

*Republic of Iraq
Ministry of Higher Education
and Scientific Research
Al-Nahrain University
College of Science*



Using Discrete Cosine Transform to Encode Approximation Wavelet Subband

*A Thesis Submitted to the College of Science, Al-Nahrain
University in Partial Fulfillment of the Requirements for
The Degree of Master of Science in Physics*

By

Sara Adnan Mahmood

(B.Sc. 2005)

Supervised by

Dr. Loay E. George

February 2008

Safar 1429

Supervisor Certification

I certify that this thesis was prepared under our supervision in the Physics Department/ College of science/ Al-Nahrain University, by **Sara Adnan Mahmood** as partial fulfillment of requirements for the degree of Master of Science in Physics Science.

Signature: 

Name : **Dr. Loay E. George**

Title : **Assistant Professor**

Date : / / **2008**

In view of the available recommendations, I forward this thesis for debate by the examination committee.

Signature: 

Name : **Dr. Ahmad K. Ahmad**

Title : **Head of the department of Physics,
Al-Nahrain University.**

Date : / / **2008**

Certification of the Examination Committee

We certify that we have read this thesis entitled "*Using Discrete Cosine Transform to Encode Approximation Wavelet Subband*" and as an Examining Committee, examined the student *Sara Adnan Mahmood* in its contents and in what is related with it, and in opinion, it has adequate standard of a thesis of the degree of Master of Science in Physics.

Signature:

Name: **Dr. Ayad A. Al-Ani**

Title: **Professor**

Date: / / 2008

(Chairman)

Signature:

Name: **Dr. Bushra K. Al-Abudi**

Title: **Assistant Professor**

Date: / / 2008

(Member)

Signature:

Name: **Dr. Ban N. Thanon**

Title: **Lecturer**

Date: / / 2008

(Member)

Signature:

Name: **Dr. Loay E. George**

Title: **Assistant Professor**

Date: / / 2008

(Supervisor)

Signature:

Name: **Dr. LAITH ADBUL AZIZ AL-ANI**

Title: **Dean of College of Science**

Date: / / 2008

Dedication

**To
My Family
and**

**For all those who burdened themselves
to fathom the deep truth for the sake
of enlightening humans path
and relieve their pain.**

Sara...



ACKNOWLEDGMENTS

I would like to express my gratitude to the following people:

*To **Dr. Loay E. George** - For his guidance, valuable advice, generosity and constant help throughout the course of this work*

To Bashar Jasim - For his help and support

To my husband - For his patience

To my family - For their encouragement

To all my friends - For their joy and laughter

Abstract

Various compression methods have been proposed to achieve high compression ratios and high image qualities in low computation time relatively. In this research work a combined transform coding scheme was proposed, the adopted system utilize both Discrete Cosine Transform and Wavelet transform. The advantages of both transforms were taken into consideration to encode the image. First, the system transform the color components of the image from (RGB) to (YUV), the U and V bands are downsampled due to their poor spatial resolution, and then the wavelet transform is applied on each color band separately. Some spatial coding steps are applied on detail coefficients (like hierarchal uniform quantization, Run Length Encoding (RLE), shift coding) to gain more compression. The approximate coefficients are coded by using (DCT), uniform quantization, shift coding.

Also, some analysis tests were done to study the performance of the established system, and the effects of the involved coding parameters on the system were investigated. The test results indicated that the proposed scheme give high compression ratio and good fidelity measures (MSE and PSNR). Moreover, the proposed scheme was found need 0.7 second to compress color images of size (256x256) without making significant degradation in image quality.

List of Abbreviations

<i>Abbreviation</i>	<i>Original</i>
2-D	Two Dimensions
BitRate	Bit rate
BlkLen	Block Length
CMY	Cyan Magenta and Yellow
CPU	Central Processing Unit
CR	Compression Ratio
CRT	Cathode Ray Tube
dB	deci Bell
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
DSP	Digital Signal Processing
DWT	Discrete Wavelet Transform
FBI	Federal Bureau of Investigation
FHWT	Forward Haar Wavelet Transform
HH	High-High Band Coefficient
HL	High-Low Band Coefficient
HSL	Hue Saturation Lightness
HVS	Human Visual System
HWT	Haar Wavelet Transform
IDCT	Inverse Discrete Cosine Transform
IHWT	Inverse Haar Wavelet Transform
ISO	International Standard Organization
IWT	Inverse Wavelet Transform
JBIG	Joint Bi-level Image Group
JPEG	Joint Photographic Expert Group
LH	Low-High Band Coefficient
LL	Low-Low Band Coefficient
LS	Lifting Scheme

LTW	Lower-Tree Wavelet
LZW	Lemple-Ziv Welch Coding
MAE	Mean-Absolute Error
MSE	Mean-Squared Error
NTSC	National Television System Committee
PDF	Probability Density Function
PSNR	Peak Signal-to-Noise Ratio
RGB	Red, Green, Blue
RLD	Run Length Decoding
RLE	Run Length Encoding
RMSE	Root Mean Square Error
SPIHT	Set Partitioning In Hierarchical Trees
SQ	Scalar Quantization
STFT	Short Time Fourier Transform
TC	Transform Coding
TIFF	Tagged Image File Format
VQ	Vector Quantization

List of Contents

Chapter One: General Introduction

1.1 The Need for Compression	1
1.2 Image Compression	1
1.3 Wavelet Based Compression Scheme	3
1.4 Historical Background of Image Compression.....	4
1.5 Related Work	6
1.6 Aim of Thesis.....	10
1.7 Thesis Layout	10

Chapter Two: Theoretical Background

2.1 Introduction	11
2.2 Classification of Compression Techniques	11
2.3 Lossless Compression Methods	13
2.3.1 Huffman Coding	13
2.3.2 Arithmetic Coding	14
2.3.3 S-Shift Coding	14
2.3.4 Run Length Encoding	15
2.4 Lossy Compression	16
2.4.1 Predictive Coding	17
2.4.2 Quantization	17
2.4.3 Transform Coding (TC).....	19
2.5 Image Redundancy	19
2.6 The Color Space	21
2.7 Traditional Image Transform	25
2.7.1 Fourier Transform (FT)	26
2.7.2 Short-Time Fourier Transform (STFT)	26

2.7.3 The Discrete Cosine Transform (DCT)	27
2.8 Wavelet Transform (WT)	30
2.9 Types of Wavelet Image Decompositions	34
2.9.1 Line Decomposition	35
2.9.2 Quincunx Decomposition	36
2.9.3 Pyramid Decomposition	37
2.9.4 Standard Decomposition	38
2.9.5 Full Wavelet Decomposition (Packet)	39
2.10 Haar Wavelet Transform (HWT)	40
2.11 The Integer Wavelet Transform (Tap5/3).....	41
2.12 Float Wavelet Transform (Tap9/7)	42
2.13 Fidelity Criteria.....	42

Chapter Three: The Proposed Image Compression System

3.1 Introduction	45
3.2 System Model	45
3.3 Encoding Unit	47
3.3.1 Image Loading	47
3.3.2 Color Transform	47
3.3.3 Wavelet Transform	48
3.3.4 Quantization of Detail Coefficients	50
3.3.5 Run Length Encoding	52
3.3.6 Discrete Cosine Transform	54
3.3.7 DCT Coefficients Quantization	55
3.3.8 Mapping to Positive	57
3.3.9 Zigzag and S-Shift Optimizer	58
3.3.10 Shift Encoder	61
3.3.11 Shift Coding the Detail Coefficients	62
3.3.12 Shift Encoding the Detail Subband	64

3.4 Decoding Unit	65
3.4.1 Shift Decoder and DeZigzag	65
3.4.2 DeQuantization	67
3.4.3 Inverse Discrete Cosine Transform (IDCT)	68
3.4.4 Run Length Decoding	69
3.4.5 The Dequantization of Detail Coefficients	71
3.4.6 Inverse Wavelet Transform	72
3.4.7 Up Sampling	73
3.4.8 YUV to RGB Transform	74

Chapter Four: Tests and Results

4.1 Introduction	75
4.2 Image Test Material	75
4.3 Image Compression Performance Tests	76
4.4 Tables Notes.....	100

Chapter Five: Conclusions and Future Works

5.1 Conclusions	101
5.2 Future Works	102

(DCT)

((RGB) , ,)

((YUV))

V U

)

(RLE)

(hierarchal uniform quantization)

((shift coding)

(uniform quantization)

)

(PSNR)

(MSE)

256x256

0.7

Chapter One

General Introduction

1.1 The Need for Compression

The rapid developments in Internet and multimedia technologies have led to exponential growth in the amount of information that is handled by computers over the past decades. This information requires large amount of storage space and transmission bandwidth that the current technology is unable to handle technically and economically. One of the possible solutions to this problem is to compress the information so that the storage space and transmission time can be reduced.

A common characteristic that can be found in most images is that they contain redundant information. This redundant information can be classified as [Cro01]:

1. Spatial redundancy-correlation between neighboring pixels values
2. Spectral redundancy -correlation between different spectral bands

This thesis focuses on the compression of still images.

1.2 Image Compression

Compression is desired in many occasions. Usually, image compression algorithms can bring about 10 fold of reduction in the file size without making significant effect on the visual quality. On the other hand, the compression is at the price of extensive processing and CPU power [Zhe05].

In general, image compression techniques can be broadly classified into:

1. Lossless compression

2. Lossy compression

In lossless compression, every bit of information is preserved during the decomposition process. The reconstructed image after lossless compression is an exact replica of the original one. Such scheme only achieves a modest compression rate. It is used in applications where no loss of image data is permitted. Lossless schemes only achieve a modest compression rates. It is used in applications where no loss of image data can be compromised.

In lossy compression, a perfect reconstruction of the image is sacrificed by the elimination of some amount of redundancies in the image to achieve higher compression ratio. However, no visible loss of information should be perceived under normal viewing conditions [Cro01, Sah01].

In this thesis, the type of image compression scheme that will be focused is the lossy compression scheme. The typical lossy image encoder system consists of three main operations, as shown in Figure (1.1).

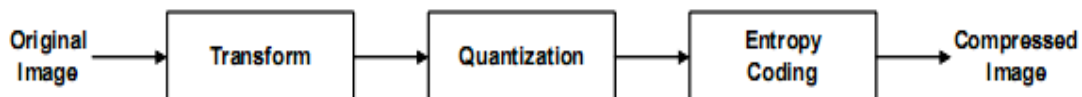


Fig (1.1) Encoder of Image Lossy Compression Scheme

The transform operation is a linear transform that aims to reduce the entropy of the coefficients of the image. This operation is reversible and does not cause any loss of information to the image. An example of such a transform operation is the discrete cosine transforms (DCT) or wavelet-based subband coding.

The quantization operation is a lossy operation that maps a large set of input data to a smaller set of output data, attempting to remove redundancies in the image. This process is irreversible and it introduces distortion. The two main

types of quantization are scalar quantization and vector quantization.

The entropy coding operation, which is a lossless operation, compresses the image data further without making loss in information. The main idea here is to reduce the average number of bits to represent an alphabet by assigning a shorter codeword to the most probable alphabet and a longer codeword to the least probable symbol. Some common examples of entropy coding are Run-Length Encoding, Huffman Encoding and Arithmetic Encoding.

To reconstruct the image, the three main of the encoder, figure 1.1, should be reversed (as shown in Figure 1.2). At each stage, an inverse operation will be carried out.

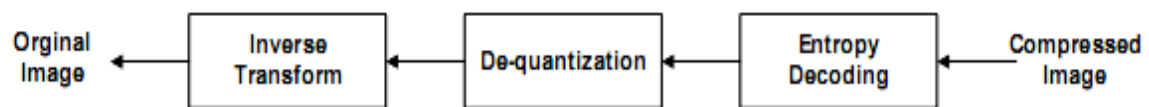


Fig (1.2) Decoder of Image Lossy Compression Scheme

1.3 Wavelet Based Compression scheme

In recent years, many studies have been made on wavelets. Wavelets have been used in various fields (like biomedical applications, wireless communications, computer graphics or turbulence) as know key solutions to various relevant problems. Image compression is one of the most visible applications of wavelets.

Wavelets are functions defined over a finite interval. The basic idea of the wavelet transform is to represent an arbitrary function $f(x)$ as a linear combination of a set of such wavelets or basis functions. These basis functions are obtained from a single prototype wavelet called the mother wavelet by

dilations (scaling) and translations (shifts). The purpose of wavelet transform is to change the data from time-space domain to time-frequency domain which makes better compression results. The simplest form of wavelets is the Haar wavelet function.

As discussed earlier, for lossy image compression, loss of some information is acceptable. Vector quantization requires many computational resources for large vectors; fractal compression is time consuming for coding; predictive coding has inferior compression ratio and worse reconstructed image quality than those of transform based coding. So, transform based compression methods are generally the best for image compression [Sah01].

1.4 Historical Background of Image Compression

Image compression algorithms have been the subject of research both in academia and industry for many years. Today, while significantly improved algorithms have been achieved and compression performance becomes better, there is still room for new technologies. The first widely adopted international image compression standard was JPEG which was introduced in the late eighties [PeMi94]. JPEG is based on DCT followed by entropy coding based on either Huffman coding or binary arithmetic coding [Sch98]. It has been widely used whether in printing industry or Internet applications. For example most images transmitted through the internet are JPEG compressed.

JPEG is intended for continuous tone images of more than one bit depth. Algorithms for binary images work in a different way, JBIG-1 and JBIG-2 are the standards covering this area.

JPEG and JBIG are part of other standards, such as facsimile transmission standards [ITU93], the FlashPix file format [HaHo96], the TIFF file format [Ado92], and page description languages like PDF.

In recent years researchers have been using the discrete wavelet transform in compression systems. In 1983 Burt and Anderson [BuAd83] were the first to introduce multiresolutional analysis in image compression. While their approach seemed counter intuitive at the first glance, given that it increased the number of samples to be coded, their results were promising. Daubechies [Dau88] had studied the discrete wavelet transform and made it a popular tool in the scientific community. Mallat [Mal89] was the first who had pointed out to the connection between multiresolutional analysis and the wavelet transform. Some of the first papers on wavelet image compression presented excellent compression performance results and gave a lot of intuition behind the use of the wavelet transform in image compression. A number of researchers have described the same principles of wavelet image compression by looking at it from a system perspective, using filter banks, and subband decomposition, and refer to wavelet coding as subband coding. Subband coding and wavelet coding essentially refer to the same system, their description to the system is slightly different. In subband coding the emphasis is in the frequency domain unlike wavelet coding where the emphasis is in the space domain [VeKo95].

Numerous organizations have been using wavelet compression algorithms as their own, internal compression standards. An example is the FBI where there was a need for storing large data-bases of finger-prints and JPEG did not satisfy their requirements. More recently there was a decision taken by the ISO to standardize a wavelet coder in JPEG2000. Until recently all proposed wavelet coders would require buffering the whole images, computing the wavelet transform in a frame buffer, and then applying a quantization on the wavelet coefficients and entropy encoding the generated indexes. Wavelet coders could indeed perform very well, but their complexity was well above the complexity of the current JPEG standard [ChOr98].

1.5 Related Works

1. Bethel (1997) [Bet97], has investigated the whole area of image compression and optimized some of the techniques that applicable to produce the best possible compressor. He refer that increasing the complexity of the compression method could improve the rate distortion performance of the method, but it does so at the expense of the speed of the compressor. It was also found that the transform stage of the compressor is not as important as an effective source coding stage. He showed that the increase in computational complexity yields better rate distortion performance, and hence a balance has to be reached between the speed of a system and its performance. Also, He showed that the transform stage of a compressor is not as important as the compression methods used on the transformed data. Also, He indicated that wavelet methods do not perform better than DCTs method in terms of MSE when the compression is done correctly.
2. Chrysafis (2000) [Chr00], had studied some wavelet based image coders. The refer that wavelet coders apart from offering superior compression ratios have also very useful features, (e.g., resolution scalability; he allow decoding a given image at a number of different resolutions depending on the application). He started by presenting in a simple manner a collection of tools and techniques to apply wavelet filtering in images, ranging from boundary extension to fast implementations, and continued by exploiting the use of rate distortion theory to achieve very high compression ratios for a wavelet coder. His results have been reported among the best in the literature. He applied rate distortion theory on a per coefficient basis combining a theoretical analysis with online probability estimation. After presenting the rate

distortion algorithm he focused on techniques to reduce the complexity of generic wavelet coders. One of the main contributions of this work is the ability to compress an image with a wavelet coder without need to buffer the complete image. The memory requirements of the proposed approach are orders of magnitude lower than other algorithms proposed up to date, which would require buffering of the entire image. This limited low memory implementation was the key in enabling widespread commercial use of wavelet image coding and has been incorporated in the informative part of the upcoming JPEG2000 standard.

3. Xiao (2001) [Xia01], his project studied image compression using wavelet transform. As a necessary background, the basic concepts of graphical image storage and currently used compression algorithm are discussed. The mathematical properties of several types of wavelets, (including Haar, Daubechies, and biorthogonal wavelets) are covered. He analyzed the compression results to compare the wavelet types, and found that the biorthogonal wavelet gave a good compression results than the other types.
4. Tan (2001) [Tan01], had studied and implemented some of the operations used in a lossy compression scheme to compress two-dimensional images. Basically he implemented a coding scheme consists of three main operations: the transform, quantization and entropy encoding operations. Higher compression ratios can be achieved at the expense of the quality of the image by quantizing the image coarsely or by using a more sophisticated entropy encoder such as Huffman encoder. However, the computation time was increased when Huffman encoder was used. One way of overcoming the long computation time is to use the RLE before Huffman. Such a

combination has shown an improvement over the Huffman encoder for all types of implemented quantizers.

5. Grgic et al (2001) [Grg01], they examined a set of wavelet functions (wavelets) for implementation in a still image compression system, and they try to highlight the benefit of wavelet transform relating to today's methods. The paper discusses the important features of wavelet transform in compressing still images, including the extent to which the quality of image is degraded by the process of wavelet compression and decompression. Image quality was measured objectively (using peak signal-to-noise ratio or picture quality scale) and subjectively (using perceived image quality). The effects of different wavelet functions, image contents and compression ratios were assessed. A comparison with a discrete-cosine-transform-based compression system is given. They found that wavelet-based image compression prefers smooth functions of relatively short length.

6. Nanda (2003)[Nan03], had presented a novel method that incorporates pre and post image processing for increasing the effective amount of compression achieved on an image. Pre-processing is performed by low pass filtering the image before compression, resulting in suppression of high frequencies, and hence allowing for fewer coefficients to represent the image. The post-processing is performed after the compression of the filtered image, basically it is an image restoration algorithm based on the gradient method that uses the information of the filter and the compression process to reverse the effects of filtering and compression. The method was tested for block-based compression on a number of images. Research results indicated that the introduced method yields considerable gain both from a subjective and an objective viewpoint, especially at high compression ratios. The results demonstrate that the

idea of low pass filtering the image before compression, and then applying the image restoration algorithm on the reconstructed blurred image, does in fact an enhancement in the compression of images.

7. Ibraheem (2004) [Ibr04], had introduced an image compression system based on wavelet transform coding. In his project, some additional coding techniques were implemented such as Differential Pulse Code Modulation (DPCM), and S-shift coding to improve the compression performance. His test results indicated that the degradation in image quality was kept as minimum as possible. He got minimum MSE with good PSNR (from 24 to 40), and Cr (from 2 to 5).
8. Kotteri (2004) [Kot04], had designed and implemented image compression by using biorthogonal tap9/7 DWT, and applied uniform scalar quantization on wavelet coefficient. He utilized the fidelity measure (MSE and PSNR) to asses the quality of the compressed image. To avoid wasting in computation he improved the efficiency of the filter bank.
9. Oliver & Malumbres (2006) [OlMa06], in their paper a very fast variation of the Lower-Tree Wavelet (LTW) image encoder was presented. LTW is a fast non-embedded encoder with state-of-the-art compression efficiency; it employs a tree structure as a fast method for coding wavelet coefficients. It is faster than other encoders like SPIHT or JPEG 2000. The alternative Huffman-based encoder presented in this paper serves to largely reduce the execution time, at the expense of loss in coding efficiency. Experimental results show that this encoder is more efficient than other very fast wavelet encoders, like the recently proposed progress (which is surpassed in up to 0.5 dB), and faster than them (from

4 to 9 times in coding). Compared with the JPEG 2000 reference software, the encoder is from 18 to 38 times faster, while PSNR is similar at low bit-rates, and about 0.5 lower at high bit-rates.

1.6 Aim of Thesis

In this research work a combined transform coding scheme is proposed, this scheme utilizes both Wavelet transform and DCT, the first will be considered as the main core engine of the compression system, while the second transform will be used as secondary coding tool. The advantages of both transforms will be taken into consideration to encode the portioned regions of the image. The requirement of low complexity is taken into consideration in designing the scheme of the proposed system.

1.7 Thesis Layout

Beside to this chapter the remaining part of thesis consists of the following four chapters:

- **Chapter Two:** this chapter discusses the image compression techniques in details, including Wavelet transform, and discrete cosine transform.
- **Chapter Three:** this chapter includes in details the designed and implemented image compression models. All the developed algorithms that used in this research work are presented.
- **Chapter Four:** this chapter contains the result of the conducted tests on some samples of images that used as test material in this work. The used performance criteria are the fidelity measures (MSE, PSNR) beside the compression ratio.
- **Chapter Five:** this chapter includes the derived conclusions and some suggestions for future works.

