Compression Efficiency for Combining Different Embedded Image Compression Techniques with Huffman Encoding

Thesis submitted in partial fulfillment of the requirements for the degree

of

Master of Technology

i*n*

Electronics and Communication Engineering

(Specialization: Electronics and Instrumentation)

by

Sure Srikanth

Roll No: 211EC3320

Department of Electronics & Communication Engineering

National Institute of Technology Rourkela

Rourkela, Odisha-769008

Compression Efficiency for Combining Different Embedded Image Compression Techniques with Huffman Encoding

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Electronics and Communication Engineering

(Specialization: Electronics and Instrumentation)

by

Sure Srikanth

Roll No: 211EC3320

Under the Supervision of

Dr. Sukadev Meher

Department of Electronics & Communication Engineering

National Institute of Technology Rourkela

Rourkela, Odisha-769008

Dedicated to

My Family

Department of Electronics and Communication Engg
National Institute of Technology Rourkela
Rourkela - 769008, Odisha, India.

CERTIFICATE

This is to certify that the Thesis Report titled "**Compression Efficiency for Combining Different Embedded Image Compression Techniques with Huffman Encoding**", submitted by **Mr. Sure Srikanth** bearing roll no**. 211EC3320** in partial fulfillment of the requirements for the award of **Master of Technology in Electronics and Communication Engineering** with specialization in "**Electronics and Instrumentation**" during session 2011 - 2013 at National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

.

Place: Rourkela
Date: 31st May, 2013

Prof. (Dr.) SUKADEV MEHER

Acknowledgement

The real spirit of achieving a goal is through the way of excellence and austere discipline. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by various personalities.

Foremost, I would like to express my sincere gratitude to my advisor, ***Prof. Sukadev Meher*** *for providing me with a platform to work on challenging areas of Image Compression, Compression Efficiency for Combining Different Embedded Image Compression Techniques with Huffman Encoding. His profound insights and attention to details have been true inspirations to my research.*

I am also thankful to all the professors of the Electronics & Communication department for their constant support and encouragement.

I would like to thank all my friends and lab–mates for their constant encouragement and understanding. Their help can never be penned with words.

Most importantly, none of this would have been possible without the love and patience of my family. My family, to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heartfelt gratitude to them.

Sure Srikanth

211ec3320

Abstract

This thesis presents a technique for image compression which uses the different embedded Wavelet based image coding in combination with Huffman- encoder(for further compression). There are different types of algorithms available for lossy image compression out of which Embedded Zerotree Wavelet(EZW), Set Partitioning in Hierarchical Trees (SPIHT) and Modified SPIHT algorithms are the some of the important compression techniques. EZW algorithm is based on progressive encoding to compress an image into a bit stream with increasing accuracy. The EZW encoder was originally designed to operate on 2D images, but it can also use to other dimensional signals. Progressive encoding is also called as embedded encoding. Main feature of ezw algorithm is capability of meeting an exact target bit rate with corresponding rate distortion rate(RDF).

Set Partitioning in Hierarchical Trees (SPIHT) is an improved version of EZW and has become the general standard of EZW. SPIHT is a very efficient image compression algorithm that is based on the idea of coding groups of wavelet coefficients as zero trees. Since the order in which the subsets are tested for significance is important in a practical implementation the significance information is stored in three ordered lists called list of insignificant sets (LIS) list of insignificant pixels (LIP) and list of significant pixels (LSP). Modified SPIHT algorithm and the preprocessing techniques provide significant quality (both subjectively and objectively) reconstruction at the decoder with little additional computational complexity as compared to the previous techniques. This proposed method can reduce redundancy to a certain extend. Simulation results show that these hybrid algorithms yield quite promising PSNR values at low bitrates.

Contents:

LIST OF FIGURES:

LIST OF Tables:

ABBREVIATIONS:

DICOM	Digital Imaging and Communications in Medicine
JPEG	Joint Photographic Experts Group
SPIHT	Set Partioning in Hierarchical Trees
ISPIHT	Improved Set Partioning in Hierarchical Trees
ROI	Region of Interest
CT	Computed Tomography
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
WT	Wavelet Transform
LIS	List of Insignificant Sets
LIP	List of Insignificant Pixels
LSP	List of Significant Pixels
PSNR	Peak Signal to Noise Ratio
MSE	Mean Squared Error
US	Ultrasound
PROI	Primary Region of Interest
SROI	Secondary Region of Interest

Chapter- 1

Introduction

Basic Steps of Image Compression

History

Literature Survey

Objective of Thesis

Thesis Organization

1. INTRODUCTION

Now a day's Medical science is growing very fast and hence each hospital needs to store high volume of information or data about the patients. And medical images are one of the most important data about patients. As a result hospitals have a high volume of images with them and require a huge hard disk space and transmission bandwidth to store these images. Most of the time transmission bandwidth has not sufficient into storing all the image data. Image compression is the process of encoding the information using fewer bits (or other information-bearing units) than an un-encoded representation which would use of specific encoding schemes. Compression is mainly useful to reduce the consumption of expensive memory or resources, such as hard disk space or transmission bandwidth (computing). On the downside, compressed data must be decompressed, and this extra processing may be detrimental to some applications. For instance, a compression scheme for image may require expensive hardware for the image to be decompressed fast enough to be viewed as its being decompressed (the option of decompressing the image in full before watching it may be inconvenient, and requires storage space for the decompressed image). The design of data compression schemes therefore involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (if using a lossy compression scheme), and the computational resources required to compress and uncompressed the data [1][2]. Data compression is the process of converting data files into smaller ones for efficiency of storage and transmission. Each compression algorithm has its corresponding decompression algorithm provides compressed file, should reproduce the original one. Image compression schemes come under two categories: lossless and lossy compression. Lossless compression uses coding techniques to compress the data while retaining all information content. However, in this case the achieved file size reduction is not sufficient for many applications. Lossy image compression, as its name implies, output image is the loss of some information content while the file size reduction can be much more significant than the lossless image compression. Since most digital images are intended for human observers, much research is nowadays focused on lossy image compression that minimizes visual distortion and possibly obtains visually lossless results. Image compression is the application of Data compression on digital images. Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages. This would imply the need for a compression scheme that would give a very high compression ratio usually comes with a price. This refers to the quality of the image. Given a particular compression ratio, the quality of the image reconstructed using the SPHIT algorithm make the job more better way even though the problem comes into picture is the issue of energy constraints. While compressing and transmitting an image if the coefficients to be transmitted are of very large magnitude then more resources would be required for transmission. This is taken care of by employing energy efficient compression. But again medical images cannot afford the loss of important details for the sake of meeting battery constraints for telemedicine. This is also taken care of in this work and the regions of diagnostic importance are undisturbed in course of achieving energy efficiency. Another important characteristic, which is often the criterion or even the referred basis of diagnosis in medical images, is the texture of the various regions within the image.

1.1. Basic Steps of Image Compression:

There are 2 types of image compression: lossless compression (reversible) and lossy compression (irreversible) Run-length encoded (RLE) and the JPEG lossless compression algorithms are examples of lossless compression [2][3][4]. In lossy compression, data are discarded during compression and cannot be recovered. Lossy compression achieves much greater compression than does lossless technique. Wavelet and higher-level JPEG are examples of lossy compression. JPEG 2000 is a progressive lossless-to-lossy compression algorithm. JPEG handles only still images, but there is a related standard called MPEG for motion pictures.

- **Compression Algorithms:** There are 3 basic steps:

  1. **Transformation**: The discrete wavelet transform cuts the image into blocks of 64 pixels (8 × 8) and processes each block independently, shifting and simplifying the colours so that there is less information to encode.

  2. **Quantization**: The values in each block are then divided by a quantization coefficient. This is the compression step where information loss occurs. Pixels are changed only in relation to the other pixels within their block.

  3. **Encoding**: The reduced coefficients are then encoded, usually with Huffman coding (entropy encoding that finds the optimal system of encoding based on the relative frequency of each character).With high ratio compression.

1.2. History:

In 1949 Claude Shannon and Robert Fano devised a systematic way to assign code words based on probabilities of blocks. An optimal method for doing this was then found by David Huffman in 1951. Early implementations were typically done in hardware, with specific choices of code words being made as compromises between compression and error correction. In the mid-1970s, the idea emerged of dynamically updating code words for Huffman encoding, based on the actual data encountered. And in the late 1970s, with online storage of text files becoming common, software compression programs began to be developed, almost all based on adaptive Huffman coding. In 1977 Abraham Lempel and Jacob Ziv suggested the basic idea of pointer-based encoding. In the mid-1980s, following work by Terry Welch, the so-called LZW algorithm rapidly became the method of choice for most general-purpose compression systems. It was used in programs such as PKZIP, as well as in hardware devices such as modems. In the late 1980s, digital images became more common, and standards for compressing them emerged. In the early 1990s, lossy compression methods also began to be widely used.

