

## Abstract

Color images have the potential to convey more information than a monochrome or black and white images. Each pixel for a true color image is stored as values of red, green, and blue. However, the RGB color model is not an efficient representation for compression task because there is significant correlation between RGB color components. Therefore, for compression, luminance-chrominance representations (such as YIQ) is implemented. A common approach to the color image compression was started by transform the RGB image to a desire color model, then applying compression techniques, and finally retransform the results back to RGB model.

In this work, a color image compression scheme combining the wavelet transform and a modified vector quantization (MVQ) method is proposed. In wavelet transform, the low and high Haar filters are composed to construct four 2- dimensional filters, such filters are applied directly to the YIQ image to speed up the implementation of the Haar wavelet transform. Haar wavelet transform was used to map it to frequency bands. Bit allocation process and scalar quantization are implemented on the approximation subband while modified vector quantization mechanism is employed to encode other higher frequency subbands using small block size (so as decrease the codebook size) as the subband number increases. Since the encoding process is much easier when the range of coded parameters are positive, thus the coefficients values of codebook are mapped to the positive range. Finally S-Shift encoding process is performed.

The analysis results have indicated that the proposed method offers a comparission performance up to (29/1) with little effects will be noticed on the image quality.

## الخلاصة

إن الصورة الملونة لها القدرة على حمل معلومات أكثر من الصور الأحادية اللون أو صور الأسود-أبيض. إن كل نقطة من نقاط الصور الملونة ذات الألوان الحقيقية تمثل بدلالة ثلاثة درجات لونية هي الأحمر، والأخضر، والأزرق. إن تمثيل هذه النقاط في الغالب لا يكون مطابقاً لما تراه العين المجردة. إن نظام RGB غير كفاء لعملية ضغط الصور بسبب الترابط الموجود بين مركباته لذلك تم استخدام نظام (YIQ) والتي يعتمد مركباته على صفات الإضاءة واللون.

إن الطريقة العامة لضغط الصور الملونة تبدأ بتحويلها من نظام RGB إلى إحدى الأنظمة اللونية المناسبة ومن ثم تطبيق إحدى تقنيات ضغط الصور على هذه الأنظمة الجديدة، ثم يتم إعادتها إلى النظام السابق RGB.

في هذا البحث تم اقتراح منظومة تشفير الصور الملونة اعتمدت التحويل المويجي وطريقة التكميم الاتجاهي المعدله. في التحويل المويجي إن مرشحات موجة هار العمودية والأفقية توصف لبناء أربعة مرشحات ثنائية البعد تسلط بشكل مباشر على الصورة لتسريع تنفيذ تحويل موجة هار. إن تحويل هار المويجي استخدم لنقل بيانات الصورة الى حزم تردد ثم استخدمت طريقة توزيع الثنائيات والتكميم العددي على جزء الحزمه التقريبي بينما نفذت اليه التكميم الاتجاهي المعدله لتشفير بقية حزم الموجات ذات التردد العالي بأستعمال القطع ذو حجم صغير مع تقليل حجم الكتاب المشفر مع زيادة رقم حزمة التردد. بما انه عملية التشفير تكون اسهل بكثير عندما يكون مدى المقاطع المشفره موجباً لذلك فان قيم المعاملات لكتاب الشفره نقلت ضمن المجال الموجب. أخيراً عملية تشفير ال (S-Shift) قد انجزت. إن نتائج التحليل قد أشرت بأن أداء الشكل المقترح يكون افضل جداً عندما تكون الصور المستجرحه ذات مستوى تشويه أقل مع نسبة ضغط عاليه.

إن نتائج التحليل قد قدمت بان الطريقة المقترحه تعرض لضغط يصل الى 29/1 مع تأثير قليل يلاحظ على كفاءة الصور.

# Chapter 1

## General Introduction

### 1.1 Introduction

Image processing is computer imaging where the application involves a human being in visual loop. In other words, the images are to be examined and acted upon by people. The major topics within the field of image processing include image restoration, image enhancement, and image compression [Umb98]:

- Image restoration is the process of taking an image with some known, or estimated, degradation, and restoring it to its original appearance.
- Image enhancement involves taking an image and improving it visually, typically by taking advantage of the human visual system's response.
- Image compression involves reducing the typically massive amount of data needed to represent an image.

An image may be defined as a two-dimensional function,  $f(x, y)$ , where  $x$  and  $y$  are *spatial* (plane) coordinates, and the amplitude of  $f$  at any pair of coordinates  $(x, y)$  is called the *intensity* or *gray level* of the image at that point. When  $x$ ,  $y$ , and the amplitude values of  $f$  are all finite, discrete quantities, we call the image a *digital image*. The field of *digital image processing* refers to processing digital images by means of a digital computer. Digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as *picture elements*, *image elements*, and *pixels*. *Pixel* is the term most widely used to denote the elements of a digital image [Gon02].

Color image processing is concerned with the manipulation of digital color image on computer, and is a branch of the wider of digital image processing discipline, which is itself a branch of the more general field of digital signal processing. The purpose of color image processing is to enhance or improve the color image, beside to extract information [San98].

Digital color images are usually stored using the Red, Green, and Blue or RGB color space. A RGB representation of an image is generated by spectral primary filtering an arbitrary color scene. Filters generate three channels by the spectral subbands red, green and blue, which usually overlap. Therefore the RGB model representation is a very redundant one.

The combine of these three channels of light produces a wide range of visible colors. All color spaces are three-dimensional orthogonal coordinate systems, meaning that there are three axes (in this case the red, green, and blue color intensities) that are perpendicular one to another [Bel03].

## **1.2 Image Size Problem**

Many of images are yearly received. Beside to the multispectral of these images their size may extend to reach multi\_mega bits of size level. As an example, color digital image consists of three digital images (i.e., red, green, and blue); each image is represented by 512\*512 matrixes of pixels, with 8 bits (i.e., 256 levels). Thus, the number of bits required to store this type of images is,  $S = 512 \times 512 \times 8 \times 3$  bits = 786 Kbytes, a rather large number. Thus, the goal of color image compression is to reduce this number, as much as possible, keeping in mind that reconstructed image has to be a faithful duplication of the original image.

To avoid the possibility of storing totally the information, tread-off techniques were proposed and utilized to give more flexible solution to the image archive capacity overload problem. Most of these techniques are considered as part of image compression discipline framework [Jor97].

In fact, the goal of these methods is to reduce (compress) the number of bits required to represent the image information by utilizing a suitable encoding process. Some of these compression methods offer a perfect

reconstruction of the original image, while other methods impose distortions, up to an acceptable level, upon the reconstructed image.

### **1.3 Image Compression**

As long as bandwidth and storage isn't cheap and people are impatient, image compression is going to be utilized. After compressing, image data takes up less storage space and requires less bandwidth to be transmitted over the Internet. Whether we need to send images over the Internet or squeeze multiple files into a single file for backup and storage; imaging compression components will give the code which need to make images "lean and mean" and make the applications run much faster [Bot98].

Documents can be scanned and simply compressed using JPEG or GIF. Unfortunately, the resulting files tend to be quite large if one wants to preserve the readability of the text. Compressed with JPEG, a color image of a typical magazine page scanned at 100dpi (dots per inch) would be around 100 Kbytes to 200 Kbytes, and would be barely readable. The same page at 300dpi would be of acceptable quality, but would occupy around 500 Kbytes. Even worse, not only would the decompressed image fill up the entire memory of an average PC, but also only a small portion of it would be visible on the screen at once. GIF is often used on the Web for distributing black and white document images, but it requires 50 to 100KB per page for decent quality (150dpi) [Bot98].

Considering anisotropy in luminance images in the compression stage, has led to better compression performance and better preservation of details, thus getting higher image quality [Bel03].

### **1.4 Vector Quantization**

The application of wavelet transform in signal and image compression has attracted a great deal of attention. It is known that it generates a multiresolution representation of an image. There are several sub-images or subbands that might be encoded more perfectly than the original image. The

wavelet transform technique breaks the image information into various frequency bands and encodes each subband using suitable coding system.

Vector Quantization with image compression techniques have many uses in image processing. VQ is useful in compressing data that arises in a wide range of applications and it can achieve better compression performance than any conventional coding techniques, which are based on the encoding of scalar quantities.

It is very interesting to see why VQ works in compression. A vector quantizer is composed of two operations: the encoder, and the decoder. The encoder takes an input vector and outputs the index of the codeword that offers the lowest distortion. In this case the lowest distortion is found by evaluating the Euclidean distance between the input vector and each codeword in the codebook. Once the codeword of the closest reproduction vector is found, the index of that codeword is sent through a channel (the channel could be computer storage, communications channel, and so on). When the decoder receives the index of the codeword, it replaces the index with the associated codeword.

## **1.5 An Introduction to Wavelets**

From digital signal processing to computer vision, wavelets have been widely utilized to analyze and transform discrete data. The concept of wavelets is rooted in many disciplines, including mathematics, physics, and engineering. The 1980s witnessed a new wave of wavelet discoveries, like multiresolution analysis and orthonormal compactly supported wavelets. These advances have revolutionized the field and have led to many novel applications of wavelets [Asu02].

A wavelet, which literally means little wave, is an oscillating zero-average function that is well localized in a small period of time. A wavelet function, known as a mother wavelet, gives rise to a family of wavelets that are translated (shifted) and dilated (stretched or compressed) versions of the original mother wavelet [Asu02].

Wavelets have great utility in the area of digital signal processing. A digital signal can be represented as a summation of wavelets that are fundamentally identical except for the translation and dilation factors (or coefficients). Hence, a signal can be represented entirely by wavelet coefficients. These coefficients provide important frequency and temporal information which can be used to analyze a signal. Furthermore, the signal can be processed in the wavelet coefficient domain before being transformed back to the normal time-amplitude representation. Thus, wavelets facilitate a unique framework for digital signal processing [Asu02].

## 1.6 Related Work

There are many coding techniques applicable to color image. In our work, some adaptive coding methods have been presented for performing the image compression process; all these methods are principally based on using wavelet transform and vector quantization.

*J. R. Goldschneider* [1997] developed lossy compression algorithms based on the wavelet transform and vector quantization. Optimal bit-allocation algorithms based on pruned tree-structures vector quantization techniques are developed for the discrete wavelet transform and for the more general discrete wavelet packet transform. These algorithms systematically find all quantizers on the lower convex hull of the rate-distortion curve, while for the wavelet packet transform, simultaneously selecting based basis.

*S. Paek* and *L. Kim* [2000] proposed zerotree wavelet vector quantization (WVQ) algorithm focuses on the problem of how to reduce the computation time to encode wavelet images with high coding efficiency. Conventional wavelet image compression algorithm exploits tree-structure of wavelet coefficients coupled with scalar quantization. However, they can not provide the real-time computation because they use iterative methods to decide zerotrees. In contrast, the zerotree WVQ algorithm predicts in real-time zero-vector trees of insignificant wavelet vectors by a non-iterative decision rule and then encodes significant wavelet vectors by the classified VQ (CVQ). These cause the zerotree WVQ algorithm to provide the best compromise between the coding performance and the computation time.

*B. K. Al-Abudi* [2002] proposed hierarchical multilevel block truncation coding (HMBTC) based on wavelet transform. Zero tree and transmission progressive mechanism were designed and implemented through the encoding and decoding process, and also proposed Wavelet Modified Vector Quantization.

*O. O. Khalifa* [2003] a modified version of Linde-Buzo-Gray (LBG) algorithm using Partial Search Partial Distortion (PSPD) is presented for coding the wavelet coefficients. The proposed scheme can save 70 - 80 % of the Vector Quantization (VQ) encoding time compared to fully search VQ and reduced arithmetic complexity with out sacrificing performance.

*F. Murtagh* [2005] considers the wavelet transform of a rooted, node-ranked, p-way tree. he focus on the case of binary trees, and a variant of the Haar wavelet transform. They distinguish between two cases: firstly, where the binary tree represents a hierarchy of embedded, non-overlapping subsets of a given set; and secondly, where the binary tree represents an ultrametrically related set of points. Wavelet transforms allow for multiresolution analysis through translation and dilation of a wavelet function.

## **1.6 Aim of Thesis**

The aim of thesis is implementing a color image compression scheme based on the modified vector quantization coding (VQ) method and wavelet transform (WT). Modified VQ was adapted to encode color images. Because the RGB image has extensive correlation among color components, luminance-chrominance representation was implemented. In this scheme, scalar quantization method was implemented to encode the approximation subband of wavelet transform while modified VQ was employed to encode the high frequency subbands.



## **1.7 Thesis Layout**

The reminder of this thesis will be structured into the following chapters:

**Chapter 2:** Entitle “*Color Models & Image Compression*”. This chapter implies a brief overview about color, Luminance, Chrominance, Pixels and Bitmaps, Types of Digital Images, Computer Color Image Representation types of images, representation of color images and description of various mathematical models, including the equations and coefficients required for converting color pixel values between different color models. The second part will be dedicated to discuss the based of some image compression techniques with clear emphasis on their properties and transformation methods.

**Chapter 3:** Entitle “*Hybrid Wavelet Modified Vector Quantization*”. This chapter will be devoted to present a full description for the color image compression technique based on wavelet transform and modified VQ. The reasons for implementing luminance-chrominance representation for compression will be investigated. The experimental results of these methods will be investigated and discussed.

**Chapter 4:** Entitle “*Conclusions & Suggestions*”. A brief review will be devoted to demonstrate the success of the presented ideas to match the requirements of color image compression task. A further adaptive proposals will be given as future work

# Chapter 2

## Color Models and Image Compression

### 2.1 Introduction

A digital color image is a collection of pixels with each pixel a 3-dimensional (3-D) color vector. The vector elements specify the pixel's color with respect to a chosen color space. Joint Photographic Experts Group (JPEG) is a commonly used standard to compress digital color images. JPEG compresses by quantizing the discrete cosine transform (DCT) coefficients of the image's three color planes [Nee04].

*Mass storage* capability is a must in image processing applications. An image of size 1024\*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage for image processing applications falls into three principal categories [Gon02]:

- (1) Shortterm storage for use during processing,
- (2) On-line storage for relatively fast recall, and
- (3) Archival storage, characterized by infrequent access.

Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion bytes), and Tbytes (meaning tera, or one trillion bytes).

### 2.2 Basic Concepts:

The basic concepts in the image are:

### 2.2.1 What is Color?

Color is the perceptual result of light in the visible region of the spectrum, having wavelengths in the region of 400 nm to 700 nm, incident upon the retina. Physical power (or *radiance*) is expressed in a *spectral power distribution* (SPD).

The human retina has three types of color photoreceptor *cone* cells, which respond to incident radiation with somewhat different spectral response curves. A fourth type of photoreceptor cell, the *rod*, is also present in the retina. Rods are effective only at extremely low light levels (colloquially, *night vision*), and although important for vision play no role in image reproduction.

Because there are exactly three types of color photoreceptor, three numerical components are necessary and sufficient to describe a color, providing that appropriate spectral weighting functions are used. This is the concern of the science of *colorimetry*. In 1931, the Commission Internationale de L'Éclairage (CIE) adopted standard curves for a hypothetical *Standard Observer*. These curves specify how an SPD can be transformed into a set of three numbers that specifies a color [Poy97].

### 2.2.2 Luminance

The luminance of a color is a measure of its perceived brightness. The computation of luminance takes into account the fact that the human eye is far more sensitive to certain colors (like yellow-green) than to others (like blue) [Sac99].

### 2.2.3 Chrominance

Chrominance is a complementary concept to luminance. If you think of how a television signal works, there are two components—a black and white image which represents the luminance and a color signal which contains the chrominance information. Chrominance is a 2-dimensional color space that represents hue and saturation, independent of brightness [Sac99].

## 2.3 Types of Digital Images

The digital image  $f(x, y)$  is represented as two-dimensional array of data, where each pixel value corresponds to the brightness of the image at the point  $(x, y)$ . The simplest type of image is the monochrome (one color, this is what we normally refer to as black and white) image data, other types of image data required extension or modification to this model, typically there are multiband images (color, multispectral), and they can be modeled by different function  $f(x, y)$  corresponding to each separate band of brightness information. The image can be classified into the following types [Umb98]:

### 2.3.1 Black and White Images

A black and white image is made up of pixels each of which holds a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps, normally 256 different grays. Since the eye can barely distinguish about 200 different gray levels, this is enough to give the illusion of a stepless tonal scale as illustrated below:

Assuming 256 gray levels, each black and white pixel can be stored in a single byte (8 bits) of memory [Sac99].

### 2.3.2 Color Images

A color image is made up of pixels each of which holds three numbers corresponding to the red, green, and blue levels of the image at a particular location. Red, green, and blue (sometimes referred to as RGB) are the primary colors for mixing light—these so-called additive primary colors are different from the subtractive primary colors used for mixing paints (cyan, magenta, and yellow). Any color can be created by mixing the correct amounts of red, green, and blue light. Assuming 256 levels for each primary, each color pixel can be stored in three bytes (24 bits) of memory. This corresponds to roughly 16.7 million different possible colors while for images of the same size, a black and white version will use three times less memory than a color version [Sac99].

### 2.3.3 Binary or Bilevel Images

In a binary image, each pixel assumes one of only two discrete values. Essentially, these two values correspond to on and off. A binary image is stored as a two-dimensional matrix of 0's (off pixels) and 1's (on pixels). A binary image can be considered a special kind of intensity image, containing only black and white. Other interpretations are possible, however; you can also think of a binary image as an indexed image with only two colors [Mat99].

### 2.3.4 Indexed Color Images

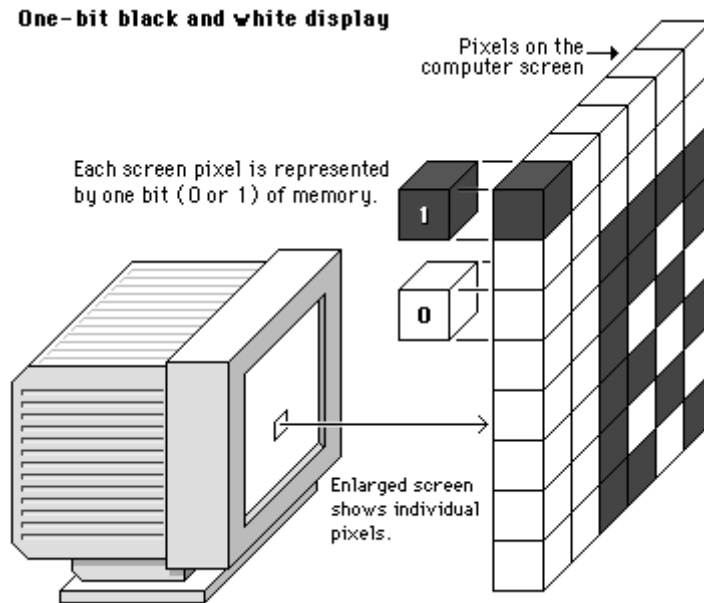
*Indexed color* (or *pseudocolor*), is the provision of a relatively small number of discrete colors – often 256 – in a *colormap* or *palette*. The frame buffer stores, at each pixel, the index number of a color. At the output of the frame buffer, a lookup table uses the index to retrieve red, green and blue components that are then sent to the display [Poy97].

There are several problems with using indexed color to represent photographic images. First, if the image contains more different colors than are in the palette, techniques such as dithering must be applied to represent the missing colors and this degrades the image. Second, combining two indexed color images that use different palettes or even retouching part of a single indexed color image creates problems because of the limited number of available colors [Sac99].

## 2.4 Computer Color Image Representation

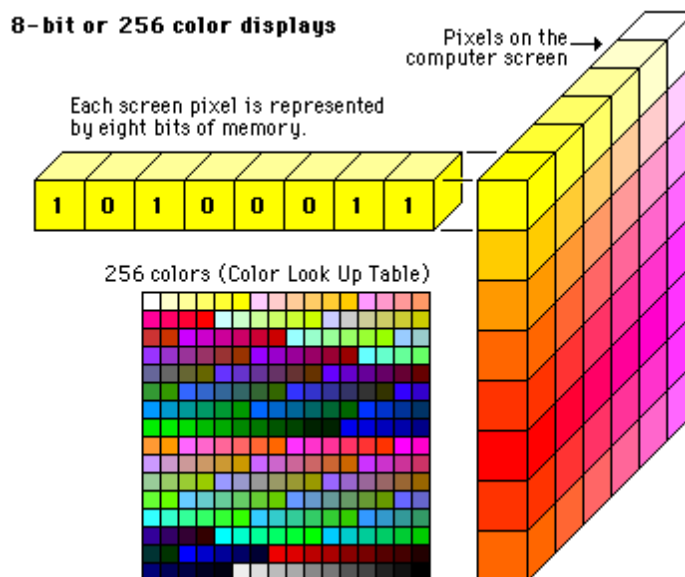
The computer's operating system displays, the (RGB) color model, which is called "additive" because a combination "add up" the three pure colors lead to white light.

In the simplest form of black and white computer displays a single bit of memory is assigned to each pixel. Since each memory bit can only be (0 or 1), a one-bit display system can only manage two colors (black or white) for each pixel on the screen as shown in figure (2.1) [Wei96].



**Fig. (2.1)** One bit black and white image display.

When eight bits of memory are dedicated to each pixel, each pixel could be one of 256 colors ( $256=2^8$ ). This kind of computer display is called an “eight bit” or “256 color” display as shown in figure (2.2). In eight bit images the 256 colors that make up the image are referenced to as “palette” or “index” (also called a color look up table, CLUT). The main point for eight bit images is that they can never contain more than 256 colors.



**Fig. (2.2)** 8-bit color image display.

True-color or 24-bits color displays can show million of unique colors on the computer screen. True color images are composed by dedicating 24 bits of memory to each pixel; eight bit for each component (red, green, and blue) ( $8+8+8=24$ ) as shown in figure (2.3)

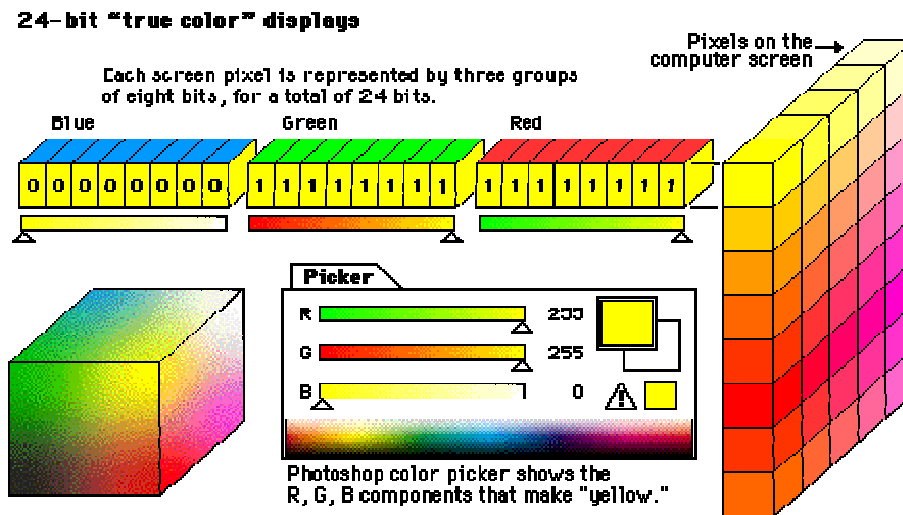


Fig. (2.3) 24-bit "True color image" –display.