Chapter Two

Theoretical Background

2.1 Introduction

Image compression technique is especially important for compact storing and fast communication of the image. To reach this target a lot of research work appears yearly in the literature. Some of them involved with the use of sub-block DCT coding and others include DWT coding. Although the sub-block DCT method is efficient for the image compression a block artifact sometimes appears when the low average bit rate is employed. This is the inherent shortcoming of the sub-block DCT method. In order to avoid this defect the DWT method is exploited. But the DWT itself is not so efficient unlike DCT; the resulting reconstructed image is not necessarily good for a high picture quality coding. In this work, a combine scheme makes use of both DWT and DCT.

In this chapter, some relevant concepts to image compression discipline given. Also some of the common image compression techniques are illustrated.

2.2 Classification of Compression Techniques

Compression process takes an input \mathbf{X} and generates a representation \mathbf{X}_c that hopefully requires fewer storage sizes. While the reconstruction algorithm operates on the compressed representation \mathbf{X}_c to generate the reconstruction Y .

Based on the difference between the original and reconstructed versions, data compression schemes can be divided into two broad classes, see figure (2.1). The first is lossless compression, at which Y is identical to

X. While the second is lossy compression, which generally provides much higher compression than lossless compression but makes Y not exactly as X [Add00].

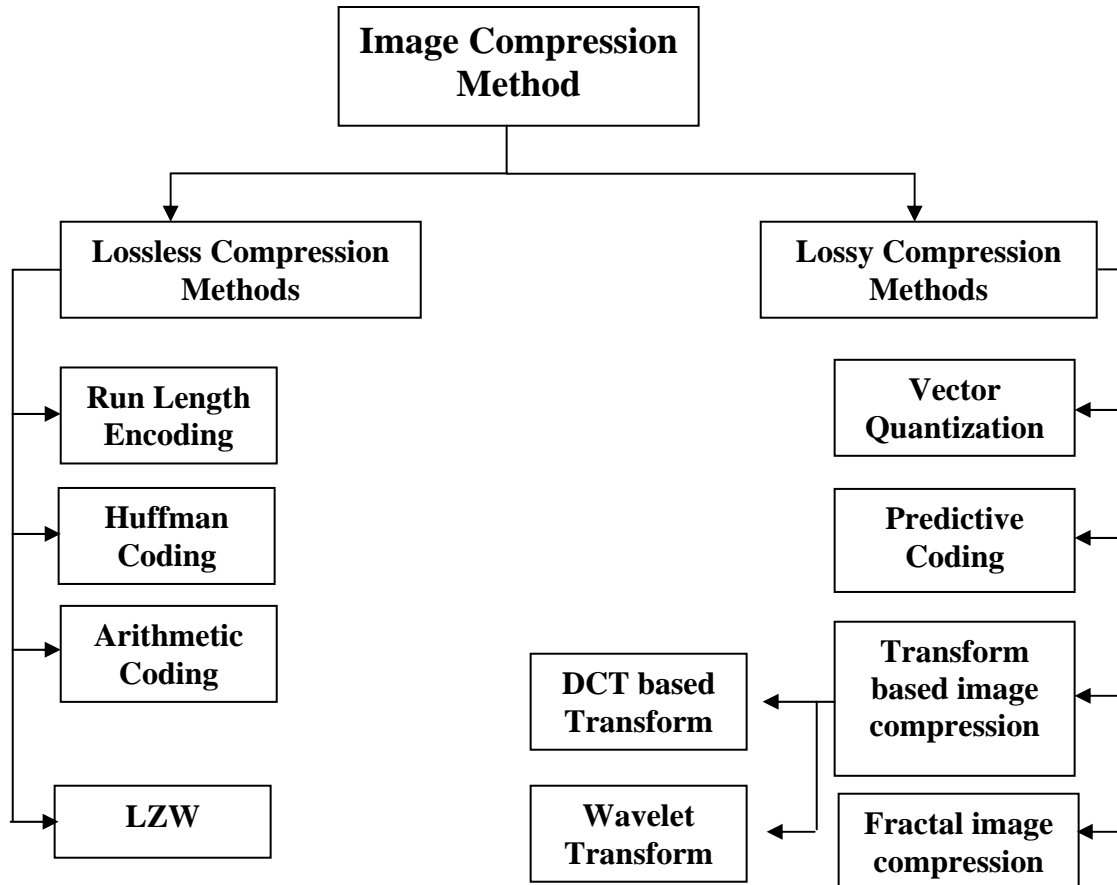


Fig (2.1) The most popular image compression methods [Add00]

There are two primary types of image compression methods. Some compression methods preserve the data, while the other allows some loss of data. Therefore, image compression techniques are classified into two categories [Umb98]:

1. Lossless compression methods.
2. Lossy compression methods.

2.3 Lossless Compression Methods

Lossless compression techniques provide the guarantee that no pixel difference will occur between the original and decompressed image, in other words lossless schemes result in reconstructed data that exactly matches the original. It is generally used for applications that cannot allow any difference between the original and reconstructed data. The most popular lossless compression methods are: *Huffman coding, Arithmetic coding, S-shift coding, and Run length coding* [Avc02].

2.3.1 Huffman Coding

Huffman compression [Huf62] is designed to reduce the bits representation for a data source, such that its bit rate should be close to the entropy of the data source. The entropy is determined by using the following equation:

$$E = \sum_x P(x) \log_2(P(x)) \dots \dots \dots (2.1)$$

Where, $P(x)$ is the probability density distribution of symbol x .

Huffman encoder represents the common data symbols with short codewords and the rare data symbols with long codewords. The average effect of this method is to reduce the redundancy of each compressed symbol to a minimum. Huffman encoder determines the codewords (compression symbols) by forming a data tree from the original data symbols and their associated probabilities.

2.3.2 Arithmetic Coding

Arithmetic coding [Nel96] was introduced in 1976. It works by treating a stream of data symbols as a whole and does not replace individual data symbols with compressed versions. The coder is always implemented in binary. An arithmetic coder takes upper and lower limits, and defines a range between these upper and lower limits to be equivalent to a symbol with the probability of [0, 1]. Symbols are encoded by modifying the range of the arithmetic coder. Implementations of arithmetic coding are very complicated and need to overcome the precision problem. The other disadvantage is its decoding speed.

2.3.3 S-Shift Coding

The idea of this method is to encode a set of numbers by codewords whose bit length is less than the bit length required to represent the maximum value of the set. While the numbers whose values are large may encode using sequence of codewords, each sequence may consist of many short-length codewords, the values of the codewords are determined according to the following equation [Gon00]:

$$X = nW_m + W_r \dots \dots \dots (2.2)$$

Where,

X is the number to be coded.

n is the number of codewords that used to encode the number X.

W_m is the lowest value which cannot be coded by using a single codeword.

W_r is the value of the last word used to encode X.

The values of W_m , W_r , and n are determined using the following equations:

$$W_m = 2^b - 1 \dots\dots\dots (2.3)$$

$$W_r = X \bmod W_m \dots\dots\dots (2.4)$$

$$n = X \operatorname{div} W_m \dots\dots\dots (2.5)$$

Where b is the number of bits used to represent each single s -codeword.

The performance of Huffman coding and shift coding are better when the histogram of the sequence of coded numbers is highly peaked. The performance of shift coding is better than Huffman and arithmetic coding when the histogram has long tails [Ibr04, Gon00].

2.3.4 Run Length Encoding

Run length encoding is a lossless algorithm that only offers decent compression ratios in specific types of data. It is based on the idea of encoding a consecutive occurrence of the same symbol. This is achieved by replacing a series of repeated symbols with a count and the symbol.

For example, if the data contains large number of consecutive zeros, the data size can be greatly reduced using RLE. On the other hand, if the content of the data is random, then this encoding technique might increase the data size.

To allow RLE to be used in encoding the consecutive runs of other characters, a control symbol could be added to the RLE pair as shown in Figure (2.2) the following example illustrates the use of this control symbol.

Original data : {12 13 13 13 13 14 14 14 14 0 0 0 0 9 9 9 0}

Encoded data: 12 ! 4 13 ! 4 14 ! 5 0 9 9 9 0

CTRL (1 byte)	Count (1 byte)	Symbol (1 byte)
------------------	-------------------	--------------------

Figure (2.2) RLE symbols (2)

For this type of RLE encoder, the compression takes place when the number of consecutive characters is more than three. Such characters will be replaced with the RLE symbol (shown in Figure 2.2). If the number of consecutive characters is less than two, then the characters will not be encoded as runs of numbers. In the decoding process, the encoded data will be expanded according to the count of characters in the encoded-symbols. It is important to know that there are many different run-length encoding schemes. The above example has just been used to demonstrate the basic principle of RLE encoding. Sometimes the implementation of RLE is adapted to the type of data that are being compressed. This algorithm is very easy to implement and does not require much CPU horsepower. RLE compression is only efficient with files that contain lots of repetitive data. These can be text files if they contain lots of spaces for indenting, the line-art images that contain large areas of constant colors are far more suitable. Computer generated color images (e.g. architectural drawings) can also give fair compression ratios [Sch01, Dat01].

2.4 Lossy Compression

Lossy compression involves elimination of "less important" information with respect to the "goodness" of the reconstructed data in the process of compression.

The most popular lossy compression methods are:

1. Predictive Coding.

2. Quantization.
3. Transform Coding.
4. Sub-Band coding.
5. Fractal Image Compression.

Similar to the DCT based image compression techniques; the image compression techniques based on wavelet transform are lossy. This involves eliminating the wavelet transform coefficients which are "less important" in contributing to the image's appearance and keeping the rest. More specifically, the coefficients with higher magnitudes are more important than coefficients with lower values [Wan00].

2.4.1 Predictive Coding

Predictive coding has been used extensively in image compression. Predictive image coding algorithms are used primarily to exploit the correlation between adjacent pixels. They predict the value of a given pixel using the values of the surrounding pixels. Due to the correlation property among adjacent pixels in image, the use of a predictor can reduce the amount of bits needed to represent image. Predictive coding can be implemented as a lossy image compression technique, but is not as competitive as transform coding techniques, because predictive techniques have inferior compression ratios and lead to worse reconstructed image quality when they compared with those produced by transform coding [Say00].

2.4.2 Quantization

Quantization involved in image processing. Quantization techniques generally compress a range of values to a single quantum value. By reducing the number of discrete symbols in a given stream, the stream

becomes more compressible. There are two main types of quantization methods:

1. Scalar Quantization (SQ)

It is used to reduce the number of bits needed to store each individual of a set of real numbers. This can be done by dividing each number by a quantization factor and approximate it to the nearest integer before it is stored. To retrieve the number again, the stored quantized integer (called Quantization Index) should multiplied by quantization factor. Quantization process is not lossless, because the retrieved number doesn't have the exact value of the original; the degree of closeness depends on the value of the quantization factor [Aal96].

2. Vector Quantization (VQ)

Scalar quantization can be improved using **vector quantization (VQ)**; Vector Quantization (VQ) is also a lossy compression method. It uses a codebook containing pixel patterns with corresponding indexes, each represent one of them. The main idea of VQ is to represent arrays of pixels by an index in the codebook. In this way, compression is achieved because the size of the index (in bits) is usually a small fraction of that of the block of pixels.

The main advantages of VQ are the simplicity of its idea and the possible efficient implementation of the decoder. Moreover, VQ is theoretically an efficient method for image compression, and superior performance is gained for large vectors. However, in order to use large vectors, VQ becomes complex and requires many computational resources (e.g. memory, computations per pixel) in order to efficiently construct and search a codebook. More research on reducing this complexity needs to be done in order to make VQ a practical image compression method with superior quality [Say00].

2.4.3 Transform Coding (TC)

Although the prediction is a prediction-residual coding method can be made two dimensional, it is very hard to fully exploit the two dimensional correlation using only prediction. A better way is to transform the pixels in the image domain into another domain, where the representation is more natural and therefore more compact. This representation should preferably be such that some coefficients give the bulk of the energy in the image, while others are very likely to be very small or zero. When the latter ones are small, it signifies that the correlation that was expected is present. To have this kind of representation there is a need to decorrelate the coefficients with a reversible transformation (at least almost reversible), while still maintaining the same amount of total energy in the basis coefficients. Various fast algorithms were developed to perform the necessary transformations like [Nis98]:

1. Fourier Transform
2. Cosine Transform
3. Wavelet Transform

2.5 Image Redundancy

The common characteristic of most images is that the neighboring pixels contain redundant information [Sah01]. Digital image data compression can be performed by removing all redundancies that existed in an image data, so it takes up less storage space and require less bandwidth to be transmitted. There are many types of redundancies, among these are the following:

- A. **Interpixel Redundancy:** Interpixel redundancy implies that the intensity value of a pixel can be predicated from its neighboring

pixel, because the values of adjacent pixels tend to be highly correlated [ShSu00]. It is the result of the fact that in most images the brightness levels do not change rapidly, but change gradually, so that the adjacent pixels tend to be relatively close to each other in value [Gon02]. This kind of redundancy can be removed by transforming the image into a state where the interpixel redundancy can be discovered and eliminated, and this kind of transformation process is called a mapping. Sometimes the interpixels redundancy is called geometric redundancy, spatial redundancy or interframe redundancy (for video and motion image).

B. Coding Redundancy: Gray levels are usually coded with equal-length binary codes, and coding redundancy may exist in such codes. For example, the 8 bit/pixel image allows 256 gray level values but sometimes the actual image may contain only 16 gray level values, then as a suboptimal coding only 4 bit/pixel is actually needed. Also, more efficient coding can be achieved if variable-length coding is employed. Variable-length coding assigns fewer bits to the gray levels with higher occurrence probabilities in an image. The average length required to represent a pixel within a compressed image using variable-length coding method, is given by [Sal98]:

$$\bar{n} = \sum_{k=0}^{L-1} n(r_k)P(r_k) \dots\dots\dots (2.6)$$

Where, r_k is the gray level of an image.

$P(r_k)$ is the probability of occurrence of r_k .

$n(r_k)$ is the number of bits used to represent each value of r_k .

K is the number of gray levels.

In variable coding the symbol with a higher occurrence probability is coded using a short codeword [ShSu00].

C. Psychovisual Redundancy: Psychovisual redundancy originates from the characteristics of the Human Visual System (HVS). The human perception is not a constant pixel oriented mechanism. This implies that every area in an image is not processed with same amount of sensitivity, and the image areas which do not contribute to the valuable visual content can be removed without a major loss in quality for the human perceiver. The elimination of these redundancies can be considered as a sort of quantization, which is an irreversible process [Fri95]. For color images, this kind of redundancy can be used to reduce the size of the chromatic components, since the human eyes are less sensitive to the variation in chromaticity than the variation in light intensity [ShSu00].

D. Spectral Redundancy: It is the correlation between different spectral bands [ShSu00].

E. Temporal Redundancy: It is the correlation between different subsequence frames in video subsequence [Haf01]. Since this thesis will focus only on still image compression, so the temporal redundancy is not taken into consideration.

2.6 The Color Space

A color space model is a method used to specify, creation and visualization. A color is usually specified by using three coordinates or parameters which describe the position of the color within the color space

being used. There are various reasons for applying a color transformation. The choice of an appropriate color space can be an important factor to determine the results of processing on a color image (quality of image segmentation, compression ratio). In practice, there is no ideal color space for all image processing applications. Some of the more common computer related color spaces are the following [FoRo98]:

- 1. RGB (Red, Green, and Blue):** This is an additive color system. It is the most frequently used color space for image processing. Since color camera, scanners and systems that use a CRT to display images (like computer, television, video etc) are most often provided with direct RGB signal input or output. This color space is the basic one, which is, if necessary, transformed into another color space. As shown in figure (2.3).

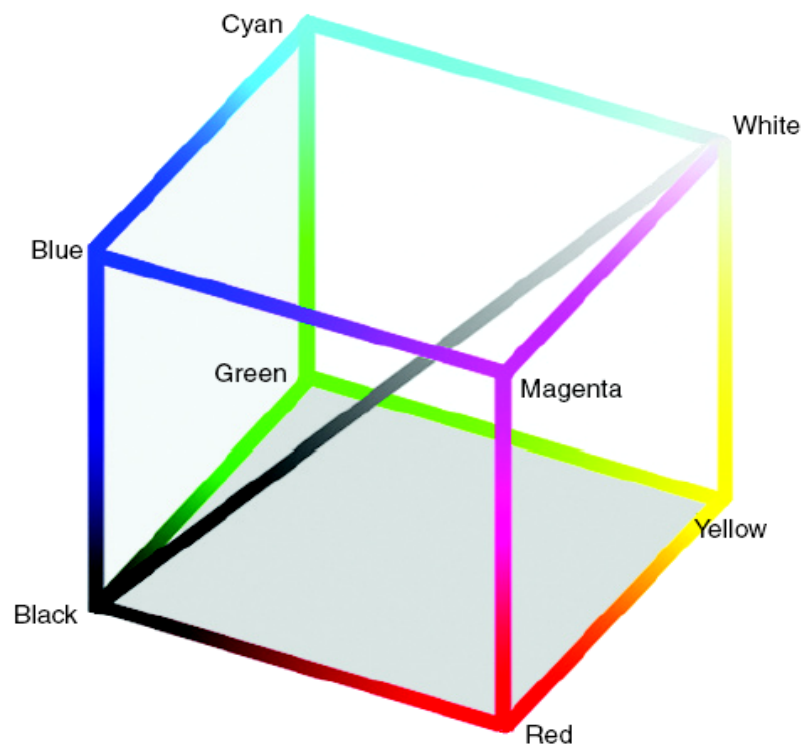


Fig (2.3) Representation of Color in RGB Color Space [Rus06]

The main advantage of RGB color space in the applications involving natural images is the existence of high correlation between its components. But, this makes RGB space unsuitable for compression [SaHo98]. All gray colors are placed on the main diagonal of this cube from black ($R=G=B=0$) to white ($R=G=B=Max$).

2. **CMY (Cyan, Magenta, and Yellow):** This is subtractive based color space. It is mainly used in printing and hard copy output. CMY is fairly easy to implement but the perfect transfer from RGB to CMY is very difficult [FoRo98].
3. **HSL (Hue, Saturation, Lightness):** It has a wealth of similar color spaces; alternative names include HIS (Intensity), HSV (Value), and HCL (Chroma/Colorfulness). Most of these color spaces are non-linearly transformed from RGB. Their advantage lies in the extremely intuitive manner of specifying color. It is very easy to select a color with desired hue and then modify it slightly by adjusting its corresponding saturation and intensity [SaHo98].

The separation of the luminance component from chrominance (color) information is stated to have advantages in different image processing applications.

4. **YUV, YIQ, YC_bC_r, YCC (Luminance-Chrominance):** These are the television transmission color spaces, sometimes known as transmission primaries. YUV and YIQ are analogue spaces for NTSC and PAL systems, respectively. While YC_bC_r is a digital standard. These color spaces separate RGB into luminance information and are

useful in compression application.

YUV color space is widely used in coding color images and color video to convert from RGB to YUV spaces, the following equations can be used: [Dun99]

$$Y = 0.299R + 0.587G + 0.114B \dots\dots\dots (2.7)$$

$$U = 0.492(B - Y) \dots\dots\dots (2.8)$$

$$V = 0.877(R - Y) \dots\dots\dots (2.9)$$

Any errors in the resolution of the luminance (Y) are more important than the errors in the chrominance (U, V) values. The luminance information can be coded using higher bandwidth than the chrominance information. Figure (2.4) presents an example of YUV components, the Y-components shows more spatial details than U & V components.