1.3. Literature survey:

Walter B. Richardson, revised the challenges faced as technology moved toward digital mammography, presented a necessarily brief overview of multiresolution analysis, and finally, gave current and future applications of wavelets to several areas of mammography.

Armando Manduca, have developed software modules (both stand-alone and in the biomedical image analysis and display package analyze) that could perform wavelet- based compression on both 2-D and 3-D gray scale images. He presented examples of such compression on a variety of medical images, and comparisons with JPEG and other compression schemes.

Arve Kjoelen et.al, studied the rapid growth in the field of diagnostic imaging has produced several new class of digital images, resulting from computerized tomography, magnetic resonance imaging, and other imaging modalities. In addition, well-established imaging modalities such as X-rays and ultrasound will increasingly be processed and stored in a digital format. It has been estimated that a 600-bed hospital would need almost 2 terabytes (2,000 gigabytes) of storage per year if all images produced at the hospital were to be stored in a digital format. The need for high performance compression algorithms to reduce storage and transmission costs is evident. The compression techniques described herein were likely to be effective for a far wider range of images than the skin tumor images employed in this research.

Mislav GrgiC et.al, discussed the features of wavelet filters in compression of still images and characteristic that they showed for various image content and size. The aim of this work was to create a palette of functions (filters) for implementation of wavelet in still image processing and to emphasize the advantage of this transformation relating to today's methods. Filters taken in the test are some of the most used: Haar filter (as the basis), orthogonal and Biorthogonal filter. All these filters gave various performances for images of different content. Objective and subjective picture quality characteristics of images coded using wavelet transform with different filters were given. The comparison between JPEG coded picture and the same picture coded with wavelet transform was given. For higher compression ratios it was shown that wavelet transform had better S/N.

Reto Grüter et.al, addressed the problem of progressive lossless image coding, A nonlinear decomposition for progressive lossless compression was presented. The decomposition into sub bands was called rank-order polynomial decomposition (ROPD) according to the polynomial prediction models used. The decomposition method presented here was a further development and generalization of the morphological sub band decomposition (MSD) introduced earlier by the same research group. It was shown that ROPD provides similar or slightly better results than the compared coding schemes such as the codec based on set partitioning in hierarchical trees (SPIHT). The proposed lossless compression scheme had the functionality of having a completely embedded bit stream, which allowed for data browsing. It was shown that the ROPD had a better lossless rate than the MSD but it had also a much better browsing quality when only a part of the bit stream is decompressed. Finally, the possibility of hybrid lossy/lossless compression was presented using ultrasound images. As with other compression algorithms, considerable gain could be obtained if only the regions of interest are compressed lossless.

Wael Badawy et.al, found that with the recent explosion of the Internet, the implementation of technologies such as telemedicine, video conferencing, and wireless data communication have been constrained due to the Internet's finite bandwidth. With the limited technology at hand, techniques were needed to compress a large amount of data into a feasible size before transmission. Data compression had even a greater importance in many applications that involved storage of large data sets, such as magnetic resonance imaging, digital television, and seismic data collection. There were number proposed compression techniques in the literature, but none had unique properties of sub band coding algorithms. One such technique presented here was the discrete wavelet transform (DWT), which was one of the most efficient compression algorithms because of its perfect reconstruction property .

Karthik Krishnan et.al, studied that the goals of telemedicine was to enable remote visualization and browsing of medical volumes. There was a need to employ scalable compression schemes and efficient client-server models to obtain interactivity and an enhanced viewing experience. First, they presented a scheme that used JPEG2000 and JPIP (JPEG2000 Interactive Protocol) to transmit data in a multi-resolution and progressive fashion. JPEG2000 for remote volume visualization and volume browsing applications. The resulting system was ideally suited for client-server applications with the server maintaining the compressed volume data, to be browsed by a client with a low bandwidth constraint.

Charalampos Doukas et.al, studied that Medical imaging had a great impact on medicine, especially in the fields of diagnosis and surgical planning. However, imaging devices continue to generate large amounts of data per patient, which require long-term storage and efficient transmission. Current compression schemes produce high compression rates if loss of quality is considerable. However, in most of the cases physicians may not afford any deficiency in diagnostically important regions of images, called regions of interest (ROIs). An approach that brings a high compression rate with good quality in the ROI was thus necessary. The general theme was to preserve quality in diagnostically critical regions while allowing lossy encoding of the other regions. The aim of the research focused on ROI coding is to allow the use of multiple and arbitrarily shaped ROIs within images, with arbitrary weights describing the degree of importance for each ROI including the background (i.e., image regions not belonging to ROI) so that the latter regions may be represented by different quality levels. In this context, this article provided an overview of state-of the- art ROI coding techniques applied on medical images. These techniques are classified according to the image type they apply to; thus the first class included ROI coding schemes developed for two-dimensional (2- D) still medical images whereas the second class consists of ROI coding in the case of volumetric images. In the third class, a prototype ROI encoder for compression of angiogram video sequences is presented.

In 2008 Ultrasound, computed tomography (CT), magnetic resonance imaging (MRI) medical imaging produce human body pictures in digital form. These medical applications have already been integrated into mobile devices and are being used by medical personnel in treatment centers, for retrieving and examining patient data and medical images. Storage and transmission are key issues in such platforms, due to the significant image file sizes. Wavelet transform has been considered to be a highly efficient technique of image compression resulting in both lossless and lossy compression of images with great accuracy, enabling its use on medical images. On the other hand, in some areas in medicine, it may be sufficient to maintain high image quality only in the region of interest i.e. in diagnostically important regions. This paper proposes a framework for ROI based compression of medical images using wavelet based compression techniques (i.e. JPEG2000 and SPIHT). Results are analyzed by conducting the experiments on a number of medical images by taking different region of interests. The performance is evaluated using various image quality metrics like PSNR, MSE.

In 2009, Dr.R.Sudhakar, Advanced medical imaging requires storage of large quantities of digitized clinical data [22]. Due to the constrained bandwidth and storage capacity, a medical image must be compressed before transmission and storage. However, the compression will reduce the image fidelity, especially when the images are compressed at lower bit rates. The reconstructed images suffer from blocking artifacts and the image quality will be severely degraded under the circumstance of high compression ratios. Medical imaging poses the great challenge of having compression algorithms that reduce the loss of fidelity as much as possible so as not to contribute to diagnostic errors and yet have high compression rates for reduced storage and transmission time. To meet this challenge several hybrid compression schemes exclusively for medical images are developed in the recent years. This paper presents overview of various compression techniques based on DCT, DWT and ROI.

In 2009 the proposed algorithm presents an application of SPIHT algorithm to color volumetric dicom medical images using wavelet decomposition [7]. The wavelet decomposition is accomplished with biorthogonal 9/7 filters and 5/3 filters. SPIHT is the modern-day benchmark for three dimensional image compressions. The three-dimensional coding is based on the observation that the sequences of images are contiguous in the temporal axis and there is no motion between slices. Therefore, the discrete wavelet transform can fully exploit the inter-slices correlations. The set partitioning techniques involve a progressive coding of the wavelet coefficients. The SPIHT is implemented and the Rate-distortion (Peak Signal-to-Noise Ratio (PSNR) vs. bit rate) performances are presented for volumetric medical datasets by using biorthogonal 9/7. The results are compared with the previous results of JPEG 2000 standards. Results show that SPIHT method exploits the color space relationships as well as maintaining the full embeddedness required by color image sequences compression and gives better performance in terms of the PSNR and compression ratio than the JPEG 2000.

In 2010 Most of the commercial medical image viewers do not provide scalability in image compression and/or encoding/decoding of region of interest (ROI) [27]. This paper discusses a medical application that contains a viewer for digital imaging and communications in medical (DICOM) images as a core module. The proposed application enables scalable wavelet-based compression, retrieval, and decompression of DICOM medical images and also supports ROI coding/decoding. Furthermore, the presented application is appropriate for use by mobile devices activated in a heterogeneous network. The methodology involves extracting a given DICOM image into two segments, compressing the region of interest with a lossless, quality sustaining compression scheme like JPEG2000, compressing the unimportant regions (background, et al.,) with an algorithm that has a very high compression ratio and that does not focus on quality (SPIHT). With this type of the compression work, energy efficiency is achieved and after respective reconstructions, the outputs are integrated and combined with the output from a texture based edge detector. Thus the required targets are attained and texture information is preserved.