True-color or 24 bits images are typically much larger than eight bits images in their uncompressed state, because each pixel in a 24-bits image has 24 bits of memory dedicated to it, typically in three monochrome layers (red, green, and blue) as shown in figure (2.4) [Wei96].

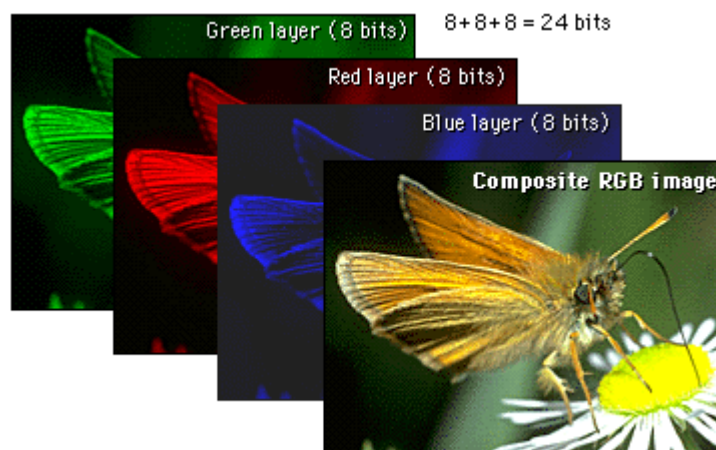


Fig. (2.4) Composite RGB images from three layers.

Since this type of images is the most common format in today's high-quality computer display and storage devices, therefore, in the present work this type of images will be utilized, also, the selected file format will be the **BMP (bitmap image)** file due to the popularity of this format [Lus94]. This type of images contains both header and image pixel data as shown in figure (2.5).



**Fig. (2.5)** The BMP file format of 24-bits image.

The header contains information about the image (like, image size, width, height, number of bits per pixel...etc)

## 2.5 Color Models

For computer applications; color is usually described in terms of the red, green, and blue, or RGB color model [Gon92]. However, the RGB color model does not model well the human perception of color. Applying image processing techniques in the RGB model will often produce color distortion and artifacts. Therefore, color models based on the human perception of color may be beneficial. The color models presented in this section are the most popular in the image processing community. Equations describing transformations between color models and the reasons for using color models other than RGB will be presented in the following sections.

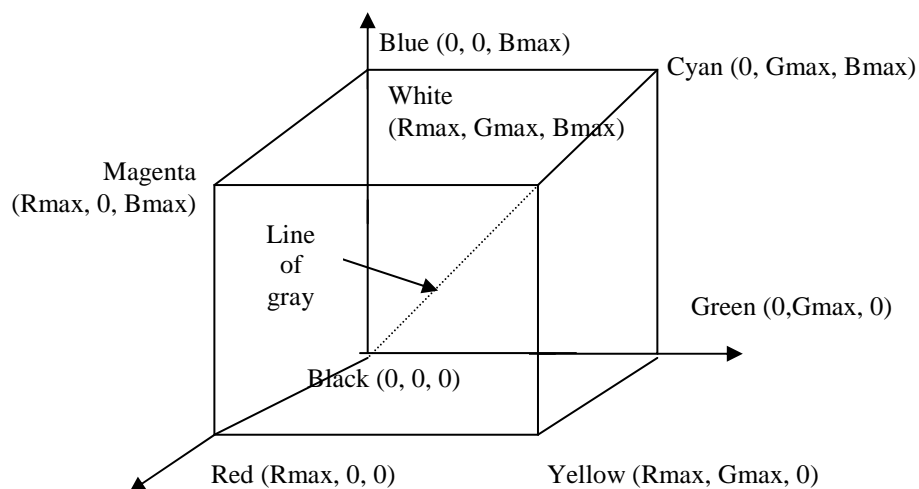
### 2.5.1 RGB Color Model

The RGB model is the most frequently used color model for image processing. Since color cameras, scanners and displays are most often provided with direct RGB signal input or output, this color model is the basic



one, which is, if necessary, transformed into other color models. The color gamut in RGB model forms a cube (see figure 2.6). Each color, which is described by its RGB components, is represented by a point and can be found either on the surface or inside the cube. All gray colors are placed on the main diagonal of this cube from black ( $R=G=B=0$ ) to white ( $R=G=B=\max$ ). The main disadvantage of RGB color model in applications involving natural images comes from the high correlation between its components; about 0.78 for B-R, 0.98 for R-G and 0.94 for G-B components. This makes the RGB model unsuitable for compression. The other disadvantages of RGB model can be summarized as follow [San98]:

1. Psychological non-intuitively, i.e. it is hard to visualize a color based on R, G, and B components.
2. Non-uniformity, i.e. it is impossible to evaluate the perceived differences between colors on the basis of distances in RGB model.



**Fig. (2.6)** RGB color cube. The gray scale spectrum lies on the line joining the black and white vertices.

## 2.5.2 CIEXYZ Color Model

The CIE system is based on the description of color as a luminance component Y, and two additional components X and Z. The spectral

weighting curves of X and Z have been standardized by the CIE based on statistics from experiments involving human observers. *XYZ tristimulus values* can describe any color. (RGB tristimulus values will be described later.)

The magnitudes of the XYZ components are proportional to physical energy, but their spectral composition corresponds to the color matching characteristics of human vision [Poy97].

In the (CIE) spectral and N.T.S.C primary systems, the tristimulus values required to achieve a color match are sometimes negative. The CIE has developed a color standard based on imaginary primary colors XYZ, which has been used in colorimetry. It is designed to yield a non-negative tristimulus value for each color. It is related to RGB tristimulus values according to the following relationship [San98]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.431 & 0.342 & 0.178 \\ 0.220 & 0.707 & 0.071 \\ 0.020 & 0.130 & 0.939 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.1)$$

### 2.5.3 CMYK Color Model

CMYK is a color model used for printing. It generates by combining pigments, or paints, for the model's three primary colors: cyan, magenta and yellow. For convenience, CMYK also uses black (K, not B, to avoid confusion with blue) because to generate black with CMYK would take mixing the paints from the three primaries; it's much easier to have black ink to create a really dark black on paper [Asi03].

Like the RGB color model, we'll need three color quantities to produce a color in the CMYK color model. Also like the RGB model, each color component assumes the same range of values and each color component uses the same unit of measurement (unlike the hue-based models, where the hue is measured as an angle and the other components are measured as a number between 0.0 and 1.0). The CMYK color model can be represented as a cube similar to the RGB cube. Figure (2.7) shows the CMYK cube [Asi03].

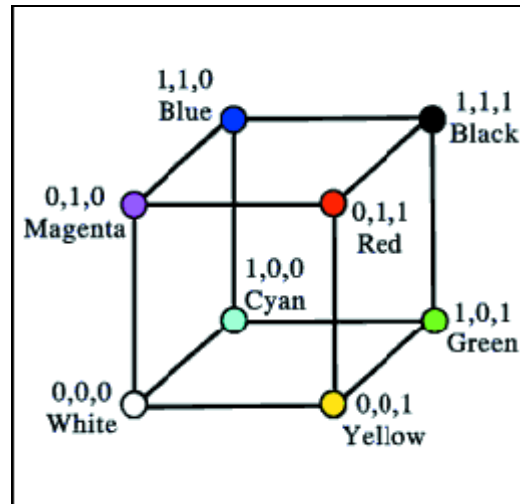


Fig. (2.7) The CMYK Color Cube.

### 2.5.4 YUV Color Model

In Europe the YUV color model (the basis of the PAL TV signal coding system) is used [San98]:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.2)$$

The Y (luminance) component is identical to the Y component in XYZ model. The YUV model is widely used in coding of color images and in color video. Because the color representation of an image sequences requires less detail than the luminance, the data rate can be shared as follows: 80% to the Y component and 10% each to the U and V components [San98].

The color models discussed thus far were designed for specific purposes: RGB for color image display, CMYK for printing and HSV (HSI, HSB) for an artistic view of color. However, there is one glaring omission from these color models. None of them leverage an important property of human vision. Human vision is more sensitive to changes in light intensity than changes in color. Perhaps that's why we have more rods than cones on our retinas. The YUV color model takes advantage of this interesting property of human vision [Asi03].

### 2.5.5 YIQ Color Model

The YIQ color model is useful in color TV broadcasting, and is a simple linear transform of the RGB representation [Son99]:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.3)$$

The Y (luminance) component is identical to the Y component in XYZ model. The inverse transform from YIQ to RGB is performed as follows:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \quad (2.4)$$

In this model the luminance (Y component) is separated from the chrominance ((I and Q components), which is designed to be more sensitive to changes in Luminance than to changes in Chrominance. Separating out the luminance from other components has several advantages:

1. Since the human eyes are more sensitive to the luminance than to chrominance, thus, the bits can be distributed for encoding in a more effective way, Q and I components can be represented by fewer numbers of bits compared to the Y component (i.e., Q and I components can be limited without noticeable degradation).
2. We can drop the chromatic part altogether if we want achromatic image (this is how black and white TVs can pickup the same signal as color ones).

### 2.5.6 YC<sub>b</sub>C<sub>r</sub> Color Model

This color model is closely related to the YUV model .It is appropriate for digital coding of standard TV images and is given as follows [San98]:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.418 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.5)$$

The YCbCr format is widely used for digital video. In this format, luminance information is stored as a single component (Y), and chrominance information is stored as two color difference components (Cb and Cr). Cb represents the difference between the blue component and a reference value, and Cr represents the difference between the red component and a reference value [Mat99].

Other current applications in image compression (e.g. JPEG format) often employ YCbCr model as quantization model [Son99].

### 2.5.7 IHS Color Model

In the perception process, a human can easily recognize basic attributes of color intensity (I), hue (H) and saturation (S). The hue (H) represents the impression related to the dominant wavelength of the perceived color, the saturation corresponds to relative color purity (lack of white in the color) and in the case of a pure color it is equal to 100%. For example, for a vivid red S=100% and for a pale red (pink) S=50%. Colors with zero saturation are gray levels. Maximum intensity is sensed as pure white, minimum intensity as pure black [San98]. As a three dimensions representation, the IHS color model (see Figure 2.8) can give a hexagonal volume with vertical axis representing intensity (i.e., from black (0) to white (1)), the distance from this axis representing saturation (from 0 to 100%), and the horizontal angle representing the change in the hue (from 0 to 360 degree, i.e., hue is measured from red. This model, as in the YIQ model, intensity (I) is decoupled from the color information, which is described by hue and saturation components.

As hue varies from 0 to 1.0, the corresponding colors vary from red, through yellow, green, cyan, blue, and magenta, back to red. As saturation varies from 0 to 1.0, the corresponding colors vary from unsaturated (shades of gray) to fully saturated (no white component). As value, or brightness,

varies from 0 to 1.0, the corresponding colors become increasingly brighter [Mat99].

It is sometimes useful to convert from RGB color cube to IHS hexagonal cone, and vice versa. The transform from RGB to IHS model is accomplished as follows: [Mat87]

```

Intensity = max(R, G, B)
M = min(R, G, B)
Range = Intensity-M
if intensity # 0 then Saturation = Range/Intensity else Saturation=0
  if S # 0 then
    r1 = (Intensity-R)/ Range
    g1 = (Intensity-G)/ Range
    b1 = (Intensity-B)/ Range
    if Intensity = R then
      if M = G then hue = 5+b1 else hue =1-g1
    else if Intensity = G then
      if M = B then hue =1+r1 else hue=3-b1
    else if M= R then hue =3+g1 else hue =1-r1
    hue = hue*60
    else hue = undefined
  end if S # 0
end.

```

While the inverse transform (IHS to RGB) is:

```

if hue =360 then hue = 0
hue = hue / 60
I = int (hue); f = hue -i
P1= intensity × (1- Saturation)
P2= intensity × (1- (f × Saturation))
P3= intensity × (1- Saturation × (1-f))
case of i
  0: R= intensity: G = P3: B = P1
  1: R = P2: G = intensity: B = P1

```

2:  $R = P1: G = \text{intensity}: B = P3$

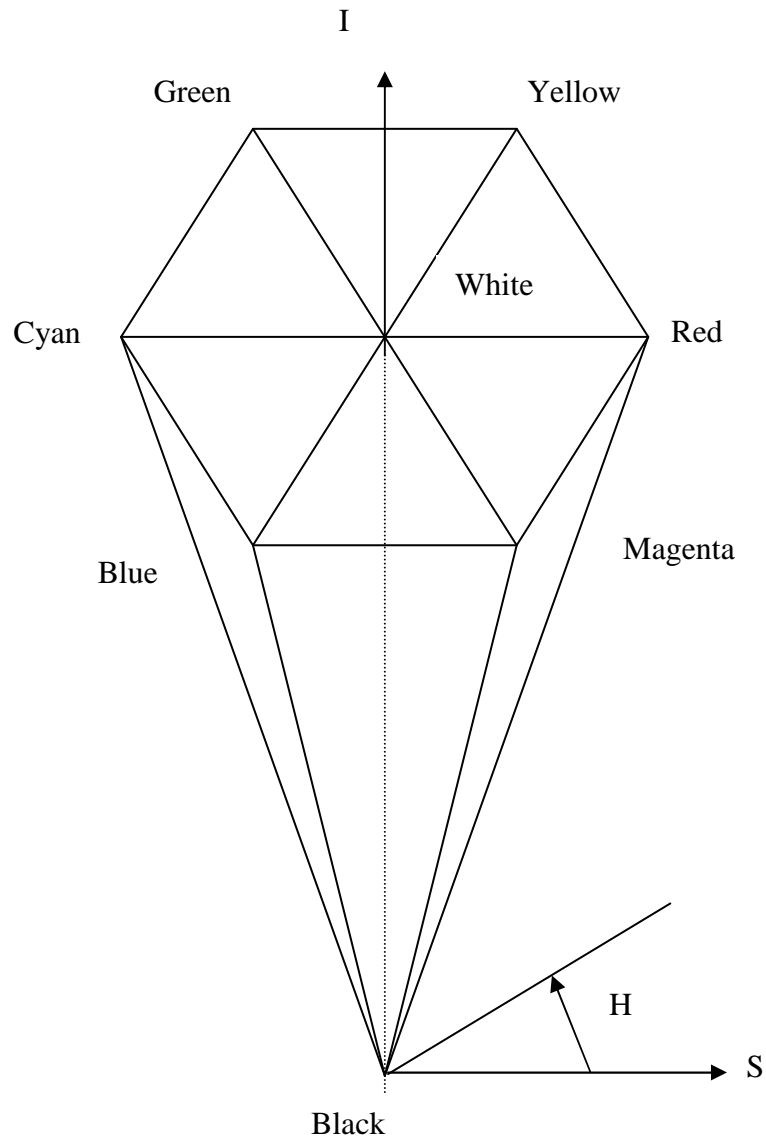
3:  $R = P1: G = P2: B = \text{intensity}$

4:  $R = P3: G = P1: B = \text{intensity}$

5:  $R = \text{intensity}: G = P1: B = P2$

*end case i*

*end*



**Fig. (2.8)** HIS hexagonal cone.

### 2.5.8 CIELAB Color Model

The CIELAB was designed to approach a perceptually uniform color model, in which the perceived color differences recognized as equal by the human eye would correspond to equal Euclidean distances.

It is based on the differences of three elementary color pairs: red-green, yellow-blue, and black-white. The values of CIELAB are  $L^*$ , which stand for lightness with a scale value ranging from 0(black) to 100(white) and is orthogonal to  $a^*$  and  $b^*$  (see figure 2.9).  $a^*$  denotes the redness-greenness and  $b^*$  denotes the yellowness-blueness [Hei00].

A color in CIELAB color model is computed directly from the color components in CIEXYZ and the component of reference white ( $X_o$ ,  $Y_o$ ,  $Z_o$ ). The CIELAB color model is defined by the following expressions [San98]:

$$L^* = 116 f\left(\frac{Y}{Y_o}\right) - 16 \quad (2.6)$$

$$\begin{aligned} a^* &= 500 \left[ f\left(\frac{X}{X_o}\right) - f\left(\frac{Y}{Y_o}\right) \right] \\ b^* &= 200 \left[ f\left(\frac{Y}{Y_o}\right) - f\left(\frac{Z}{Z_o}\right) \right] \end{aligned} \quad (2.7a)$$

where

$$f(x) = \begin{cases} x^{1/3} & x > 0.00885 \\ 7.787x + \frac{16}{116} & \text{otherwise} \end{cases} \quad (2.7b)$$

The reverse transformation (for  $x > 0.008856$ ) is:

$$X = X_o(P + a^*/500)^3 \quad (2.8a)$$

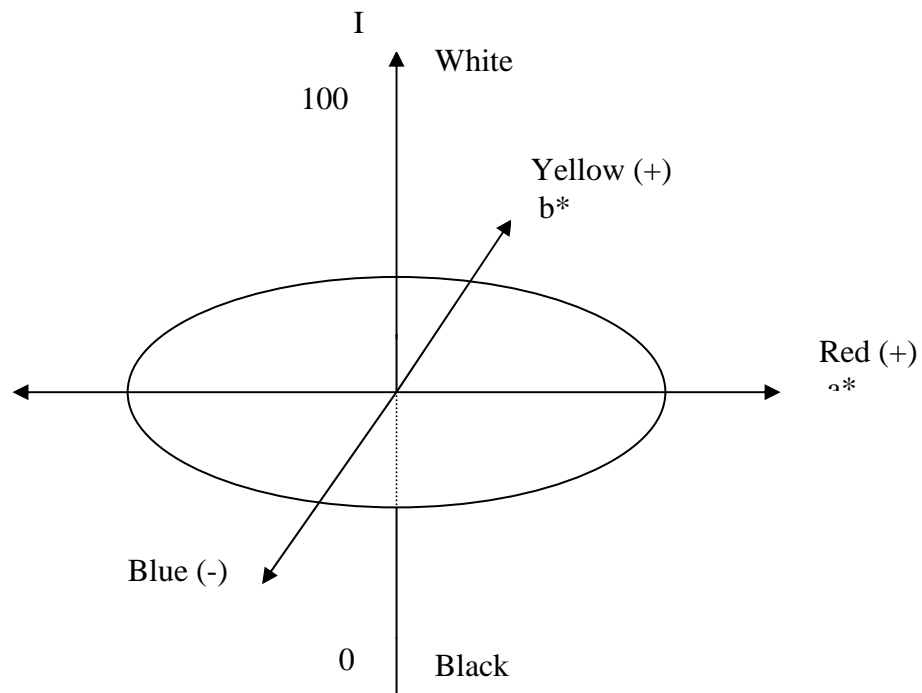
$$Y = Y_o P^3 \quad (2.8b)$$

$$Z = Z_o(P - b^*/200)^3 \quad (2.8c)$$

Where

$$P = (L^* + 16)/116 \quad (2.8d)$$





**Fig. (2.9)** A three dimensional representation of the CIELAB color mode.

### 2.5.9 CIELUV Color Model

The CIELUV model is recommended by the CIE for applications in additive light source conditions (e.g. display). The definition of  $L^*$  (which means the lightness) is the same as for the CIELAB color model which is given in equation (2.6). The  $u^*$  stands for the chromaticity variation approximately from green to red, and  $v^*$  represent the chromaticity variation approximately from blue to yellow. They are defined as follows [San98]:

$$u^* = 13L^* (u' - u'_0) \quad (2.9)$$

$$v^* = 13L^* (v' - v'_0) \quad (2.10)$$

The quantities  $u'_0$  and  $v'_0$  refer to reference white or the light source; for CIE standard illuminant,  $u'_0=0.2009$ ,  $v'_0= 0.4610$  [Chr01]. The value of  $u'$  and  $v'$  can be determined by using the following:

$$u' = \frac{4X}{(X + 15Y + 3Z)} \quad (2.11)$$

$$v' = \frac{9Y}{(X + 15Y + 3Z)} \quad (2.12)$$

Most of the physical display device are driven by RGB signals, so the transformation from  $L^*, u^*, v^*$  color model to RGB model must be done. The inverse mapping to RGB model is done in two steps.  $L^*, u^*, v^*$  to XYZ and XYZ to RGB. The transformation from CIELUV to XYZ is performed as follows [Poy97]:

$$u' = u^*/(13L^*) + u'_o \quad (2.13)$$

$$v' = v^*/(13L^*) + v'_o \quad (2.14)$$

$$Y = ((L^* + 16)/116)^3 \quad (2.15)$$

$$X = -9Y u' / ((u' - 4)v' - u' v') \quad (2.16)$$

$$Z = (9Y - 15v' Y - v' X) / 3v' \quad (2.17)$$

Then,

$$R = 0.431X - 1.393Y - 0.476Z \quad (2.18)$$

$$G = -0.969X + 1.876Y + 0.427Z \quad (2.19)$$

$$B = 0.068X - 0.229Y + 1.069Z \quad (2.20)$$

Since displays are most often provide output images with direct RGB model, images in any other color models must be transformed back to the RGB color model after handling the processed components in these models. In the present work, we applied the model (YIQ) as shown in chapter-3.

## 2.6 Color Image Compression

Compressed images are representations that require less storage than the nominal storage. This is generally accomplished by coding of the data based on measured statistics, rearrangement of the data to exploit patterns and redundancies in the data, and (in the case of lossy compression), quantization of information. The goal is that the image, when decompressed, either looks very much like the original despite a loss of some information (lossy compression), or is not different from the original (lossless compression) [Gib00].

Three primary types of redundancy can be found in digital images [Yan98; Shi00]:

1. Spatial redundancy due to the correlation between neighboring pixel value.
2. Spectral redundancy due to the correlation between color planes or spectral bands.
3. Temporal redundancy due to the correlation between different frames in sequence of images.

Image compression aims to reduce the number of bits by removing these redundancies. Based on the difference between original and reconstructed version, data compression schemes can be divided into two broad classes: [Add00]

### 2.6.1 Lossless coding

Also, called noiseless, or reversible compression. Lossless compression is usually limited to compression ratios of 4:1 or less, and even these ratios are achievable only on sources that have considerable structure. For images and scientific data, lossless compression ratios are usually much lower, frequently 2:1 or less. Codes designed for noiseless compression may even cause data expansion [Bot98].

The goal of lossless image compression is to represent an image signal with the smallest possible number of bits without loss of any information, thereby speeding up transmission and minimizing storage requirements. The number of bits representing the signal is typically expressed as an average bit rate (average number of bits per sample for still images, and average number of bits per second for video). The function of *compression* is often referred to as *coding*, for short [Gib00]. The most popular lossless compression methods are: *Run length encoding*, *LZW*, *Arithmetic coding*, *Huffman coding*, *S-shift coding*.

### **2.6.2 Lossy coding**

The second type of image compression is lossy, noisy, or irreversible compression [Bot98].

The goal of lossy compression is to achieve the best possible fidelity given an available communication or storage bit-rate capacity, or to minimize the number of bits representing the image signal subject to some allowable loss of information. In this way, a much greater reduction in bit rate can be attained as compared to lossless compression, which is necessary for enabling many real-time applications involving the handling and transmission of audiovisual information [Gib00]. The most well known lossy compression methods are Transform Coding, Vector Quantization, Block Truncation and Subband Coding. The proposed technique utilizes two of the compression techniques presented in figure (2.10).