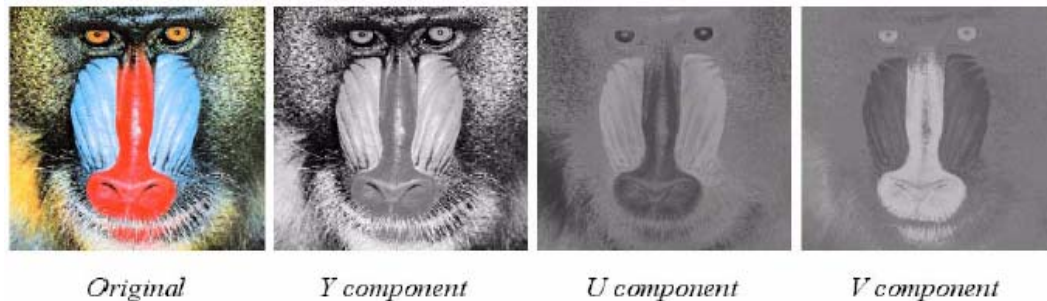


Fig (2.4) Example of YUV Space [Dun99]

YIQ color space is useful in color image segmentation and it is useful for color image coding, it is similar to YUV, except that its color space is rotated 33 degrees clockwise, so that the component I is the orange- blue axis, and Q is the purple-green axis. The equations to convert from RGB to YIQ are [Dun99]:

$$I = 0.596R - 0.275G - 0.321B \dots\dots\dots (2.10)$$

$$Q = 0.212R - 0.523G + 0.311B \dots\dots\dots (2.11)$$

YCbCr color space was used in compression schemes of both video sequences and still images. For example JPEG image compression employs YCbCr spaces. Greater compression is achieved when the spatial resolution for the color components C_b and C_r are reduced, and then coarser quantization is applied on all components for (Y, C_b, C_r). YCbCr color space was used by Kodak for encoding images on photo CD system [SaHo98]. The linear transform from RGB to YCbCr generates one luminance space Y and two chrominance (C_r and C_b) spaces:

$$C_r = \frac{B - Y}{2} + 0.5 \dots\dots\dots (2.12)$$

$$C_b = \frac{R - Y}{1.6} + 0.5 \dots\dots\dots (2.13)$$

2.7 Traditional Image Transforms

Transform is a powerful tool in many DSP application areas. It can effectively be used to serve as an effective approach to image compression. An image can be compressed by transforming its pixels (which are correlated) to a representation where they are decorrelated. Compression is achieved if the new values are smaller, on average, than the original ones. Lossy compression can be achieved by quantizing the transformed values. The decoder inputs the transformed values from the compressed stream and reconstructs the (precise or approximate) original data by applying the opposite transform. The term decorrelated means that the transformed values are independent of one another. [Sal02]

2.7.1 Fourier Transform (FT)

In Fourier transform the sinusoids are the basis functions. Such functions have infinite energy across the domains, and it is valuable in analyzing time invariant or stationary phenomena. It can be computed using the following equation:

$$F(\omega) = \int_{-\infty}^{\infty} \exp(-j\omega t) f(t) dt \dots\dots\dots (2.14)$$

A sudden change in the input signal causes significant changes in the frequency components throughout the entire duration of the signal. Thus, information about one instant of a signal cannot be obtained. Therefore this transformation is not suitable for non-stationary, time-varying phenomena whose frequency content changes with time [Gra95, Pol01]

2.7.2 Short-Time Fourier Transform

To overcome the above mentioned limitation of Fourier transform, a window-version of Fourier transform known as Short Time Fourier Transform (STFT) was developed. In STFT, the non-stationary signal is divided into small segments, where each segment of the signal is assumed to be stationary. Then STFT could be applied on these segments using the following formula:

$$STFT(t, \omega) = \int_{-\infty}^{\infty} \exp(-j\omega s) f(s) g(s - t) ds \dots\dots\dots (2.15)$$

But here a resolution problem appears. Once the size Δs of the STFT window is chosen, the time-frequency resolution is fixed for the entire time-frequency plane, Moreover, the resolution in time Δt and frequency Δf can

not be chosen arbitrarily small at the same time, because their product is lower bounded by the Heisenberg inequality [Hub96]:

$$\Delta t \Delta f \geq \frac{1}{4\pi} \dots\dots\dots (2.16)$$

This inequality means that a trade off between time resolution and frequency resolution should be done. That is, if a good frequency resolution is required then a poor time resolution should settle. Likewise, if a good time resolution is needed then poor frequency resolution is settled.

The wavelet transform have been developed independently in applied mathematics and signal processing. It is gradually substituting other transforms in some signal processing applications. The wavelet transform was substituted instead of STFT (which was extensively used in speech signal processing) and discrete cosine transform (DCT) (which was widely used for image compression) due to its better resolution properties and high compression capabilities [Gra95, Pol01].

2.7.3 The Discrete Cosine Transform (DCT)

The DCT is a technique for converting a signal into elementary frequency components. The discrete cosine transform (DCT) was first applied to image compression in the work by Ahmed, Natarajan, and Rao [Str99]. It is a popular transform used by the JPEG (Joint Photographic Experts Group) image compression standard for lossy compression of images. Since it is used so frequently, DCT is often referred to in the literature as JPEG-DCT, which indicated that DCT is used in JPEG. JPEG-DCT is a transform coding method consists of four steps. The source image is first partitioned into sub-blocks of size 8X8 pixels in dimension. Then,

each block is transformed from spatial domain to frequency domain using a 2-D DCT basis function. The resulting frequency coefficients are quantized and finally output to a lossless entropy coder. DCT is an efficient image compression method since it can decorrelate pixels in the image (because the cosine bases are orthogonal) and compact most of the image energy into a few transformed coefficients. Moreover, DCT coefficients can be quantized according to some human visual characteristics. Therefore, the JPEG image file format is very efficient. This makes it very popular, especially in the World Wide Web. However, in JPEG2000 the wavelet transform is used instead of DCT due to its better compression performance [Cab02, Tru99].

The forward DCT formula is given by [Sal02]:

$$C_{ij} = \frac{2}{\sqrt{mn}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} P_{xy} \cos\left(\frac{(2y+1)j\pi}{2m}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right) \dots \quad (2.17)$$

Where

$$C_f = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } f = 0 \\ 1 & \text{otherwise} \end{cases} \dots \dots \dots (2.18)$$

C_{ij} represents the transform coefficients

$0 \leq i < n$ and $0 \leq j < m$ are the indexes of the transform coefficients

P_{xy} is the value of the pixel (x, y)

n is the image width (number of columns).

m is the image height (number of rows).

To turn the image back to its original domain the inverse transform must be applied, the inverse DCT is given by:

$$P'_{xy} = \frac{2}{\sqrt{mn}} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} C_i C_j C_{ij} \cos\left(\frac{(2x+1)i\pi}{2n}\right) \cos\left(\frac{(2y+1)j\pi}{2m}\right) \dots\dots(2.19)$$

Where

P' is the reconstructed image.

$0 \leq x < n$, $0 \leq y < m$ are the image pixel's coordinates

C is the transformed image.

n , m is the number of pixels.

The main advantages of JPEG are its simplicity, satisfactory performance, and the availability of a dedicated hardware for implementation. However, because the input image is blocked, then correlation across the block boundaries cannot be eliminated. This results in noticeable and annoying “blocking artifacts” particularly at low bit rates as shown in figure (2.5) [Sah01].



a. Original



b. reconstructed

Fig (2.5) The original and reconstructed Lena image

2.8 Wavelet Transform (WT)

During the last decade, the wavelet transform has gained the attention of many researchers in the field of image compression. What makes this transform a better choice than Fourier transform is its ability to localize in frequency and time simultaneously. This is extremely useful when analyzing time-varying or non-stationary phenomena that are commonly found in images [Bur98]. In images, fine information content is generally found in the high frequencies whereas the coarse information content exists in the low frequencies. The multi-resolution capability of wavelet transform is used to decompose the image into multiple frequency bands. The wavelet transform has its roots in Fourier transform.

The fundamental idea behind wavelets is to analyze the signal at different scales or resolutions, which is called multiresolution. Wavelets are functions used to localize a given signal in both space and scaling domains. A family of wavelets can be constructed from a mother wavelet. Compared to Windowed Fourier analysis, a mother wavelet is stretched or compressed to change the size of the window. In this way, big wavelets give an approximate image of the signal, while smaller and smaller wavelets zoom in on details. Therefore, wavelets automatically adapt to both the high-frequency and the low-frequency components of a signal by different sizes of windows. Any small change in the wavelet representation produces a correspondingly small change in the original signal, which means local mistakes will not influence the entire transform. The wavelet transform is suited for non stationary signals (like, very brief signals and signals with interesting components at different scales) [Hub95].

In the discrete wavelet transform, an image signal can be analyzed by passing it through an analysis filter bank followed by a decimation

operation. This analysis filter bank consists of a low pass and a high pass filter at each decomposition stage [Gra95].

When a signal passes through these filters, it is split into two bands. The low pass filter, which corresponds to an averaging operation, extracts the coarse information of the signal. The high pass filter, which corresponds to a differencing operation, extracts the detail information of the signal. The output of the filtering operations is then decimated by two [HiJaSe94].

The two-dimensional wavelet transform can be accomplished by performing two separate one-dimensional transforms, as depicted in figure (2.6). First, the image is filtered along the x-dimension and decimated by two. Then, it is followed by filtering the sub-image along the y-dimension and decimated by two. Finally, the image data contents are split into four bands denoted by LL, HL, LH and HH after one-level decomposition (see figure 2.6b) [Mul97].

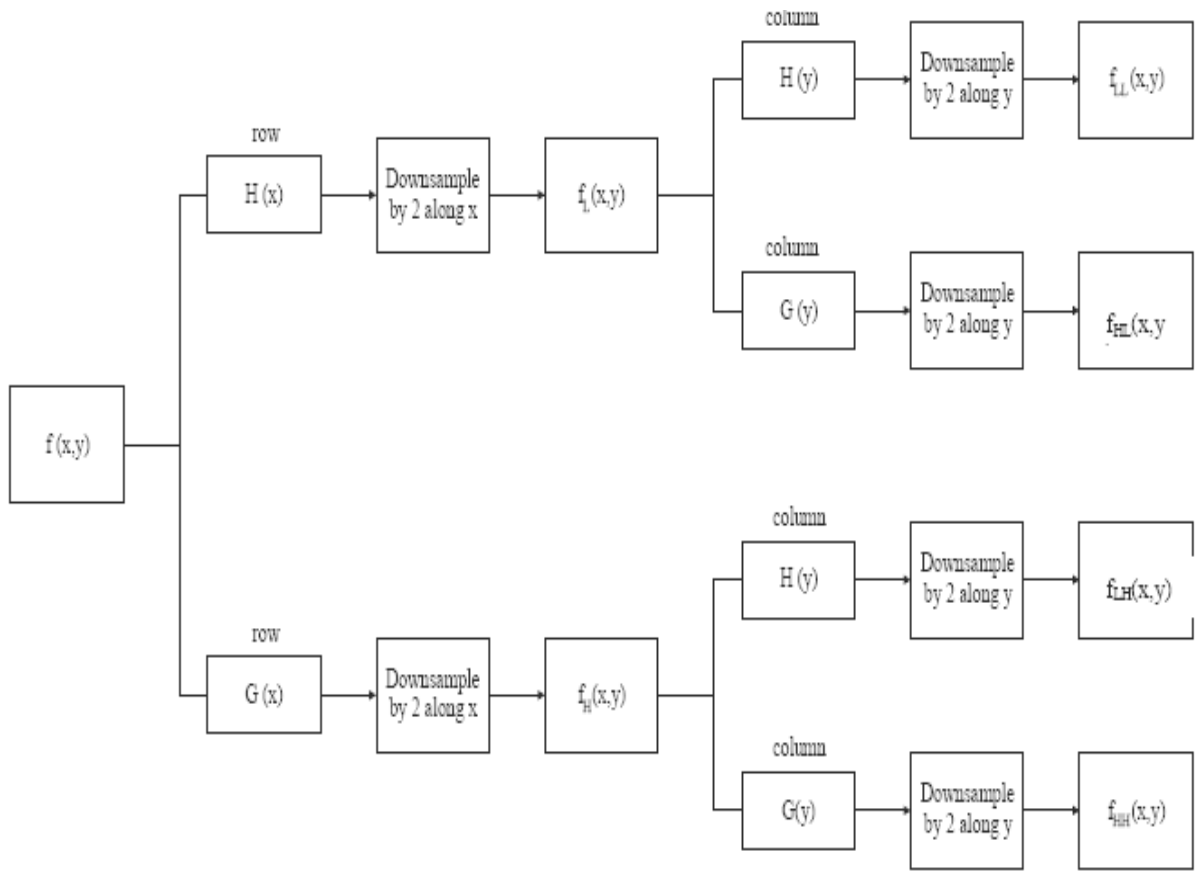


Fig (2.6) Wavelet decomposition

Further decompositions can be achieved by acting upon the LL subband successively, and then the resultant image is split into multiple bands, as shown in Figures (2.7c) and (2.7d) [StDeSa94, Bur98].

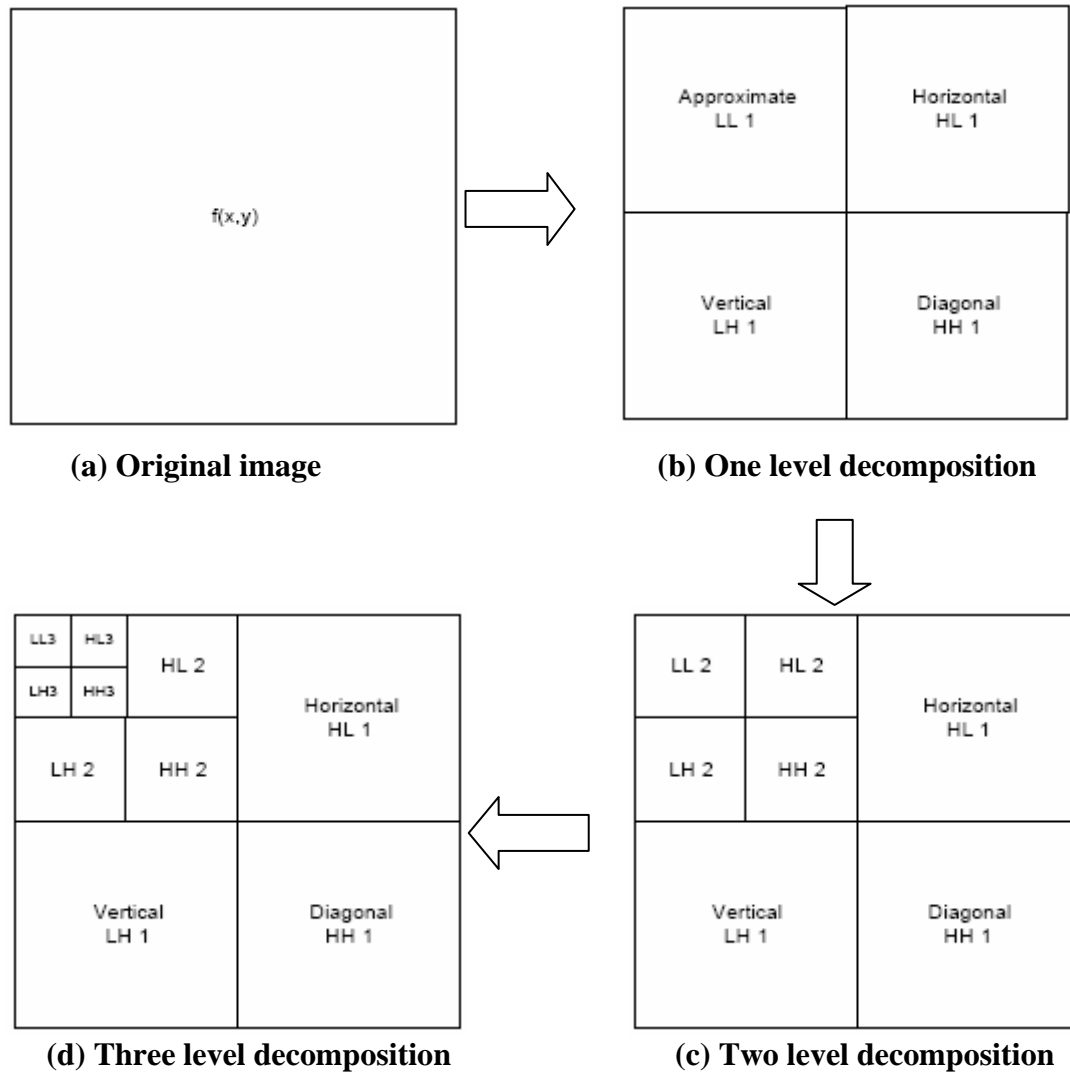


Fig (2.7) Two-dimensional discrete wavelet transforms

In mathematical terms, the averaging operation (or low pass filtering) is the inner product between the signal and the scaling function whereas the differencing operation (or high pass filtering) is the inner product between the signal and the wavelet function [Cro01].

The reconstruction of the image can be carried out by the following procedure. First, the four subbands are up-sampled by a factor of two at the coarsest scale, and the subbands are filtered in each dimension. Then the sum of the four filtered subbands is determined to reach the low-low

subband for the next finer scale. This process is repeated until the image is fully reconstructed, as depicted in Figure (2.8) [Val01].

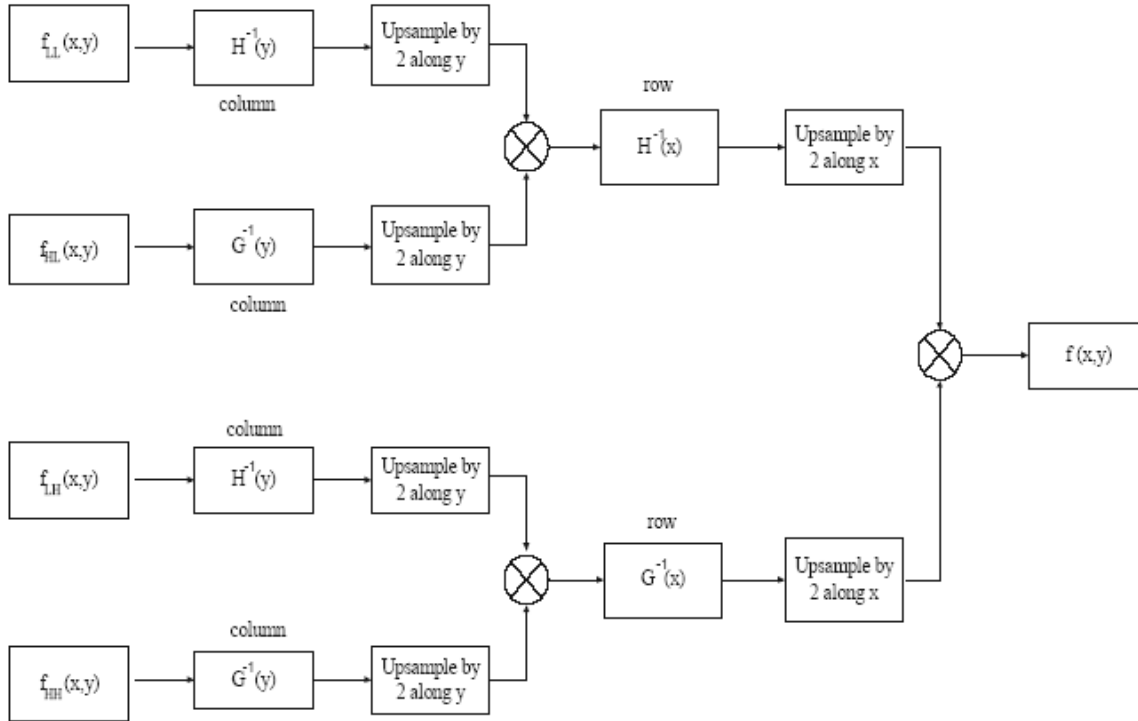


Fig (2.8) The two-Dimensional inverse discrete wavelet transform

2.9 Types of Wavelet Image Decompositions

This section discusses several ways for decomposing an image, each involves a different algorithm and resulting in subbands with different energy compactions. It is important to realize that the wavelet filters and the decomposition method are independent. The DWT of an image can use any set of filters and decompose the image in any way. The only limitation is that there must be enough data points in the subbands to cover all the filter taps. The main decomposition types considered with wavelet transform are described in the following sub-sections [Sal02].

2.9.1 Line Decomposition

In this method the DWT is applied on each row of the image, resulting in smooth coefficients on the left (subband L1) and detail coefficients on the right (subband H1), as shown in figure (2.9). Then the subband L1 is partitioned into L2 and H2, and the process is repeated until the entire coefficient matrix is turned into detail coefficients. In the second stage of this decomposition scheme, the wavelet transform is applied recursively to the leftmost column, resulting in one smooth coefficient at the top left corner of the coefficient matrix. This last step may be omitted if the used decomposition method requires that the image rows be individually compressed [Sal02].

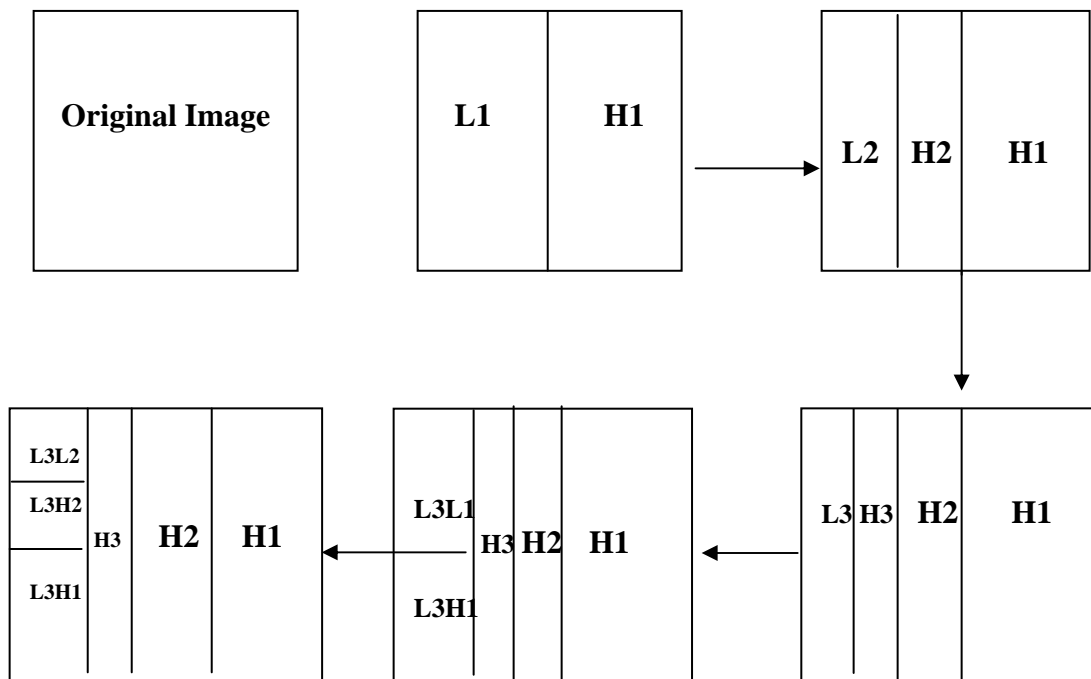


Fig (2.9) Line wavelet decomposition

2.9.2 Quincunx Decomposition

Quincunx decomposition, as shown in figure (2.10), proceeds level by level and decomposes subband L_i of level i into subbands H_{i+1} and L_{i+1} of level $i+1$. It is efficient and computationally simple. On average, it achieves more than four times energy compaction in comparison with the line method. It results in fewer subbands than most other wavelet decomposition, a feature that may lead to reconstruct images with slightly lower visual quality. This method is not used much in practice, but it may perform extremely well and may be the best performer in many practical situations [Sal02].