In 2010 an **Improved SPIHT** algorithm based on double significant criteria according to define relation between a threshold value and a boundary rate-distortion slope [2], The significant coefficient and trees has been chosen. The selected significant coefficient and trees are quantized.

Yumnam Kirani Singh Proposed a new sub band coding scheme entitled **ISPIHT** (Improved SPIHT). It is simpler in its coding approach yet it is more efficient in time and memory keeping the performance of SPIHT preserved [3]. It requires less no of compression operations during the coding. The memory requirement for ISPIHT is about two times less than SPIHT.

Yin-hua Wu, Long-xu Jin studied that the current stringent need to the real-time compression algorithm of the high-speed and high-resolution image, such as remote sensing or medical image and so on, in this paper, No List SPIHT (NLS) algorithm has been improved, and a fast parallel SPIHT algorithm is proposed, which is suitable to implement with FPGA [5]. It can deal with all bit-planes simultaneously, and process in the speed of 4pixels/period, so the encoding time is only relative to the image resolution. The experimental results show that, the processing capacity can achieve 200MPixels/s, when the input clock is 50MHz, the system of this paper need 2.29ms to complete lossless compression of a 512×512×8bit image, and only requires 1.31ms in the optimal state. The improved algorithm keeps the high SNR unchanged, increases the speed greatly and reduces the size of the needed storage space. It can implement lossless or lossy compression, and the compression ratio can be controlled. It could be widely used in the field of the high-speed and high-resolution image compression.

1.4. OBJECTIVE OF THE THESIS:

The amount of data associated with visual information is so large that its storage would require enormous storage capacity. Although the capacities of several storage media are substantial, their access speeds are usually inversely proportional to their capacity. Typical television images generate data rates exceeding 10 million bytes per second. There are other image sources that generate even higher data rates. Storage and/or transmission of such data require large capacity and/or bandwidth, which could be very expensive. Image data compression techniques are concerned with reduction of the number of bits required to store or transmit images without any appreciable loss of information.

The basic objective of the thesis is to reduce the memory size of original image without degrading the quality, by using different lossy and lossless image compression techniques. Already some of the researchers implemented compressing the original image using either lossless or lossy technique. The proposed research work aims to combine the different embedded lossy image compression techniques with Huffman encoding.

1.5. THESIS ORGANIZATION:

The remaining part of the thesis is organized as follows. **Chapter 2** presents the basic concepts of wavelets, wavelet transform, purpose of wavelets, classification of wavelet transform, properties and architecture of wavelet. **Chapter 3** presents introduction about image compression, Techniques of Image compression, different types of lossy image compression algorithms and Terms use in Image Compression. **Chapter 4** presents the implementation of different lossy image compression algorithms with Huffman encoding. **Chapter 5** presents the experimental results of different lossy image compression algorithms with & without Huffman encoding. Finally **chapter 6** concludes the thesis work with the suggestions for future research.

Chapter- 2

Wavelet Transform

Wavelet Analysis

Purpose Wavelet Analysis

Wavelet Transform

Reason for preferring Wavelet

Classification of Wavelet Transform

Property of Wavelet

Architecture of Wavelet

2.Wavelet Transforms:

INTRODUCTION TO WAVELETS AND WAVELET TRANSFORMS:

2.1. Wavelet Analysis:

Wavelet analysis represents the next logical step: a windowing technique with variable-sized regions [5]. Wavelet analysis allows the use of long time intervals where more precise low frequency information is required and shorter regions where high-frequency information.

Fig 2.1 Wavelet Analysis

Here's what this looks like in contrast with the time-based, frequency-based, and STFT views of a signal: You may have noticed that wavelet analysis does not use a time-frequency region, but rather a time-scale region.

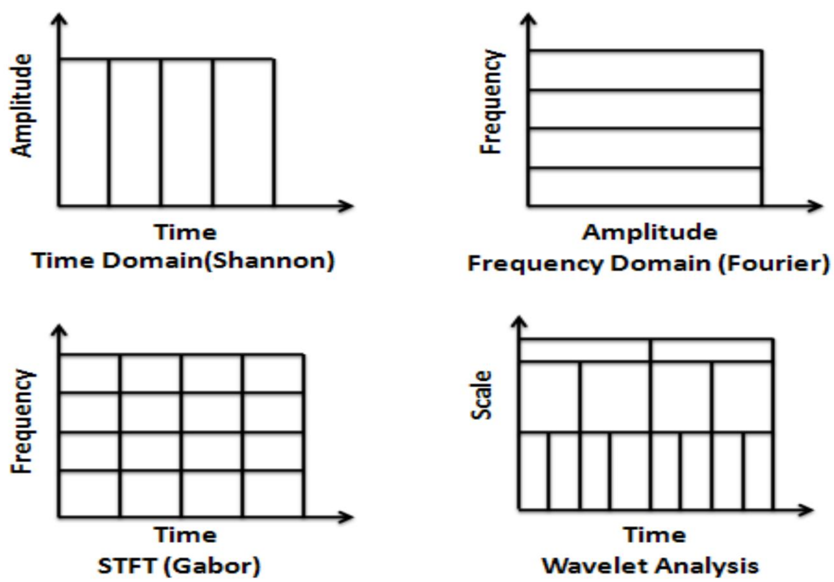


Fig 2.2: Time-based, frequency-based, and STFT views of a signal

2.2. WAVELET:

Wavelet means a wavelet is a waveform of effectively limited duration that has an average value of zero [5][6]. Compare wavelets with sine waves, which are the basis of Fourier analysis. Sinusoids do not have limited duration they extend from minus to plus infinity. And where sinusoids are smooth and predictable, wavelets tend to be irregular and symmetric.

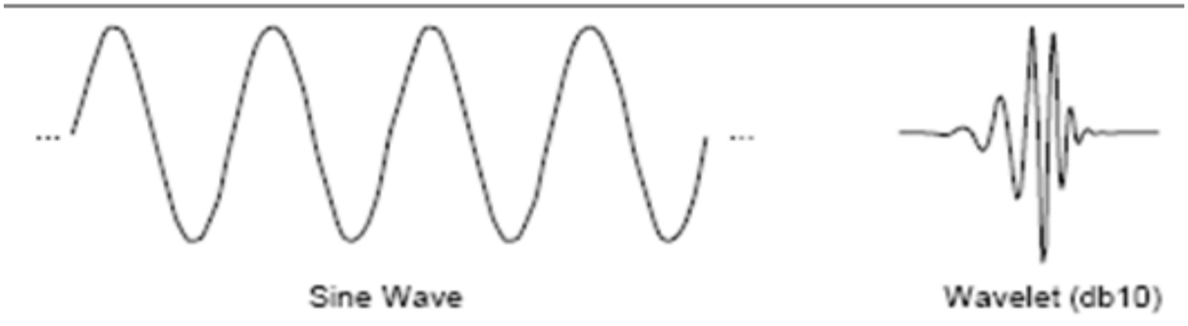


Fig 2.3: Sinusoidal wave

Fourier analysis consists of breaking up a signal into sine waves of various frequencies. Similarly, wavelet analysis is the breaking up of a signal into shifted and scaled versions of the original (or mother) wavelet. Just looking at pictures of wavelets and sine waves, you can see intuitively that signals with sharp changes might be better analyzed with an irregular wavelet than with a smooth sinusoid, just as some foods are better handled with a fork than a spoon.

2.3. Purpose of Wavelet Analysis:

One major advantage afforded by wavelets is the ability to perform local analysis, that is, to analyze a localized area of a larger signal [6][19]. Consider a sinusoidal signal with a small discontinuity one so tiny as to be barely visible. Such a signal easily could be generated in the real world, perhaps by a power fluctuation or a noisy switch. However, a plot of wavelet coefficients clearly shows the exact location in time of the discontinuity.