The most common approaches in color image compression implies the application of separated encoding of the three image components. Lossy coding mostly operates on the luminance-chrominance representation with subsampled chrominance. The rationale for this is that the chrominance band is narrower than that for luminance. The chrominance components are often more compressed than luminance, at least in high compression applications. Thus the amount of chrominance data in the compressed bitstream can be 20% less than the amount of compressed luminance data [San98]. See figure (2.10).

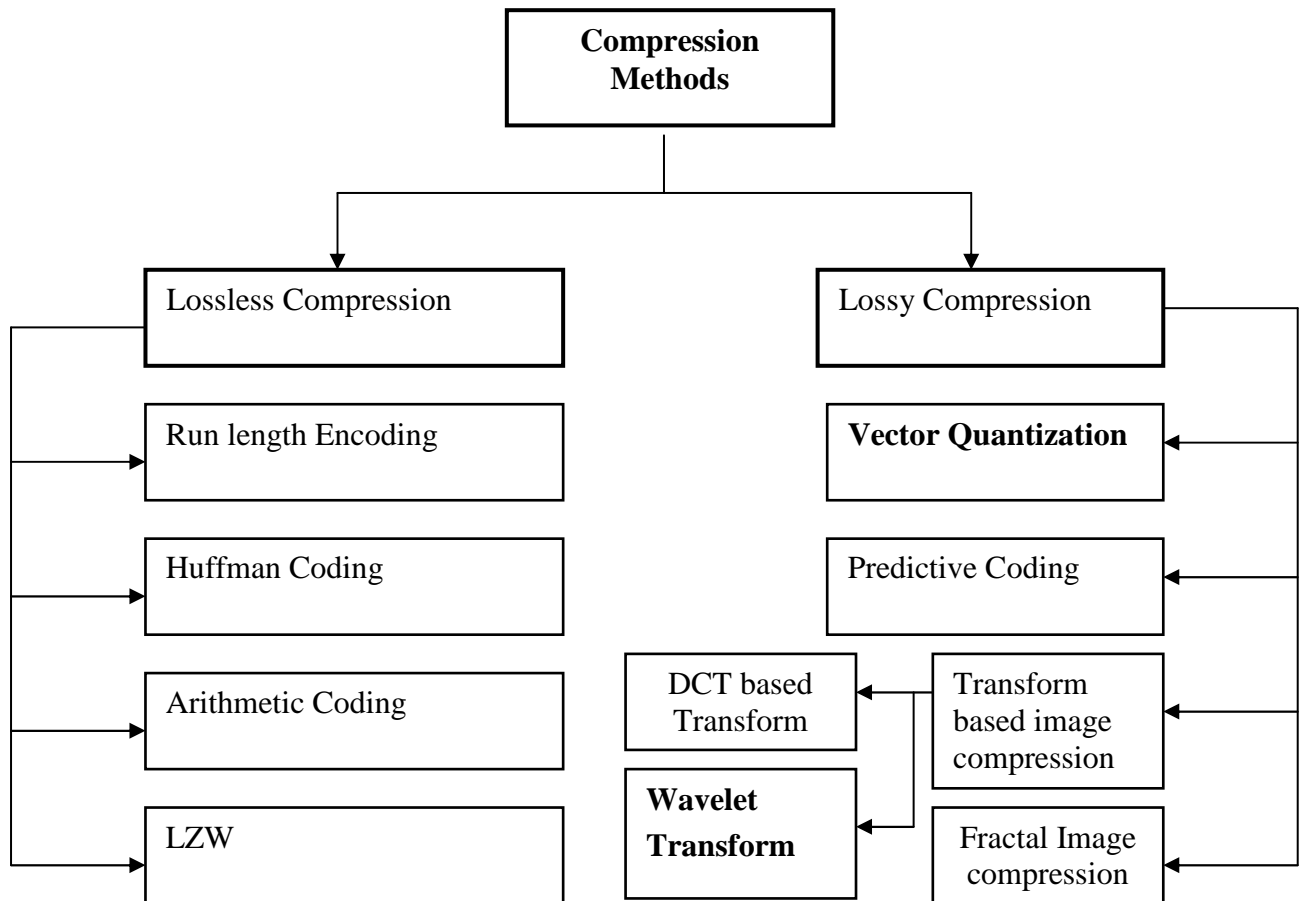


Fig. (2.10) The Most Popular Image Compression Methods.

## 2.7 Vector Quantization

Vector quantization is used for both image and sound compression. In practice, vector quantization is commonly used to compress data that have been digitized from an analog source, such as sampled sound and scanned images (drawings or photographs). Such data is called *digitally sampled analog data* (DSAD). Vector Quantization is based on two facts:

1. The compression methods that compress strings, rather than individual symbols, can, in principle, produce better results.
2. Adjacent data items in an image (i.e., pixels) and in digitized sound (i.e., samples) are correlated. There is a good chance that the near neighbors of a pixel P will have the same values as P or very similar values. Also consecutive sound samples rarely differ by much.

For images, VQ works as follows:

1. Two Operations : Encoder and Decoder
2. Encoder takes an input vector and outputs the index of the codeword that offer lowest distortion.
3. This index is sent to decoder through a channel.
4. Decoder replaces index with associated codeword. [Agg06].

Following is a simple, intuitive vector quantization method for image compression. Given an image, divide it into small blocks of pixels, typically  $2 \times 2$  or  $4 \times 4$ . Each block is considered a vector. The encoder maintains a list (called a *codebook*) of vectors and compresses each block by writing on the compressed stream a pointer to the block in the codebook. The decoder has the easy task of reading pointers, following each pointer to a block in the codebook, and joining the block to the image-so-far (see figure (2.11)). Vector quantization is thus an asymmetric compression method.

In the case of  $2 \times 2$  blocks, each block (vector) consists of four pixels. If each pixel is one bit, then a block is four bits long and there are only  $2^4=16$  different blocks. It is easy to store such a small, permanent codebook in both encoder and decoder. However a pointer to a block in such a codebook is, of course, four bits long, so there is no compression gain by replacing blocks with pointers. If each pixel is  $k$  bits, then each block is  $4k$  bits long and there are  $2^{4k}$  different blocks [Sal00].

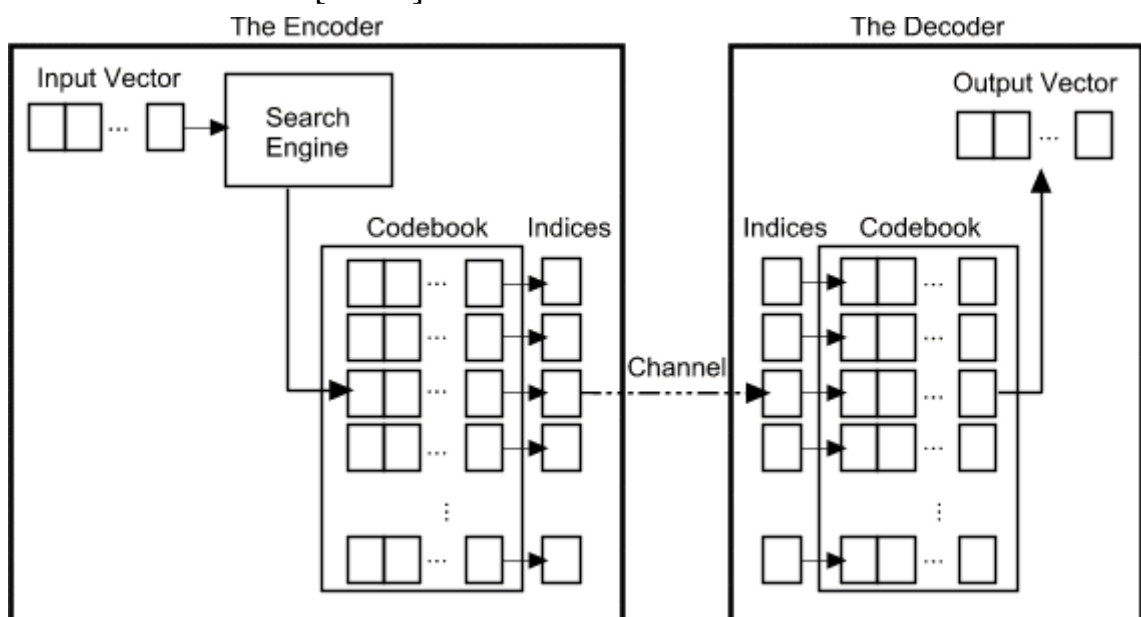


Fig. (3.11) The encoder and decoder in a vector quantization.

### 2.7.1 The LBG Design Algorithm

Designing a codebook that best represents the set of input vectors is very-hard. That means that it requires an exhaustive search for the best possible codewords in space, and the search increases exponentially as the number of codewords increases, therefore, we resort to suboptimal codebook design schemes, and the first one that comes to mind is the simplest. It is named **LBG** algorithm for **Linde-Buzo-Gray**. It sometimes is also known as K-means clustering .

This algorithm is in fact designed to iteratively improve a given initial codebook. The design of a codebook with  $N$ -codeword can be stated as follows [Lin80]:-

1. Determine the number of codewords,  $N$ , or the size of the codebook.
2. Select  $N$  codewords at random, and let that be the initial codebook. The initial codewords can be randomly chosen from the set of input vectors.
3. Using the Euclidean distance measure clusterize the vectors around each codeword. This is done by taking each input vector and finding the Euclidean distance between it and each codeword. The input vector belongs to the cluster of the codeword that yields the minimum distance.
4. Compute the new set of codewords. This is done by obtaining the average of each cluster. Add the component of each vector and divide by the number of vectors in the cluster.

$$y_i = \frac{1}{m} \sum_{j=1}^m x_{ij} \quad (2.21)$$

where  $i$  is the component of each vector (x, y, z, ... directions),  $m$  is the number of vectors in the cluster.

Repeat steps 1, 2 and 3 until the either the codewords don't change or the change in the codewords is small .

This algorithm is by far the most popular, and that is due to its simplicity. Although it is locally optimal, yet it is very slow. The reason it is slow is because for each iteration, determining each cluster requires that each input vector be compared with all the codewords in the codebook.

## 2.8 Scalar Quantization

The function of scalar quantization is to convert a quantity to a finite precision representation (such as converting real numbers to integers in digital computing) [Gib00].

The scalar quantization can be represent as [Ala02]:

1. Determine the minimum ( $M_n$ ) and maximum ( $M_x$ ) values of the image and then determine their dynamic range ( $R$ ).

$$R = M_x - M_n$$

2. Determine the number of bits allocated,  $B$ , for using image by using the following equation:

$$B = \text{Log}_2(R)$$

3. Determine the quantization coefficients of image by applying the following expression:

$$Q = \frac{R}{2^B - 1}$$

5. For each quantization coefficient of image, determine the quantization index ( $Q_I$ ) by applying scalar quantization to reduce the number of bits needed to represent, approximately, the coefficients as follows:

$$Q_I(x, y) = \frac{\text{img}(x, y) - M_n}{Q} \quad (2.22)$$

Where  $\text{img}(x, y)$  is the image coefficient at the position  $(x, y)$ , and  $Q_I(x, y)$  is the corresponding quantization index.



## 2.9 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), from these transformation methods is:

### 2.9.1 Wavelets Transform

A wavelet, which literally means little wave, is an oscillating zero-average function that is well localized in a small period of time. A wavelet function, known as a mother wavelet, gives rise to a family of wavelets that are translated (shifted) and dilated (stretched or compressed) versions of the original mother wavelet [Asu02].

The *wavelet transform* can be described as a transform that has basis functions that are shifted and expanded versions of themselves. Because of this, the wavelet transform contains not just frequency information as well. One of the most common models for a wavelet transform uses the Fourier transform and highpass filters. To satisfy the conditions for a wavelet transform, the filters must be *perfect reconstruction filters*, which means that any distortion introduced by the forward transform will be canceled in the inverse transform (an example of these types of filters are *quadrature mirror filters*).

The wavelet transform breaks an image down into four subsampled, or decimated, images. They are subsampled by keeping every other pixel. The results consist of one image that has been highpass filtered in both the horizontal and vertical direction, one that has been highpass filtered in vertical and lowpass filtered in the horizontal, one that has been lowpassed in the vertical and highpassed in the horizontal, and one that has been lowpass filtered in both directions.

Numerous filters can be used to implement the wavelet transform, and two of the commonly used ones, the Daubechies and the Haar. These are separable, so can be used to implement a wavelet transform by first convolving them with the rows and then the columns [Umb98].

### 2.9.1.1 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 (2.12). 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  $x_g$  and  $x_h$ . A DWT is obtained by further decomposing the low-pass output signal  $x_h$  by means of a second identical pair of analysis filters. This process may be repeated, and the number of such stages defines the level of the transform [Bur98].

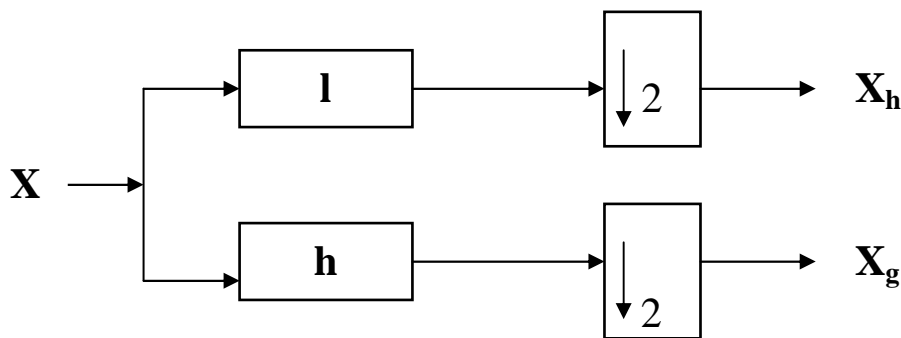


Fig. (2.12) One level wavelet decomposition.

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  $h(n)$  and low-pass filter  $l(n)$ . 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 [Alh01]:

$$\left. \begin{aligned} y_{high}(k) &= \sum_n x(n).h(2k - n) \\ y_{low}(k) &= \sum_n x(n).l(2k - n) \end{aligned} \right\} \quad (2.23)$$

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 (2.12) illustrate this procedure, where  $x(n)$  is the original signal to be decomposed, and  $l(n)$  and  $h(n)$  are low-pass and high-pass filters respectively [Alh01].

### 2.9.1.2 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) Line decomposition

In this method, the DWT is applied to each row of the image, resulting in smooth coefficients on the left (sub-band L1) and detail coefficients on the right (sub-band H1). Sub-band L1 is then partitioned into L2 and H2, and the process is repeated until the entire coefficient matrix is turned into detail coefficients (see figure 2.13). 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].

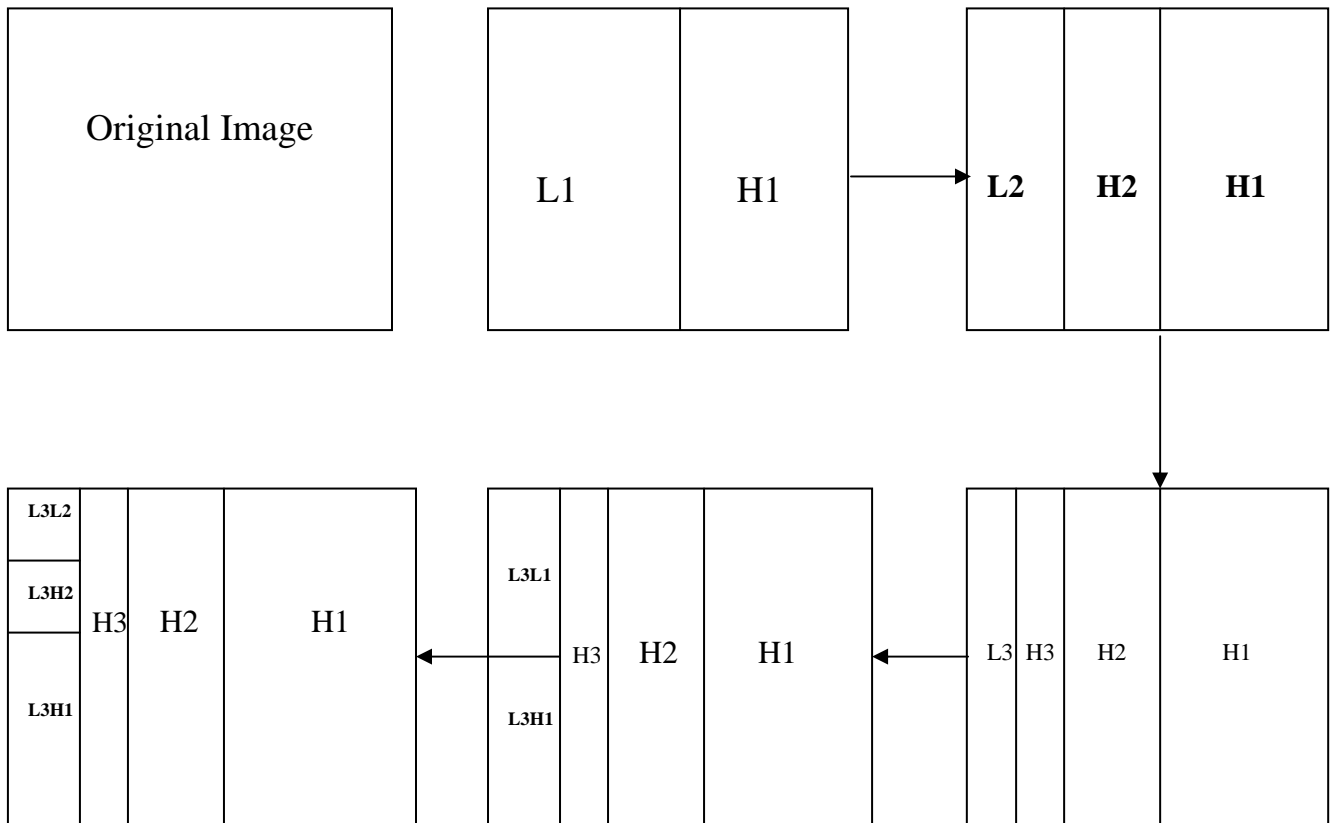


Fig. (2.13) Line Wavelet Decomposition.

**ii) Quincunx decomposition**

Quincunx decomposition, proceeds level by level and decomposes sub-band  $L_i$  of level  $i$  into sub-bands  $H_{i+1}$  and  $L_{i+1}$  of level  $i+1$ . It is efficient and computationally simple (as shown in figure 2.14). On average, it achieves more than four times the energy compaction of the line method. It results in fewer sub-bands 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 performs extremely well and may be the best performer in many practical situations [Sal00].

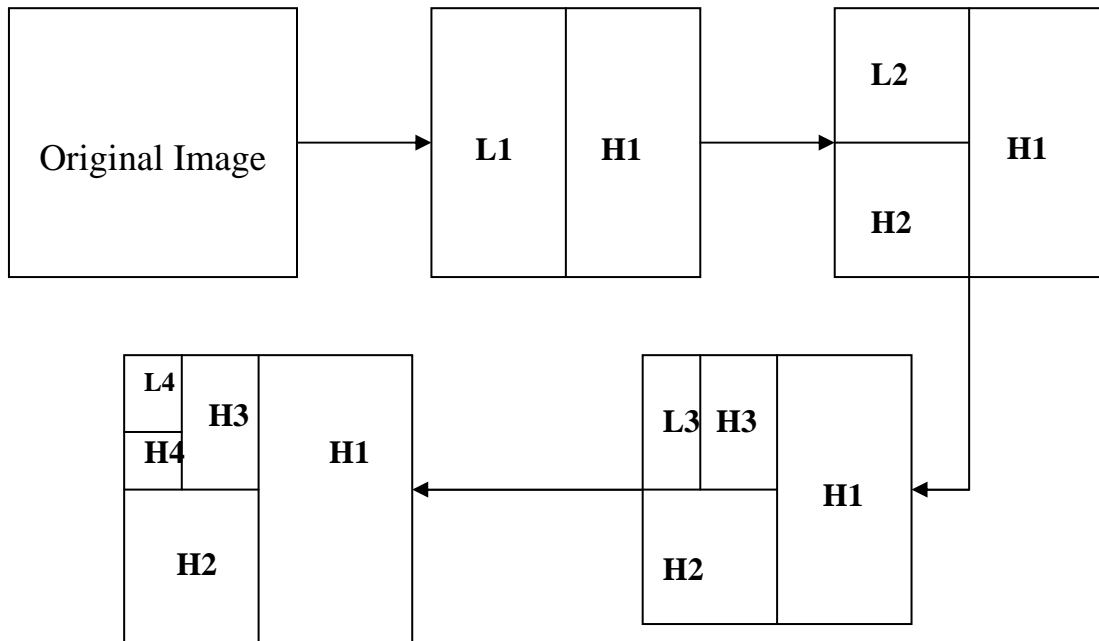


Fig. (2.14) Quincunx wavelet decomposition.

### iii) Pyramid decomposition

The pyramid decomposition (figure 2.15) 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].

