IMPROVED IMAGE COMPRESSION WITH COMPARATIVE ANALYSIS OF PROGRESSIVE CODING TECHNIQUES

By
Natasha Khan

Submitted to the Faculty of Department of Electrical Engineering

National University of Sciences and Technology, Rawalpindi in Partial fulfillment for the requirements of

MS in Electrical Engineering

March 2018

Dedicated to

To my dear Parents, Siblings and Friends

Whose love, affection and support enabled me to achieve yet another milestone in my professional life and also to my supervisor Col Dr Imran Touqir for being forthcoming and helping in fulfillment of this research work.

ABSTRACT

Images play a cardinal role in innumerable fields of life. Image compression is nothing but reduction of an image size in bytes by compromising on the quality of original image to a tolerable level. In present era, image compression is preferred by wavelet transform, which is the most in demand “time-frequency” transformation. Wavelet transforms are used due to their intrinsic property that they are redundant and shift invariant.

The wavelet transform surpassed the discrete cosine transform (DCT) and its ancestors due to its unique quality that defined the image both in frequency and time domains, while DCT defined it with sine and cosine waves. In low bit rates, artifacts are caused in DCT because it tries to distribute numerous bits to approximations furthermore, a couple of bits are assigned to fine details, whereas DWT has no artifacts.

Embedded encoding has witnessed a remarkable progress during the last two decades. With discrete wavelet transform as their basis and concatenation with entropy encodings, have paved the way towards the compression optimization. This paper is an endeavor to present a survey on the above mentioned encoding techniques by underlining their facets and drawbacks in detail. It has been discussed that as to how SPIHT has outperformed EZW not only by addressing few of its inherent shortcomings but also exhibited improvements in the basic parameters of PSNR and Compression Ratios. On the other hand EBCOT has brought improvements by preserving the edges lost by SPIHT.

ACKNOWLEDGEMENTS

In the name of Allah, the Beneficent, the Merciful. Praise and Gratitude be to Allah for giving me strength and guidance, so that this dissertation can be finished accordingly. I extend my deep gratitude to Military College of Signals, NUST, for providing an immense conducive environment during complete time of my research thesis. This careful assignment was impractical without the help of my Chief Col. Dr. Imran Tauqeer who harnessed my potentials to the optimum levels. I am obligated to him for his auspicious direction, significant consolation and developmental patronage. His warm and kind thought towards my examination helped me to continue with my undertaking in odd conditions. I m also very grateful to my committee members, Lt. Col. Dr. Adil Masood Sidiqui, and Asst. Prof. Dr. Mohammad Imran for their involvement in this research work. This work could not have been completed without untiring support and prayers of my siblings Sana, Hammad and Umair, who guided me in light of their experience of research work. They bolstered me at whatever point I needed them. I would also like to express my special thanks to my friends Irum Malik, Sehrish Zaryat and Sadaf Ahmed who always remained available to help me. This work could not have been completed without their support. My deepest gratitude goes to Maj. Asim Ashfaq, my accomplishment of this work couldn't be conceivable without his dedicated help. Last but not the least i would like to give my deepest gratitude

to my beloved husband Maj. Qaiser Lohani whose unconditional love, patience, and continual support of my academic endeavors enabled me to complete this thesis.

LIST OF ACRONYMS

1.	Peak Signal to Noise Ratio	**PSNR**
2.	Zero Tree Root	**ZTR**
3.	Isolated Zero	**IZ**
4.	Compression Ratio	**CR**
5.	Discrete Fourier Transform	**DFT**
6.	Discrete Wavelet Transform	**DWT**
7.	Dominant Pass	**D**
8.	Embedded Zero-tree Wavelet	**EZW**
9.	Mean Square Error	**MSE**
10.	Set Partitioning In Hierarchical Trees	**SPIHT**
11.	One Dimensional	**1 D**
12.	Subordinate Pass	**S**
13.	Two Dimensional	**2 D**
14.	Successive Approximation Quantization	**SAQ**
15.	Fast Fourier Transform	**FFT**
16.	Inverse Fourier Transform	**IFT**
17.	Inverse Discrete Fourier Transform	**IDWT**

18.	Positive	**Pos**
19.	Negative	**Neg**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter-1

Introduction

1.1 Background

The demand of image with a high resolution and low image size has increased in the present era, which requires more storage capacity and bandwidth Wavelet Change, a successor of discrete cosine change (DCT), is a numerical plan which is growing quickly in the field of picture compression. In this system interpreted and scaled adaptation of specific wavelets are utilized to decay a picture [1]. This is a kind of multi-resolution filter which analyzes the signal in frequency as well as in time domain simultaneously. The concept of multi-resolution of wavelet allows us to understand an image as a sum of details appearing as different resolutions [2].

The wavelet transform surpassed the discrete cosine transform (DCT) and its ancestors due to its unique quality that defined the image both in frequency and time domains, while DCT defined it with sine and cosine waves. In low bit rates, artifacts are caused in DCT because it tries to distribute numerous bits to approximations what's more, a couple of bits are assigned to fine details, whereas DWT has no artifacts [3].

Discrete Wavelet Transform (DWT) is a type of wavelet change for which wavelets are discretely examined. Now days DWT based various techniques of image compression are being used to attain better PSNR and Compression Ratio of reconstructed image. Different technique of image compression is used to attain better PSNR and compression ratio of reconstructed image i.e. EZW, SPIHT and EBCOT. In EZW a discrete wavelet transform is used, which de-correlated the coefficients to encode the more significant bits in zero trees, which predicts the insignificant across the scale. Successive-approximation to terminate the code at any point, utilizing zero-tree multiple significance maps are being coded. Entropy encoder is used by adopted arithmetic coding to code the symbols.

EZW works on an image that has been de-correlated by using DWT to encode the more insignificant bits in zero trees, which predicts the insignificance across the scale. It is having less computational complexity but better efficiency. SPIHT an improved version of EZW gives better PSNR, fast computation and high compression ratio than EZW. Using SPIHT algorithm at different bit rate produces an implanted piece stream through which best image can be reconstructed with minimum distortion rate [4]. EBCOT is also a wavelet based compression technique which gives excellent compression ratio with modest computational complexity [5]. This algorithm uses wavelet decomposition to produce sub-bands which are then quantized and coded into the block-codes. These blocks are coded independent of each other. They have an embedded bit streams which enable the efficient use of post-compression rate-distortion optimization. This allows the successful exploitation of visual image which lacked in the previous techniques [6].

1.2 Research motivation

Pictures are essential reports these days; to work with them in a few applications they should be compacted, pretty much relying upon the motivation behind the application. There are a few calculations that play out this pressure in various ways; some are lossless and keep an indistinguishable data from the first picture, some others misfortune data when packing the picture. Some of these pressure techniques are intended for particular sorts of pictures, so they won't be so useful for different sorts of pictures. A few calculations even let you change parameters they use to modify the pressure better to the picture. This exploration concentrates on comparing three different techniques on saving comparatively more number of bits without compromising the quality of the image.

1.3 Research Objectives

The main objectives of this thesis are:-

1. Study embedded codec image compression techniques based on discreet wavelet transformation (DWT).
2. Analysis of three different techniques that is EZW, SPHIT and EBCOTand its variants in detail to sift out their various facets for apt utilization.
3. In depth analysis of latest progressive coding techniques to make their better use to bring further improvements in image compression.
4. Propose a better image compression model for enhanced storage and bandwidth requirement.

1.4 The thesis Organization

The current these is organized as follows:

- Chapter 1 includes the introduction, motivation and research objects.
- Chapter 2 underlines need for image compression, its principles and processes.
- Chapter 3 describes about wavelet transform in general and DWT in particular.
- Chapter 4 highlights the EZW transform along with its advantages and shortcomings.
- Chapter 5 explains the concept as to how SPIHT has outperformed the EZW.
- Chapter 6 introduction and details about EBCOT
- Chapter 7 the results along with future work have been presented.

1.5 Image compression

This is the period of Digital communication, to store, processed and transmits digital data large amount of storage is required in so doing necessitates that adept methods be adopted or devised to meet the requirements of storage space and address the limitations of bandwidth. Image compression is the reduction of size of digital image without compromising on the quality and is achieved by minimizing the number of bytes of an image file with compromising on the quality of an image to a bare minimum level. A marked progress has been made in the field of image compression and its application in various branches of engineering. Image compression is associated with removing redundant information of image data.

1.6 Principle of Image Compression

An image is comprised of lot of pixels which are correlated with each other, which is why the neighboring pixels are very similar. Due to this correlation, only a small amount of redundant information can be get rid of because if the information from an image is removed without de-correlating it, there are chances that some of the important information is also lost as result thereby affecting the image quality. Therefore, there is a need that image be first de-correlated before subjecting it to the compression. Above in view, the digital images are first converted into statistically uncorrelated dataset before transmission and storage. Original or approximated image can be regenerated after the decompression process. Image compression deals with the redundancy. It eradicates or reduces the redundant and irrelevant data which is duplicated in the image there by preserving the image quality to a level which is acceptable by human eye. Two of the commonly known types of redundancies are Statistical Redundancy and Psycho Visual Redundancy:-

1.6.1 Statistical Redundancy

It is further classified into two types:-

1.6.1.1 Inter-Pixel Redundancy

Regular pictures have high level of connection among its pixels. This relationship is alluded as between pixel repetition or spatial excess and is evacuated by either prescient coding or change coding. It has been observed that neighboring pixels have similar values. Inter-Pixel redundancy is further divided in to Spatial, Spectral and Temporal redundancies. Correlation or redundancy of neighboring pixels is dealt by spatial redundancy. Spectral is related to different bands and color plans where as Temporal deals with adjacent frames in a sequence of image. While compressing an image, much of reliance is made on removal of maximum of spatial and spectral redundancies.

1.6.1.2Coding Redundancy

variable-length codes coordinating to the factual model of the picture or its handled rendition misuses the coding repetition in the picture. It is based on the principle that some pixel values are more common than others. It can also be related to the representation of information which has been expressed in the form of codes. Gray levels of an image are allocated more than the required number of code symbols which causes coding redundancy.

1.6.2 Psycho Visual Redundancy

Pictures are regularly implied for utilization of human eyes, which does not react with level with affectability to all visual data. The relative importance of different picture data segments can be abused to wipe out or decrease any measure of information that is psycho-outwardly excess. The procedure, which expels or decreases Psycho-visual excess, is alluded as quantization. It has been observed after innumerable experiments that all visual information is not necessarily picked up completely by human eye with the same sensitivity rather it differentiates some information as more important than the other. Therefore, extraction of only such information which is considered important to the human eye of a particular user by eliminating the unimportant information, which is termed as psycho visual redundant, comes under Psycho Visual Redundancy ambit.

1.7 Image Compression Process

Image compression process works over three tires. Starting from de-correlation that is done by many ways few are transformation, predictive coding and sub-band coding. This is followed by quantization which is to reduce the precision and achieve the better compression ratios. Last but not the least is the entropy encoding for optimizing the compression results. Compression processes are lossy and losses depending upon the techniques being used. Hence in general it can be said that there are three components in compression concatenated closely as shown in the figure 1.1.

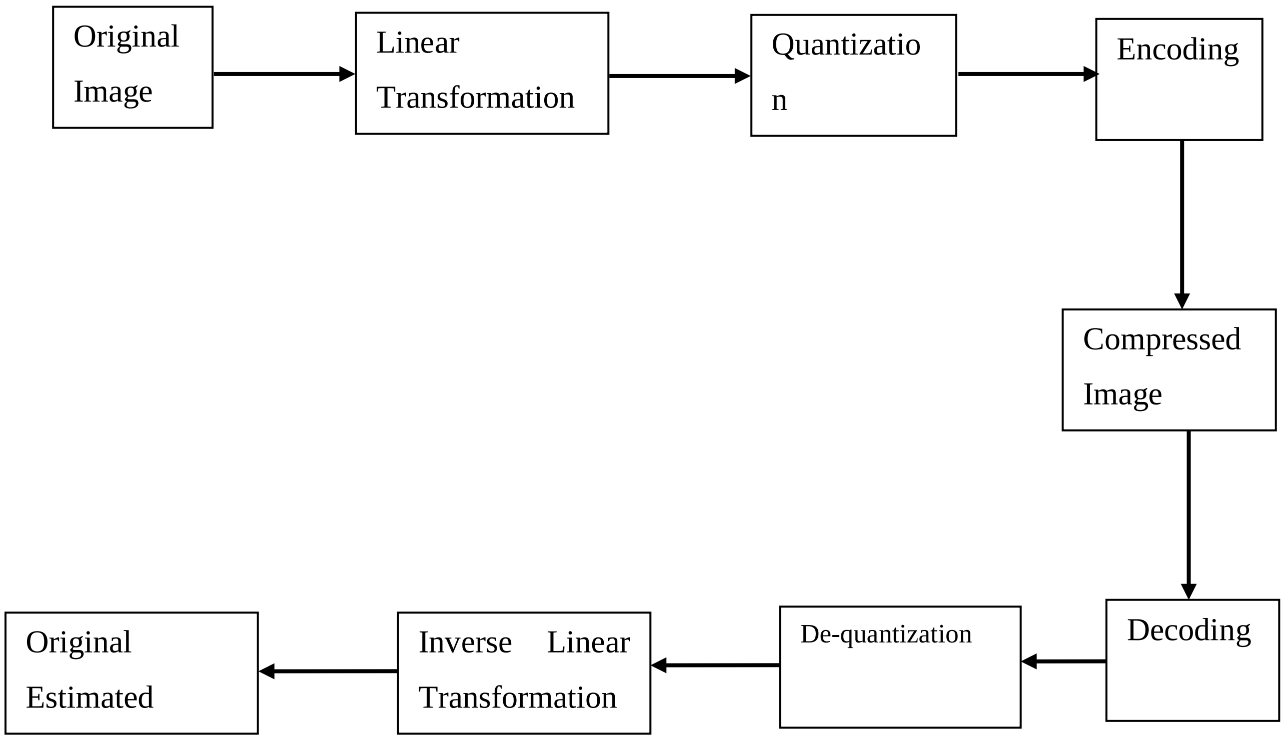


Figure-1.1: Compression and Decompression Process

Chapter-2

Wavelet Analysis and Transform

2.1 Introduction

This chapter will dilate upon wavelet, its analysis and wavelet transformation in general and Discrete Wavelet Transform, which forms the basis of present embedded encoding techniques, in particular. The facts as to how the wavelet has outperformed its predecessors and why DWT is being preferred over DCT or CWT as the basis of embedded codec will also come under discussion to establish the a coherent relationship with the upcoming chapters.

2.2 Wavelets

Wavelets are small wavelike mathematical functions of varying frequency and limited duration [2][3]. A single function f(x) that generates all these function is called the mother wavelet. Mother wavelet function is represented by the equation:-

$$f(x) = \sum_{k} \propto_k \ \varphi_{j,k(x)}$$

Where

$$\varphi_{j,k(x)=2^{\frac{j}{2}}\varphi\left(2^{j}x-k\right)}$$

Position of $\varphi_{j,k}$ (x) along the x-axis is defined by K whereas j tells about width of the function [4]. Higher frequency wavelet corresponds to narrow width and lower frequency corresponds to wider width[5].

2.3 Background

The signals nowadays are mostly time-domain signals in their raw format which once plotted, gives “Time Amplitude” representation. But then it is not the best method for presenting a signal as most of the distinct information is hidden in frequency contents. Since frequency is something that deals with the rate of change of that thing, therefore, if changes are rapid we say frequency is high and if these are slow then we say that frequency is low. Fourier Transform (FT) helps in measuring the frequency contents of a signal. If the Fourier Transform of a signal is taken in time domain, then representation of frequency amplitude of a signal is the outcome.It can be concluded that frequency information of a signal can be obtained by using Fourier Transform but it provides data about quantity of frequency in each signal and no information is rendered about at what time it existed.

Same time there is requirement when a signal needs information both in frequency and time domain for its apt utilization. Albeit, FT is reversible transform but only one representation i.e either frequency or time is present at a time therefore, could not meet desired requirements. However, with the introduction of wavelet transform representation of signal both in time and frequency became a reality.

2.4 Wavelet Transform

Data is converted in various frequency components with the help of wavelets. Out of these frequency components each one is representing to a specific resolution corresponding to the concerned scale [6]. The essential idea of this transform is to signify that any random function can be represented as a superposition of a basis function. Baby functions are obtained when mother wavelet like the one in figure 2.1 is scaled and shifted.

