ENTROPY CODING BASED IMAGE COMPRESSION USING SET PARTITIONING IN HIERARCHICAL TREES (SPIHT)

By

Anam Hijab Mohsin

SUBMITTED TO THE FACULTY OF ELECTRICAL ENGINEERING

MILITARY COLLEGE OF SIGNALS,

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,

RAWALPINDI, PAKISTAN

July 2021

In The name of Allah the most Beneficent and the most Merciful.

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by Ms **Anam Hijab Mohsin,** Registration No. **00000205568** of **Military College of Signals** has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the student have been also incorporated in the said thesis.

Signature: ___________________________

Name of Supervisor: Asst Prof. Lt. Col (R) Imran Touqir, PhD

Date: _______________________________

Signature (HoD): ______________________

Date: _______________________________

Signature (Dean): _____________________

Date: _______________________________

DEDICATION

Dedicated to my family who has always been a source of motivation for me and also to my supervisor Dr Imran Touqir who had been a great source of help throughout this period. I oblige to him for his support.

ABSTRACT

Images have become a vital part of our daily life. With the increase of images in use the storage and transmission has come into consideration. In the era where image compression was in consideration, emergence of Wavelet Transform (WT) has made it a lot easier as it represents a signal in terms of functions localized in both frequency and time domain, there are many wavelet transform based Image compression methods with progressive coding schemes. With the Embedded Zero Tree Wavelet (EZW) massive improvement was witnessed in image compression. EZW and SPIHT are used to attain better PSNR and compression ratios.

EZW works on DWT to predict the absence of significant information by exploiting self-similarities across the scale. However, coefficient position is missing, didn't cater for intra-band correlation and its performance with single embedded file was not much pronounced. And we had to share bit plane with it to the decoder. The improvements in EZW were brought in with the introduction of SPIHT, which is again a fully embedded codec algorithm. It uses principal of partial ordering by magnitude, set partitioning by importance of magnitude of the coefficients, self-similarity across the scale and ordered bit plan transmission.

SPIHT encodes the transformed coefficients according to their significance. Statistical analysis has shown that the output bit-stream of SPIHT comprises of long series of zeroes which can be further compressed, therefore SPIHT cannot be used as sole mean of compression. To this end, additional compression is being done by making use of different kinds of entropy encoding schemes. One of the entropy encoding scheme which is concatenated with SPIHT for further compression is Huffman encoding.

This research is motivated by the requirement of a viable solution for fast transmission and less storage space. This research concentrates on saving

comparatively more number of bits without compromising the quality of the image by combining two encodings “Set Partitioning in Hierarchical Tree and Huffman coding. This is done by making deft use Huffman encoding where it yields the optimized results and saves more numbers of bits thereby reducing the storage space and increasing the transmission time.

CERTIFICATE

It is to certify that final copy of thesis written by **Miss Anam Hijab Mohsin** of Military College of Signals has been evaluated by me, found complete in all respect as per the specified format of NUST Statutes / Regulations, is free of plagiarism, errors and mistakes.

Dated: _______________ _______________________

Col Dr Imran Tauqir

ACKNOWLEDGEMENTS

I am thankful to Allah Almighty for helping me achieve another milestone in my academics. Secondly, I want to thank Military College of Signals, NUST and faculty, for giving me a research environment at the institute. This would have not possible without the support of my supervisor Assoc Prof. Dr. Imran Tauqir who not only provided timely guidance, profound encouragement and positive criticism but also ensured that I complete the assigned tasks in time frame. His kind consideration helped me during all this time. I m also very obliged to my committee members, Brig. Dr. Adil Masood and Dr. M Imran for their association in my work.

This would’ve not been possible without the support and prayers of my Parents, Husband and family.

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**
	Positive	**Pos**
	Negative	**Neg**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Background:

In present era, huge amount of information is stored, processed and transmitted digitally thereby necessitates different 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.

Images are the most important form of information. In representation of digital data, new researches have been made as use of images has increased in many fields. Hence the need to reduce storage in compliance with bandwidth requirements has increased. Process of Image compression is converting the larger image size to smaller, keeping picture quality in mind or Information of digital image to lesser bits. In transmitting and receiving the data memory is saved. Hence, compression of binary data and image compression is a lot more different as image compression done using binary data compression will not yield good results in terms of compression ratio and other parameters. So there was a need of image compression techniques.

There are few cases when a few parameters of image compression can be ignored and a few information is lost during the image compression, those methods are known as lossy image compressions. Lossless image compression techniques are used where image quality is important.

In order to meet the requirements, for a last past couple of decades, many wavelet based efficient image compression methods with progressive bit stream output have come up as an efficient solution [1]. With the emergence of Embedded Zero Tree Wavelet (EZW) Encoding algorithm WT has gone through a lot of improvement. Search of improvements in EZW led to another better encoding technique “Set Partitioning in Hierarchical Tree” (SPIHT).

SPIHT encoding has yielded efficient compression performance than EZW. This has been done by making use of property of self-similarity of the coefficients. The output of SPIHT once further fed to entropy coding, it paves the way towards better results. To this end, several entropy encoding schemes like Huffman, Arithmetic and Lempel Ziv Welch have remained prime considerations, both for the lossless and lossy compression, for the researcher.

Although, the researchers have proposed a variety of combinations to obtain efficient compression results yet the room for improvements still exists. There is a possibility that entropy coding, once used by utilizing its best possible way, may produce more efficient compression by saving extra number of bits than before without compromising on the quality of original image and its basic facet like peak signal to noise ratio (PSNR). The purpose of this research is directed to work on image compression that uses set partitioning in hierarchical tree(SPIHT) along with entropy encoding with a view to bring some improvements in image storage capacity and transmission time by saving more number of bits.

1.2 Research motivation

As our most of daily life information is based on images and by the invention of smart devices image usage has increased, This put some extra constraint on bandwidth and storage space because extra bits are required to sustain the image quality. To the problem stated above, there is a dire need of tackling the above highlighted issues of bandwidth and storage space.

An image of size 1024 x 1024 is nearly equal to 3 MB and can be compressed to 300 KB by using proper compression technique.

This research concentrates on saving comparatively more number of bits without compromising the quality of the image by combining two encodings "Set Partitioning in Hierarchical Tree and Entropy coding". Hence, this research is motivated by the requirement of a viable solution for fast transmission and less storage space.

1.3 Objective:

Study of discreet wavelet transformation (DWT) and embedded entropy coding technique. Analysis of SPIHT cascaded with Huffman entropy coding to save more number of bits.

1.4 Principle of Image Compression

If images are observed, correlation of images can be seen between neighboring pixels. Due to this correlation information is almost negligible in those pixels. In image compression we remove data but not on the cost of quality of pixels. An uncorrelated set of data is formed using pixels of images before storing or transmitting. At receiver's side we get approximated/real image after decompression.

Redundancy has different types. A few are listed below.

1.4.1 Spatial redundancy is correlation among same objects and space. It can be divided in further two categories.

1.4.1.1 Inter-Pixel Redundancy*:* 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 whereas 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.

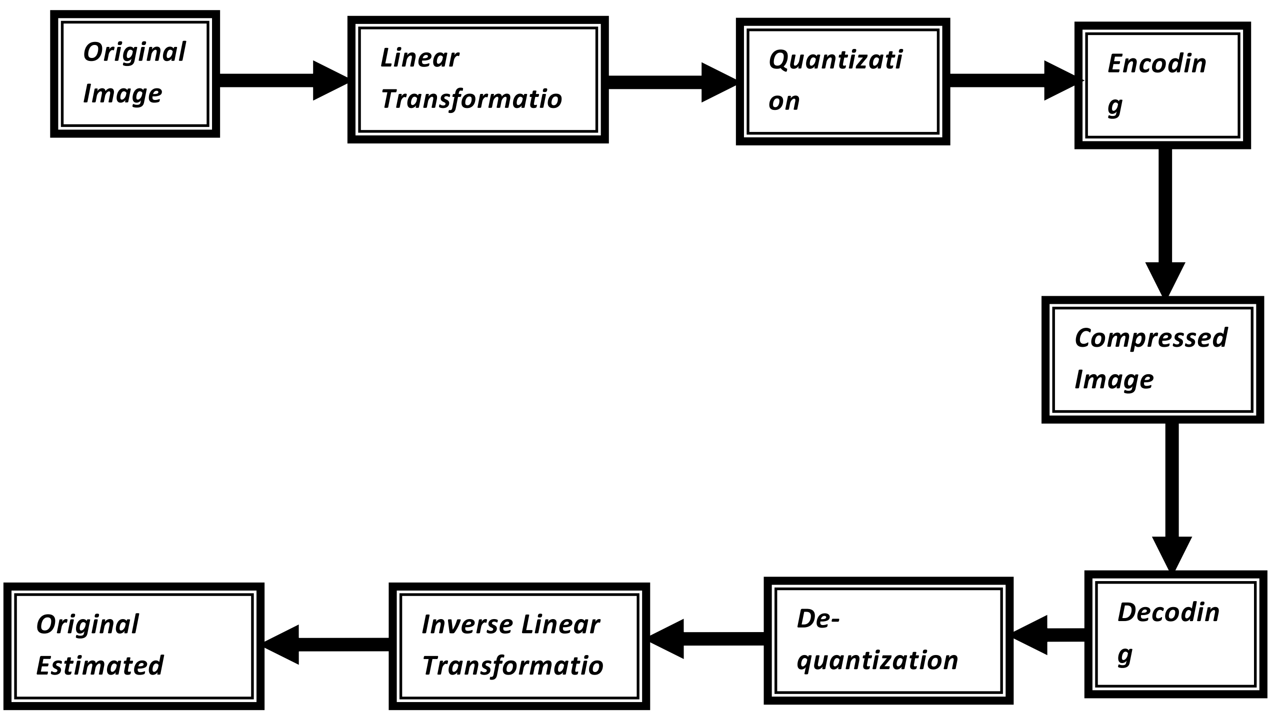
1.4.1.2 Coding Redundancy: It is based on the principle that some pixel values are more common than others. It can also be related to the representation of information which has been expressed in the form of codes. Gray levels of an image are allocated more than the required number of code symbols which causes coding redundancy.

1.4.2 Spectral redundancy is correlation among numerous bands of spectrum.

1.4.3 Temporal redundancy is correlation among consecutive objects or frame of image.

1.5 Image Compression Process:

Process of Image compression is based on three tiers. First of all, self-similarity of pixels are exploited using transformation/ predictive coding or 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 lossless 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.



1.1 Process of image compression

1.6 Organization of thesis

This research work has seven chapters. Chapter 1 is about the research incorporating background, motivation and contributions. 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 enunciates about entropy encoding and its apt utilization for optimal results that also includes the proposed methodology in detail and finally in chapter 7 the results along with future work have been presented.

2 Wavelet Transform

2.1 Introduction

A transformation method has taken place of sinusoidal transformation techniques like 'Discrete Fourier Transform' DFT and 'Discrete Cosine Transform' DCT for performing better on lower bit rates. Wavelet part was introduced here, that support in time and frequency duration. Wavelets enable us to observe the image at different frequency and time level i.e. different resolutions as these can be scaled and shifted. i.e. Multi Resolution Analysis (MRA). In addition to this wavelet analysis is a tool to analyze frequency at different spatial locations also known as 'Space Frequency Localization', this property was absent in the previous sinusoidal transformation techniques.

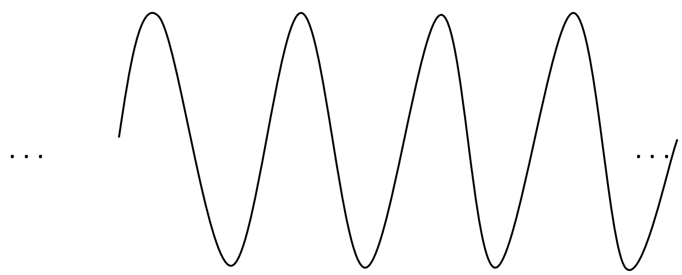
In image pixels are located in two dimensions, using sinusoidal we can't get both the information at the same time. Wavelet transform caters this issue and provides effective MRA and coding capability. It is a powerful mean for representing [2].

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 a coherent relationship with the upcoming chapters. [3]

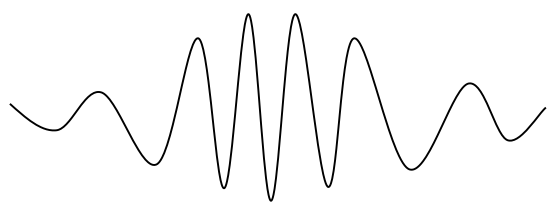
This characteristic of wavelets distinguishes it from the others. Moreover, wavelet coefficients also help in plotting the exact position of the discontinuity in time domain.

2.2 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



(a) Sine Wave



(b) wavelet

2.3 Importance and benefits of wavelets

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. An image is basically a matrix of color intensity also known as pixels, ranging from 0-255. To analyze small objects, higher resolution is needed.

Concept of mother wavelet W(t) is employed in Wavelet Transform (WT) to represent any wavelet by scaling and translation in W(2kt-m). Where k is the scaling factor. With the increase in scaling factor wavelet gets narrower.

Same time there is requirement when a signal needs information both in frequency and time domain for its apt utilization. Although, 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. 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.

2.4 Basics of Wavelets

Concept of mother wavelet W(t) is employed in Wavelet Transform (WT) to represent any wavelet by scaling and translation in W(2kt-m). Where W(t) is a representation of a wave for t=0 till t=T, whereas k scales and m shifts. In W(2kt-m) is which exists from t=m to t=m+T and contract by factor With the increase in scaling factor wavelet gets narrower. [4]. Higher frequency wavelet corresponds to narrow width and lower frequency corresponds to wider width[5].