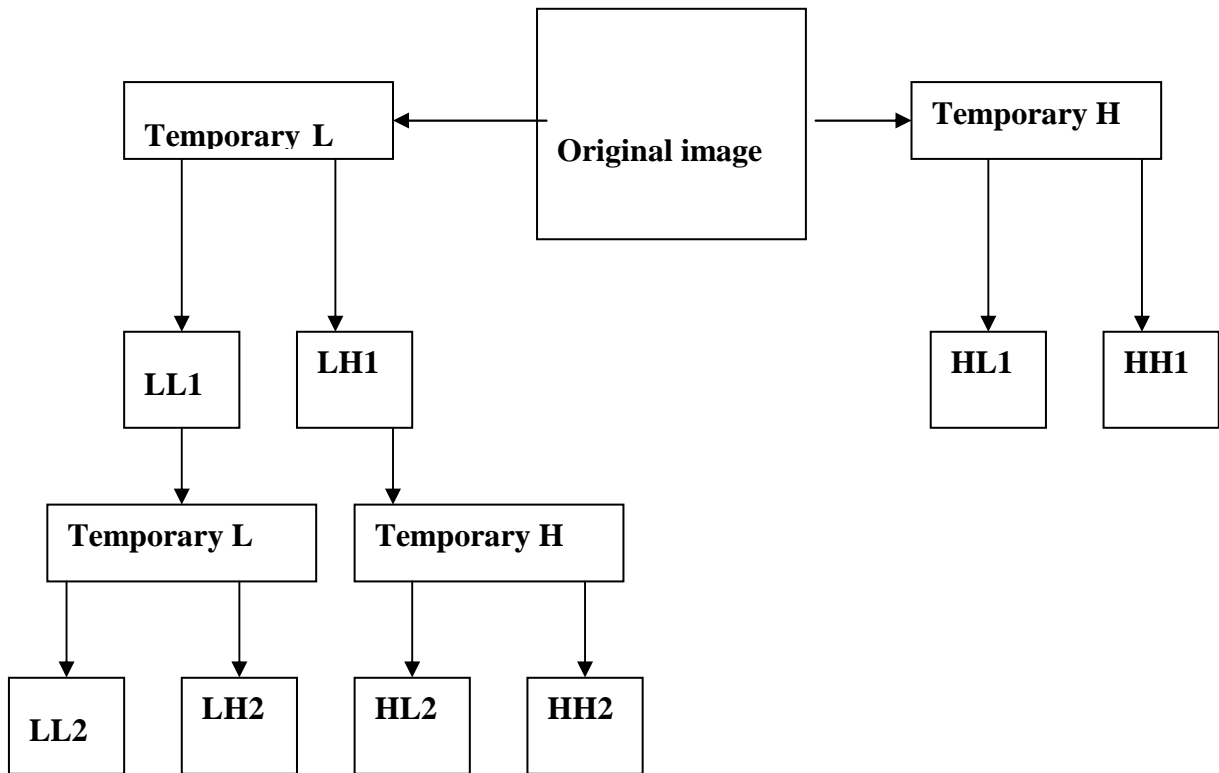
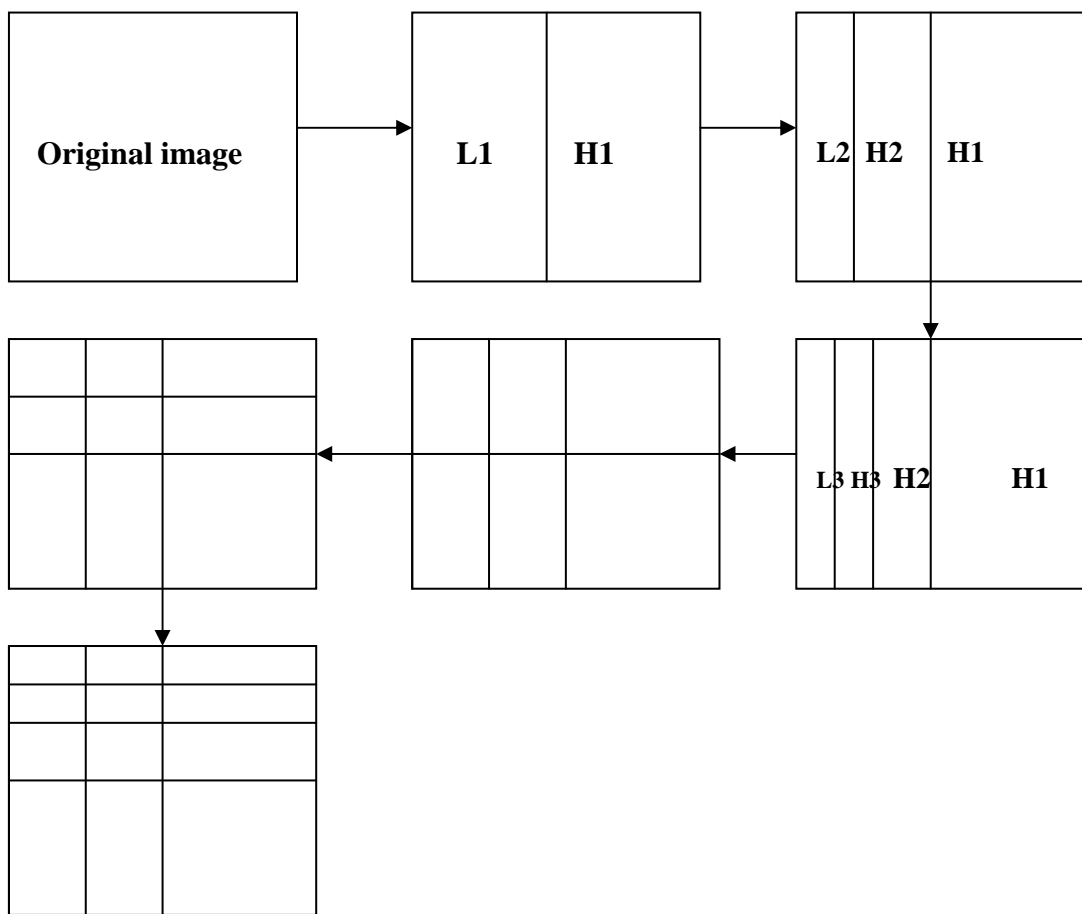


Fig. (2.15) Pyramid Wavelet Decomposition.

#### iv) Standard decomposition

The first step in the standard decomposition is to apply whatever discrete wavelet filter is being used to all rows of the image, obtaining sub-bands L1 and H1. This is repeated on L1 to obtain L2 and H2, and so on  $k$  times (as shown in figure 2.16). 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.

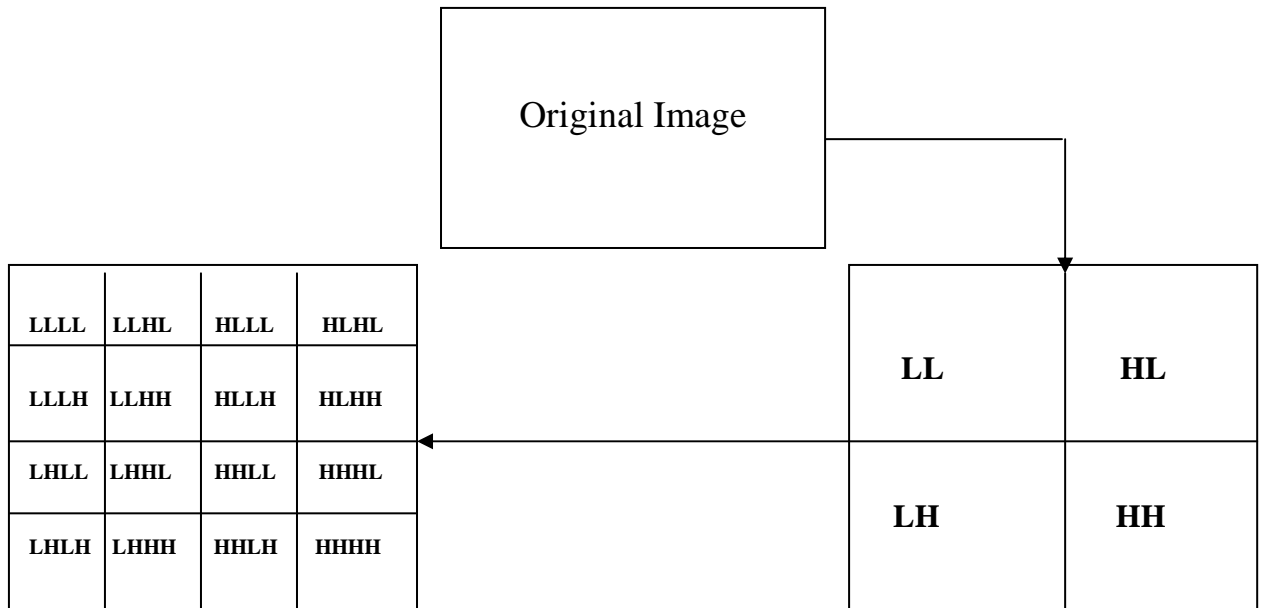
The result is to have one smooth coefficient at the top-left corner of the coefficient matrix. This method is somewhat similar to line decomposition. An important feature of standard decomposition is that when a coefficient is quantized, it may affect a long, thin rectangular area in the reconstructed image. Thus, very coarse quantization may result in artifacts in the reconstructed image in the form of horizontal rectangles [Sal00].



**Fig. (2.16)** Standard Wavelet Decomposition.

### v) Full wavelet decomposition

This type of decomposition is also called the *Wavelet Packet* transform. It is shown in figure (2.17).



**Fig. (2.17)** 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 sub-bands. The same 2-D DWT (i.e., using the same wavelet filters) is then applied recursively to each of the four sub-bands individually. The result is a coefficient matrix  $I_2$  consisting of 16 sub-bands. When this process is carried out  $r$  times, the result is a coefficient matrix consisting of  $2^r \times 2^r$  sub-bands, each of size  $2^{l-r} \times 2^{l-r}$ . The top-left sub-bands 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 sub-bands contain detail coefficients. Each sub-band corresponds to a 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].



### 2.9.1.3 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)

Haar wavelet transform consists of both: low pass and high pass filters is the preferred wavelet because it can be readily implemented in hardware [San00]. The high pass and low pass filters are called the decomposition filters because they break the image down or decompose the image into detailed and approximation coefficients, respectively.

The approximation band (LL) is the result of applying low pass filter in vertical and horizontal directions, the (LH) band is the result of applying horizontal low pass filter and vertical high pass filter, while the (HL) band is the result of horizontal high pass filter and vertical low pass filter, and finally (HH) band is the result of horizontal and vertical high pass filter. In this transform each (2x2) adjacent pixels are picked as group and passed simultaneously through four filters (i.e., LL, HL, LH, and HH) to obtain the four wavelet coefficients, the bases of these 4-filters could be derived as follows: [Ala02]

The low and high filters are:

$$\begin{aligned} L &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \end{pmatrix} \\ H &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \end{pmatrix} \end{aligned} \quad (2.24)$$

Thus the horizontal low pass followed by the vertical low pass filter is equivalent to:

$$LL = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad (2.25)$$

The horizontal high pass filter followed by vertical low pass filter is:

$$HL = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \quad (2.26)$$

While the horizontal low pass filter followed by vertical high pass filter is equivalent:

$$LH = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix} \quad (2.27)$$

And finally, the horizontal high pass filter followed by vertical high pass filter is:

$$HH = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad (2.28)$$

## ii) Inverse Haar Wavelet Transform (IHWT)

The inverse one-dimensional HWT is simply the inverse to those applied in the FHWT; the IHWT equation are [Jia03]:

i) If N is even

$$\left. \begin{aligned} x(2i) &= \frac{L(i) + H(i)}{\sqrt{2}} & , i=0 \dots N/2-1 \\ x(2i+1) &= \frac{L(i) - H(i)}{\sqrt{2}} & , i=0 \dots N/2-1 \end{aligned} \right\} \quad (2.29)$$

ii) If N is odd

$$\left. \begin{aligned} x(2i) &= \frac{L(i) + H(i)}{\sqrt{2}} & , i=0 \dots (N-1)/2 \\ x(2i+1) &= \frac{L(i) - H(i)}{\sqrt{2}} & , i=0 \dots (N-1)/2 \\ x(N-1) &= L\left(\frac{N+1}{2}\right)\sqrt{2} \end{aligned} \right\} \quad (2.30)$$

Where

N is the number of pixels.

L is the low frequencies subbands.

H is the high frequencies subbands.

### **2.9.1.4 Integer Wavelet Transform(IWT)**

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) [Maj97].

### **2.9.2 Wavelets features for Image Compression**

This is a summary of some features of image compression using Wavelets [Kha03]:

1. Wavelet transform has a good energy compact, it is preserved across the transform, i.e. the sum of squares of the wavelet coefficients is equal the sum of squares of the original image.
2. Wavelets can provide a good compression, it can perform better than JPEG2000, both in terms of SNR and image quality. Thus show no blocking effect unlike JPEG2000.
3. The entire image is transformed and compressed as a single data object using wavelet transforms, rather than block by block. This allows for uniform distribution of compression error across the entire image and at all scales.

4. The wavelet transform methods have been shown to provide integrity at higher compression rates than other methods where integrity of data is important e.g., medical images and fingerprints, etc.
5. Multiresolution properties allow for progressive transmission and zooming, without extra storage.
6. It is a fast operation performance, in addition to symmetry: both the forward and inverse transform have the same complexity, in both compression and decompression phases.
7. Many image operations such as noise reduction and image scaling can be performed on wavelet transformed images.

## 2.10 Compression Efficiency Parameters

As mentioned previously, image compression techniques can be distinguished in two classes if our concern is the exact reconstruction of the original data or not. These include lossy and lossless. All the adopted coding methods in the present work are classified as a lossy compression method. Therefore, to evaluate the compression efficiency of the proposed methods, the following parameters were utilized in the current analysis,

### 2.10.1 Compression Ratio

It is defined as the ratio between the size of the original image data (file) and the size of overall compressed data file [Ros82].

$$C.R. = \frac{\text{Original File Size}}{\text{Compressed File Size}} \quad (2.31)$$

### 2.10.2 Fidelity Criteria

Mostly, compression techniques cause some information losses, up to a certain tolerated level. Thus a use of fidelity criteria is required to measure the amount of lossess. Two basic ways of fidelity criteria are used [Umb98]:

- 1-Subjective fidelity criteria, and
- 2- Objective fidelity criteria.

Subjective fidelity criteria require the definition of a qualitative scale to assess image quality. This scale can then be used by human test subjects to determine image fidelity. In order to provide unbiased results, evaluation with subjective measures requires careful selection of the test subjects and carefully designed evaluation experiments.

When the level of information loss can be expressed as a function of the original and reconstructed image, it is said to be based on an objective fidelity criterion. Different objective criteria have been considered in the literature, among the most commonly used is the *mean square error (MSE)*,

$$MSE = E\{(f - f')^2\} = \frac{1}{N \times M} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} [f(x, y) - f'(x, y)]^2 \quad (2.32)$$

Where  $f, f'$  represent the original and the reconstructed image.

Considering the difference between the original and the reconstructed images as a noise a closely related objective fidelity criteria is the *mean signal to noise ratio (MSNR)* is used which is given by [Jor97]:

$$MSNR = \frac{E\{f^2\}}{E\{(f - f')^2\}} \quad (2.33)$$

Or

$$MSNR = \frac{\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f^2(x, y)}{\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} [f(x, y) - f'(x, y)]^2} \quad (2.34)$$

The equivalent definition to the (MSNR) is the *Peak Signal –to- Noise Ratio (PSNR)* defined as:

$$PSNR = \frac{[\text{peak to peak of } f(x, y)]^2}{MSE} \quad (2.35)$$

The PSNR value is usually measured in decibel (dB), which is given by

$$PSNR(dB) = 10 \log_{10} \left[ \frac{(\text{gray scale of image})^2}{MSE} \right] \quad (2.36)$$

For 8-bit gray scale image the PSNR is defined as:

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \quad (2.37)$$

Here, PSNR is calculated relative to the dynamic range rather than to signal variation, while SNR was calculated relative to the signal variance. The values of PSNR are about 6dB above those of SNR. Therefore, PSNR is not only easier to calculate than SNR but it also gives more optimistic results, and this may account for its popularity in reporting the results of compression.

For color images, the reconstruction of all three planes must be considered in the PSNR calculation. The MSE is calculated for the reconstruction of each color plane. The average of these three MSE is used to generate the PSNR of the reconstructed RGB image:

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE_{RGB}} \right) \quad (2.38)$$

Where,

$$MSE_{RGB} = \frac{MSE_{red} + MSE_{green} + MSE_{blue}}{3} \quad (2.38a)$$

Where  $MSE_{red}$  (or green or blue) is similar to equation(2.34) for each color plane.

Equation (2.40a) give the same weight for each plane, but as known the human visual system is more sensitive to green plane and less to red plane and much less to blue plane as shown the following equation [San98]:

$$Y = 0.299R + 0.587G + 0.114B \quad (2.39)$$

Therefore, in the present work the  $MSE_{RGB}$  was calculated according to the following weighted expression as follows:

$$MSE_{RGB} = 0.299MSE_{Red} + 0.587MSE_{Green} + 0.114MSE_{Blue}$$

Nevertheless, there is no commonly accepted method for calculation of the PSNR of color images. Moreover, the results obtained using PSNR often do not coincide with those obtained in subjective tests. It is well known that some small subjective errors randomly distributed over the whole image may reflect badly in objective measures, i.e., lead to small values of PSNR but they often correspond to invisible degradation of image quality. On the other hand, relatively high valued errors concentrated in a particular part of an image cannot affect the objective error, i.e., can permit the PSNR to be relatively high, while the subjective assessment is very low because of an annoying corruption of a particular portion of the picture. These observations prompt questions about the usefulness of the objective measures, which however, exhibit a very important advantage, i.e., they are easy to calculate. This fundamental advantage makes PSNR very popular.

Image compression research literature typically reports performance based solely on the quantitative PSNR metric. A recent survey of quality measurements indicates the widely used PSNR measure often does not reflect spurious artifacts and does not always correlated with visual error perception.

## 2.11 Programming Work

All the computations involved with the proposed coding methods were accomplished by utilizing a computer programming, the programs had been designed and implemented to achieve the coding and testing tasks. *Visual Basic (VB)* (version 6) under the Windows XP operating system was adopted to perform the programming work. The programs are implemented and executed for testing purpose using Asus personal computer, with processor Pentium-4. All the test tasks were performed on three color images, each image has 24b/p and its size is 256×256 pixel.

## Chapter 3

# Hybrid Wavelet Modified Vector Quantization (WMVQ)

### 3.1 Introduction

In this chapter, we proposed method which combines the WT and modified VQ. Wavelet transform decompose the image, most of the information is concentrated in the lowest frequency subband. Any distortion in the lowest frequency subband causes critical defects on the quality of the reconstructed image.

### 3.2 Color Images Transformation

Most color images are recorded in RGB model, which is the most well known color model. However, RGB model is not suited for some image processing purposes due to a number of reasons. Firstly, there exist significant information redundancy amongst the color planes; therefore it may not be suitable for image processing tasks such as image coding. Secondly, it is not a uniform color model, which means that the distance between some pairs of points in the RGB model corresponds to an unnoticeable subjective color difference while the same distance in another part of the model corresponds to a quite significant difference in color sensation [San98].

For compression, a luminance-chrominance representation is considered superior to the RGB representation. Therefore, RGB images are transformed to one of the luminance-chrominance models, performing the compression process, and then transform back to RGB model because displays are most often provided output image with direct RGB model. The



luminance component represents the intensity of the image and looks like a gray scale version. The chrominance components represent the color information in the image.

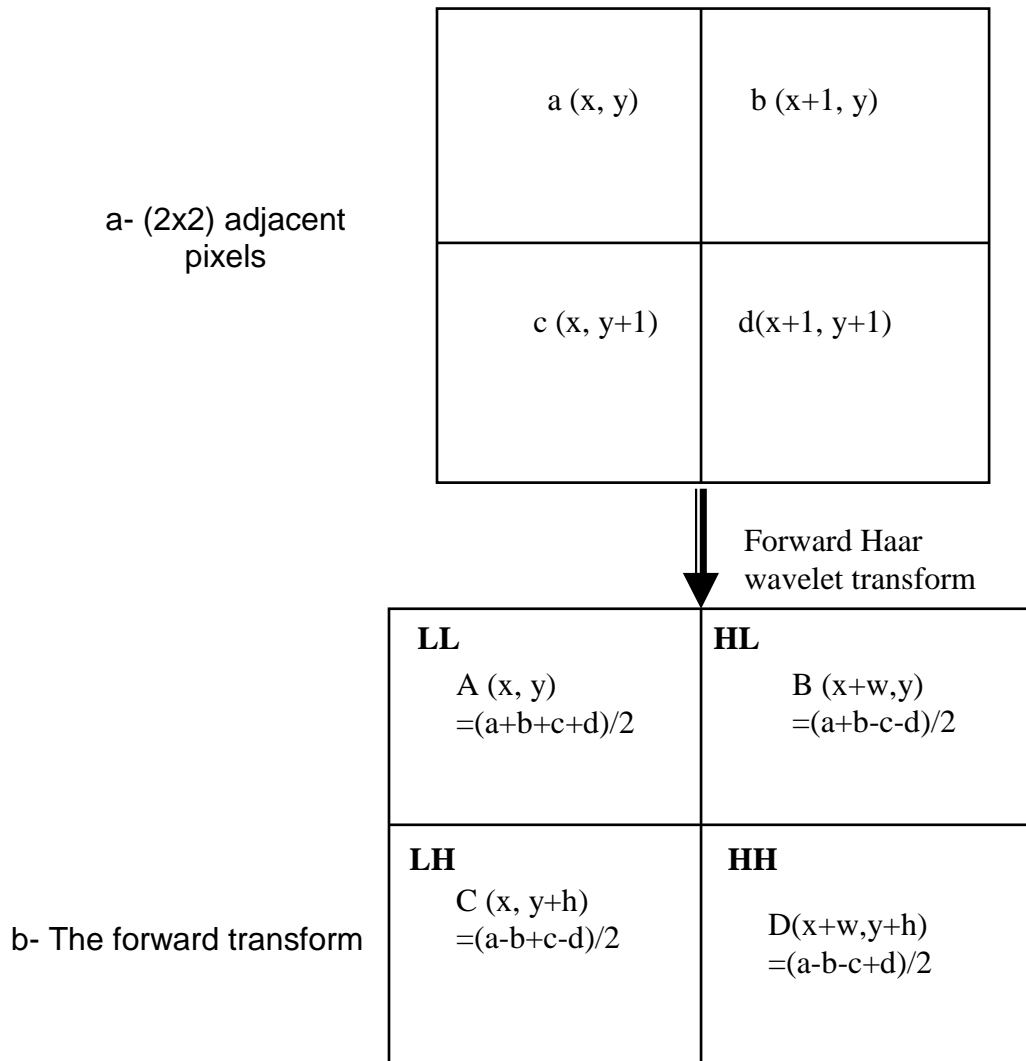
### **3.3 Suggested Wavelet Transform**

This section is dedicated to demonstrate the suggested scheme of wavelet compression using Haar filter. In the present work, the mechanism of the WT system is based on the idea of combining the horizontal lowered high pass filters with the vertical corresponding filters to constructed four 2-dimensional Haar filters, the direction implementation of these filters will lead to significant reduction in both coding and decoding process time in comparison with the traditional mechanism which is based on the sequential application of the horizontal followed by vertical filter.

The suggested wavelet transform can be shown as in the following steps:

#### **3.3.1 Forward Haar Wavelet Transform**

Figure (3.1) illustrates the steps of forward Haar wavelet transform, where  $a$ ,  $b$ ,  $c$ , and  $d$  are the image pixel values, while  $A$ ,  $B$ ,  $C$ , and  $D$  are the corresponding wavelet coefficients,  $w$  and  $h$  are half of the image width and height, respectively.



**Fig. (3.1)** The forward Haar wavelet transform.

To apply forward Haar wavelet transform on the test color images, those shown in figure (3.2). Of the beginning of decomposition, the image will be decomposed into four subbands LL, HL, LH, and HH (see figures, 3.3b, 3.4b, and 3.5b on luminance component). This procedure defines first level subband coding, which contains the detailed features of the image (also referred to as the high frequency or fine resolution wavelet coefficients). While figures (3.3c, and d), (3.4c, and d) and figures (3.5c, and d) define the second and third subband levels, respectively. The top level, i.e. third level, contains the coarse image features (low frequency or coarse resolution wavelet coefficients). It is clear that the lower levels can be quantized coarsely without much loss of important image information, while the higher

levels should be quantized finely. These ten subbands can be recombined to produce the original image by applying inverse transform, which is explained in the next section.