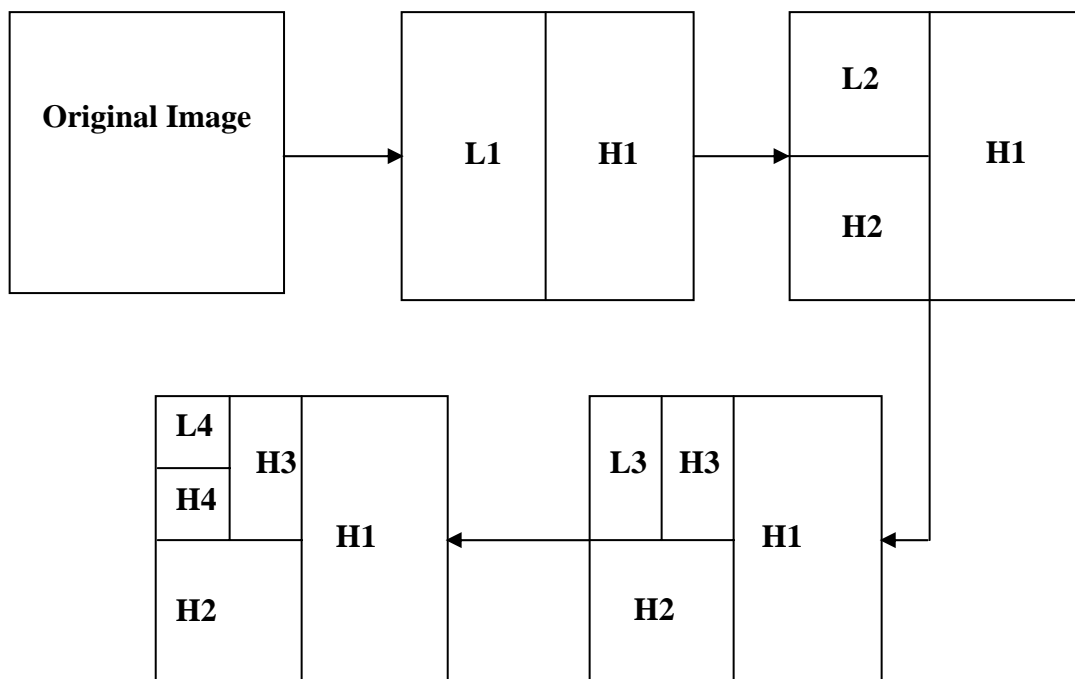


Fig (2.10) Quincunx Wavelet decomposition

2.9.3 Pyramid Decomposition

The pyramid decomposition, as shown in figure (2.11), is by far the most common method used to decompose images that are wavelet transformed. It results in subbands with horizontal, vertical, and diagonal image details. The three subbands at each level contain horizontal, vertical, and diagonal image features at a particular scale, and each scale is divided by an octave in spatial frequency (division of the frequency by two). Pyramid decomposition turns out to be very efficient way of transforming significant visual data to the detail coefficients. Its computational complexity is about 30% higher than that of the quincunx method, but its image reconstruction abilities are higher. The reasons for the popularity of the pyramid method may be that it is symmetrical, and its mathematical description is simple [Sa102].

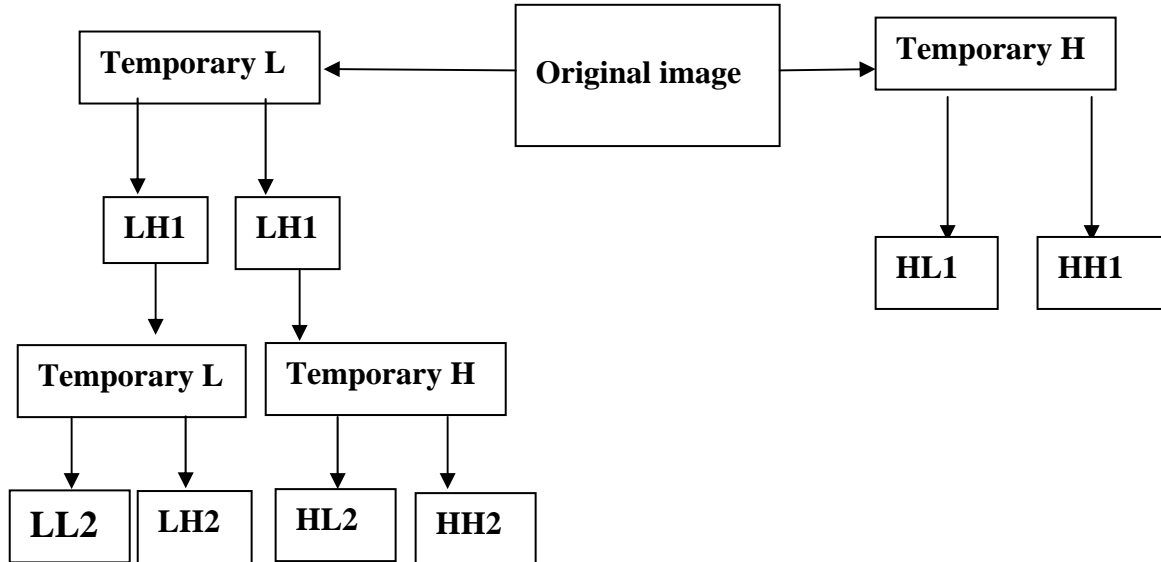


Fig (2.11) Pyramid wavelet decomposition

2.9.4 Standard Decomposition

The first step in the standard decomposition, shown in figure (2.12), is to apply a discrete wavelet filter upon all rows of the image, obtaining subbands L1 and H1. This step is repeated on L1 to obtain L2 and H2, and so on for k times. This is followed by the second step where a similar calculation is applied k times on the columns.

The result is to have one smooth coefficient at the top-left corner of the coefficients matrix. This method is somewhat similar to line decomposition. It has an important feature that is when a coefficient is quantized it may affect a long and thin rectangular area in the reconstructed image. Thus, very coarse quantization may result in artifacts in the reconstructed image in the form of horizontal rectangles [Sal02].

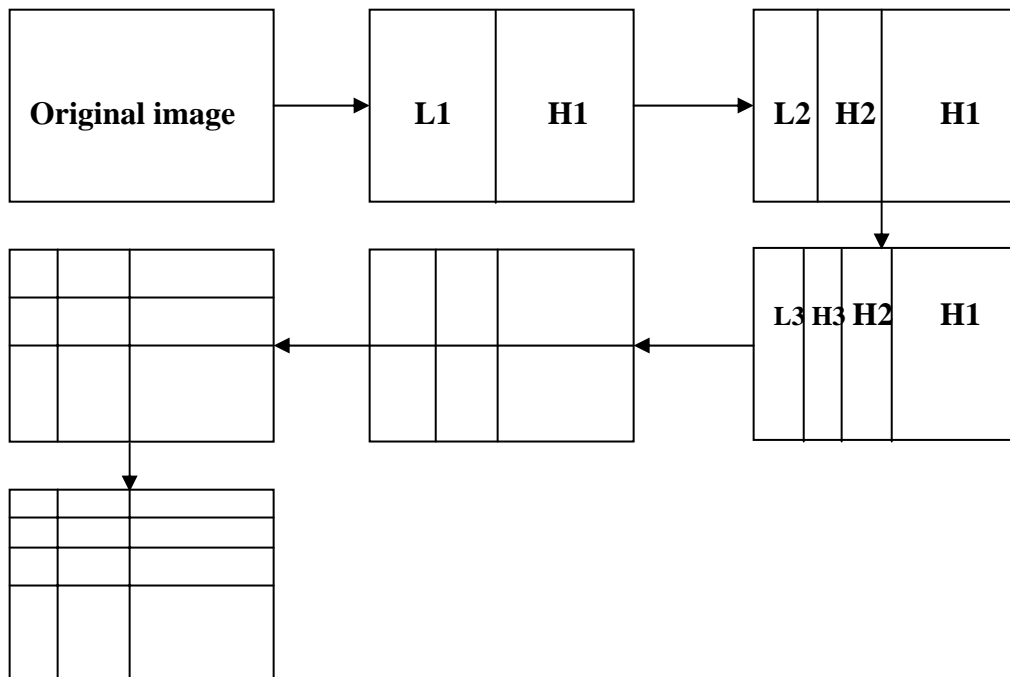


Fig (2.12) Standard wavelet decomposition

2.9.5 Full Wavelet Decomposition (Packet)

This type of decomposition is also called Wavelet Packet transform. It is shown in figure (2.13).

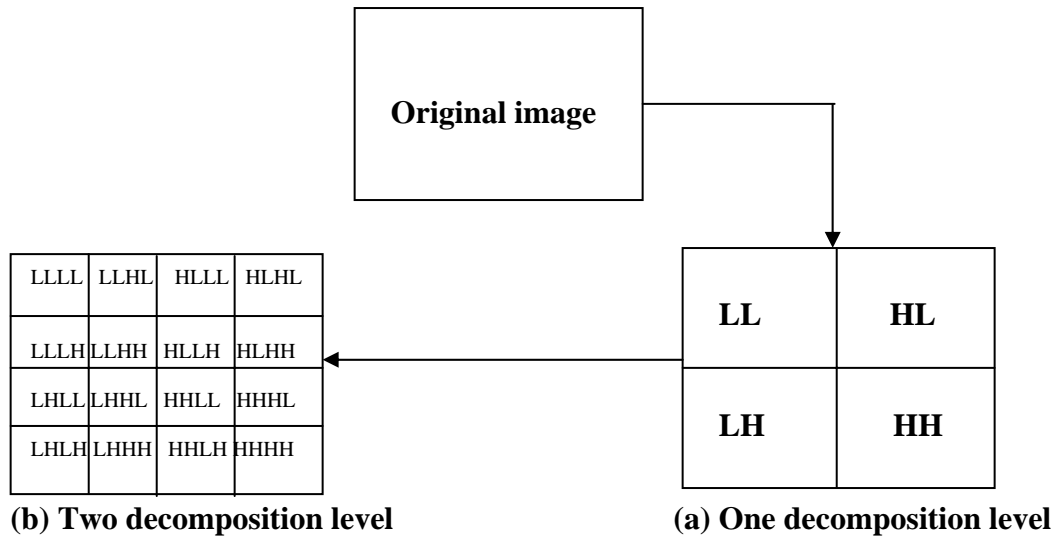


Fig (2.13) Full wavelet decomposition

Let us denote the original image by I_0 . It is assumed that its size is $2^l \times 2^l$ when applying the 2-D discrete wavelet transform on it, it ends up with a matrix I_1 partitioned into four subbands. The same 2-D DWT (i.e., using the same wavelet filters) could be applied recursively on each of the four subbands individually. The result is a coefficient matrix I_2 consist of 16 subbands. When this process is carried out r times, the result is a coefficient matrix consists of $2^r \times 2^r$ subbands, each of size $2^{l-r} \times 2^{l-r}$.

The top-left subbands contains the smooth coefficients, depending on the particular wavelet filter used, it may look like a small versions of the original image. The other subbands contain detail coefficients. Each subband corresponds to its frequency band, while each individual transform coefficient corresponds to a local spatial region. By increasing the recursion

depth r , the frequency resolution is increased at the expense of spatial resolution [Sal02].

2.10 Haar Wavelet Transform (HWT)

The oldest and most basic wavelet system had been constructed from the Haar basis function. The equations for forward Haar wavelet transform and inverse Haar wavelet transform are given in the following subsections.

1. Forward Haar Wavelet Transform (FHWT) [Jia03]

Given an input sequence $(x_i)_{i=0, \dots, N-1}$, then FHWT produces L_i and H_i (where $i=0, \dots, N/2 - 1$) by using the following transforms equations:

A. If N is even

$$\left. \begin{aligned} L_i &= \frac{x_{2i} + x_{2i+1}}{\sqrt{2}}, \quad i=0, \dots, (N/2) - 1 \\ H_i &= \frac{x_{2i} - x_{2i+1}}{\sqrt{2}}, \quad i=0, \dots, (N/2) - 1 \end{aligned} \right\} \dots\dots\dots (2.20)$$

B. If N is odd

$$\left. \begin{aligned} L_i &= \frac{x_{2i} + x_{2i+1}}{\sqrt{2}}, \quad i=0, \dots, (N-1/2) - 1 \\ H_i &= \frac{x_{2i} - x_{2i+1}}{\sqrt{2}}, \quad i=0, \dots, (N-1/2) - 1 \\ L_{(N+1)/2} &= \sqrt{2}x_{(N-1)} \\ H_{(N+1)/2} &= 0 \end{aligned} \right\} \dots\dots\dots (2.21)$$

2. Inverse Haar Wavelet Transform (IHWT) [Jia03]

The inverse one-dimensional HWT is, simply, the inverse to the equations of FHWT; so, the IHWT equations are:

A. if N is even

$$\left. \begin{aligned} X_{2i} &= \frac{L_i + H_i}{\sqrt{2}}, \quad i=0 \dots N/2 - 1 \\ X_{2i+1} &= \frac{L_i - H_i}{\sqrt{2}}, \quad i=0 \dots N/2 - 1 \end{aligned} \right\} \dots\dots\dots (2.22)$$

B. If N is odd

$$\left. \begin{aligned} X_{2i} &= \frac{L_i + H_i}{\sqrt{2}}, \quad i=0 \dots (N-1)/2 \\ X_{2i+1} &= \frac{L_i - H_i}{\sqrt{2}}, \quad i=0 \dots (N-1)/2 \\ X_{N-1} &= \sqrt{2}L_{(N+1)/2} \end{aligned} \right\} \dots\dots\dots (2.23)$$

2.11 The Integer Wavelet Transform (TAP 5/3)

Wavelet transform operate on integer values to produce integer valued wavelet coefficient. Integer wavelet transform have been effectively used for lossless compression of images, the results of the invertible integer wavelet transform are integer, while the computations are done with floating point numbers. Due to rounding each filter output to an integer value, then the transform maps integers to integers. This kind of transform is named IWT and its implementation is based on a simple lifting scheme (LS) [DrLi01].

2.12 Float Wavelet Transform (Tap 9/7)

The biorthogonal filter (Tap 9/7) was chosen as the basis of the JPEG2000 lossy image compression standard for still images. The coefficients of this filter are given as floating-point numbers. The float filter can be lifted (factorized) in order to speed up the convolution step. It is primarily suited to high visual quality compression. The use of floating-point arithmetic in the DWT, and the associated rounding errors, make it unsuitable for strictly lossless compression [Mah05].

The (9/7) floating-point wavelet transform is computed by executing four "lifting" steps followed by two "scaling" steps on the extended pixel values P_k through P_m . Each step is performed over all the pixels in the tile before the next step starts [Sal02].

2.13 Fidelity Criteria

Lossy compression techniques cause some information losses up to a certain tolerated level. Thus the use of fidelity criteria is required to measure or estimate the amount of losses. Two kinds of fidelity criteria are normally used; they are objective and subjective measures [Umb98]:

a. Objective

It is measured, mathematically, as the amount of error in the reconstructed image. They are useful as a relative measure for comparing different versions of the same image. The error (e) between an original (uncompressed) pixel value and reconstructed (decompressed) pixel value can be defined as:

$$e(x, y) = f'(x, y) - f(x, y) \dots\dots\dots (2.24)$$

Where, f' is the reconstructed image.

f is the original image.

e is the error.

The total error (e_T) of a decompressed image ($N \times M$) is expressed mathematically as:

$$(e_T) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} [f'(x, y) - f(x, y)] \dots\dots\dots (2.25)$$

The most widely used objective fidelity criteria are [Kom94]:

1. Mean Square Error (MSE)

$$\text{MSE} = \frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (f'(x, y) - f(x, y))^2 \dots\dots\dots (2.26)$$

2. Root Mean Square Error (RMSE)

It is square root of MSE. The smaller the value of the error metrics (RMSE, MSE and MAD) the better the compressed image represents the original image.

$$\text{RMSE} = \sqrt{\frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (f'(x, y) - f(x, y))^2} \dots\dots\dots (2.27)$$

3. Peaks-Signal to Noise Ratio (PSNR)

PSNR express the ratio of the maximal signal power to that of the error:

$$\text{PSNR} = \frac{[\text{Max}(f(x, y)) - \text{Min}(f(x, y))]^2}{\text{MSE}} \dots\dots\dots (2.28)$$

Mostly, it is measured in units of decibel, thus the PSNR is given as:

$$\text{PSNR(dB)} = 10 \log_{10} \left[\frac{(L-1)^2}{\text{MSE}} \right] \dots\dots\dots (2.29)$$

Where L is the number of grey levels

4. Compression Ratio (CR): It is a basic measure for the performance of any compression algorithm. It is defined as the ratio of the original data (s_o) to the compressed data size (s_c):

$$\text{CR} = \frac{s_o}{s_c} \dots\dots\dots (2.30)$$

b. Subjective

Subjective fidelity criteria depends on human evaluation, the evaluation can be classified into three categories [Gon02], [Umb98]:

1. Impairment test: where the test subject scores the images in terms of how bad they are.
2. Quality test: where the test subject rates the images in terms of how good they are.
3. Comparison test: this test provides a relative measure, which is the easiest metric for most people to determine.

Chapter Three

The Proposed Image Compression System

3.1 Introduction

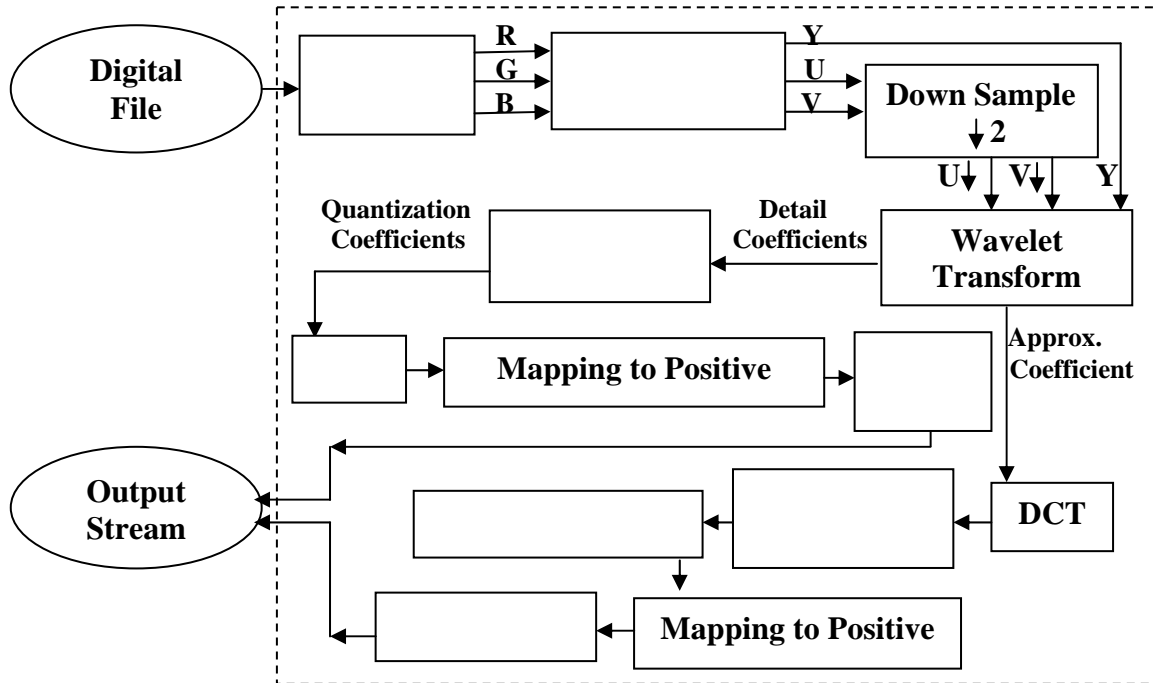
In this chapter the layout of the proposed image compression scheme is investigated, and its implementation steps are illustrated. The techniques used in this work are the Sub-Band Coding (using Wavelet transform) and Transform Coding (using DCT). Both transforms show some advantages and disadvantages and they perform in different ways within image regions. In the proposed transform scheme most of the benefits of both transforms are taken. The DCT shows the superior compression performance when the image area has poor power concentration in high frequency area, for this reason it is utilized to encode the low-low subband of the wavelet domain. While, the subband coding mechanism is adopted to encode the detail (high) subbands of the wavelet transform.

In this chapter, the implementation steps of the established image compression system are given. The data of the color components (R, G, and B) are transformed to (Y, U, and V) components, to take the advantage of the existing spectral correlation and consequently to gain more compression. Also, the low spatial resolution characteristic of the human vision system to the chromatic components (U and V) is utilized to increase the compression ratio without making significant subjective distortions.