2.5 Wavelet Transform

Wavelets are formed by the frequency components of the Data. Each component represents a specific resolution. [6]. The core idea of this is that main wavelet can be represented by scaled and shifted version of mother wavelets. Fig 2.1

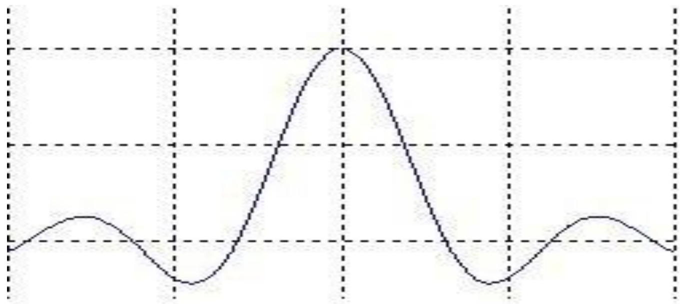
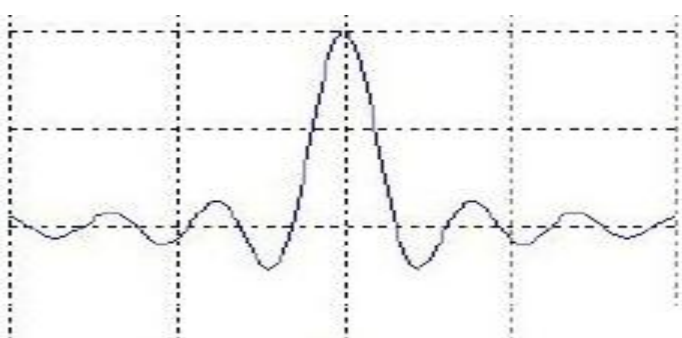
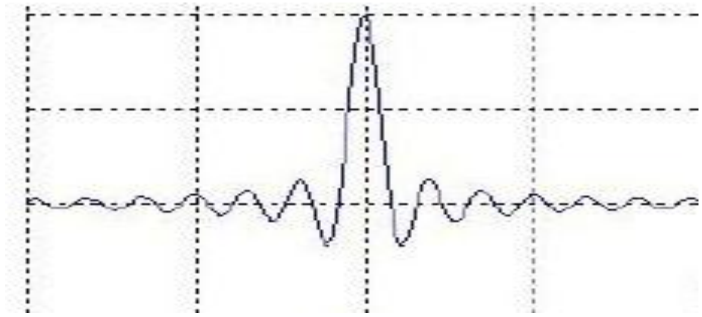


Figure-2.1: Mother wavelet w(t)

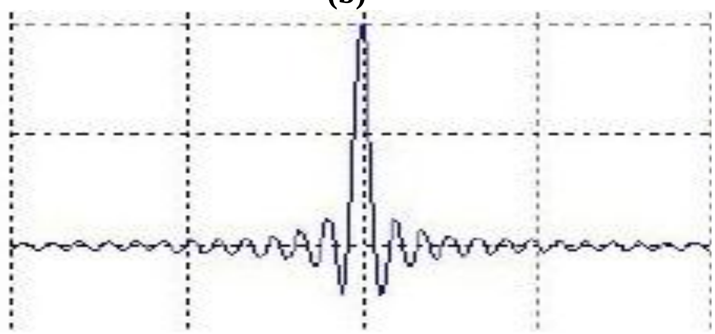
Wavelets are obtained by using mother wavelet. Whereas the mother wavelet w(t) is a short interval wave from time t=0 to t=T. Similarly, wavelet that starts at t=m and terminates at t m+T is w (t-m) and w (2kT) is a scaled version obtained by starting mother wavelet at T=0 and terminating at t=T/2k as shown in figure 2.2. Wavelet gets narrower with the increase in the scaling factor and increase in frequency. Moreover, wavelets with zero inner product are called orthogonal to each other. [7] [8] [9].



(a)



(b)



(c)

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

2.6 Role of Multi Resolution Analysis (MRA)

In multi resolution analysis, a series of approximations of a signal are produced with the help of a scaling function [10] and alteration in data between neighboring approximations are encoded with by using wavelet function. Higher resolution is required for areas with high detail. MRA gives us knowledge location wise using self-similarity property. Wavelet analysis is a method of Multi Resolution Analysis. In the image transformation sub band coding is the part of Multi Resolution Analysis.

2.7 Two Channel Wavelet Filter Bank:

when X(z) is fed to High Pass analysis $H_1(z)$ and low pass analysis Filters, Smaller bandwidth is obtained. Down sampling is done, this completes the

“Analysis”. In Synthesis phase, received signal is up-sampled first, and then synthesis filter like low pass Go(z) and high pass synthesis filter Gi(z) is applied then their combination is represented as reconstructed signal.

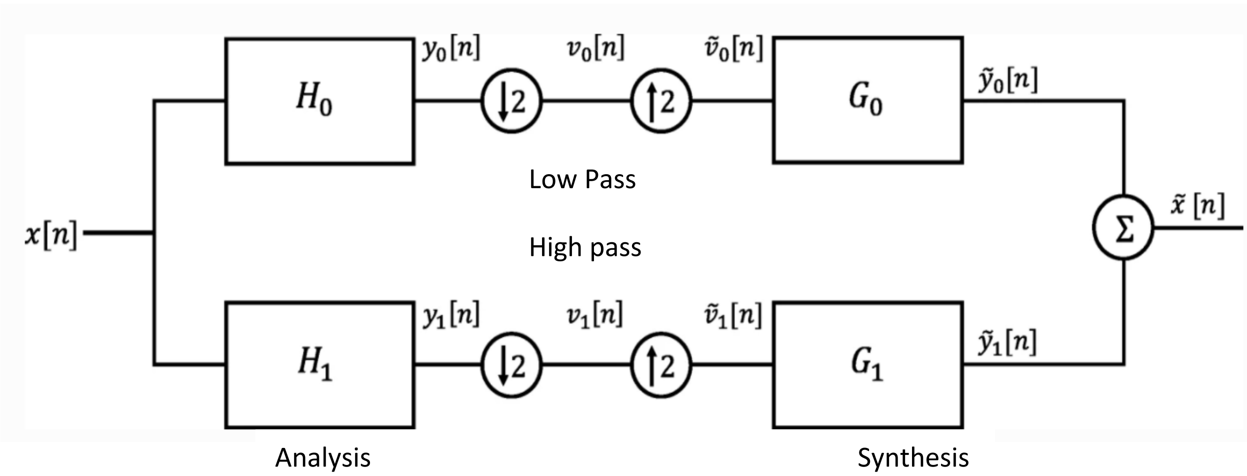


Fig.2.3 A two-channel filter bank (Analysis and synthesis)

Aliasing is induced after down sampling. Analysis filter induces distortion. Whereas, the synthesis filter is to reduce these distortions. Analysis and synthesis can be expressed as follows

Go(z) Ho(z) + Gi(z) Hi(z) = 2 (2.1)

Go(z) Ho (-z) + Gi(z)Hi(-z) = 0 (2.2)

2.8 Classification of Wavelet Transform

One of the most renowned time–frequency transform 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. Wavelets are used to examine signals, there are two major transforms

1. Continuous Wavelet Transform
2. Discrete Wavelet Transform

2.8.1 Continuous Wavelet Transform (CWT)

The continuous wavelet transform is the natural extension of discrete transform. A continuous function is transformed into 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 identification and detection of concealed transients (difficult to detect the small details of signal) are specifically treated with the help of CWT[11]. Fourier analysis in Fourier transform is mathematically expressed as:

$$F(w) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t}\, dt \tag{2.3}$$

The exponential $''-j\omega t''$ is the superposition, the CWT is mathematically defined as under

$$C\,(\text{Scale}, \text{position}) = \int_{-\infty}^{\infty} f(t)\psi\,(\text{Scale}, \text{position}, t)\, dt \tag{2.4}$$

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

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left[\frac{t-b}{a}\right] \;;\; a, b \epsilon R^1 \;\; \text{and} \;\; a > 0 \tag{2.5}$$

Where ‘a’ represents scaling factor to define width of a basis function and ‘b’ to translate the wavelet in the time domain. So the continuous wavelet transform can be defined as:

$$W_f(a, b) = \int_{-\infty}^{\infty} x(t)\, \psi_{a,b}(t) dt \tag{2.6}$$

2.8.1.1 The inverse Wavelet Transform

IWT can be defined as :

$$x(t) = \frac{1}{C} \int_{0}^{\infty} \int_{-\infty}^{\infty} W_f(a, b)\, \psi_{a,b}(t) db \frac{da}{a^2} \tag{2.7}$$

Here $C = \int_{-\infty}^{\infty} \frac{|\psi|^2}{\omega} d\omega < \infty$ (2.8)

The Fourier transform of ψ(t) (mother wavelet) results in the form of X(t) and it must fulfill two conditions. Number one, C should be finite with X(t) to have mean to be zero. This is called the admissibility condition and can be expressed mathematically as follows

$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \qquad (2.9)$$

Number two, the mother wavelet must have finite energy.

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt = \infty \qquad (2.10)$$

2.8.2 DWT Decomposition

In Fourier analysis, sinusoidal basis functions of different frequencies are obtained once Discrete Fourier Transform (DFT) decomposes a signal. In lossless transformation original signal can be formed using DFT. Whereas, DWT beaks down a signal into orthogonal wavelets basis. These functions don't match with the sinusoidal basis functions. Moreover, wavelet functions are scaled and translated versions of mother wavelet φ. DWT is invertible like that of Fourier analysis, to recover the original signals. However, DWT is a set of transforms, each with a different wavelet basis function sets unlike DFT. 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, wavelet function set makes an orthogonal set of basis functions.

2.8.2.1 DWT in One Dimensional

To localize and decompose 1-D discrete time signal, DWT one dimensional is used.[11].

Orthogonal $L^2(\mathcal{R})$ is defined as:

$$a_{j,k} = \int x(t) 2^{j/2} \phi(2^j t - k) dt b_{j,k} = \int x(t) 2^{j/2} \psi(2^j t - k) dt \quad (2.11)$$

To synthesize $L^2(\mathcal{R})$, orthogonal inverse can be defined as:

$$x(t) = 2^{N/2} \sum_k a_{N,k} \phi\, (2^N t - k) + \sum_{j=N}^{M-1} 2^{j/2} \sum_k b_{j,k} \psi\, (2^j t - k) \quad (2.12)$$

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 can be represented as

$$\tilde{a}_{j,k} = \int x(t) 2^{j/2} \tilde{\phi}(2^j t - k) dt \tilde{b}_{j,k} = \int x(t) 2^{j/2} \tilde{\psi}(2^j t - k) dt \quad (2.13)$$

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

$$x(t) = 2^{N/2} \sum_k \tilde{a}_{N,k} \phi\, (2^N t - k) + \sum_{j=N}^{M-1} 2^{j/2} \sum_k \tilde{b}_{j,k} \psi\, (2^j t - k) \quad (2.14)$$

2.8.2.2 Two-Dimensional DWT

Images are the 2D form of 1D data transmitted. 2 stages of 1D wavelet transform are cascaded in series. 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[12]. To analyze the 2D images two 1-D wavelet transforms are connected in series as they are the 2-D form of 1-D signals. 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. In figure 2.4 the perfect reconstruction filter bank Represents the DWT and Inverse DWT.

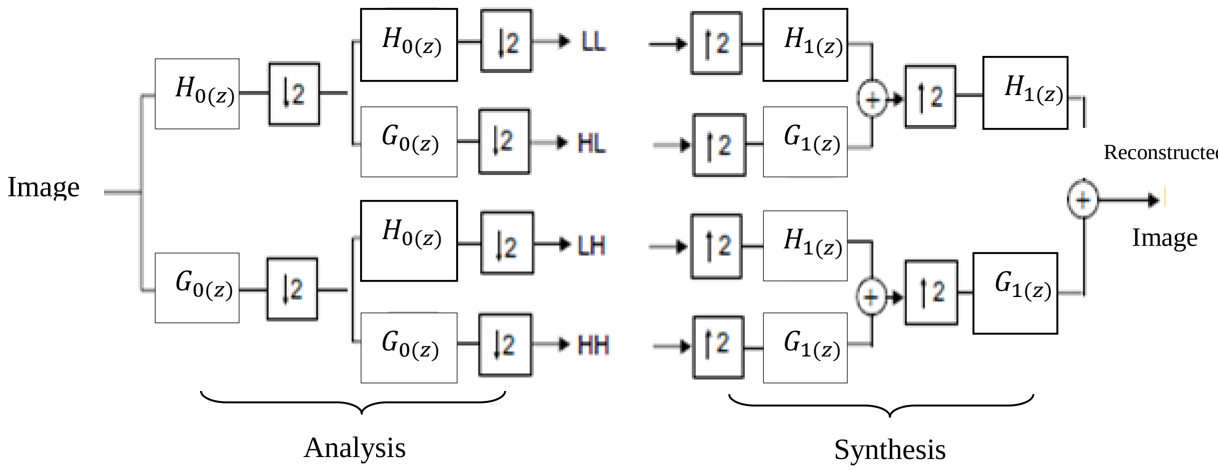


Figure-2.4: filter bank for 2-D DWT

Transformed coefficients are generated as a result of 2-D image. So, we have 4xbasis functions which have been represented in the equations from 2.25 to 2.28.

$$\phi\,(u,v) = \phi(u)\,\phi(v) \quad (2.15)$$

$$\psi_1(u,v) = \psi(u)\,\phi(v) \quad (2.16)$$

$$\psi_2(u,v) = \phi(u)\,\psi(v) \quad (2.17)$$

$$\psi_3(u,v) = \psi(u)\,\psi(v) \quad (2.18)$$

Where, $\phi\,(u,v)$ represents scaling function of the images whereas $\psi_1(u,v), \psi_2(u,v)$and $\psi_3(u,v)$ represents the wavelet functions. After transformation of the original image, it is decomposed four sub-bands. (Figure 2.6):

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

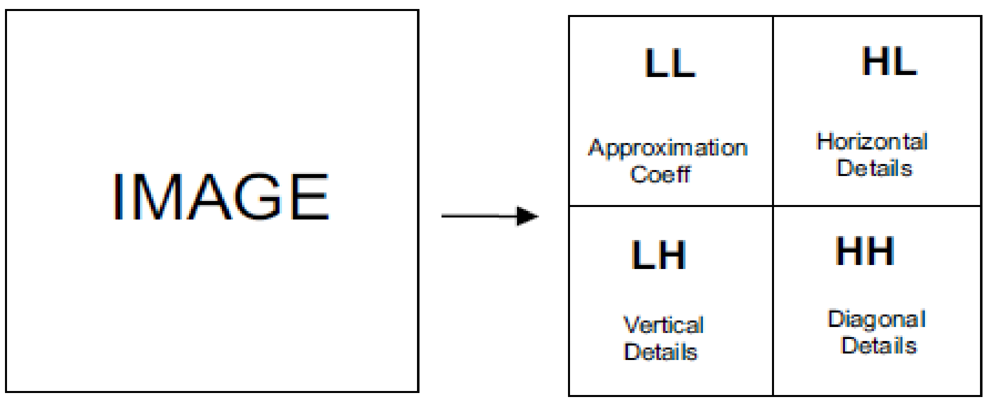


Figure-2.5: Decomposition of 2D to 1D

3 Embedded Zero-tree Wavelet (EZW) Transform

3.1 Introduction

This chapter is about an embedded coding transform EZW, it is a compression algorithm where, The E of EZW represents Embedded(progressive), Z represent data structure of Zero-trees which encodes the data and W for wavelet transform. Each term will be defined one by one in this chapter.

3.2 Process of EZW

EZW algorithm generates an embedded bit stream for compression improvement. In order to understand EZW, we need to define the connection of sub-bands and then hierarchical tree structure to identify the parent-offspring relation.

3.2.1 Relation between Sub-bands

In images hierarchical relationship exists between same spatial locations. Lower frequency components are associated as parent coefficients to the higher frequency components as off-springs. Descendants are coefficients at higher detail resolutions for a parent at the similar spatial orientation.