Satellite Image



Monaliza Image

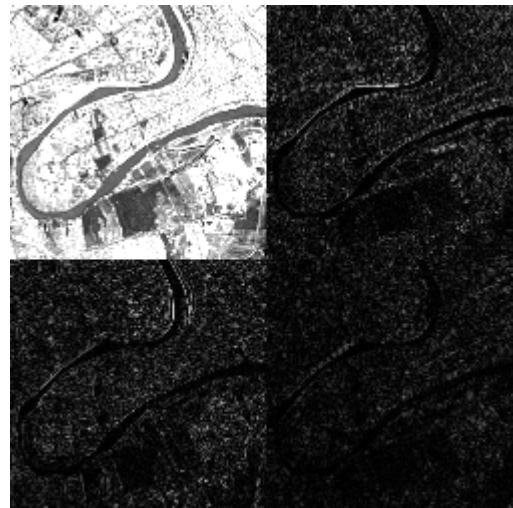


Lina Image

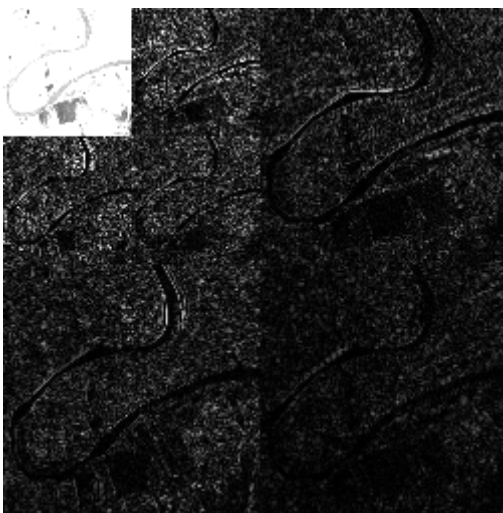
**Fig. (3.2)** The test color images.



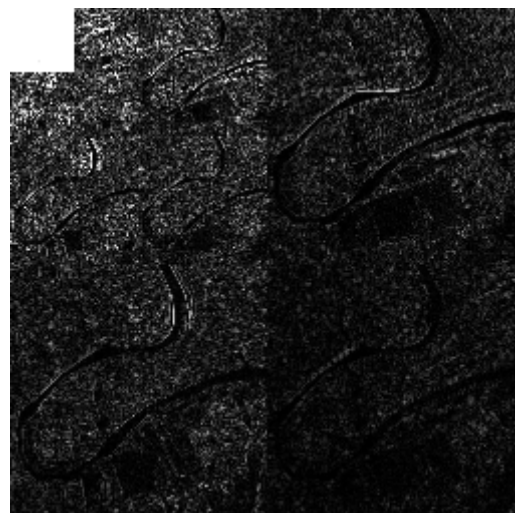
a) Luminance component of satellite image.



b) Forward Haar wavelet transform, four bands.



c) Forward Haar wavelet transform, seven bands

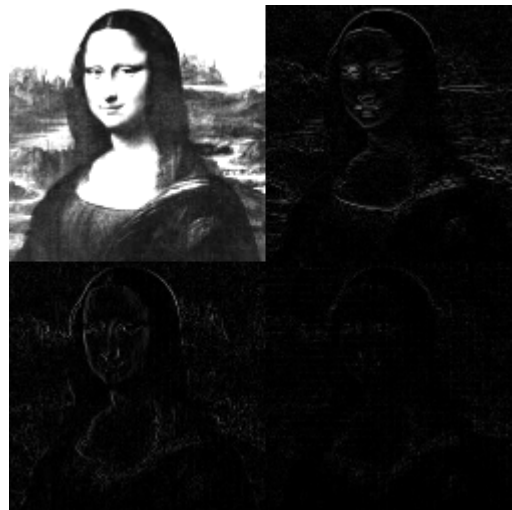


d) Forward Haar wavelet transform, ten bands.

**Fig. (3.3)** Forward Haar wavelet transform applied on luminance component of satellite image.



a) Luminance component of Monaliza image.



b) Forward Haar wavelet transform, four bands.



c) Forward Haar wavelet transform, seven bands.



d) Forward Haar wavelet transform, ten bands.

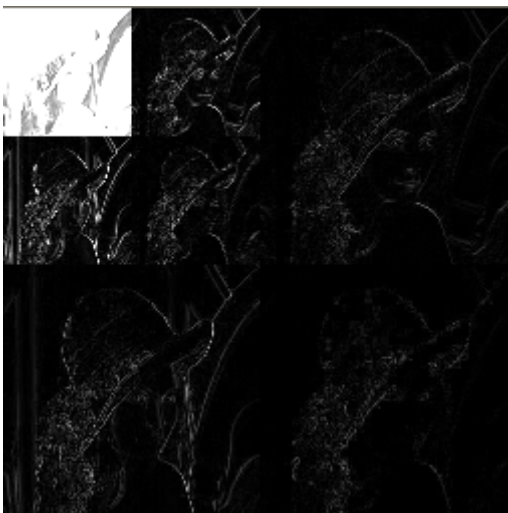
**Fig. (3.4)** Forward Haar wavelet transform applied on luminance component of Monaliza image.



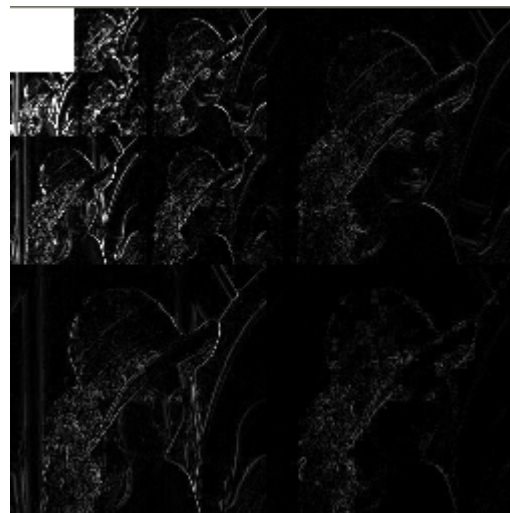
a) Luminance component of Lina image.



b) Forward Haar wavelet transform, four bands.



c) Forward Haar wavelet transform, seven bands.

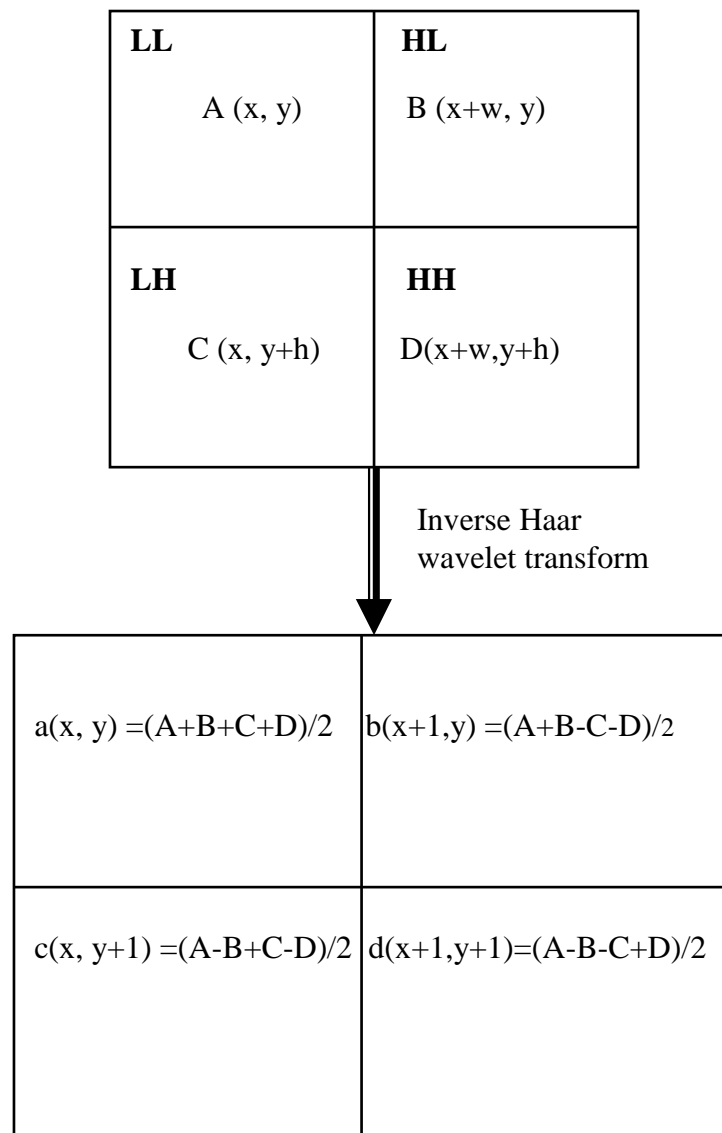


d) Forward Haar wavelet transform, ten bands.

**Fig. (3.5)** Forward Haar wavelet transform applied on luminance component of Lina image.

### 3.3.2 Inverse Haar Wavelet Transform

The output of forward Haar wavelet transform is the wavelet coefficients of the (LL, HL, LH, and HH) bands. To reconstruct the image, the same four two dimensional filters (which are mentioned in section 2.2.1) have been used. Figure (3.6) demonstrates the inverse Haar wavelet transform, where A, B, C, and D are the wavelet coefficients, while a, b, c, and d are the reconstructed pixel values.



**Fig. (3.6)** The inverse Haar wavelet transform.

### **3.4 Modified Vector Quantization (MVQ)**

In this section, the following modified algorithm for improving the quality of reconstructed images. The main steps of this modification are:

1. For an  $M \times M$  image, the image is first partitioned into fixed size square blocks, each block of size  $n \times n$ .
2. Form an initial codebook by choosing the first  $N$ -input image blocks as reproduction vectors.
3. Compare each input vector with all  $N$ -reproduction vectors. Best match is achieved when the minimum mean square error (MSE) between the reproduction and the input vectors is within a pre-specified threshold. In this case the input that matches vector with minimum distance should be given the same index of this reproduction vector.
4. For each index, find the centroid of all input vectors. The centroids are the new codebook.
5. Sort the codebook vectors in descending order from high count to low count.
6. Eliminate the last reproduction vector, which has very low count and split the first reproduction vector (i.e., high count) into two vectors by multiplying the vector contents by enlargement/reduction factors (say, 1.1/0.9) to reproduce two new vectors.
7. The above steps (3-6) are repeated until the centroids redistribution converges to solution, which is a minimum of the total reproduction error.

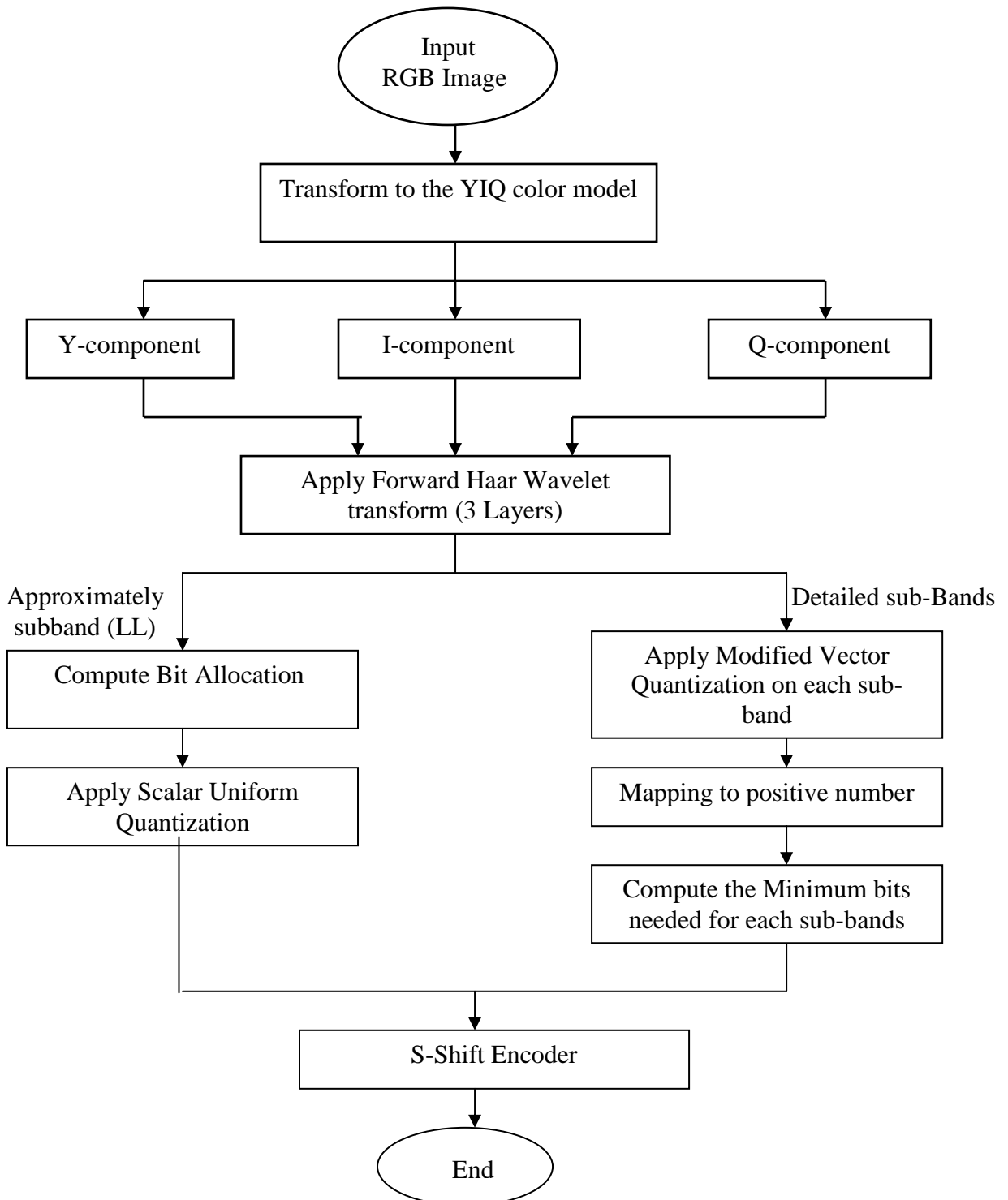
### **3.5 Wavelet Modified Vector Quantization Scheme (WMVQ)**

In this section, we will discuss the proposed compression method based on wavelet transform and modified vector quantization. The suggested method can be divided into two units:



### 3.5.1 Encoding Unit.

The encoding process can be shown in the flowchart of figure (3.7) and implies in the following steps:



**Fig. (3.7)** Flowchart of encoder of suggested method.

1. Convert the RGB image to YIQ space. Algorithm(3.1) shows this conversion.

**Algorithm (3.1) conversion the color image to YIQ image.**

**Input:**  $R(), G(), B()$  2D array of RGB color model

$W$  is the image width

$H$  is the image height

$bitY, bitI, bitQ$  the number of bits needed for each component  $Y, I,$  and  $Q$

**Output:**  $Y(), I(), Q()$  2D array of YIQ color model

**Procedure:**

For all coefficients  $(x, y)$  where  $0 \leq x \leq W-1$  and  $0 \leq y \leq H-1$

Set  $Y(x, y) \leftarrow 0.299 * R(x, y) + 0.587 * G(x, y) + 0.114 * B(x, y)$

Set  $I(x, y) \leftarrow 0.596 * R(x, y) - 0.274 * G(x, y) - 0.322 * B(x, y)$

Set  $Q(x, y) \leftarrow -0.211 * R(x, y) - 0.523 * G(x, y) + 0.312 * B(x, y)$

End Loop  $x, y$

Find the minimum and maximum value for  $Y, I,$  and  $Q$ .

Find the range between maximum value and minimum value for each  $Y, I,$  and  $Q$ .

Set  $oneY \leftarrow (2^{bitY} - 1) / rangY$

Set  $oneI \leftarrow (2^{bitI} - 1) / rangI$

Set  $oneQ \leftarrow (2^{bitQ} - 1) / rangQ$

For all coefficients  $(x, y)$  where  $0 \leq x \leq W-1$  and  $0 \leq y \leq H-1$

Set  $Y(x, y) \leftarrow (Y(x, y) - minY) * oneY$

Set  $I(x, y) \leftarrow (I(x, y) - minI) * oneI$

Set  $Q(x, y) \leftarrow (Q(x, y) - minQ) * oneQ$

End Loop  $x, y$

2. Apply forward Haar wavelet transform on luminance component and chrominance components, individually, to produce (L) number of layers. The total number of subbands will be  $(3 \times L + 1)$  as it is shown in figure (3.8). Algorithm (3.2) decomposes the image into subbands, such kind of decomposition reduces the image data correlation and provides a useful data structure. The results have been gathered in a single array named `img()`.

**Algorithm (3.2) Haar Wavelet Transform**

**Input:** `img()` 2D array represent the coefficients (Y, I, Q)

*W* is the image width

*H* is the image height

*no\_of\_level* the number of layer of wavelet

**Output:** rec of sub images `rec().LL()`, `rec().LH()`, `rec().HL()`, and `rec().HH()`.

**Procedure:**

Loop *i* = 0 to *no\_of\_layer-1*

For all coefficients (*x*, *y*) where  $0 \leq x \leq W/2-1$  and  $0 \leq y \leq H/2-1$

Set `rec(i).LL(x, y)`  $\leftarrow 0.5 * (\text{img}(2 * x, 2 * y) + \text{img}(2 * x, 2 * y + 1) + \text{img}(2 * x + 1, 2 * y) + \text{img}(2 * x + 1, 2 * y + 1))$

Set `rec(i).HL(x, y)`  $\leftarrow 0.5 * (\text{img}(2 * x, 2 * y) + \text{img}(2 * x, 2 * y + 1) - \text{img}(2 * x + 1, 2 * y) - \text{img}(2 * x + 1, 2 * y + 1))$

Set `rec(i).LH(x, y)`  $\leftarrow 0.5 * (\text{img}(2 * x, 2 * y) - \text{img}(2 * x, 2 * y + 1) + \text{img}(2 * x + 1, 2 * y) - \text{img}(2 * x + 1, 2 * y + 1))$

Set `rec(i).HH(x, y)`  $\leftarrow 0.5 * (\text{img}(2 * x, 2 * y) - \text{img}(2 * x, 2 * y + 1) - \text{img}(2 * x + 1, 2 * y) + \text{img}(2 * x + 1, 2 * y + 1))$

End Loop *x, y*

Set *H*  $\leftarrow (H / 2)$

Set *W*  $\leftarrow (W / 2)$

For all coefficients (*x*, *y*) where  $0 \leq x \leq W-1$  and  $0 \leq y \leq H-1$

Set `img(x, y)`  $\leftarrow \text{rec}(i).LL(x, y)$

End Loop *x, y*

End Loop *i*

I=3 k=10	I=3 k=7	I=2 k=4	I=1 k=1
I=3 k=8	I=3 k=9		
I=2 k=5		I=2 k=6	
I=1 k=2			I=1 k=3

**Fig. (3.8)** An example illustrates the relationship between subband number (k) and the subband layer number (I), where the total number of subband layers is taken (L).

- Determine the quantization coefficients of LL subband by applying the equations used in section 3.7.2:

Algorithm (3.3) illustrates the bit allocation and uniform quantization implementation for subband LL, the results have been gathered in a single array name `img()`.

**Algorithm (3.3) Mapping by using bit allocation and Uniform Quantization**

**Input:** *img()* 2D array represent the LL-band.

*H* is the subband LL width.

*W* is the subband LL height.

**Output:** *q(x, y)* 2D array that represent LL-band quantized.

*Bitsclr* represent the number of bits needed to quantized LL-band.

**Procedure:**

*Find the minimum and maximumu number of the subband LL.*

*Set rang*  $\leftarrow$  *max - min*

*Set bitsclr*  $\leftarrow$   $\text{Log}_2(\text{rang})$

*For all coefficients (x, y) where*  $0 \leq x \leq W-1$  *and*  $0 \leq y \leq H-1$

*Set*  $q(x, y, k) \leftarrow (\text{img}(x, y, k) - \text{min}) / \text{rang} * (2^{\text{bitsclr}} - 1)$

*End Loop* *x, y*

4. Eliminate the highest frequency subbands.
5. Apply (MVQ) mechanism which described in section (3.4) for middle subbands using small block size. Decrease the codebook size as the subband number increases.

Algorithm (3.4) illustrates MVQ mechanism. The results form each component and band is gathered by array *VQ()* that input to the algorithm.

**Algorithm (3.4) Modified Vector Quantization**

**Input:**  $VQ()$  array of (LH, HL, HH) of YIQ Coefficients.

$BlkSiz$  represent the block size.

$MaxNoblk$  represent the CodeBook size .

$W$  is the subband width.

$H$  is the subband Height.

**Output:** Codebook() 3D array that contain the average blocks from subband.

$Idx()$  represent the index of each block (x, y) in subband.

**Procedure:**

\*\*\*\*\* Initialize Codebooks Vectors \*\*\*\*\*

$NoBlk \leftarrow -1$

Loop for all blocks in the subband

$Blk() \leftarrow VQ()$

If  $NoBlk < (MaxNoblk - 1)$  Then  $Flg \leftarrow 0$

Call algorithm (3.5) to find if block similar to block in CodeBook()  
or not

If  $Flg = 0$  Then

$NoBlk \leftarrow NoBlk + 1$

Initial Codebook() for  $NoBlk$  by  $Blk()$

End If

End If

End Loop

$TotErr \leftarrow 9.999E+19$ ;  $Iter \leftarrow 0$

Do

$Iter \leftarrow Iter + 1$ ;  $OTotErr \leftarrow TotErr$

For all blocks( $NoBlk$ ) in codebooks initial account() for that blocks by  
zero and the  $acm()$  for that block by zero.

To be continue

*\*\*\*\*\* Calculate the Centroids \*\*\*\*\**

*TotErr ← 0*

*Make for loop for all blocks in the subband and then*

*Call algorithm (3.5) to found what block in the CodeBook is similar to that block in the subband and then add account for that block by 1*

*Add acm for that block in the codebook with the block in the subband.*

*TotErr ← TotErr + Dist*

*End Loop*

*TotErr ← TotErr / (H \* W)*

*If TotErr < OTotErr Then*

*Loop For k = 0 To NoBlk*

*If Count1(k) > 0 Then*

*Divide acm() for that block k by account(k) and then put in the CodeBok for block(k)*

*Else*

*Set Codebook() for block k by zero*

*End If*

*End Loop k*

*Sort the vectors in the Codebook() in ascending order depend on the account of blocks in subband.*

*k ← 0*

*M1 ← NoBlk*

*While Count1(M1) < 1*

*For all coefficients (x, y) where  $0 \leq I_x \leq Blksz$  and  $0 \leq I_y \leq Blksz$*

*j ← Codebook(k, I<sub>x</sub>, I<sub>y</sub>)*

*Codebook(M1, I<sub>x</sub>, I<sub>y</sub>) ← CInt(j \* 0.9)*

*j ← CInt(j \* 1.1)*

*If j > 255 Then j ← 255*

*Codebook(k, I<sub>x</sub>, I<sub>y</sub>) ← j*

*End Loop x, y*

*To be continue*

```

    k ← k + 1: M1 ← M1 - 1
  End While M1
End If
Loop Until (Iter = NoIter) Or (OTotErr < TotErr)

Loop for each block in the subband
  Dist = 255 * bsz2: M1 = 0
  Call algorithm (3.5) to number the index for that block in the
  subband by the number block in the CodeBook
  Idx(Bx, By) ← M1
End Loop

```

### Algorithm (3.5) Find Block

**Input:** *Blk()* array of 2D.

*BlkSiz* represent the block size.

*CodeBook()* array of 3D that contains number of blocks.

*NoBlk* represent number of blocks in the *CodeBook()*.

**Output:** *M1* represent the number block in the *CodeBook()*.

*Flg* represent flag number.