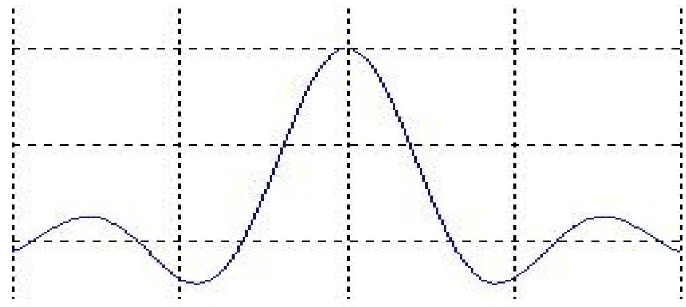
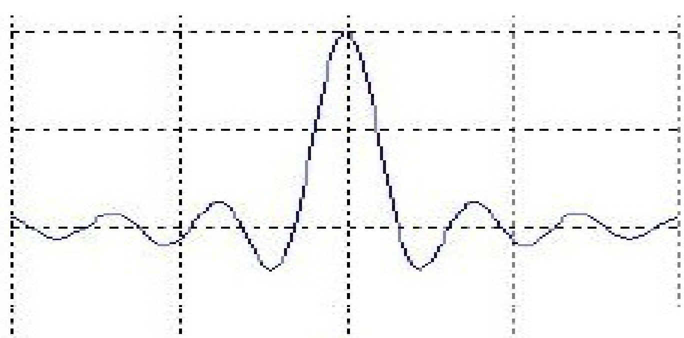
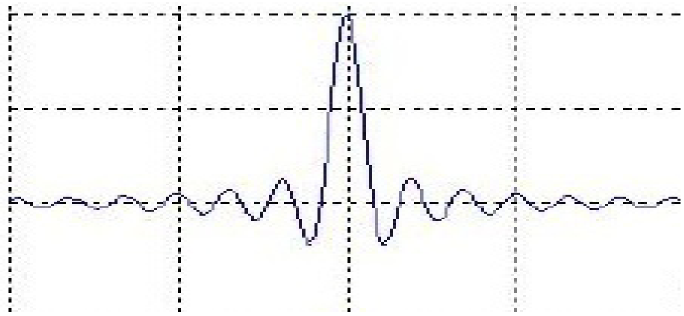


Figure-2.1: Mother Wavelet w(t)

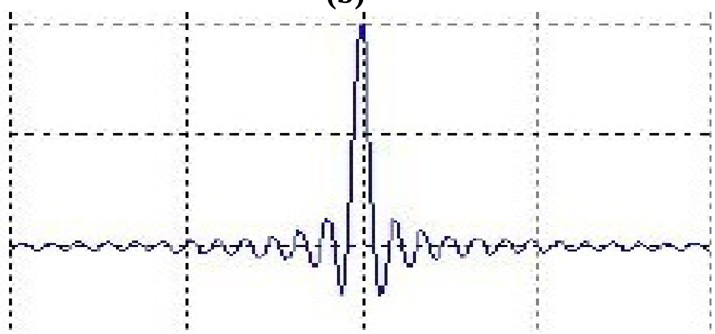
During the last two decades wavelet transform is being utilized in several fields of life like compression of an image, prediction of earthquake, turbulence and human vision. The analysis of wavelet creates a scale versus time plot of signal amicably. All functions of a wavelet w(2kt-m) can be obtained by utilizing a single mother wavelet w(t)as shown in figure 2.2. Wavelet gets narrower with the increase in the scaling factor. A wavelet which is wide in shape and size is equivalent to low frequency sinusoid of Fourier transform whereas narrow wavelets are the high frequency sinusoid of Fourier transform. Moreover wavelets with zero inner product are called orthogonal to each other.



(a)



(b)



(c)

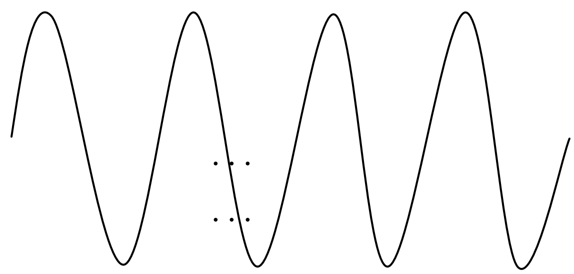
Figure-2.2: Scaling wavelet (a) k=1, (b) k=2 and (c) k=3

2.5 Wavelet Transform and Its Importance

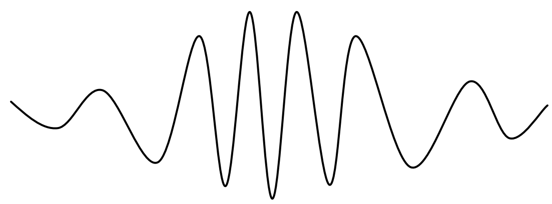
Nowadays wavelet transform has become very popular and is being is preferred over other techniques in many areas. One of the main areas out of them is Data compression. In data compression, wavelet transform is given priority due to the reason of its ability to compress data and image at various resolutions levels[7][8][9]. Local analysis of larger signal can also be performed with the help of wavelets[3][8]. This characteristic of wavelets distinguishes it from the others. Moreover, wavelet coefficients also help in plotting the exact position of the discontinuity in time domain.

2.6 Preference over Fourier Transform

As discussed earlier Fourier transform can provided representation either in time or frequency domain at a time whereas wavelets transform can express the properties of a signal both in time and frequency domain simultaneously. The basis functions in Fourier transform are the sine waves that extends from –ve infinity to +ve infinity means there is no existence in a defined interval. As we know that sine waves are relatively predictable for being smooth as compared to the wavelets which are symmetric and irregular therefore, to analyze signals with sharp changes an irregular wavelet is a better option than a smooth sine wave. Same is evident from the figure 2.3.



(a)



(b)

Figure-2.3:Compression of Sine Wave and Wavelet (a) Sine Wave (b) Wavelet

2.7 Multi Resolution Analyses (MRA)

In multi resolution analysis, a series of approximations of a signal are produced with the help of a scaling function [15] and alteration in data between neighboring approximations are encoded with by using wavelet function .In equation 2.1 g(x) is a signal that can be studied as expansion function's linear combination

$$g(x) = \sum_{k} \alpha_k \phi_k(x) \tag{2.1}$$

Here V as given under is the function space of expansion set $\{\phi_k(x)\}$

$$\mathrm{V} = span_k\{\phi_k(x)\} \qquad 2.2$$

And $g(x) \in V$ means that $g(x)$ is in the span of $\{\phi_k(x)\}$ and can be written in the form of Eq. 2.3. The coefficients α_k are computed by taking the integral inner products of the dual $\hat{\phi}_k(x)$'s and function $g(x)$. That is

$$\alpha_k = \langle \hat{\phi}_k(x), f(x) \rangle = \int \hat{\phi}_k^*(x)\, g(x)\, dx \qquad 2.3$$

If $\{\phi_k(x)\}$ is an orthonormal basis for *V*, then $\phi_k(x) = \hat{\phi}_k(x)$. If $\{\phi_k(x)\}$ are not orthonormal but rather an orthogonal reason for V, at that point the premise capacities and their duals are called bi-orthogonal.

$$\langle \phi_j(x), \tilde{\phi}_k(x) \rangle = \begin{cases} \delta_{jk} = 0 & , j \neq k \\ \delta_{jk} = 1 & , j = k \end{cases} \qquad 2.4$$

Presently consider the arrangement of development capacities $\{\boldsymbol{\phi}_{\mathbf{j,k}}(\mathbf{x})\}$ composed of whole number interpretations and twofold scalings of the genuine, square-integrable work $\boldsymbol{\phi(x)}$ which is called a scaling function where

$$\phi_{j,k}(x) = 2^{j/2}\, \phi\, (2^j x - k) \qquad 2.5$$

for $k \in Z$ and $\phi(x) \epsilon\, L^2(R)$. Because the shape of $\phi_{j,k}(x)$ changes with *j*. We denote the subspace spanned over *k* for any *j* as

$$V_j = span_k\{\phi_{j,k}(x)\} \qquad 2.6$$

The scaling capacity has four key prerequisites of multi-determination examination:-

The development elements of any subspace can be gathered from twofold determination duplicates of themselves. That is,

$$\phi(x) = \sum_k h_\phi(n)\sqrt{2}\,\phi(2x-n) \qquad 2.7$$

Where $h_\emptyset(n)$ as scaling function coefficients.

If we have a scaling capacity that qualifies the multi-resolution prerequisites, we can characterize a wavelet function $\psi(x)$ that covers the contrast between any two adjoining scaling subspaces, V_j and V_{j+1}. We can define the set $\{\psi_{j,k}(x)\}$ of wavelets

$$\psi_{j,k}(x) = 2^{j/2}\,\psi\,(2^j x - k) \qquad 2.8$$

for all $k \in \mathbf{Z}$ that spans the space W_j where

$$W_j = span_k\ \psi_{j,k}\ (x) \qquad 2.9$$

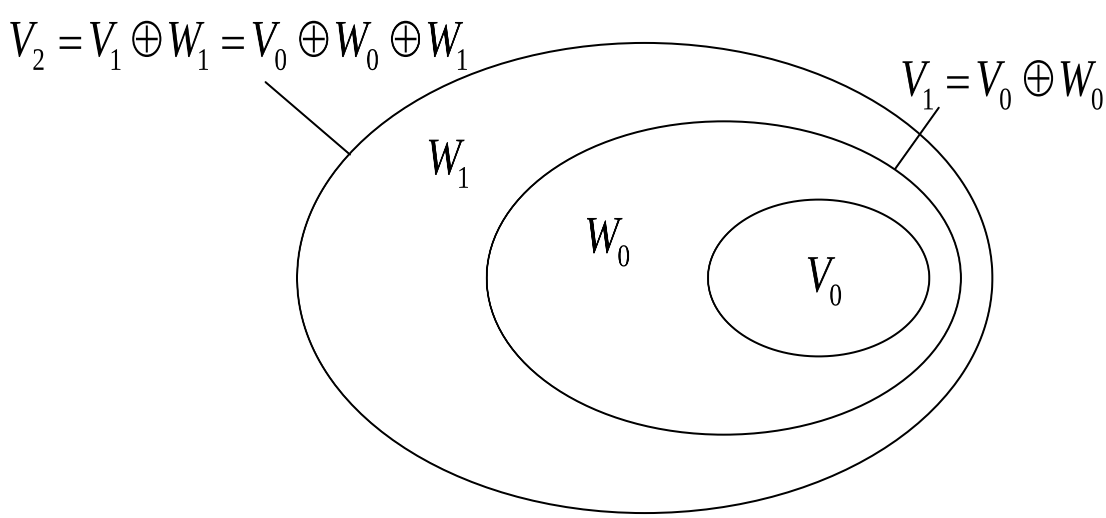


Figure-2.4: The relationship of scaling and wavelet function spaces

Scaling & wavelet functions subspaces as shown in figure 2.4 are linked by

$$V_{j+1} = V_j \oplus W_j \qquad 2.10$$

Hence, space of all quantifiable, square-integral capacity can be spoken to as

$$L^2(R) = V_0 \oplus W_0 \oplus W_1 \oplus W_2 \oplus \dots \qquad 2.11$$

Similar to the scaling function, the wavelet function can be stated as a weighted total of moved, twofold determination scaling capacities. That is,

$$\psi(x) = \sum_n h_\psi(n)\sqrt{2}\,\phi(2x-n) \qquad 2.12$$

Where $h_\psi(n)$ are called the wavelet function coefficients. It can be shown that $h_\psi(n)$ is related to $h_\phi(n)$by

$$h_\psi(n) = (-1)^n\, h_\phi(1-n) \qquad 2.13$$

2.8 Types of Wavelet Transform

One of the most renowned time – frequency transform of the time is wavelet transform. For the analysis of frequency components in time domain wavelet functions are used on the same lines like sine and cosine waves are utilized in Fourier transform to carry out the analysis of a signal. Wavelet transforms can be discussed under following :-

2.8.1 Continuous Wavelet Transform (CWT)

The continuous wavelet transform is the natural extension of discrete transform. It transforms a continuous function in to a much redundant function of two continuous variables which are scale and translation. The CWT offers the description of a signal in time and frequency domain which is kind of redundant but finely detailed. Problems related to signal identification and detection of concealed transients (difficult to detect the short lived elements of signal) are specifically treated with the help of CWT[16]. Fourier analysis in Fourier transform is mathematically expressed as:-

The result obtained will be CWT when a signal is multiplied with by a wavelet. Wavelet is used to define the basis functions of the wavelet transform.

2.8.1.1 The inverse Wavelet Transform

The Fourier transform of the mother wavelet ψ(t)results in the form of X(t) and it must fulfill two conditions. Number one, to stay away from This is called the admissibility condition and can be expressed mathematically as follows

$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \qquad (2.19)$$

Number two, the mother wavelet must have finite energy.

$$\int_{-\infty}^{\infty} |\psi(t)|^2 \mathrm{dt} = \infty \qquad (2.20)$$

2.8.2 DWT Decomposition

In Fourier analysis, sinusoidal basis functions of different frequencies are obtained once Discrete Fourier Transform (DFT) decomposes a signal. This is a lossless transformation in which original signal can be completely recovered from its DFT representation. However, in case of wavelet investigation, the DWT breaks down a flag into an arrangement of commonly orthogonal wavelet premise capacities. These functions don’t match with the sinusoidal basis functions .Moreover, wavelet functions are translated, scaled and dilated versions of mother wavelet φ. DWT is invertible like that of Fourier analysis, so that the original signal can be completely recovered. Haar wavelets and Daubechies set of wavelets are two of the most popular wavelets. The common properties of the two describe that wavelet functions are spatially localized, scaled, dilated and translated version of the mother wavelet. Moreover, each set of wavelet function makes an orthogonal set of basis functions.

2.8.2.1 DWT in One Dimensional

One-dimensional DWT is a multi-determination recurrence decay and limitation of a one-dimensional, discrete-time flag [11].

Here $\phi(t)$ is the scaling function which is orthogonal and $a_{j,k}$is to express the scaling. Orthogonal wavelet function has been expressed as $\psi(t)$ and $b_{j,k}$ is used to express wavelet coefficients. The analysis equation of bi-orthogonal DWT for a signal that belongs to $L^2(\mathcal{R})$ is written as

$$\tilde{a}_{j,k} = \int x(t) 2^{j/2} \tilde{\phi}(2^j t - k) dt \tilde{b}_{j,k} = \int x(t) 2^{j/2} \tilde{\psi}(2^j t - k) dt \quad (2.23)$$

For a signal that belongs to $L^2(\mathcal{R})$ the the synthesis equation of bi-orthogonal IDWT is defined as:

$$x(t) = 2^{N/2} \sum_k \tilde{a}_{N,k} \phi\ (2^N t - k) + \sum_{j=N}^{M-1} 2^{j/2} \sum_k \tilde{b}_{j,k} \psi\ (2^j t - k) \quad .24)$$

2.8.2.2 Two Dimensional DWT

It is of a great utilization for the processing of images and applications related to computer vision. One can say that it is a kind of straight forward extension of the one dimensional discrete wavelet transform. It can be implemented by utilizing down-samplers and digital filters [17]. It goes without saying that 2-D separable wavelet transform is made when two of 1-D wavelet transforms are connected in series. The data is passed through the rows and then through the columns of the 1-D wavelet transform.

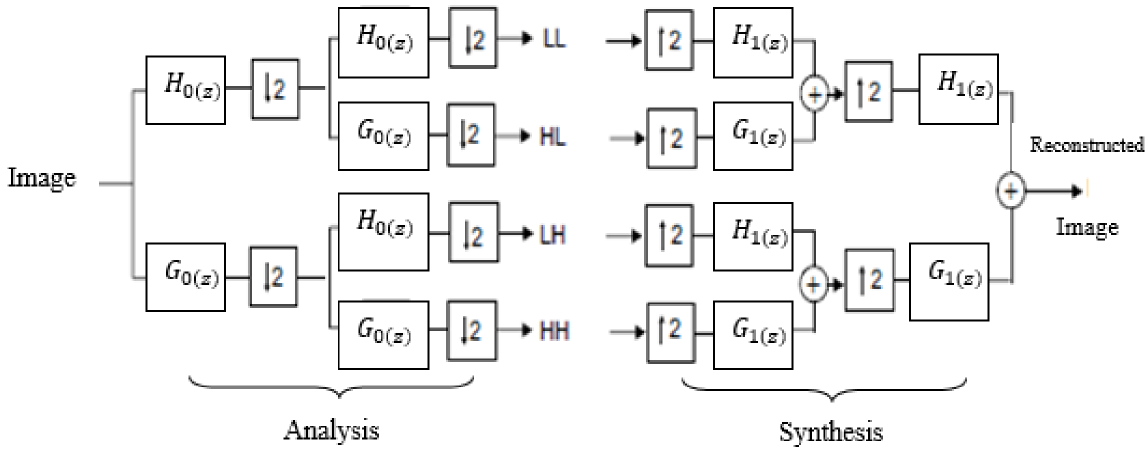


Figure-2.5: One level filter bank for computation of 2-D DWT

Projection of the image into basis of 2-D will result into transform coefficients. When two 1-D basis are multiplied the consequence is a 2-D separable basis function. Therefore for the images, we have 4xbasisfunctions which have been represented in the equations from 2.25 to 2.28.

$$\phi\,(u,v) = \phi(u)\,\phi(v) \quad (2.25)$$

$$\psi_1(u,v) = \psi(u)\,\phi(v) \quad (2.26)$$

$$\psi_2(u,v) = \phi(u)\,\psi(v) \quad (2.27)$$

$$\psi_3(u,v) = \psi(u)\,\psi(v) \quad (2.28)$$

The scaling function of the images is $\phi\,(u,v)$ whereas $\psi_1(u,v), \psi_2(u,v)$and $\psi_3(u,v)$ expresses the wavelet functions.(Figure 2.6):-

LL Sub-bands (Approximations)

- LH Sub-bands (Vertical Details)
- HL Sub-bands (Horizontal Details)
- HH Sub-bands (Diagonal Details)

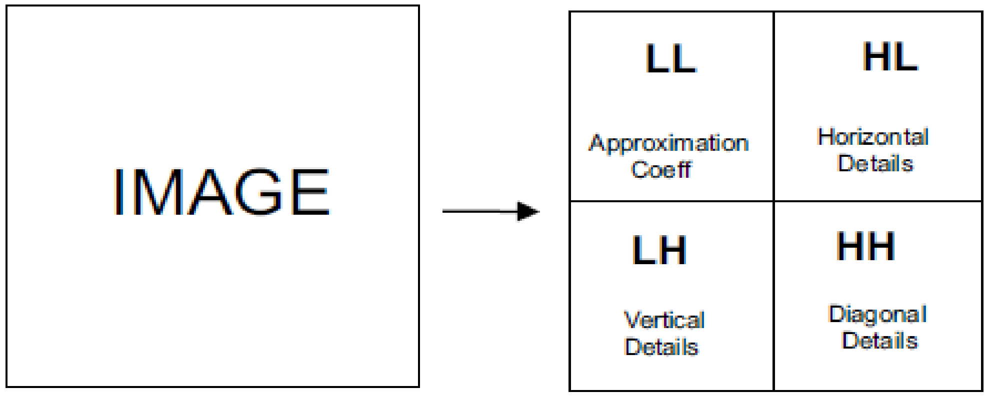


Figure-2.6: 2-D Decomposition up to one level

Chapter-3

Embedded Zero-Tree Wavelet (EZW) Transform

3.1 Introduction

EZW is a compression algorithm which performs compression to a good extent with many types of images. The E of EZW is for embedded thereby depicting that it is a progressive coding. Z stands to represent data structure of Zero-trees which encodes the data and W for wavelet transform on which EZW encoders works on. In this chapter, EZW encoding has been discussed in detail.