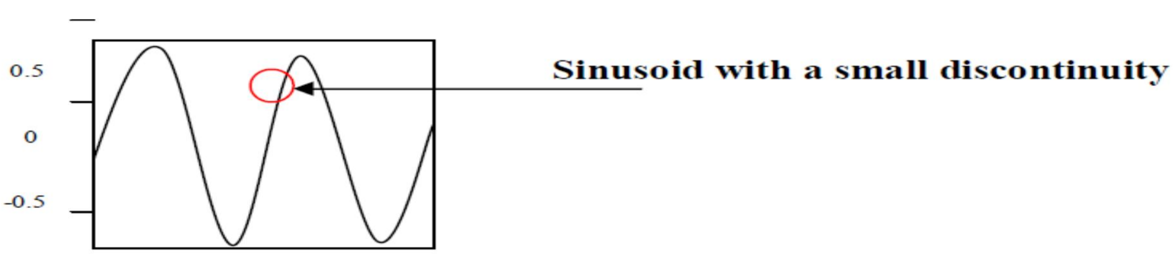


Fig 2.4 Sinusoidal Signal

2.4. Wavelet Transform:

When the signal in time for its frequency content is analyzed, Unlike Fourier analysis, in which signals using sines and cosines are analyzed, wavelet functions is used.

- **The Continuous Wavelet Transform**

Mathematically, the process of Fourier analysis is represented by the Fourier transform:

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

Which is the sum over all time of the signal f(t) multiplied by a complex exponential. (Recall that a complex exponential can be broken down into real and imaginary sinusoidal components) [20] The results of the transform are the Fourier coefficients F(w), which when multiplied by a sinusoid of frequency w yields the constituent sinusoidal components of the original signal. Graphically, the process looks like:

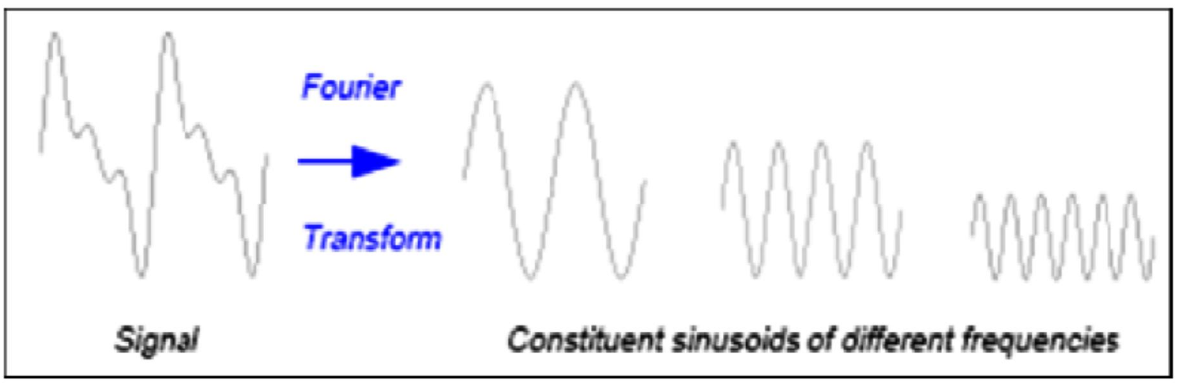


Fig 2.5 Continuous Wavelets of different frequencies

Similarly, the continuous wavelet transform (CWT) is defined as the sum over all time of signal multiplied by scaled, shifted versions of the wavelet function.

$$C(scale, position) = \int_{-\infty}^{\infty} f(t)\psi(scale, position, t)dt$$

The result of the CWT is a series many wavelet coefficients **C**, which are a function of scale and position**.** Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelets of the original signal:

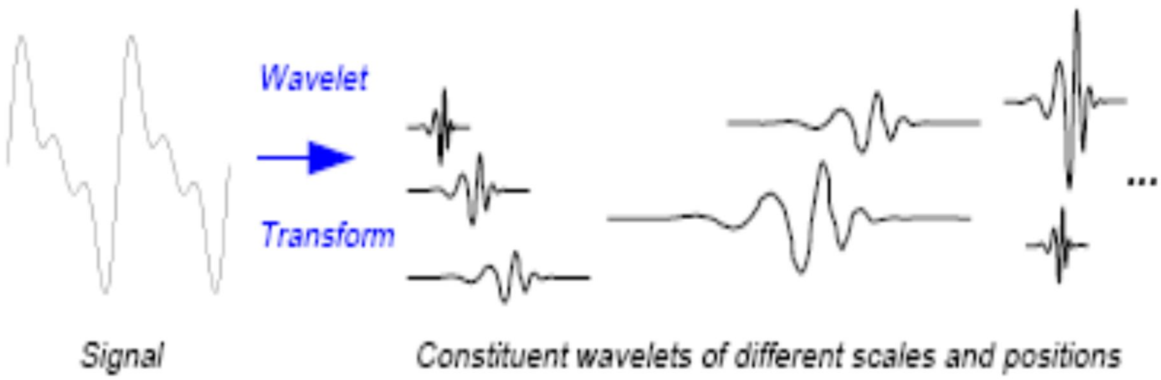


Fig 2.6 Continuous Wavelets of different scales and positions

The wavelet transform involves projecting a signal onto a complete set of translated and dilated versions of a mother wavelet $\psi(t)$. The strict definition of a mother wavelet will be dealt with later so that the form of the wavelet transform can be examined first. For now, assume the loose requirement that $\psi(t)$ has compact temporal and spectral support (limited by the uncertainty principle of course), upon which set of basis functions can be defined. The basis set of wavelets is generated from the mother or basic wavelet is defined as

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right); \text{ a,b} \in \mathrm{R}^1 \; and \;\; \text{a>0}$$

The variable 'a' (inverse of frequency) reflects the scale (width) of a particular basis function such that its large value gives low frequencies and small value gives high frequencies. The variable 'b' specifies its translation along x-axis in time. The term 1/ a is used for normalization. The 1-D wavelet transform is given by:

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

The inverse 1-D wavelet transform is given by

$$\text{x(t)} = \frac{1}{C}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w_f(a,b)\psi_{a,b}(t)db\frac{da}{a^2}$$

$$Where \;\; \text{C} = \int_{-\infty}^{\infty}\frac{|\psi(\omega)|^2}{\omega}d\omega < \infty$$

X (t) is the Fourier transform of the mother wavelet $\psi(t)$. *C* is required to be finite, which leads to one of the required properties of a mother wavelet. Since *C* must be finite, then x (t) =0 to avoid a singularity in the integral, and thus the x(t) must have zero mean. This condition can be stated as

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

and known as the admissibility condition. The other main requirement is that the mother wavelet must have finite energy:

$$\int_{-\infty}^{\infty} |\psi(t)|^2\,dt < \infty$$

A mother wavelet and its scaled versions are depicted in figure 2.10 indicating the effect of scaling.

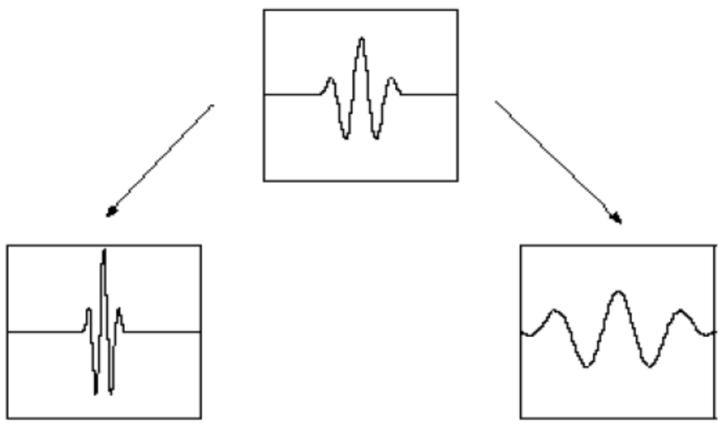


Fig 2.7 Mother wavelet and its scaled versions

Unlike the STFT which has a constant resolution at all times and frequencies, the WT has a good time and poor frequency resolution at high frequencies, and good frequency and poor time resolution at low frequencies.

2.5. Reasons for preferring Wavelet

The compression in wavelet domain is preferred, because it has many advantages:

a) Wavelet-based compression provides multi-resolution hierarchical characteristics. Hence an image an be compressed at different levels of resolution and can be sequentially processed from low resolution to high resolution [7][20][22].

b) High robustness to common signal processing.

c) Real time signals are both time-limited (or space limited in the case of images) and band-limited. Time-limited signals can be efficiently represented by a basis of block functions (Dirac delta functions for infinitesimal small blocks). But block functions are not band-limited. Band limited signals on the other hand can be efficiently represented by a Fourier basis. But sines and cosines are not time-limited. Wavelets are localized in both time (space) and frequency (scale) domains. Hence it is easy to capture local features in a signal.

d) Another advantage of a wavelet basis is that it supports multi resolution. Consider the windowed Fourier transform. The effect of the window is to localize the signal being analyzed.

