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

The Effect of Quantization On Image Compression Using Transform Coding

A Thesis
Submitted to College of Science of Al-Nahrain University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

By
Waffaa Shehab Ahmed Al-Mashhadani
(B. Sc. 2003)

Supervisors
Dr. Abdul Kareem A.-R. Kadhim **Dr. Taha S. Bashaga**

February 2007 **Safar 1428**

بسم الله الرحمن الرحيم

(يَرْفَعْ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ)

صدق الله العظيم

المجادلة(11)

Supervisor Certification

We certify that this thesis was prepared under our supervision at the Department of Computer Science/ College of Science/ Al-Nahrain University, by Waffaa Shehab Ahmed Al-Mashhadani As partial fulfillment of the requirement for the degree of Master of Science in Computer Science.

Signature:
Name: **Abdul Kareem A.-R. Kadhim**
Title: **Assistant Professor**
Date: / / **2006**

Signature:
Name: **Taha S. Bashaga**
Title: **Lecturer**
Date: / / **2006**

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

Signature:
Name: **Taha S. Bashaga**
Title: **Head of the Department of Computer Science, Al-Nahrain University.**
Date: / / **2006**

Certification of the Examination Committee

We chairman and members of the examination committee, certify that we have studied the thesis entitled "**The Effect of Quantization On Image Compression Using Transform Coding**" presented by the student **Waffaa Shehab Ahmed Al-Mashhadani** and examined her its content and in what is related to it, and we have found it worthy to be accepted for the degree of Master of Science in Computer Science.

Signature:
Name: **Dr. Laith Abdul Aziz Al-Ani**
Title: Assist. Prof.
Date: / /2007
(Chairman)

Signature:
Name: **Dr. Loay E. George**
Title: Assist. Prof.
Date: / /2007
(Member)

Signature:
Name: **Dr. Ban N. Thannon**
Title: Lecturer
Date: / /2007
(Member)

Signature:
Name: **Dr. Abdul Kareem A.-R. Kadhim**
Title: Assist. Prof.
Date: / /2007
(Supervisor)

Signature:
Name: **Dr. Taha S. Bashaga**
Title: Lecturer
Date: / /2007
(Supervisor)

Approved by the Dean of the Collage of Science, Al-Nahrain University
Signature:
Name: **Dr. Laith Abdul Aziz Al-Ani**
Title: Assist. Prof.
Date: / /2007

Dedication

To My Father Mother and To All My Family

Waffaa

Acknowledgment

First of all, my great thanks to Allah who helped me and gave me the ability to achieve this work.

I would like to express my deep gratitude and sincere thanks to my supervisors **Dr. Abdul-Kareem A.R.** and **Dr. Taha S. Bashaga** for their able guidance, supervision and untiring efforts during the course of this work.

Special thanks to the College of Science, dean of the college for the continuous support and encouragement during the period of my studies.

Grateful thanks for the Head of Department of Computer Science **Dr. Taha S. Bashaga,** staff and employees.

Special thanks to all my friends and group of M.Sc. in the Department of Computer Science.

Thanks to my family: my Father, Mother, Brothers, and Sisters without whose prays I would not be able to succeed in doing this work.

Waffaa

Abstract

One of the most important problems in computer applications are the storage and transmission of images that makes the field of developing the image compression. For that, various compression methods have been proposed using different techniques to achieve high compression ratios and high image quality. Among these techniques are the ***Wavelet Transform (WT) and Transform Coding (TC)*** methods.

In the wavelet transform method, which is the subject of this work, a wavelet transform may be used to divide the image into sub-bands. A type of wavelet transform is used here, the integer wavelet transform (IWT). The sub-band division can be repeated more than once (or the wavelet process can be of one or more passes). Then after the decomposition, the resultant wavelet coefficients are rounded to nearest integer to get the compressed form. In the second approach of the work, that is the transform coding, the image data is partitioned into blocks, and each block is transformed and then compressed. This method is studied and implemented for comparison reason.

Two types of images have been used for testing the result; they are either grayscale or color images. For grayscale image, the wavelet transform method achieved 1/1 to 3/1 compression ratios depending on number of compressed bits (b), while TC method achieved 3/1 to 7/1 compression ratios. For color images the WT method achieved 1/1 to 2/1 compression ratios. While TC achieved 4/1 to 7/1, with acceptable error. The obtained Peak Signal to Noise Ratio (PSNR) is well beyond 24 dB for both methods.

List of Abbreviations

2D	Two Dimensions
BMP	Bit-Map image file
C.R	Compression Ratio
dB	deci Bell
DCT	Discrete Cosine Transform
DWT	Discrete wavelet Transform
FHWT	Forward Haar Wavelet Transform
HVS	Human Vision System
HWT	Haar Wavelet Transform
IDCT	Inverse Discrete Cosine Transform
IHWT	Inverse Haar Wavelet Transform
IWT	Integer Wavelet Transform
JPEG	Joint Photographic Experts Group
MSE	Mean Square Error
PSNR	Peak Signal –to-Noise Ratio
Q	Quantizer
RGB	Red Green Blue
SBC	Sub-Band Coding
SNR	Signal -to- Noise Ratio
TC	Transform Coding
WT	Wavelet Transform
YC_bC_r	Weber- Fechner Color System

List of Symbols

Δ	Threshold value
b	Number of compressed bits
$\mathbf{G_{ij}}$	Transformed image
g	high pass filter
H(i)	High frequency sub-band
h	low pass filter
L(i)	Low frequency sub-band
$\mathbf{M_1}$	Minimum value
$\mathbf{M_2}$	Maximum value
N	Number of pixels
$\mathbf{P_{xy}}$	Original image
$\mathbf{V_i}$	Segment value
X	Image data
$\mathbf{X_c}$	Compressed image
x(n)	Original Data
Y	Reconstructed image
y(n)	Transformed coefficients
$\mathbf{y_{high}}$	Output of high pass filter
$\mathbf{y_{low}}$	Output of low pass filter
$\lfloor x \rfloor$	The greatest integer less than or equal to x (floor x)

Table of Contents

CHAPTER ONE

General Introduction

1.1 Introduction

As the amount of digital information was grown and increased, the need for efficient ways of storing and transmission this information were increased as well. Data compression is concerned with the means of providing an efficient way to represent information by using techniques to exploit different kinds of structures that may present in the data.

Data compression is the process of converting data files into smaller files to improve the efficiency of storage and transmission, (in other worlds, compression is the process of representing information in a compact form). As one of the enabling technologies of the multimedia revolution, data compression is a key to rapid progress being made in information technology. It would not be practical to put images, audio, and video data as there are on web sites without compression **[Xia01**].

In order to be useful, a compression algorithm has a corresponding decompression algorithm that reproduce the original file from the compressed one. Many types of compression algorithms were developed, these algorithms classified into two broad types: *lossless algorithms* and *lossy algorithms*. Alossless algorithm reproduces the exact original file. A lossy algorithm, as its name implies some loss of data, which may be unacceptable in many applications. The use of these methods depends on the data used, for example in text any loss of information cause an error in the text that means a loss of small information cause an error. So the text compression requires the losseless method. There are also many situations where loss may be either unnoticeable or acceptable. In image compression, for example, the exact reconstructed value of each sample of the image is not necessary. Depending on the quality required for the reconstructed image, varying amounts of information loss can be accepted.

Efficient image compression solutions are becoming more critical with the recent growth of data intensive and multimedia-based web application.

1.2 Image Compression

Image compression has been pushed to the forefront of the image processing fields. This is largely a result of the rapid growth in computer power, the corresponding growth in the multimedia market, and the advent of the World Wide Web (WWW), which makes the internet easily accessible for every one. Additionally, the advances in video technology create a demand for new, better and faster image compression algorithms. Compression algorithms development starts with applications to two-dimentional (2D) still images. Because video and television signals consist of consecutive frames of (2D) image data, the compression methods are developed for (2D) still images **[Umb98**].

Image compression involves reducing the size of image data files, while retaining the necessary information. The reduced file is called the compressed image file, while the original image is called the uncompressed image file **[Umb98]**.

One of the image compression methods is transforms based image compression using wavelet transform, which is main purpose of this thesis, also DCT method which of same type of compression method.

1.3 Literature survey

1. A.A.Al-Azraqi (1994)

M.Sc. thesis designed and implemented an image data compression system using Classified Vector Quantization (CVQ) technique. The image to be coded is decomposed into small blocks of size 4×4. Also, in this thesis, a complete implementation of CVQ is proposed in four stages. These are building training sets, codebook, encoder and decoder. Each stage is constructed from several methods.

2. A.P.Beegan (2001).

This thesis develops new models of the Human Vision System (HVS) and illustrates their performance for various scalar wavelet and multiwavlet transforms. The performance is measured quantitatively (PSNR) and qualitatively using the new perceptual testing procedure.

The results of extensive trials indicate that HVS model improves both quantitative and qualitative performance. For grayscale images, although the HVS scheme reduce PSNR (0.5 to 1.0 dB), it improves subjective quality as the auther claim. For color images, the HVS model improves both PSNR and subjective quality.

3. P. Xiao (2001).

M. Sc. thesis studies image compression with wavelet transform. As a necessary background, the basic concepts of graphical image storage and currently used compression algorithms are discussed. The mathematical properties of several types of wavelets, including Haar, Daubechies, and biorthogonal spline wavelets are covered and the Embedded Zerotree Wavelet (EZW) coding algorithm is introduced.

The conclusion reached in their thesis are:

a. The Haar wavelet transforms is the simplest one to implement, and it is the fastest. However, because of its discontinuity, it is not optimal to simulate a continuous signal. Based on experiments.

b. Haar wavelet obtained the worst compression result, which proves the above statement.

c. The EZW coding method is one of the more efficient quantization and coding methods for wavelet coefficients.

4. T.Z.Abdul Hameed (2003).

In this thesis the lossless compression is applied for multi media files (i.e. such as, image, audio, text). The implemented data compression algorithms are belong to three main classes: Run Length Encoding (RLE) techniques, Statistical techniques, and Dictionary techniques.

For each of the proposed approaches, an algorithm is constructed. These algorithms are tested by changing their parameters. A standard test files for given application are used in the tests. The Statistical Coder with order 3 model named Prediction with Partial string Matching (PPM) produced best compression ratio that the well known software WinZip(V.8) reduction ratio to about 3% on average and over 8% for best case.

5. A.A.Hussien (2003)

M. Sc. Thesis concerned with developing a system model for image compression based on fractals. The developed system consists of two major units; the first is the encoding unit and the second one is the decoding unit.

The main purpose of this thesis is to reduce the encoding time of this method. To do so, two approaches are proposed, the first is a new mathematical approach (called Improved Searching Mechansim ISM) that is responsible for determining the Iterated Function System IFS-codes between the range and domain blocks by reduction the total number of the searching opertion, while in the second approach (called Loosely Coupled Multiprocessing LCM), the encoding operations is excuted using loosely coupled multiprocessing system.

6. I.N.Ibraheem (2004).

In this thesis, image compression system had been applied by using wavelet transform (WT). The data after loading was divided into three layers red, green, and blue, each layer was processed individually. After applying two passes of the wavelet transform on each layer, the output of each layer was mapped into seven subbands or quadrants.

The spatial encoder used in this thesis are contain two coding methods which can be implemented on image data, the first spatial encoder is the adaptive run length encoder combined with classical fixed length coding method while the second encoder is the adaptive runlength encoder with merged hybrid coding method. These spatial encoders produce different code size. The optimizer will select the small of two sizes. The optimizer will select the small of two sizes and compared it with the original size (which is the size of bitplanes when it is coded by using, fixed codewords only), the optimizer will select the small one to store data.

The achieved compression ratio was between (2-15) with acceptable error, and the PSNR is more than 24dB, which is within the acceptable range.