3.2 Embedded Encoding

The embedded coding is defined by the fact that the order of the coded bits is set in accordance with their significance and lower code rates are adjusted at the start of the bit stream. To achieve the intended bit rate specified by channel, progressive encoding is capable of terminating encoding process at any stage. This is done by maintaining the bit calculation and truncating stream of bit by encoder, whenever the set bit rate is attained. Albeit, EZW uses more simple and state of the art progressive coding, yet we can compare it with one, where most significant bit plan is the starting point for the coding and gradually carries on with the most significant bit plan coming next and so forth. Reconstruction error at receiver will occur, if before addition of the less significant bit plan to bit stream we meet the target, reconstruction error is reduced at given target bit rate with the help of " significant ordering" of the embedded bit stream.

Accordingly, compression algorithm which generates embedded code, first thing that should be sent on the transmission network is coarser variant of the picture took after by refinement details within the framework of progressive broadcast.

The diagram of an embedded image coding system is as under:

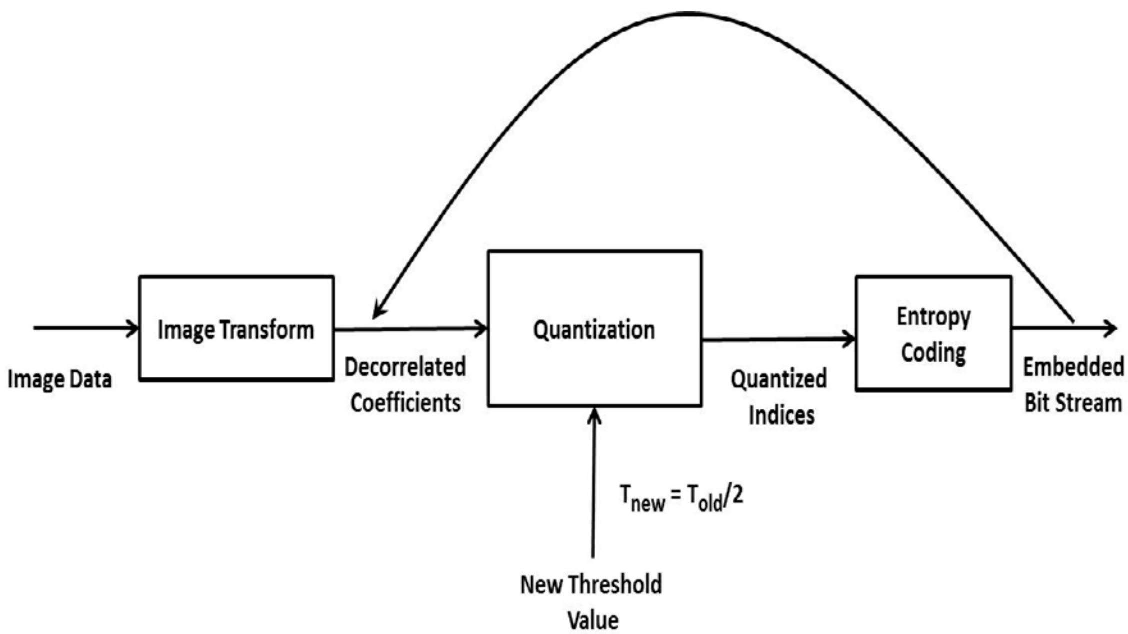


Figure-3.1:Embedded Coding Scheme

3.3 Zero-Tree Structure

Now let us see what is a zero tree, a quad-tree is termed as Zero-tree when a root node in a tree structure is more prominent than or, on the other hand equivalent to alternate nodes but lesser than the given threshold that compare the wavelets against it.

A quad-tree with all the nodes smaller or equal to the root is termed as zero- tree. A single symbol is used to code it and the decoder reconstructs it as quad-tree which is filled with the zeros. A root smaller than the threshold, for which the coefficients are presently being measure, is required to clutter this definition. [18].

The encoder for EZW works on two basic observations, number one is that all natural images are low pass spectrum images. If passed through the wavelet transform energy in sub-bands decrease with the decrease in the scale (low scale means high resolution). It means that progressive encoding appear to be the best choice as higher bands add the details only

.Number two is that the larger wavelet coefficients are given more importance over those of smaller ones..

3.4 EZW Encoding Process

The EZW algorithm results in a fully progressive bit streams for image coding[19] and the compression enactment of this technique is comparatively better than previously known methods. The EZW process is based on following major theories:

1) Hierarchical sub-band breakdown.
2) Zero-tree coding.
3) Entropy coded successive-approximation quantization.
4) A prioritization technique to define importance of coefficients basing on various characteristics [20].
5) Lossless data compression through entropy coding schemes.

In this algorithm encoding becomes the most important part. The test image undergoes the filters for DWT which yields the transform coefficients. This transforms results into de-correlated coefficients with as fewer dependencies among the samples as possible. Then the symbols are produced by quantizing these transformed coefficients for compression process. Here the embedded coding is done by using successive approximation quantization. It has been seen that it is the quantization phase where most of the information is lost[21].Resultantly, to find the significance in quantization stage the threshold value is fixed. In the final stage of encoding, the bit stream of symbols is sent for compression. At the decoder end, reverse process as enunciated at the encoder is applied. EZW algorithm has a privilege [22] that user can select bit rate as per his desire and encode the image according to that. Figure 3.2 explains the details as that how EZW coding algorithm is applied.

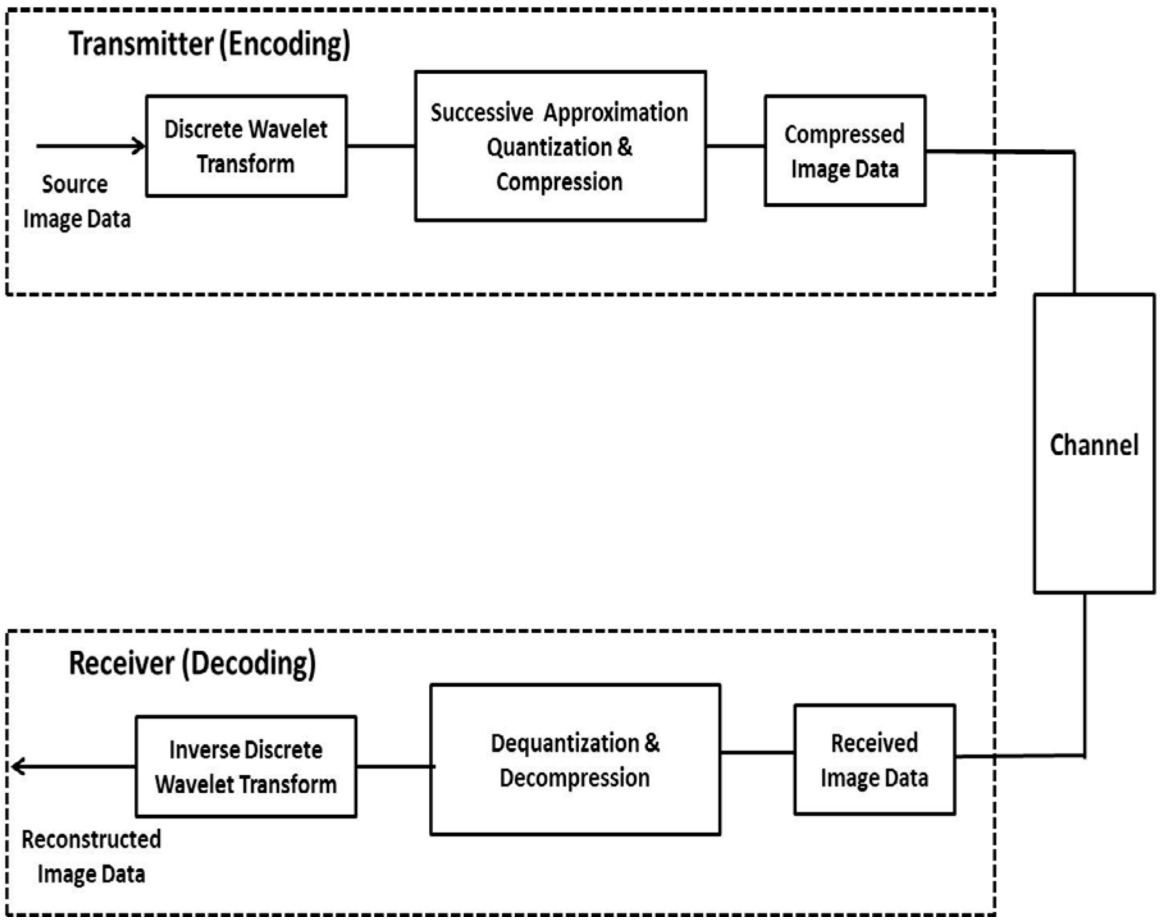


Figure 3.2: EZW Compression Diagram

In EZW image compression algorithm, some of the information is lost due to residual matrix left at transmitter end. It is because that the real images are made up of mostly low frequency information which is highly correlated. Moreover, the importance of high frequency information (such as edges) cannot be overlooked at the same time as it is of significance due to human perception of the image quality. Therefore, in high quality coding scheme it is necessary to precisely signify the high frequency components. At root node the transformed coefficients can be measured as the tree or the trees with the lowest frequency components also, with the offspring of each tree hub being the spatially related coefficients in the following higher recurrence band. It has been observed with great probability that one or more sub-trees will have zero or nearly zero coefficients [23].

In EZW, the measurable properties of the trees are used to proficiently code the areas of the substantial coefficients [24]. Since a large portion of the coefficients will either be zero or approximately zero, the spatial areas of the critical coefficients consist of a large portion of the entire size of a compressed image. The significance of a coefficient is determined by comparing its modulus value or value of a node and its children in the case of a tree above with a specific threshold. Initially, the threshold is taken by considering the magnitude of maximum coefficient and then by iteratively lowering the threshold as per algorithm. Four kinds of different symbols[19] are used by the EZW to represent the wavelet coefficients:-

- Zero-Tree Root (ZTR)
- Isolated Zero(IZ)
- Significant Positive (Pos)
- Significant Negative (Neg)

Two binary bits are used by EZW to represent the above mentioned symbols. After every dominant pass, the existing threshold is updated by a factor of two. By scanning the trees and emitting one of the four symbols, the dominant pass encodes the significant coefficients which have not yet been found significant in previous iterations. Conditions for scanning of the children of a coefficient is either should be significant or an isolated zero. One bit (MSB) is produced by the subordinate pass for each coefficient found significant in the previous significance passes.

3.5 Encoding Concept of EZW

3.5.1 Progressive Coding

In EZW algorithm, bits emerge as per their importance order in a bit stream. It is due to this feature that the beginning of the bit streams contains all the low rate codes[25]. This progressive code signifies a structure of binary conclusions that differentiate a picture from the invalid or all grey picture. The encoding stops when the user set target rate or distortion metric is reached [20]. This implies that the embedded coder can terminate encoding on user set parameter [26] and offers the best representation of the image.

Stream which is binary coded, can comprehend progressive broadcast by utilizing multi- threshold EZW coding, as a result, coding rate / distortion metric can be restricted accurately. The coding process can be terminated either when bit budget is consumed [27] or compression ratio is reached. So at any given rate of coding, the coefficients required to represent an image will always contain the required information that was required at much lesser rates. Therefore, this may be done by choosing a target bit rate which fixed and decoder retains the option of terminating the decoding process at any point of time. So it is concluded that the decoder has the capability to interfere [28] the process of decoding at any time in the bit stream and still has the ability to reconstruct the image. For that reason, the compression technique which is progressive in nature sends the low frequency information at the start. This is followed by the high frequency components i.e details within the framework of progressive broadcast.

3.5.2 Significance Map Encoding

It is has been observed in EZW scheme that a reasonable amount of the overall bits strength is needed for the coding of the position information [29]. Therefore, a significance map can be defined as a binary function the value of which tells whether a coefficient is significant or insignificant. A coefficient is quantized to zero if it not significant. So it is known to a decoder that no further information is needed by the significance map about that coefficient. However, non-zero value is assigned to a significant coefficient. It is achieved by encoding the location of the zeros [19]. After lot of statistical analysis it has been experimentally proved that in the wavelet transform, across different scales zeros can be forecasted accurately. Assumption on which EZW is based stats if at coarse scale, a wavelet coefficient is irrelevant as for a given scale then at same spatial location at finer scales all the coefficients of same orientation are also likely to be insignificant [20][23].In the significance map, the location of significant and insignificant wavelet coefficients signified specifically for every threshold T. Zeros are used to specify the positions of insignificant coefficients and locations of the significant coefficients are represented by values of “one”[23].A Zero-tree Representation is used to code the significance map .It permits that insignificant coefficients are identified and predicted accurately across the scales. With this technique, total cost of encoding the significance maps is reduced by grouping the insignificant coefficients in exponentially growing trees across the scales, and by encoding these coefficients with zero-tree symbols [20].Samples of a significance map and quantized coefficients are presented in the table 3.1:

Quantized Coefficients

64	56	48	32	24	16	0	0
56	28	40	24	16	23	0	0
40	40	30	24	16	8	0	8
32	32	32	24	24	16	0	0
24	24	16	8	0	0	8	0
16	16	8	0	0	8	0	0
0	0	0	8	0	0	0	0
0	0	0	0	0	0	0	0

Significance Map

1	1	1	1	1	1	0	0
1	1	1	1	1	1	0	0
1	1	1	1	1	1	0	1
1	1	1	1	1	1	0	0
1	1	1	1	0	0	1	0
1	1	1	0	0	1	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0

Table 3.1: Quantized Coefficients & Significance Map

The significance map is about location of location of non zero valued coefficient which are going to be transmitted and the coding of the significance is one of the important views of low bit rate image coding. After quantization followed by entropy coding, the zero symbol which occurs with the highest probability, should be extremely high in order to attain very low bit rates. Therefore, in this way a large portion of the bit budget is utilized in encoding the significance map. Due to this not only efficient encoding the significance map is achieved but also it offers a higher efficiency in compression.

To understand the significance of map coding in a better way, let us consider a encoding system with a typical transform method. There are three basic parts of a typical low bit-rate image coder which are shown in the figure 3.3 below[30].

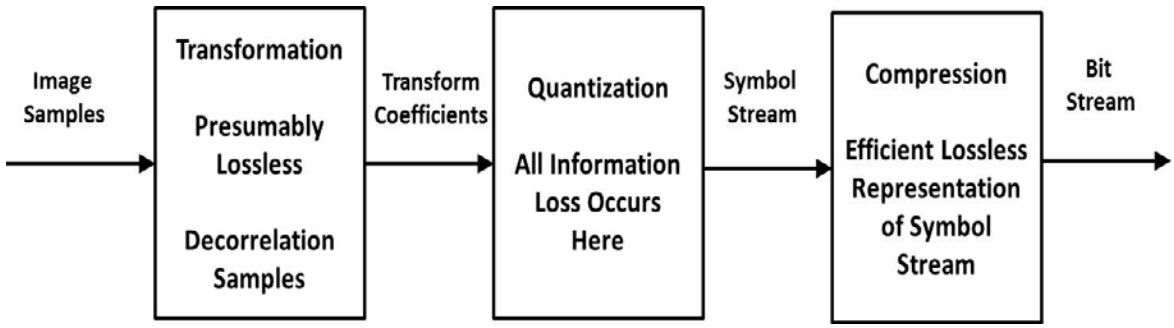


Figure 3.3: Low bit rate image coder

3.6 The Successive Approximation Quantization (SAQ)

The SAQ [26]provides a vital facet to embedded compression algorithms. It goes without saying that symbol stream produced by these algorithms consists of the bit streams for all possible lower rates. An embedded code is characterized by J.M Shapiro by defining two properties [20]:

1) While same data is encoded at different rates, for the level of smaller one the two resulting images must match exactly. This represents that for a given data rate a coded symbol consists of all the symbols for smaller data rates. As we add more symbols to them, the representations get more precise.

2) For a given data rate there should be a good representation.