Because a single window is used for all frequencies, the resolution of the analysis is same at all frequencies. To capture signal discontinuities (and spikes), one needs shorter windows, or shorter basis functions. At the same time, to analyze low frequency signal components, one needs longer basis functions. With wavelet based decomposition, the window sizes vary. Thus it allows analyzing the signal at different resolution levels.

2.6. Classification of Wavelets:

The wavelets can be classify into two classes: (a) orthogonal and (b) biorthogonal. Based on the application, either of them can be used.

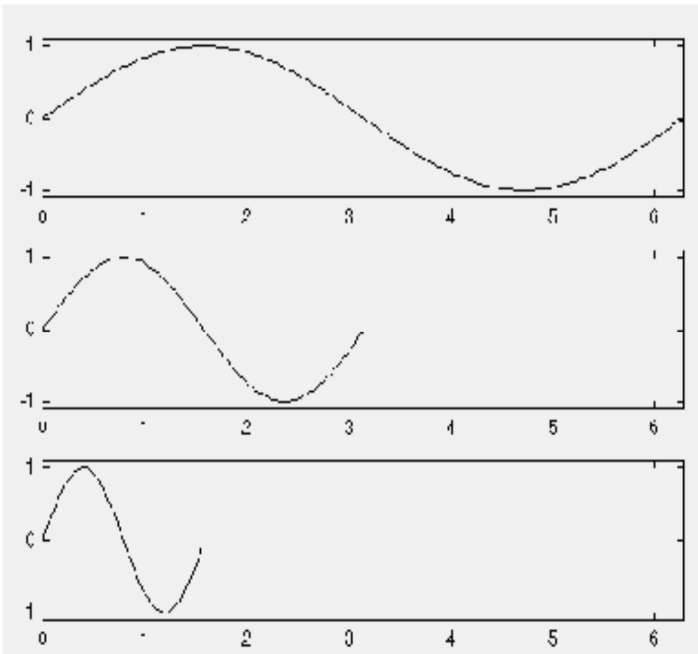
- **Features of orthogonal wavelet filter banks :** The coefficients of orthogonal filters are real numbers. The filters are of the same length and are not symmetric.

  The low pass filter, G0 and the high pass filter, H0 are related to each other by $H0(z) = z^{-N} G0(-z^{-1})$. The two filters are alternated flip of each other. The alternating flip automatically gives double-shift orthogonality between the low pass and high pass filters, i.e., the scalar product of the filters, for a shift by two is zero. i.e., $\sum G[k]H[k-2l] = 0, \text{ where k,l} \in Z$ Filters that satisfy equation are known as Conjugate Mirror Filters (CMF) [27]. Perfect reconstruction is possible with alternating flip. Also, for perfect reconstruction, the synthesis filters are identical to the analysis filters except for a time reversal. Orthogonal filters offer a high number of vanishing moments. This property is useful in many signal and image processing applications. They have regular structure which leads to easy implementation and scalable architecture.
- **Features of biorthogonal wavelet filter banks:** In the case of the biorthogonal wavelet filters, the low pass and the high pass filters do not have the same length. The low pass filter is always symmetric, while the high pass filter could be either symmetric or anti-symmetric. The coefficients of the filters are either real numbers or integers. For perfect reconstruction, biorthogonal filter bank has all odd length or all even length filters. The two analysis filters can be symmetric with odd length or one symmetric and the other anti symmetric with even length. Also, the two sets of analysis and synthesis filters must be dual. The linear phase biorthogonal filters are the most popular filters for data compression applications.

2.7. Property of Wavelet:

Scaling and Shifting:

Scaling a wavelet simply means stretching (or compressing) it. The effect of the scaling factor is very easy to see:



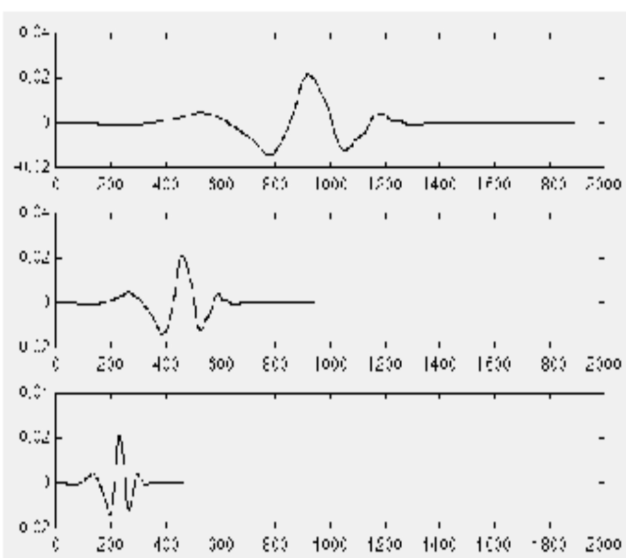
$$f(t) = \sin(t)\,; \quad a = 1$$

$$f(t) = \sin(2t)\,; \quad a = \frac{1}{2}$$

$$f(t) = \sin(4t)\,; \quad a = \frac{1}{4}$$

Fig 2.8: General scaling

The scale factor works exactly the same with wavelets. The smaller the scale factor, the more "compressed" the wavelet.



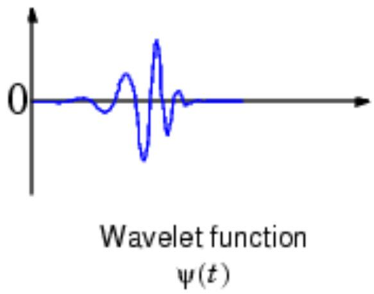
$$f(t) = \psi(t) \quad ; \quad a = 1$$

$$f(t) = \psi(2t)\,; \quad a = \frac{1}{2}$$

$$f(t) = \psi(4t)\,; \quad a = \frac{1}{4}$$

Fig 2.9: Scaling in wavelets

It is clear from the diagram that, for a sinusoid sin(*wt*) , the scale factor *a* is related (inversely) to the radian frequency *w*. Similarly, with wavelet analysis, the scale is related to the frequency of the signal. Shifting a wavelet simply means delaying (or hastening) its onset.
Mathematically, delaying a function f(t) by *k* is represented by f(t-k).



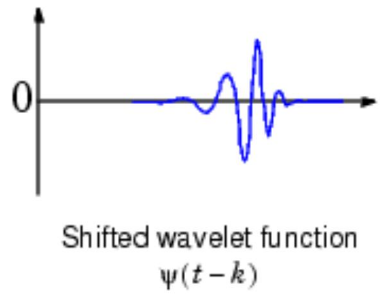


Fig 2.10: Shifting in wavelets

Various other properties of wavelet transforms is described below:

1) Regularity

2) The window for a function is the smallest space-set (or time-set) outside which function is identically zero.

3) The order of the polynomial that can be approximated is determined by number of vanishing moments of wavelets and is useful for compression purposes.

4) The symmetry of the filters is given by wavelet symmetry. It helps to avoid de phasing in image processing. The Haar wavelet is the only symmetric wavelet among orthogonal. For biorthogonal wavelets both wavelet functions and scaling functionsthat are either symmetric or antisymmetric can be synthesized.

5) Filter length: Shorter synthesis basis functions are desired for minimizing distortion that affects the subjective quality of the image [22]. Longer filters (that correspond to longer basis functions) are responsible for ringing noise in the reconstructed image at low bit rates.

2.8. Architecture of Wavelet:

Wavelet compression involves a way analyzing an uncompressed image in a recursive fashion, resulting in a series of higher resolution images, each “adding to” the information content in lower resolution images. The primary steps in wavelet compression are performing a discrete wavelet Transformation (DWT), quantization of the wavelet-space image sub bands, and then encoding these sub bands.

Wavelet images by and of themselves are not compressed images; rather it is quantization and encoding stages that do the image compression and to store the compressed image. Wavelet compression inherently results in a set of multi-resolution images; it is well suited to working with large imagery which needs to be selectively viewed at different resolution, as only the levels containing the required level of detail need to be decompressed. The following diagram shows wavelet based compression.