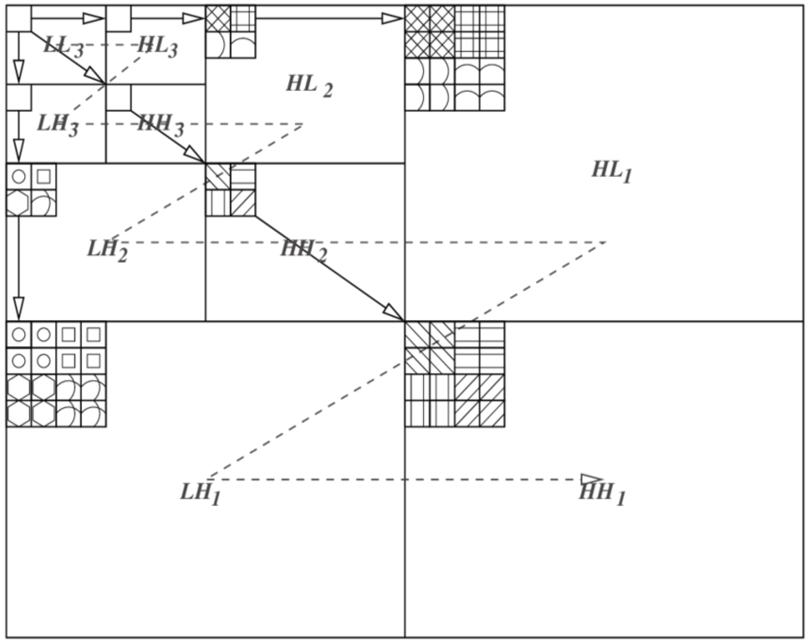


Figure 3.1 Parent-Offspring Relation between sub-bands

3.2.2 Zero-tree Root

DWT coefficients are scanned in the form of the alphabet Z, starting from the top left LL3 band which is known as lowest frequency sub-band as in Figure 3.2

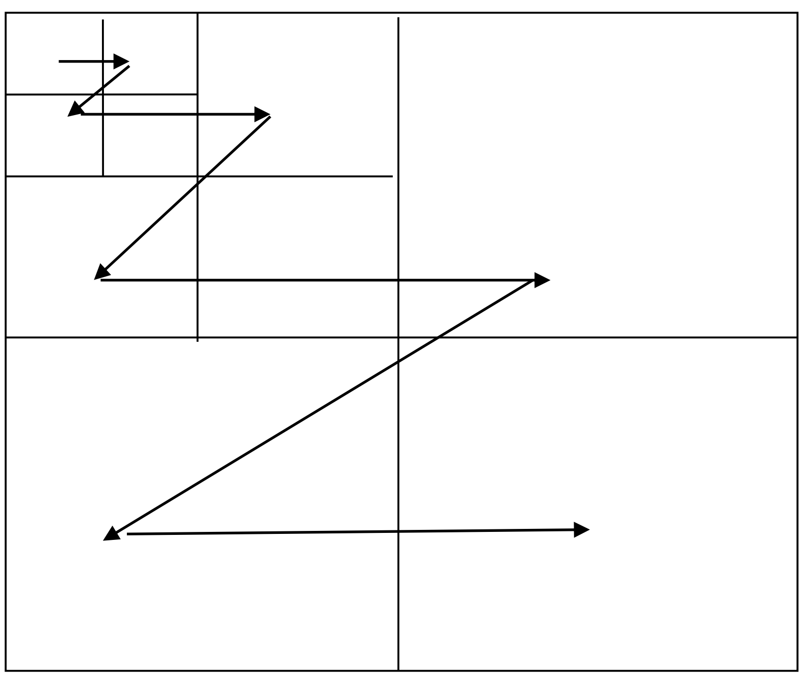


Figure 3.2: Scanning procedure in 'Z' manner

For a Zero-Tree the quad-tree must be insignificant with respect to the provided threshold. Usually, if a coefficient at lower frequency is insignificant then all its descendants will be insignificant. ZTR is coded with one symbol and regenerated as a quad-tree filled with zero.

But not all the insignificant coefficients are termed ZTR. In that case it is termed as Isolated Zero (IZ).

The EZW encoder works on two 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 appears 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. [13]

3.2.3 Progressive (Embedded) Coding

Embedded coding which is also known as progressive coding follows order of coded bits is set according to their importance, starting with the lower rates. [20][14][15]. Progressive coding can stop encoding process at any stage to attain the required bits for which, bit calculation and truncation is maintained by the encoder. EZW starts with the most Significant bit plan and gradually carries on with the most significant bit plan coming next and forth. Reconstruction error can be avoided with the help of "significant ordering" of the embedded bit stream.
First thing that is shared on transmission network is coarser details of the image and afterwards the details according to generation of embedded code in compression algorithm. [21]

"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 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 mush lesser rates. [22]

The diagram of an embedded codded scheme is as under:

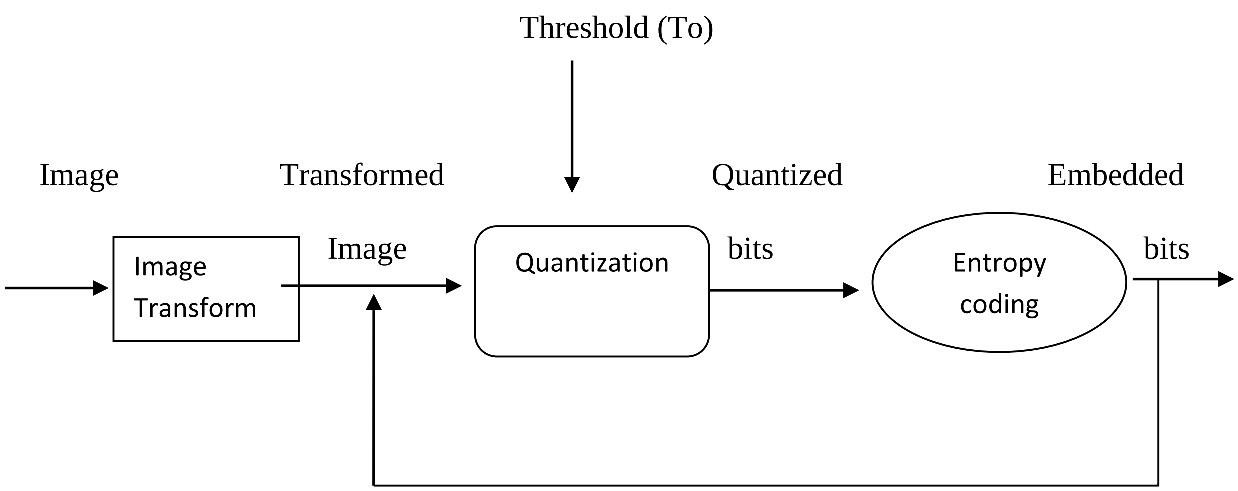


Figure 3.3: Progressive encoding Scheme

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 decorrelated 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 [16]. 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 [17] that user can select bit rate as per his desire and encode the image according to that. Figure 3.3 explains the details as that how EZW coding algorithm is applied and desired bit rate is achieved using feedback.

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 [23] decoding any time 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. The refinement details are followed by this within the framework of progressive broadcast”

In figure 3.4 encoding and decoding both are elaborated in the diagram.

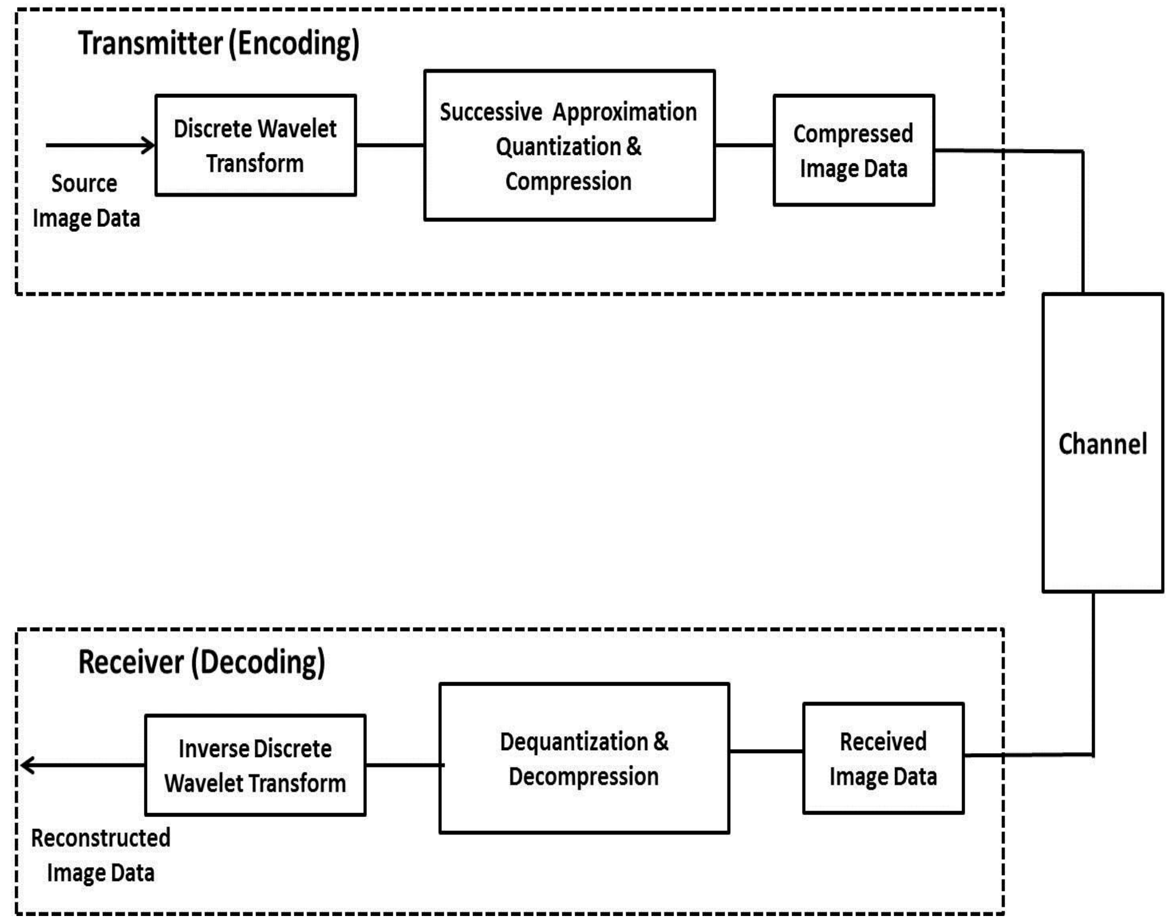


Figure 3.4: EZW Compression and decompression Diagram

3.2.4 Significance Map Encoding

In EZW a few bits were required for the position information[24], known as significance maps. A significance map basically determines significance and insignificance of a coefficient in binary. A coefficient is quantized to zero if it not significant. However, a significant coefficient is given a non-zero value. EZW used a proficient method for significance map. It is achieved by encoding zero's location[14]. After lot of statistical analysis, it has been experimentally proved that in the wavelet transform, zeros can be predicted across the scale precisely. Usually in EZW we assume if a pixel coefficient is insignificant then its descendants will be insignificant too [15][18]. Zero tree helps code the insignificance of coefficients across the map accurately. Using Zero-Tree technique where insignificant data is grouped, the significance map cost is reduced exponentially[15]. Significance map and quantized coefficients map can be seen 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

In significance map only non-zero values are transmitted; it helps in the reduction of low bit rate image coding. In order to achieve the lower bit rate coding probability of zero should be higher. Due to this not only efficient encoding the significance map is achieved but also it offers a higher efficiency in compression. As huge budget is used in encoding the significance map.

The importance of significance map encoding can be understood by understanding a typical transform coding. In order to understand the coding cost of position its amplitude and sign information. A low bit-rate image coder is comprised of three basic components: transformation, followed by quantization and data compression, as shown in the figure3.3 below” [25].

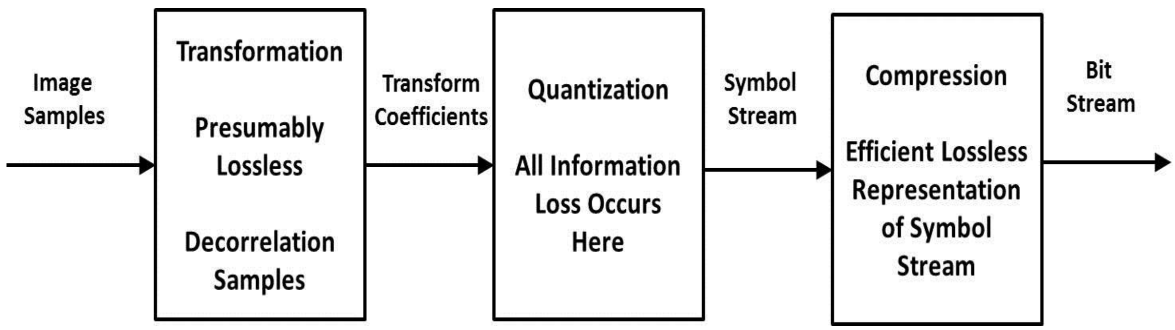


Figure 3.3: Low bit rate image coder

3.2.5 Successive Approximation Entropy-Coded Quantization

The successive approximation quantization (SAQ) [26] provides a vital facet to embedded compression algorithms. Symbol streams produced by these for one rate has bit streams for all lower rates. J.M. Shapiro defined an embedded code [15] as:

1) While encoding for different data rates, results of two images at different data rates must match. In short, adding more symbols gives us the precise results.

2) A better result should be produced for any data rate.

SAQ is used for progressive coding to encode magnitudes[27]. The SAQ applies a number of thresholds sequence wise, to check the significance of the data. The starting threshold T0 is selected in such a way that $|Xj| < 2T_o$for all transform coefficients. After this the subsequent thresholds are selected as Ti = Ti-1/2. Wavelet coefficients are present in two separate lists in the process of encoding and decoding. Dominant list has the coordinates of the insignificant coefficients following the order followed during the process of the initial scan. Sub-bands are ordered in this scan, and the set of coefficients are well-arranged.

Absolute value of significant coefficients is stored in the subordinate list. We scan the list once for each threshold.

We compare the coefficients during a dominant pass to check the significance. Next, if a coefficient is significant, it's sign is checked. This significance map is coded as zero-tree later. If a coefficient is found to be significant it is coded as POS or NEG then each time its absolute value is attached to subordinate list. Wavelet transform coefficients' range is set zero to prevent the occurrence of significant coefficient as a zero-tree at smaller threshold. 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. '0' is the representation of the value to be in lower half interval. In a subordinate pass the string of binary alphabets are generated. And are fed to entropy encoder. After subordinate pass, these are arranged in descending order magnitude wise. The process keeps alternating between subordinate passes and dominant passes. Threshold is reduced to half before each dominant pass.

3.3 The EZW algorithm

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 that is highly correlated. And higher frequency components are of great importance too. Hence, in high quality coding scheme it is necessary to precisely signify the high frequency components. The transformed coefficients may be considered lower frequency trees with the children being spatially related to the higher frequency bands. More than one sub trees will be Zero-trees[18].

In EZW, coefficients are coded using their statistical properties[19]. As most of the coefficients are insignificant, it helps in compression of data. The significance is determined by comparing the coefficient and it's descendants with the 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 [14] are used by the EZW to represent the wavelet coefficients:

- Zero Tree Root (ZTR)
- Isolated Zero (IZ)
- Positive Significant (P)
- Negative Significant (N)

Two binary bits are used by EZW to represent the above-mentioned symbols. After every dominant pass, threshold is divided by '2'. One symbol is transmitted of the four scanned symbols, which means dominant pass encodes the coefficients that are not significant yet. The descendants of a coefficient are checked only if the coefficient is significant, otherwise it is termed as an Isolated Zero. In the subordinate pass MSB of each significant coefficient is produced. This depicts that it is like bit plane coding.

The coefficients' magnitude is compared by the encoder with the initially selected threshold. Encoder sends the inform the decoder whether the magnitude is greater or Positive than the given threshold. For nearly precise results at the decoder end, encoder must also send the information about threshold value.

In order to reduce the number of bits by sharing the threshold information, we predefine the rule for threshold, the thresholds represents binary value of magnitude of the coefficients. Hence, it is known as bit plane coding[15].

The decoder needs information about the position of the coefficients to reconstruct the transmitted signal. Efficient encoder is differentiated by the inefficient ones by coding of the positions.

A predefined sequence for scanning is followed for encoding the positions. (as shown in figure 3.4). zero-trees code the positions perfectly[28]. Different orders can be used for scanning depending on the encoder.