1.4 Aim of Thesis

This thesis is concerned with the development of lossy image compression system. The aim here is to propose image compression algorithm that is based on integer wavelet transform by applying subband filter banks approach. The integer wavelet transform may be very useful due to its less operations required in the calculations, thus reducing the cost of the system. A comparison with the available schemes that use discrete Fourier transform for standard compression algorithm is to be performed. Different image types are to be considered in the work mainly Gray and color images.

1.5 Thesis Layout

The following are the outline of the thesis contents:

1. **Chapter Two:** Introduces a background to image compression, methods of compression, Discrete Cosine Transform (DCT) and Wavelet Transform (WT) with its types, uses and benefits.
2. **Chapter Three:** Presents the proposed compression with the obtained experimental results.
3. **Chapter Four:** Gives some conclusions and suggestions for future work.

CHAPTER TWO

Compression Methods

(Theoretical Background and Analysis)

2.1 Introduction

With the advance of the information age, the need for mass information storage and fast communication links was grown. Images are widely used in computer applications and storing these images in less memory leads to a direct reduction in storage cost and faster data transmission. These facts justify the efforts of a cadmic and industrial research centers and personal on new better performance image compression algorithms.

Images are stored on computers as collections of bits representing pixels or points forming the image elements. Since the human eye can process large amounts of information, many pixels are required to store moderate quality images. Most data contains some amount of redundancy, which can sometimes be removed for storage and replaced for recovery, but this redundancy does not lead to high compression ratios. An image can be changed in many ways that are either not detectable by the Human Vision System (HVS), or do not contribute to the degradation of the image. The standard methods of image compression come in several forms. This chapter concerned with exploring the popular compression methods, one of these methods is (Integer Wavelet Transform) IWT, which will be studied in details through this chapter, also a DCT transform method were introduced as a popular compression method to make a comparsion with IWT method which is the subject of this work.

2.2 Compression Methods

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

Based on the difference between original and reconstructed version, 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 other is lossy compression, which generally provides much higher compression than lossless compression but makes Y not exactly as X **[Add00].**

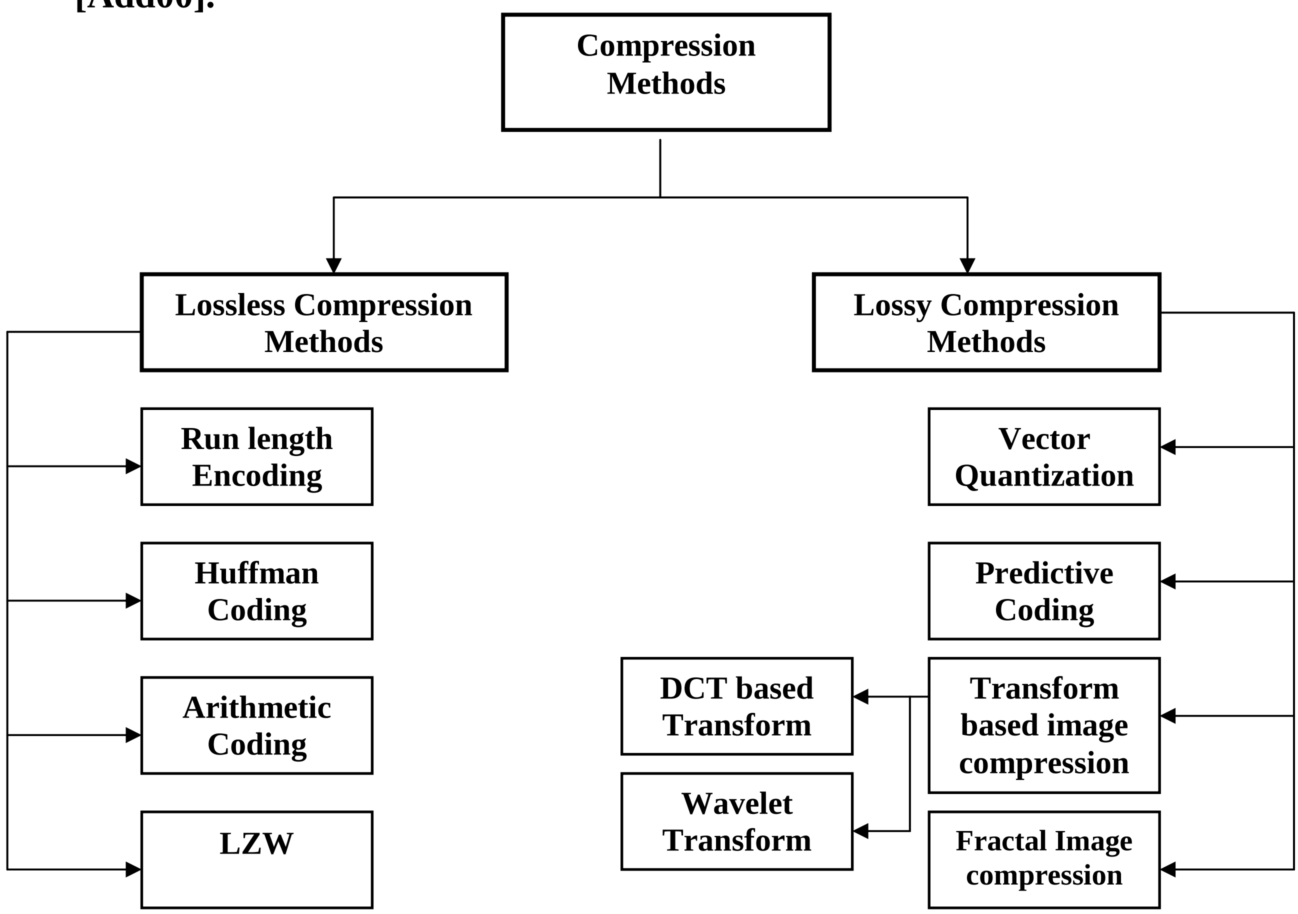


Figure 2.1:The Most Popular Image Compression Methods.

2.2.1 Lossless Compression Methods

Lossless compression techniques provide the guarantee that any pixel in the decompressed image is exactly the same as in the original one, i.e losseless 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: *Run length encoding, LZW, Arithmetic coding, Huffman coding, S-shift coding* **[Avc02]**.

2.2.2 Lossy Compression Methods

Lossy compression techniques involve some loss of information, and data cannot be recovered or reconstructed exactly. In some applications, exact reconstruction is not necessary. For example, it is acceptable that the reconstructed video signal is different from the original as long as the differences do not result in annoying artifacts. Generally, lossy compression can produce a higher compression ratio than is possible with lossless compression **[Bor03].** The most well known lossy compression method are:

a) **Vector Quantization**.

The definition of the term “quantization” is “to restrict a variable quantity to *discrete values* rather than to a continuous set of values”. In the field of data compression, quantization is used as follows: **[Sal00, Mor98]**

If the data samples to be *compressed* are large numbers samples, quantization is used to convert them to small numbers. Small numbers take less space than large ones, so quantization generates compression. On the other hand, small numbers generally contain less information than large ones, so quantization results in lossy compression. This aspect of quantization is used by several *image compression* methods **[Sal00].**

Quantization theory says that a quantizer can be modelled as the addition of a uniform distributed random signal e and the original unquantized signal X as shown in figure (2.2).

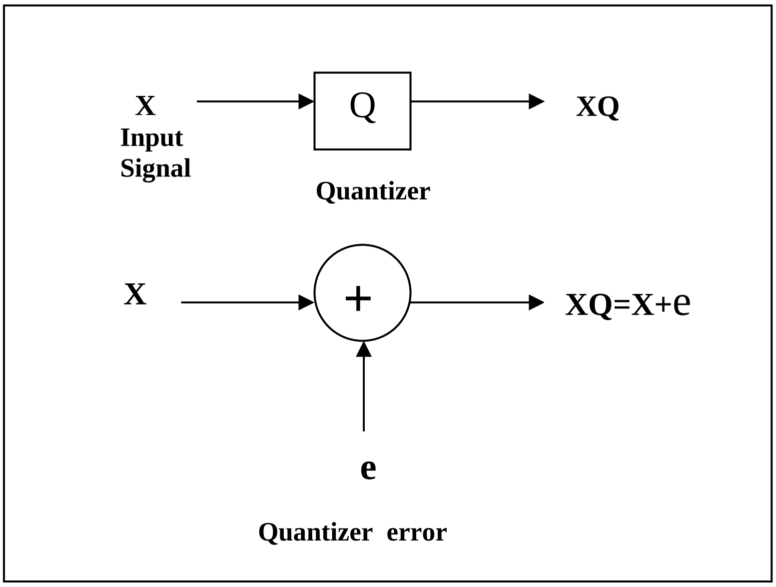


Figure 2.2:A Quantizer

There are two types of quantization: scalar Quantization and vector Quantization. In scalar quantization, each input pixel is treated separately in producing the output, while in vector quantization the input pixels are clubbed together in groups called vectors, and processed to give the output. This clubbing of data and treating them as a single unit increases the optimality of the vector quantizer, but at the cost of increased computational complexity [**Sal00**].

A quantizer can be specified by its input partitions and outputs levels (also called reproduction points). If the input range is divided into levels of equal spacing, then the quantizer is termed as a uniform quantizer, and if not, it is termed as a Non-Uniform Quantizer. A uniform quantizer can be easily specified by its lower bound and the step size. Also, implementing a uniform quantizer is easier than a non-uniform quantizer.

The quantization error (X-XQ) is used as a measure of the optimality of the quantizer and dequantizer **[Sal00].**

This work concerned with Transform Based Image Compression (TBIC).

b) Transform Based Image Compression

Transform based compression implies the most popular and efficient coding schemes. Combined with other compression techniques this technique allows efficient transmission, storage, and display of images that otherwise would be impractical **[Xia01].**

This basic transform encoding method for image compression works as follows:

i) **Image Transform:** Divide the source image into blocks and apply the transformations to each block.

ii) **Parameter Quantization:** The data generated by the transformation are quantized to reduce the amount of information. Quantization is irreversible operation because of its lossy property.

iii) **Encoding:** Encoding the result of the quantization. This last step can be error free by using Run Length encoding or Huffman coding. It can also be lossy if it optimizes the representation of the information to further reduce the bit rate.

2.3 Transformation Methods

These methods are used to map the signal from one domain representation to another (e.g. from the time domain to the frequency domain), these transformation methods are:

2.3.1 The Discrete Cosine Transform (DCT)

The DCT is a technique for converting a signal into elementary frequency components. It is widely used in image compression. It is popular transform used by JPEG (Joint Photographic Expert 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. JPEG-DCT is a transform coding method comprising 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 2D 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 decor relate pixels in the image since the cosine basis is orthogonal. Orthogonal waveforms are independent of each other and compact most image energy to a few transformed coefficients. Moreover, DCT coefficients can be loosely quantized according to some human visual characteristics **[Cab02, Tru99].**

The DCT formula is given by:**[Sal00]**

$$G_{ij} = \frac{1}{4} C_i C_j \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p_{xy} \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right) \qquad ,\ldots\ldots\ldots\ldots(2.1)$$

$$\text{Where } C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0, \\ 1, & f > 0, \end{cases}$$

and i= 0…N-1
j= 0…N-1

To turn the image back to its original domain the inverse transform must be applied to the transformed signal,

The Inverse DCT is given by:

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} C_i C_j G_{ij} \cos\left(\frac{(2x+1)i\pi}{2N}\right)\cos\left(\frac{(2y+1)j\pi}{2N}\right), \qquad ,\ldots\ldots\ldots\ldots(2.2)$$

Where p is the original image, x,y = 0,…N-1

G is the transformed image,

N is the number of pixels.