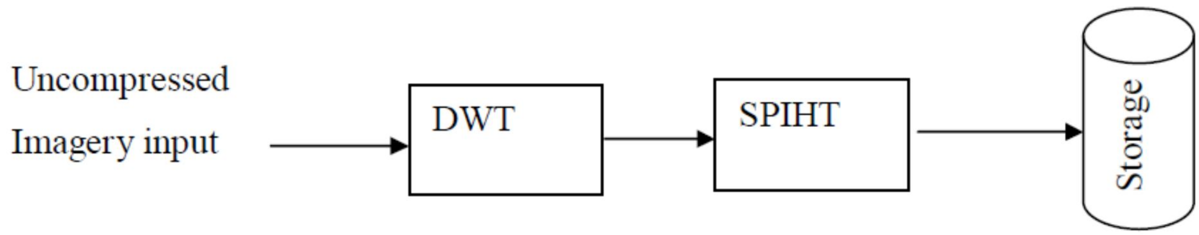


Fig 2.11:Wavelet Based Image compression

2.8.1. Decomposition Process

The image is high and low-pass filtered along the rows. Results of each filter are downsampled by two. The two sub-signals correspond to the high and low frequency components along the rows, each having a size N by N/2. Each of the sub-signals is then again high and low-pass filtered, but now along the column data and the results are again down-sampled by two.

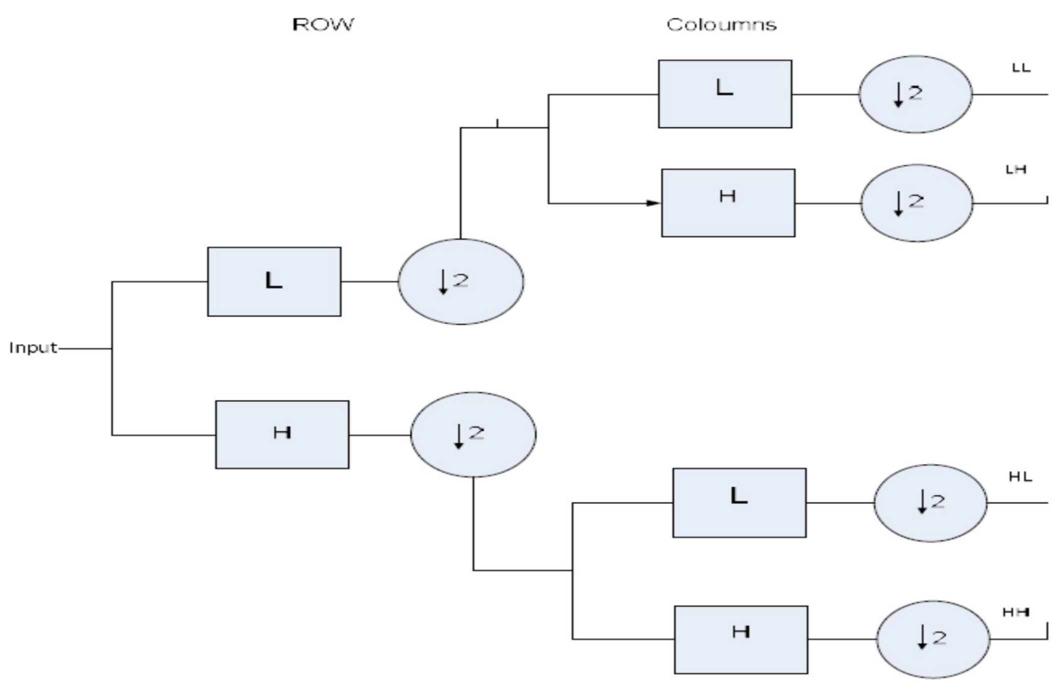


Fig 2.12: One decomposition step of the two dimensional image

Hence, the original data is split into four sub-images each of size N/2 by N/2 and contains information from different frequency components [27]. Figure 3.15 shows the block wise representation of decomposition step.

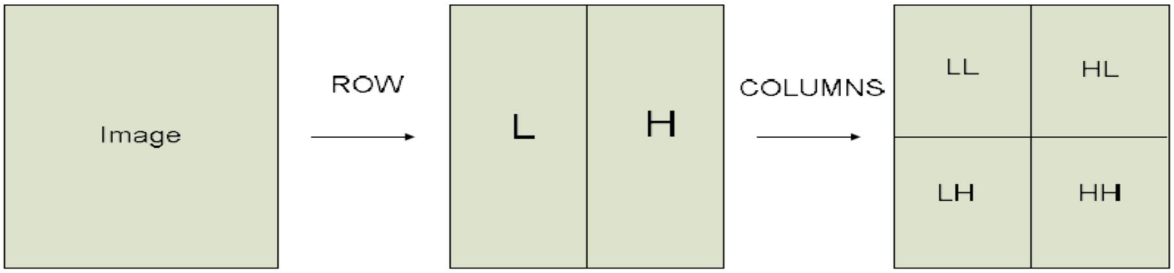


Fig 2.13: One DWT decomposition step

The LL subband obtained by low-pass filtering both the rows and columns, contains a rough description of the image and hence called the approximation subband. The HH Subband, high-pass filtered in both directions, contains the high-frequency components along the diagonals. The HL and LH images result from low-pass filtering in one direction and highpass filtering in the other direction. LH contains mostly the vertical detail information, which corresponds to horizontal edges. HL represents the horizontal detail information from the vertical edges. The subbands HL, LH and HH are called the detail subbands since they add the high-frequency detail to the approximation image.

2.8.2. Composition Process:

Figure 2.14 corresponds to the composition process. The four sub-images are up-sampled and then filtered with the corresponding inverse filters along the columns [27]. The result of the last step is added together and the original image is retrieved, with no information loss.

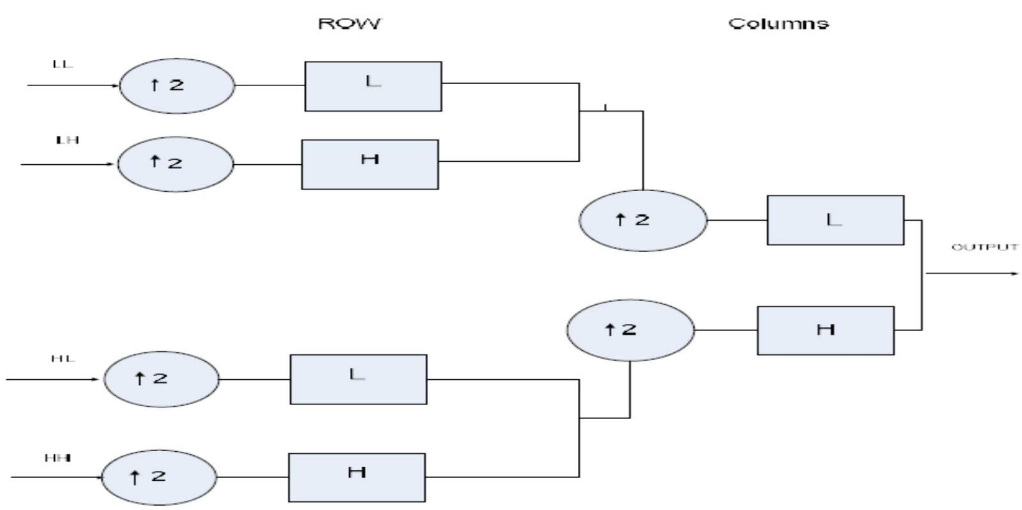


Fig 2.14: One composition step of the four sub images

Chapter- 3

Image Compression

Introduction

Image compression techniques

Types of image compression algorithms

Terms used in image compression

3.IMAGE COMPRESSION

3.1. Introduction:

Image compression is the process of encoding information using fewer bits (or other information-bearing units) than an encoded representation would use through use of specific encoding schemes. Compression is useful because it helps reduce the consumption of expensive resources, such as hard disk space or transmission bandwidth (computing). On the downside, compressed data must be decompressed, and this extra processing may be detrimental to some applications. For instance, a compression scheme for image may require expensive hardware for the image to be decompressed fast enough to be viewed as it's being decompressed (the option of decompressing the image in full before watching it may be inconvenient, and requires storage space for the decompressed image). The design of data compression schemes therefore involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (if using a lossy compression scheme), and the computational resources required to compress and uncompress the data.

Image compression is an application of data compression on digital images. Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages. There are several different ways in which image files can be compressed. For Internet use, the two most common compressed graphic image formats are the JPEG format and SPIHT format.

3.2. Image Compression Techniques:

Image compression can be lossy or lossless. Lossless compression is sometimes preferred for artificial images such as technical drawings, icons or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossless compression methods may also be preferred for high value content, such as medical imagery or image scans made for archival purposes. Lossy methods are especially suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate.