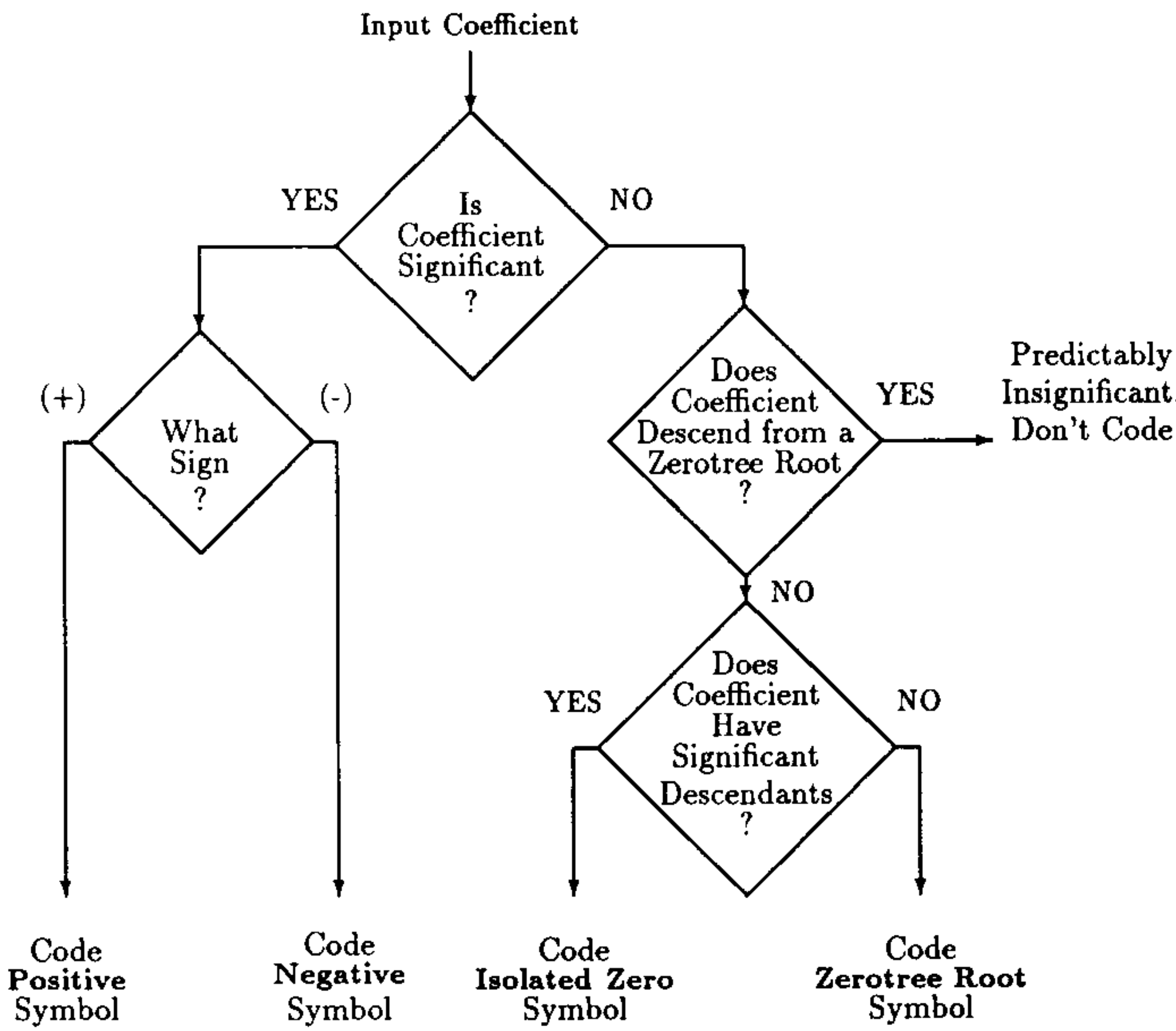


Figure 3.6: encoding a significance map (Flow chart)

Final results of compression do get affected by the order of scan. Initial threshold is defined as

$$t_0 = 2^{\lfloor \log_2(\text{MAX } (|\gamma\ (x,y)|)) \rfloor} \quad (3.1)$$

3.4 Example

To illustrate the above stated algorithm we take an example shown in Table 3.2 and figure 3.7.

26	**6**	**13**	**10**
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

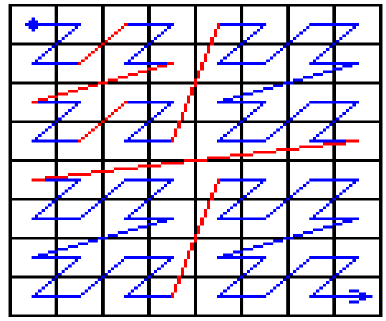


Table 3.2: (a) data set **Figure 3.7:(b) scanning order (Morton scan)**

[16], table 3.2 and figure 3.7 is showing that the order of scan used is Morton scan. Initial threshold as calculated by putting in relevant equation 3.1 is t_0=16. After the process of one pass the EZW algorithm generates following bit stream".

- **sp**: If magnitude of a coefficient is greater than the threshold, and sign is positive it is called positive Significant Coefficients.
- **sn**: If magnitude of a coefficient is greater than the threshold, and sign is negative, it is called **negative significant**
- **zr**: if the magnitude of a coefficient and it's descendants are less than T. the coefficient is called **zerotree root**.
- **iz**: If the coefficient is Insignificant but all of it's descendants are not, it is called **isolated zero**

1st **Dominant Pass:** Initial threshold=To= 16

- 26 > 16 sp
- 6<16
  descendants < 16 zr

- -7<16
  descendants < 16 zr
- 7<16
  descendants<16   zr

Transmitted labels will be

sp zr zr zr

*****	**6**	**13**	**10**
-7	**7**	**6**	**4**
4	**-4**	**4**	**-3**
2	**-2**	**-2**	**0**

Subordinate Pass:

Ls={26}

The significant coefficient reconstructed value

1.5To=24

Correction bits: 16/4=4

24	**0**	**0**	**0**
0	**0**	**0**	**0**
0	**0**	**0**	**0**
0	**0**	**0**	**0**

24+4= 28	**0**	**0**	**0**
0	0	0	0
0	0	0	0
0	0	0	0

Table 3.3 (a)Reconstruction   (b)Reconstruction using correction bits

2nd Dominant Pass

T_1=   ½*T0=   ½*16=8

6   <8
descendants >8   iz
-7   <8
descendants <8   zr
7   <8
descendants<8   zr
13
no descendants>8   sp
10
no descendants > 8   sp
6

Table 3.4 (a)Matrix for second pass

no descendants < 8 iz
4 no descendants < 8 iz

transmitted symbols

iz zr zr sp sp iz iz

14 bits

Total bits required = 9 + 14 =23

Subordinate Pass:

The significant coefficient
1.5T1 = 1.5 * 8 = 12

Ls = {26, 13, 10}
reconstructed bits

28	**0**	**12**	**12**
0	0	0	0
0	0	0	0
0	0	0	0

26	**0**	**14**	**10**
0	0	0	0
0	0	0	0
0	0	0	0

Table 3.4 b: Reconstructed c: using correction bits

3rd Dominant Pass:

*T2 = ½ * T1 = ½ *8=4*
6 > 4 sp
|-7| > 4 sn
7 > 4 sp
6 > 4 sp
4 = 4 sp
4 = 4 sp
|-4| = 4 sn
2, -2 iz
4 = 4 sp
-3, -2, 0 iz

*	**6**	*	*
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

Table 3.5(a) Matrix for 3rd dominant pass

sp sn sp sp sp sp sn iz iz sp iz iz iz
26 bits

Total bits required = 26 + 26 =52

3rd subordinate pass:

- *The significant coefficient 1.5 * T_2* = 1.5 * 4 = 6
- *Ls* = {26, 13, 10, 6, -7, 7, 6, 4, 4, -4, 4} reconstructed bits

*****	**6**	*****	*****
-6	6	6	6
6	-6	6	-3
2	-2	-2	0

26	**6**	**14**	**10**
-6	6	6	6
6	-6	6	0
0	0	0	0

27	**7**	**13**	**11**
-7	7	7	5
5	-5	5	0
0	0	0	0

Table 3.5 (b) Reconstruction (c) using correction bits

Considering that EZW algorithm needs 2 bits to code symbols in alphabet {POS, NEG, ZT, IZ}, **52 bits** were used after the third pass.

3.5 Shortcomings:

EZW has two major shortcomings. First, not as optimal in terms of defining the threshold and needs to be optimized according to required bit rate.

Second, it does not efficiently encode the insignificant transformed coefficients. Hence the encoding efficiency is compromised.

Chapter-4

Set Partitioning in Hierarchical Trees Algorithm

4.1 Introduction

We discussed EZW in previous chapter, there are two major points. Firstly, we arrange bit stream I order of their significance and it is progressive in nature. Hence, bit stream can be stopped at any point according to the channel. Secondly, it exploits the self-similarity between sub-bands. It helps in the data trim. It’s shortcomings and strong points both are discussed in previous chapter in detail, hence that leads to the need of another better algorithm.

Amir Said and A. Pearlman 1996 presented another method called ‘Set Partitioning in Hierarchical Trees (SPIHT)”.

4.2 SPIHT over EZW

SPIHT is the modified version of EZW. Which has overcome the shortcomings of the EZW in terms of optimization and well performance [32]. SPIHT Algorithm works on the concept of ordering partially by the magnitude and self-similarity is being exploited across different scales of an image transform. SPIHT transmits coefficients in order, this property is inherited by EZW. Partitioning refers to splitting of coefficients in significant and insignificant sets. However, SPIHT gives us higher PSNR as compared to EZW because a special symbol indicates the child and parents node’s significance [29][30][31]. SPIHT exhibits better results than EZW because of utilization of insignificant coefficient set.

SPIHT bit stream possesses the unique 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, in SPIHT no information is sent to the decoder. Decoder recovers the information and execution path itself. For smooth execution and recovery, the time of execution should be same, which isn't usually the case in other schemes.

4.3 Partial Ordering and Progressive Transmission

ordering is done by comparing the transformed coefficient's magnitude with a set of decreasing thresholds [32]. The hierarchical sub-band transformation like wavelet can be described as

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

"Here c is the output array of the transformed coefficient, which are produced once Ω sub-band is applied on the input image. Both the output and the input image have same dimensions of the coefficient array. Encoder and decoder process the coefficients as per the defined SPIHT algorithm". the inverse transformation is applied to estimate the image. given by

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

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

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

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

Pixel intensity at n1,n2 is represented by s_{n_1,n_2} Since the sub-band transformation is lossless therefore, the mean square error (MSE) is independent of it.

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

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

Coefficient of transformation is represented by c_{n_1,n_2} at the position n_1, n_2. Initially, $\hat{c}_{n_1,n_2}$=0 , then MSE is reduced by $\frac{(c_{n_1,n_2})^2}{N}$.This indicates that the coefficients with the higher 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 improved reconstruction as compared to the smaller valued coefficients. Therefore, the coefficients are arranged with respect to their magnitudes in the embedded stream coding. In EZW we arrange the data in the subordinate pass. We can arrange the bit-planes when we order coefficients. In equation form this can be represented as

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

In equation 4.2 $c_n(k)$represents the coefficients which have been ordered according to their magnitude values.

This example will help us understand the ordering concept in a better way. Here, coefficient array is as follow: -3, -9, 5, 8, 2, -12, 14, 16, -17, -6, 38, 25, -57 and -7. By using ordering equation, the array can be arranged as follows (Table 4.1):

Magnitude	57	38	25	16	17	14	12	9	8	7	6	5	3	2
Sign	1	0	0	0	1	0	1	1	0	1	1	0	1	0
Bit5 (msb)	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Bit4	1	0	1	1	1	0	0	0	0	0	0	0	0	0
Bit3	1	0	1	0	0	1	1	1	1	0	0	0	0	0

Bit2	0	1	0	0	0	1	1	0	0	1	1	1	0	0
Bit1	0	1	0	0	0	1	0	0	0	1	1	0	1	1
Bit0 (lsb)	1	0	1	0	1	0	0	1	0	1	0	1	1	0

Table 4.1: binary representation coefficients

Coefficients are arranged according to the magnitude and signs. The order of coefficients follow the equation of ordering but nor the rule for magnitude. All the bits are transmitted from highest to the lowest coefficient.

One sorting pass is sent in embedded transmission, where the coefficients are arranged according to the relation $2^n \leq |c_{n_1,n_2}| < 2^{n+1}$. and then bit stream is sent which is set according to their significance, which is a result of refinement pass.

4.4 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. Execution path helps in recovering the ordering information at decoder end, if the encoder shares magnitude comparison results".

So it can be said that the set partitioning is devoid of explicit ordering of the coefficients are observed for a certain value of n, following $2^n \leq |c_{n_1,n_2}| < 2^{n+1}$. A coefficient is significant if $|c_{n_1,n_2}| > 2^n$ and is insignificant if it does not. We coefficients 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. If $T_n$is significant then we classify it to significant and insignificant subsets. We continue partitioning Until we come across a single significant coefficient. A sub-band hierarchical framework is followed in set partitioning technique. The main purpose of the set partitioning approach is to have maximum elements as significant/ insignificant according to the requirement.

4.5 Spatial Orientation Tree

Lowest frequency components contain most of the energy of an image. Hence, from highest to lowest levels of sub-bands value of variance reduces [32]. connection of the sub-bands can be seen in figure 4.1 in the form recursively split four bands, pixels represent nodes. 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 top 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. Rest of the three 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 pixel in LL band which is marked with ‘*’, as displayed in figure 4.1 does not determines any descendants”.

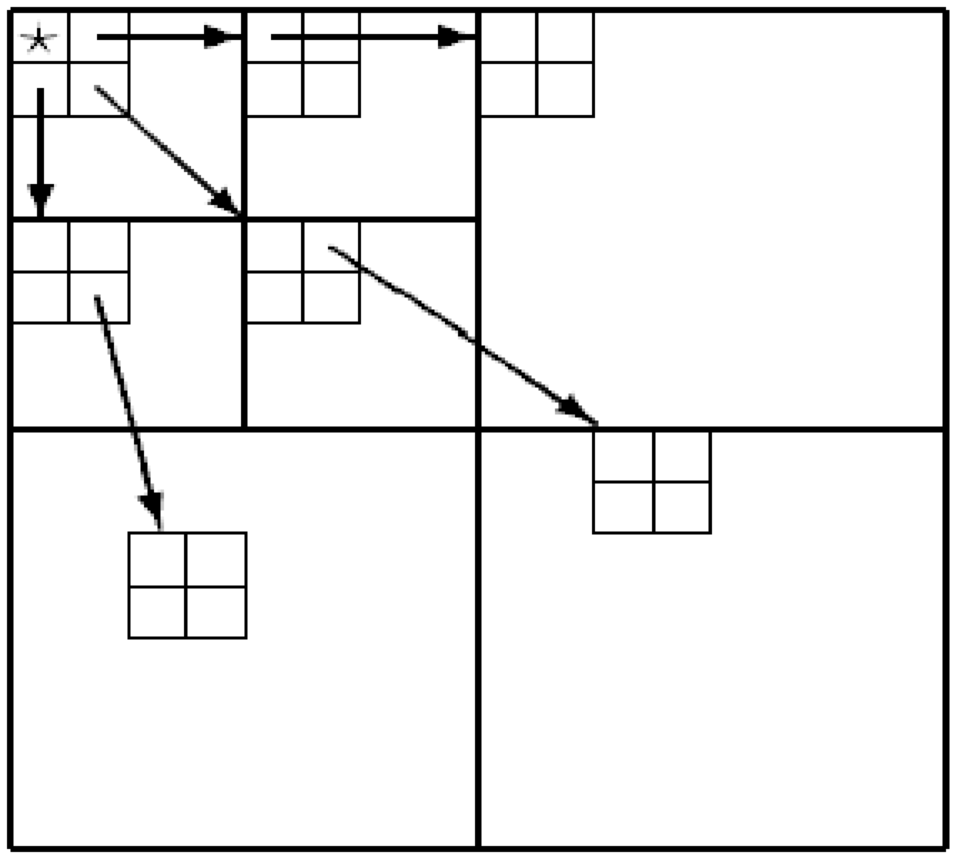


Figure 4.1: Spatial orientation tree

4.6 Rules of Set Partitioning Algorithm

For better understanding of the concept, few important sets of notations which are used 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 represents the offspring and $\mathbf{n_1}$&$\mathbf{n_2}$ are the coordinates of pixels of the offspring or it can be said that these represents the set of offspring of the node(n_1, n_2). For example, the $O(0,1)$ has coordinates of the pixels b_1, b_2, b_3 and b_4.