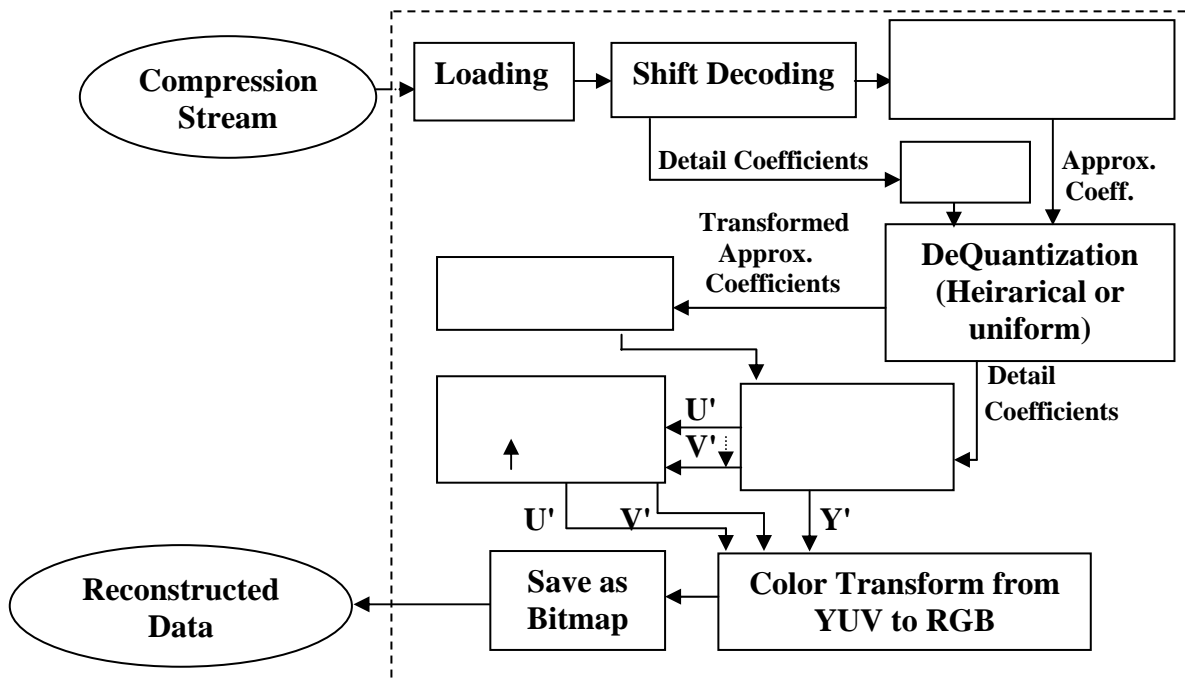
3.2 System Model

Digital image compression system consists of two units: the first unit is called "Encoding unit", and the second one is called "Decoding unit".

Each one of these two units consists of many parts, as shown in Figure (3.1).



A. Encoder



B. Decoder

Fig (3.1) The Encoder/Decoder of the proposed system

3.3 Encoding Unit

As shown in Figure (3.1a) this unit consists of thirteen modules which are all together responsible for reducing the data size of the desired color image, and generate compressed stream of data that represent the image. In the following subsections a functional description and the implemented steps for each module are given.

3.3.1 Image Loading

In this part, the color image data is loaded and put it in three arrays, each has size $(H \times W)$, where H denotes the height of the image, and W denotes its width. Figure (3.2) presents a typical RGB color image (256×256) with its three RGB color bands.



Fig (3.2) Lena Image and its RGB components

3.3.2 Color Transform

One of the main disadvantages of using RGB color space in some image processing applications is due to the fact that the contents of R, G, B bands of the natural images are correlated to some extent, this makes the RGB space unsuitable for compression.

Today the YUV color space is widely used for coding color images and color video, taking into consideration that some previous studies indicated that more than 80% of the color image information is held by

Y component, and 10% in each other two (i.e. U and V) components [SaHo98].

In this research project the loaded RGB color image was transformed to YUV color space by using the equations (2.7- 2.9).

The inverse transform of YUV to RGB is done by using the following equations:

$$R = Y + 0.00000 U + 1.40200 V \dots\dots\dots (3.1)$$

$$G = Y - 0.34414U - 0.71417V \dots\dots\dots (3.2)$$

$$B = Y + 1.77200U + 0.00000V \dots\dots\dots (3.3)$$

In order to get an effective compression the (U, V) component have been down sampled by 2. The adopted downsampling method was the averaging method, where the average value of each (2x2) block is determined, and taken as a value represent that block in the downsampled image. Figure (3.3) shows the results of applying downsampling step on the (U, V) color subbands of Lena image.

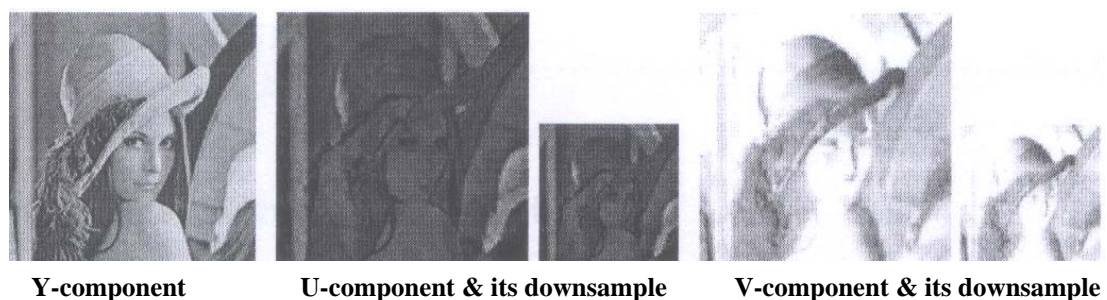


Fig (3.3) YUV component of Lena RGB color image

3.3.3 Wavelet Transform

In this proposed system Haar wavelet transform was applied on each color transform band, individually. After each wavelet transform pass four subbands (LL, LH, HL, and HH) are produced. The goal of wavelet

transform is to map the image data into an alternative representation, in this representation the image data is decomposed into subbands each hold certain kind of image information; such that most of the image information energy is concentrated in the lowest frequency subband (i.e., LL subband). List (3.1) illustrates the implemented steps of Haar wavelet transform

Algorithm (3.1) Haar Wavelet transforms

Input:

T () is the array of the color transform coefficients

Wid is the image width

Hgt is the image height

NoPass is the number of wavelet transform levels or passes

Output:

T_w () an array consists, at least, of four subbands (LL, LH, HL, HH)

Steps:

W= wid: H= Hgt

For all I where $1 \leq I \leq NoPass$

Hh= H div 2: Wh= W div 2

*For Y=0 Hh-1: Y1 = 2*Y : Y2 = Y1+1*

*For X=0 Wh-1: X1 = 2*X : X2 = X1+1*

$T_w(X, Y) = (T(X1, Y1) + T(X2, Y1) + T(X1, Y2) + T(X2, Y2)) / 2$

$T_w(X+Wh, Y) = (T(X1, Y1) + T(X2, Y1) - T(X1, Y2) - T(X2, Y2)) / 2$

$T_w(X, Y+Hh) = (T(X1, Y1) - T(X2, Y1) + T(X1, Y2) - T(X2, Y2)) / 2$

$T_w(X+Wh, Y+Hh) = (T(X1, Y1) - T(X2, Y1) - T(X1, Y2) + T(X2, Y2)) / 2$

End Loop X

End Loop Y

End Loop I

3.3.4 Quantization of Detail Coefficients

Image quantization is the process of reducing the number of possible values of a quantity, and consequently reducing the number of bits needed to represent it.

In our proposed system, the uniform quantization was adopted to quantize the coefficients of each wavelet (detail) subband individually. The quantization step used to quantize the coefficients of each subband was determined according to the following equation:

$$Q_{\text{step}} = \begin{cases} Q\alpha^{n-1} & \text{for LH, HL in } n^{\text{th}} \text{ level} \\ Q\beta\alpha^{n-1} & \text{for HH in } n^{\text{th}} \text{ level} \end{cases} \dots\dots\dots(3.4)$$

Where, n is the wavelet level number (i.e., the pass number), (Q, α, β) are quantization parameters (such that, $Q \geq 1$, $\alpha \leq 1$, $\beta \geq 1$). According to the above equation the value of the quantization step is reduced with the increase of the wavelet level, and its value for HH-subband is greater than its value for the corresponding HL and LH subbands.

The quantization index for each wavelet (detail) coefficient is determined by using the following equation:

$$T_q(x, y) = \text{round}\left(\frac{T_w(x, y)}{Q_{\text{step}}}\right) \dots\dots\dots (3.5)$$

Where,

$T_w()$ is the array of the wavelet transform coefficients.

$T_q()$ its quantization index array.

Algorithm (3.2) illustrates the implemented steps of the applied uniform quantization method.

Algorithm (3.2) Image Quantization**Input:**

$T_w()$ is the array of the wavelet transform coefficients.

W is the image width.

H is the image height.

Q is the initial quantization step for (LH, HL, HH)

, β are the quality numbers.

Output:

Quantization indices $T_q()$

Steps:

$W_w = W: H_h = H$

For all j where $1 \leq j \leq Nopass$

$W_m = (W_w + 1) \text{ div } 2: H_m = (H_h + 1) \text{ div } 2$

$W_{m1} = W_w - 1: H_{m1} = H_h - 1$

$X_s = W_m: X_e = W_{m1}: Y_s = 0: Y_e = H_m - 1$

$Q_{step} = Q * \alpha^{j-1}$

For all x, y where $X_s \leq x \leq X_e$ and $Y_s \leq y \leq Y_e$

$T_q(x, y) = \text{Cint}(T_w(x, y) / Q_{step})$

End loop x, y

$X_s = 0: X_e = W_m - 1: Y_s = H_m: Y_e = H_{m1}$

For all x, y where $X_s \leq x \leq X_e$ and $Y_s \leq y \leq Y_e$

$T_q(x, y) = \text{Cint}(T_w(x, y) / Q_{step})$

End loop x, y

$X_s = W_m: X_e = W_{m1}: Y_s = H_m: Y_e = H_{m1}$

If $j = 1$ then $Q_{step} = Q_{step} * \beta$

For all x, y where $X_s \leq x \leq X_e$ and $Y_s \leq y \leq Y_e$

$T_q(x, y) = \text{Cint}(T_w(x, y) / Q_{step})$

End loop x, y

$W_w = W_m: H_h = H_m$

End loop j

3.3.5 Run Length Encoding (RLE)

It is based on the idea of encoding a consecutive occurrence of the same symbol. This is achieved by replacing a series of repeated symbols with a count and the symbol. In the proposed system the indices coming from the quantization process usually hold long runs of zeros. In the quantization stage the wavelet coefficients are divided by large quantization step value and the result rounded to the nearest integer, this process causes the production of a sequence of small integer numbers which holds long runs of zeros.

Algorithm (3.3) illustrates the implemented steps of the run length encoding stage.

Algorithm (3.3) Run Length Encoding (RLE)

Input:

$T_q()$ is the array of (LH, HL, HH) subbands from quantization

W is the image width

H is the image height

Output:

$A()$ Vector consist Typ and Lng and TotBits needed.

Procedure:

$W = \text{Wid}; H = \text{Hgt}$

For all I where $1 \leq I \leq \text{NoPasses}$

$Wh = W \text{ div } 2; Hh = H \text{ div } 2$

For all J where $1 \leq J \leq 3; L = -1$

Select Case J

Case 1: $X_1 = Wh; X_2 = W - 1; Y_1 = 0; Y_2 = Hh - 1$

Case 2: $X_1 = 0; X_2 = Wh - 1; Y_1 = Hh; Y_2 = H - 1$

Case 3: $X_1 = Wh; X_2 = W - 1; Y_1 = Hh; Y_2 = H - 1$

End Select

To be continue

```

If  $T_q(X_1, Y_1) = 0$  Then
  Typ = 0: Lng = 1: L = L + 1: A (L) = Typ
Else
  Typ = 1: Lng = 0: L = L + 1: A (L) = Typ
  L = L + 1: A (L) =  $T_q(X, Y)$ 
End If
Flg = 0
For all Y where  $Y_1 \leq Y \leq Y_2$ 
  If Flg = 0 Then
    Xs =  $X_1$ : Xe =  $X_2$ : Stp = 1: Flg = 1
  Else
    Xs =  $X_2$ : Xe =  $X_1$ : Stp = -1: Flg = 0
  End If
  For all X where  $Xs \leq X \leq Xe$  Step Stp
    If  $X \neq X_1$  or  $Y \neq Y_1$  Then
      If  $T_q(X, Y) = 0$  Then
        If Typ = 0 Then Lng = Lng + 1 Else Lng = 1: Typ = 0
      Else
        If Typ = 0 Then
          Typ = 1: L = L + 1: A (L) = Lng
          L = L + 1: A (L) =  $T_q(X, Y)$ 
        Else
          L = L + 1: A (L) = C
          L = L + 1: A (L) =  $T_q(X, Y)$ 
        End If
      End If
    End If
  End loop X, Y
For all K where  $1 \leq K \leq 2$ : M = K: P = -1
  For all N where  $M \leq N \leq L$  Step 2
    P = P + 1: B (P) = A (N)
  End loop N

```

```

Max = B (0)
For all N where 1 ≤ N ≤ P
    If Max < B (N) Then Max = B (N)
End loop N
NoBits = Log (Max) / Log (2)
If NoBits < Max Then NoBits = NoBits + 1
TotBits = TotBits + (P + 1) * NoBits
End loop K
End loop J
W = W / 2: H = H / 2
End loop I

```

3.3.6 Discrete Cosine Transform (DCT)

In this stage the low-low (approximation) subband of the wavelet transformed image is firstly partitioned into (8x8) fixed blocks, then the data of each block is decomposed using the DCT transform. The two dimensional DCT was performed by applying equation (2.17).

Each block (8x8) of the approximate coefficients is DCT transformed. The following code illustrates the implemented steps of the DCT transformation stage.

Algorithm (3.4) Discrete Cosine Transformation (DCT)**Input:**

$T_w()$ is the array of approximation coefficients

W is the image width

H is the image height

$BlkSiz$ is the block size

Output:

$C()$ is the transformed block of an (8×8) array of DCT coefficients

Procedure:

$BlkSiz = 8$: $Sum = 0$

$W = Wid$: $H = Hgt$

$C_0 = 1 / Sqr(2)$

$C2 = 1 / Sqr(2 * BlkSiz)$

$C1 = 3.14159 / (2 * BlkSiz)$

For all I where $0 \leq I \leq BlkSiz - 1$

For all J where $0 \leq J \leq BlkSiz - 1$

For all Y where $0 \leq Y \leq BlkSiz - 1$

For all X where $0 \leq X \leq BlkSiz - 1$

$W1 = (2 * Y + 1) * J * C1$

$W2 = (2 * X + 1) * I * C1$

$Sum = Sum + T_w(X, Y) * Cos(W1) * Cos(W2)$

If $I = 0$ Then $C_i = C_0$ Else $C_i = 1$

If $J = 0$ Then $C_j = C_0$ Else $C_j = 1$

$C(I, J) = C2 * C_i * C_j * Sum$

End loop X

End loop Y

End loop J

End loop I

3.3.7 DCT Coefficients Quantization

The produced DCT coefficients are real valued, and in order to increase the compression, they must be quantized before storage.

As a first step in this stage the uniform quantization was applied on the AC-transform coefficients, $C(i, j)$, using the following equation:

$$C_q(i, j) = \text{round}\left(\frac{C(i, j)}{Q_c(i, j)}\right) \dots\dots\dots (3.6)$$

Where, $Q_c(i, j)$ is the quantization step of the $(i, j)^{\text{th}}$ AC-coefficients, and it is determined by the following equation:

$$Q_c(i, j) = Q_{\text{low}}(1 + \gamma(i + j - 1)) \dots\dots\dots (3.7)$$

Where,

Q_{low} is the lowest quantization step value for AC-coefficients

γ is the increamentation rate

(i, j) are the frequency indices, whose values are within the range $[0\dots, \text{Block length} - 1]$

As a second step in this stage, the DCT coefficient (i.e, $C(0, 0)$) is the quantized using:

$$C_q(0,0) = \text{round}\left(\frac{C(0,0)}{Q_{\text{DC}}}\right) \dots\dots\dots(3.8)$$

The way of applying the uniform quantization on DCT coefficients is illustrated in code list (3.5)

Algorithm (3.5) Quantization DCT Coefficients**Input:**

$C()$ is the transformed block of an (8x8) array of DCT coefficients

W is the image width

H is the image height

Q_{low} is the quantization step of the AC-coefficients

Q_{DC} is the quantization step of the DC- coefficient

γ is the quantization parameter

Output:

$Q_{DC}()$ is the array of quantization of DCT coefficients

Procedure:

For all I where $0 \leq I \leq \text{BlkSiz}-1$

For all J where $0 \leq J \leq \text{BlkSiz}-1$

If $I = 0$ and $J = 0$ then

$$Q_{DC}(0, 0) = \text{round}(C(0, 0)/Q_{DC})$$

Else

$$Q = Q_{low} * (1 + \gamma * (I + J))$$

$$Q_{DC}(I, J) = \text{round}(C(I, J) / Q)$$

End if

End loop J

End loop I

3.3.8 Mapping to Positive

In this stage, all the determined quantization indices of DCT coefficients and the quantization indices of the detail subbands are mapped to be positive, the following mapping equation was used to convert the signed integer into positive integers:

$$C' = \begin{cases} 2C & \text{If } C \geq 0 \\ 2|C|+1 & \text{If } C < 0 \end{cases} \dots\dots\dots(3.9)$$

Where, C represents the signed integer value of the quantization index. This kind of mapping insure that all coefficients values are mapped as positive integers, and is to keep the optimal number of bits needed to shift encode the quantized coefficients as small as possible, taking into consideration that the histogram of the coded coefficients is highly peaked around zero, see algorithm (3.6)

Algorithm (3.6) Mapping to Positive

Input:

$Q_{DC}()$ is the array of quantization of DCT coefficients

W is the image width

H is the image height

Output:

Positive quantization indices

Steps:

For all I where $0 \leq I \leq \text{BlkSiz} - 1$

For all J where $0 \leq J \leq \text{BlkSiz} - 1$

If $Q_{DC}(I, J) > 0$ then

$$Q_{DC}(I, J) = 2 * Q_{DC}(I, J)$$

Else If $Q_{DC}(I, J) < 0$ then

$$Q_{DC}(I, J) = 2 * \text{abs}(Q_{DC}(I, J)) + 1$$

End if

End loop J

End loop I

3.3.9 Zigzag and S-Shift Optimizer

In this stage the 2-dimentional array of the quantization indices of DCT coefficients are arranged in one dimensional array by using Zigzag scanning method. And before applying an adaptive shift coding method on the array of coefficients its optimized coding parameters should

determined using shift-optimizer. The values of the DCT quantization indices are usually concentrated around zero, and their histogram is highly peaked, so the use of shift encoder is very suitable, and can lead to good compression (packing) result.

The mechanism applied to compute the optimal length (N), in bits, of the shift codewords is based on scanning all possible codeword lengths, starting from one bit and proceeding more till N_{\max} bits; the number (N_{\max}) represent the minimum number of bits required to represent the maximum (Max) coefficient value in the set of DCT quantization indices. Also, the number (N_{\max}) is considered as the length (in bits) of the second (auxiliary) codeword. The scan method was applied to test all possible values of bits that can be assigned to the first (shortest) codeword, so the tested range of the number of bits of the first codeword is $[1, N_{\max}]$. For each possible codeword length (N) the total number of all bits, $S[n]$, required for encoding the quantization indices is determined by using the following equations:

$$S(N) = N \sum_{i=0}^{R-1} H(i) + (N + N_{\max}) \sum_{i=R}^{\max} H(i) \dots\dots\dots (3.10)$$

$$N_{\max} = \lceil \log_2(\text{Max}) \rceil \dots\dots\dots (3.11)$$

$$R = 2^N - 1 \dots\dots\dots (3.12)$$

Where $\lceil . \rceil$ is the lowest integer value greater than (.).

The value of (N) which leads to the lowest determined value of $S(N)$ is considered as the optimal length of the first codeword.

The code list (3.7) illustrates the implemented steps to perform zigzag method and determine the optimal length of the shift codewords.

Algorithm (3.7) Zigzag and S-Shift Optimizer**Input:**

$Q_{DC}()$ array of quantization indices of (LL) coefficients

W is the image width

H is the image height

$BlkLen$ is the length of the block

Output:

Number of required bits (N, N_{max})

Steps:

Load the ScanOrder [0, ..., 63]:

{0,1,8,16,9,2,3,10,17,24,32,25,18,11,4,5,12,19,26,33,40,48,41,34,27,20,13,6,7,14,21,28,35,42,49,56,57,50,43,36,29,22,15,23,30,37,44,51,58,59,52,45,38,31,39,46,53,60,61,54,47,55,62,63}

For all I where $0 \leq I \leq 63$

$X = \text{ScanOrder}[I] \bmod 8$

$Y = \text{ScanOrder}[I] \text{ div } 8$

$Z(I) = Q_{DC}(X, Y)$

End loop I

"" Compute the maximum number of bits required to encode each DCT Coefficients

$Max = Z(0)$

For all $Z(i)$ coefficients (i) (where)

If $Max < Z(i)$ then $Max = Z(i)$

End loop i

"" Compute maximum number of required bits

$N_{max} = 1; K = 1$

While $Max \geq K$

$K = 2 * K + 1$

$N_{max} = N_{max} + 1$

Wend

To be continue

```

''' Compute the histogram  $H()$  of the quantized transform coefficients
Set  $H(i) = 0$ 
For all  $i$  where  $0 \leq i \leq \text{Max}$ 
     $j = Z(i)$ 
     $H(j) = H(j) + 1$ 
End loop  $i$ 
''' Shift coding optimizer to compute the optimal number of required bits ( $N, N_{\text{max}}$ )
 $\text{Len} = \text{BlkLen} * \text{BlkLen}$ 
 $\text{MinBits} = N_{\text{max}} * \text{Len}$ 
 $N = N_{\text{max}}$ 
For all  $N$  where  $1 \leq N \leq N_{\text{max}} - 1$ 
     $R = 2^N - 1$ 
     $S_m = 0$ 
    For all  $J$  where  $0 \leq J \leq \text{Max}$ 
        If  $J < R$  then
             $S_m = S_m + N * H(J)$ 
        Else
             $S_m = S_m + (N + N_{\text{max}}) * H(J)$ 
        End if
    End loop  $J$ 
    If  $S_m < \text{MinBits}$  then
         $\text{MinBits} = S_m$ 
    End if
End loop  $N$ 

```

3.3.10 Shift Encoder

In this stage, the input data are the quantization indices of (DCT) coefficients of the LL-subband. The codewords produced by applying shift-coding are sent to the compression bitstream (which represents the compressed data file), see algorithm (3.8).