SAQ is applied to perform the progressive / embedded encoding which is linked to the bit-plan encoding of the magnitudes [32]. The SAQ applies a series of thresholds sequentially to check the significance of given data. The initial threshold To is selected in such a way that |Xj|< 2Tofor all transform coefficients. After this the subsequent thresholds are selected as Ti = Ti-1/2. Two separate lists of wavelet coefficients are present in the process of encoding and decoding. The coordinates of the coefficients, not found significant so far in the same order during the process of initial scan, are part of a list known as "Dominant list". The sub-bands are ordered in this scan, and with each sub-band, the set of coefficients are well-arranged. Absolute value of significant coefficients is stored in the subordinate list. For each threshold the list is scanned once.

To govern their significance, the coefficients are compared to the threshold To during the dominant pass. In the next step, if a coefficient is proved to be significant then its sign is ascertained as positive or negative. This significance map is coded as zero-tree later. A coefficient is coded as Pos if it is found to be significant then each time its absolute value is attached to a list called subordinate list. The coefficient in the wavelet transform range is set to zero to prevent the occurrence of significant coefficient as a zero-tree on future dominant passes at smaller thresholds. All those values which were previously found significant will now be subjected to the subordinate pass. Binary '1' is used to code the refinement for every absolute value in the subordinate list. This points out that the old uncertainty interval contains the true value in the upper half. However, '0' symbol shows that the value is in the lower half of the old uncertainty interval. The entropy encoding is done to the string of the symbols that is generated from this from this binary alphabet during a subordinate pass. The magnitudes on the subordinate list are sorted in decreasing magnitude after the completion of a subordinate pass. The process keeps alternating between subordinate passes and dominant passes and the threshold is halved before each dominant pass.

3.7 The EZW Algorithm

The coefficients' magnitude is compared by the encoder with the initially selected threshold. Encoder sends the signal to the decoder whether the magnitude is smaller or greater than the given threshold. For nearly precise results at the decoder end, encoder must also send the information about threshold value. The process is repeated till we locate the littlest coefficient (wanted to be sent) gets bigger than the last processed edge to get immaculate remaking.

If the both encoder and decoder use a predefined criterion for the threshold rather than transmitting the limit in each pass, then bandwidth can be saved as there will be no requirement of sending the threshold to the decoder. The threshold represents the quantity of bits to explain the paired estimation of the coefficients size if the predefined paradigm for the limit is a genuine of the forces of two [20]. As per Shapiro it is known as “bit plane coding”.

The decoder needs information about the position of the coefficients to reconstruct the transmitted signal. Efficient encoders are differentiated from the inefficient ones with the help of coding of the positions.

EZW encoder uses a predetermined sequence of scanning to encode the extraordinary positioning of the coefficients as demonstrated in the figure 3.4.Using zero-trees a considerable measure of positions are coded flawlessly. Many orders scan can be used by the encoder[33], as long as it checks the lower coefficients of the sub-bands before scanning the higher coefficients of sub-bands.

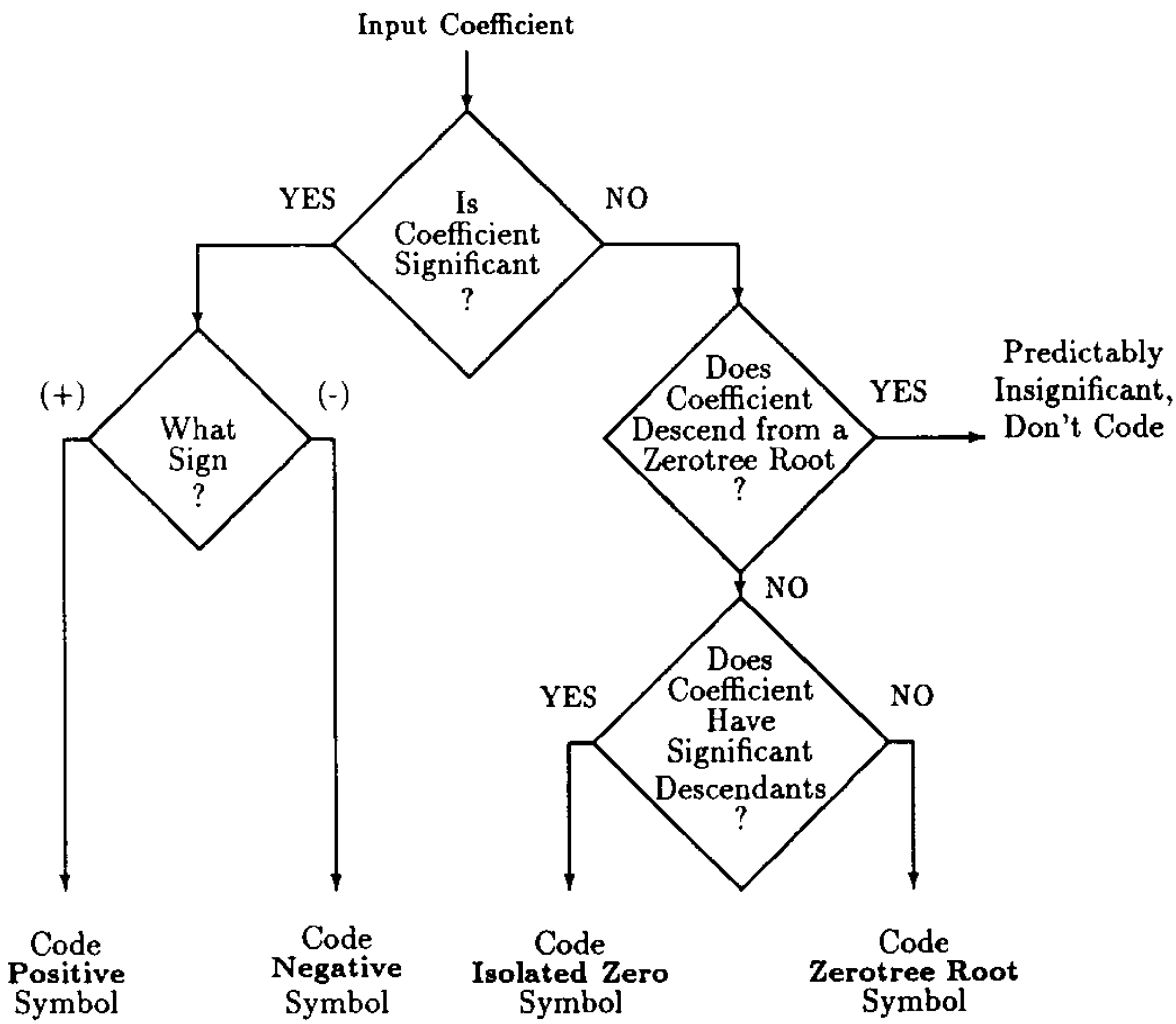


Figure 3.4: Flow chart of encoding a significance map coefficient

The final results of compression are also affected by the order of scan. Initial threshold tobe calculated by adopting bit plane coding by using the relation as under

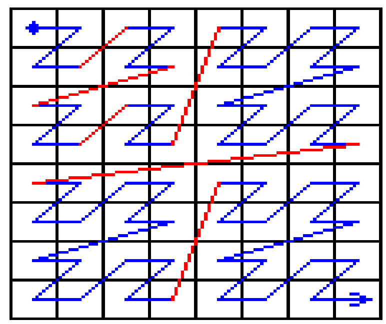
$$t_0 = 2^{\lfloor \log_2 (\mathrm{MAX}\ (|\gamma\ (x,y)|)) \rfloor} \qquad)$$

Where MAX (.) indicates the highest value of the wavelet coefficients $\gamma(x, y)$

3.8 Example

To illustrate the above stated algorithm we take an example shown in Table 3.2 and figure 3.5.

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4



(a) (b)

Table 3.2: (a) data set **Figure 3.5:(b) scanning order (Morton scan)**

Here the example is being explained after taking the data from table 3.2 and running the Morton scan in figure 3.5[20]. With initial threshold calculated as t_0=32 the EZW algorithm generates following bit stream after one pass.

D1: pnztptttztttttttptt

S1: 1010

D2: ztnptttttttt

S2: 100110

D3: zzzzzppnppnttnnptpttntttttttpttptttttttttpttttttttttt

S3: 10011101111011011000

D4: zzzzzzztztznzzzzpttptpptpnptntttttptpnpppptttttptptttpnp

S4: 1101111101100100000111011010001001010110 0

D5: zzzzztzzzzztpzzzttptttnptppttptttnppnttttpnnpttpttppttt

S5: 101111001101000101111101011011001000000001101101100110 00111

D6: zzzttztttztttttnnttt

With the consideration in mind that EZW algorithm will require initially (at least) two bits for coding of the symbols in the alphabet { Pos, Neg, ZT, IZ} and another bit to code the symbol Z, therefore, a total of 33=26+7 bits were utilized after the first pass.

Chapter-4

SPIHT (Set Partitioning in Hierarchical Trees)

4.1 Introduction

Albeit, EZW was an efficient and computationally simple technique yet it had room for improvements which were done by Amir Said and A. Pearlman in 1996[37]. Their technique works the concept of ordering which is partial by the magnitude having an algorithm of partitioning sorting, bit planes on different scales of an image transform have ordered transmission and self similarity is being exploited.SPIHT is a modified EZW algorithm proposed by Amir Said and Pearlman. The transmission of the ordered coefficients, and sub-bands of equivalent orientation having self-similarity, which were considered as the best features of EZW algorithm, were also incorporated in SPIHT. The reason of high PSNR of SPIHT than EZW is a special symbol that demonstrates the significance of child nodes of significance parent, and separation of child nodes from second generation descendants [34][35][36].

SPIHT algorithm outperforms the performance of its predecessor. SPIHT bit stream is in possession of a distinct property of compactness. It is due to this characteristic that only marginal gain is obtained once the output bit stream of SPIHT is further passed through the entropy encoding schemes.

Unlike EZW no ordering information is clearly transmitted to the decoder and this becomes another signature of SPIHT algorithm. Moreover, the decoder reproduced the execution path of the encoder and the ordering information is recovered. But then for smooth execution or recovery of the actual information the there is need that both the encoder and decoder posses the same execution time.

4.2 Progressive Transmission Scheme

To use partial ordering, a comparison of the magnitudes of the coefficients is drawn with a set of octave decreasing threshold [37]. The hierarchical sub-band transformation can be expressed like wavelet as under

$$c = \Omega(s) \tag{4.1}$$

Here c is the output array of the transformed coefficient, which are produced once Ω sub-band transformation is applied on the original image array s. The coefficient array has the same dimensions for the output and the original image. Encoder and decoder process the coefficients as per the defined SPIHT algorithm. To reform an estimated image $\hat{s}$ inverse transformation needs to be taken from the estimated array of coefficients $\hat{c}$ as under

$$\hat{s} = \Omega\,(\hat{c}) \tag{4.2}$$

For reconstruction of the estimated image at the decoder end, the mean-squared error is calculated with the help of following

$$D_{mse}\,(s - \hat{s}) = \frac{\|s-\hat{s}\|^2}{N} \tag{4.3}$$

$$\frac{\|s-\hat{s}\|^2}{N} = \frac{1}{N}\sum_{n_1}\sum_{n_2}(s_{n_1,n_2} - \hat{s}_{n_1,n_2})^2 \tag{4.4}$$

Where s_{n_1,n_2} is the intensity value of the pixel at location n_1, n_2 of image having N number of pixels. Mean square error (MSE) is independent because the sub-band transformation is lossless.

$$D_{mse}\,(s - \hat{s}) = D_{mse}\,(c - \hat{c}) \tag{4.5}$$

$$D_{mse}\,(c - \hat{c}) = \frac{1}{N}\sum_{n_1}\sum_{n_2}(c_{n_1,n_2} - \hat{c}_{n_1,n_2})^2 \tag{4.6}$$

In the above mentioned equation, c_{n_1,n_2} shows the transformation coefficient at the position n_1, n_2. For every coefficient, $\hat{c}_{n_1,n_2}$ is set to zero by the decoder. If c_{n_1,n_2} is the encoder send coefficient value, then mean square error is decreased by $\frac{(c_{n_1,n_2})^2}{N}$. This is the indication of the coefficients with greater magnitude have high significance in embedded bit stream encoding than those of smaller ones. It is because they play an important role in reducing the mean square error and thereby producing better reconstruction as compared to the smaller valued coefficients. Therefore, the coefficients are arranged with respect to their magnitudes in the embedded stream coding. Whereas in EZW algorithm significant coefficients are arranged in the subordinate pass. Representation in general form as

$$\left|\log_2|c_n(k)|\right| \geq \left|\log_2|c_n(k+1)|\right| \qquad k = 1,2,\ldots\ldots..,N \tag{4.7}$$

In equation 4.20 $c_n(k)$represents the coefficients which have been ordered according to their magnitude values.

Following example will enable us to assimilate ordering concept in a better way. Here -3, -9, 16, 5, -57, 8, 38, 2, -12, 14, -17, -6, 25, and -7 are the array of coefficients. They can be arranged by using above ordering equation as follows (Table 4.1):

Coefficient magnitude	57	38	25	16	17	14	12	9	8	7	6	5	3	2
Sign bit	1	0	0	0	1	0	1	1	0	1	1	0	1	0
Bit-5 (msb)	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Bit-4	1	0	1	1	1	0	0	0	0	0	0	0	0	0
Bit-3	1	0	1	0	0	1	1	1	1	0	0	0	0	0
Bit-2	0	1	0	0	0	1	1	0	0	1	1	1	0	0
Bit-1	0	1	0	0	0	1	0	0	0	1	1	0	1	1
Bit-0 (lsb)	1	0	1	0	1	0	0	1	0	1	0	1	1	0

Table 4.1: Binary representation of order of coefficients

4.3 Set Partitioning Sorting Technique

To send the coefficients explicitly there is no need of ordering information for the Set partitioning technique. However, both decoder and encoder follow the same implementation path. If the encoder transmits the magnitude comparison results then the execution path helps the decoder to recover the sorting information.

So it can be said that the set partitioning is devoid of explicit ordering of the coefficients rather the coefficient values are observed for any given value of n, if followed by the inequality $2^n \leq |c_{n_1,n_2}| < 2^{n+1}$. A significant coefficient must follow the inequality $|c_{n_1,n_2}| > 2^n$ and if it this inequality is not being followed then it is not a significant coefficient. The coefficient subset T_n is examined if

$$\max_{n_1,n_2 \in T_m} |c_{n_1,n_2}| \geq 2^n \tag{4.8}$$

Same holds well for the subset T_n, it stands significant or insignificant if it satisfies or does not satisfy the inequality respectively. We classify significant and insignificant subsets by portioning T_n if it is significant. So until we come across a single significant coefficient we continue with the partitioning of significant subsets into significant and insignificant coefficients. A sub-band hierarchical framework is followed in set partitioning technique.

4.4 Spatial Orientation Tree

It has been observed that frequency components of an image that are low contain most of the energy of an image. Therefore, as we move from highest to lowest levels of sub-band pyramid there is a decrease in the value of the variance [37]. Moreover, spatial similarity between the sub-bands can also be observed as sub-bands are linked spatially with each other. In the spatial orientation tree, the connection of the sub-bands is presented in figure 4.1 in the form recursively split four bands. The coordinates of a pixel are made use of to represent the associated node. The offspring, which are four in number for every node, are represented with likewise pixel location in the orientation pyramid of next lower level as explained with the help of arrows in the diagram below. The LL sub-band which resides at the highest level of the pyramid is exempted and does not hold any such relationship. It is the pixel in sub-band which forms the root and composes adjacent 2x2 pixels' group. Out of the four pixels of LL band three has offspring in HH, LH and HL sub-bands which are exiting in the same scale since only three sub-bands which decides the descendants. Whereas one pixels in LL band which is marked with '*', as displayed in figure 4.1 does not determines any descendants.

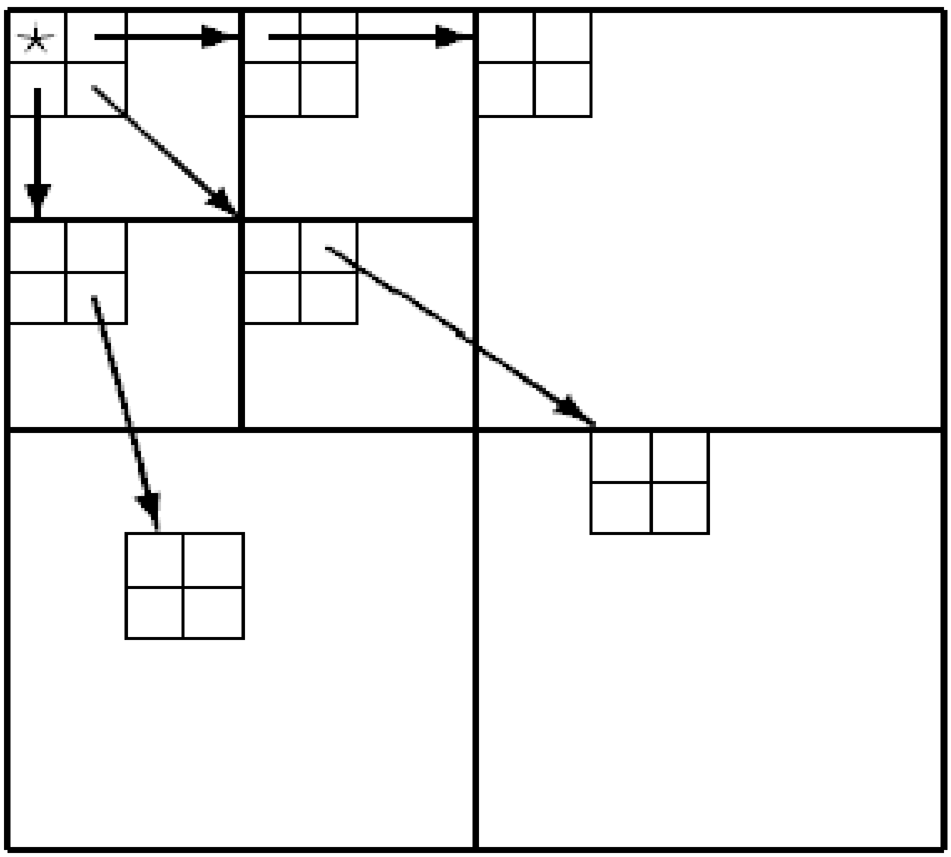


Figure 4.1: Spatial orientation tree

4.5 Set Partitioning Rules and Algorithm

For better assimilation of the concept, few important sets of notation which are utilized in SPIHT algorithm need to be understood well. It is also underlined that some of the SPIHT algorithm rules are also derived from these notations.