2) $\mathbf{D}(\mathbf{n_1}, \mathbf{n_2})$ **:** represents the set of descendants. D represents descendants whereas $\mathbf{n_1}$&$\mathbf{n_2}$ represents the positions of the pixels. 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 pixels $b_1 - b_4, b_{11} - b_{14}, b_{41} - b_{44}$. 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) $\mathbf{L}(\mathbf{n_1}, \mathbf{n_2})$ **:** is the remining pixels of $D(n_1, n_2)$ and $O(n_1, n_2)$. This notation is used to represent the set which has the coordinates of a descendants at position at (n_1, n_2) less than the offspring or we can say that It may be represented as .

$$L(n_1, n_2) = D(n_1, n_2) - O(n_1, n_2) \tag{4.9}$$

4) **H:** This consists of pixels of LL sub-band that belongs to the highest level.

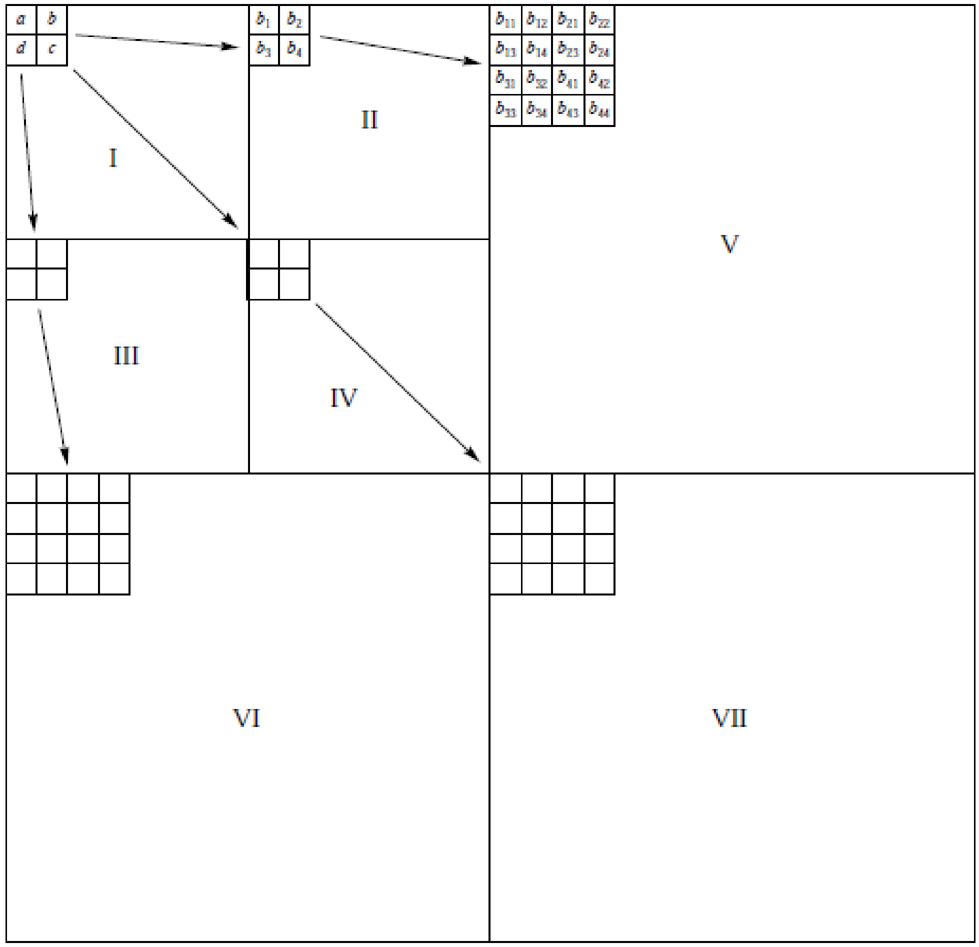


Figure 4.2: Data structure in SPIHT algorithm

For the set partitioning technique there is a need to follow three basic rules sternly. These are as under

1) the initial division is of $D(n_1, n_2)$. Where $(n_1, n_2) \in H$.

2) if $D(n_1, n_2)$ is significant, and then we divide it more to $L(n_1, n_2)$ and four sets of only one pixel.

3) if (n_1, n_2) is significant, then we divide it further to four sets.

4.7 SPIHT Encoding and Decoding

At both encoder and decoder ends similar algorithm is being run. In addition to that, ordering information is not sent to decoder like other algorithm of embedded transmission which makes SPIHT algorithm more efficient than the others. The set of lists updated during the process to achieve the better compression rate are enlisted below.

I. List of Insignificant Pixels (LIP)
II. List of Significant Pixels (LSP)
III. List of Insignificant Sets (LIS)

Entries are identified by its pixel location(i,j). entries in LIP and LIS are pixels. Whereas, elements in LIS are sets. 'n' is found with the help of the coefficients' that is the amount of magnitude required in refinement pass. There are three passes after initialization:

1) The sorting pass
2) The refinement pass
3) Improvement of quantization step pass

"The recursive process of these passes continues and keep repeating iteratively itself till least significant bit is shared. During the sorting pass insignificant pixels are checked if significant or not. If yes, these are moved to LSP. Similarly, the significance of the LIS is checked and those found significant are partitioned and taken out from the LIS. Pixels are adjusted in the LSP or LIP in accordance with their significance. In refinement pass, the significant pixels are encoded for most significant bit. With this preview the encoding algorithm can be summarized as:

4.7.1 Initialization:

1) $n = \lfloor \log_2(\max_{(n_1,n_2)} \{|c_{n_1,n_2}|\}) \rfloor$
2) LSP = { }
3) LIP = $\{(n_1, n_2) \in H\}$ and LIS= $\{D(n_1, n_2), (n_1, n_2) \in H\}$

4.7.2 Sorting pass

1) LIP is checked for the significant pixels. '1' is sent for significant and '0' for insignificant. Significant entries are added to LSP and removed from LIP.

2) LIS is checked for the significant pixels. Sign of significance is output for the significant entry. Divide L or D.

Keep updating the LSP, LIP and LIS based on the significance.

4.7.3 Refinement pass

Most significant bit is transmitted for all the elements in LSP.

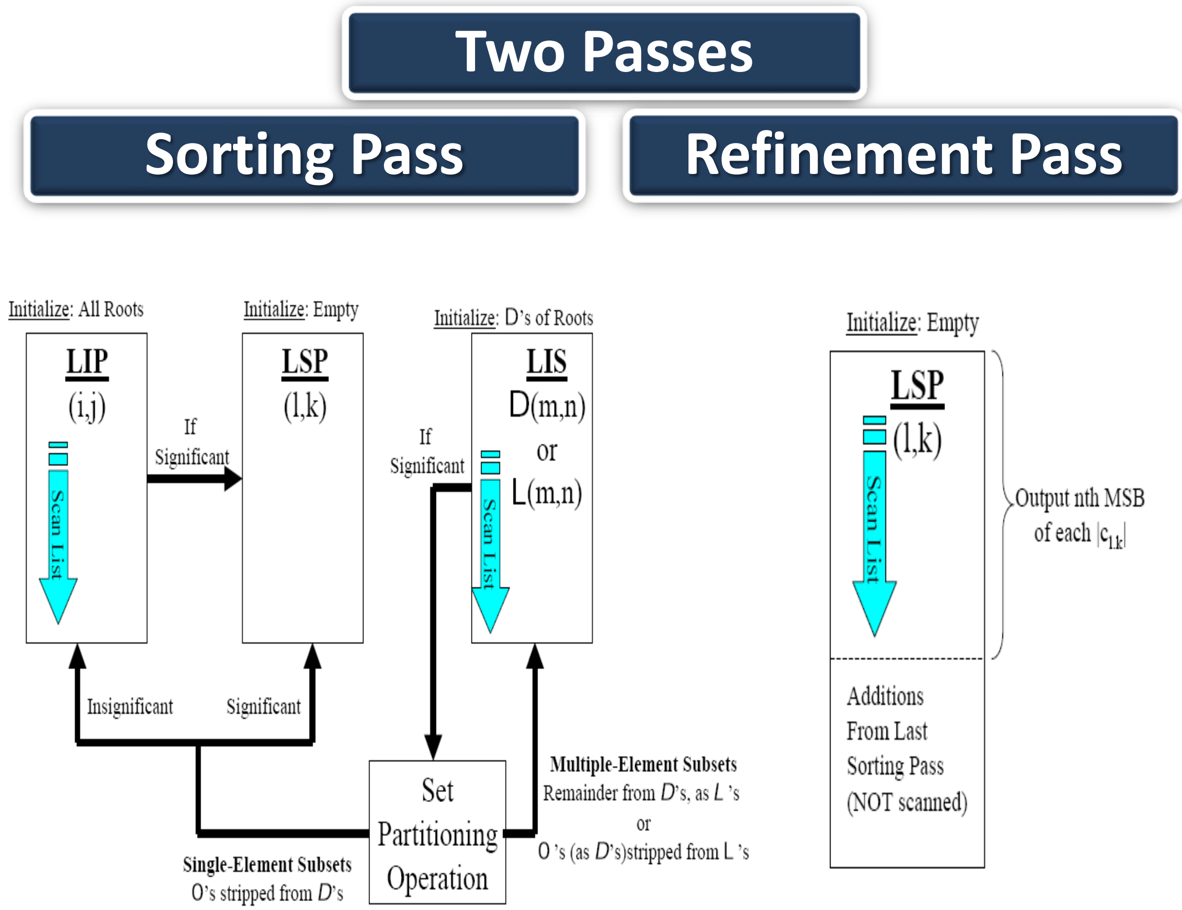


Figure 4. Initialization

4.7.4 Renewing quantization step pass

1. n is reduced by 1 and previous steps are repeated. Until n=0
2. Decoder and encoder follow the same steps. Decoder is inputted encoder's output. Entropy coder can be attached to improve the efficiency.

4.8 Example

SPIHT algorithm is applied to same data as was used for EZW. A few important definitions are here

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

Table4.2: Set of Image Wavelet Coefficients used by example.

LIS (List of insignificant set): Set of coefficients whose magnitude is smaller than the threshold. A set has minimum 4 elements.

LIP (List of insignificant pixels): pixels whose magnitude is smaller than the threshold.

LSP (List of significant pixels): pixels whose magnitude is greater than the threshold (are significant).

O(i, j): the set of direct/ immediate descendants of a node(i,j) only, defined by pixel coordinates (i, j).

D(i, j): set of descendants of node defined by pixel location (i, j).

L(i, j): set defined by L(i, j) = D(i, j) − O(i, j).

Initialization:

At initialization these three lists are generated as.

Assumptions:

1= Significant bit

0= Insignificant bit

1= negative sign

0 = positive sign

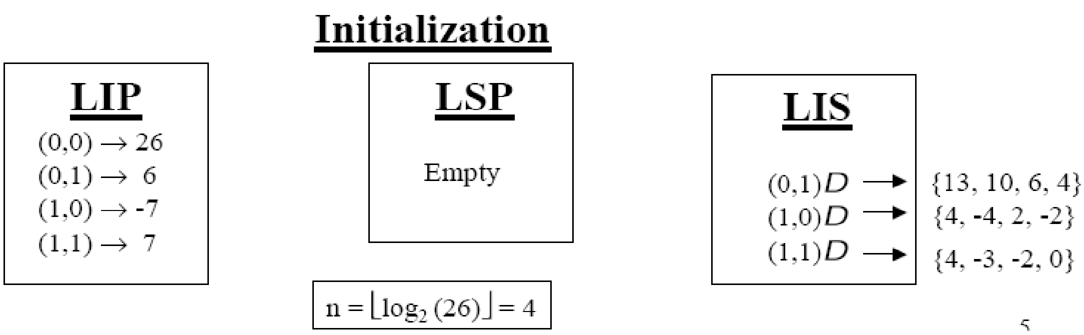


Figure 4.3 Initialization

First Pass:

To= 2^n and n=4. Hence, To= 16

First Sorting Pass:

At (0,0) the entry is greater than 16. Rest of LIP the coefficients are insignificant hence 0 is transmitted for those and as it's descendants are insignificant of (0,1) , (1,0) , (1,1) Hence '0' is transmitted for each entry. (0,0) is sent to LSP.

First Refinement pass:

'1' is transmitted and followed by '0'to represent it's sign. Rest of LIP the coefficients are insignificant hence 0 is transmitted.

Transmitted bits: 10000000

And updatedlists are as follows:

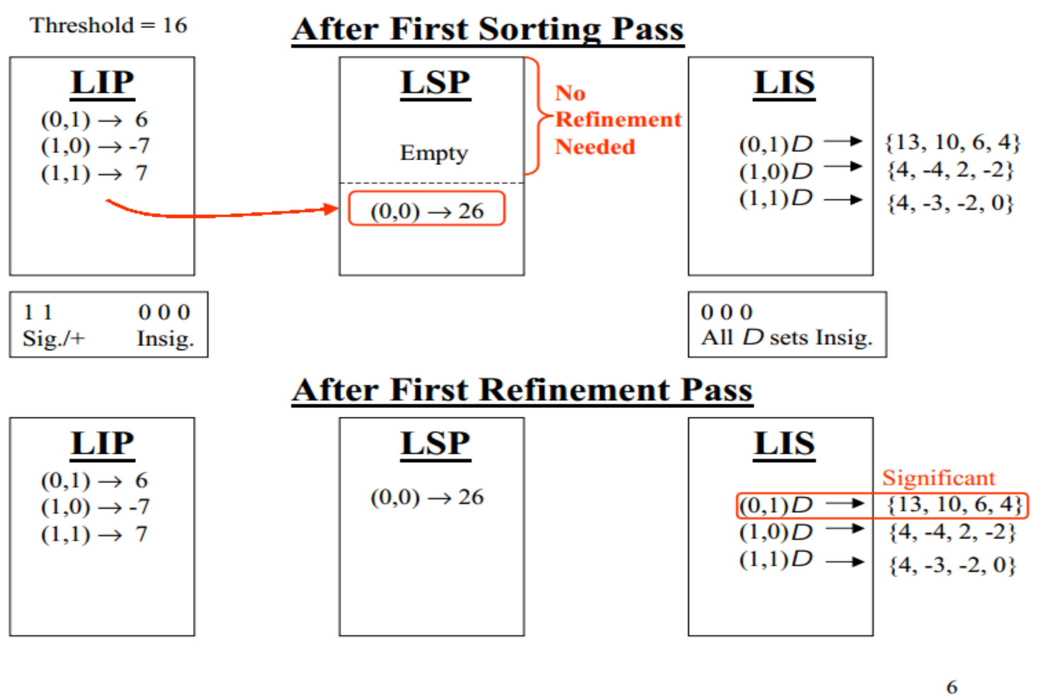


Figure 4.4 1st sorting and refinement pass

Second Pass:

N=3 hence threshold=2^3=8

Sorting pass:

Firstly, LIP entries are checked, if insignificant then these will be kept in the list.