2.3.2 The Wavelet Transform

Wavelets are mathematical functions that cut up data into different frequency components, and then study each component with a resolution matched to its scale. They have advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Wavelets were developed independently in the fields of mathematics, quantum physics, electrical engineering, and seismic geology. Interchanged between these fields during the last twenty of years have led to many new wavelet applications such as image compression, turbulence, human vision, radar, and earthquake prediction **[Gra95].**

a) Discrete Wavelet Decomposition

A time-scale representation of a digital signal is obtained using digital filtering techniques. The heart of the DWT implies two filters h and g, low-pass and high-pass respectively. The block diagram of one level DWT is shown in figure (2.3). The one dimensional signal, x, is convoluted with high-pass filter to analyze the high frequencies, and it is convoluted with low-pass filter to analyze the low frequencies, and each result is down sampled by two, yielding the transformed signal $\mathbf{x_g}$ and $\mathbf{x_h}$. A DWT is obtained by further decomposing the low-pass output signal $\mathbf{x_h}$

by means of a second identical pair of analysis filters. This process my be repeated, and the number of such stages defines the level of the transform **[Bur98]**.

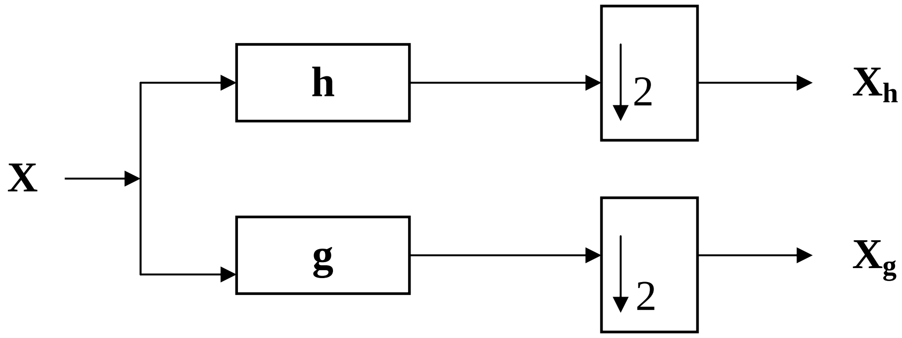


Figure 2.3: One level wavelet decomposition

The DWT analyze the signal at different frequency bands with different resolutions by decomposing the signal into coarse approximation and detail information. DWT employs two sets of functions, called the scaling function and wavelet function, which are associated with low-pass and high-pass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive low-pass and high-pass filtering of the time domain signal. The original signal x(n) is first passed through a half band high-pass filter g(n) and low-pass filter h(n). After filtering half of the samples can be eliminate. The signal can therefore be subsampled by two. These consitute one level of decomposition and can mathematically be expressed as follows:

$$\left.\begin{aligned} y_{high}(k) &= \sum_n x(n).g(2k-n) \\ y_{low}(k) &= \sum_n x(n).h(2k-n) \end{aligned}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.3)$$

where $y_{high}(k)$ and $y_{low}(k)$ are the output of the high-pass and low-pass filters respectively, after subsampling by 2.

The above procedure, which also known as subband coding, can be repeated for further decomposition. At every level, the filtering and subsampling will result in half the number of half of the number of samples (and hence half the time resolution). Figure (2.3) illustrate this procedure, where x(n) is the original signal to be decomposed, and h(n) and g(n) are low-pass and high-pass filters respectively **[Taw01].**

b) Wavelet Image Decompositions

This section discusses several ways for decomposing an image, each involving 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 sub bands to cover all the filter taps. The main decomposition types considered with wavelet transform are **[Sal00]**:

i) Morelet Decomposition

In this method (figure 2.4), the DWT is applied to each row of the image, resulting in smooth coefficients on the left (subband L1) and detail coefficients on the right (subband H1). Subband L1 is then partitioned into L2 and H2, and the process is repeated until the entire coefficient matrix is turned into detail coefficients. The wavelet transform is then 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 a decomposition method requires that the image rows be individually compressed **[Sal00]**.

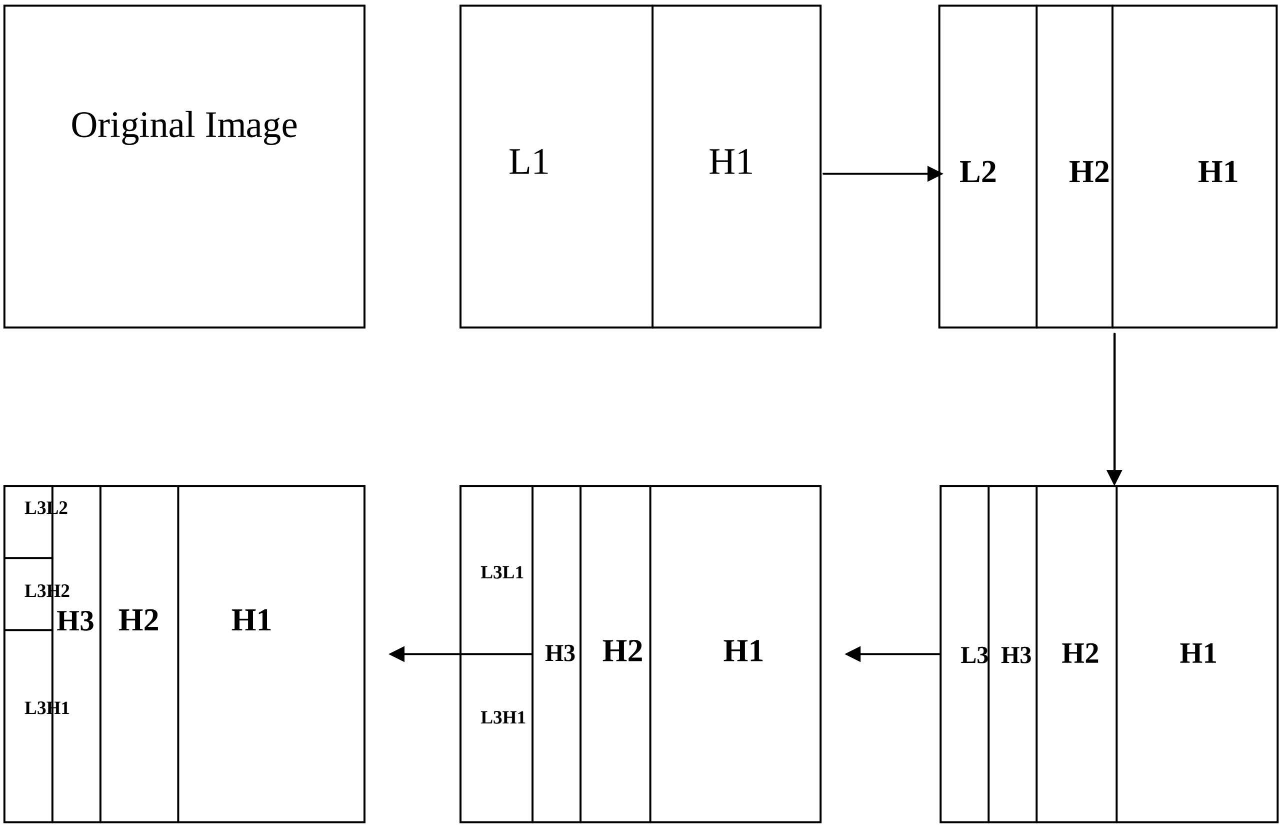


Figure 2.4: Morelet Wavelet Decomposition

ii) Quincunx Decomposition

Quincunx decomposition (figure 2.5), 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 the energy compaction of 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 [**Sal00]**.

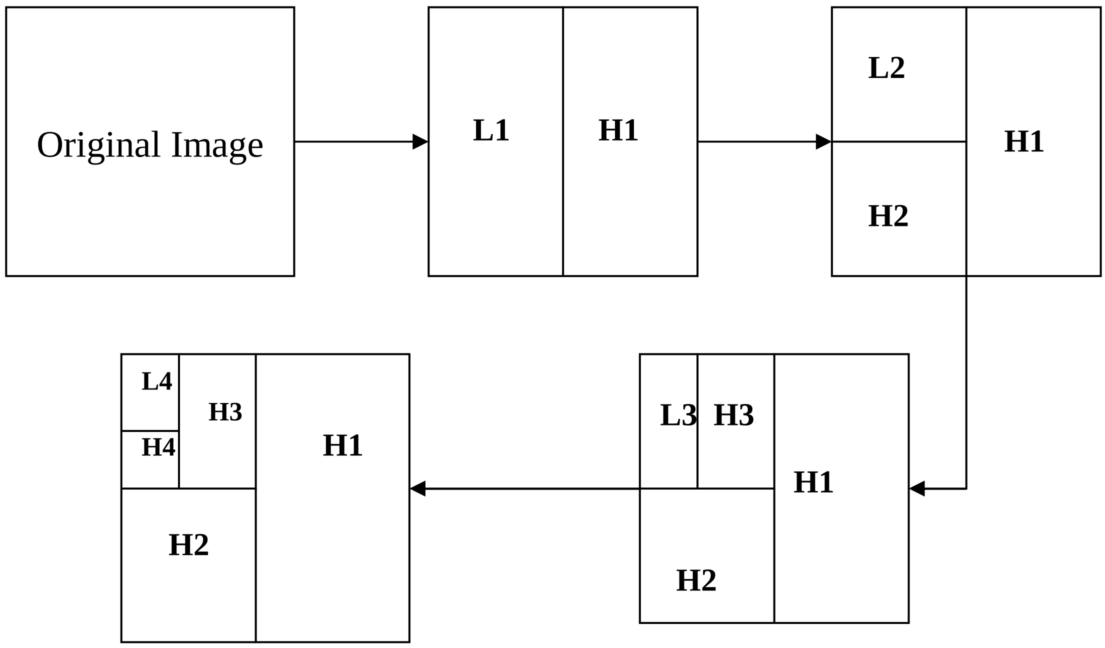


Figure (2.5): Quincunx wavelet decomposition

iii) Pyramid Decomposition

The pyramid decomposition (figure 2.6) 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 [**Sal00]**.

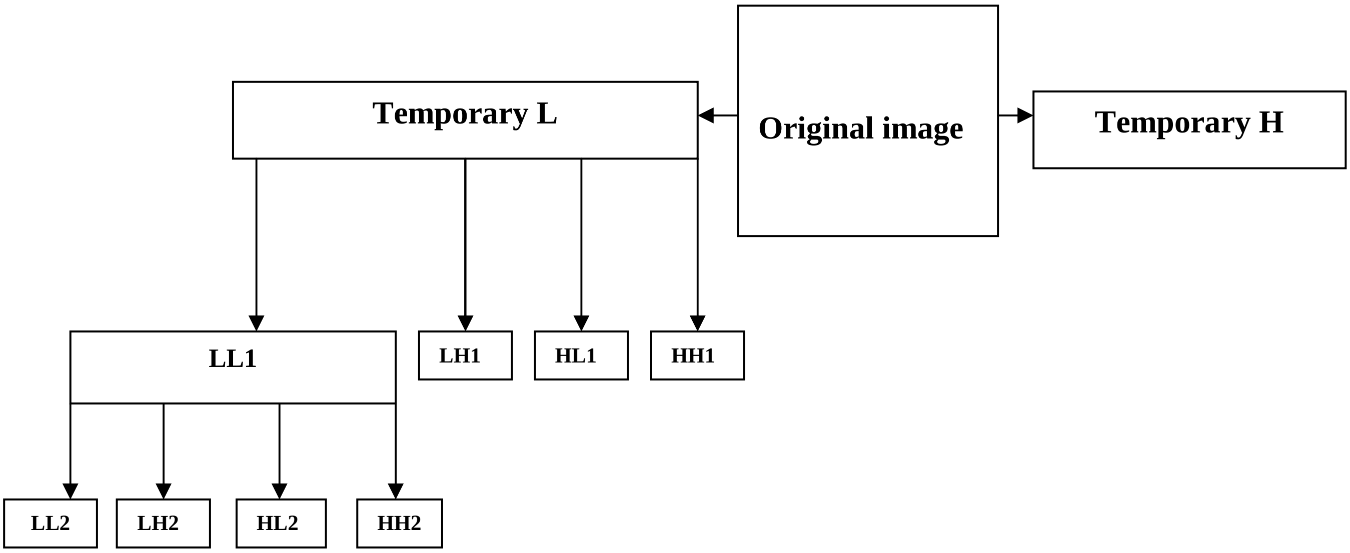


Figure 2.6: Pyramid Wavelet Decomposition

iv) Standard Decomposition

The first step in the standard decomposition (figure 2.7) is to apply whatever discrete wavelet filter is being used to all rows of the image, obtaining subbands L1 and H1. This is repeated on L1 to obtain L2 and H2, and so on k times. This is followed by a second step where a similar calculation is applied k times to the columns. If k=1, the decomposition alternates between rows and columns, but k may be greater than 1.