Algorithm (3.8) Shift Encoder**Input:**

$Z()$ 1D array of quantization indices of (LL) coefficients

N_{max} number of bits required to encode the largest number of bit required to encode length binary representation

N the length (in bits) of the first shift codeword

$BlkLen$ is the length of block

Output:

A set of integers whose lengths are either N or N_{max}

Procedure:

$$R = 2^N - 1$$

For all I where $0 \leq I \leq BlkLen * BlkLen$

If $Z(I) < R$ then

Output the value $Z(I)$ as an integer has a length N bits {Putword ($Z(I)$, N)}

Else

Output the value of max as an integer has a length N bits {Putword (R , N)}

Output the value of $(Z(I) - R)$ as an integer has a length N_{max} bits

{Putword ($Z(I) - R$, N_{max})}

End if

End loop I

3.3.11 Shift Coding the Detail Coefficients

In this stage, all the determined RLE indices of the subbands (LH, HL, HH) are mapped to be positive, equation (3.13) have been used to convert the signed integer into positive integers. The mechanism applied to compute the optimal length is same like that illustrated in section (3.3.8). Algorithm (3.9) illustrates the implemented steps to perform

positive mapping step and to determine the optimal length of the shift codewords.

Algorithm (3.9) Mapping and S-Shift Optimizer

Input:

$A ()$ Vector consist RLE indices

L is the length of the vector $A ()$

Output:

Number of required bits (N_b, N_{bmax})

Steps:

"" Do the positive mapping

For all I where $0 \leq I \leq L$

If $A (I) > 0$ then $A (I) = 2 * A (I)$

Else If $A (I) < 0$ then $A (I) = 2 * \text{abs} (A (I)) + 1$

End loop I

"" Compute the maximum number of bits required to encode each DCT Coefficients

$Max = A(0)$

For all $A(i)$ coefficients (i) (where $0 \leq i \leq L$)

If $Max < A (i)$ then $Max = A (i)$

End loop i

"" Compute maximum number of required bits

$N_{max} = 1: K = 1$

While $Max \geq K$

$K = 2 * K + 1: N_{max} = N_{max} + 1$

Wend

To be continue

```

''' Compute the histogram  $H()$  of the RLE coefficients
Set  $H(i) = 0$ 
For all  $i$  where  $0 \leq i \leq \text{Max}$ 
     $j = A(i): H(j) = H(j) + 1$ 
End loop  $i$ 
''' Shift coding optimizer to compute the optimal number of required bits
( $N, N_{\text{max}}$ )
     $\text{MinBits} = N_{\text{max}} * L$ 
     $N = N_{\text{max}}$ 
    For all  $N$  where  $1 \leq N \leq N_{\text{max}} - 1$ 
         $R = 2^N - 1$ 
         $S_m = 0$ 
        For all  $J$  where  $0 \leq J \leq \text{Max}$ 
            If  $J < R$  then  $S_m = S_m + N * H(J)$ 
            Else  $S_m = S_m + (N + N_{\text{max}}) * H(J)$ 
        End loop  $J$ 
        If  $S_m < \text{MinBits}$  then  $\text{MinBits} = S_m$ 
    End loop  $N$ 

```

3.3.12 Shift Encoding the Detail Subband

In this stage, the input data are the mapped indices of (RLE) coefficients of (LH, HL, HH) subbands. The codewords produced by applying shift-coding are sent to the compression bitstream, which represents the compressed data file, see code list (3.8).

3.4 Decoding Unit

This unit consists of ten parts, as shown in figure (3.1b); it starts with loading the compressed data and ends with output the reconstructed image.

The constructed decoding process implies the following stages:

1. Loading the compressed data as one dimensional array of bits.
2. Decoding the compressed data of LL coefficients: (i) Shift decoding the quantized indices of DCT coefficients, (ii) deZigzag the quantized indices to convert to map the array of indices from 1D to 2D array, (iii) dequantize the quantization indices to get the DCT coefficients of the LL coefficients, and then (iv) apply IDCT to reconstruct the coefficients of LL subband.
3. Decoding the compressed data of LH, HL, HH coefficients: (i) shift decoding the quantized indices of RLE coefficients, (ii) run length decoding (RLD) the detail subbands to get the quantization indices, and then (iii) apply the hierarical dequantization process on the subbands (LH, HL, HH) to get the wavelet coefficients.
4. Inverse wavelet transform: apply the inverse wavelet transform on the constructed LL, LH, HL, HH coefficients to produce the coefficients of Y, U, and V color components.
5. Inverse color transforms from the three components Y, U, V to the color components (R, G, B) to produce the reconstructed image.

3.4.1 Shift Decoder and DeZigzag

Code list (3.10) illustrates the steps of the implemented shift decoder.

Algorithm (3.10) S-Shift Decoder and DeZigzag**Input:**

$A ()$ 1D array holds the shift-codewords of (LH, HL, HH) subbands

$Z ()$ is 1D array hold the shift-codewords of LL subband

N is number of bits of the first (short) shift codeword

N_s is number of bits of the second (long) shift codeword

L is the length of input array

Output:

$QD ()$ 2D array output (quantization indices and DCT coefficients codewords)

$A()$ holds the indices of RLE of (LH, HL, HH) subbands

Procedure:

$$R = 2^N - 1$$

"" Shift Decoder for LL subband

For all I where $0 \leq I \leq L$

$E = \text{GetBits}(N)$ "" load N bits from the compression stream

If $\text{abs}(E) < R$ then

$$\text{DA}(I) = E$$

Else

$$\text{DA}(I) = E + \text{GetBits}(N_s)$$

End if

End loop I

Load the scan order $[0, \dots, 63]$ array:

$\{0, 1, 8, 16, 9, 2, 3, 10, 17, 24, 32, 25, 18, 11, 4, 5, 12, 19, 26, 33, 40, 48, 41, 34, 27, 20, 13, 6, 7, 1$
 $4, 21, 28, 35, 42, 49, 56, 57, 50, 43, 36, 29, 22, 15, 23, 30, 37, 44, 51, 58, 59, 52, 45, 38, 31, 39, 4$
 $6, 53, 60, 61, 54, 47, 55, 62, 63\}$

For $I = 0$ to 63

$$X = \text{ScanOrder}(I) \bmod 8$$

$$Y = \text{ScanOrder}(I) \div 8$$

$$QD(x, y) = \text{DA}(I)$$

End loop I

```

''' Shift Decoder for LH, HL, HH subbands
For all I where 0 ≤ I ≤ L
    S = GetBits (N) ''' load N bits from compression stream
    If abs (S) < R then
        A (I) = S
    Else
        A (I) = S + GetBits (Ns)
    End if
End loop I

```

3.4.2 DeQuantization

Code list (3.11) illustrates the implemented steps to perform the dequantization of the DCT coefficients of the LL-subband.

Algorithm (3.11) Dequantization

Input:

$QD ()$ is a block of an 8×8 array of quantization indices of DCT coefficients

Q_{low} is the lowest value of quantization step of AC() coefficients

Q_{DC} is the quantization step of the DC coefficient

is the quantization rate parameter

Output:

$D ()$ is dequantized DCT coefficients

Procedure:

For all I_y where $0 \leq I_y \leq BlkLen$

For all I_x where $0 \leq I_x \leq BlkLen$

For all I where $0 \leq I \leq BlkLen$

For all J where $0 \leq J \leq BlkLen$

If $I = 0$ and $J = 0$ then

$$Q_q (I, J) = Q_{DC} * Q(I, J)$$


```

Else
   $Q_q = Q_{low} + (I + * (I + J))$ 
   $D(I, J) = Q_q * Q(I, J)$ 
End if
End loop J
End loop I
End loop Ix
End loop Iy

```

3.4.3 Inverse Discrete Cosine Transform (IDCT)

The inverse transform turns the quantized DCT coefficients to LL-subband coefficients. The code list (3.12) illustrates the implemented steps to perform the inverse DCT.

Algorithm (3.12) IDCT

Input:

$D()$ is dequantized DCT coefficients for LL subband

Output:

$B()$ is array of quantization indices for LL subband

Procedure:

Set $BlkSiz = 8$; $S_m = 0$

$C_0 = 1 / \text{Sqrt}(2)$

$C_1 = 3.14159 / (2 * BlkSiz)$

$C_2 = 1 / \text{Sqrt}(2 * BlkSiz)$

For all I where $0 \leq I \leq BlkSiz - 1$

For all J where $0 \leq J \leq BlkSiz - 1$

For all Y where $0 \leq Y \leq BlkSiz - 1$

For all X where $0 \leq X \leq BlkSiz - 1$

$WI = (2 * Y + 1) * J * C_1$

To be continue

```

W2 = (2 * X + 1) * I * C1
Sm = Sm + D (I, J) * Cos (W1) * Cos (W2)
If I = 0 Then Ci = C0 Else Ci = 1
If J = 0 Then Cj = C0 Else Cj = 1
    B(X, Y) = C2 * Ci * Cj * Sm
End loop X
End loop Y
End loop J
End loop I

```

3.4.4 Run Length Decoding

Code list (3.13) illustrates the implemented steps to perform the Run length decoding.

Algorithm (3.13) Run Length Decoding

Input:

$A ()$ is a vector consist of the type of the first run (i.e., zero or non zero), the length of zero runs, and the value of non-zero elements.

W is the image width

H is the image height

L length of vector $A ()$

Output:

$q ()$ is the array of quantization indices for (LH, HL, HH) subbands

Procedure:

$W1 = W: H1 = H$

For all I where $1 \leq I \leq NoPass$

$Wh = W1 \text{ div } 2: Hh = H1 \text{ div } 2$

To be continue

```

For J = 1 To 3
  F = -1 : M = 0
  If A (0) = 0 Then Flg = 0 Else Flg = 1
  While M < L
    M = M + 1 : N = A (M)
    If Flg = 0 Then
      For all K where 1 ≤ K ≤ N : F = F + 1 : V (F) = 0 : End loop K
    Else
      F = F + 1 : V (F) = N
    End If
    If Flg = 0 Then Flg = 1 Else Flg = 0
  Wend
  Select Case J
    Case 1 : X1 = Wh : X2 = W1 - 1 : Y1 = 0 : Y2 = Hh - 1
    Case 2 : X1 = 0 : X2 = Wh - 1 : Y1 = Hh : Y2 = H1 - 1
    Case 3 : X1 = Wh : X2 = W1 - 1 : Y1 = Hh : Y2 = H1 - 1
  End Select
  Flg = 0 : F = -1
  For all Y where Y1 ≤ Y ≤ Y2
    If Flg = 0 Then
      Xs = X1 : Xe = X2 : Stp = 1 : Flg = 1
    Else
      Xs = X2 : Xe = X1 : Stp = -1 : Flg = 0
    End If
    For all X where Xs ≤ X ≤ Xe : Step Stp
      F = F + 1 : q(X, Y) = V (F)
    End loop X
  End loop Y
End loop J
Wh = Wh / 2 : Hh = Hh / 2
End loop I

```

3.4.5 The Dequantization of Detail coefficients

Code list (3.14) illustrates the implemented steps to perform the dequantization of LH, HL and HH subbands coefficients.

Algorithm (3.14) Dequantization of Detail coefficients

Input:

$q()$ is the array of the quantization indices for (LH, HL, HH) subbands

W is the image width

H is the image height

Q is quantization step for (LH, HL, HH) coefficients

α, β is quantization parameters

Output:

$Dimg()$ is the dequantized wavelet coefficients

Procedure:

$W_m = W; H_m = H$

For all j where $1 \leq j \leq NoPass$

$W_{m1} = W_m; H_{m1} = H_m$

$W_m = (W_m + 1) \text{ div } 2$

$H_m = (H_m + 1) \text{ div } 2$

$X_s = W_m; X_e = W_{m1} - 1; Y_s = 0; Y_e = H_m - 1$

$Q_{step} = Q * \alpha^{j-1}$

For all x, y where $X_s \leq x \leq X_e$ and $Y_s \leq y \leq Y_e$

$Dimg(x, y) = q(x, y) * Q_{step}$

End loop x, y

$X_s = 0; X_e = W_m - 1; Y_s = H_m; Y_e = H_{m1} - 1$

For all x, y where $X_s \leq x \leq X_e$ and $Y_s \leq y \leq Y_e$

$Dimg(x, y) = q(x, y) * Q_{step}$

End loop x, y

To be continue

```

 $X_s = W_m : X_e = W_{m1} - 1 : Y_s = H_m : Y_e = H_{m1} - 1$ 
If  $j = 1$  then  $Q_{step} = Q_{step} * \beta$ 
For all  $x, y$  where  $X_s \leq x \leq X_e$  and  $Y_s \leq y \leq Y_e$ 
     $Dimg(x, y) = q(x, y) * Q_{step}$ 
End loop  $x, y$ 
End loop  $j$ 

```

3.4.6 Inverse Wavelet Transform

Code list (3.15) illustrates the steps of the implemented inverse wavelet transform.

Algorithm (3.15) Inverse Wavelet Transform

Input:

$Dimg()$ is the input array i.e., (wavelet coefficients)
 W is the image width
 H is the image height
 $NoPass$ is the number of wavelet transform levels or passes

Output:

$T()$ is the array of the color transform coefficients of $W * H$

Procedure:

```

 $W1 = W : H1 = H$ 
 $Hm = H1 - 1 : Wm = W1 - 1$ 
Put the two array  $B()$  of approximation coefficients and  $Dimg()$  of detail
coefficients in one array  $A()$ 
For all  $I$  where  $1 \leq I \leq NoPass - 1$ 
     $W1 = W1 \text{ div } 2 : H1 = H1 \text{ div } 2$ 
End loop  $I$ 

```

To be continue

```

For I = NoPass - 1 to 1 Step - 1
  Wh = W1 div 2: Hh = H1 div 2
  For all Y where 0 ≤ Y ≤ Hh - 1
    Y1 = 2 * Y: Y2 = Y1 + 1: Yy = Y + Hh
    For all X where 0 ≤ X ≤ Wh - 1
      X1 = 2 * X: X2 = X1 + 1: Xx = X + Wh
      T(X1, Y1) = (A(X, Y) + A(Xx, Y) + A(X, Yy) + A(Xx, Yy)) / 2
      T(X2, Y1) = (A(X, Y) + A(Xx, Y) - A(X, Yy) - A(Xx, Yy)) / 2
      T(X1, Y2) = (A(X, Y) - A(Xx, Y) + A(X, Yy) - A(Xx, Yy)) / 2
      T(X2, Y2) = (A(X, Y) - A(Xx, Y) - A(X, Yy) + A(Xx, Yy)) / 2
    End loop X
  End loop Y
  For all Y where 0 ≤ Y ≤ H1 - 1
    For all X where 0 ≤ X ≤ W1 - 1
      A(X, Y) = T(X, Y)
    End loop X
  End loop Y
  W1 = W1 * 2: H1 = H1 * 2
End loop I

```

3.4.7 Up Sampling

This stage implies the up sampling of (U, V) color components. The applied steps of this stage could be expressed by the following simple equations:

$$A(2x, 2y) = B(x, y) \dots\dots\dots (3.13)$$

$$A(2x + 1, 2y) = B(x, y) \dots\dots\dots (3.14)$$

$$A(2x, 2y + 1) = B(x, y) \dots\dots\dots (3.15)$$

$$A(2x + 1, 2y + 1) = B(x, y) \dots\dots\dots (3.16)$$

Where, A () is the up sampled component

B () is the down sample component

3.4.8 YUV to RGB Transform

After the reconstruction of (Y, U, V) color components, they must transformed to the space (R, G, and B), and then the output will be the reconstructed image.

Chapter Four

Tests and Results

4.1 Introduction

This chapter is devoted to present the results of some conducted tests to study the compression performance of the suggested image compression scheme, and to evaluate the effects of the involved coding parameters on the overall system performance. Some of the well known fidelity measures (i.e. MSE, PSNR) have been used to assess the quality of the reconstructed image.

The proposed compression scheme had been established using Visual Basic (version 6.0) programming language. The tests have been conducted by using personal computer (processor Pentium 4, 2.6 GHz), dual cash memory, and the operating system was windows XP.

4.2 Image Test Material

Two different images were adopted as test samples. The specifications of both test images are listed in table (4.1). Figure (4.1) presents these images.

Table (4.1) The test images attributes

Image	Type	Width (pixels)	Height (pixels)	Sample Resolution (bits/sample)
Lena	Bitmap	256	256	24
Baboon	Bitmap	256	256	24



a. Lena Image



b. Baboon

Fig (4.1) The original test images

4.3 Image Compression Performance Tests

The listed tables in this section illustrate the compression results of the proposed system. In these tables the values of MAE, MSE, PSNR, BitRate and compression ratio (CR) are listed. It is obvious that their values are significantly affected by the system parameters (i.e., number of wavelet passes, and the quantization parameters values). The effect of each system parameter was investigated separately, and the relevant test results have been listed in one of the following tables:

1. Table (4.2) shows the effects of the number of wavelet passes (NoPass) on the compression performance. Figures (4.2), (4.3) show some of the reconstructed images when the number of wavelet transform pass was varied between (1- 4).

Table (4.2) The effect of NoPass parameters

(Where, $Q_{High}=30$; $Q_{low}=3$; Alpha=0.6; Beta= 1.5; $\gamma=0.75$; BkLen=4)

Image	NoPass	MAE	MSE	PSNR	CR	BitRate
Lena	1	3.55	24.04	34.32	16.816	1.427
	2	4.72	41.02	32.00	29.248	0.821
	3	5.56	54.45	30.77	33.793	0.710
	4	6.06	63.12	30.13	34.092	0.704
Baboon	1	3.96	31.53	33.14	18.444	1.301
	2	5.52	54.25	30.79	33.631	0.714
	3	6.37	70.32	29.66	38.718	0.620
	4	6.80	79.13	29.15	39.056	0.615



NoPass = 1, CR = 16.82
MSE = 24.04
PSNR = 34.32
BitRate= 1.427



NoPass = 2, CR = 29.25
MSE = 41.02
PSNR = 32.00
BitRate=0.821

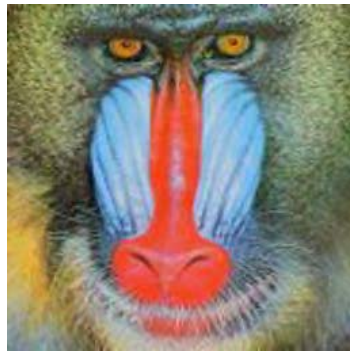


NoPass = 3, CR = 33.79
MSE = 54.45
PSNR = 30.77
BitRate=0.710



NoPass = 4, CR = 34.09
MSE = 63.12
PSNR = 30.13
BitRate=0.704

Fig (4.2) Samples of Lena reconstructed image when the number of wavelet passes is varied



NoPass = 1, CR = 18.44
 MSE = 31.53
 PSNR = 33.14
 BitRate=1.301



NoPass = 2, CR = 33.63
 MSE = 54.25
 PSNR = 30.79
 BitRate=0.714



NoPass = 3, CR = 38.72
 MSE = 70.32
 PSNR = 29.66
 BitRate=0.620



NoPass = 4, CR = 39.06
 MSE = 79.13
 PSNR = 29.15
 BitRate=0.615

Figure (4.3) Sample of Baboon reconstructed image when the number of wavelet passes is varied

2. Table (4.3) illustrates the effects of the quantization step (Q_{low}) on the compression performance, the value of Q_{low} was varied from 1 to 8. Figures (4.4), (4.5) show some of the reconstructed images.