LIS is examined after that, first element of LIS has two descendants Significant i.e. 10 and 13. And these two pixels are moved to LSP and remaining are moved to LIP as the set is open.

Refinement pass:

'0' are transmitted for LIP entries. For 13 '10' and for 10 '10' is transmitted where '1' represents significant bit and followed by the representation of positive sign '0'.

Transmitted bits after 2nd pass:

0001101000001

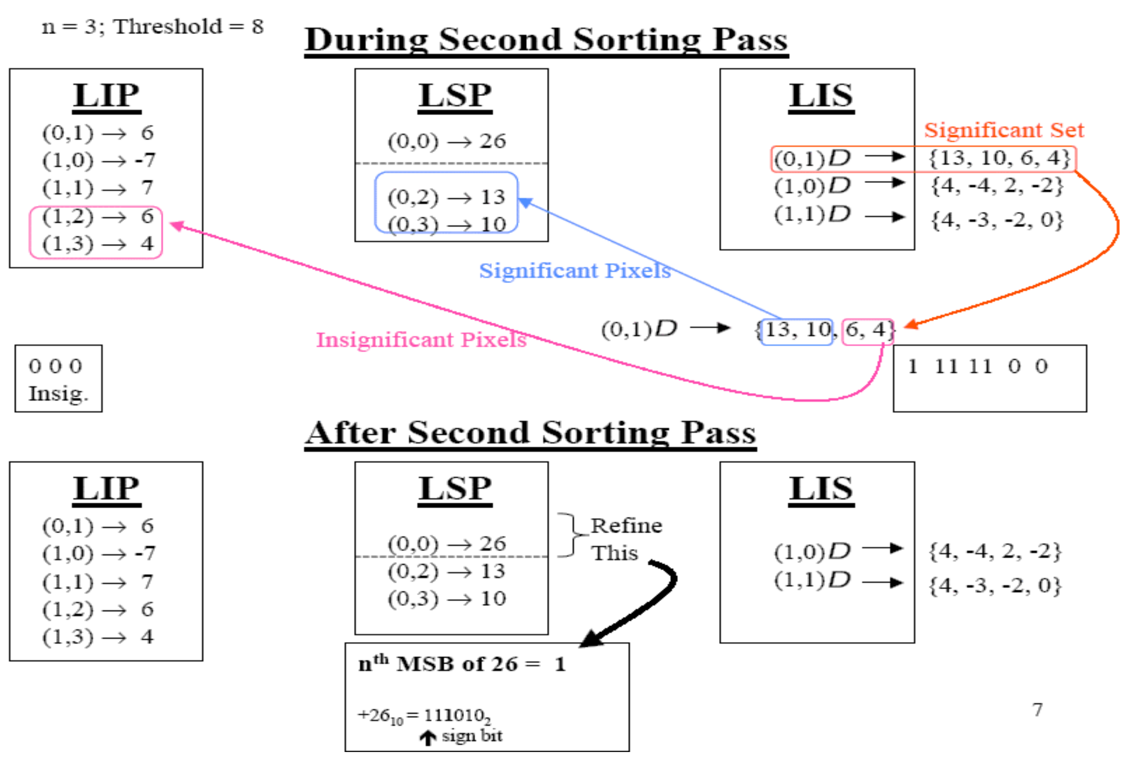


Figure 4.5 2nd sorting and refinement pass

3rd Pass:

N=2 To= 2^2=4

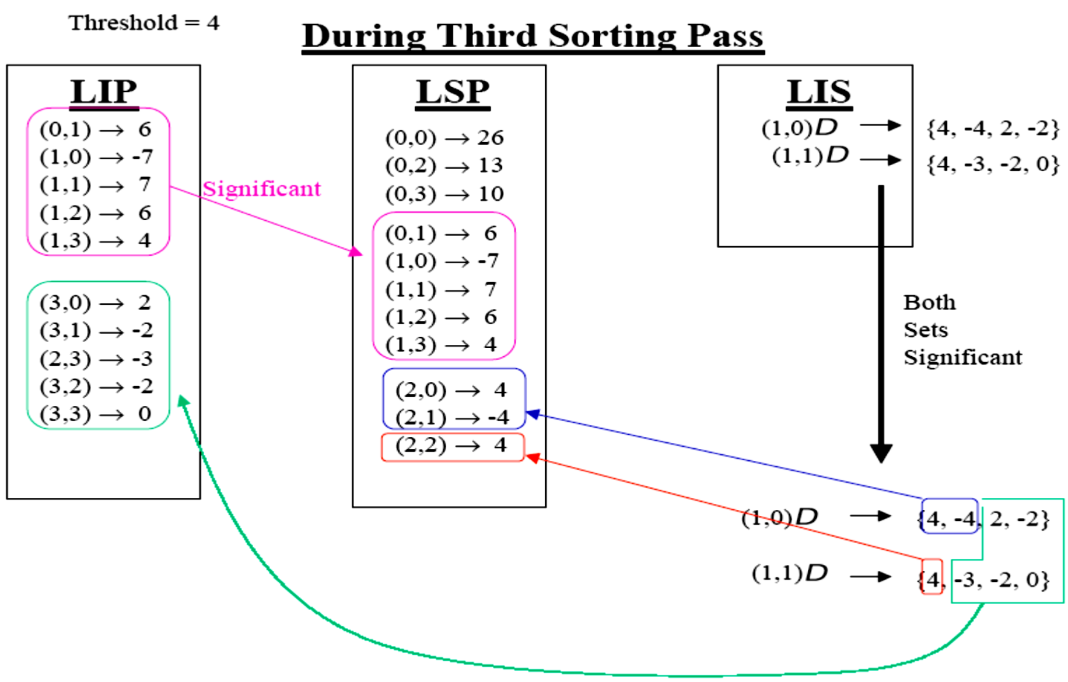
3rd Sorting pass:

Following the same method of checking lists and sets we will get these transmitted bits.

Transmitted bits:

10111010101101100110000010

and lists are as follows:



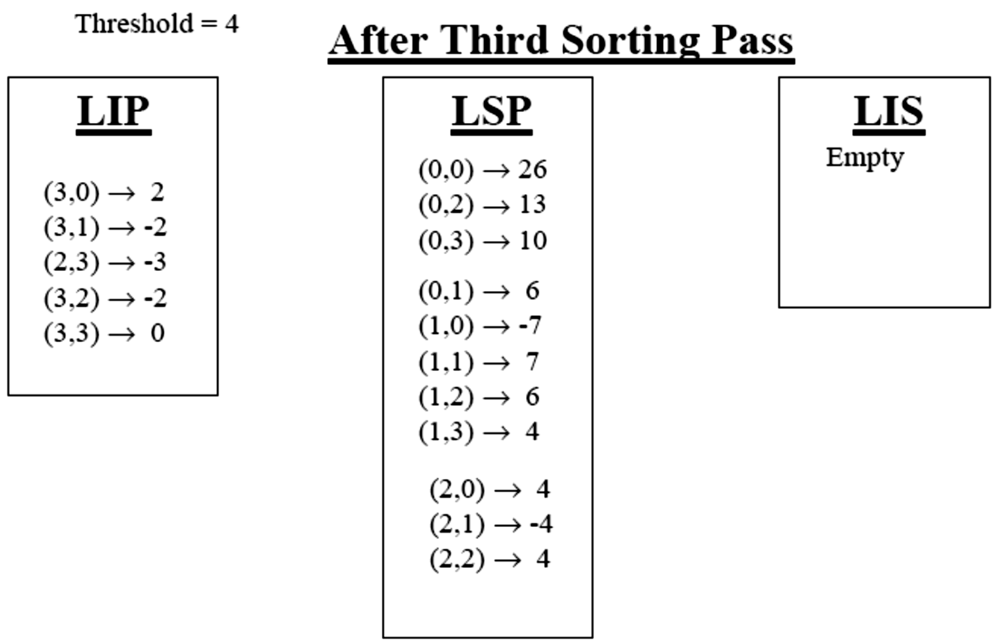


Figure 4.6 **3rd sorting and refinement pass**

Decoding Process:

Initialization:

- LIP: {(0,0), (0,1),(1,0),(1,1)}
- LIS: {D(0,1), D(1,0) , D(1,1)}
- LSP: {}

24	**0**	**0**	**0**
0	**0**	**0**	**0**
0	**0**	**0**	**0**
0	**0**	**0**	**0**

First Pass:

- LIP: {(0,1), (1,0),(1,1)}
- LIS: {D(0,1), D(1,0) , D(1,1)}
- LSP: {(0,0)}

1st Transmitted bits: 11000000

Second Pass:

LIP: {(0,1),(1,0),(1,1),(1,2),(1,3)}

LIS: {D(1,0), D(1,1)}

LSP: {(0,0),(0,2),(0,3)}

2nd Transmitted bits: 000 1 1111 00 00 1

Third pass:

LIP: {(3,0), (3,1)}

LIS: {}

LSP: {(0,0),(0,2),(0,3), (0,1), (1,0), (1,1), (1,2), (2,0), (2,1), (3,2)}

3rd transmitted bits: 11 10 11 11 11 1 11 10 00 1 11 000

4.9 Feature of SPIHT:

These are a few good features of SPIHT:

1. Good PSNR with improved image quality
2. Progressive coded output
3. Efficient Decoding and encoding
4. Can be modified to lossless compression
5. Bit stream can be stopped at any point

4.10 Limitations of Basic SPIHT Technique

The SPIHT divides the wavelets coefficients in three lists for all the quantization levels. Since the information of significance is kind of compact therefore this type of partitioning is very efficient. Moreover, as compare to all existing schemes, even much better performance is attained by binary coded transmission. For a 5-level decomposition of 512x512 and concatenation with arithmetic coding PSNR improves whereas the bit rate remains the same. But, for the same size of image and lesser levels of decomposition the compression is not good because in LL band the pixels are not less. Hence time is consumed. LL band is not reduced if the Images is too small or large.

For small images, LL band is decomposed into smaller size. Decomposition can not be done to many levels as overflow might occur. Arithmetic scheme is of no help here due to increase in time elapsed.

Initially, LIS has all the roots of LL band. If LL sub-band is not small then output is not very efficient.

Modified SPIHT (MSPIHT) algorithm is shared which is based on varying threshold for arrangement. This improves output using different decomposition levels.

ENTROPY CODING
(Huffman Coding)

Chapter 5

Entropy Encoding

5.1 Introduction

Statistical analysis reveals that the output bit-stream of SPIHT comprises of long trail of zeroes which can be further compressed, thus SPIHT is not advocated to be used as sole mean of compression. Images are fed for wavelet transformation, and is compressed by SPIHT followed by entropy encoder (Huffman or arithmetic) to achieve improved compression. In this research we will be carrying out Huffman entropy encoder only. Experimentally concatenation with Huffman is proven to be efficient.

5.2 Huffman Coding

Frequency of pixel occurrence in an image is exploited by Huffman coding. Lower number of bits are used to encode data with higher frequency. A code book is made for the encoding and reconstruction of each image. Code book is shared with the decoder. It has the following features:

1) Fixed symbols for variable length code-words.
2) Symbols are decoded without any repetition i.e. prefix of any code is not used.
3) Every symbol is assigned with a unique code.

5.2.1 Key Principles of Huffman Coding

A few rules used in Huffman coding

I. Least number of bits represent larger probability symbols. And vice versa. Different lengths of code-word are assigned to a fixed group symbols.

II. Huffman coding is a distinctive decoding scheme, as we assign distinctive Code-word to each symbol.

We will concatenate this coding scheme with SPIHT and check results.

5.2.2 Flowchart for Huffman Coding

The flowchart represents the algorithm. Firstly, the symbols are arranged with decreasing probabilities. Two least probable symbols are combined and then assigned a bit ‘1’ and ‘0’ to upper and lower one respectively. Unmerged groups are checked to merge, and code words are assigned.

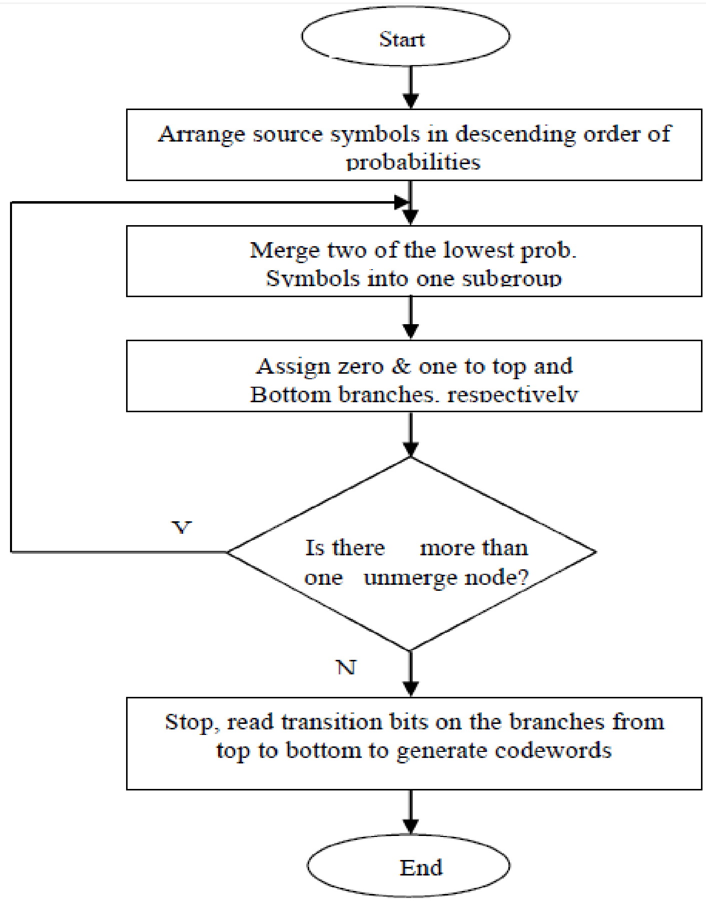


Figure 5.1: Flowchart (Huffman Coding)

5.2.3 Example

Here is an example to apply the algorithm and clarify the concept. Symbols with the occurrence (Table 5.1)".

Symbols	Frequency
222	5
136	7
14	9
2	10
0	100

Table 5.1: Symbols with their respective Frequencies

Step-1: symbols are arranged with respect to decreasing frequencies (Table 5.2).

Symbols	Frequency
0	100
2	10
14	9
136	7
222	5

Table 5.2: Tables w.r.t arranged probabilities

Step-2: two least probable symbols are merged and their occurrence is added (Figure 5.2)".