1) $\mathbf{O(n_1, n_2)}$**:** O stands for the offspring and $\mathbf{n_1}$&$\mathbf{n_2}$are the pixel coordinates of the offspring or it can be said that these represents the set of offspring of the node$(\mathrm{n}_1, \mathrm{n}_2)$. The size can be four or zero determined by the number of offspring. For example, the $\mathrm{O}(0,1)$ in the figure 4.2 has coordinates of the pixels$\mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3$ and b_4.

2) $\mathbf{D(n_1, n_2)}$ **:** Notation D is for the descendants whereas $\mathbf{n_1}$&$\mathbf{n_2}$ represents the positions of the pixels. So this represents the set of descendants. By descendants here includes the offspring, offspring's offspring and so on depending upon the number of sub-bands. For example the descendants set D(0,1) consists of the coordinates of the pixels $\mathrm{b}_1, \ldots\ldots\ldots, \mathrm{b}_4, \mathrm{b}_{11}, \ldots\ldots\ldots, \mathrm{b}_{14}$, ... $\mathrm{b}_{41}, \ldots\ldots\ldots, \mathrm{b}_{44}$. Since by now we know that every node may have four or no offspring therefore the size of this node may be either four or zero.

3) $\mathbf{L(n_1, n_2)}$ **:** This notation is used to represent the set which has the coordinates of a descendants at position at $(\mathrm{n}_1, \mathrm{n}_2)$ less than the offspring or we can say that $\mathbf{L(n_1, n_2)}$is the dissimilaritiy between $\mathrm{D}(\mathrm{n}_1, \mathrm{n}_2)$ and $\mathrm{O}(\mathrm{n}_1, \mathrm{n}_2)$.It may be represented as .

$$\mathrm{L}(\mathrm{n}_1, \mathrm{n}_2) = \mathrm{D}(\mathrm{n}_1, \mathrm{n}_2) - \mathrm{O}(\mathrm{n}_1, \mathrm{n}_2) \tag{4.9}$$

4) **H:** It is comprised of all the roots of special orientation tree which belongs to highest level pyramid (i.e LL Sub-band).

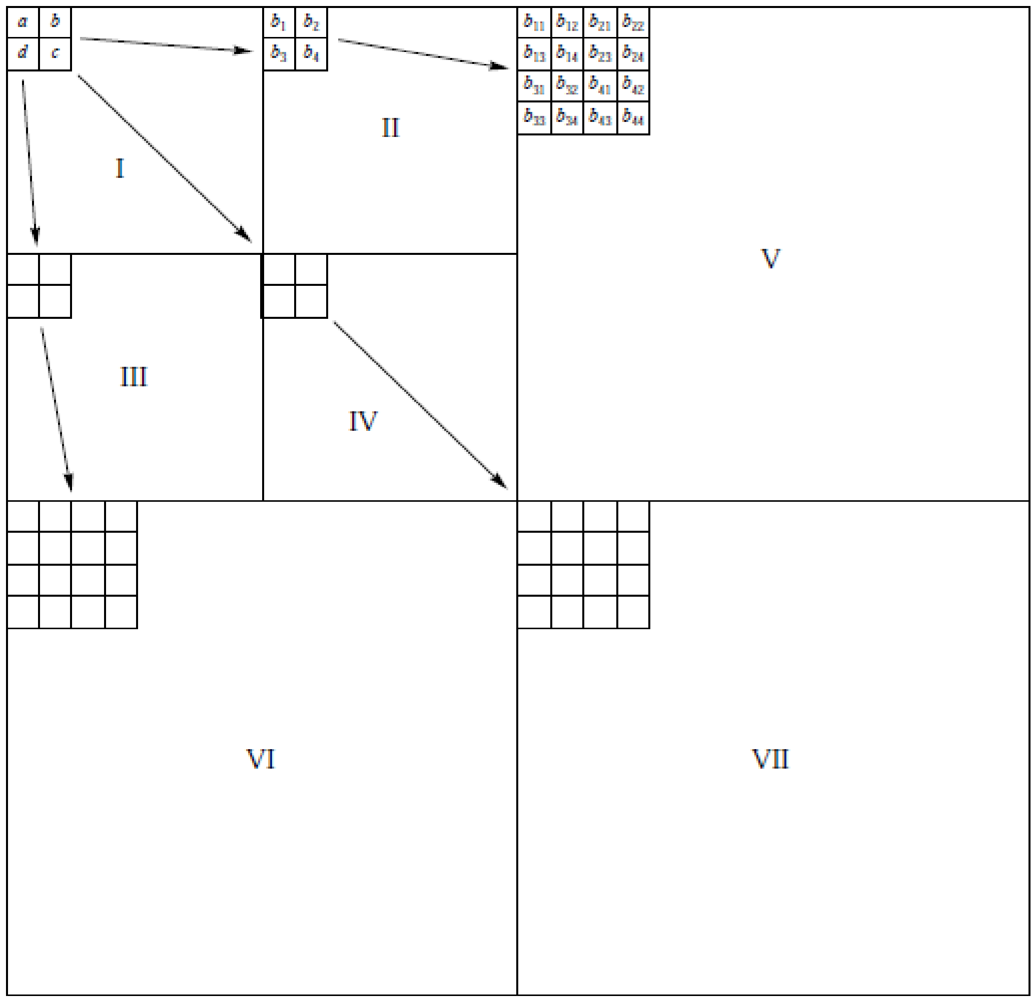


Figure 4.2: Data structure of SPIHT used in the algorithm

4.6 SPIHT Encoding and Decoding

As name states rule of set partitioning technique is observed by SPIHT. The good thing is that at both encoder and decoder ends similar algorithm is being run. Moreover, ordering information is not sent explicitly like other algorithm of embedded transmission which makes SPIHT algorithm more efficient than the others as similar algorithm for both encoder and decoder. The set of lists which are continuously maintained / updated during the process are as under:-

1) LIP- List of insignificant pixels.
2) LSP- List of significant pixels.
3) LIS- List of insignificant sets.

The position of coordinates is the mean of identification for every element.LIP and LIS contains the elements which are individual pixels whereas, LIS has the sets of either as $L(n_1, n_2)$ orset $D(n_1, n_2)$.

From the outset every pixel and set is considered insignificant. 'n' is found with the help of the coefficients' ceiling magnitude. The number of fundamental passes are three these are : the sorting pass, the refinement pass and the upgrading of quantization step pass.

As long as the encoder transmits the least significant bit, the recursive process of these passes continues and keeps repeating itself iteratively. It is during the sorting pass that insignificant pixels of LIP are ascertained whether they are significant or otherwise. If found significant, then they are passed to the LSP. Similarly, significance of the sets of LIS is ascertained and those found significant are partitioned. These are also taken out from the LIS. Subsets having multiple entries are placed in the LIS. On the same lines single pixels are adjusted in the LSP or LIP keeping in view their status. With this preview encoding algorithm may be summarized in four steps as under:-

4.6.1 Step-1: Initialization

- Output $n = \lfloor \log_2 (\max_{(x_1, x)} \{|c_{x_1, x_2}|\}) \rfloor$
- Set the LSP = $\{\phi\}$
- Set the LIP = $\{(x_1, x_2) \in H\}$and LIS= $\{D(x_1, x_2), (x_1, x_2) \in H\}$

4.6.2 Step-2: The Sorting Pass

1) Each and every element of the LIP is checked for the significance. Give the output '1' or '0' to each entry depending upon whether it is found as significant or insignificant respectively. If significant then discard it from the LIP and place it in the LSP.

2) Now the significance of each set of LIS should be checked. Output its sign as significance if it is found to be significant. By using rule 2 or 3, this element is partitioned as per the set if it is $L(x_1, x_2)$ or $D(x_1, x_2)$. All the three lists i.e

LIP, LSP and LIS be upgraded throughout the process depending upon the significance.

4.6.3 Step-3: The Refinement Pass

In refinement pass, other than the elements which were having the same ‘x’ in the process of sorting pass and have been moved to LSP, the most significant bit from all the elements of the LSP be placed at the x^{th} position.

4.6.4 Step-4: Renewing Quantization Step Pass

In this step keep decreasing ‘x’ by 1 and keep repeating the steps of sorting pass and refinement pass until x=0.

The decoder follows the exactly the same steps as those of encoder. The output generated by the encoder becomes the input to the decoder.

4.7 Example 4.1

In this example I have applied the SPIHT algorithm, for one pass only, on the same example as was taken in case of EZW so that the comparison between the two techniques can be drawn. For better assimilation DWT matrix of an 8x8 image has been shown in table 4.2 at the cost of repetition. The results of application of one pass of SPIHT algorithm on table 4.2 have been shown in the table 4.3. Table 4.3 is clearly indicating about the data coded and updating of the control lists. It can be observed that after one pass SPIHT algorithm has used 29 bits without using any other kind of entropy encoding which are 4 bits less as compared to EZW.

	0	1	2	3	4	5	6	7
0	63	−34	49	10	7	13	−12	7
1	−31	23	14	−13	3	4	6	−1
2	15	14	3	−12	5	−7	3	9
3	−9	−7	−14	8	4	−2	3	2
4	−5	9	−1	47	4	6	−2	2
5	3	0	−3	2	3	−2	0	4
6	2	−3	6	−4	3	6	3	6
7	5	11	5	6	0	3	−4	4

Table 4.2: Set of Image Wavelet Coefficients used by example. The numbers outside the box indicate the set of co-ordinates used.

Comm.	Pixel or Set Tested	Output Bit	Action	Control Lists
(1)				LIS = {(0,1)A,(1,0)A,(1,1)A} LIP = {(0,0),(0,1),(1,0),(1,1)} LSP = ∅
(2)	(0, 0)	1+	(0,0) to LSP	LIP = {(0,1),(1,0),(1,1)} LSP = {(0,0)}
	(0, 1)	1–	(0,1) to LSP	LIP = {(1,0),(1,1)} LSP = {(0,0),(0,1)}
	(1, 0)	0	none	
	(1, 1)	0	none	
(3)	$\mathcal{D}(0,1)$	1	test offspring	LIS = {**(0,1)A**,(1,0)A,(1,1)A}
	(0, 2)	1+	(0,2) to LSP	LSP = {(0,0),(0,1),(0,2)}
	(0, 3)	0	(0,3) to LIP	LIP = {(1,0),(1,1),(0,3)}
	(1, 2)	0	(1,2) to LIP	LIP = {(1,0),(1,1),(0,3),(1,2)}
	(1, 3)	0	(1,3) to LIP	LIP = {(1,0),(1,1),(0,3),(1,2),(1,3)}
(4)			type changes	LIS = {(1,0)A,(1,1)A,(0,1)B}
(5)	$\mathcal{D}(1,0)$	1	test offspring	LIS = {**(1,0)A**,(1,1)A,(0,1)B}
	(2, 0)	0	(2,0) to LIP	LIP = {(1,0),(1,1),(0,3),(1,2),(1,3),(2,0)}
	(2, 1)	0	(2,1) to LIP	LIP = {(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1)}
	(3, 0)	0	(3,0) to LIP	LIP = {(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0)}
	(3, 1)	0	(3,1) to LIP	LIP = {(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1)}
			type changes	LIS = {(1,1)A,(0,1)B,(1,0)B}
(6)	$\mathcal{D}(1,1)$	0	none	LIS = {**(1,1)A**,(0,1)B,(1,0)B}
(7)	$\mathcal{L}(0,1)$	0	none	LIS = {(1,1)A,**(0,1)B**,(1,0)B}
(8)	$\mathcal{L}(1,0)$	1	add new sets	LIS = {(1,1)A,(0,1)B,(2,0)A,(2,1)A,(3,0)A,(3,1)A}
(9)	$\mathcal{D}(2,0)$	0	none	LIS = {(1,1)A,(0,1)B,**(2,0)A**,(2,1)A,(3,0)A,(3,1)A}
(10)	$\mathcal{D}(2,1)$	1	test offspring	LIS = {(1,1)A,(0,1)B,(2,0)A,**(2,1)A**,(3,0)A,(3,1)A}
	(4, 2)	0	(4,2) to LIP	LIP = {(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2)}
	(4, 3)	1+	(4,3) to LSP	LSP = {(0,0),(0,1),(0,2),(4,3)}
	(5, 2)	0	(5,2) to LIP	LIP = {(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2),(5,2)}
	(5, 3)	0	(5,3) to LIP	LIP = {(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2),(5,2),(5,3)}
(11)			(2,1) removed	LIS = {(1,1)A,(0,1)B,(2,0)A,(3,0)A,(3,1)A}
(12)	$\mathcal{D}(3,0)$	0	none	LIS = {(1,1)A,(0,1)B,(2,0)A,**(3,0)A**,(3,1)A}
	$\mathcal{D}(3,1)$	0	none	LIS = {(1,1)A,(0,1)B,(2,0)A,(3,0)A,**(3,1)A**}
(13)				LIS = {(1,1)A,(0,1)B,(2,0)A,(3,0)A,(3,1)A} LIP = {(1,0),(1,1),(0,3),(1,2),(1,3),(2,0),(2,1),(3,0),(3,1),(4,2),(5,2),(5,3)} LSP = {(0,0),(0,1),(0,2),(4,3)}

Table 4.3: SPHIT Process.

Chapter 5

Embedded Block Coding with Optimized Truncation

5.1 Introduction

Embedded Block Coding with Optimized Truncation (EBCOT) algorithm in various degrees, is related to Shapiro's EZW (Embedded Zero-Tree Wavelet compression) algorithm, Said and Pearlman's SPIHT (Spatial Partitioning of Images into Hierarchical Trees) algorithm and Taubman and Zakhor's LZC (Layered Zero Coding) algorithm [[i]]. It is the core of the new image compression standard JPEG 2000. EBCOT and wavelet transforms are the most complex and computationally expensive parts of the standard.

5.2 EBCOT Fundamentals

EBCOT uses independent block based fractional bit-plane coding on quantized indices after wavelet transform. Each coefficient is visited three times. A separate bit stream is generated for each block without using any information from the other blocks. These bit-streams can be truncated to a number of lengths (truncation points).

These lengths are decided by a Post Compression Rate Distortion Optimization Algorithm (PCRD-OPT) which uses Lagrange optimization and convex hull calculations. It selects the share of each block bit-stream in final bit-stream by keeping in view the target bit rate or allowed distortion. Different quality layers are formed by including truncated bit streams from each block or selected blocks. These quality layers provide distortion scalability (image with multiple qualities). Generally these steps are portioned into two tiers; fractional bit plan coding and PCRD-OPT processes comprise tier 1 and layering is classified as tier 2 of EBCOT. Fractional bit plan coding is the most complex part of the algorithm. In the proceeding part of the chapter we will focus on it in detail.

EBCOT has some distinguishing features that made it most suitable for coding algorithm of JPEG 2000. It has superior low bit-rate performance in terms of visual quality and PSNR (below 0.25 bit/pixel) compared to the baseline JPEG. Random access and compressed domain processing are supported in EBCOT. Randomly code blocks can be extracted from the compressed bit stream (region of interest is supported by this feature). Functions of compressed-domain processing include cropping, flipping, rotation, translation, scaling, feature extraction, etc. These two features enable object-based functionality. Error resilience is achieved by small independent code blocks. Errors in one block do not affect other blocks.

5.3 Independent Block Coding

In EBCOT quantized indices of each sub band are partitioned into small code blocks (normally 32x32 or 64x64). Each block is coded independently producing a separate bit stream.

Figure5.1 shows blocks with their elementary bit streams. We can define different points (truncation points) in the elementary bit stream of a block where we can truncate it depending on the allowed bit rate or distortion. Each truncation point adds more bits (length L) from elementary to final bit stream. Final bit stream is formulated by contribution of bit stream from each block or some to the blocks.