**Procedure:**

Loop For  $k = 0$  To *NoBlk*

$Dis \leftarrow 0$

For all coefficients  $(x, y)$  where  $0 \leq x \leq Blksz$  and  $0 \leq y \leq Blksz$

$j \leftarrow CodeBook(k, x, y) - Blk(x, y)$

$Dis \leftarrow Dis + Abs(j)$

End Loop  $x, y$

If  $Dis < Dist$  Then

$Dist \leftarrow Dis: M1 \leftarrow k: Flg = 1$

End If

End Loop  $k$



6. For the detailed bands the coefficients values of codebook  $W_c(k, x, y)$  will have positive and negative values. Since the encoding process is much easier when the range of coded parameters is positive, thus the determined coefficients are mapped to the positive range by using the following mapping formula:

$$W'_c(x, y, k) = \begin{cases} 2W_c(x, y, k) & \text{if } W_c(x, y, k) \geq 0 \\ 2|W_c(x, y, k)| - 1 & \text{otherwise} \end{cases}$$

In this case all positive values will mapped to even values and all negative values will correspond to odd values.

7. Compute the minimum bits for every codebook in each detailed subbands by computing the maximum number in the codebook and compute the minimum bits need it for that subbands as shown in algorithm (3.6).

**Algorithm (3.6) Mapping and S-Shift Optimizer****Input:** *img()* array of coefficients values of detailed sub-bands.*W* is the image width.*H* is the image height.**Output:** Number of required bits (*Nb*, *Nbmax*)**Procedure:**

"" Do the positive mapping""

*For all coefficients (x, y) where  $0 \leq x \leq W-1$  and  $0 \leq y \leq H-1$* *If  $img(x, y, k) > 0$  then  $Img(x, y, k) \leftarrow 2 * img(x, y, k)$* *Else**If  $img(x, y, k) < 0$  then  $Img(x, y, k) \leftarrow 2 * abs(img(x, y, k)) - 1$* *End if*

"" Compute the max number""

*max*  $\leftarrow 0$ *For all coefficients (x, y) where  $0 \leq x \leq W-1$  and  $0 \leq y \leq H-1$* *If  $img(x, y, k) > max$  Then  $max \leftarrow img(x, y, k)$* *End Loop x, y*

"" Compute maximum number of required bits for max number""

*p*  $\leftarrow 1$ ; *o*  $\leftarrow 1$ *While  $max \geq o$* *o*  $\leftarrow 2 * o + 1$ *p*  $\leftarrow p + 1$ *End Loop While*

"" Compute the histogram Hist() of the quantized transform coefficients""

*Loop For j = 0 To max**hist(j) = 0**End Loop j**To be continue*

```

For all coefficients (x, y) where  $0 \leq x \leq W-1$  and  $0 \leq y \leq H-1$ 
   $j \leftarrow \text{img}(x, y, k)$ 
   $\text{hist}(j) \leftarrow \text{hist}(j) + 1$ 
End Loop x, y

""Shift coding optimizer to compute the number of required bits (Nb,
Nbmax)
siz  $\leftarrow W * H$ 
nbmax  $\leftarrow p$ 
minbits  $\leftarrow \text{nbmax} * \text{siz}$ 
nb  $\leftarrow \text{nbmax}$ 
Loop For j = 2 To nbmax - 1
  m  $\leftarrow 2^j - 1$ 
  sm  $\leftarrow 0$ 
  Loop For s = 0 To Max
    If s < m Then
      sm  $\leftarrow sm + j * \text{hist}(s)$ 
    Else
      sm  $\leftarrow sm + (j + \text{nbmax}) * \text{hist}(s)$ 
    End If
  End Loop s
  If sm < minbits Then
    minbits  $\leftarrow sm$ 
    nb  $\leftarrow j$ 
  End If
End Loop j

```

## 8. S-Shift Encoder

In this stage, the input data are the quantization indices values  $Q(x, y, k)$  of subband LL and the coefficients values of the subbands (LH, HL, HH) both results have been gathered in a single array named S(). The codewords produced by applying Shift-coding on the array S() are send to

the compression bit stream, which represents the compressed data file (see algorithm (3.7)).

**Algorithm (3.7) S-Shift Encoder**

**Input:**  $A()$  2D array represent the mapped values of the sub-band coefficients (LL, LH, HL, HH) of record from  $(Y, I, Q)$ .

$W$  is the image width

$H$  is the image height

$nb$  the length (in bits) of the first shift codeword

**Output:** A set of integers whose lengths are either  $Nb$  or  $Nbmax$

**Procedure:**

$l \leftarrow 0$

$Max \leftarrow (2^{nb}) - 1$

For all coefficients  $(x, y)$  where  $0 \leq x \leq W-1$  and  $0 \leq y \leq H-1$

If  $Abs(A(x, y)) < Max$  Then

$A1(l) \leftarrow A(x, y)$

$l \leftarrow l + 1$

Else

If  $A(y, x) > 0$  Then

$A1(l) \leftarrow Max$

$A1(l + 1) \leftarrow A(x, y) - Max$

$l \leftarrow l + 2$

Else

$A1(l) \leftarrow -Max$

$A1(l + 1) \leftarrow (A(x, y) + Max)$

$l \leftarrow l + 2$

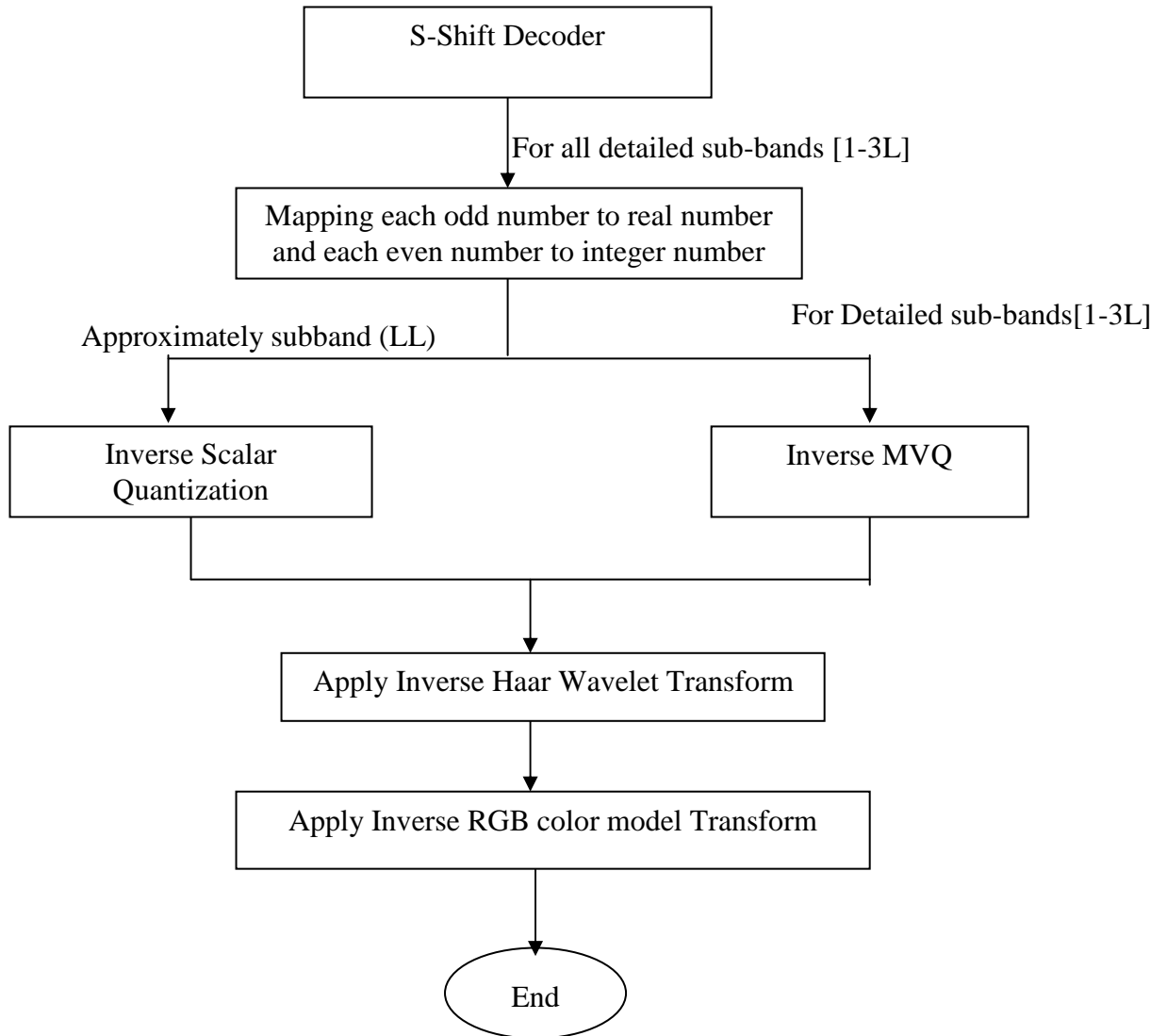
End If

End If

End Loop  $x, y$

### 3.5.2 Decoding Unit.

The decoding process of the proposed method can be shown in the Flowchart of figuer (3.9) and implies the following steps:



**Fig. (3.9)** Flowchart of decoder of suggested method.

1. Loading the compressed data as a one dimensional array of bits.
2. Perform S-Shift decoding, which is opposite to S-Shift encoding. The input to the S-Shift decoder is the length (in bits) of the coefficient value which is previously archived in the compression file by using S-Shift encoder, and the output is the coefficient value. Algorithm (3.8) illustrates this process.

**Algorithm (3.8) S-Shift decoder**

**Input:**  $A()$  1D array represent the mapped values of the sub-band coefficients (LL, LH, HL, HH) of record from  $(Y, I, Q)$ .

$W$  is the image width.

$H$  is the image height.

$Nb$  max number of bits required to encode the largest number by using fixed length binary representation.

$Nb$  the length (in bits) of the first shift codeword.

**Output:**  $NA()$  2D a set of integers

**Procedure:**

$l \leftarrow 0$

$Max \leftarrow (2^{nb}) - 1$

For all coefficients  $(x, y)$  where  $0 \leq x \leq h-1$  and  $0 \leq y \leq H-1$

If  $Abs(A(l)) < Max$  Then

$NA(x, y) \leftarrow A(l)$

$l \leftarrow l + 1$

Else

$NA(x, y) \leftarrow A(l) + A(l + 1)$

$l \leftarrow l + 2$

End If

End Loop  $x, y$

3. For each quantization index value  $Q_i(x, y, k)$  belongs to subband LL, apply the dequantization process as follows:

$$W'(x, y, k) = \frac{Q_i(x, y, k) \times R}{2^{B(k)} - 1} + M_n$$

Where  $R$  is the range between maximum and minimum computed in algorithm (3.3)  $W'(x, y, k)$  is the reconstructed wavelet coefficient.

Algorithm (3.9) represents inverse scalar quantization.

### **Algorithm(3.9) Inverse Scalar Quantization**

**Input:**  $Q()$  2D array for all component (Y, I, Q).

$W$  is the image width.

$H$  is the image height.

$bitsclr$  is the number of bit quantized in algorithm(3.3).

$min$  is the minimum number in LL-band.

$rang$  is the number of range between max number and minimum number in LL-band.

**Output:**  $img1()$  2D array of LL-band.

**Procedure:**

For all coefficients  $(x, y)$  where  $0 \leq x \leq W-1$  and  $0 \leq y \leq H-1$

$img(x, y, k) \leftarrow ((Q(x, y, k) * rang) / (2^{bitsclr} - 1)) + min$

End Loop  $x, y$

For all coefficients  $(x, y, k)$  where  $0 \leq x \leq W/2-1$  and  $0 \leq y \leq H/2-1$

$img1(x, y, k) \leftarrow img(x, y, k)$

End Loop  $x, y$

4. For each coefficient value belongs to detailed subbands, use the inverse MVQ method (see algorithm 3.10).

**Algorithm (3.10) Inverse MVQ method**

**Input:** Codebook() array of Coefficients that contain each block and index in image before compression.

*Idx()* array contain the index of each block in the codebook.

*Blksiz* is the size of block

*W* is the subband width.

*H* is the subband Height.

**Output:** *imgb()* compressing image

**Procedure:**

*By* ← 0

For *y* = 0 To *W* Step *BlkSiz*

*Bx* ← 0

For *x* = 0 To *H* Step *BlkSiz*

*k* ← *Idx(Bx, By)*

For all coefficients (*Ix, Iy*) where  $0 \leq Ix \leq W-1$  and  $0 \leq Iy \leq H-1$

*Z* ← Codebook(*k, Ix, Iy*)

*imgb(x + Ix, y + Iy)* ← *Z*

End Loop *Ix, Iy*

*Bx* ← *Bx* + 1

End Loop *y*

*By* ← *By* + 1

End Loop *x*

5. For each coefficient value  $W'_c(x, y, k)$  belongs to detailed subbands, use the following inverse mapping equation as shown in algorithm (3.11):

$$W_c(x, y, k) = \begin{cases} \frac{W'_c(x, y, k)}{2} & \text{if } W'_c(x, y, k) \text{ is even} \\ -\frac{W'_c(x, y, k) + 1}{2} & \text{if } W'_c(x, y, k) \text{ is odd} \end{cases}$$



**Algorithm (3.11) Mapping array from positive numbers to integer and real numbers**

**Input:** *img()* array of quantization indices of (LL, LH, HL, HH) Coefficients.

*W* is the image width.

*H* is the image height.

**Output:** *img()* array of integer and real numbers.

**Procedure:**

For all coefficients  $(x, y)$  where  $0 \leq x \leq W-1$  and  $0 \leq y \leq H-1$

    If  $\text{img}(x, y) \text{ Mod } 2 > 0$  then

$\text{Img}(x, y) \leftarrow 2 / \text{img}(x, y)$

    Else

        If  $\text{img}(x, y) < 0$  then

$\text{Img}(x, y) \leftarrow (1 - \text{img}(x, y)) / 2$

    End if

End Loop  $x, y$

6. Apply inverse Haar wavelet transform on the reconstructed wavelet coefficients to reconstruct the image (see algorithm (3.12)).

**Algorithm (3.12) Inverse Wavelet Transform**

**Input:** *rec()* array of record for all subbands that output from algorithm(3.10).

*no\_of\_level - 1* represent number of level of wavelet.

*W* is the image width

*H* is the image height

**Output:** *img()* record of wavelet off all sub-bands

**Procedure:**

Loop For  $i = no\_of\_level - 1$  To 0 Step -1

For all coefficients  $(x, y)$  where  $0 \leq x \leq W/2 - 1$  and  $0 \leq y \leq H/2 - 1$

$img(2 * x, 2 * y) \leftarrow 0.5 * (rec(i).ll(x, y) + rec(i).hl(x, y) + rec(i).lh(x, y) + rec(i).hh(x, y))$

$img(2 * x, 2 * y + 1) \leftarrow 0.5 * (rec(i).ll(x, y) + rec(i).hl(x, y) - rec(i).lh(x, y) - rec(i).hh(x, y))$

$img(2 * x + 1, 2 * y) \leftarrow 0.5 * ((rec(i).ll(x, y) - rec(i).hl(x, y) + rec(i).lh(x, y) - rec(i).hh(x, y))$

$img(2 * x + 1, 2 * y + 1) \leftarrow 0.5 * ((rec(i).ll(x, y) - rec(i).hl(x, y) - rec(i).lh(x, y) + rec(i).hh(x, y))$

End Loop  $x, y$

If  $(i > 0)$  Then

For all coefficients  $(x, y)$  where  $0 \leq x \leq W - 1$  and  $0 \leq y \leq H - 1$

$rec(i - 1).ll(x, y) \leftarrow img(x, y)$

End Loop  $x, y$

End If

$H \leftarrow H * 2$

$W \leftarrow W * 2$

End Loop  $i$

7. Finally, retransform the results into RGB color model as shown in algorithm (3.13).

**Algorithm (3.13) Inverse transform from YIQ model to RGB model****Input:**  $Y()$  2D array of Y model. $I()$  2D array of I model. $Q()$  2D array of Q model. $bitY, bitI, bitQ$  number of bits input in the algorithm(3.1). $minY, minI, minQ$  number of minimum number compute in the algorithm(3.1). $rangY, rangI, rangQ$  the range between maximum and minimum of components Y, I and Q that compute in the algorithm(3.1) . $W$  is the image width. $H$  is the image height.**Output:**  $R()$  2D array of R model $G()$  2D array of G model $B()$  2D array of B model**Procedure:** $oneY \leftarrow rangY / (2^{bitY} - 1)$  $oneI \leftarrow rangI / (2^{bitI} - 1)$  $oneQ \leftarrow rangQ / (2^{bitQ} - 1)$ For all coefficients  $(x, y)$  where  $0 \leq x \leq W-1$  and  $0 \leq y \leq H-1$  $Y(x, y) \leftarrow R(x, y) * oneY + minY$  $I(x, y) \leftarrow G(x, y) * oneI + minI$  $Q(x, y) \leftarrow B(x, y) * oneQ + minQ$ End Loop  $x, y$ For all coefficients  $(x, y)$  where  $0 \leq x \leq W-1$  and  $0 \leq y \leq H-1$  $R(x, y) \leftarrow Y(x, y) + 0.956 * I(x, y) + 0.621 * Q(x, y)$  $G(x, y) \leftarrow Y(x, y) - 0.272 * I(x, y) - 0.647 * Q(x, y)$  $B(x, y) \leftarrow Y(x, y) - 1.106 * I(x, y) + 1.703 * Q(x, y)$ End Loop  $x, y$

### **3.6 Experimental Results**

This section is dedicated to present the results of applying the proposed WMVQ method. The proposed method was applied separately on the luminance component and both chrominance components.

All the involved parameters of this method, have been tested as control parameters to investigate the compression performance of the proposed method.

These involved parameters are:

1. Number of layers.
2. Codebook Size.
3. Block Size.

The effect of these parameters were investigated by considering several cases within the allowable range of their values as shown in tables (3.1 - 3.28) for YIQ color models. While figures (3.10 – 3.12) present the reconstructed RGB images for Satellite, Monaliza and Lina images respectively. It is clear from the results that the compression performance decreases with the increase of block size and decreases with the codebook size. The relationships between compression ratio and PSNR at different values of codebook sizes are shown in figures, (3.13 - 3.15). These figures show how PSNR decreases while C.R increases with the decreasing in the codebook size. Better compression ratio is obtained for smaller codebook size, as less number of bits required to store the index of the codevector.

**Table (3.1):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Satellite image with block size  $2 \times 2$ , and the number of subband layers=3.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Sat.	10	scalar	X	X	X
	7, 8, 9	vector	128	5.801	22.909
	4, 5, 6	vector	64		
	1, 2	vector	32		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	64	7.613	21.085
	4, 5, 6	vector	32		
	1, 2	vector	16		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	32	10.063	19.683
	4, 5, 6	vector	16		
	1, 2	vector	8		
	3	eliminate	X	X	X

**Table (3.2):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Monaliza image with block size  $2 \times 2$ , and the number of subband layers=3.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Mona.	10	scalar	X	X	X
	7, 8, 9	vector	128	5.777	30.209
	4, 5, 6	vector	64		
	1, 2	vector	32		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	64	7.529	27.653
	4, 5, 6	vector	32		
	1, 2	vector	16		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	32	9.881	26.029
	4, 5, 6	vector	16		
	1, 2	vector	8		
	3	eliminate	X	X	X

**Table (3.3):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Baboon image with block size  $2 \times 2$ , and the number of subband layers=3.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Lina	10	scalar	X	X	X
	7, 8, 9	vector	128	0,902	27,816
	4, 5, 6	vector	64		
	1, 2	vector	32		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	64	7,706	20,767
	4, 5, 6	vector	32		
	1, 2	vector	16		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	32	10,113	23,797
	4, 5, 6	vector	16		
	1, 2	vector	8		
	3	eliminate	X	X	X

**Table (3.4):** The effects of control parameters on the compression performance of the proposed method applied on chrominance component(I) of Satellite, Monaliza and Baboon images with block size  $2 \times 2$ ,  $4 \times 4$ , and the number of subband layers=3.

Test Color Image	Subband Number	Type of Quantization	Block Size	Codebook Size	C.R.	PSNR. (dB)
Sat.	10	scalar	X	X	X	X
	4,5,6,7,8,9	vector	$2 \times 2$	4	36.571	50.103
	1	vector	$4 \times 4$	4		
	2, 3	eliminate	X	X	X	X
Mona.	10	scalar	X	X	X	X
	4,5,6,7,8,9	vector	$2 \times 2$	4	34.133	49.301
	1	vector	$4 \times 4$	4		
	2, 3	eliminate	X	X	X	X
Lina	10	scalar	X	X	X	X
	4,5,6,7,8,9	vector	$2 \times 2$	4	23,133	01,601
	1	vector	$4 \times 4$	4		
	2, 3	eliminate	X	X	X	X

**Table (3.5):** The effects of control parameters on the compression performance of the proposed method applied on chrominance component(Q) of Satellite, Monaliza and Baboon images with block size 2×2, 4×4, and the number of subband layers=3.

Test Color Image	Subband Number	Type of Quantization	Block Size	Codebook Size	C.R.	PSNR. (dB)
<b>Sat.</b>	10	scalar	X	X	X	X
	4,5,6,7,8,9	vector	2×2	4	47.080	54.721
	1	vector	4×4	4		
	2, 3	eliminate	X	X	X	X
<b>Mona.</b>	10	scalar	X	X	X	X
	4,5,6,7,8,9	vector	2×2	4	47.080	55.261
	1	vector	4×4	4		
	2, 3	eliminate	X	X	X	X
<b>Lina</b>	10	scalar	X	X	X	X
	4,5,6,7,8,9	vector	2×2	4	46.945	56.052
	1	vector	4×4	4		
	2, 3	eliminate	X	X	X	X

**Table (3.6):** The compression performance of the proposed method of Satellite, Monaliza, and Baboon images with block size 2×2 of Y-component and number of layers=3.