The result is to have one smooth coefficient at the top-left corner of the coefficient matrix. This method is somewhat similar to line decomposition. An important feature of standard decomposition is that when a coefficient is quantized, it may affect a long, 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 **[Sal00]**.

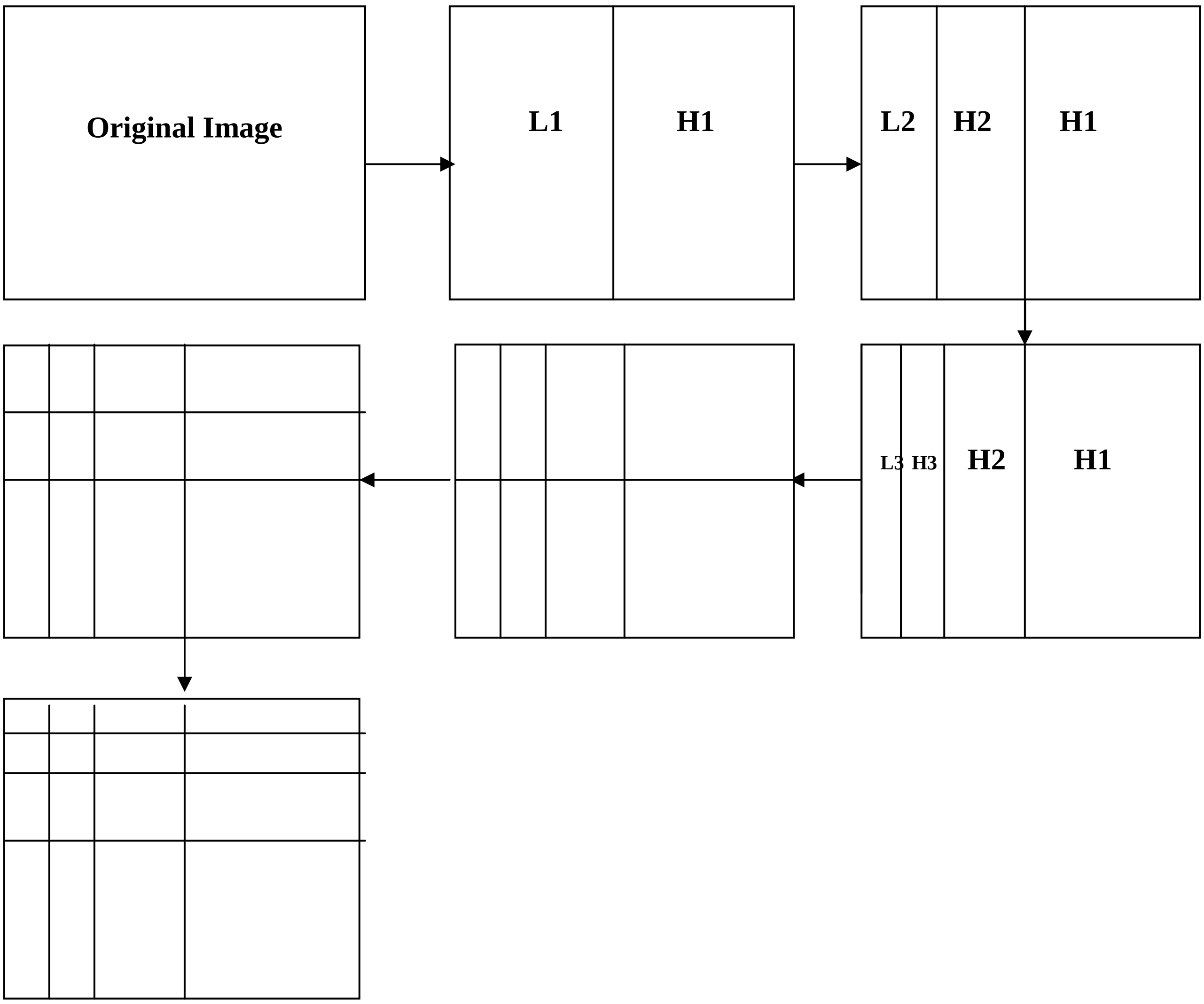


Figure 2.7: Standard Wavelet Decomposition

v) Full wavelet decomposition

This type of decomposition is also called the *Wavelet Packet* transform. It is shown in (figure 2.8).

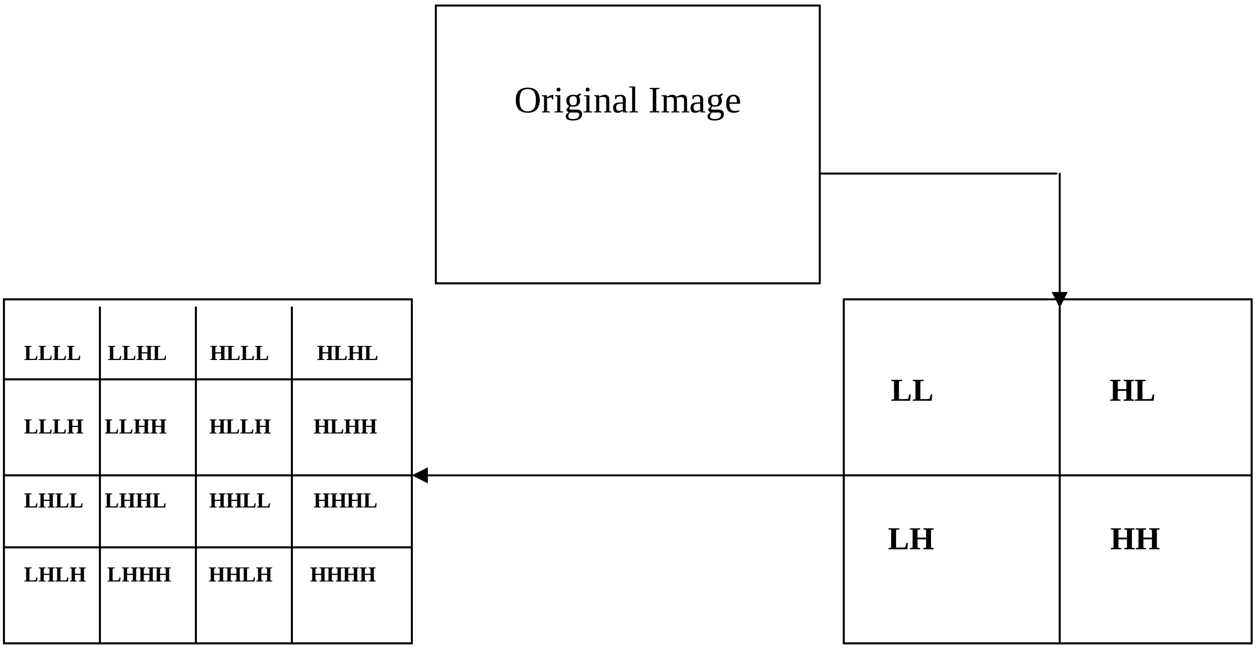


Figure 2.8:Full Wavelet Decomposition
(Wavelet packet of an image for one, two decomposition levels)

Denote the original image by I_0. It is assumed that its size $2^l \times 2^l$. After applying the 2-D discrete wavelet transform to 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) is then applied recursively to each of the four subbands individually. The result is a coefficient matrix I_2 consisting of 16 subbands. When this process is curried out *r* times, the result is a coefficient matrix consisting 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) and the other subbands contain detail coefficients. Each subband corresponds to s 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 **[Sal00]**.

c) Haar Wavelet Transform (HWT)

The oldest and most basic of the wavelet systems has constructed from the Haar basis function. The equations for forward Haar wavelet transform and inverse Haar wavelet transform, are given by:

i) Forward Haar Wavelet Transform (FHWT) [Jia03]

Given an input sequence (x_i) i=0…N-1, it is FHWT produce (L_i) i=0…N/2-1 and (H_i) i=0… N/2-1 by using the following transform equations:

1. If N is even

$$\left.\begin{aligned} L(i) &= \frac{x(2i)+x(2i+1)}{\sqrt{2}} \quad ,i=0\ldots(N/2)-1 \\ H(i) &= \frac{x(2i)-x(2i+1)}{\sqrt{2}} \quad ,i=0\ldots(N/2)-1 \end{aligned}\right\} \quad ,\ldots\ldots\ldots\ldots\ldots\ldots(2.4)$$

2. If N is odd

$$\left.\begin{aligned} L(i) &= \frac{x(2i)+x(2i+1)}{\sqrt{2}} \quad ,i=0\ldots(N-1)/2 \\ H(i) &= \frac{x(2i)-x(2i+1)}{\sqrt{2}} \quad ,i=0\ldots(N-1)/2 \\ L\left(\frac{N+1}{2}\right) &= x(N-1)\sqrt{2} \\ H\left(\frac{N+1}{2}\right) &= 0 \end{aligned}\right\} \quad ,\ldots\ldots\ldots\ldots\ldots\ldots(2.5)$$

ii) Inverse Haar Wavelet Transform (IHWT) [Jia03]

The inverse one-dimensional HWT is simply the inverse to those applied in the FHWT; the IHWT equation are:

i) If N is even

$$\left.\begin{aligned} x(2i) &= \frac{L(i)+H(i)}{\sqrt{2}} \qquad ,i=0\ldots N/2-1 \\ x(2i+1) &= \frac{L(i)-H(i)}{\sqrt{2}} \qquad ,i=0\ldots N/2-1 \end{aligned}\right\} \quad ,\ldots\ldots\ldots\ldots\ldots\ldots(2.6)$$

ii) If N is odd

$$\left.\begin{aligned} x(2i) &= \frac{L(i)+H(i)}{\sqrt{2}} \qquad ,i=0\ldots(N-1)/2 \\ x(2i+1) &= \frac{L(i)-H(i)}{\sqrt{2}} \qquad ,i=0\ldots(N-1)/2 \\ x(N-1) &= L\left(\frac{N+1}{2}\right)\sqrt{2} \end{aligned}\right\} \quad ,\ldots\ldots\ldots\ldots\ldots\ldots(2.7)$$

Where

N is the number of pixels.

L is the low frequencies subbands.

H is the high frequencies subbands.

d) Integer Wavelet Transform(IWT) [Maj97]

One level of IWT decomposes the signal into a low frequency part and high frequency part, both at lower resolutions. The low part can be used with the high part to reconstruct the original signal. This decomposition represents one level of the IWT.

Typically several levels of forward IWT can be computed, by iterating the procedure just described upon the low-frequency part (in a tree scheme), or reapply the procedure upon both low-frequency and high frequency parts (in a packet scheme).

In the next two paragraphs, we described one level of forward IWT and inverse IWT.

i) Two-Dimensional Forward IWT

Given an input sequence (x_i) i=0…N-1, its forward IWT (y_i) i=0…N-1, will also be an integer sequence, its computation depends on the length N of the integer sequence weather its even or odd.