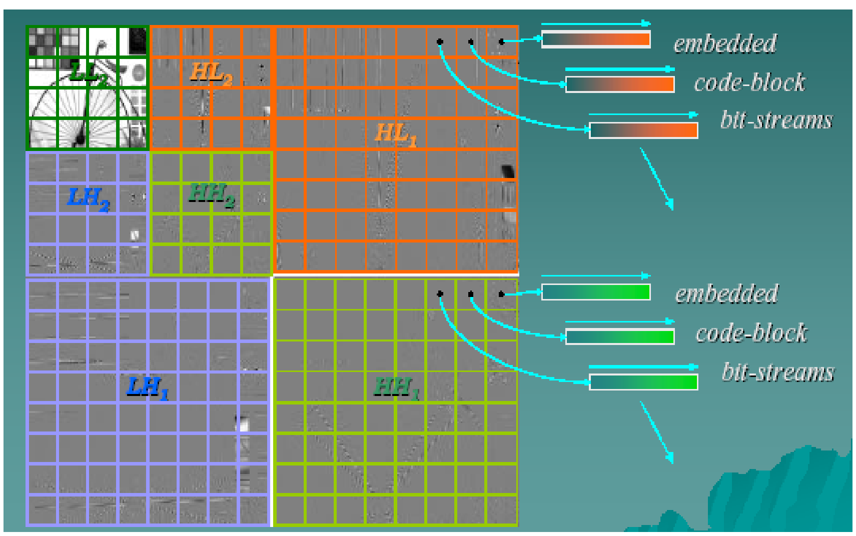


Figure 5.1: Blocking with Independent Bit streams

The net distortion of the reconstructed image is sum of distortions from each code block. Calculation of the block distortion depends on sub band. The final bit stream is constrained by maximum length allowed. The selection of truncation points should decrease distortion. This can only be calculated when all blocks are coded in full and their truncation points with associated distortions are known. The algorithm used for optimal selection of truncation points is known as Post Compression Rate Distortion Optimization (PCRD-Opt). It has a potential disadvantage of extra coding of bits which are not used. This algorithm is thoroughly studied in literature and proved by many practitioners [38].

Another shortcoming of independent block coding is to unable to exploit inter-block redundancy within sub band and inter-sub band. Also the redundancy in parent child samples as used in EZW and SPIHT is not coded. These disadvantages are compensated by optimized contributions of each code block independently to the final bit stream.

5.4 Bit Plane Coding

In bit plane coding (BPC) samples are coded bit plane by bit plane with most significant bits (MSB) of all quantized coefficients (most significant bit plane) are coded first then next level bits (MSB-1) then next (MSB-2) until least significant bit plane comes or required compression rate achieved. Each bit plane is partitioned into strips of 4 rows and same no of columns as the block (see Figure 5.2).

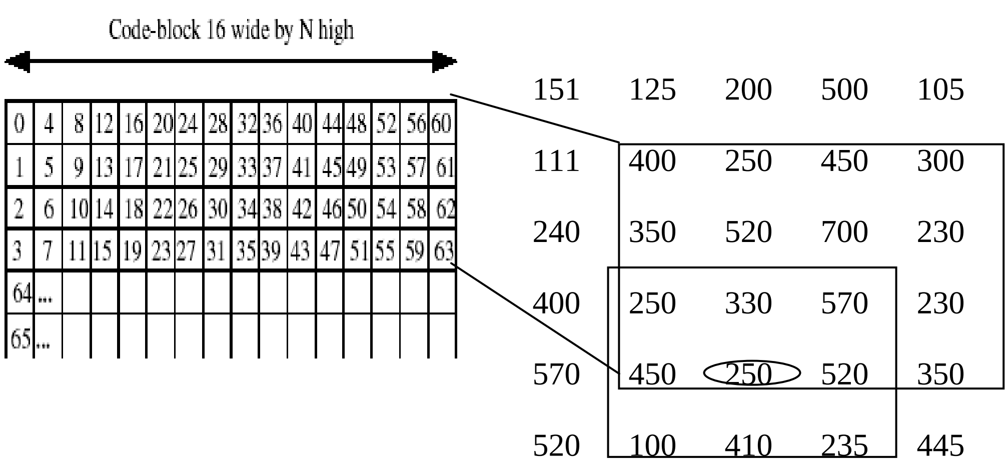


Figure 5.2 : Scan Pattern and Formation of Context in EBCOT

After complete scan of a strip, control shifts to the next consecutive strip. In the left part of the figure sequence of samples being coded is shown whereas right part shows context window (8 connected neighbors) of a sample being coded.

5.4.1 Primitive Coding Operations

Four primitive coding operations are used to generate the output of BPC. The output of BPC is a string of context (CX) and decision (D) pairs. CX is a value from 0-18 (different implementations may use different range for contexts) while D is a binary value of 0 or 1.

If a sample (single bit in a bit plane) has value 1 then it is said to be significant. If significance of a sample from bit plane p at location i, j is represented as sig(i, j) then it can be defined as in equation 5.1.

$$\begin{aligned} sig(i, j) &= 1 \quad \text{if } magnitude(i, j) > 0 \\ sig(i, j) &= 0 \quad \text{if } magnitude(i, j) = 0 \end{aligned} \qquad \textbf{(5.1)}$$

Significance state sig(i, j) is initialized to zero for each every block. The four coding operations, Significance (zero) coding, Run length coding (RLC), Sign coding and Magnitude Refinement coding (MRC) employed by EBCOT depend on significance state of a sample. RLC is used when entire strip (4 bits in a column) has insignificant samples with entirely insignificant neighbors.

For a sample s at location i, j; if sig(i, j) = 0 and mag(i, j) = 1 then "significance coding" (also known as zero coding) along with RLC (Run length coding) is used where mag(i, j) means magnitude of sample s at location i, j. After this operation sig(i, j) becomes 1. Sign coding is used in conjunction with significance coding to code sign of a nonzero element.

The significance of a sample is based on the significance of its eight connected neighbors (see Figure 5.3: Significance Coding Context Figure 5.3). There are nine possible states of interest in a sample context. These states are represented by nine context values (0-8). Theses state combinations are different for different sub-bands. Table 5.1 shows this context assignment to different states. This also depends on three intermediate values given in equation 5.2.

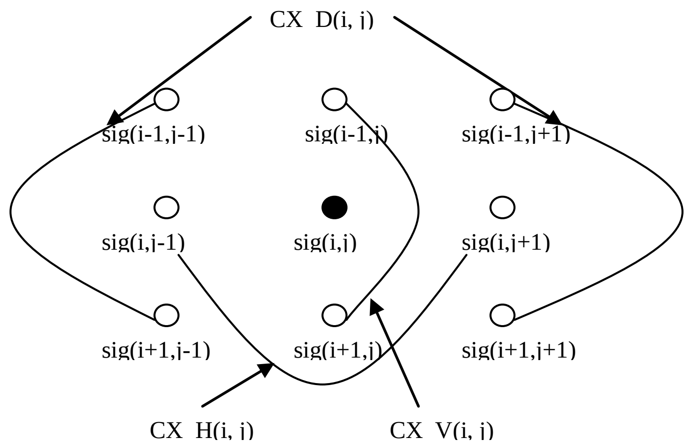


Figure 5.3: Significance Coding Context

The context assignment is based on sub band. In LH(vertically high) sub band significant sample arise from horizontal features for the image and vice versa in HL (horizontally high pass) whereas in HH diagonal regions are more significant. These context design rules are based on empirical studies.

CX_H (i, j) = sig(i, j-1) + sig(i, j+1) two horizontal neighbors **(5.2)**

CX_V(i, j) = sig(i-1, j) + sig(i+1, j) two vertical neighbors

CX_D(i, j) = Sum of 4 diagonal neighbors

LL and LH blocks (X means don't care condition)			
CX (i, j)	**CX_H(i, j)**	**CX_V(i, j)**	**CX_D(i, j)**
8	2	X	X
7	1	>=1	X
6	1	0	>=1
5	1	0	0

4	0	2	X
3	0	1	X
2	0	0	>=2
1	0	0	1
0	0	0	0
For HL sub band			
CX (i, j)	**CX_H(i, j)**	**CX_V(i, j)**	**CX_D(i, j)**
8	X	2	X
7	>=1	1	X
6	0	1	>=1
5	0	1	0
4	2	0	X
3	1	0	X
2	0	0	>=2
1	0	0	1
0	0	0	0
For HH sub band			
CX (i, j)	**CX_H(i, j)**	**CX_V(i, j)**	**CX_D(i, j)**
8	X		>=3
7	>=1		2
6	0		2
5	>=2		1
4	1		1

3	0	1
2	>=2	0
1	1	0
0	0	0

Table 5.1: Context Labels for Significance Coding

In sign coding normally sign is treated as independent random variable and separate 1 bit is used to represent it in most of the coding algorithms. The signs of neighboring samples exhibit statistical redundancy. In EBCOT 5 context labels are used to code sign of a sample.

Magnitude refinement coding (MRC) codes next magnitude bit, i.e., the bit in next bit plane (sig(i, j) for p+1 where p is current bit plane). A state variable is used to note whether MRC is applied to a sample or not. Three context values are used to label MRC. Simple scheme for this assignment is: If MRC is applied on a sample and then context label (CX) = 16; If MRC is not applied on current sample and sum of significance of 8-connected neighbors is greater than 0 then CX=15; And if MRC is not applied on current sample and significance of neighbors <= 0, then CX = 14 might be used.

5.4.2 Coding Passes

Rate-distortion performance of bit plane coding (with truncation) can be improved by first coding those bits which result in largest reduction in distortion relative to the increase in code length. This observation results in notion of fractional bit plane coding. EBCOT uses three coding passes – significance propagation pass (SPP), magnitude refinement pass (MRP) and clean up pass (CUP) to code each bit plane, with a part of the bit-plane being coded in one of the pass. The first bit-plane is only coded in CUP whereas remaining is coded from SPP, MRP and CUP. Selection of pass to encode current sample depends on the criteria given in Table 5.2.

Pass 1 (SPP)	If sig(i, j) is not equal to 1 and at least one neighbor is significant
Pass 2 (MRP)	If sig(i, j) = 1
Pass 3 (CUP)	If sig(i, j) is NOT 1 and significance of Neighbors is NOT 1

Table 5.2: Selection of Pass to Encode Current Sample

In each bit plane Significance Propagation Pass (SPP) is the first pass . If it happens to be a insignificant sample but one of its eight neighbors is significant then s is coded in this pass. Using our previous notations it can be expressed as sig (i, j) = 0 and CX(i, j) > 0.. Only significance coding primitive is used in this pass followed by sign coding. Once a sample becomes significant its four connected neighbors which are not visited yet have a significant neighborhood, so it is called “significance propagation pass”.

$$Z = 3n - 2 \qquad \textbf{(5.3)}$$

Magnitude refinement coding primitive is used in this second pass. It codes only those bits which are significant and are not coded in previous pass (SPP). Cleanup Pass coded the earlier two passes. Run-length coding is also incorporated to help in coding a string of zeros.

CBS(MSBP)_p3, CBS(MSBP-1)_p1, CBS(MSBP-1)_p2, CBS(MSBP-1)_p3,
………………, CBS(0)_p1, CBS(0)_p2, CBS(0)_p3 **(5.4)**

In EBCOT compressed bit stream (CBS) is formed with concatenated bit stream from each pass with coded bits from most significant bit-plane to coded bits from least significant bit plane. This provides a natural set of truncation point at the end of bit stream from each pass. If n passes are applied on a block then number of truncation points Z given in equation 5.3. Equation 5.4 presents a compressed bit stream for EBCOT which makes bit-stream embedded because it can be truncated after any pass. Most of the information lies in significant bit planes; lower order bit planes add details to the image and improve quality.

5.5 Conclusion

EBCOT coding methodology is discussed in this chapter. Concepts like independent block coding and bit plane coding provide the basic building blocks for the algorithm understanding. Three coding primitives’ significance coding, sign coding and magnitude refinement coding are elaborated in next section. Three coding passes Significance propagation pass, magnitude refinement pass and clean up pass make the next section.

Chapter 6

Results and Simulations

6.1 Introduction

The wavelet transforms based embedded coding techniques discussed in the previous chapters has its own inherent advantages and draw-backs. To sift out their distinct features, there is a need to carry out a comparative analysis so as to come up with best compression solution compatible to given situation and requirement. The three techniques use DWT which is preferred over DCT. The reason behind this can be, DCT is defined as the block based transform partition and it partitions image into overlapping blocks and then process each block separately, at very low bit rate. The transformed coefficients need to be coarsely quantized and so there will be significant reconstruction error after decoding [30]. The artifact known as blocking artifact caused by the discontinuity in the image is more visible at the block boundaries. The best way to reduce this errors allow basis functions to decay towards zero at the block artifacts points or to overlap transform [32], wavelet transform is a special of this type transform.

This chapter gave the brief description on the comparison between these three wavelet based embedded compression technique. In this portion, a comparison is drawn by using different parameters like Peak signal to Noise Ratio (PSNR) and bit saving capacity i.e. Compression Ratio(CR).

PSNR is used to compare the amount of distortion, that can be done been done by using

$$\text{PSNR}= 10\ \log10_{10}(\frac{(max\ (E(i,j)))^2}{MSE})$$

PSNR is based on the MSE that indicates the comparison of mean squared error of an image transmitted and the one reconstructed at decoder end. MSE can be calculated mathematically as under

$$\text{MSE} = \sum_{\text{IJ}} \frac{\text{g(i, j)} - \tilde{\text{g}}\text{(i, j)}}{\text{I x J}}$$

More over the Bit Saving Capacity can be calculated by making use of Compression Ratio.

$$\text{Bit Saving Capacity} = 1 - \frac{1}{\text{CR}}$$

Whereas CR can be calculated by

$$\text{CR} = 1 - \frac{\text{Compressed}}{\text{uncompressed}}$$

6.2 Analysis of Important Features and Draw-back of EZW

6.2.1 Important Features of EZW

Zero-tree and successive approximation combined to form a new compression technique to give efficient compression to the wavelet coefficients as well as to obtain spatial and SNR scalability.

Some interesting features of EZW which make it significant are enlisted below:-

1) The similarities are exploited in the bands those which have same orientation in the zero-trees (ZTs) structure with the help of which the numbers of symbols, required to be coded, are reduced.

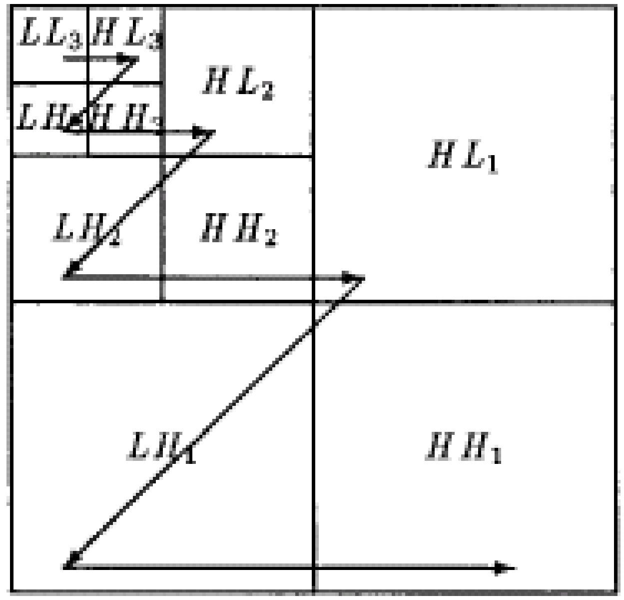


Figure 6.1: ZT-structure of EZW

2) The adaptive arithmetic coding, that was a part of EZW structure in the beginning, is very efficient because of using a small alphabets to represent the image. The reason behind this is because if there is any change in the Statistics of the symbols, it adopts the changes itself quickly.
3) The current threshold bounds the maximum distortion level of coefficients therefore, for every pass it is the current threshold that gives the average distortion level which is same for all the bands.
4) As defined by EZW algorithm the coefficients having magnitude greater than the threshold will be given the non-zero values. This implies that the coefficient with higher magnitude will be coded first given the priority over the coefficients with lower magnitude. This information from EZW shows that it accords importance to the most significant information in the coding process. So the bits are best used at given bit rate as per the requirements.
5) The successive approximation quantization process is demonstrated in EZW. It helps the addition of a symbol to the bit string such as + ve, -ve, ZT and Z, that helps to further refinement of reconstructed image. The encoding and decoding can be halted at any interval depending upon the distortion and bit rate. The symbol is being added to the string so that it becomes the part of bit stream. Corresponding to the symbol that are encoded and decoded, image can be

recovered up to the level of refinement corresponding to the symbols. Therefore as the bit-rate budget is exhausted the encoding and decoding of the image can be stopped that's what makes the EZW extremely precise bit-rate control. Moreover as the priority is given to the most important information so if the decoding of the bit-stream is stopped it will not matter and the best possible image quality for that bit rate can be achieved.

6) In addition to above features, one of the most important that was introduced in EZW is spatial and SNR scalability. In this scheme different scanning methods are used to achieve both of them. Spatial scalability can be achieved by scanning the wavelet coefficients sub-band by sub-band from the lowest to the highest frequency sub-bands.

   On the other hand PSNR scalability is achieved by scanning the wavelet coefficients in every tree from the peak to base.

6.2.2 Draw-backs of EZW

EZW is an efficient technique with comparatively better compression performance than previously known compression techniques. However, still there were certain grey areas, as in the transmission of coefficients position was not well defined. EZW didn't cater for inter-band correlation; it correlates intra-bands only. Moreover, the performance with single embedded file was not much pronounced.

6.3 Analysis of Important feature and Limitation of SPIHT

SPHIT introduced by Said and Pearlman [36] outperforms EZW even without arithmetic coding. It has also addressed few of the issues discussed above being faced by EZW.