3.2.1. Lossy image compression:

A lossy compression method is one where compressing data and then decompressing it retrieves data that may well be different from the original, but is close enough to be useful in some way [23][28]. Lossy compression is most commonly used to compress multimedia data (audio, video, still images), especially in applications such as streaming media and internet telephony. On the other hand lossless compression is required for text and data files, such as bank records, text articles, etc. Lossy compression formats suffer from generation loss: repeatedly compressing and decompressing the file will cause it to progressively lose quality. This is in contrast with lossless data compression. Information-theoretical foundations for lossy data compression are provided by rate-distortion theory. Much like the use of probability in optimal coding theory, rate-distortion theory heavily draws on Bayesian estimation and decision theory in order to model perceptual distortion and even aesthetic judgment.

3.2.2. Lossless Image Compression:

Lossless or reversible compression refers to compression techniques in which the reconstructed data exactly matches the original [23]. Lossless compression denotes compression methods, which give quantitative bounds on the nature of the loss that is introduced. Such compression techniques provide the guarantee that no pixel difference between the original and the compressed image is above a given value. It finds potential applications in remote sensing, medical and space imaging, and multispectral image archiving. In these applications the volume of the data would call for lossy compression for practical storage or transmission. However, the necessity to preserve the validity and precision of data for subsequent reconnaissance, diagnosis operations, forensic analysis, as well as scientific or clinical measurements, often imposes strict constraints on the reconstruction error. In such situations lossless compression becomes a viable solution, as, on the one hand, it provides significantly higher compression gains vis-à-vis lossless algorithms, and on the other hand it provides guaranteed bounds on the nature of loss introduced by compression.

Another way to deal with the lossy-lossless dilemma faced in applications such as medical imaging and remote sensing is to use a successively refinable compression technique that provides a bit stream that leads to a progressive reconstruction of the image. Using wavelets, for example, one can obtain an embedded bit stream from which various levels of rate and distortion can be obtained. In fact with reversible integer wavelets, one gets a progressive reconstruction capability all the way to lossless recovery of the original. Such techniques have been explored for potential use in tele-radiology where a physician typically requests portions of an image at increased quality (including lossless reconstruction) while accepting initial renderings and unimportant portions at lower quality, and thus reducing the overall bandwidth requirements. In fact, the new still image compression standard, JPEG 2000, provides such features in its extended form.

Various Loss-Less Compression Method are:

- Run-length encoding – used as default method in PCX and as one of possible in BMP, TGA, TIFF
- Entropy coding
- Adaptive dictionary algorithms such as LZW – used in GIF and TIFF
- Deflation – used in PNG, MNG and TIFF.

The usual steps involved in compressing and decompressing of image are:

Step 1: Specifying the Rate (bits available) and Distortion (tolerable error) parameters for the target image.

Step 2: Dividing the image data into various classes, based on their importance.

Step 3: Dividing the available bit budget among these classes, such that the distortion is a minimum.

Step 4: Quantize each class separately using the bit allocation information derived in step 2.

Step 5: Encode each class separately using an entropy coder and write to the file.

Step 6: Reconstructing the image from the compressed data is usually a faster process than compression. The steps involved are

Step 7: Read in the quantized data from the file, using an entropy decoder (reverse of step 5).

Step 8: Dequantize the data. (Reverse of step 4).

Step 9: Rebuild the image. (Reverse of step 2).

3.3. Different Types of Image Compression algorithms:

The various types of image compression methods are described below:

3.3.1. Embedded zero wavelet coding {EZW}:

The EZW algorithm was one of the first algorithms to show the full power of wavelet-based image compression. It was introduced in the groundbreak ing paper of Shapiro [4]. We shall describe EZW in some detail because a solid understanding of it will make it much easier to comprehend the other algorithms we shall be discussing. These other algorithms build upon the fundamental concepts that were first introduced with EZW.

Our discussion of EZW will be focused on the fundamental ideas underly ing it; we shall not use it to compress any images. That is because it has been superseded by a far superior algorithm, the SPIHT algorithm. Since SPIHT is just a highly refined version of EZW, it makes sense to first describe EZW.

EZW stands for Embedded Zerotree Wavelet. We shall explain the terms Embedded, and Zerotree, and how they relate to Wavelet-based compression. An embedded coding is a process of encoding the transform magnitudes that allows for progressive transmission of the compressed image. Zerotrees are a concept that allows for a concise encoding of the positions of significant values that result during the embedded coding process. We shall first discuss embedded coding, and then examine the notion of zerotrees.

The embedding process used by EZW is called bit-plane encoding. It consists of the following five-step process:

Bit-plane encoding:

*Step*1(*Initialize*): Choose initial threshold, T=T_0, such that all transform values satisfy $|w(m)| < T_0$ and atleast one transform value satisfies $|w(m)| \geq T_0/2$

Step 2(Update threshold): Let $T_k = T_{k-1}/2$

Step 3(Significant pass): Scan through insignificant values using baseline algorithm scan order. Test each value $w(m)$ *as follows*

If $|w(m)| \geq T_{k,}$ *then*

Output sign of $w(m)$

Set $w_Q(m) = T_k$

Else if $|w(m)| < T_k$ then

Let $w_Q(m)$ retain its initial value 0.

Step 4(Refinement pass): Scan through significant values found with higher threshold values T_j, for j<k(if k=1 skip this step). For each significant value $w(m)$, do the following:

If $|w(m)| \in [w_Q(m), w_Q(m) + T_k)$, then

Output bit 0

Else if $|w(m)| \in [w_Q(m) + T_k, w_Q(m) + 2T_k)$, then

Output bit 1

Repeat value of $w_Q(m)$ by $w_Q(m) + T_k$.

Step 5(Loop): Repeat steps 2 through 4.

This bit-plane encoding procedure can be continued for as long as necessary to obtain quantized transform magnitudes $w_Q(m)$ which are as close as desired to the transform magnitudes $|w(m)|$. During decoding, the signs and the bits output by this method can be used to construct an approximate wavelet transform to any desired degree of accuracy. If instead, a given compression ratio is desired, then it can be achieved by stopping the bit-plane encoding as soon as a given number of bits (a bit budget) is exhausted. In either case, the execution of the bit-plane encoding procedure can terminate at any point (not just at the end of one of the loops).

3.3.2. Set Partitioning in Hierarchical Trees (SPIHT):

SPIHT is the wavelet based image compression method. It provides the Highest Image Quality, Progressive image transmission, fully embedded coded file, Simple quantization algorithm, fast coding/decoding, completely adaptive, Lossless compression, Exact bit rate coding and Error protection[6][11]. SPIHT makes use of three lists – the List of Significant Pixels (LSP), List of Insignificant Pixels (LIP) and List of Insignificant Sets (LIS). These are coefficient location lists that contain their coordinates. After the initialization, the algorithm takes two stages for each level of threshold – the sorting pass (in which lists are organized) and the refinement pass (which does the actual progressive coding transmission). The result is in the form of a bit stream. It is capable of recovering the image perfectly (every single bit of it) by coding all bits of the transform. However, the wavelet transform yields perfect reconstruction only if its numbers are stored as infinite imprecision numbers. Peak signal-to noise ratio (PSNR) is one of the quantitative measures for image quality evaluation which is based on the mean square error (MSE) of the reconstructed image. The MSE for K x M size image is given by:

$$MSE = \frac{1}{MK} \sum_{i=0}^{K-1} \sum_{j=0}^{M-1} \left| f(i, j) - f_1(i, j) \right|^2$$

Where *f(i,j)* is the original image data and *f1(i,j)* is the compressed image value. The formula for PSNR is given by:

$$PSNR = 10 \log((255)^2 / MSE)$$

SPIHT CODING ALGORITHM:

Since the order in which the subsets are tested for significance is important in a practical implementation the significance information is stored in three ordered lists called list of insignificant sets (LIS) list of insignificant pixels (LIP) and list of significant pixels (LSP). In all lists each entry is identified by a coordinate (i, j) which in the LIP and LSP represents individual pixels and in the LIS represents either the set D (i, j) or L (i, j) [13][17]. To differen tiate between them it can be concluded that a LIS entry is of type A if it represents D (i,j) and of type B if it represents L(i, j). During the sorting pass the pixels in the LIP-which were insignificant in the previous pass-are tested and those that become significant are moved to the LSP. Similarly, sets are sequentially evaluated following the LIS order, and when a set is found to be significant it is removed from the list and partitioned. The new subsets with more than one element are added back to the LIS, while the single coordinate sets are added to the end of the LIP or the LSP depending whether they are insignificant or significant respectively. The LSP contains the coordinates of the pixels that are visited in the refinement pass. Below the new encoding algorithm in presented.