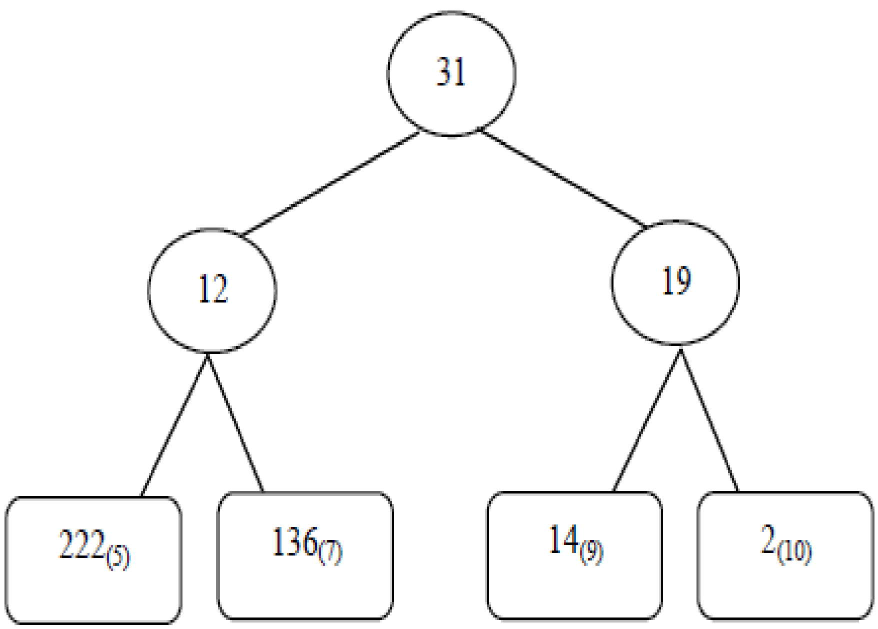


Figure 5.2: Subgroup formation

Step-3: unmerged node is checked (Figure 5.3).

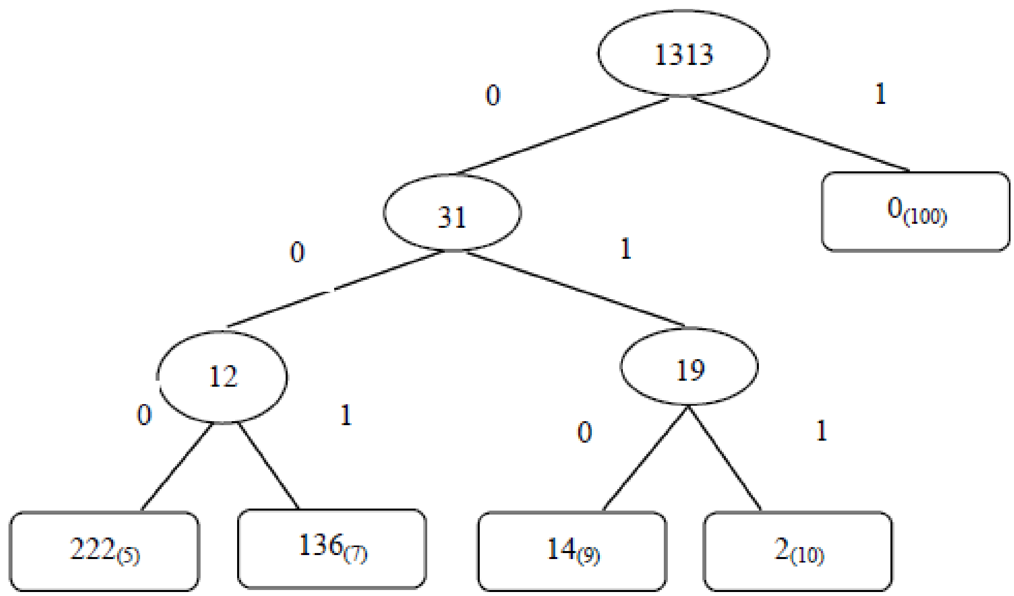


Figure 5.3: Processing of Huffman tree

Step-4: Code-words are assigned to symbols (Table 5.3).

Symbols	**Code-word**	**Frequency**
0	1	100
2	011	10
14	010	9
136	001	7
222	000	5

Table 5.3: Code-words

Results and simulations

Chapter 6

Results and Simulations

6.1 Introduction

In this chapter we will discuss Huffman concatenated with the SPIHT in order to receive better compression results and saving storage space. We will be comparing existing work and our proposed method to exhibit as to how new method saves number of bits without degrading the quality of image while Mean Squared Error (MSE) and PSNR remain unchanged. This research work is an extension of SPIHT cascaded with the Huffman. Efforts are made to improve the bit saving capacity of the aforementioned concatenation by making appropriate use of Huffman entropy coding. The results of existing and proposed methods have been compared by using various perimeters, like bit saving capacity, elapsed timings, MSE, PSNR etc

PSNR is used to compare the distortion. It has been done by using

$$\text{PSNR} = 10\ \log 10_{10}\left(\frac{\left(\max(f(m,n))\right)^2}{\text{MSE}}\right) \tag{6.1}$$

Input image is denoted by $f(m, n)$. usually gray scale images are assumed

$$\text{Max}(f(m,n)) = 255 \tag{6.2}$$

mean squared error (MSE) is calculated between original and reconstructed image as

$$\text{MSE} = \sum_{MN} \frac{f(m,n) - \tilde{f}(m,n)}{M \times N} \tag{6.3}$$

After reconstruction we represent image by $\tilde{f}(m, n)$

6.2 Analysis of Concatenation of SPIHT with Huffman

Let us take the example in Table 6.1(a) and 6.1(b) below, 3 level decomposed image is compressed using one pass of SPIHT. In that case the initial threshold is going to be To = 32 due to largest coefficient being 63. With this, the binary output after 3 passes were bits in all, 11100011100010000001010110000. It is evident from the binary output steam that there is large series of zeros that can be further compressed. Moreover, it has been observed that output of SPIHT has trails of zeros, that can be compressed further. Hence, SPIHT itself is not a good mean of compression.

34.2329	22.9106	8.0819	-9.5783	2.4702	9.6024	17.4720	20.9260
3.1444	0.0473	-10.7578	5.7983	15.2621	5.7212	-6.8773	-26.2526
-9.4979	-7.2971	8.1126	10.0352	10.4049	3.1472	-12.044	-15.2028
6-7.7991	1.9334	12.7445	13.1993	11.3390	6.9783	4.1331	5.5305
12.7476	13.6053	24.2530	24.4590	21.2853	16.8028	13.9673	14.3350
0.4514	7.7005	16.9633	23.2157	20.2790	16.1249	13.9673	14.3350
7.9368	-4.9201	2.2329	0.4116	29.5710	22.8518	22.6126	6.1349

	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 6.1 (a): Original Matrix (b): Dwt coefficients of arbitrary data set

Hence, there is still room for further compression. And that is achieved by cascading Huffman entropy coding with SPIHT.

Statistical analysis reveals that occurrence of '000' is P('000')> 0.25. if we divide the output bit stream into 3 bits, group of 3 bits is recorded as a symbol. Hence, there will be 2^3= 8 kind of symbols. These are encoded using Huffman encoding to attain the improved compression i.e. '000', '001' , '010' , '011', '100' , '101' , '110' and '111' is the division of bit stream into a group of 3. These groups are then fed to Huffman Coder. Remaining bits are then added to the Huffman coder block as header.

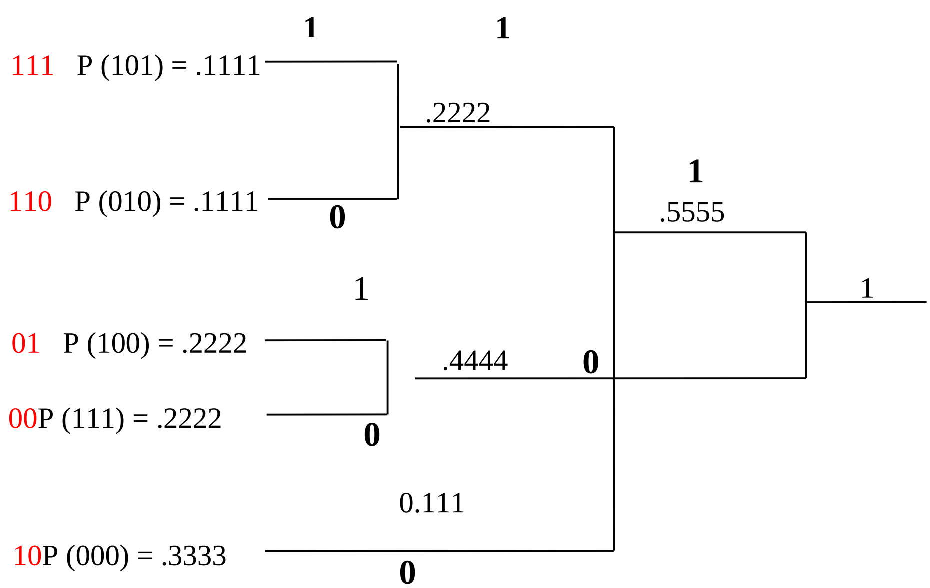
For Leena512 image the probabilities are recorded and a code-word is generated. The symbols along with the occurrence probabilities are shown as under:

Serial number	Symbol	Probability		
1	000	0.333	0.555	0.555
2	111	0.222		
3	100	0.222	0.3331	
4	010	0.1111		0.444
5	101	0.1111	0.1111	
6	110	0		
7	001	0	0	
8	011	0		

Table 6.1: Probabilities and code-word for symbols

The output stream fed to Huffman encoding for further compression is shown below and the codeword book that results according to the probabilities enunciated above, by using Huffman encoding, is as in Table 6.1

For 3 Bits:



Expected Codeword Length = L_C

It can be calculated a tree with minimum weight path length from the root.

$L_c = 1 + 0.5555 + 0.2222 + 0.4444 = 2.222$

$$\mathbf{H = \Sigma P_i \log_2 P_{i=} \; 2.197}$$

$\mathbf{L_C \approx H}$ $\qquad\qquad$ $\mathbf{2.222 \approx 2.197}$

$$\mathbf{L_C \leq H \leq L_C + 1}$$

Var = 2×{0.1111(3-2.222)} + 2×{0.2222 (2-2.2222)} + 0.3333}
= 0.000207

'C00'	→	'01'	'100'	→	'11'
'C01'	→	'1000C0'	'101'	→	'101'
'C10'	→	'1001'	'110'	→	'10001'
'C11'	→	'1000C1'	'111'	→	'00'

Table 6.2: Code-word Table

The output stream of corresponding code is: 10,00,01,00,01,11,01,1001,101,11,00,. These happens to be 25 bits in total. For the remainder bits, 10 lies in the head and appears at the start indicating that two bits were taken along as the remainder and these two bits are '00' that has appeared in the last of the code. It has been seen that 4 bits have been saved after carrying out this concatenation. Decoding can be done by adopting the reverse process.

The output stream fed to Huffman encoding for further compression is shown below and the codeword book that results according to the probabilities enunciated above, by using Huffman encoding, is as in Table 6.1
Transmitted bits are **11100011100010000001010110000**

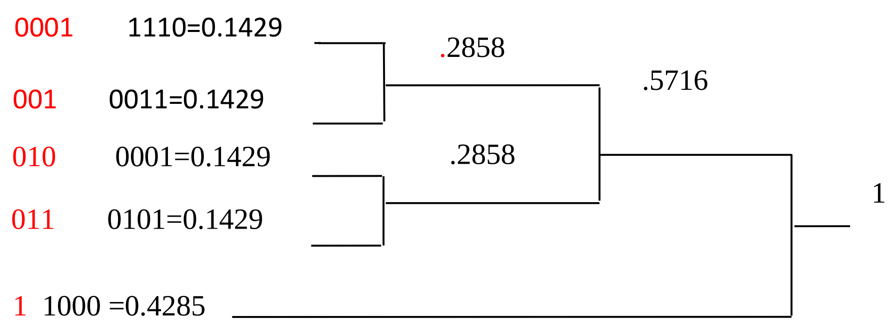
6.3 Improvements:

We saw reduction in bits by concatenation of Huffman with SPIHT. Which can still be improved. Redundancy = Lc-H. where Lc is code length and H entropy.

The largest probability is around 15-20%. If we induce variance in bits, we see that it is preferable only where encoder transmit the compressed data, as it is being generated over the network. If large variance is induced, the bit rate will vary. Hence we have taken varying bit length to see the effect.

<u>For 4 Bits</u>

Data: 1110 0011 1000 1000 0001 0101 1000 0



Compressed data: 0001 001 1 1 010 1 0
Bit saved=29- 14= 15

Expected length of Code

1 + .5710 + .2858 + .2858

$L_C = 2.1432$

H= 2.128 $\qquad$ 2.1432 ≈ 2.128

$\mathbf{L_C \leq H \leq L_C + 1}$

Var = 4× [.1429× (3-2.1432)] + .4285 (1-2.1432)

= - 0.0001612

For 6 Bits

111000 111000 100000 010101 10000

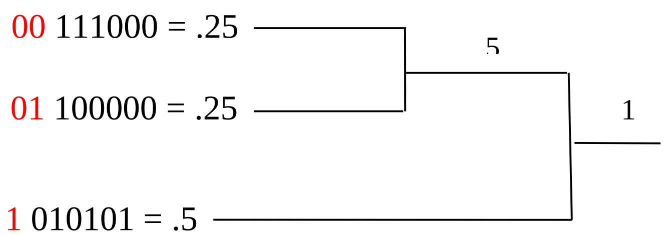
111000	111000	100000	010101	10000
A	A	B	C	Remain

6 Bits mean that will $2^6 = 64$ symbols

P (11000) = 2/4 = .5 P(A)

P (100000) = 1/4 = .25 P(B)

P (010101) = 1/4 = .25 P(C)



Compressed data: 00 00 01 1 10000

Bits saved= 29-12= 17

6.4 Simulations and Results

Simulation results are shared here using MATLAB to check the efficiency of increasing the bit data. Performance parameters like number of bits saved, MSE, PSNR value, Bit Saving Capacity and Elapsed Time or Execution time for algorithm were checked. Tables 6.3-6.7 exhibit the results using same 8x8 DWT matrix that has been used in examples to illustrate EZW and SPIHT in the previous chapters. Table 6.3 - 6.7 enunciate a no of performance measures like output bits, MSE, PSNR, Bit saving capacity and elapsed timings of 8 x 8 matrix of example 6.1 at various given bit rates for 3, 4, 6 bit symbols. Efforts have been made to show the graphical representation of the results in figure 6.1 – 6.5 at 0.1-0.9 bit rates.