Test Color Image	Subband Number	Type of Quantization	Codebook Size of Y-component	C.R.	PSNR. (dB)
Sat.	10	scalar	X	X	X
	7, 8, 9	vector	128	29.817	22.402
	4, 5, 6	vector	64		
	1, 2	vector	32		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	64	30.421	20.747
	4, 5, 6	vector	32		
	1, 2	vector	16		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	32	31.238	19.436
	4, 5, 6	vector	16		
	1, 2	vector	8		
	3	eliminate	X	X	X
Mona.	10	scalar	X	X	X
	7, 8, 9	vector	128	28.996	30.039
	4, 5, 6	vector	64		
	1, 2	vector	32		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	64	29.581	28.315
	4, 5, 6	vector	32		
	1, 2	vector	16		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	32	30.365	27.042
	4, 5, 6	vector	16		
	1, 2	vector	8		
	3	eliminate	X	X	X
Lina	10	scalar	X	X	X
	7, 8, 9	vector	128	28.993	28.262
	4, 5, 6	vector	64		
	1, 2	vector	32		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	64	29.595	26.423
	4, 5, 6	vector	32		
	1, 2	vector	16		
	3	eliminate	X	X	X
	10	scalar	X	X	X
	7, 8, 9	vector	32	30.397	24.608
	4, 5, 6	vector	16		
	1, 2	vector	8		
	3	eliminate	X	X	X



**Table (3.7):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Satellite image with block size  $4 \times 4$ , and the number of subband layers=3.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Sat.	10	Scalar	X	X	X
	7, 8, 9	Vector	32	14.813	17.675
	4, 5, 6	Vector	16		
	1, 2	Vector	8		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	16	23.011	16.874
	4, 5, 6	Vector	8		
	1, 2	Vector	4		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	8	34.276	16.385
	4, 5, 6	Vector	4		
	1, 2	Vector	2		
	3	Eliminate	X	X	X

**Table (3.8):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Monaliza image with block size  $4 \times 4$ , and the number of subband layers=3.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Mona.	10	Scalar	X	X	X
	7, 8, 9	Vector	32	14.707	25.402
	4, 5, 6	Vector	16		
	1, 2	Vector	8		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	16	22.021	23.835
	4, 5, 6	Vector	8		
	1, 2	Vector	4		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	8	32.125	22.748
	4, 5, 6	Vector	4		
	1, 2	Vector	2		
	3	Eliminate	X	X	X

**Table (3.9):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Baboon image with block size  $4 \times 4$ , and the number of subband layers=3.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Lina	10	Scalar	X	X	X
	7, 8, 9	Vector	32	15.723	22.721
	4, 5, 6	Vector	16		
	1, 2	Vector	8		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	16	24.165	21.632
	4, 5, 6	Vector	8		
	1, 2	Vector	4		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	8	35.083	20.919
	4, 5, 6	Vector	4		
	1, 2	Vector	2		
3	Eliminate	X	X	X	

**Table (3.10):** The effects of control parameters on the compression performance of the proposed method applied on chrominance component(I) of Satellite, Monaliza and Baboon images with block size  $4 \times 4$ ,  $8 \times 8$ , and the number of sub-band layers=3.

Test Color Image	Subband Number	Type of Quantization	Block Size	Codebook Size	C.R.	PSNR. (dB)
Sat.	10	Scalar	X	X	X	X
	4,5,6,7,8,9	Vector	$4 \times 4$	4	60.235	49.431
	1	Vector	$8 \times 8$	8		
	2, 3	Eliminate	X	X	X	X
Mona.	10	Scalar	X	X	X	X
	4,5,6,7,8,9	Vector	$4 \times 4$	4	52.851	48.823
	1	Vector	$8 \times 8$	8		
	2, 3	Eliminate	X	X	X	X
Lina	10	Scalar	X	X	X	X
	4,5,6,7,8,9	Vector	$4 \times 4$	4	54.251	51.067
	1	Vector	$8 \times 8$	8		
	2, 3	Eliminate	X	X	X	X

**Table (3.11):** The effects of control parameters on the compression performance of the proposed method applied on chrominance component(Q) of Satellite, Monaliza and Baboon images with block size 4×4, 8×8, and the number of subband layers=3.

Test Color Image	Subband Number	Type of Quantization	Block Size	Codebook Size	C.R.	PSNR. (dB)
<b>Sat.</b>	10	Scalar	X	X	X	X
	4,5,6,7,8,9	Vector	4×4	4	117.028	54.173
	1	Vector	8×8	8		
	2, 3	Eliminate	X	X	X	X
<b>Mono.</b>	10	Scalar	X	X	X	X
	4,5,6,7,8,9	Vector	4×4	4	99.902	54.874
	1	Vector	8×8	8		
	2, 3	Eliminate	X	X	X	X
<b>Lina</b>	10	Scalar	X	X	X	X
	4,5,6,7,8,9	Vector	4×4	4	102.4	55.045
	1	Vector	8×8	8		
	2, 3	Eliminate	X	X	X	X

**Table (3.12):** The compression performance of the proposed method of Satellite, Monaliza, and Baboon images with block size 4×4 of Y-component and number of layers=3.

Test Color Image	Subband Number	Type of Quantization	Codebook Size of Y-component	C.R.	PSNR. (dB)
Sat.	10	Scalar	X	X	X
	7, 8, 9	Vector	32	64.025	17.500
	4, 5, 6	Vector	16		
	1, 2	Vector	8		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	16	66.758	16.728
	4, 5, 6	Vector	8		
	1, 2	Vector	4		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	8	70.513	16.254
	4, 5, 6	Vector	4		
	1, 2	Vector	2		
	3	Eliminate	X	X	X
Mona.	10	Scalar	X	X	X
	7, 8, 9	Vector	32	55.820	26.480
	4, 5, 6	Vector	16		
	1, 2	Vector	8		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	16	58.258	25.159
	4, 5, 6	Vector	8		
	1, 2	Vector	4		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	8	61.626	24.191
	4, 5, 6	Vector	4		
	1, 2	Vector	2		
	3	Eliminate	X	X	X
Lina	10	Scalar	X	X	X
	7, 8, 9	Vector	32	57.458	23.564
	4, 5, 6	Vector	16		
	1, 2	Vector	8		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	16	60.272	22.531
	4, 5, 6	Vector	8		
	1, 2	Vector	4		
	3	Eliminate	X	X	X
	10	Scalar	X	X	X
	7, 8, 9	Vector	8	63.911	21.848
	4, 5, 6	Vector	4		
	1, 2	Vector	2		
	3	Eliminate	X	X	X

**Table (3.13):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Satellite image with block size  $2 \times 2$ , and the number of subband layers=2.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Sat.	7	scalar	X	X	X
	4, 5, 6	vector	128	3.792	24.581
	1, 2, 3	vector	64		
	7	scalar	X		
	4, 5, 6	vector	64	4.530	22.987
	1, 2, 3	vector	32		
	7	scalar	X		
	4, 5, 6	vector	32	5.417	21.419
	1, 2, 3	vector	16		

**Table (3.14):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Monaliza image with block size  $2 \times 2$ , and the number of subband layers=2.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Mona.	7	scalar	X	X	X
	4, 5, 6	vector	128	3.820	33.128
	1, 2, 3	vector	64		
	7	scalar	X		
	4, 5, 6	vector	64	4.556	30.848
	1, 2, 3	vector	32		
	7	scalar	X		
	4, 5, 6	vector	32	5.446	29.213
	1, 2, 3	vector	16		

**Table (3.15):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Baboon image with block size  $2 \times 2$ , and the number of subband layers=2.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Lina	7	Scalar	X	X	X
	4, 5, 6	Vector	128	4.633	30.175
	1, 2, 3	Vector	64		
	7	Scalar	X		
	4, 5, 6	Vector	64	5.535	28.396
	1, 2, 3	Vector	32		
	7	Scalar	X		
	4, 5, 6	Vector	32	6.548	26.984
	1, 2, 3	Vector	16		

**Table (3.16):** The effects of control parameters on the compression performance of the proposed method applied on chrominance component(I) of Satellite, Monaliza and Baboon images with block size 2×2, 4×4, and the number of subband layers=2.

Test Color Image	Subband Number	Type of Quantization	Block Size	Codebook Size	C.R.	PSNR. (dB)
Sat.	7	Scalar	X	X	X	X
	4, 5, 6	Vector	2×2	4	22.803	51.103
	1, 2	Vector	4×4	4		
	3	Eliminate	X	X	X	X
Mona.	7	Scalar	X	X	X	X
	4, 5, 6	Vector	2×2	4	19.309	52.871
	1, 2	Vector	4×4	4		
	3	Eliminate	X	X	X	X
Lina	7	Scalar	X	X	X	X
	4, 5, 6	Vector	2×2	4	19.309	55.570
	1, 2	Vector	4×4	4		
	3	Eliminate	X	X	X	X

**Table (3.17):** The effects of control parameters on the compression performance of the proposed method applied on chrominance component(Q) of Satellite, Monaliza and Baboon images with block size 2×2, 4×4, and the number of subband layers=2.

Test Color Image	Subband Number	Type of Quantization	Block Size	Codebook Size	C.R.	PSNR. (dB)
Sat.	7	Scalar	X	X	X	X
	4, 5, 6	Vector	2×2	4	35.694	55.866
	1, 2	Vector	4×4	4		
	3	Eliminate	X	X	X	X
Mono.	7	Scalar	X	X	X	X
	4, 5, 6	Vector	2×2	4	35.386	57.545
	1, 2	Vector	4×4	4		
	3	Eliminate	X	X	X	X
Lina	7	Scalar	X	X	X	X
	4, 5, 6	Vector	2×2	4	35.386	58.981
	1, 2	Vector	4×4	4		
	3	Eliminate	X	X	X	X

**Table (3.18):** The compression performance of the proposed method of Satellite, Monaliza and Baboon images with block size  $2 \times 2$  of Y-component and number of layers=2.

Test Color Image	Subband Number	Type of Quantization	Codebook Size of Y-component	C.R.	PSNR. (dB)
<b>Sat.</b>	7	Scalar	X	20.763	24.074
	4, 5, 6	Vector	128		
	1, 2, 3	Vector	64		
	7	Scalar	X	21.009	22.632
	4, 5, 6	Vector	64		
	1, 2, 3	Vector	32		
	7	Scalar	X	21.305	21.168
	4, 5, 6	Vector	32		
	1, 2, 3	Vector	16		
<b>Mono.</b>	7	Scalar	X	19.505	32.892
	4, 5, 6	Vector	128		
	1, 2, 3	Vector	64		
	7	Scalar	X	19.750	31.376
	4, 5, 6	Vector	64		
	1, 2, 3	Vector	32		
	7	Scalar	X	20.047	30.133
	4, 5, 6	Vector	32		
	1, 2, 3	Vector	16		
<b>Lina</b>	7	Scalar	X	19.776	30.722
	4, 5, 6	Vector	128		
	1, 2, 3	Vector	64		
	7	Scalar	X	20.077	29.119
	4, 5, 6	Vector	64		
	1, 2, 3	Vector	32		
	7	Scalar	X	20.414	27.808
	4, 5, 6	Vector	32		
	1, 2, 3	Vector	16		

**Table (3.19):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Satellite image with block size  $4 \times 4$ , and the number of subband layers=2.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Sat.	7	Scalar	X	X	X
	4, 5, 6	Vector	128	4.686	20.949
	1, 2, 3	Vector	64		
	7	Scalar	X		
	4, 5, 6	Vector	64	6.736	19.385
	1, 2, 3	Vector	32		
	7	Scalar	X		
	4, 5, 6	Vector	32	8.943	18.490
	1, 2, 3	Vector	16		

**Table (3.20):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Monaliza image with block size  $4 \times 4$ , and the number of subband layers=2.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Mona.	7	Scalar	X	X	X
	4, 5, 6	Vector	128	5.007	30.633
	1, 2, 3	Vector	64		
	7	Scalar	X		
	4, 5, 6	Vector	64	7.013	28.628
	1, 2, 3	Vector	32		
	7	Scalar	X		
	4, 5, 6	Vector	32	9.102	27.432
	1, 2, 3	Vector	16		

**Table (3.21):** The effects of control parameters on the compression performance of the proposed method applied on luminance component of Baboon image with block size  $4 \times 4$ , and the number of subband layers=2.

Test Color Image	Subband Number	Type of Quantization	Codebook Size	C.R.	PSNR. (dB)
Lina	7	Scalar	X	X	X
	4, 5, 6	Vector	128	5.158	26.916
	1, 2, 3	Vector	64		
	7	Scalar	X		
	4, 5, 6	Vector	64	7.474	25.271
	1, 2, 3	Vector	32		
	7	Scalar	X		
	4, 5, 6	Vector	32	9.846	24.217
	1, 2, 3	Vector	16		



**Table (3.22):** The effects of control parameters on the compression performance of the proposed method applied on chrominance component(I) of Satellite, Monaleza and Baboon images with block size 4×4, 8×8, and the number of subband layers=2.

Test Color Image	Subband Number	Type of Quantization	Block Size	Codebook Size	C.R.	PSNR. (dB)
Sat.	7	Scalar	X	X	30.796	50.531
	4, 5, 6	Vector	4×4	4		
	1, 2	Vector	8×8	4	X	X
	3	Eliminate	X	X		
Mono.	7	Scalar	X	X	24.899	52.149
	4, 5, 6	Vector	4×4	4		
	1, 2	Vector	8×8	4	X	X
	3	Eliminate	X	X		
Lina	7	Scalar	X	X	24.899	54.725
	4, 5, 6	Vector	4×4	4		
	1, 2	Vector	8×8	4	X	X
	3	Eliminate	X	X		

**Table (3.23):** The effects of control parameters on the compression performance of the proposed method applied on chrominance component(Q) of Satellite, Monaleza and Baboon images with block size 4×4, 8×8, and the number of subband layers=2.

Test Color Image	Subband Number	Type of Quantization	Block Size	Codebook Size	C.R.	PSNR. (dB)
Sat.	7	Scalar	X	X	65.015	55.492
	4, 5, 6	Vector	4×4	4		
	1, 2	Vector	8×8	4	X	X
	3	Eliminate	X	X		
Mono.	7	Scalar	X	X	65.015	57.007
	4, 5, 6	Vector	4×4	4		
	1, 2	Vector	8×8	4	X	X
	3	Eliminate	X	X		
Lina	7	Scalar	X	X	61.134	58.092
	4, 5, 6	Vector	4×4	4		
	1, 2	Vector	8×8	4	X	X
	3	Eliminate	X	X		

**Table (3.24):** The compression performance of the proposed method of Satellite, Monaliza and Baboon images with block size 4×4 of Y-component and number of layers=2.

Test Color Image	Subband Number	Type of Quantization	Codebook Size of Y-component	C.R.	PSNR. (dB)		
<b>Sat.</b>	7	scalar	X	X	X		
	4, 5, 6	vector	128				
	1, 2, 3	vector	64				
	33.499	20.705	7	scalar	X	X	
			4, 5, 6	vector	64	34.183	19.217
			1, 2, 3	vector	32		
	34.918	18.350	7	scalar	X	X	
			4, 5, 6	vector	32	34.918	18.350
			1, 2, 3	vector	16		
<b>Mona.</b>	7	scalar	X	X	X		
	4, 5, 6	vector	128				
	1, 2, 3	vector	64				
	31.640	31.120	7	scalar	X	X	
			4, 5, 6	vector	64	32.309	29.598
			1, 2, 3	vector	32		
	33.005	28.619	7	scalar	X	X	
			4, 5, 6	vector	32	33.005	28.619
			1, 2, 3	vector	16		
<b>Lina</b>	7	scalar	X	X	X		
	4, 5, 6	vector	128				
	1, 2, 3	vector	64				
	30.397	27.721	7	scalar	X	X	
			4, 5, 6	vector	64	31.169	26.168
			1, 2, 3	vector	32		
	31.960	25.155	7	scalar	X	X	
			4, 5, 6	vector	32	31.960	25.155
			1, 2, 3	vector	16		



Number of layer = 3  
 Codebook size of Y-component = 128, 64, 32.  
 Block size of Y-component = 2×2.  
 C.R. = 29.817.  
 PSNR. = 22.402.



Number of layer = 2  
 Codebook size of Y-component = 128, 64.  
 Block size of Y-component = 2×2.  
 C.R. = 20.763.  
 PSNR. = 24.074.

**Fig. (3.10)** The reconstructed RGB Satellite images from applying WMVQ.



Number of layer = 3  
 Codebook size of Y-component = 128, 64, 32.  
 Block size of Y-component = 2×2.  
 C.R. = 28.996.  
 PSNR. = 30.039.



Number of layer = 2  
 Codebook size of Y-component = 128, 64.  
 Block size of Y-component = 2×2.  
 C.R. = 19.505.  
 PSNR. = 32.892.

**Fig. (3.11)** The reconstructed RGB Monaliza images from applying WMVQ.



Number of layer = 3  
 Codebook size of Y-component =128, 64, 32.  
 Block size of Y-component =2x2.  
 C.R. = 28.993.  
 PSNR. = 28.262.



Number of layer = 2  
 Codebook size of Y-component =128, 64.  
 Block size of Y-component =2x2.  
 C.R. = 19.776.  
 PSNR. = 30.722.

Fig. (3.12) The reconstructed RGB Lina images from applying WMVQ.

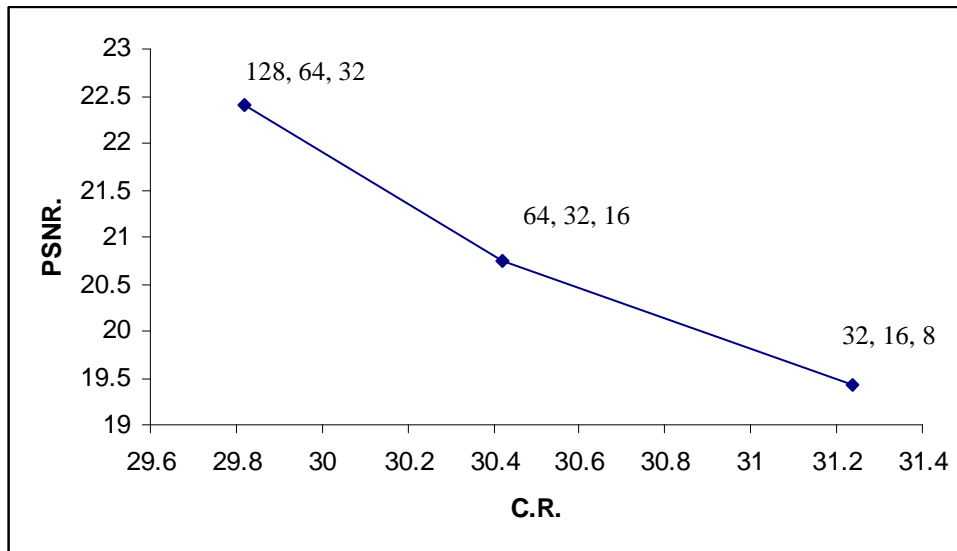
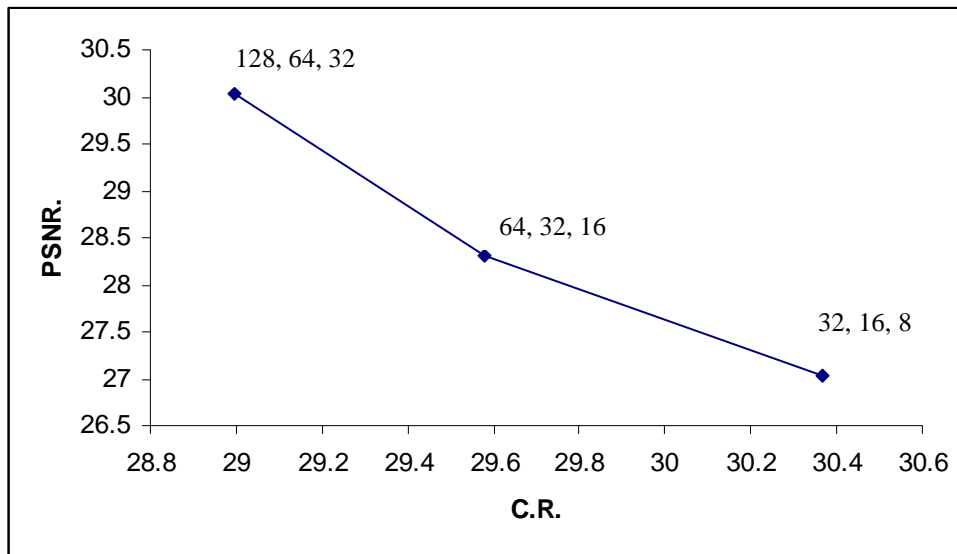
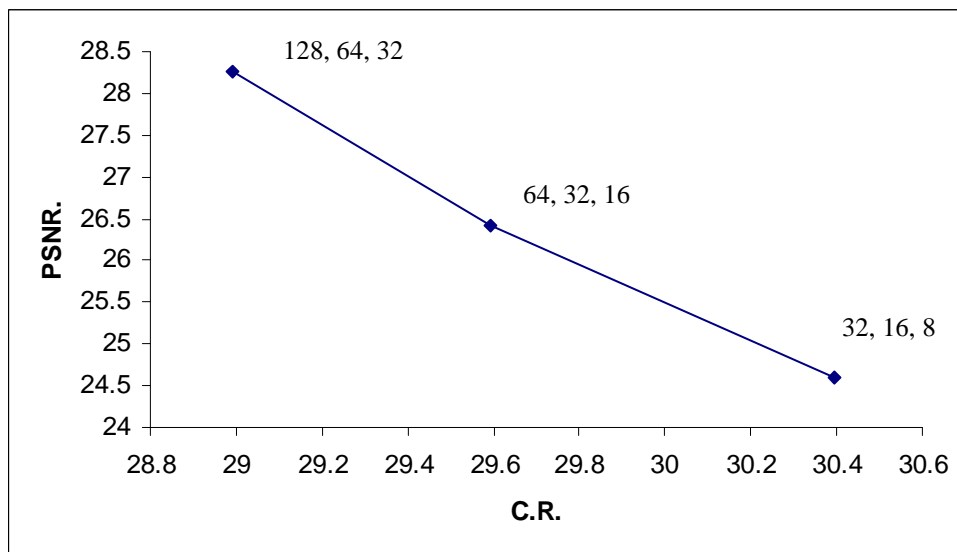


Fig. (3.13) Compression ratio versus PSNR of suggested method with number of layers=3, for different codebook sizes for Satellite image.



**Fig. (3.14)** Compression ratio versus PSNR of suggested method with number of layers=3, for different codebook sizes for Monaliza image.



**Fig. (3.15)** Compression ratio versus PSNR of suggested method with number of layers=3, for different codebook sizes for Lina image.

# Chapter 4

## Conclusions & Suggestions

### 4.1 Conclusions

Wavelet transform and modified VQ was considered. In this work, the colored images were often processed by first performing a color transformation from RGB to YIQ color model performing the compression process, and then transform back to RGB model. From the results presented in the previous chapters, some remarks related to the performance of the proposed methods were reported. Some of the important conclusions are presented in the following:

1. The proposed method offers a compression performance up to (29/1) with little effects will be noticed on the image quality (22/1).
2. The experimental results show that the three-level decomposition scheme (i.e., 10 subbands) is a good practical choice for the luminance and chrominance components.
3. The codebook size refers to the total number of code vectors in the codebook. As the size of codebook increases, the quality of the reconstructed images improves, but the compression ratio reduces and the computational complexity increases. Therefore, there is a trade off between the quality of the reconstructed images and the amount of compression achieved.
4. The compression performance of the proposed scheme is effected by the textural attributes of the image, such that the image regions which imply high contrast regions (as in urban regions in the Satellite image), will suffer as objective degradation, when they compressed up to high compression ratio.

5. Very small size of codebooks are utilized for both chrominance components comparing with codebook size for luminance component because most of the image energy is distributed in luminance component and to maintain a relatively high fidelity coding of luminance component to satisfy the human visual system.

## **4.2 Future work**

Among the different suggestions that could be simulated during our previous discussion, the following suggestions were presented as a material for future research work;

1. The performance of the wavelet transform can be increased by using better filters (e.g., the Daubechies, Tab9\7). Using better filters perhaps could yield additional improvement in the compression performance.
2. Study in more details, the performance of the considering coding scheme with CIELAB, YC<sub>b</sub>C<sub>r</sub>, YUV and compare between them.
3. Coding wavelet coefficients using fast algorithm for VQ-based wavelet coding system and implementing Partial Search Partial Distortion (PSPD). In this modification, the combination of variance  $\alpha^2$  and mean  $m$  is used to reject a large number of code vectors from the search consideration without calculating their distortion distance from the training vectors. Then, the partial search is used to find the closed or best match code vectors from the remaining possible match of the codebook.
4. Using method of coding super high definition (SHD) still images based on vector quantization of wavelet coefficients will use. A compression scheme for SHD images should achieve data compression without ant visible picture quality deterioration.