If the signal length N is even (i.e. N=2k), then the integer transform sequence is computed by implementing the following two steps:

1. Determine the odd coefficients

$$\left.\begin{array}{ll} y_{2i+1} = x_{2i+1} - \lfloor (x_{2i} + x_{2i+2})/2 \rfloor & ,i=0\ldots k-2 \\ y_{2i+1} = x_{2i+1} - x_{2i} & ,i=k-1 \end{array}\right\} \quad ,\ldots\ldots\ldots\ldots\ldots\ldots(2.8)$$

2. Determine the even coefficients

$$\left.\begin{array}{ll} y_{2i} = x_{2i} + \lfloor y_{2i+1}/2 \rfloor & ,i=0 \\ y_{2i} = x_{2i} + \lfloor (y_{2i-1} + y_{2i+1})/4 \rfloor & ,i=1\ldots k-1 \end{array}\right\} \quad ,\ldots\ldots\ldots\ldots\ldots\ldots(2.9)$$

If the signal length N is odd (i.e. N=2k+1), then the integer transform is computed in the following two steps:

1. Determine the odd coefficients

$$y_{2i+1} = x_{2i+1} - \lfloor (x_{2i} + x_{2i+2})/2 \rfloor \quad ,i=0\ldots k-1,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.10)$$

2. Determine the even coefficients

$$\left.\begin{array}{ll} y_{2i} = x_{2i} + \lfloor y_{2i+1}/2 \rfloor & ,i=0 \\ y_{2i} = x_{2i} + \lfloor (y_{2i-1} + y_{2i+1})/4 \rfloor & ,i=1\ldots k-1 \\ y_{2i} = x_{2i} + \lfloor y_{2i-1}/2 \rfloor & ,i=k \end{array}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots(2.11)$$

ii) Two-Dimensional Inverse IWT

The inverse one-dimensional IWT is simply the inverse procedure from that applied when computing the forward one-dimensional IWT.

If the length N of the transformed signal y is even (i.e. N=2k), then the integer transform sequence x is computed in the following two steps:

1. Determine the even coefficients

$$\left.\begin{array}{ll} x_{2i} = y_{2i} - \lfloor y_{2i+1}/2 \rfloor & ,i=0 \\ x_{2i} = y_{2i} - \lfloor (y_{2i-1} + y_{2i+1})/4 \rfloor & ,i=1\ldots k-1 \end{array}\right\} \quad ,\ldots\ldots\ldots\ldots(2.12)$$

2. Determine the odd coefficients

$$\left.\begin{array}{ll} x_{2i+1} = y_{2i+1} + \lfloor (x_{2i} + x_{2i+2})/2 \rfloor & ,i=0\ldots k-2 \\ x_{2i+1} = y_{2i+1} + x_{2i} & ,i=k-1 \end{array}\right\} \quad ,\ldots\ldots\ldots\ldots(2.13)$$

If the length N of transform signal y is odd (i.e. N=2k+1), then inverse integer transform x is computed in the following two steps:

1. Determine the even coefficients

$$\left.\begin{array}{ll} x_{2i} = y_{2i} - \lfloor y_{2i+1}/2 \rfloor & ,i=0 \\ x_{2i} = y_{2i} - \lfloor (y_{2i-1} + y_{2i+1})/4 \rfloor & ,i=1\ldots k-1 \\ x_{2i} = y_{2i} - \lfloor y_{2i-1}/2 \rfloor & ,i=k \end{array}\right\} \quad ,\ldots\ldots\ldots\ldots(2.14)$$

2. Determine the odd coefficients

$$x_{2i+1} = y_{2i+1} + \lfloor (x_{2i} + x_{2i+2})/2 \rfloor \quad ,i=0\ldots k-1 \quad ,\ldots\ldots\ldots\ldots(2.15)$$

e) Wavelet Transform Charactersitics [Bur98]

Wavelet transform have proven to be very efficient and effective in analyzing a very wide class of signals and phenomena. The reasons for that are:

i) The size of the wavelet expansion coefficients drop off rapidly with its indices for a large class of signals. This property is being called *an unconditional basis* and it is why wavelets are very effective in signal and image compression, denoising, and detection.

ii) The wavelet expansion allows a more accurate description and separation of signal characteristics. A wavelet expansion coefficient representation a component that is itself local and is easier to interpret. The wavelet expansion may allow a separation of components of a signal that overlap in both time and frequency.

iii) Wavelets are adjustable and adaptable. Because there is not just one wavelet, they can be designed to fit individual applications. They are ideal for adaptive systems that adjust themselves to suit the signal.

iv) The generation of wavelets and the calculation of the discrete wavelet transform is well matched to the digital computer. The defining equation for a wavelet uses no calculus. There are no derivatives or integrals, just multiplication and additions operations that are basic to a digital computer.

2.4 Fidelity Measures [Umb98]

Fidelity measures can be divided into two classes:

a. Objective Fidelity Criteria.
b. Subjective Fidelity Criteria.

The objective fidelity criteria are borrowed from digital signal processing and information theory and provide us with equations that can be used to measure the amount of error in the reconstructed signal (image, sound or video).

Subjective fidelity criteria require the definition of a qualitative scale to asses signal quality. This scale can then be used by human test subjects to determine signal fidelity.

The commonly used objective measures are the Mean Square Error (MSE), Signal to Noise Ratio (SNR) and Peak Signal to Noise Ratio (PSNR).

The MSE is found by taking the summation of the square of the difference between the original and reconstructed signal and finally divide it by the total number of samples as shown below:

$$MSE = \frac{1}{size}\sum_{i=0}^{size-1}\left(R_i - P_i\right)^2 \qquad ,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.16)$$

Where

R= Reconstructed image.

P= Original image.

Size= number of image pixels.

The smaller value of MSE mean the better the reconstructed signal represents the original signal.

The SNR metrics consider the reconstructed signal to be the “image” and the error to be “noise”. The SNR can be defined as:

$$SNR = 10\log_{10}\left(\frac{\frac{1}{Size}\sum_{i=0}^{Size-1}P_i^{2}}{MSE}\right) \qquad ,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2.17)$$

Where

Size= number of pixels.

P= Original image.

The large value of SNR implies a better reconstructed image.

The PSNR metrics consider the “maximum peak value” and the error to be “noise”. The PSNR can be defined as:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \qquad ,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.18)$$

PSNR is like SNR where the large value means a better-reconstructed image represents the original image.

2.5 Compression Efficiency

Historically, there are two main types of applications of data compression: transmission and storage. The term “compression rate” comes from the transmission camp, while “compression ratio” comes from the storage camp.

$$CompressionRatio(CR) = \frac{Uncompressedfilesize}{Compressedfilesize} \quad \text{,.....................(2.19)}$$

$$CompressionRate = \frac{Compressedfilesize}{Numberofpixels} \quad \text{,.............................(2.20)}$$

Compression rate is the rate of the compressed data (which we imagined to be transmitted in “real-time”). Typically, it is in units of bits/sample, bits/character, bits/pixels, or bits/second. Reduction ratio is the ratio of the size or rate of the original data to the size or rate of the compressed data. For example, if a gray-scale image is originally represented by 8 bits/pixel (bpp) and it is compressed to 2 bpp, we say that the compression ratio is 4-to-1. Sometimes, it is said that the reduction ratio is 75%.

Compression rate is an absolute term, while compression ratio is a relative term **[Wan00]**.

Chapter Three
Proposed Compression Methods

3.1 Introduction

In this chapter image compression method based on the IWT will be discussed and implemented. The well known DCT compression method also implemented to compare the result obtained by the previous method which is the main objective of this project. The techniques used in this work is a sort of Wavelet Transform (WT) and Transform Coding (TC) using DCT.

In the wavelet transform the source image is fed into an analysis filter bank consisting of M bandpass filters which are contiguous in frequency so that the set of sub_band images can be recombined additively to produce the original image or a close version thereof. Each decimated filter output is quantized separately.

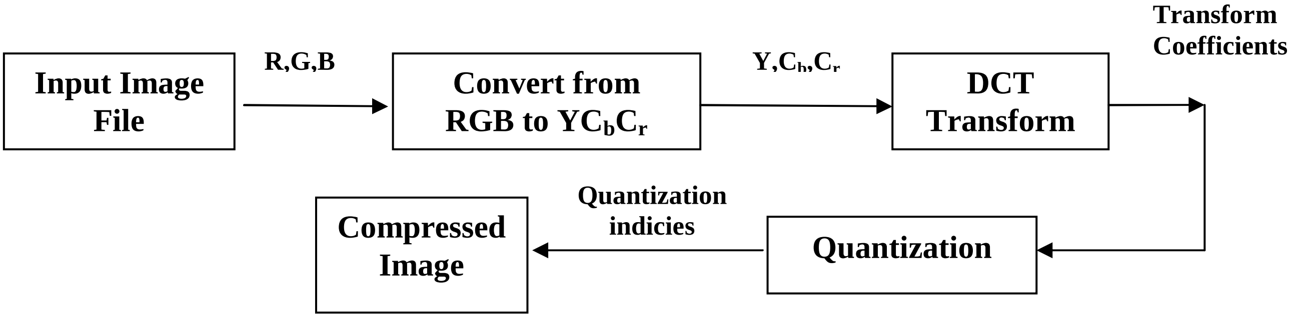
3.2 Transform Coding Compression Method

The Transform Coding compression method consists of two main phases the compression phase and decompression phase. In compression phase the image transforms from spacial domain to frequency domain and then compressed. In the second phase which is the reconstruction phase using the output of the first phase is used to reconstruct the original image by using inverse transform operation on compressed image.

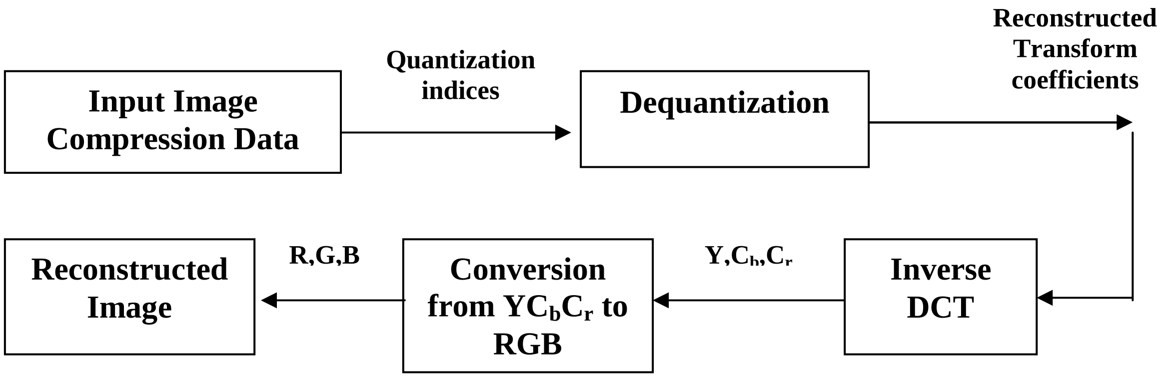
The full process is summerized in figure (3.1).