3 Bits Symbol

S/No	Rate	Output Bits	MSE	PSNR	Bit Saving Capacity	Elapsed Time (Sec)
1	.2	31	49.8906	31.1506	0.0606	0.351054
2	.4	49	41.6406	31.9356	0.1250	0.413168
3	.6	55	32.6406	32.9932	0.1129	0.391448
4	.8	68	26.2656	33.9369	0.0811	0.386800
5	1.0	78	24.3906	34.2586	0.0824	0.425324

Table 6.3: Performance measures at given bit rates for 3 bits symbols

4 Bits Symbol

S/No	Rate	output Bits	MSE	PSNR	Bit Saving Capacity	Elapsed Time (Sec)
1	.2	21	49.8906	31.1506	0.3636	0.375059
2	.4	30	49.8906	31.1506	0.3750	0.328439
3	.6	49	32.6406	32.9932	0.2097	0.404674
4	.8	59	26.2656	33.9369	0.2027	0.421750
5	1.0	73	24.3906	34.2586	0.1412	0.413899

Table 6.4: Performance measures at given bit rates for 4 bits symbols

6 Bits Symbol

S/No	Rate	output Bits	MSE	PSNR	Bit Saving Capacity	Elapsed Time (Sec)
1	.2	16	49.8906	31.1506	0.5152	0.502989
2	.4	21	49.8906	31.1506	0.5625	0.435720

3	.6	35	32.6406	32.9932	0.4355	0.497648
4	.8	42	26.2656	33.9369	0.4324	0.498170
5	1.0	50	24.3906	34.2586	0.4118	0.501804

Table 6.5: Performance measures at given bit rates for 6 bits symbols

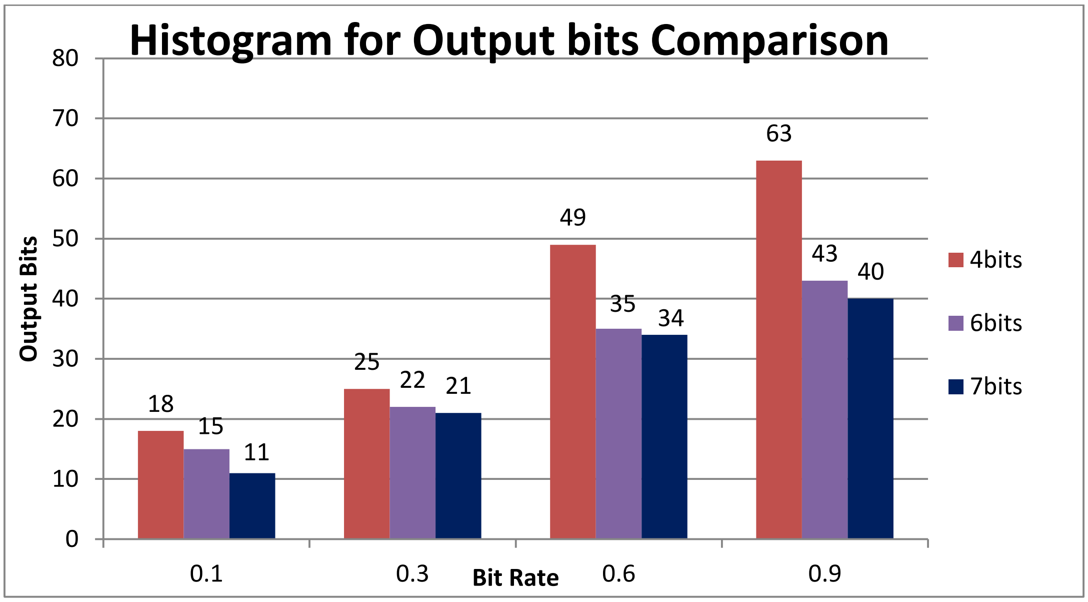


Figure 6.1: Output bit performance at given bit rates for 3, 4, 6 bits per symbol

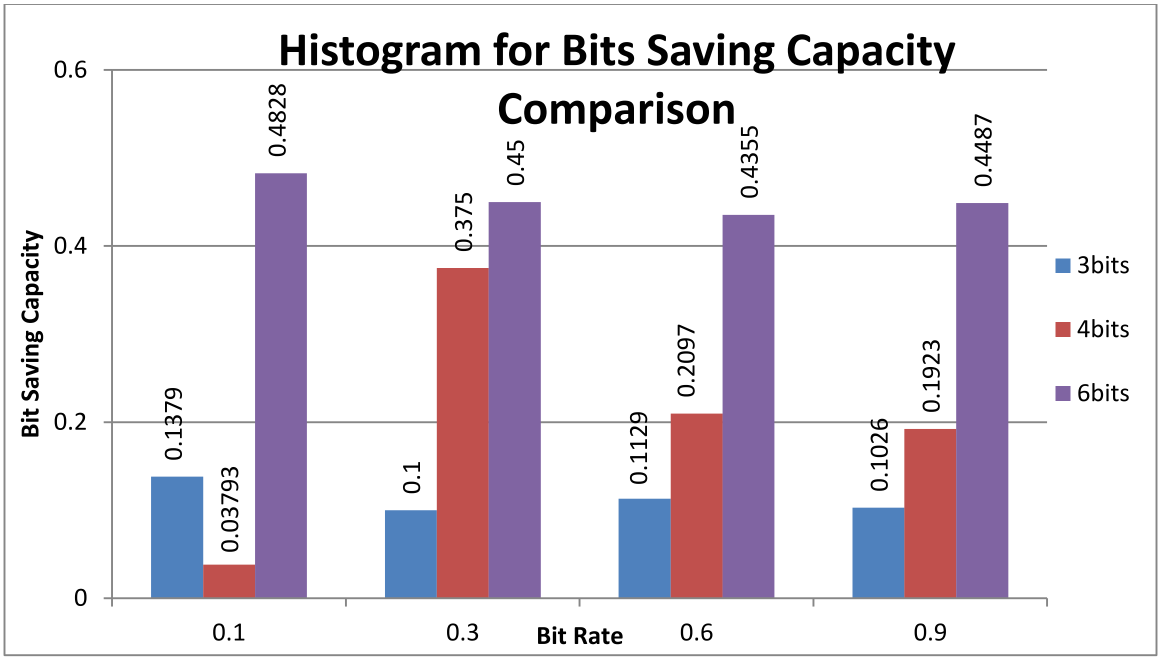


Figure 6.2: Bits saving capacity performance at given bit rates 3, 4, 6 bits symbol

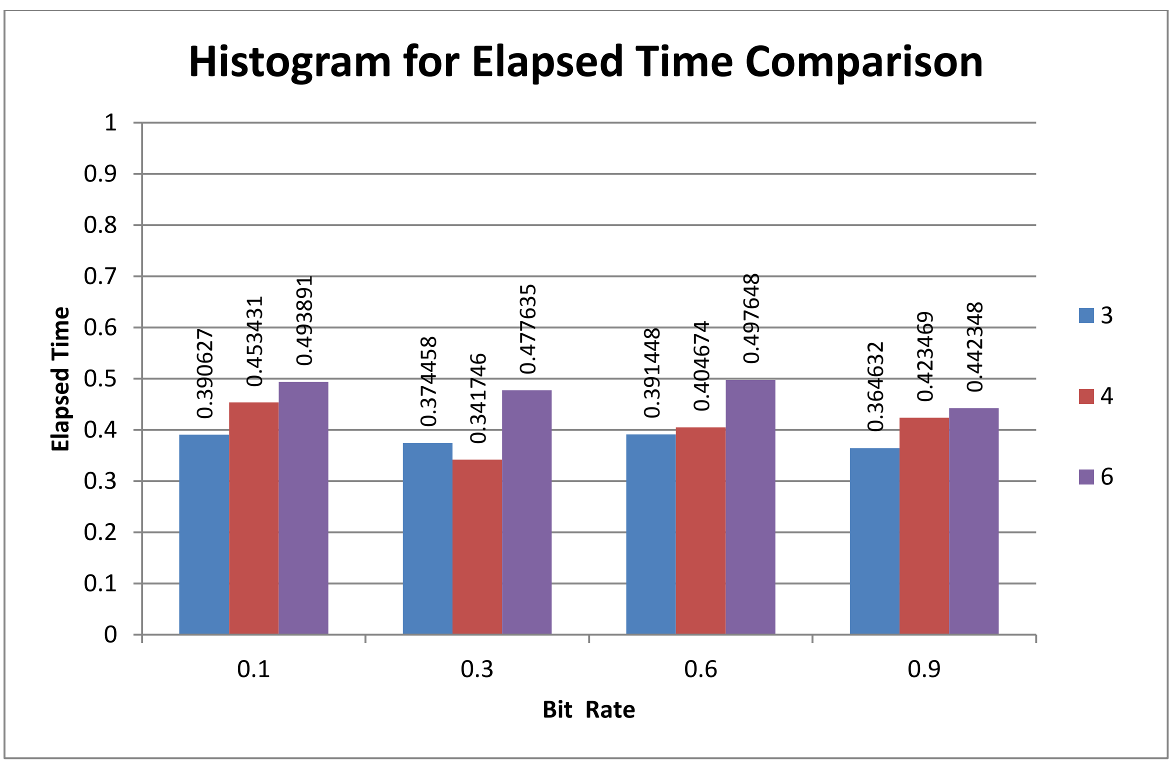


Figure 6.3: Elapsed timing performance at given bit rates 3, 4, 6 bits symbol

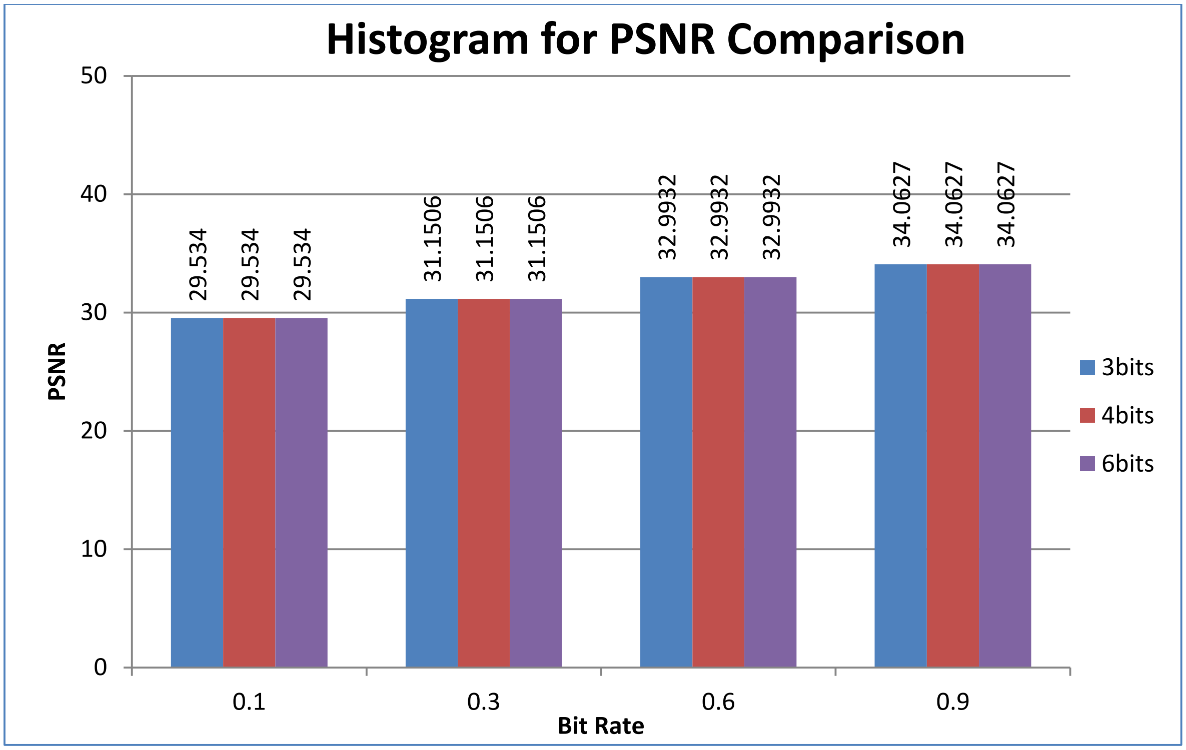


Figure 6.4: PSNR performance at given bit rates 3, 4, 6 bits per symbol

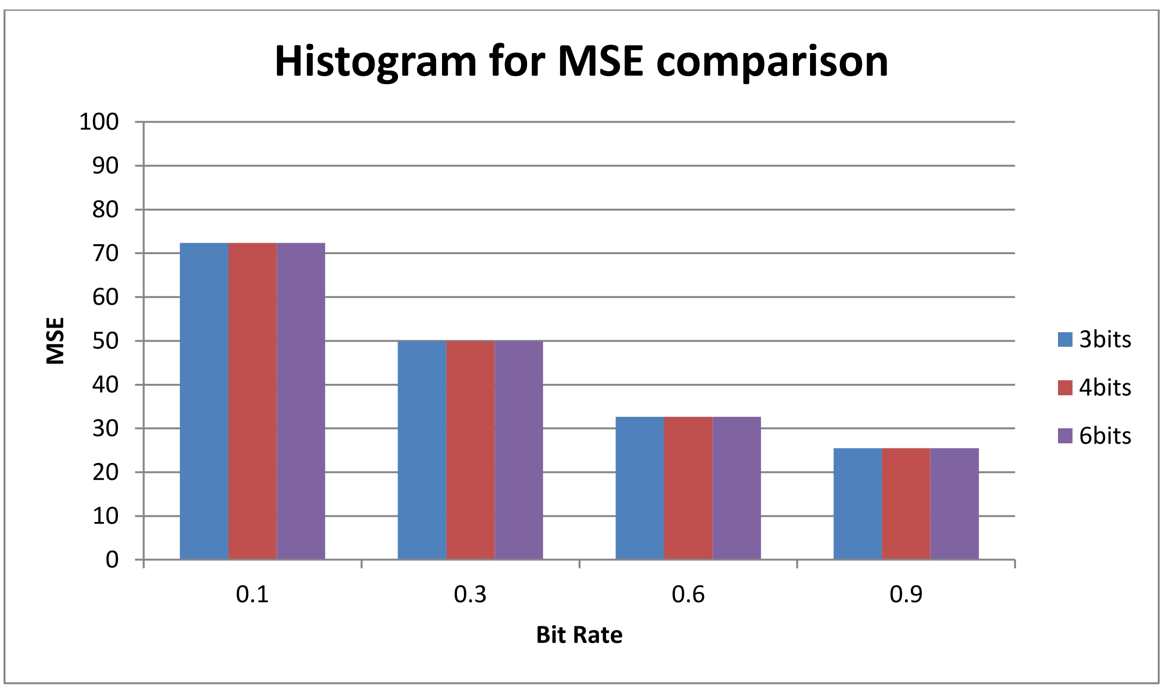


Figure 6.5: MSE performance at given bit rates 3, 4, 6 bits per symbol

More symbol offers improvement in saving no of bits without degrading picture quality of original image where PSNR and MSE remains constant as it can be seen from the results. However, it has been observed there is an increase in the elapsed timings as we proceed further from 3 bits to 4 bits and so on.

Lena 512x512 is taken to check the effect of different bits per symbol.

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

(b) **Reconstructed at 0.2 bpp**

(c) **Reconstructed at 0.4 bpp**

(d) **Reconstructed at 0.6 bpp**

Figure 6.6: Image compression on Lena image of size 512 x 512 using different rates.

6.5 Conclusion

A fully embedded codec algorithm SPIHT is used to compress the already wavelet transformed images followed by Huffman coder. With the increase in bits used for a symbol in Huffman coding it yields better compression ratio thereby enhance the storage space capacity without degradation of the picture quality as MSE and PSNR are constant.

7.2 Future work

In this work picture quality was constant Future work can be done to improve the picture quality instead of these parameters like PSNR being constant.

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] Pandian,1.(2012). Hybrid Algorithm for lossless Image Compression-Survey. International Journal og Emerging Technology and Advanced Engineering (pp. 652-656, Volume 3). Chandigarh: IJETAE

[3] Qi-bin, F. (2008). WaveletAnalysis. Wuhan:: Wuhan University Press.

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

[5] 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.

[6] 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.

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

[8] 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.

[9] Nicholas Kolokotronis, AlikiVassilarakou, SergiosTheodoridis, DionisisCavouras "Wavelet-based medical image compression EleftheriosKofidis" Department of Informatics, Division of Communications and Signal Processing, University of Athens, Panepistimioupolis, TYPA Buildings, GR-15784 Athens, Greece .

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

[11] Penedo, S.R.M., Netto, M.L. & Justo, J.F. Designing digital filter banks using wavelets. *EURASIP J. Adv. Signal Process.* 2019, 33 (2019). https://doi.org/10.1186/s13634-019-0632-6

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

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

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

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

[16] “Introduction to Data Compression” By Khalid Sayood. 5th Edition. Issued: 2017 ISBN: 9780128094747

[17] 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.

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

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

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

[21] 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.

[22] 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.

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

[24] Oktem, Levent, RusenOktem, Karen Eguiazarian, and JaakkoAstola. "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.

[25] 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.

[26] 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.

[27] 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.

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

[29] 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.

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

[31] 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.

[32] 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)