Table (4.3) The effect of Q_{low}
 (Where, NoPass=3 Q_{High} =30; Alpha=0.6; Beta= 1.5; γ =0.75; BkLen=4)

Image	Q_{low}	MAE	MSE	PSNR	CR	BitRate
Lena	1	5.30	50.29	31.12	33.049	0.726
	2	5.43	52.15	30.96	33.477	0.717
	3	5.56	54.45	30.77	33.793	0.710
	4	5.73	57.13	30.56	34.027	0.705
	5	5.88	59.97	30.35	34.240	0.701
	6	6.06	63.13	30.13	34.384	0.698
	7	6.24	67.16	29.86	34.547	0.695
	8	6.50	71.90	29.56	34.718	0.691
	9	6.67	76.17	29.31	34.829	0.689
	10	6.90	81.06	29.04	34.934	0.687
	11	7.07	85.98	28.79	35.009	0.686
Baboon	1	6.14	65.74	29.95	37.729	0.636
	2	6.25	67.58	29.83	38.318	0.626
	3	6.37	70.32	29.66	38.718	0.620
	4	6.51	73.12	29.49	39.041	0.615
	5	6.67	76.13	29.31	39.267	0.611
	6	6.86	80.59	29.07	39.607	0.606
	7	7.03	84.50	28.86	39.791	0.603
	8	7.24	89.65	28.61	40.010	0.600
	9	7.48	95.74	28.32	40.149	0.598
	10	7.64	99.08	28.17	40.330	0.595
	11	7.80	102.96	28.00	40.463	0.593



Figure (4.4) Samples of Lena reconstructed image for different values of Q_{low}

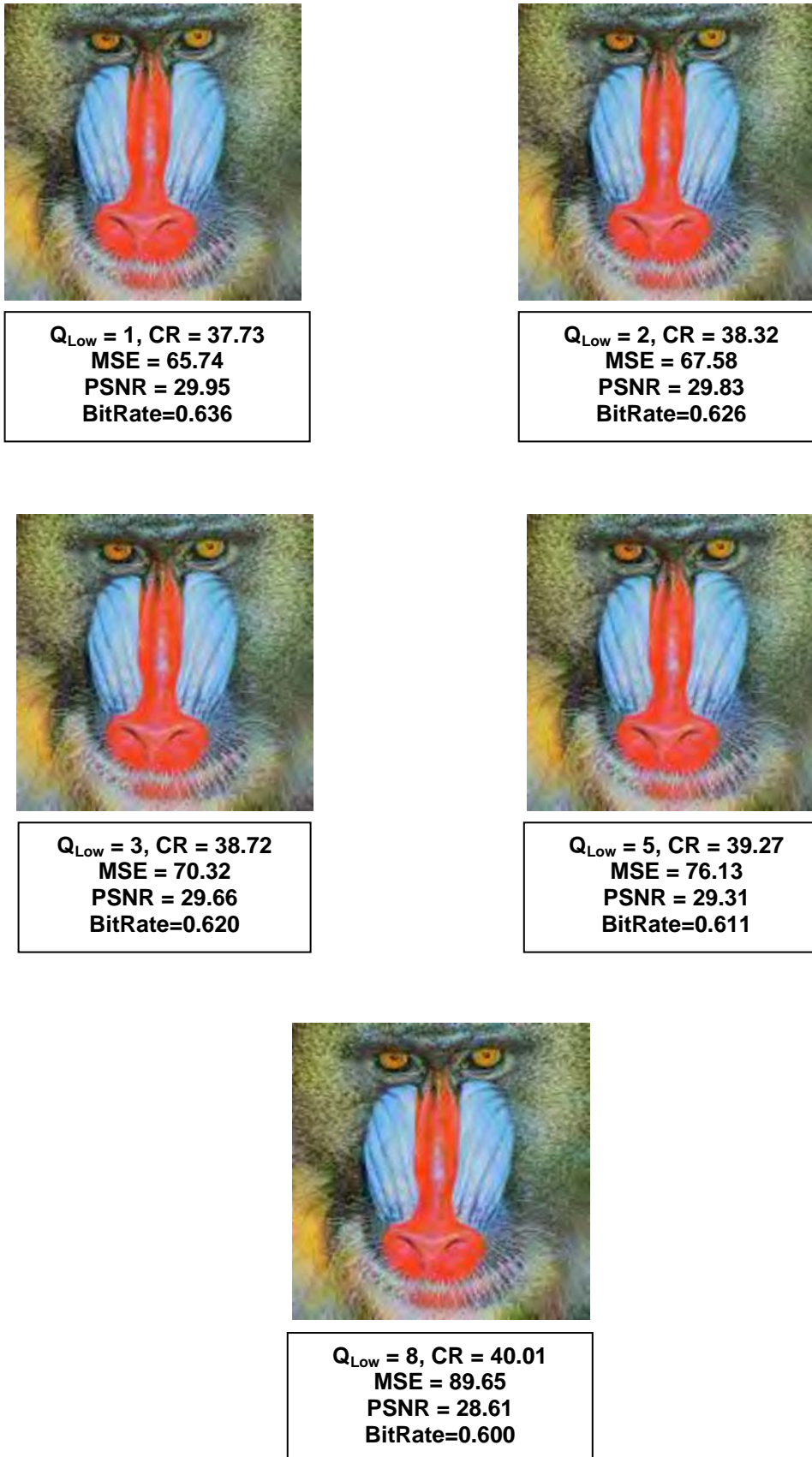


Figure (4.5) Samples of Baboon reconstructed image for different values of Q_{low}

3. Table (4.4) shows the effects of the parameter (Increment rate (γ)) on the compression performance. Figures (4.6) (4.7) show some of reconstructed images when the value of γ was varied from 0.25 to 2.

Table (4.4) The effect of (γ)

(Where, NoPass=3; $Q_{low} = 3$; $Q_{High} = 30$; $\alpha = 0.6$; BlkLen =4; $\beta = 1.5$)

Image	γ	MAE	MSE	PSNR	CR	BitRate
Lena	0.25	5.35	50.87	31.07	33.290	0.721
	0.50	5.44	52.22	30.95	33.574	0.715
	0.75	5.56	54.45	30.77	33.793	0.710
	1	5.70	56.72	30.59	33.974	0.706
	1.25	5.80	58.54	30.46	34.145	0.703
	1.50	5.92	60.74	30.30	34.240	0.701
	1.75	6.02	60.62	30.16	34.324	0.699
	2	6.18	62.11	29.93	34.444	0.697
	2.75	6.61	75.17	29.37	34.761	0.690
	3.5	7.06	85.54	28.81	34.915	0.687
Baboon	0.25	6.18	66.62	29.89	38.117	0.630
	0.50	6.27	67.85	29.82	38.453	0.624
	0.75	6.37	70.32	29.66	38.718	0.620
	1	6.46	72.18	29.55	38.917	0.617
	1.25	6.60	74.90	29.39	39.142	0.613
	1.50	6.74	77.71	29.23	39.298	0.611
	1.75	6.85	80.24	29.09	39.495	0.608
	2	7.02	84.38	28.87	39.647	0.605
	2.75	7.47	95.56	28.33	39.993	0.600
	3.5	7.85	106.07	27.87	40.322	0.595



Figure (4.6) Samples of Lena reconstructed image for different values of γ



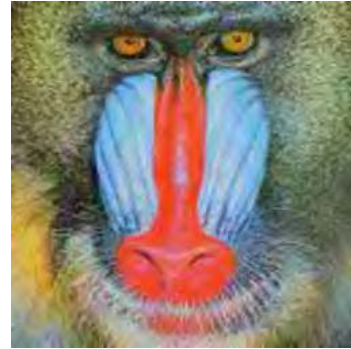
$\gamma = 0.25$, CR = 38.12
MSE = 66.62
PSNR = 29.89
BitRate=0.630



$\gamma = 0.75$, CR = 38.72
MSE = 70.32
PSNR = 29.66
BitRate=0.620



$\gamma = 1.25$, CR = 39.14
MSE = 74.90
PSNR = 29.39
BitRate=0.613



$\gamma = 1.75$, CR = 39.50
MSE = 80.24
PSNR = 29.09
BitRate=0.608



$\gamma = 2$, CR = 39.65
MSE = 84.38
PSNR = 28.87
BitRate=0.605

Figure (4.7) Samples of Baboon reconstructed images for different values of γ

4. Table (4.5) illustrates the effects of the quantization parameter (Q_{High}) on the compression performance parameters, the value of Q_{High} was varied from 5 to 50. Figures (4.8), (4.9) show some of reconstructed images.

Table (4.5) The effect of Q_{High}
(Where, NoPass=3; $Q_{low} = 3$; Alpha=0.6; Beta= 1.5; $\gamma=0.75$; BkLen=4)

Image	Q_{High}	MAE	MSE	PSNR	CR	BitRate
Lena	5	3.12	16.61	35.93	8.265	2.904
	10	3.66	22.64	34.58	13.926	1.723
	20	4.70	37.95	32.34	23.207	1.034
	30	5.56	54.45	30.77	33.793	0.710
	40	6.29	71.34	29.60	44.004	0.545
	50	6.93	87.38	28.72	55.633	0.431
Baboon	5	3.34	22.32	34.64	8.930	2.688
	10	4.01	29.98	33.36	14.200	1.690
	20	5.26	49.04	31.22	26.323	0.912
	30	6.37	70.32	29.66	38.718	0.620
	40	7.29	90.44	28.57	50.816	0.472
	50	8.14	112.92	27.60	62.734	0.383



$Q_{High} = 5$, CR = 8.27
MSE = 16.61
PSNR = 35.93
BitRate=2.904



$Q_{High} = 10$, CR = 13.93
MSE = 22.64
PSNR = 34.58
BitRate=1.723



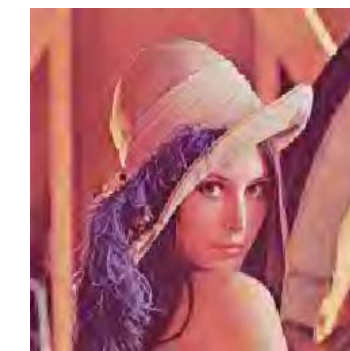
$Q_{High} = 20$, CR = 23.21
MSE = 37.95
PSNR = 32.34
BitRate=1.034



$Q_{High} = 30$, CR = 33.79
MSE = 54.45
PSNR = 30.77
BitRate=0.710



$Q_{High} = 40$, CR = 44.00
MSE = 71.34
PSNR = 29.60
BitRate=0.545

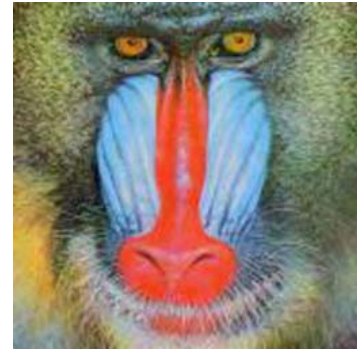


$Q_{High} = 50$, CR = 55.63
MSE = 87.38
PSNR = 28.72
BitRate=0.431

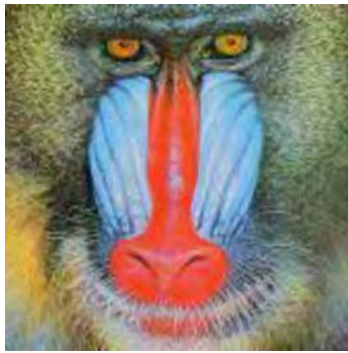
Figure (4.8) Samples of Lena reconstructed images for different Q_{High} values



$Q_{High} = 5$, CR = 8.93
MSE = 22.32
PSNR = 34.64
BitRate=2.688



$Q_{High} = 10$, CR = 14.20
MSE = 29.98
PSNR = 33.36
BitRate=1.690



$Q_{High} = 20$, CR = 26.32
MSE = 49.04
PSNR = 31.22
BitRate=0.912



$Q_{High} = 30$, CR = 38.72
MSE = 70.32
PSNR = 29.66
BitRate=0.620



$Q_{High} = 40$, CR = 50.82
MSE = 90.44
PSNR = 28.57
BitRate=0.472



$Q_{High} = 50$, CR = 62.73
MSE = 112.92
PSNR = 27.60
BitRate=0.383

Figure (4.9) Samples of the reconstructed Baboon images for different Q_{High} values

5. Table (4.6) shows the effect of the increasing the quantization parameter (α). Figures (4.10), (4.11) show some of reconstructed images when α was varies from 0.1 to 0.9.

Table (4.6) The effect of α

(Where, NoPasss=3; $Q_{low} = 3$; $Q_{High} = 30$; Beta= 1.5; $\gamma=0.75$; BkLen=4)

Image	α	MAE	MSE	PSNR	CR	BitRate
Lena	0.1	3.99	28.64	33.56	14.694	1.633
	0.2	4.16	30.88	33.23	18.387	1.305
	0.3	4.42	34.66	32.73	22.837	1.051
	0.4	4.77	40.11	32.10	26.800	0.896
	0.5	5.15	46.41	31.46	30.468	0.788
	0.6	5.56	54.45	30.77	33.793	0.710
	0.7	6.07	65.12	29.99	36.275	0.662
	0.8	6.55	76.06	29.32	38.994	0.615
	0.9	6.99	87.62	28.70	41.071	0.584
Baboon	0.1	4.35	36.15	32.55	15.083	1.591
	0.2	4.53	38.76	32.25	19.055	1.260
	0.3	4.83	42.97	31.80	23.977	1.001
	0.4	5.26	49.74	31.16	29.011	0.827
	0.5	5.75	58.08	30.49	33.927	0.707
	0.6	6.37	70.32	29.66	38.718	0.620
	0.7	7.03	84.17	28.88	43.574	0.551
	0.8	7.71	100.65	28.10	48.509	0.495
	0.9	8.46	121.29	27.29	52.809	0.454



$\alpha = 0.1$, CR = 14.69
MSE = 28.64
PSNR = 33.56
BitRate=1.633



$\alpha = 0.3$, CR = 22.84
MSE = 34.66
PSNR = 32.73
BitRate=1.051



$\alpha = 0.5$, CR = 30.47
MSE = 46.41
PSNR = 31.46
BitRate=0.788

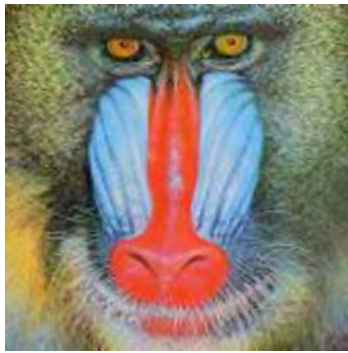


$\alpha = 0.6$, CR = 33.79
MSE = 54.45
PSNR = 30.77
BitRate=0.710

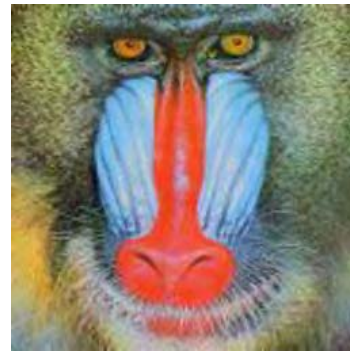


$\alpha = 0.8$, CR = 38.99
MSE = 76.06
PSNR = 29.32
BitRate=0.615

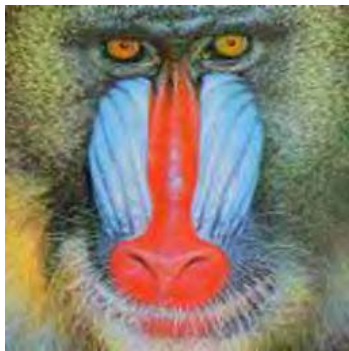
Figure (4.10) Samples of Lena reconstructed image for different values of α



$\alpha = 0.1$, CR = 15.08
MSE = 36.15
PSNR = 32.55
BitRate=1.591



$\alpha = 0.3$, CR = 23.98
MSE = 42.97
PSNR = 31.80
BitRate=1.001



$\alpha = 0.5$, CR = 33.93
MSE = 58.08
PSNR = 30.49
BitRate=0.707



$\alpha = 0.6$, CR = 38.72
MSE = 70.32
PSNR = 29.66
BitRate=0.620



$\alpha = 0.8$, CR = 48.51
MSE = 100.65
PSNR = 28.10
BitRate=0.495

Figure (4.11) Samples of Baboon reconstructed image for different values of

α

6. Table (4.7) shows the effects of the quantization parameter (β) on the compression performance. Figures (4.12), (4.13) show some of reconstructed images for different values of β -coefficient (i.e., from 1.1 to 1.9)

Table (4.7) The effect of β

(Where, NoPass=3; $Q_{low} = 3$; $Q_{High} = 30$; $\alpha = 0.6$; $\gamma=0.75$; BlkLen=4)

Image	β	MAE	MSE	PSNR	CR	BitRate
Lena	1.1	5.55	54.07	30.80	32.497	0.739
	1.3	5.56	54.27	30.78	33.306	0.721
	1.5	5.56	54.45	30.77	33.793	0.710
	1.7	5.57	54.67	30.75	34.306	0.700
	1.9	5.58	54.86	30.74	34.706	0.692
	2.5	5.59	55.21	30.71	35.431	0.677
	3.5	5.60	55.59	30.68	35.858	0.669
	4.5	5.60	55.70	30.67	36.015	0.666
Baboon	1.1	6.37	70.32	29.66	38.657	0.621
	1.3	6.37	70.32	29.66	38.702	0.620
	1.5	6.37	70.32	29.66	38.718	0.620
	1.7	6.37	70.33	29.66	38.725	0.620
	1.9	6.37	70.34	29.66	38.725	0.620
	2.5	6.37	70.34	29.66	38.771	0.619
	3.5	6.37	70.34	29.66	38.771	0.619
	4.5	6.37	70.34	29.66	38.771	0.619

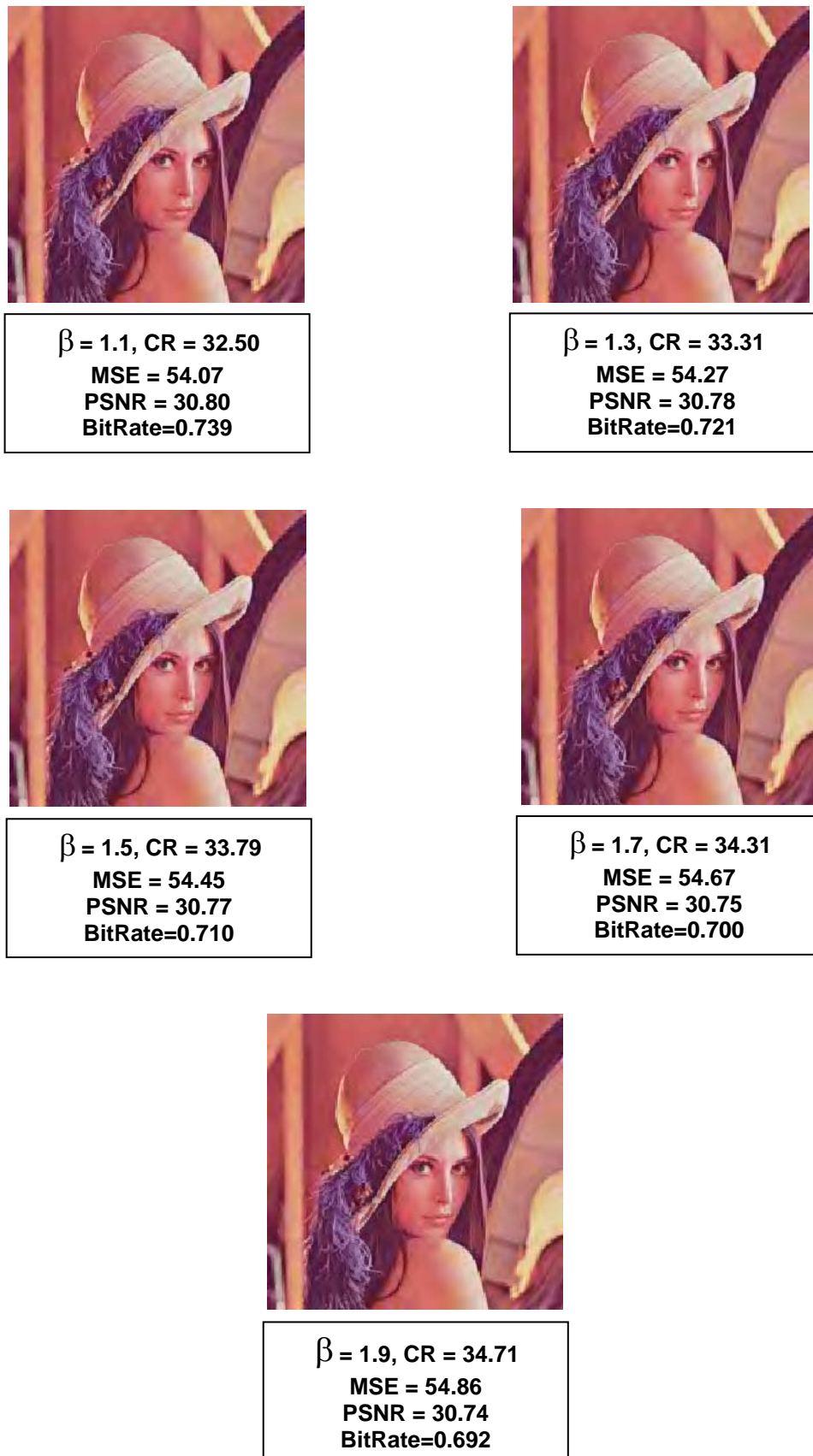


Figure (4.12) Samples of Lena reconstructed images for different values of β coefficient