# Contents

<b>Chapter 1 General Introduction</b>	<b>1-7</b>
1.1 Introduction	1
1.2 Image Size Problem	2
1.3 Image Compression	3
1.4 Vector Quantization	3
1.5 An Introduction to Wavelets	4
1.6 Related Work	5
1.7 Aim of Thesis	6
1.8 Thesis Layout	7
<b>Chapter 2 Color Models &amp; Image Compression</b>	<b>8-45</b>
2.1 Introduction	8
2.2 Basic Concepts	8
2.2.1 What is Color?	9
2.2.2 Luminance	9
2.2.3 Chrominance	9
2.3 Types of Digital Images	10
2.3.1 Black and White Images	10
2.3.2 Color Images	10
2.3.3 Binary or Bilevel Images	11
2.3.4 Indexed Color Images	11
2.4 Computer Color Image Representation	11
2.5 Color Models	14
2.5.1 RGB Color Model	14
2.5.2 CIEXYZ Color Model	15
2.5.3 CMYK Color Model	16
2.5.4 YUV Color Model	17
2.5.5 YIQ Color Model	18
2.5.6 $YCbCr$ Color Model	18
2.5.7 IHS Color Model	19
2.5.8 CIELAB Color Model	21
2.5.9 CIELUV Color Model	23
2.6 Color Image Compression	24
2.6.1 Lossless coding	25
2.6.2 Lossy coding	26
2.7 Vector Quantization	27
2.7.1 The LBG Design Algorithm	29



<b>2.8</b>	Scalar Quantization .....	30
<b>2.9</b>	Transformation Methods .....	31
<b>2.9.1</b>	Wavelets Transform .....	31
<b>2.9.1.1</b>	Discrete Wavelet Decomposition .....	32
<b>2.9.1.2</b>	Wavelet Image Decompositions .....	33
<b>2.9.1.3</b>	Haar Wavelet Transform (HWT) .....	39
<b>2.9.1.4</b>	Integer Wavelet Transform(IWT) .....	41
<b>2.9.2</b>	Wavelet Transform Charactersitics .....	41
<b>2.10</b>	Compression Efficiency Parameters .....	42
<b>2.10.1</b>	Compression Ratio .....	42
<b>2.10.2</b>	Fidelity Criteria .....	42
<b>2.11</b>	Programming Work .....	45

### **Chapter 3 Hybrid Wavelet Modified Vector Quantization**

	<b>(WMVQ) .....</b>	<b>46-91</b>
<b>3.1</b>	Introduction .....	46
<b>3.2</b>	Color Images Transformation .....	46
<b>3.3</b>	Suggest Wavelet Transform .....	47
<b>3.3.1</b>	Forward Haar Wavelet Transform .....	47
<b>3.3.2</b>	Inverse Haar Wavelet Transform .....	53
<b>3.4</b>	Modified Vector Quantization (MVQ) .....	54
<b>3.5</b>	Wavelet Modified Vector Quantization method (WMVQ) .....	54
<b>3.5.1</b>	Encoding Unit .....	55
<b>3.5.2</b>	Decoding Unit .....	67
<b>3.6</b>	Experimental Results .....	74

### **Chapter 4 Conclusions & Suggestions .....**

<b>4.1</b>	Conclusions .....	92
<b>4.2</b>	Future work .....	93

<b>References</b> .....	94-98
-------------------------	-------

# Tables

## Chapter 3

<b>3.1</b> The effects of control parameters on the compression performance of the proposed method applied on luminance component of Satellit image with block size $2 \times 2$ , and the number of subband layers=3. ....	75
<b>3.2</b> The effects of control parameters on the compression performance of the proposed method applied on luminance component of Monaliza image with block size $2 \times 2$ , and the number of subband layers=3. ....	75
<b>3.3</b> The effects of control parameters on the compression performance of the proposed method applied on luminance component of Baboon image with block size $2 \times 2$ , and the number of subband layers=3. ....	76
<b>3.4</b> The effects of control parameters on the compression performance of the proposed method applied on chrominance component(I) of Satellite, Monaliza and Baboon images with block size $2 \times 2$ , $4 \times 4$ , and the number of subband layers=3 ....	76
<b>3.5</b> The effects of control parameters on the compression performance of the proposed method applied on chrominance component(Q) of Satellite, Monaliza and Baboon images with block size $2 \times 2$ , $4 \times 4$ , and the number of subband layers=3. ....	77
<b>3.6</b> The compression performance of the proposed method of Satellite, Monaliza, and Baboon images with block size $2 \times 2$ of Y-component and number of layers=3 ....	78
<b>3.7</b> The effects of control parameters on the compression performance of the proposed method applied on luminance component of Satellite image with block size $4 \times 4$ , and the number of subband layers=3 ....	79
<b>3.8</b> The effects of control parameters on the compression performance of the proposed method applied on luminance component of Monaliza image with block size $4 \times 4$ , and the number of subband layers=3. ....	79
<b>3.9</b> The effects of control parameters on the compression performance of the proposed method applied on luminance component of Baboon image with block size $4 \times 4$ , and the number of subband layers=3. ....	80
<b>3.10</b> he effects of control parameters on the compression performance of the proposed method applied on chrominance component(I) of Satellite, Monaliza and Baboon images with block size $4 \times 4$ , $8 \times 8$ , and the number of sub-band layers=3. ....	80
<b>3.11</b> The effects of control parameters on the compression performance of the proposed method applied on chrominance component(Q) of Satellite, Monaliza and Baboon images with block size $4 \times 4$ , $8 \times 8$ , and the number of subband layers=3. ....	81

<b>3.12</b>	The compression performance of the proposed method of Satellite, Monaliza, and Baboon images with block size 4×4 of Y-component and number of layers=3 .....	82
<b>3.13</b>	The effects of control parameters on the compression performance of the proposed method applied on luminance component of Satellite image with block size 2×2, and the number of subband layers=2 .....	83
<b>3.14</b>	The effects of control parameters on the compression performance of the proposed method applied on luminance component of Monaliza image with block size 2×2, and the number of subband layers=2 .....	83
<b>3.15</b>	The effects of control parameters on the compression performance of the proposed method applied on luminance component of Baboon image with block size 2×2, and the number of subband layers=2 .....	83
<b>3.16</b>	The effects of control parameters on the compression performance of the proposed method applied on chrominance component(I) of Satellite, Monaliza and Baboon images with block size 2×2, 4×4, and the number of subband layers=2 .....	84
<b>3.17</b>	The effects of control parameters on the compression performance of the proposed method applied on chrominance component(Q) of Satellite, Monaliza and Baboon images with block size 2×2, 4×4, and the number of subband layers=2. ....	84
<b>3.18</b>	The compression performance of the proposed method of Satellite, Monaliza and Baboon images with block size 2×2 of Y-component and number of layers=2. ....	85
<b>3.19</b>	The effects of control parameters on the compression performance of the proposed method applied on luminance component of Satellite image with block size 4×4, and the number of subband layers=2 .....	86
<b>3.20</b>	The effects of control parameters on the compression performance of the proposed method applied on luminance component of Monaliza image with block size 4×4, and the number of subband layers=2. ....	86
<b>3.21</b>	The effects of control parameters on the compression performance of the proposed method applied on luminance component of Baboon image with block size 4×4, and the number of subband layers=2. ....	86
<b>3.22</b>	The effects of control parameters on the compression performance of the proposed method applied on chrominance component(I) of Satellite, Monaliza and Baboon images with block size 4×4, 8×8, and the number of subband layers=2. ....	87
<b>3.23</b>	The effects of control parameters on the compression performance of the proposed method applied on chrominance component(Q) of Satellite, Monaliza and Baboon images with block size 4×4, 8×8, and the number of subband layers=2. ....	87

3.24 The compression performance of the proposed method of Satellite, Monaliza and Baboon images with block size 4×4 of Y-component and number of layers=2. ....	88
------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

## List of figures

### Chapter 2

2.1 One bit black and white image display. ....	12
2.2 8-bit color image display. ....	12
2.3 24 –bit “True color image” –display. ....	13
2.4 Composite RGB images from three layers. ....	13
2.5 The BMP file format of 24-bits image. ....	14
2.6 RGB color cube. The gray scale spectrum lies on the line joining the black and white vertices. ....	15
2.7 The CMYK Color Cube. ....	17
2.8 HIS hexagonal cone. ....	21
2.9 A three dimensional representation of the CIELAB color mode. ....	23
2.10 The Most Popular Image Compression Methods. ....	27
3.11 The encoder and decoder in a vector quantization. ....	28
2.12 One level wavelet decomposition. ....	32
2.13 Line Wavelet Decomposition. ....	34
2.14 Quincunx wavelet decomposition. ....	35
2.15 Pyramid Wavelet Decomposition. ....	36
2.16 Standard Wavelet Decomposition. ....	37
2.17 Full Wavelet Decomposition (Wavelet packet of an image for one, two decomposition levels). ....	38

### Chapter 3

3.1 The forward Haar wavelet transform. ....	48
3.2 The Test color images. ....	49
3.3 Forward Haar wavelet transform applied on luminance component of satellite image. ....	50
3.4 Forward Haar wavelet transform applied on luminance component of Monaliza image. ....	51
3.5 Forward Haar wavelet transform applied on luminance component of Baboon image. ....	52
3.6 The inverse Haar wavelet transform. ....	53
3.7 Flowchart of encoder of suggested Method ....	55

<b>3.8</b>	An example illustrates the relationship between subband number (k) and the subband layer number (I), where the total number of subband layers is taken (L) .....	58
<b>3.9</b>	Flowchart of decoder of suggested method .....	67
<b>3.10</b>	The reconstructed RGB Satellite images from applying WMVQ. ....	89
<b>3.11</b>	The reconstructed RGB Monaliza images from applying WMVQ. ....	89
<b>3.12</b>	The reconstructed RGB Lina images from applying WMVQ. ....	90
<b>3.13</b>	Compression ratio versus PSNR of WMVQ method with number of layers=3, for different codebook sizes for Satellite image .....	90
<b>3.14</b>	Compression ratio versus PSNR of WMVQ method with number of layers=3, for different codebook sizes for Monaliza image .....	91
<b>3.15</b>	Compression ratio versus PSNR of WMVQ method with number of layers=2, for different codebook sizes for Lina image .....	91

DEDICATED TO

**MY PARENTS ...**

**MY BROTHERS ...**

(Wa'al and Mohammad)

*Zainab*

## *Acknowledgment*

Praise is to our almighty Allah most gracious for enabling me to finish what I started and for helping me to present this work.

I would like to express my sincere appreciation to my supervisor, **Dr. Bushra K. Al-Abudi**, for giving me the major steps to go on to explore the subject, and discuss the points that I felt they are important.

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

Also, I wish to thank the staff of Computer Science Department at **Al-Nahrain University** for their help.

I would like to say "thank you" to my faithful friends for supporting and giving me advises.

*Zainab*

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ  
(وَيَسْأَلُونَكَ عَنِ الرُّوحِ  
قُلِ الرُّوحُ مِنْ أَمْرِ رَبِّي  
وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ  
إِلَّا قَلِيلًا)

صدق الله العظيم  
سورة الإسراء (الآية ٨٥)



## المعلومات الشخصية

الاسم الثلاثي: زينب فتوح أحمد سعيد الصوفي

عنوان السكن: بغداد | حي تونس | محله ٣٢٦ | زقاق ٣٠ | دار ٥٤

رقم الهاتف: موبايل : ٠٠٩٦٤٧٩٠١٥٠٨٤٣١

العنوان البريدي : [zozo83\\_2005@yahoo.com](mailto:zozo83_2005@yahoo.com)

تاريخ المناقشة : 19\11\٢٠٠٧

أسم الرسالة:

Color Image Compression Using Hybrid Wavelet

Transform and Vector Quantization Method

أسم المشرف: الأستاذ المساعد - الدكتورة بشرى قاسم العبودي

# References

- [Add00] J. Addel, "*Lossy and Lossless Image Compression*", Prentice Hall PTR, 2000.
- [Ala02] B. k. Al-Abudi, "*Color Image Data Compression Using Multilevel Block Truncation Coding Technique*", Ph.D. thesis, department, Collage of Science, University of Baghdad, 2002.
- [Alh01] M. D. T. Al-Hassani, "*Design Finger Print Recognition System Using Wavelet Transform*", M.Sc. Thesis, department, Computer Science, Al-Nahrain University, 2001.
- [Asi03] Asiva, "*Color Spaces, Color Models and Digital Image Representation*", Texas, 2003, [www.asiva.com](http://www.asiva.com).
- [Asu02] A. Asuncion, "*Signal Processing Applications of Wavelets*", Information and Computer Science, University of California, Irvine, 2002, [http://www.ics.uci.edu/~asuncion/180/Asuncion\\_Wavelet\\_Final.pdf](http://www.ics.uci.edu/~asuncion/180/Asuncion_Wavelet_Final.pdf).
- [Bel03] A. N. Belbachir and P. M. Goebel, "*Color Image Compression: Early Vision and the Multiresolution Representations*", Vienna University of Technology, Institute of Computer Aided Automation, 2003, <http://www.prip.tuwien.ac.at/~goe/AbstractDAGM05Color.pdf>.
- [Bot98] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio and Y. LeCun, "*High Quality Document Image Compression with DjVu*", AT&T Labs, Lincroft, NJ, 1998, {fleonb,ha\_ner,pgh,patrice,yoshua,yang}@research.att.com.
- [Bur98] C. S. Burrus, R. A. Gopinath and H. Guo, "*Introduction to Wavelets and Wavelet Transform: A Primer*", Prentice Hall, Inc., 1998.

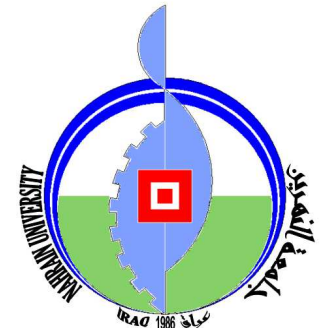
- [Chr01] G. Christensen, “*Advanced Image Processing*”, 2001, <http://www.icaen.uiowa-edu/sgautham/public-html/>.
- [Gib00] J. D. Gibson, “*Image and Video processing*”, submitted to department of electrical and computer engineering the university of Texas at Austin, 2000, <http://www.hbuk.co.uk/apl>.
- [Gol97] J. R. Goldschneider, “*Lossy Compression of Scientific Data via Wavelets and Vector Quantization*”, Doctor of Philosophy, submitted to department of electrical engineering, University of Washington, 1997, <http://citeseer.ist.psu.edu/259335.html>.
- [Gon92] R. C. Gonzalez and R. E. Woods, “*Digital Image Processing*”, Addison-Wesley, 1992.
- [Gon02] R. C. Gonzalez and R. E. Woods, “*Digital Image Processing*”, University of Tennessee, Prentice Hall, Upper Saddle River, New Jersey, 2002.
- [Gra95] A. Graps, “*An Introduction to Wavelets*”, IEEE Computational and Engineering, 1995, <http://www.amara.com/current/wavelet.html>.
- [Hei00] D. Heidelberger, “*CIELAB Color Space*”, 2000, <http://www.linecolor.com/>.
- [Jai07] N. Jain and R. Vig, “*Wavelet based vector quantization with tree code vectors for EMG Signal compression*”, proceeding of the 6<sup>th</sup> WSEAS international conference on signal processing, Dallas, Texas, USA, March 22-24-2007.
- [Jia03] X. Jianyun, A. H. Sung, P. Shi and Q. Lin, “*Text Steganography Using Wavelet Transform*”, Dep. Computer Science, New Mexico Tech, Socorro, USA, 2003.

- [Jor97] L. A. Jorj, “*New Coding Methods for Compression Remotely Sensed Images*”, Ph.D. Thesis, submitted to College of Science, University of Baghdad, 1997.
- [Kha03] O. O. Khalifa, “*Fast algorithm for VQ-based wavelet coding system*”, Electrical & Computer Department, University Malaysia, 2003, [http://sprg.massey.ac.nz/ivcnz/Proceedings/IVCNZ\\_25.pdf](http://sprg.massey.ac.nz/ivcnz/Proceedings/IVCNZ_25.pdf).
- [Lin80] Y. Linde, A. Buzo, and R. M. GRAY "An algorithm for Vector Quantizer Design ;IEEE Trans .On Commun .“, Vol .COM-28,no.1,pp.84-95, 1980.
- [Lin91] J. Lin and J. S. Vitter, “*Nearly Optimal Vector Quantization*”, Department of Computer Science, Brown University, 1991, [http://www.cs.duke.edu/~jsv/Papers/LV92.vector\\_quantization.pdf](http://www.cs.duke.edu/~jsv/Papers/LV92.vector_quantization.pdf).
- [Lus94] M. Luse, “*The BMP File Format*”, Dr.Dobb’s Journal, 1994.
- [Maj97] E. E. Majani, “*Algorithm For Lossy Progressive Losseless Image Compression*”, National Astronautics and Space Administration (NASA TECH BRIEF Vol. 21, No. 12), 1997, <http://www.rdrop.com/~cary/mirror/NPO20141.pdf>.
- [Mat87] P. M. Mather, “*Computer Processing of Remotely Sensed Images*”, John Wiley&Sons, 1987.
- [Mat99] Mathworks, “*Image Processing Toolbox*”, <http://www.mathworks.com>, 1999.
- [Muk98] D. Mukherjee and S. k. Mitra, “*Vector Set-Partitioning with Successive Refinement Voronoi Lattice VQ for Embedded Wavelet Image Coding*”, Department of Electrical and Computer Engineering, University of California, Santa Barbara, 1998, <http://www.comsoc.org/sac/private/2000/jun/pdf/18sac06-mukherjee.pdf>.

- [Mur05] F. Murtagh, “*The Haar Wavelet Transform of a Dendrogram*”, Department of Computer Science, University of London, June 26-2005, <http://www.cs.rhul.ac.uk/home/fionn/papers/fm37.pdf>.
- [Nee04] R. Neelamani, R. de Queiroz, Z. Fan, S. Dash, and R. G. Baraniuk, “*JPEG Compression History Estimation*”, Accepted by the IEEE Transactions on Image Processing, 2004, <http://signal.ece.utexas.edu/~queiroz/papers/neelsh-journal.pdf>.
- [Oli02] J. Oliver, “*Fast and Efficient Spatial Scalable Image Compression Using Wavelet Lower Trees*”, Department of Computer Engineering (DISCA), Technical University of Valencia, 2002, <http://www.grc.upv.es/docencia/tdm/referencias/VideoCoding/wavelets/LTW.pdf>.
- [Pae00] S. Paek and L. Kim, “*A Real-Time Wavelet Vector Quantization Algorithm and its VLSI Architecture*”, IEEE Transactions on circuits and systems for video technology, Department of Electrical Engineering, Korea, 2000, [http://mvlsi.kaist.ac.kr/~webmast/paper/csvt99\\_skp.pdf](http://mvlsi.kaist.ac.kr/~webmast/paper/csvt99_skp.pdf).
- [Poy97] C. A. Poynton, “*Frequently Asked Questions about Color*”, 1997, <http://www.poynton.com/PDFs/ColorFAQ.pdf>.
- [Ros82] A. Rosenfeld and A. Kak, “*Digital Picture Processing*”, 2<sup>nd</sup> Ed., Academic Press, New York, 1982.
- [Sac99] J. Sachs, “*Digital Image Basics*”, Digital Light & Color, 1999, <http://www.dl-c.com/basics.pdf>.
- [Sal00] D. Salomon, “*Data Compression*”, 2<sup>nd</sup> Ed., Springer, New York, 2000.
- [San98] S. J. Sangwine and R. E. Horne, “*The Colour Image Processing Handbook*”, Chapman & Hall, 1<sup>st</sup> Ed., 1998.

- [Shi00] Y. Q. Shi and H. Huifang, “*Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards*”, CRC.Press LLC., 1<sup>st</sup> Ed., 2000.
- [Son99] M. Sonka, V. Halva, and T. Boyle, “*Image Processing Analysis and Machine Vision*”, Brooks/Cole Publishing Company, 2<sup>nd</sup> Ed., 1999.
- [Umb98] S. E. Umbaugh, “*Computer Vision Image Processing: A practical Approach Using CVIP Tools*” Prentice Hall, INC., 1998
- [Wei96] L. Weinman, “*Color Display Primer*”, 1996, <http://www.info.med.yale.edu/caim/...>
- [Yan98] C. K .Yang and W. H. Tsai, “*Color Image Compression Using Quantization, Thresholding, and Edge Detection Techniques all Based on the Moment-preserving Principle*”, Pattern Recognition Letters, vol.19, pp.205-215, 1998

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



# *Color Image Compression Using Hybrid Wavelet Transform and Vector Quantization Method*

*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 Computer  
Science*

**By  
Zainab Fatooh Al-Soufi  
(B.Sc. 2004)**

**Supervised By  
Dr. Bushra K. Al-Abudi**

September 2007

Ramadan 1428



جمهورية العراق  
وزارة التعليم العالي والبحث العلمي  
جامعة النهرين  
كلية العلوم

# ضغط بيانات الصورة الملونه باستخدام طريقة هجين التحويل المويجي والتكميم الاتجاهي

رساله

مقدمه الى كلية العلوم في جامعة النهرين كجزء من  
متطلبات نيل درجة الماجستير في علوم الحاسبات

من قبل

زينب فتوح الصوفي

(بكالوريوس جامعة النهرين ٢٠٠٤)

إشراف

أ. م. د. بشرى قاسم العبودي

ايلول ٢٠٠٧

رمضان ١٤٢٨



desktop

[.ShellClassInfo]

LocalizedResourceName=@%SystemRoot%\system32\shell32.dll,-21815

# *Certification of the Examination Committee*

We chairman and members of the examination committee, certify that we have studied this thesis "**Color Image Compression Using Hybrid Wavelet Transform and Vector Quantization Method**" presented by the student **Zainab Fatooh Al-Soufi** and examined her in its contents and that we have found it worthy to be accepted for the degree of Master of Science in Computer Science with very good degree.

Signature:

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

Title : **Assistant Professor**

Date : / /2007

**(Chairman)**

Signature:

Name: **Dr. Bara'a Ali Attea**

Title : **Assistant Professor**

Date : / /2007

**(Member)**

Signature:

Name: **Dr. Ban N. Al-Kallak**

Title : **Lecturer**

Date : / /2007

**(Member)**

Signature:

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

Title : **Assistant Professor**

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

**(Dean of Collage of Science)**

## **Supervisor Certification**

I certify that this thesis was prepared under my supervision at the Department of Computer Science/College of Science/Al-Nahrain University, by **Zainab Fatooh Al-Soufi** as partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Signature:

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

Title : **Assist. Prof.**

Date : **26 / 9 / 2007**

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

Signature:

Name : **Dr. Taha S. Bashaga**

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

Date : **26 / 9 / 2007**