3.2.1 Compression Phase

After the required information is loaded as either colored or gray image, the color image then converted from RGB to YC_bC_r space. After that partition the data to blocks of pixels, each block can be transformed to the frequency domain, then quantizing the transform coefficients.



a. The block diagram of the compression process,



b. The block diagram of the decompression process

Figure 3.1: Transform Coding Compression Method

This phase as shown in figure (3.1a) which can be summerised into:

a. Convert from RGB to YC_bC_r

The input data (as image) given as two-dimensional array (H ×W), where H denotes the height of the image, and W denotes its width, the following steps have been implemented to convert the inputted image from RGB to YC_bC_r space, if the input image is color one, using the following equations are used **[Sal00]**:

$$\left.\begin{aligned}
Y &= (77/256)R+(150/256)G+(29/256)B \\
C_b &= -(44/256)R-(87/256)G+(131/256)B+128 \\
C_r &= (131/256)R-(110/256)G-(21/256)B+128
\end{aligned}\right\} \qquad \ldots\ldots\ldots\ldots\ldots\ldots(3.1)$$

Generally, YC_bC_r consist of a luminance component Y and two different chrominance components C_b and C_r. The Y component consists of the luminance and black & white image information, while C_b represents the difference between R and Y, and C_r represents the difference between Y and B. In YC_bC_r space, most of the image information is in the Y component. This representation is used during JPEG compression. The JPEG compression grossly removes large portions of the C_b and C_r components without damaging the image. The JPEG algorithm uses compression to reduce the Y component since it has more effect on the quality of the compressed image.

b. DCT Transform

By using the DCT transform described by equation (2.1), each block of the pixels will be transformed from the original time domain to the frequency domain. The following algorithm was utilized to DCT transform the array of image data samples:

Algorithm (1): DCT Transform

Input:

A: is a block of an 8×8 array of pixels

Output:

G: the transformed block of an 8×8 array of DCT coefficients

1. Set blocksize =8, Set C_1=1/sqrt(2), Set C_2=1/sqrt(2*blocksize)

2. For i=0 … blocksize-1

3. For j=0 … blocksize-1

To Continue

4. Set Sum=0
5. For x=0…blocksize-1
6. For y=0…blocksize-1
7. Set w=((2*y+1)*j*3.14159)/(2*blocksize)
8. Set w2=((2*x+1)*i*3.14159)/(2*blocksize)
9. Set Sum=Sum+A[x,y]*cos(w)*cos(w2)
10. end
11. end
12. If (i=0) Then $C_i=C_1$ Else $C_i=1$
13. If (j=0) Then $C_j=C_1$ Else $C_j=1$
14. Set $G[i,j]=C_2*C_i*C_j*Sum$
15. End.

c. Quantization

DCT based image compression relies on two techniques to reduce the data required to represent the image, the first is the ***quantization*** of the image's DCT coefficients; the second is ***coding*** the quantized coefficients.

Quantization is the process of reducing the number of bits needed to represent it **[Sal00]**.

<u>Algorithm (2): Quantization</u>

Input:
G: Block of DCT coefficients obtained from algorithm (1)
R: Integer factor ∈ [1…3]
Output:
QT: is block of an 8×8 array of quantized coefficients

1. Set blocksize=8
2. For i=0…blocksize-1
3. For j=0…blocksize-1

To Continue

4. Set Q[i,j]=1+(i+j)*R
5. Set QT[i,j]=round(G[i,j]/Q[i,j])
6. end
7. end
8. **End.**

d. **Output compressed image**

The outputs of the quantization algorithm will be the compressed image. Then after the completion process, the compression ratio will be computed.

3.2.2. Decompression phase

In Decompression the image is reconstructed image from the compressed image, and this phase is shown in figure (3.1b), which can be summerized into:

a. Dequantization

Dequantization is the inverse process of the quantization. The dequantization data are not identical to original data.

Algorithm (3): Dequantization

Input:

QT: Is a block of an 8×8 array of quantized coefficients

R: Integer factor ∈ [1…3]

Output:

DQ: is a block of an 8×8 array of Reconstructed coefficients

1. Set blocksize=8
2. For i=0…blocksize-1
3. For j=0…blocksize-1
4. Set Q[i,j]=1+(i+j)*R

To Continue

5. Set DQ[i,j]=QT[i,j]*Q[i,j]
6. end
7. end
8. **End.**

b. IDCT

The inverse transform turns the data back to its original representation in the time domain (refer to equation (2.2)).

The following algorithm was implemented to perform the process of the IDCT work in current system:

<u>Algorithm (4): IDCT</u>

Input:

DQ: is a block of an 8×8 array of reconstructed coefficients

Output:

A2: is a block transformed of an 8×8 array of IDCT coefficients

1. Set blocksize=8, Set Sum=0, Set C_1=1/sqrt(2), Set C_2=1/sqrt(2*blocksize)
2. For x=0…blocksize-1
3. For y=0…blocksize-1
4. For i=0…blocksize-1
5. For j=0…blocksize-1
6. Call Compute_Idct (DQ, Sum) //algorithm(5)
7. end
8. end
9. Set A2[x,y]=C_2*Sum
10. end
11. end
12. **End.**

Algorthim (5): Compute_Idct

Input:

DQ: Is block of an 8×8 array of dequantization coefficientss

Output:

Sum: Is represent the result of sumation

1. If (i=0) Then $C_i=C_1$ Else $C_i=1$
2. If (j=0) Then $C_j= C_1$ Else $C_j=1$
3. Set w=((2*x+1)*i*3.14159)/(2*blocksize)
4. Set w2=((2*y+1)*j*3.14159)/(2*blocksize)
5. Set Sum=Sum+C_i*C_j*DQ[i,j]*cos(w)*cos(w2)

c. **Convert from YC_bC_r to RGB**

In this step the image data is converted from YC_bC_r components to RGB components by using these equation **[Sal00]**:

$$\left.\begin{aligned} R&=Y+1.371*(C_r-128)\\ G&=Y-0.698*(C_r-128)-0.336*(C_b-128)\\ B&=Y+1.732*(C_b-128) \end{aligned}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.2)$$

d. Output the Reconstructed Image File

The output of this phase is the reconstructed image and in this step will compute the peak_signal to noise ratio (PSNR) of this image.

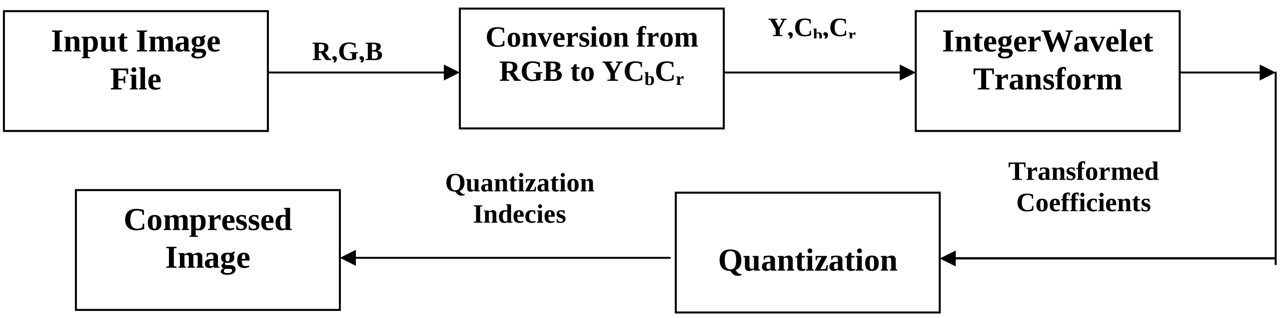
3.3 Wavelet Based Compression Method

The wavelet based compression method also consists of two main phases compression phase and decompression phase. In compression phase the source image is divided into number of sub bands by using IWT and then compressed. In the second phase, obtaining the original image from constructed one by using the inverse operation on compressed image.

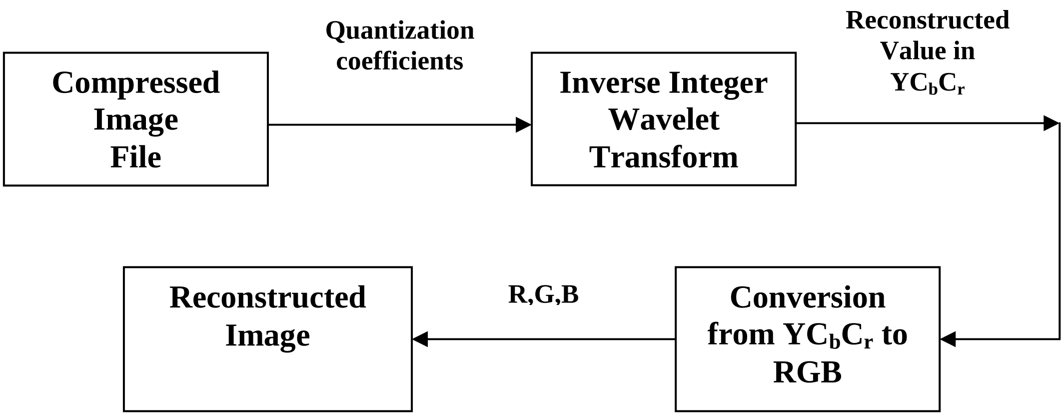
The proposed compression method is presented in figure (3.2) :

3.3.1 Compression Phase

After the required information loaded as gray image. The integer wavelet transform will be applied on the source image. After that quantize the transform coefficients.



a. The block diagram of compression processs



b. The block diagram of decompression process

Figure 3.2: Wavelet Based Compression Method

This phase as shown in figure (3.2a) which can be summerized into:

a. Integer Wavelet Transform (IWT)

The integer wavelet transform referred in equations(2.8…2.11) can be used to decompose an image. The transform is reversible i.e., the image can be fully reconstructed from the (integer) transform coefficients. This IWT can be used to compress the image lossily (by quantizing the transform coefficients).

Algorithm (6): Integer wavelet transform

Input:

X : a vector represent array of image data

j : Number of pixels in this vector

Output:

L : Low subband (approximation) coefficeints

H : High subband (detailed) coefficients

1. Set k=j/2
2. For i=0…k-2
3. Set Y[2i+1]=X[2i+1]-(X[2i]+X[2i+2]) div 2
4. Set Y[2k-1]=X[2k-1]-X[2k-2]
5. Set Y[0]=X[0]+(Y[1] div 2)
6. For i=1… k-1
7. Set Y[2i]=X[2i]+(Y[2i-1]+Y[2i+1]) div 4
8. Split the vector Y[] into L[] and H[], where L[]=Evens Y's, and H[]=Odd Y's
9. **End.**

There are several algorithms to decompose an image. From these algorithms will explain three algorithms**[Sal00]**:

Algorithm (7): Pyramid wavelet decomposition

Input:

Img: is (H×W) array of pixels

N: Number of pixel in block

Nolevel: Number of level decomposition

To Continue

Output:

WT: Array of wavelet coefficients

1. Set j=N
2. For h=1…Nolevel
3. For r=0…j-1
4. X= row r of Img
5. Call Integer wavelet transform (X, j, L, H1) // algorithm (6)
6. row r of Img=concatinate (L[ ] , H1[ ])
7. end
8. For c=j-1…0
9. X=column c of Img
10. Call Integer wavelet transform (X, j, L, H1) // algorithm (6)
11. column c of Img=concatinate (L[ ] , H1[ ])
12. end
13. Set j=j/2
14. end
15. WT=Img
16. **End.**

<u>Algorithm (8): Standard wavelet decomposition</u>

Input:

Img:Is (H×W) array of pixels

Nolevel: Number of level decomposition

Output

WT: Is (H× W) array of wavelet coefficients

1. Set j=H
2. For h=1…Nolevel
3. For r=0…H-1
4. X=row r of Img
5. Call Integer wavelet transform (X, j, L, H1) // algorithm (6)
6. row r of Img=concatinate (L[ ] , H1[ ])
7. end
8. Set j=j/2
9. end

To Continue

10. Set j=W
11. For h=1…Nolevel
12. For c=W-1…0
13. X=column c of Img
14. Call Integer wavelet transform (X, j, L, H1) // algorithm (6)
15. column c of Img=concatinate (L[ ] , H1[ ])
16. end
17. Set j=j/2
18. end
19. WT=Img
20. End.

Algorithm (9): Block_based wavelet transform

Input:

Img2: Is a block of an (16×16) array of pixels

Nolevel: Number of level decomposition

Output:

WT2: Is a block of an (16×16) array of wavelet coefficients

1. Call Pyramid wavelet decomposition (Img2, 16, Nolevel, WT2) // algorithm (7)
2. **End.**

b. Quantization

A uniform scalar quantizer was used to quantize the transform coefficients (data). The uniform scalar quantizer divides the total coefficient range into M uniform segments.

The quantization process computes the quantization index Q_i. In the current work, using the following equations applied to performed the uniform quantization:

$$\Delta = \frac{M_2 - M_1}{2^b} \qquad \text{……………………………………(3.3)}$$

Where:
M_1, M_2 are the minimum and maximum values for the (integer) WT coefficients.