6.3.1 Important feature of SPIHT

In SPIHT the wavelets coefficients subsets are partitioned and significant information is obtained. The results of comparison of branching point of the SPIHT algorithm define the execution path; this is the main feature of transmitting the ordering data so that the sorting algorithm is similar for the encoder and decoder. In this way if the decoder receives the magnitude comparison then encoder's execution path can be duplicated by it so that execution path can recover the ordering information. Some main features of SPIHT coding are:

1) SPIHT gives good quality reconstructed images, high PSNR, especially for color images.
2) It is optimized for progressive image transmission, producing a fully embedded coded file.
3) It's quantization algorithm is simple.
4) The encoding and decoding of SPIHT is nearly symmetric.

6.3.2 Similarities between EZW and SPIHT

SPIHT has brought in a different concept of coding according to which wavelets coefficients can be encoded by using successive approximation quantization. It has outperformed EZW even without incorporating arithmetic coding. One of the similarities between SPIHT and EZW that they make use of encoding which is based on the concept of spatial tree technique. In it, correlation of magnitude is exploited across the bands of decomposition. By encoding a fidelity bit steam which is progressive in nature is generated by each. To efficiently isolate and encode larger magnitude coefficient a significance test on sets of coefficients is used.

6.3.3 Differences between EZW and SPIHT

1)There is a contrast in spatial orientation tree (SOT) of EZW and SPIHT, In EZW top band (LL band) has three offspring one in each high-frequency sub-band at the same decomposition level, all other coefficients have four children in the lower decomposition sub-band at the same orientation. On the hand SPIHT, 2X2 root nodes in top LL bands, top left nodes has no descendant and other three have four offspring each in high-frequency band of the corresponding as shown in figure 2.

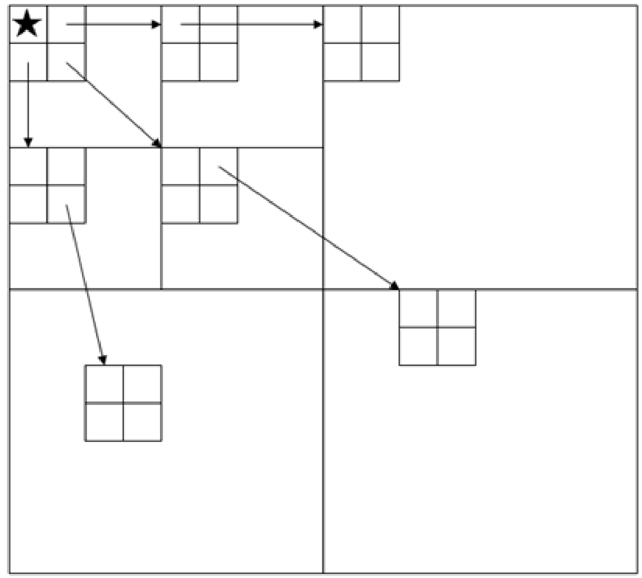


Figure 6.2: Set Orientation Tree of SPIHT

Thus SPIHT use less no of trees with more elements per tree than in EZW.

2) There is an additional step in SPIHT according to which four individual child are created by splitting a descendant (Type-A) set and grant descendant set (type-B). Moving from coarser to finer sub-bands, a breadth first search of hierarchical trees is performed by EZW explicitly. A roughly breadth first search is also performed by SPIHT though it in not explicit. LIS contains four new descendants which are placed by SPIHT after partitioning a grand descendant set. It is the appending to the LIS those results in the approximate breadth first traversal.

6.4 Analysis of Important Features and Limitation of EBCOT

6.4.1 Important Feature of EBCOT

While creating a bit-stream with a rich set of features as in SNR and resolution scalability along with a "random access property" an advanced compression performance is exhibited by EBCOT algorithm. Without substantial decrease in compression performance, all these features coexist in a single-bit stream. Sub bands coefficients are made by the EBCOT algorithm by making use of wavelet transform which are later on quantized and coded. Albeit, commonly used dyadic wavelet decomposition is utilized more often than not, yet other packet decomposition are also occasionally supported and proffered. The original image is considered in the form of a sub-band collation that may be set into growing resolution levels. The smallest level of resolution consists of the single LL sub- band. There are additional sub-bands in each successive resolution level which are needed to build the image with twice the vertical and horizontal resolution. EBOCT algorithm being a scalable kind of compression mechanism offers an advantage that there is no need to know the reconstruction resolution or the target bit rate at the time of reconstruction. Another benefit of profound importance is that unlike existing JEPEG compression technique, there is no need to compress the image multiple times to achieve a target bit rate. Small blocks of samples are created by EBCOT by dividing each sub-bands and then produces a bit stream that is separate and highly scalable to exhibit each code block. EBOCOT algorithm is modest in complexity and is well suited to applications which incorporate remote browsing of large compressed images. It makes use of 46 x 64 size code blocks with 16x16 size sub-blocks. In EBCOT bit stream is consists of a collection of quality layers and PSNR scalability is obtained by removing undesired layers. Superior interpretation of texture and significantly less ringing around the edges are shown by the EBCOT images. However through simulations it has been observed that some details which are totally lost in SPIHT can be well preserved in EBCOT. So to this end it has been seen that the EBCOT performance has been competitive with the present state of the art compression methods and it has outperformed the SPIHT to a considerable extent.

6.4.2 Comparison of EBCOT with EZW and SPIHT

EBCOT and EZW have almost similar kind of progressive coding strategy but their data structures are different. EZW is capable of compressing an image with excellent rate time distortion performance and encoding for all rates can precisely be controlled with optimal performance. EZW utilizes the fact that some wavelet presents the same location in the image though they are in different sub bands. In EBCOT, since each block is coded independently, it is suitable for parallel hardware coding and decoding. Unlike EZW and SPIHT that uses the zero-tree structure, the similarities between sub-bands are not exploited in EBCOT. But the more efficient context-based arithmetic coding and the post-compression rate distortion optimization are used. EBCOT avoids the tree data structure, but uses a data structure which is a quad-tree [35].

We have 2x2 set of blocks of the coefficients at the lowest level which are in turn ordered into sets of 2x2 quads and so on. A node in a tree will be significant at a level n if at that level any of its descendants are significant. A coefficient c_{ij} is said to be significant at level n if $|c_{ij}| \geq 2^n$. The algorithm of EBCOT, like that of EZW and SPIHT, also makes multiple passes as in significant map encoding pass and refinement pass. Arithmetic coding is used to encode the bits generated during this procedure[38].

An entropy-coded bit-stream is extremely sensitive to channel errors. In EBCOT, each code block is independently encoded, and the individual quality packets can be independently decoded, and thus effects of the channel errors are confined to a small area. In contrast, due to the zero-tree structure together with entropy coding, a single bit error may destroy a whole picture encoded by EZW or SPIHT. EZW coding exploits the inter-band dependencies of the wavelet coefficients, whereas in EBCOT coding the inter-band dependencies is not exploited.

Spatial and SNR scalability of images are among the many requirements from JPEG2000 standard. In EZW and SPIHT is SNR scalable, which is done by successive approximation or bit-plane coding. The bit-stream of EZW and SPIHT is SNR scalable only.

ZT-structure used in the EZW and SPIHT involves downward dependencies achieved by the successive decomposition of wavelets on the sub-bands. Due to these downward dependencies error propagates through the band.

This problem can be sort-out by coding the sub-band in a block-code independently. Which is more flexible, each block code has similarity within blocks, but independent with the different blocks. The blocks size is depend upon the distortion-rate to what extent the coding efficiency can be sacrifice, parallel processing is possible with the embedded block coding, multiple blocks can be coded and decoded simultaneously. Due to the flexibility of rearrangement of bit-streams. In EBCOT both spatial and SNR scalability is possible, error in one block will not affect the other block so robustness improved. Compression performance is having deficiency as the similarity between sub-bands is not exploited. This can be solved by post compression rate distortion (PCRD).

6.4.3 Gray area of EBCOT

The number of layers in EBCOT increases performance of the coding decreases. Moreover its suits the application involving remote browsing of large compressed images.

6.5 Results and Simulations

6.5.1 Results Comparison between EZW and SPIHT

In this part, attributes of simulation and inferred results of the simulation have been included to have a comparison of the three embedded coding techniques. I have used MATLAB 7.11.0 (R2014b) and calculated some performance measures like PSNR value, Bit saving capacity and elapsed time for algorithm. Tables 6.1-6.14 exhibit the results using same 8x8 DWT matrix that has been used in examples to illustrate EZW and SPIHT in the previous chapters. Tables 6.1-6.14 enunciate a no of performance measure like PSNR, bit saving capacity and elapsed time. Efforts have been made to show the graphical representation of the results in Figure 6.1

Bit Rate	**PSNR**	**Bit Saving Capacity**	**Elapse time**
0.0625	16.0922	0.2727	1.937534
0.1	16.2522	0.2507	1.999975
0.125	18.9465	0.2212	1.602176
0.25	19.0465	0.2201	1.597808
0.375	22.0183	0.0330	1.904488
.5	24.5878	0.0424	2.498722
.625	25.7093	0.0309	3.490076
.75	26.7093	0.0312	3.493514

.875	26.9434	0.0103	4.788779
1	27.8614	0.0030	6.426141

Table 6.1: Performance of EZW for Lena 128x128

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	18.3333	0.2157	6.738255
0.1	18.9133	0.2112	6.354275
0.125	21.0026	0.0294	8.295849
0.25	22.0026	0.0394	8.935108
0.375	23.5146	0.0691	9.888978
.5	26.2212	0.0425	14.967418
.625	27.6810	0.0470	24.254871
.75	27.6810	0.0470	23.552671
.875	27.8907	0.0320	37.074200
1	27.8558	0.0208	54.267545

Table 6.2: Performance of EZW for Lena 256x256

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	20.5475	0.0453	55.356887
0.1	21.5475	0.0553	36.648125
0.125	22.8248	0.0653	57.714014

0.25	22.8248	0.0653	54.762184
0.375	25.5438	0.0596	102.073352
.5	28.4667	0.0486	194.473277
.625	30.4102	0.0386	347.725214
.75	30.9102	0.0396	331.911903
.875	31.3067	0.0318	629.387061
1	31.3617	0.0365	1079.343058

Table 63: Performance of EZW for Lena 512x512

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	15.0089	0.1714	1.350978
0.1	16.0089	0.1824	1.659237
0.125	17.5562	0.0775	1.757398
0.25	18.8562	0.0742	1.796083
0.375	20.5518	0.0792	2.176902
.5	22.7974	0.0660	3.014438
.625	23.8255	0.0275	4.243626
.75	24.7255	0.0275	4.541441
.875	23.8152	0.0116	5.925847
1	23.7457	0.0043	8.121057

Table 6.4: Performance of EZW for Barbara 128x128

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	16.7938	0.0569	9.304642
0.1	17.7938	0.0679	9.789559
0.125	19.2888	0.0761	9.418228
0.25	20.2888	0.0821	10.320276
0.375	22.0375	0.0722	11.704219
.5	24.4294	0.0507	18.069329
.625	24.8599	0.0492	30.032173
.75	24.9599	0.0362	33.552018
.875	25.0787	0.0201	53.576583
1	25.0245	0.0104	81.843499

Table 6.5: Performance of EZW for Barbara 256x256

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	17.3791	0.0657	39.781169
0.1	18.4791	0.0673	42.967542
0.125	19.0773	0.0689	65.834382
0.25	19.7573	0.0369	66.610547
0.375	20.3013	0.1405	147.583429
.5	20.6504	0.0712	359.142647
.625	20.7494	0.0328	639.607126

.75	21.6394	0.0298	595.639464
.875	21.7965	0.0151	889.613307
1	22.7845	0.0047	999.241632

Table 6.6: Performance of EZW for Barbara 512x512

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	16.2074	0.0750	2.625688
0.1	17.2482	0.0882	1.651509
0.125	18.5324	0.0935	1.480082
0.25	19.0637	0.1008	1.689000
0.375	22.5522	0.0865	2.079960
.5	24.8873	0.0540	2.656129
.625	26.0845	0.0233	3.475520
.75	26.0655	0.0228	3.526251
.875	26.3613	0.0217	4.872582
1	26.3513	0.0023	6.065420

Table 6.7: Performance of EZW for Obama 128x128

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	18.5684	0.0901	6.978849
0.1	18.9644	0.0894	7.178823

0.125	18.5684	0.0873	7.374846
0.25	21.8243	0.0855	8.191558
0.375	25.2587	0.0653	10.759329
.5	28.5346	0.0368	15.083447
.625	30.6117	0.0154	21.066015
.75	30.9123	0.0198	20.840841
.875	31.5947	0.0245	30.126190
1	31.9009	0.0131	41.282123

Table 6.8: Performance of EZW for Obama 256x256

Bit Rate	**PSNR**	**Bit Saving Capacity**	**Elapse time**
0.0625	21.4589	0.0629	36.690096
0.1	22.6784	0.0629	35.141296
0.125	24.9262	0.0617	50.788133
0.25	28.6839	0.0679	80.866452
0.375	30.7345	0.0543	120.854532
.5	32.5629	0.0489	133.852233
.625	35.4020	0.0165	191.403938
.75	37.5110	0.0214	298.858808
.875	38.3679	0.0056	444.827015
1	39.4673	0.0041	958.346532

Table 6.9: Performance of EZW for Obama 512x512

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	20.1231	0.0245	0.949339
0.1	21.4876	0.0354	0.680644
0.125	22.5118	0.0432	0.652056
0.25	24.6078	0.0374	1.193544
0.375	26.2666	0.0391	1.703823
.5	27.9533	0.0329	2.387230
.625	29.2112	0.0351	2.836960
.75	30.3505	0.0397	3.301959
.875	31.2874	0.0355	3.868087
1	32.2590	0.0335	4.393763

Table 6.10: Performance of SPIHT for Lena 128x128

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	22.9790	0.0393	1.811724
0.1	24.3707	0.0438	1.947220
0.125	25.2263	0.0348	2.607012
0.25	27.8733	0.0407	4.630700
0.375	29.7893	0.0378	7.044084
.5	31.7402	0.0361	9.500859
.625	32.7609	0.0306	11.577292
.75	34.0505	0.0314	13.886916

.875	35.2069	0.0293	16.087196
1	36.1021	0.0275	18.464599

Table 6.11: Performance of SPIHT for Lena 256x256

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	26.4007	0.0423	5.571137
0.1	28.1304	0.0431	8.100437
0.125	29.1026	0.0425	9.646198
0.25	32.0069	0.0381	19.817739
0.375	33.8474	0.0316	32.345064
.5	35.4990	0.0305	44.308956
.625	36.4088	0.0248	54.976277
.75	37.3706	0.0193	71.311724
.875	38.1187	0.0226	89.128423
1	38.9583	0.0213	117.509010

Table 6.12: Performance of SPIHT for Lena 512x512

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	17.8084	0.0389	0.909845
0.1	18.8559	0.0336	0.700206
0.125	19.5199	0.0357	0.822328

0.25	21.8878	0.0396	1.282473
0.375	23.5351	0.0422	1.760517
.5	25.0262	0.0415	2.293588
.625	26.2205	0.0309	2.865506
.75	27.6089	0.0324	3.430575
.875	28.8726	0.0309	4.602290
1	29.7633	0.0325	4.598936

Table 6.13: Performance of SPIHT for Barbara 128x128

Bit Rate	**PSNR**	**Bit Saving Capacity**	**Elapse time**
0.0625	20.3359	0.0296	1.847549
0.1	21.7433	0.0391	2.146391
0.125	22.5713	0.0387	2.602938
0.25	25.0247	0.0403	4.942821
0.375	26.4118	0.0315	7.182017
.5	27.6835	0.0312	9.711893
.625	29.0118	0.0289	12.612310
.75	30.0442	0.0280	15.432655
.875	31.0409	0.0221	17.475625
1	32.0401	0.0221	21.768642

Table 6.14: Performance of SPIHT for Barbara 256x256

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	20.2957	0.0393	5.850970
0.1	21.1965	0.0363	8.257450
0.125	21.6925	0.0401	10.039333
0.25	24.0293	0.0213	21.491511
0.375	26.0871	0.0239	33.790723
.5	27.6717	0.0172	46.533642
.625	29.2643	0.0211	56.912755
.75	30.5804	0.0194	71.052532
.875	31.7058	0.0147	90.636404
1	32.7821	0.0153	109.103444

Table 6.15: Performance of SPIHT for Barbara 512x512

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	19.7110	0.0459	0.917165
0.1	21.3688	0.0269	0.598972
0.125	22.1376	0.0376	0.757006
0.25	25.6474	0.0437	1.230643
0.375	27.8854	0.0484	1.746531
.5	29.8691	0.0485	2.368685
.625	31.5827	0.0444	2.939001
.75	32.8666	0.0447	3.596977

.875	34.4429	0.0448	4.000348
1	35.9826	0.0384	4.452730

Table 6.16: Performance of SPIHT for Obama 128x128

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	24.4724	0.0376	1.850808
0.1	26.5090	0.0443	2.215765
0.125	28.0697	0.0376	2.597376
0.25	32.7178	0.0507	5.113187
0.375	35.5752	0.0445	7.281796
.5	38.4374	0.0410	10.330521
.625	39.8249	0.0399	12.013635
.75	41.4305	0.0385	14.797350
.875	42.5652	0.0351	17.206571
1	43.5483	0.0338	19.888620