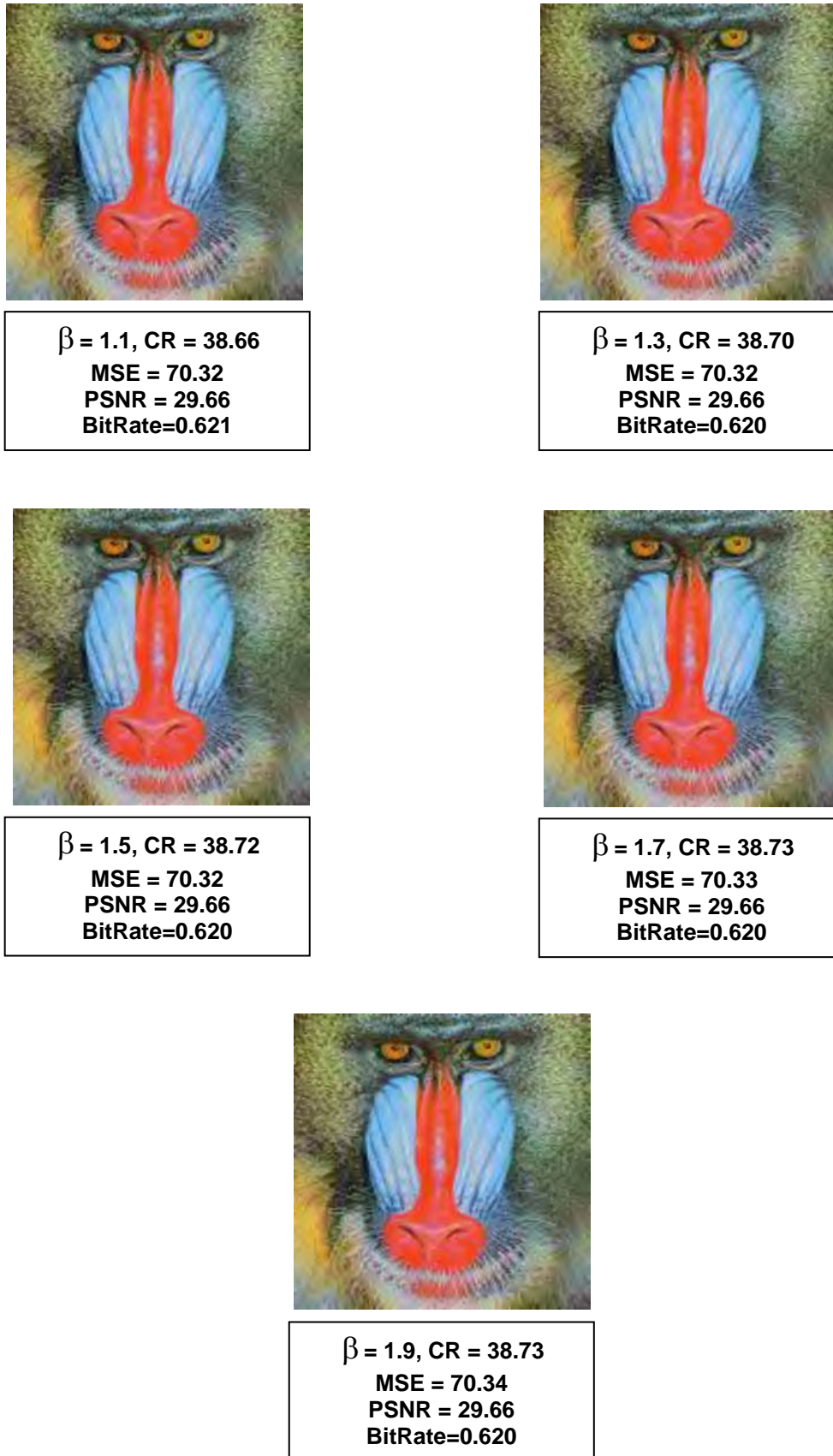


Figure (4.13) Samples of Baboon reconstructed images for different values of β coefficient

7. Table (4.8) shows the effects of the block length (BlkLen) on compression performance parameters. Figures (4.14), (4.15) show some of the reconstructed images, where BlkLen was varied from 2 to 8 pixels.

Table (4.8) The effect of block length

(Where, NoPass=3; $Q_{low} = 3$; $Q_{high} = 30$; $\alpha = 0.6$; $\gamma=0.75$; $\beta=1.5$)

Image	BlkLen	MAE	MSE	PSNR	CR	BitRate
Lena	2	5.39	51.48	31.01	33.155	0.724
	4	5.56	54.45	30.77	33.793	0.710
	8	6.17	65.52	29.97	33.927	0.707
Baboon	2	6.23	67.53	29.84	37.860	0.634
	4	6.37	70.32	29.66	38.718	0.620
	8	6.98	83.39	28.92	39.010	0.615



Figure (4.14) Samples of Lena reconstructed images for different block length values

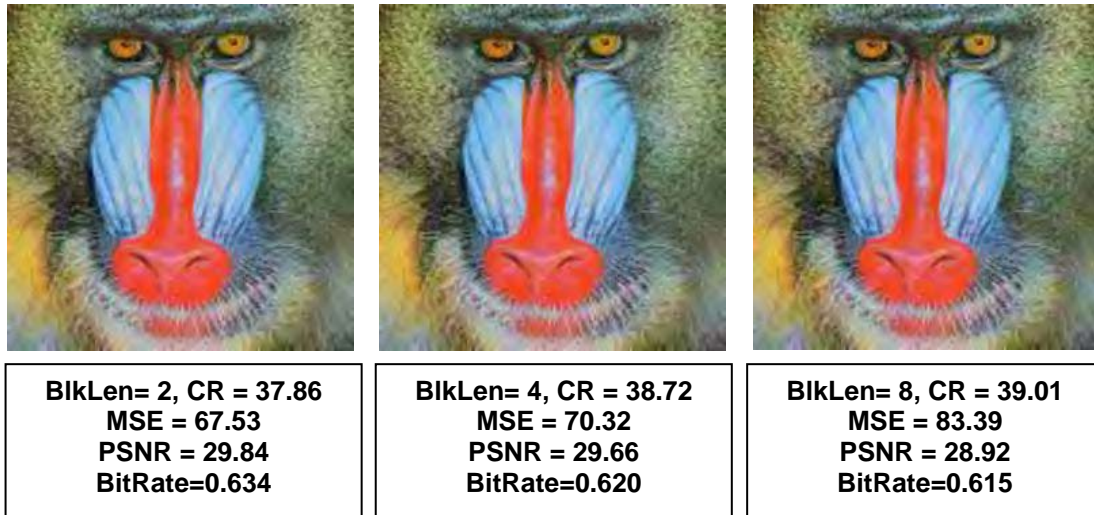


Figure (4.15) Samples of Baboon reconstructed images for different Block Length values

8. Table (4.9) shows the effects of increasing the quantization step (Q_{DC}) parameter of the DC-coefficient on the compression performance. Figures (4.16), (4.17) show some of reconstructed images, where Q_{DC} was varied from 1 to 8.

Table (4.9) The effect of Q_{DC} (Where, NoPass=3; $Q_{low} = 3$; $Q_{High} = 30$; $\alpha = 0.6$; BIkLen =4; $\beta = 1.5$; $\gamma = 0.75$)

Image	Q_{DC}	MAE	MSE	PSNR	CR	BitRate
Lena	1	5.55	54.25	30.79	33.648	0.713
	2	5.55	54.29	30.78	33.718	0.712
	3	5.56	54.43	30.77	33.752	0.711
	4	5.56	54.45	30.77	33.793	0.710
	5	5.58	54.68	30.75	33.810	0.710
	6	5.58	54.86	30.74	33.816	0.710
	7	5.61	55.28	30.70	33.851	0.709
	8	5.63	55.58	30.68	33.863	0.709
Baboon	1	6.36	70.00	29.68	38.520	0.623
	2	6.36	69.99	29.68	38.619	0.621
	3	6.37	70.19	29.67	38.649	0.621
	4	6.37	70.32	29.66	38.718	0.620
	5	6.37	70.24	29.67	38.733	0.620
	6	6.39	70.59	29.64	38.748	0.619
	7	6.39	70.72	29.64	38.786	0.619
	8	6.42	71.21	29.61	38.809	0.618



Q_{Dc} = 1, CR = 33.65
MSE = 54.25
PSNR = 30.79
BitRate=0.713



Q_{Dc} = 3, CR = 33.75
MSE = 54.43
PSNR = 30.77
BitRate=0.711



Q_{Dc} = 4, CR = 33.79
MSE = 54.45
PSNR = 30.77
BitRate=0.710



Q_{Dc} = 6, CR = 33.82
MSE = 54.86
PSNR = 30.74
BitRate=0.710



Q_{Dc} = 8, CR = 33.86
MSE = 55.58
PSNR = 30.68
BitRate=0.709

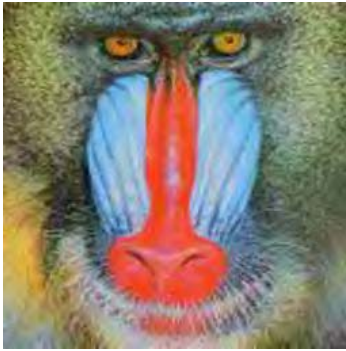
Figure (4.16) Samples of Lena reconstructed images for different values of Q_{Dc}



Q_{Dc} = 1, CR = 38.52
MSE = 70.00
PSNR = 29.68
BitRate=0.623



Q_{Dc} = 3, CR = 38.65
MSE = 70.19
PSNR = 29.67
BitRate=0.621



Q_{Dc} = 4, CR = 38.72
MSE = 70.32
PSNR = 29.66
BitRate=0.620



Q_{Dc} = 6, CR = 38.75
MSE = 70.59
PSNR = 29.64
BitRate=0.619



Q_{Dc} = 8, CR = 38.81
MSE = 71.21
PSNR = 29.61
BitRate=0.618

Figure (4.17) Samples of Baboon reconstructed images for different values of Q_{Dc}

4.4 Tables Notes

1. The increase in the number of wavelet passes (level) causes increase in compression ratio (CR), and decrease in quality.
2. The increase in quantization step Q_{low} causes decrease in the image quality and increase in CR, and vice versa.
3. The increase in quantization parameter γ causes decrease in the image quality and increase in CR, and vice versa.
4. The increase in quality factors α , β causes decrease in image quality and increase in CR, and vice versa
5. The increase in Low-Low block size causes increase in CR and decrease in image quality.
6. The increase in quantization step Q_{DC} causes increase in CR, and decrease in image quality.
7. The increase in quantization parameter Q_{High} causes increase in CR, and decrease in image quality.

Chapter Five

Conclusions and Future Works

5.1 Conclusions

From the test results presented in previous chapter, some remarks related to the behavior and performance of the investigated image coding scheme was stimulated. Among these remarks are the following:

1. The quantization parameters mainly affect MSE, PSNR, BitRate, and CR. It was found that the suitable quantization parameters values are ($Q_{low}=3$, $Q_{High}=30$, $\alpha=0.6$, $\beta=1.5$, $Q_{DC}=4$, $BlkLen=4$) they led to good PSNR (i.e., low distortion) and relatively high CR. For these values of quantization parameters, the value of PSNR is 30.77 dB, CR is 33.793 for Lena image, while they are PSNR=29.66 dB, CR=38.718 for Baboon Image.
2. The optimal values of the quantization parameters decrease with the increase of number of wavelet passes.
3. The effect of DCT on CR decreases with the increase of wavelet passes.
4. The time required to compress the image by this coding scheme is (0.7) second and decoding is (0.655) second.

5.2 Future Works

The following suggestions are introduced for future work:

1. Utilize Fractal Method with wavelet transform, such that it is applied on LH, HL, HH subbands.
2. Utilize Quadtree coding instead of RLE to compress the image.
3. Using another types of entropy coding (like, LZW coding) instead of shift coding.
4. The functionality of the proposed image compression could be extended from still image to Video. With growing market of multimedia applications this subject needs greater considerations.

References

- [Ado92], Adobe, "TIFF, Tag based Image File Format, Revision 6.0", <http://www.adobe.com/Support/TechNotes.html/>, 1992.
- [Aal96] Aalmoes, R., "Video Compression Techniques over Low-Bandwidth Line", M.Sc. thesis, Twente University, 1996.
- [Add00] Addel, J., "Lossy and Lossless Image Compression", Prentice Hall PTR, 2000.
- [Avc02] Avcbas, I., Memon, N., Sankur, B., and Sayood, K., "A Progressive Lossless/Near-Lossless Image Compression Algorithm", IEEE Signal Processing Letters, vol. 9, no. 10, pp. 312-314, October-2002.
- [BuAd83] Burt, P., Adelson, E., "The Laplacian Pyramid as a Compact Image Code", IEEE Trans. Computers, 31(4):532, 540, 1983.
- [Bet97] Bethel, D., "Optimization of Still Image Compression Techniques", PhD thesis, University of Bath, 1997.
- [Bur98] Burrus, C., "Introduction to Wavelets and Wavelet Transforms: A Primer, Prentice Hall", New Jersey, 1998.
- [ChOr98] Chrysafis, C., Ortega, A., "Line Based Reduced Memory Wavelet Image Compression". In Proc. IEEE Data Compression Conference, pages 398, 407, 1998.

- [Chr00] Chrysafis, C., "Wavelet Rate Distortion Optimization and Complexity Reductions", PhD thesis, University of Southern California, 2000.
- [Cro01] Crosswinds, "An Introduction to Image Compression", <http://www.crosswinds.net/~sskr/imagecmp/index.htm>, (current May 10, 2001)
- [Cab02] Cabeen, K., "Image Compression Using Distributed Systems", College of the Redwoods, 2002.
- [Dau88] Daubechies, I., "Orthonormal Bases of Compactly Supported Wavelets", 1988.
- [Dun99] Dunn, "Digital Color", 1999, <http://davis.wpi.edu/~matt/courses/color>
- [Dat01] Data Compression References Center, "RLE-Run Length Encoding", <http://www.rasip.fer.hr/research/compress/algorithms/fund/rl/index.html>, 2001.
- [DrLi01] Dror, I., Lischinski, D., "Fast Multi-Resolution Image Operations in the Wavelet Domain", Hebrew University, Israel, IEEE TVCG paper, pp.1-32, 2001.
- [Fri95] Frigaard, C., "Fast Fractal 2D/3D Image Compression", Report, Institute of Electronic System, Aalborg University, Laboratory of Image Analysis
- [FoRo98] Ford, A., Roberts, A., "Color Space Conversions", Reports, 1998, <http://WWW.poynton.com/PDFs/coloureq.pdf>

- [Gra95] Graps, A., "An Introduction to Wavelets", IEEE Computational Sciences and Engineering, Vol. 2, No. 2, pp 50-61, Summer 1995.
- [Gon00] Gonzales, R. C., and Woods, R. E., "Digital Image Processing", Addison-Wesley Publishing Company, 2000.
- [Grg01], Grgic, S., Grgic, M., "Performance Analysis of Image Compression Using Wavelets", Member, IEEE, 2001.
- [Gon02] Gonzalez, R., Woods, R., "Digital Image Processing", Pearson Education International, Prentice Hall, Inc. 2nd Edition, 2002.
- [Huf62] Huffman, D.A., "A method for Construction of Minimum Redundancy Codes", Proc. IRE, Vol 40, pp1016-1021, 1962.
- [HiJaSe94] Hilton, M., Jawerth, B., Sengupta, A., "Compressing Still and Moving Images with Wavelets", Multimedia Systems, Vol. 2 and No. 3, 1994.
- [Hub95] Hubbard, B., "The World According to Wavelets", A K Peters Wellesley, Massachusetts, 1995.
- [HaHo96] Hauf C., Houchin, J., "The FlashPix(TM) Image File Format". In Proceedings of the Fourth Color Imaging Conference: Color Science, Systems and Applications, pages 234,238, November 1996.
- [Hub96] Hubbard, B., "The World According to Wavelets", A.K Peters Ltd, Massachusetts, 1996.

- [Haf01] Hafren, U., "Image and Video Compression", publisher Linux Journal, 2001, <http://Ulli.linuxave.net/>
- [ITU93], ITU-T.4, "Standardization of Group 3 Facsimile Apparatus for Document Transmission", ITU-T Recommendation T.4. ITU, 1993.
- [Ibr04] Ibraheem, N., I., "Image Compression Using Wavelet Transform", M.Sc. thesis, Baghdad University, 2004.
- [Jia03] Jianyun, X., "Text Steganography Using Wavelet Transform", Dep. Computer Science, New Mexico Teach, Socorro, USA, 2003.
- [Kom94] Kominek, J., "Still Image Compression an Issue of Quality", Department of Computer Science, University of Waterloo, 1994, <http://www.links.uwaterloo.ca:/pub/Fractals/Papers/waterloo/kominek49b.xxx.ps.gz>
- [Kot04] Kotteri, A. K., "Optimal, Multiplierless Implementations of the Discrete Wavelet Transform for Image compression Applications", M.Sc. thesis, Virginia Polytechnic Institute and state university, 2004.
- [Mal89] Mallat, S., "A Theory for Multiresolution Signal Decomposition the Wavelet Representation", IEEE Trans, 1989.
- [Mul97] Mulcahy, C., "Image Compression Using The Haar Wavelet Transform", Spelman College Science & Mathematics Journal, Vol. 1, No.1, pp. 22-31, 1997.

- [Mah05] Mahmoud, G. A. H., "Robust Watermarking System Based on Wavelet Transform", M.Sc. thesis, University of Technology, 2005.
- [Nel96] Nelson, M., and Gailly, J. "The Data Compression Book", 2nd Edition, MRT Books, pp113-152, 1996.
- [Nis98] Nister, D., "Embedded Image Coding", Licentiate of Engineering, Chalmers University of Technology, 1998.
- [Nan03] Nanda, S., "Pre and Post Image processing for Enhanced image compression", M.Sc. thesis, Electrical Engineering, Florida International University, 2003.
- [OlMa06] Oliver, P., Malumbres, "Huffman Coding of Wavelet Lower Trees for Very Fast Image Compression", Computer Engineering (DISCA), Technical University of Valencia, 2006.
- [PeMi94] Pennebaker, W., Mitchell, J., "JPEG Still Image Data Compression Standard", Van Nostrand Reinhold, 1994.
- [Pol01] Polikar, R., "The Wavelet Tutorial", 2001, <http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html>
- [Rus06] Russ, J., "The Image Processing Handbook", Fifth Edition, 2006.
- [StDeSa94] Stollnitz, E., DeRose, T., Salesin, D., "Wavelets for Computer Graphics: A Primer", Tech. Report, Computer Science and Engineering Dept., Washington Uni. Seattle, 1994.

- [SaHo98] Sangwine, S., Horne, R., "The Color Image Processing Handbook", Chapman & Hall, 1998.
- [Sal98] Salomon, D., "Data Compression", Second Edition, Springer, New York, 1998.
- [Sch98] Schindler, M., "Practical Huffman Coding", <http://www.compressconsult.com/Huffman/>, 1998.
- [Str99] Strang, G., "The Discrete Cosine Transform", Society for Industrial and Applied Mathematics, 1999.
- [Say00] Sayood, K., "Introduction to Data Compression", Second edition, Academic Press, 2000.
- [ShSu00] Shi, Y., Sun, H., "Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards", CRC. Press, LLC., First Edition, 2000.
- [Sah01] Saha S., "Image Compression - from DCT to Wavelets", 2001, <http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html>.
- [Sal02] Salomon, D., "Data Compression the Complete Reference", Addison Wesley Company, Northridge California, 2002.
- [Tru99] Trulove, J., "Multimedia Networking Handbook", 1999.
- [Tan01] Tan, C., "Still Image Compression Using Wavelet Transform", Bachelor of Engineering, School of Information Technology and Electrical Engineering, University of Queensland, 2001.

- [Umb98] Umbaugh, S.E., "Computer Vision and Image Processing: A Practical Approach using CVIP Tools", Prentice Hall, Inc., 1998.
- [VeKo95] Vetterli, M., Kovacevic, J., "Wavelets and Subband coding", New Jersey, 1995.
- [Val01] Valens, C., "A Really Friendly guide to Wavelets", <http://perso.wanadoo.fr/polyvalens/clemens/wavelets/wavelets.html>, 2001.
- [Wan00] Wang, A., "Computer Basics", Dep. Information and Computer Science, California Irvin, <http://www.ics.uci.edu/~dan/pubs/compression Basics.html>, albertAcs.tut.fi, 2000.
- [Xia01] Xiong, P., "Image Compression by Wavelet Transform", M.Sc thesis, Department of Computer and Information Sciences, East Tennessee State University, 2001.
- [Zhe05] Zheng L., "Automated Feature Extraction and Content-Based Retrieval of Pathology Microscopic Images Using K-Means Clustering and Code Run-Length Probability Distribution", PhD Thesis, School of Information Sciences, University of Pittsburgh, 2005.

(DCT)

((RGB) , ,)

((YUV))

V U

)

(RLE)

(hierarchal uniform quantization)

((shift coding)

(uniform quantization)

)

(PSNR)

(MSE)

256x256

0.7

السيرة الذاتية

الاسم : سارة عدنان محمود كاظم الأنصاري

اسم الأطروحة : استخدام تحويل الجيب تمام لتشفير حزمة التحويل المويجي التقريبي

Using Discrete Cosine Transform to Encode Approximation Wavelet Subband

اسم المشرف : أ.م.د. لؤي ادور جورج

تاريخ المناقشة : 2008 / 4 / 13

من

قسم الفيزياء / كلية العلوم / جامعة النهريين

عنوان السكن : بغداد / حي الجامعة / م. 29 ز. 129 د. 25/3

الاميل : sara_alansari1984@yahoo.com

هاتف : 07901690089



جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة النهرين
كلية العلوم

استخدام تحويل الحيب تمام التفسير حزمة التحويل المويجي التقريبية

رسالة

مقدمة إلى كلية العلوم في جامعة
النهرين كجزء من متطلبات نيل درجة
الماجستير في الفيزياء

من قبل الطالبة

سارة عدنان محمود

(بكالوريوس جامعة النهرين ٢٠٠٥)

أشرف

د. لؤي أنور جورج