Δ is the quantization step.

b is the number of compressed bits.

Algorithm (10): Quantization

Input:

WT :Subband from integer wavelet coefficients

Min,Max: The minimum and maximum values for WT-coefficients

b : Number of compressed bits

N : Number of subband pixels

Output

Q : Quantization indicies for the WT coefficients

1. Set $M=2^b$, Set D=(Max-Min)/M,

2. For j=0…N-1

3. Set i=round((WT[j]-Min)/D)

4. Set Q[j]=round(i*D+D/2)+Min

5. end

6. End.

c. Output Compressed Image

The output will be the compressed image and will compute the compression ratio to this image .

3.3.2 Decompression Phase

In decompression phase will reconstruct original image from compressed image.

This phase as shown in figure (3.2b) is summerized into:

a. Inverse integer wavelet transform

The inverse IWT algorithm referred in equations (2.12…2.15) can be used to reconstruct the image from (integer) wavelet transform coefficients.

Algorithm (11): Inverse integer wavelet decomposition

Input:

Y: Vector represent the transform coefficients

j: Number of pixels in this vector

Output:

X: a vector represents the reconstructed data

1. Set k=j/2
2. Set X[0]=Y[0]-(Y[1] div 2)
3. For i=1…k-1
4. Set X[2i]=Y[2i]-((Y[2i-1]+Y[2i+1]) div 4)
5. For i=0…k-2
6. Set X[2i+1]= Y[2i+1]+((X[2i]+X[2i+2]) div 2)
7. Set X[2k-1]=Y[2k-1]+X[2k-2]
8. End.

and the inverse decompose the image are **[Sal00]**:

Algorithm (12): Inverse Pyramid wavelet decomposition

Input:

WT: Is (H×W) array of wavelet transform coefficients

N: Number of pixel in block

Nolevel: Number of level decomposition

Output:

IWT: Is Inverse wavelet transform

1. Set $j=N/2^{Nolevel-1}$
2. For h=1…Nolevel
3. For c=j-1… 0

To Continue

4 Y=column c of WT
5. Call Inverse integer wavelet transform (Y, j, X) // algorithm (11)
6. column c of WT=X
7. end
8. For r=0 … j-1
9. Y=row r of WT
10. Call Inverse integer wavelet transform that (Y, j, X) // algorithm (11)
11. row r of WT=X
12. end
13. Set j=j*2
14. end
15. IWT=WT
16. End.

Algorithm (13): Inverse standard decomposition

Input:

WT:Is (H×W) array of wavelet transform coefficients

Nolevel: Number of level decomposition

Output

IWT: Array of inverse wavelet coefficients

1. Set $j=W/2^{Nolevel-1}$
2. For h=1…Nolevel
3. For c=W-1…0
4. Y=column c of WT
5. Call Inverse integer wavelet transform (Y, j, X) // algorithm (11)
6. column c of WT=X
7. end
8. Set j=j*2
9. end
10. Set $j=W/2^{Nolevel-1}$
11. For h=1…Nolevel
12. For r=0…H-1
13. Y=row r of WT

To Continue

14. Call Inverse integer wavelet transform (Y, j, X) // algorithm (11)
15. row r of WT=X
16. end
17. Set j=j*2
18. end
19. IWT=WT
20. End.

Algorithm (14): Inverse block_based wavelet decomposition

Input:

WT2: Is a block of an (16×16) of wavelet coefficients

Nolevel: Number of levels decomposition

Output:

IWT2: Is a block of an (16×16) of inverse wavelet coefficients

1. Call Inverse pyramid wavelet decomposition (WT2,16, Nolevel, IWT2) // algorithm (12)
2. End

b. Reconstructed Image

After implement the inverse wavelet transform, if the image is color is converted from YC_bC_r to RGB according to equation (3.2). The output will be the reconstructed image; the Peak Signal to Noise Ratio (PSNR) will be computed to this image.

3.4 Results Discussion

The IWT and transform coding based on DCT transform were used for some digital images. IWT and DCT applied on grayscale and true color.

The test images used in this work are some standard bit-map (BMP) images (Bit Map image file structure is explained in appendix A). These images as shown in figure (3.3) are five 8-bit grayscale images 256×256 pixels. They used for both methods. The other images as shown in figure (3.6) are three 24-bit true color images 256×256 pixels, they used in IWT and DCT transform method.

From tables (3.1, 3.2, 3.3 and 3.4) one can conclude that:

1. The PSNR value decreased and the C.R value increased depend on the decreasing the number of compressed bits (b), as shown in these tables.
2. From tables (3.1 and 3.2) the comparison among the three wavelet decomposition methods used that, the pyramid and standard methods when number of decomposition levels are two, performs better than the block based wavelet transform, because the PSNR value for two decomposition level obtained when applying the pyramid method is better than obtained when applying the wavelet packet.

Also figures (3.4) and (3.5) show the compression images after applying the IWT and DCT method on the gray scale images, and show the C.R, PSNR, bit per pixel (bpp) values to each image.

Figures (3.6) and (3.7) show the Compression images after applying the IWT and DCT method on the color images, and show the C.R, PSNR, bit per pixel (bpp) values to each image.

Table (3.1): The PSNR and Compression ratio (C.R) values to grayscale images with different decomposition levels, decomposition methods and number of compressed bits (b) in LLLL=8, LLLH=6, LLHL=6, LLHH=4, LH=4, HL=4, HH=2.

Image	No. Level	Wav. Dec Method	Compressed Size bit/pixel	C.R	PSNR (dB)
Lena	1	Pyramid	5.5	1.45	35.05
Lena	1	Standard	5.5	1.45	35.05
Lena	1	Block_based WT	5.19	1.53	42.77
Lena	2	Pyramid	4.86	1.64	30.37
Lena	2	Standard	4.87	1.63	30.16
Lena	2	Block_based WT	4.60	1.73	27.03
Cameraman	1	Pyramid	5.49	1.45	31.07
Cameraman	1	Standard	5.49	1.45	31.07
Cameraman	1	Block_based WT	5.21	1.53	40.16
Cameraman	2	Pyramid	4.85	1.64	26.41
Cameraman	2	Standard	4.99	1.60	26.36
Cameraman	2	Block_based WT	4.60	1.73	24.72
Barbara	1	Pyramid	4.77	1.67	32.14
Barbara	1	Standard	4.77	1.67	32.14
Barbara	1	Block_based WT	5.22	1.53	40.73
Barbara	2	Pyramid	4.19	1.90	28.22
Barbara	2	Standard	4.92	1.62	28.15
Barbara	2	Block_based WT	4.64	1.72	25.99
Boat	1	Pyramid	5.5	1.45	38.98
Boat	1	Standard	5.5	1.45	38.98
Boat	1	Block_based WT	5.18	1.54	44.47
Boat	2	Pyramid	4.84	1.65	30.29
Boat	2	Standard	4.90	1.63	30.16
Boat	2	Block_based WT	4.58	1.74	27.38
Girl	1	Pyramid	4.27	1.87	40.39
Girl	1	Standard	4.27	1.87	40.39
Girl	1	Block_based WT	5.27	1.51	44.38
Girl	2	Pyramid	3.68	2.17	33.05
Girl	2	Standard	4.99	1.60	32.84
Girl	2	Block_based WT	4.69	1.70	28.15

Table (3.2): The PSNR and Compression ratio (C.R) values to color images with different decomposition levels, decomposition methods and number compressed bits (b) in LLLL=8, LLLH=6, LLHL=6, LLHH=4, LH=4, HL=4, HH=2.

Image	No. Level	Wav. Dec Method	Compressed Size bit/pixel	C.R	PSNR (dB)
Horses	1	Pyramid	14.12	1.69	36.76
Horses	1	Standard	14.12	1.69	36.76
Horses	1	Block_based WT	14.77	1.62	41.11
Horses	2	Pyramid	12.19	1.96	30.33
Horses	2	Standard	13.39	1.79	30.25
Horses	2	Block_based WT	12.81	1.87	26.31
Peppers	1	Pyramid	16.02	1.49	32.98
Peppers	1	Standard	16.02	1.49	32.98
Peppers	1	Block_based WT	15.03	1.59	40.86
Peppers	2	Pyramid	14.23	1.68	30.97
Peppers	2	Standard	14.66	1.60	30.86
Peppers	2	Block_based WT	13.08	1.83	26.93
Mandrill	1	Pyramid	16.5	1.45	25.26
Mandrill	1	Standard	16.5	1.45	25.26
Mandrill	1	Block_based WT	16.16	1.48	33.25
Mandrill	2	Pyramid	14.92	1.60	21.15
Mandrill	2	Standard	14.88	1.61	28.15
Mandrill	2	Block_based WT	14.52	1.65	20.34

Table (3.3): The PSNR and Compression ratio (C.R) values to gray scale images with different number of compressed bits (b) for DCT method.

Image	Number of compressed bits (b)	Compressed Size bit/pixel	C.R	PSNR (dB)
Lena	8	2.18	3.66	35.58
Lena	6	1.33	6.01	29.04
Cameraman	8	2.18	3.66	31.74
Cameraman	6	1.30	6.10	27.29
Barbara	8	2.28	3.50	31.20
Barbara	6	1.32	6.02	27.57
Boat	8	2.06	3.86	34.74
Boat	6	1.29	6.17	29.76
Girl	8	2.03	3.92	34.79
Girl	6	1.28	6.22	31.04

Table (3.4): The PSNR and Compression ratio (C.R) values for color images with different number of compressed bits (b) for DCT method.

Image	Number of compressed bits (b) in Y	Number of compressed bits (b) in C_b, C_r	Compressed Size bit/pixel	C.R	PSNR (dB)
Horses	8	6	4.35	5.50	35.38
Horses	8	2	4.16	5.75	33.08
Horses	7	4	3.58	6.70	30.86
Peppers	8	6	4.53	5.29	30.76
Peppers	8	2	4.25	5.64	26.49
Peppers	7	4	3.70	6.48	26.98
Mandrill	8	6	5.85	4.09	23.24
Mandrill	8	2	5.61	4.27	22.40
Mandrill	7	4	4.08	5.87	20.94

Original Lena

Original Cameraman

Original Barbara

Original Boat

Original Girl

Figure 3.3: Test Images

C.R=1.63, bpp=4.87, PSNR=30.16 dB

C.R=1.60, bpp=4.99, PSNR=26.36 dB

C.R=1.62, bpp=4.92, PSNR=28.15 dB

C.R=1.63, bpp=4.90, PSNR=30.16 dB

C.R=1.60, bpp=4.99, PSNR=32.84 dB

Figure 3.4: Reconstructed gray scale images after applying IWT.

C.R=6.01, bpp=1.33, PSNR=29.04 dB

C.R=6.10, bpp=1.30, PSNR=27.29 dB

C.R=6.02, bpp=1.32, PSNR=27.57 dB

C.R=6.17, bpp=1.29, PSNR=29.76 dB

C.R=6.22, bpp=1.28, PSNR=31.04 dB

Figure 3.5 Reconstructed gray scale images after applying DCT method with number of compressed bits (b)=6

Original Horses

C.R=1.96, bpp=12.19, PSNR=30.33 dB

Original Peppers

C.R=1.68, bpp=14.23, PSNR=30.97 dB

Original Mandrill

C.R=1.60, bpp=14.92, PSNR=21.15 dB

Figure 3.6 Original and Reconstructed color images after applying IWT with number of compressed bits (b) in LLLL=8, LLLH=6, LLHL=6, LLHH=4, LH=4, H=4, HH=2.