Table 6.17: Performance of SPIHT for Obama 256x256

Bit Rate	PSNR	Bit Saving Capacity	Elapse time
0.0625	31.4874	0.0420	5.962657
0.1	34.6612	0.0497	8.248002
0.125	36.6074	0.0483	10.240082
0.25	41.6707	0.0425	20.445356

0.375	44.2468	0.0397	30.597680
.5	45.5279	0.0361	42.215400
.625	46.2951	0.0321	56.027172
.75	46.7690	0.0316	67.742651
.875	47.1246	0.0291	81.145248
1	47.3297	0.0251	98.475749

Table 6.18: Performance of SPIHT for Obama 512x512

6.5.2 Results Comparison between EZW, SPIHT and EBCOT

To compare the above discussed encoding techniques, test images of Lena and Barbara of size 512 X 512 each have been processed through their encoding algorithms. Their results have been shown in Tab 5.15 and 5.16. Graphical representation of the same has also been exhibited in Figure 5.11 and 12

It has been observed from Tab 1 and 2 that with increase in bit rate EBCOT provides better PSNR as compare to those of EZW and SPIHT. However PSNR performance of SPIHT is comparatively better than EZW.EBCOT result is obtained by the Daubechies 9/7 bi-orthogonal wavelet filters with five level transform, code-block is size 64X64 with sub-blocks of size 16X16 are used.

bpp	**EZW**	**SPIHT**	**EBCOT**
0.0625	27.54	28.3	28.3
0.125	30.23	31.1	31.22
0.25	33.17	34.11	34.28
0.5	36.28	37.21	37.28
1	39.55	40.41	40.61

Table 6.19: Performance of EZW,SPIHT and EBCOT for Lena 512 x 512

bpp	**EZW**	**SPIHT**	**EBCOT**
0.0625	23.1	23.35	23.45
0.125	24.03	24.86	25.55
0.25	26.77	27.58	28.55
0.5	30.53	31.39	32.48
1	35.14	36.41	37.37

Table 6.20: **Performance of EZW,SPIHT and EBCOT for Barbara 512 x 512**

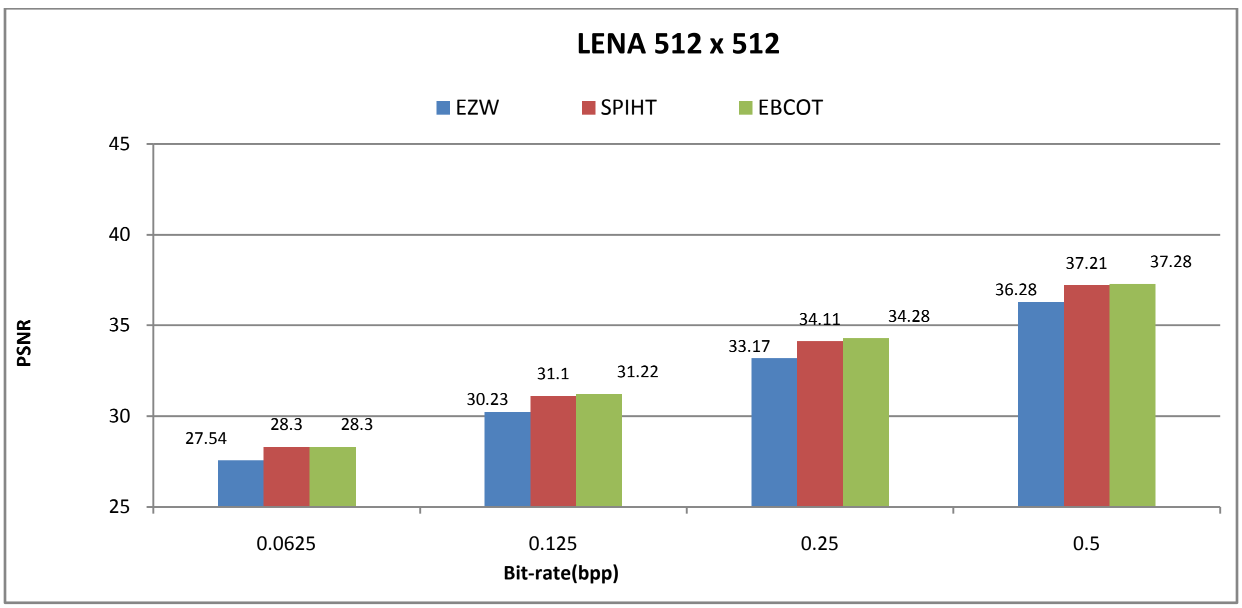


Table 6.21: Graphical representation for the performance of Lena 512x512 for EZW,SPIHT and EBCOT

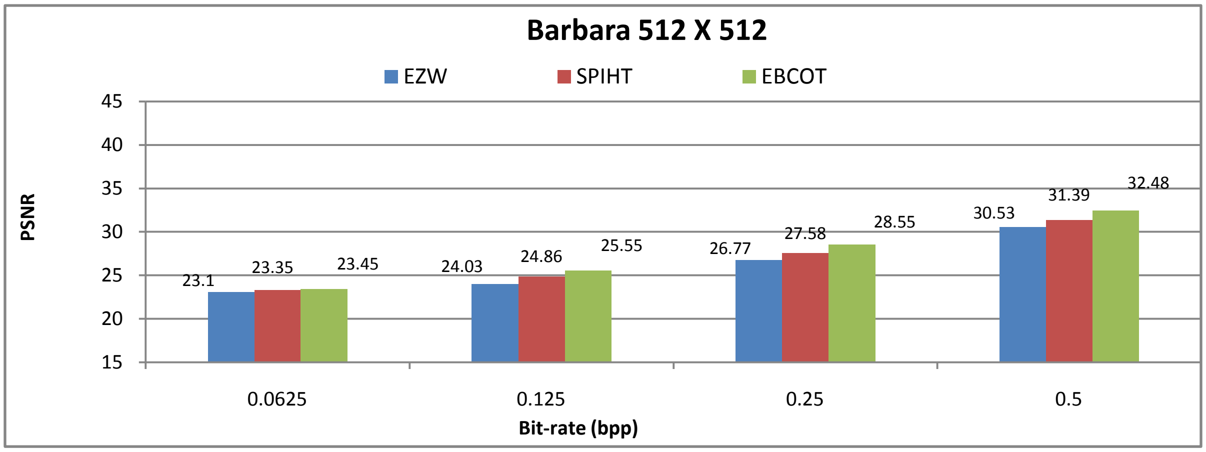


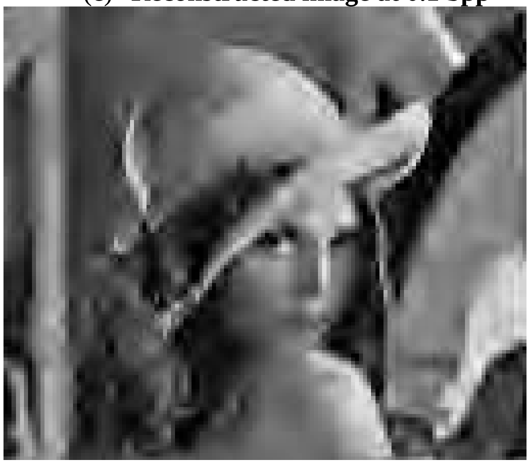
Table 6.22: Graphical representation for the performance of Barbara 512x512 for EZW,SPIHT and EBCOT

(a) **Original image of 512x512**

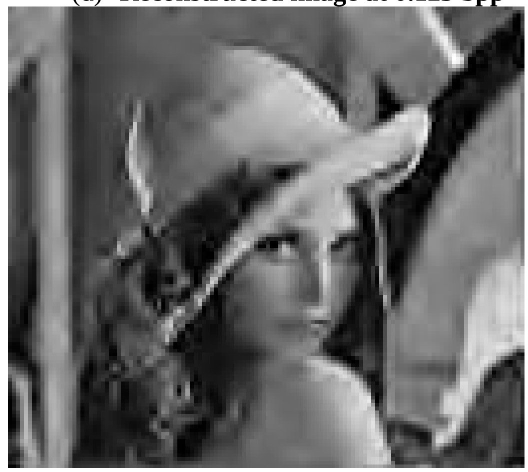
(b) **Reconstructed image at 0.0625 bpp**

(c) **Reconstructed image at 0.1 bpp**

(d) **Reconstructed image at 0.125 bpp**



(e) **Reconstructed image at 0.25 bpp**



(f) **Reconstructed image at 0.375 bpp**

(g) **Reconstructed image at 0.5 bpp**

(h) **Reconstructed image at 0.625 bpp**

(i) **Reconstructed image at 0.75 bpp**

(j) **Reconstructed image at 0.875 bpp**

(k) Reconstructed image at 1 bpp

Figure 6.3: Image compression using proposed algorithm on Lena image of size 128 x 128 using various rates.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Wavelet transform is considered as landmark for image compression because signal is represented by functions which are localized both domains of frequency and time .Wavelet sub-band coding exploits The self-similarity of pixels is exploited by Wavelet sub-band coding and exhibit resulting coefficients in different sub-bands. Different embedded codec algorithms such as EZW, SPIHT and EBCOT is widely used for the compression of wavelet transformed images. Having compared three encoding techniques it has been observed that EZW was an efficient encoding technique of its time. Albeit, it proffered better compression than its ancestors yet it was unable to give pronounced performance with single embedded file and could not cater for the inter-band similarity among the pixels. SPIHT on the other hand addressed the shortcomings of EZW and brought improvements in peak signal to noise ratio. However, edges lost by SPIHT were aptly handled by EBCOT which not only gave better PSNR but also resolution scalability. Same is evident from numerical results and plots that EBCOT encoding scheme has superseded those of EZW and SPIHT to-date. It has also been seen that EBCOT is much similar to the SPIHT as both are wavelet based algorithms however; SPIHT correlates the sub-bands while EBCOT encodes sub-bands individually

7.2 Future work

There search work investigates on three embedded coding techniques; which carried out with the comprehensive comparison of these three techniques. however there are different approaches to improve them. Future researchers are advocated to take on task of reduction of bits along with improvements in the picture quality that can be done by optimizing the PSNR and MSE.

REFERENCES

[1] T. Brahimi, A. Melit, F. Khelifi and D. Boutana, "Improvements to SPIHT for Lossless Image Coding," 2006 2nd International Conference on Information & Communication Technologies, Damascus, 2006, pp. 1445-1450

[2] Singh, G., Koranga P., and Verma, D., "Analysis of EZW and SPIHT algorithms for compression of an image", International Journal on Emerging Technologies, pp. 671-675, 2017.

[3] Shapiro, Jerome M. "Embedded image coding using zerotrees of wavelet coefficients." IEEE Transactions on signal processing 41, no. 12 (1993): 3445-3462.

[4] Amir Said, Member IEEE and William A. Pearlman, Senior Member, IEEE,"A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", IEEE Transaction on circuits and system for video technology, Vol 6,No 3, June 1996 (pp. 234-250)

[5] Ghanbari, M., "Standard Codecs Image Compression to Advanced Video Coding", British library, London institute of information and technology, Chapter 4, pp. 80-97, 2011.

[6] David Taubman,"High performance Scalable Image Compression with EBCOT", IEEE transaction on image processing, Vol. 9, No 7, July 2000 (pp. 1158-1170)

[7] J. H. Zhao, W. J. Sun, Z. Meng, Z. H. Hao, "Wavelet transform characteristics and compression coding of remote sensing images," Optics and Precision Engineering, Vol. 12(2), pp.205-210, 2004.

[8] H. L. Xu, S. H. Zhong, "Image Compression Algorithm of SPIHT Based on Block-Tree," Journal of Hunan Institute of Engineering, Vol. 19(1), pp.58-61, 2009.

[9] R.C Gonzalez & R.E Wood, "Wavelets and Multi-resolution Processing, Digital Image Processing", 2nd edition, Chapter 7, 2002

[10] Marc Antonini, Michel Barlaud, Member, Pierre Mathieu, and Ingrid Daubechies, "Image Coding Using Wavelet Transform", IEEE Transactions on Image Processing, vol. i, no 2. April 1992.

[11] Anoop Mathew, Member, IEEE and Bensiker Raja Singh D, "Image Compression using Lifting Based DWT," International Journal of Computers, Information Technology and Engineering January-June 2009.

[12] B. Yan, H. Zhang, “SPIHT Algorithm and its Improvement,” Computer Applications and Software, vol. 25(8), pp.245-247, 2008.

[13] Jose Oliver “A Fast Run-Length Algorithm for Wavelet Image Coding with Reduced Memory Usage” M.P. Malumbres Department of Computer Engineering (DISCA), Technical University of Valencia Camino de Vera 17, 46017, Spain.

[14] Nicholas Kolokotronis, AlikiVassilarakou, Sergios Theodoridis , Dionisis Cavouras “Wavelet-based medical image compression Eleftherios Kofidis” Department of Informatics, Division of Communications and Signal Processing, University of Athens, Panepistimioupolis, TYPA Buildings, GR-15784 Athens, Greece .

[15] Rosenfeld, Azriel, ed. Multiresolution image processing and analysis. Vol. 12.Springer Science & Business Media, 2013.

[16] James S. Walke, “A Primer on Wavelets and Their Scientific Applications”, Page No. 130.1999

[17] Introduction to the Discrete Wavelet Transform (DWT), 15,Feb,2004

[18] V.S.Shingate, T.R.Sontakke and S.N.Talbar “Still Image Compression using Embedded Zerotree Wavelet Encoding”

[19] Raid, A. M., W. M. Khedr, M. A. El-dosuky, and Wesam Ahmed. "Image Compression Using EZW." Signal & Image Processing 5, no. 6 (2014): 33.

[20] Shapiro, Jerome M. "Embedded image coding using zerotrees of wavelet coefficients." IEEE Transactions on signal processing 41, no. 12 (1993): 3445-3462.

[21] Li, Mei-shan, Yue Liu, and Hong Zhang. "Wavelet Transform-based Remote Sensing Image Compression." (2015).

[22] Tian, Jun, and Raymond O. Wells Jr. "Embedded image coding using wavelet difference reduction." In Wavelet image and video compression, pp. 289-301.Springer US, 2002.

[23] Shapiro, Jerome M. "Digital data compression system including zero-tree coefficient coding." U.S. Patent 5,315,670, issued May 24, 1994.

[24] Rogers, Jon K., and Pamela C. Cosman. "Wavelet zero-tree image compression with packetization." IEEE Signal Processing Letters 5, no. 5 (1998): 105-107.

[25] Cotronei, Mariantonia, Damiana Lazzaro, Laura B. Montefusco, and Luigia Puccio. "Image compression through embedded multiwavelet transform coding." IEEE Transactions on Image Processing 9, no. 2 (2000): 184-189.

[26] Zandi, Ahmad, James D. Allen, Edward L. Schwartz, and Martin Boliek. "CREW: Compression with reversible embedded wavelets." In Data Compression Conference, 1995.DCC'95. Proceedings, pp. 212-221. IEEE, 1995.

[27] Li, Jin, Po-Yuen Cheng, and C-C. Jay Kuo. "Improvements of embedded zerotree wavelet (EZW) coding."In Visual Communications and Image Processing'95, pp. 1490-1501.International Society for Optics and Photonics, 1995.

[28] Schwartz, Edward L., Ahmad Zandi, and Tibor Boros. "Method and apparatus for spatially embedded coding." U.S. Patent 5,815,097, issued September 29, 1998.

[29] Oktem, Levent, Rusen Oktem, Karen Eguiazarian, and Jaakko Astola. "Efficient encoding of the significance maps in wavelet based image compression." In Circuits and Systems, 2000.Proceedings.ISCAS 2000 Geneva. The 2000 IEEE International Symposium on, vol. 3, pp. 25-28. IEEE, 2000.

[30] Islam, Asad, and William A. Pearlman. "Embedded and efficient low-complexity hierarchical image coder."In Electronic Imaging'99, pp. 294-305.International Society for Optics and Photonics, 1998.

[31] Da Silva, Eduardo AB, Demetrios G. Sampson, and Mohammed Ghanbari. "A successive approximation vector quantizer for wavelet transform image coding." IEEE Transactions on Image Processing 5, no. 2 (1996): 299-310.

[32] Sriram, Parhasarathy, and Michael W. Marcellin. "Image coding using wavelet transforms and entropy-constrained trellis-coded quantization." IEEE Transactions on Image Processing 4, no. 6 (1995): 725-733.

[33] Algazi, V.R and R.R. Estes, Analysis based coding of image transform and sub band coefficients. Proceedings of SPIE, Vol 2564 (1995), p. 11-21.

[34] Basavanthaswami, V., and Somasekhar, T., “Image compression using SPIHT”, International Journal of Innovative Research in Computer and Communication Engineering, Volume5, No.2, pp.1839-1844, 2017.

[35] Dodla, S., Raju, Y.D.S., and Mohan, K.V.M., “Image compression using wavelet and SPIHT Encoding Scheme”, International Journal of Engineering Trends and Technology, Volume 4, No. 9, pp.3863-3865, 2013.

[36] Amir Said, Member IEEE and William A. Pearlman, Senior Member, IEEE,”A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical

Trees", IEEE Transaction on circuits and system for video technology, Vol 6,No 3, June 1996 (pp. 234-250)

[37] Wireless communication system, from RF subsystem to 4G enabling technologies by Ke-Lin Du, M.N.S. Swamy. Chapter. No 14 "image and video coding" page no. 727.

[38] Introduction to Data Compression by Khalid Sayood. Chapter No 15, "wavelet-based compression" page no. 512.