Original Horses

C.R=6.70, bpp=3.58, PSNR=30.86 dB

Original Peppers

C.R=6.48, bpp=3.70, PSNR=26.98 dB

Original Mandrill

C.R=5.87, bpp=4.08, PSNR=20.94 dB

Figure 3.7 Original and Reconstructed color images after applying DCT number of compressed bits (b) in Y=7, (b) in C_b, C_r=4

Chapter Three

Proposed Compression Methods

Chapter Four
Conclusions and Future Work

4.1 Conclusions

In this research the following remarks were concluded:

1. The resulted compression ratio for IWT and DCT could be changed when different control parameters are used, mainly when the threshold value increased. This is due to less number of coefficients being considered. As a result the PSNR is decreased.
2. The implementation of (IWT) adds some power and flexibility to the compression performance. The coefficients take integer values for both the forward and the inverse transform of the IWT.
3. The achieved rate in terms of number of bits per pixel (bpp) in DCT is changed from 8 bpp to about (1.2 to 2.3) bpp, while for color images from 24 bpp to about (3.5 to 6) bpp. This is obtained by setting the number of compressed bits(b). In IWT, 8 bpp was changed to about (3.5 to 5.5) bpp, while for color images from 24 bpp to about (12 to 16.5) according to the number of compressed bits (b).
4. It is noticed that in the DCT method, when the number of compressed bits(b) increased in very small range the C.R is increased in very large range. But the error in the image also increased, which means, loosing more information. On the other hand, in IWT the increase of the number of compressed bits(b) leads to increase the C.R smoothly with no loss of noticeable information.

4.2 Future work

1. Study the effects of threshold variation according to the given band in IWT.
2. Using different image formats and notice for the both methods if it is effect in the result or not.
3. Improve the achieved PSNR by looking for those bands that affect the quality.

Chapter Four

Conclusions and Future Work

References

- **[Abd03]** T.Z.Abdul Hameed, "Advanced Compression Tecniques for Multimedia Applications", M.Sc. Thesis, Dep. Computer Science, Saddam University, 2003.
- **[Add00]** J.Addel, "Lossy and Losseless Image Compression", Prentic Hall PTR, 2000.
- **[Ala94]** A.A.Al-Azraqi, "Image Data Compression using Classified Vector Quantization", M.Sc. Thesis. Dep. Computer Science, Saddam University, 1994.
- **[Avc02]** I.Avcibas, "A Progressive Losseless/Near Losseless Image Compression", 2002, IEEE Signal Processing Letters.
- **[Bee01]** A.P.Beegan, "Wavelet-based Image Compression using Human Visiual System", M.Sc. Thesis, Dep. Electrical Engineering, Virginia Polytechnic Institute and State University, 2001.
- **[Bor03]** J.Bormans, " Multimedia Image Compression", 2003.
- **[Bur98]** C.S.Burrus, R.A.Gopinath and H.Guo, "Introduction to Wavelets and Wavelet Transform: A Primer", Prentic Hall, Inc., 1998.
- **[Cab02]** K.Cabeen, "Image Compression Using Distributed Systems" , College of the Redwoods, 2002.

- **[Gra95]** A.Graps, "An Introduction to Wavelets", IEEE Computational and Engineering, 1995, http://www.amara.com/current/wavelet.html.

- **[Hus03]** A.A.Hussien, "Fractal Image Compression with Fasting Approaches", M.Sc. Thesis, Dep. Computer Science, Saddam University, 2003.

- **[Ibr04]** I.N.Ibraheem, "Image Compression using Wavelet Transform", M.Sc. Thesis, Dep. Computer Science, Baghdad University, 2004.

- **[Jia03]** X.Jianyun, A.H.Sung, P.Shi and Q.Lin, "Text Steganography Using Wavelet Transform", Dep. Computer Science, New Maxico Tech, Socorro, USA, 2003.

- **[Maj97]** E.E.Majani, "Algorithm For Lossy Progressive Losseless Image Compression", National Astronautics and Space Adminstration (NASA), 1997.

- **[Mor98]** J.Morris, "Center for Intelligent Information Processing Systems", Electrical and Electronic Engineer Dept., University of Western Australia, http://ciips.ee.uwa.edu.au/morris/Year2/PLDS210/ds_ToC.html, 1998.

- [**Sal00]** D.Salomon, "Data Compression the Compelet Reference", Addition Wesley Company, Second Edition, 2000.

- **[Taw01]** M.D.Tawfeeq, "Design Finger Print Recognition System Using Wavelet Transform", M.Sc. Thesis, Dep. Computer Science, Saddam University, 2001.

- [**Tru99]** J.Trulove, "Multimedia Networking Handbook", 1999.

- [**Umb98]** S.E.Umbaugh, "Computer Vision and Image Processing", Prentic Hall PTR, Inc, USA, 1998.

- [**Wan00]** A.Wang, "Computer Basics", Dep. Information and Computer Science, California Irivin, http://www.ics.uci.edu./-dan/pubs/CompressionBasics.html, albertAcs.tut.fi, 2000.

- **[Xia01]** P.Xiao, "Image Compression By Wavelet Transform", M.Sc. Thesis, College of Science, Fast Tennessee State University, 2001, http:/www.etd-ubmit.etsu.edu/etd/theses-vailable/etd-0711101-121206/unrestricted/xiaop0720.pdf.

Appendix A
The BMP File Format

The BMP file structure is very simple and is shown in Figure A.1:

File Header	Image Header	Colour Table	Pixel Data

Figure A.1 BMP File Format

File Header

Every Windows BMP begins with a BITMAPFILEHEADER structure whose layout is shown in Table A.1. The main function of this structure is to serve as the signature that identifies that file format.

Field Name	Size in Byte	Descripition
bfType	2	Contains the characters "BM" that identify the file type
bfSize	4	File size
bfReserved1	2	Unused
bfReserved2	2	Unused
bfOffBits	4	Offset to start of pixel data

Table A.1 Bit Map File header structure

Three checks can be made to ensure that the file you are reading is in fact a BMP file:

- The first two bytes of the file must contain the ASCII characters "B" followed by "M".

- If you are using a file system where you can determine the exact file size in bytes, you can compare the file size with the value in the bfSize field.
- The bfResevéd1 and bfReservsed2 fields must be zero.

The file header also specifies the location of the pixel data in the file. When decoding a BMP file you must use the bfOffbits field to determine the offset from the beginning of the file to where the pixel data starts. Most applications place the pixel data immediately following the BITMAPINFOHEADER structure or palette, if it is present. However, some applications place filler bytes between these structures and the pixel data so you must use the bfOffbits to determine the number of bytes from the BITMAPFILEHEADER structure to the pixel data.

Image Header

The image header immediately follows the BITMAPFILEHEADER structure. It comes in two distinct formats, defined by the BITMAPINFOHEADER and BITMAPCOREHEADER structures.

BITMAPCOREHEADER represents the OS/2 BMP format and BITMAPINFOHEADER is the much more common windows format. Unfortunately, there is no version field in the BMP definitions. The only way to determine the type of image structure used in a particular file is to examine the structures's size field, which is the first 4 bytes of both structure types. The size of the BITMAPCOREHEADER structrure is 12 bytes; the size of BITMAPINFOHEADER, at least 40 bytes.

The layout of BITMAPINFOHEADER is shown in Table A.2. This structure gives the dimensions and bit depth of the image and tells if the image is compressed. Windows 95 supports a BMP format that uses an enlarged versions of this header. Few applications create BMP files using this format; however; a decoder should be implemented so that it knowa that header sizes can be larger than 40 bytes. The image height is unsigned value. A negative value for the biHeight field specifies that the pixel data is oredered from the top down rather than the normal bottom up. Images with a negative biHeight value may not be compressed.

Field Name	Size	Description
biSize	4	Header size must be at least 40
biWidth	4	Image width
biHeight	4	Image height
biplanes	2	Must be 1
biBitCount	2	Bits per pixel: 1,4,8,16,24, or 32
biCompression	4	Compression type: BI_RGB=0,BI_RLE8=1, or BI_BITFIELDS=3
biSizeImage	4	Image size:may be 0 if not compressed
bixPelsPerMeter	4	Preferred resolution in pixels per meter
biyPelsPerMeter	4	Preferred resolution in pixels per meter
biClrUsed	4	Number of entries in the color map that are actually used
biClrImportant	4	Number of significant colors

Table A.2 Bit Map Info Header structure

The BITMAPCOREHEADER structure is the other image header format. Its layout is shown in Table A.3:

Field Name	Size	Descripition
bcSize	4	Header size must be 12
bcWidth	2	Image width
bcHeight	2	Image height

bcPlanes	2	Must be 1
bcBitCount	2	Bit Count: 1,4,8, or 24

Table A.3 Bit Map Core Header structure

Notice that it has fewer fields and that all have analogous fields in the BITMAPINFOHEADER structure. If the file uses BITMAPCOREHEADER rather than BITMAPINFOHEADER, the pixel data cannot be compressed.

Color Palette

The color palette immediately follows the file header and can be in one of three formats. The first two are used to map pixel data to RGB color values when the bit count is 1,4, or 8 (biBitCount or bcBitCount fields). For BMP files in the windows format, the palette consists of an array of 2 bitcount RGBQUAD structures (Table A.4). BMP files on OS/2 format use an array of RGBTRIPLE structures (Table A.5).

Field Name	Size	Description
rgbBlue	1	Blue color value
rgbGreen	1	Green color value
rgbRed	1	Red color value
rgbReserved	1	Must be zero

Table A.4 BRGBQUAD structure

Field Name	Size	Description
rgbtBlue	1	Blue color value
rgbtGreen	1	Green color value
rgbtRed	1	Red color value

Table A.5 BRGTRIPLE structure

الخلاصه

من المشاكل الهامة في تطبيقات الحاسوب هو نقل وخزن المعلومات ومنها الصور الرقمية. ومن اجل تقليص حجم المعلومات المتراسلة تطلب ذلك البحث في موضوع ضغط هذه المعلومات. لذلك أُقترحت طرق مختلفة للضغط باستخدام تقنيات مختلفة لتحقيق نسب ضغط عالية وجودة عالية للصور خصوصا. ومن بين تلك التقنيات طريقتان هما: Transform Coding (TC) و Wavelet Transform (WT).

في طريقة Wavelet Transform والذي هو موضوعنا في هذا البحث. أُستخدم التحويل المويجي لتقسيم الصورة الى حزم جزئية. وفي هذا البحث قد أُستخدم نوع من التحويل المويجي وهو Integer Wavelet Transform (IWT). يتكرر هذا التقسيم أكثر من مرة (او المعالجة المويجية نستطيع امرارها مرة او أكثر). ومن بعد ذلك تُقرب نتائج التحويل المويجي لأقرب عدد صحيح للحصول على صورة مضغوطة. التقنية الثانية هي Transform Coding، في هذه الطريقة تُقسم بيانات الصورة إلى وحدات، كل وحدة تحول إلى متسلسلة الجيب تمام ومن ثم تُقرب نتائجها لأقرب عدد صحيح. وقد أُستخدمت هذه الطريقة مبدئيا لأغراض المقارنة.

أُستخدم نوعين من الصور لأختبار النتائج وهي صورغير ملونة وصور ملونة. أُستخدمت صور غير ملونة في طريقة WT، وقد حققت نسبة ضغط من١/١ إلى ٣/١، اعتماداً على عدد البتات المضغوطة (b). بينما في طريقة TC، حققت نسبة ضغط من ٣/١ إلى ٧/١، أما في الصور الملونة فقد حققت طريقة WT نسبة ضغط من ١/١ إلى ٢/١. بينما في طريقة TC فقد حققت نسبة ضغط من٤/١ إلى ٧/١، مع نسبة خطأ مقبولة في كلا الطريقتين.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة النهرين
كلية العلوم

تأثير التكميم على ضغط الصورة باستخدام الترميز التحويلي

رسالة

مقدمة إلى كلية العلوم في جامعة النهرين

كجزء من متطلبات نيل شهادة درجة الماجستير

في علوم الحاسوب

من قبل

وفاء شهاب أحمد المشهداني

(بكالوريوس ٢٠٠٣)

المشرفون

د. عبد الكريم عبد الرحمن كاظم　　　　د. طه سعدون باشاغا

شباط ٢٠٠٧　　　　صفر ١٤٢٨