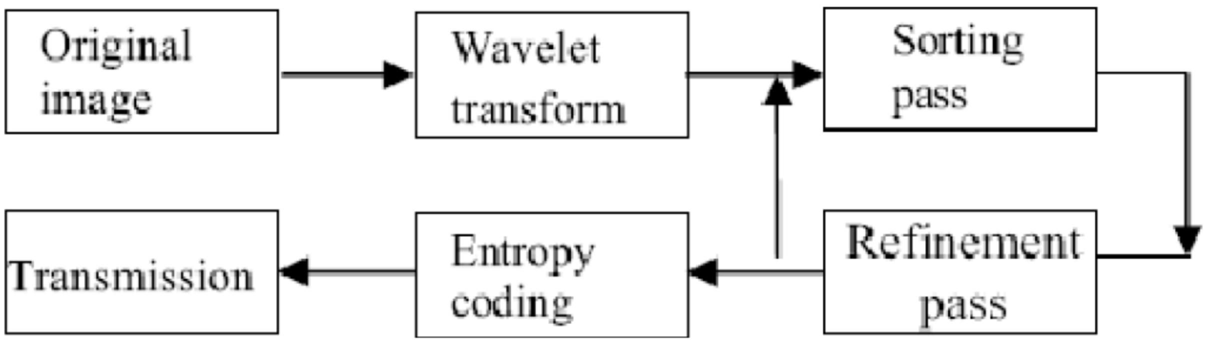


Fig 3.2: Flow Chart of SPIHT

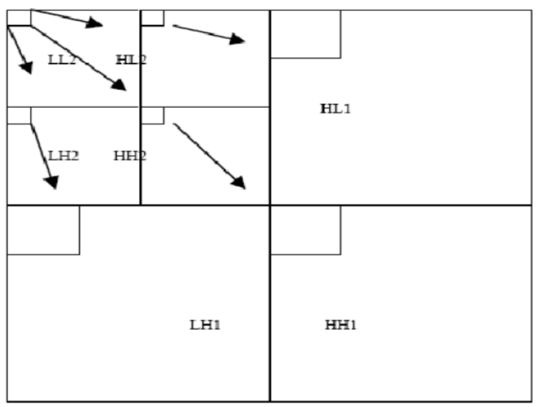


Fig 3.3: Tree Structure of SPIHT

ALGORITHM:

1. Initialization: output n= |log2 (max (i, j) {c (i, j)}|; set the LSP as an empty list, and add the coordinates (i,j) € H to the LIP, and only those with descendants also to the LIS, as type A entries [7].

2. Sorting pass:

    2.1 for each entry (i,j) in the LIP do:

        2.1.1 output Sn (i,j)

        2.1.2 if Sn (i, j) then move (i, j) to the LSP and output the sign of c(i, j)

    2.2 for each entry(i, j) in the LIS do:

        2.2.1 if the entry is of type A then

- output Sn(D(i, j));
- if Sn(D(i, j)) then

  for each (k, l) ∈ O(i, j)do:

  output Sn(k, l);

  if Sn(k, l) then add (k,l) to the LSP and output the sign of ck,l;

  if Sn (k, l) = 0 then add (k, l) to the end of the LIP;

  if L(i, j)≠0then move (i, j) to the end of the LIS, as an entry of type B and go to Step2.2.2 else, remove entry (i, j) from the LIS;

2.2.2 if the entry is of type B then

- Output Sn(L(i, j));
- if Sn (L(i, j)) = 1 then

add each (k, l) € O(i, j) to the end of the LIS as an entry of type A;

remove (i, j) from the LIS.

3. Refinement pass for each entry (i, j) in the LSP, except those included in the last sorting pass (with same n), output the n-th most significant bit of |ci, j|;

4. Quantization-step update: decrement n by 1 and go to Step 2

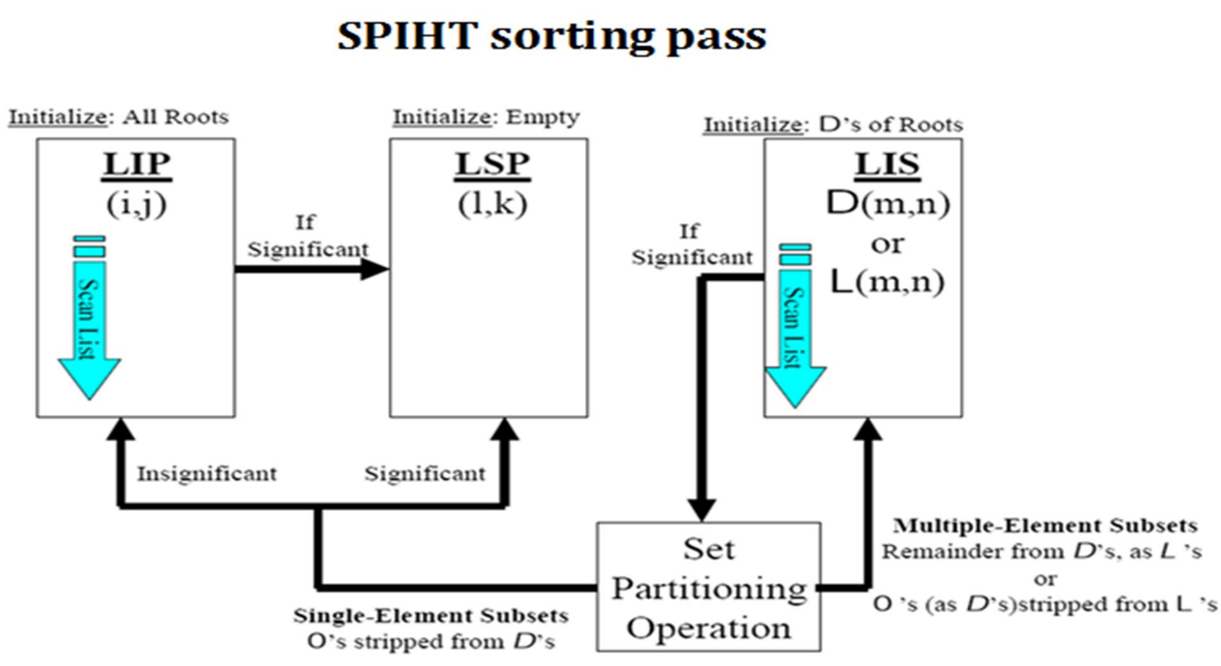


Fig:3.3 SPIHT sorting pass

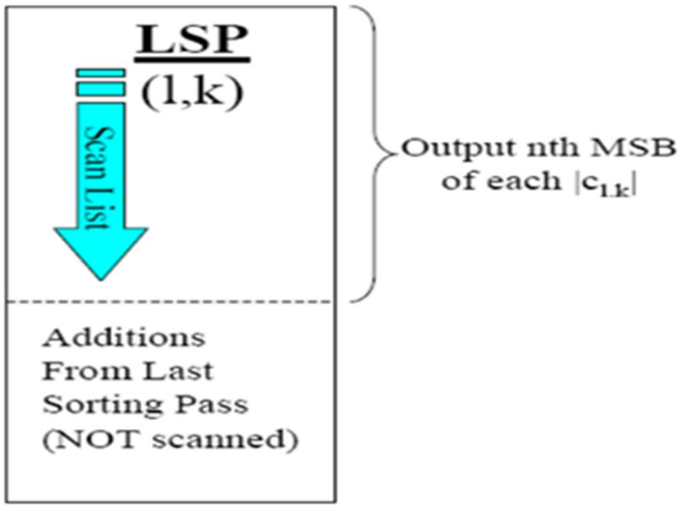
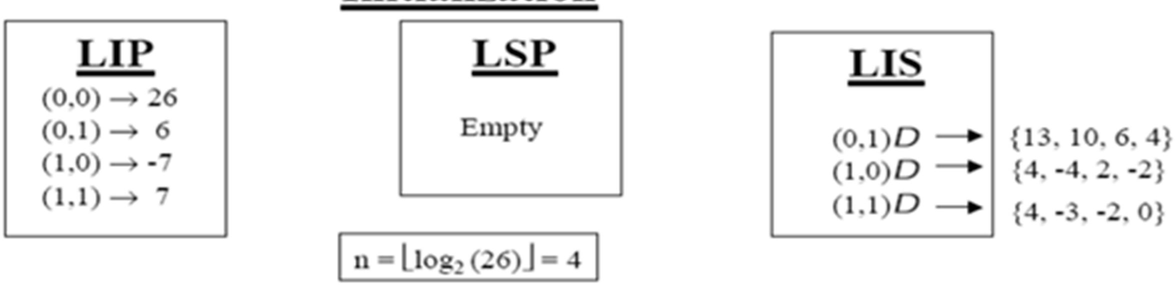


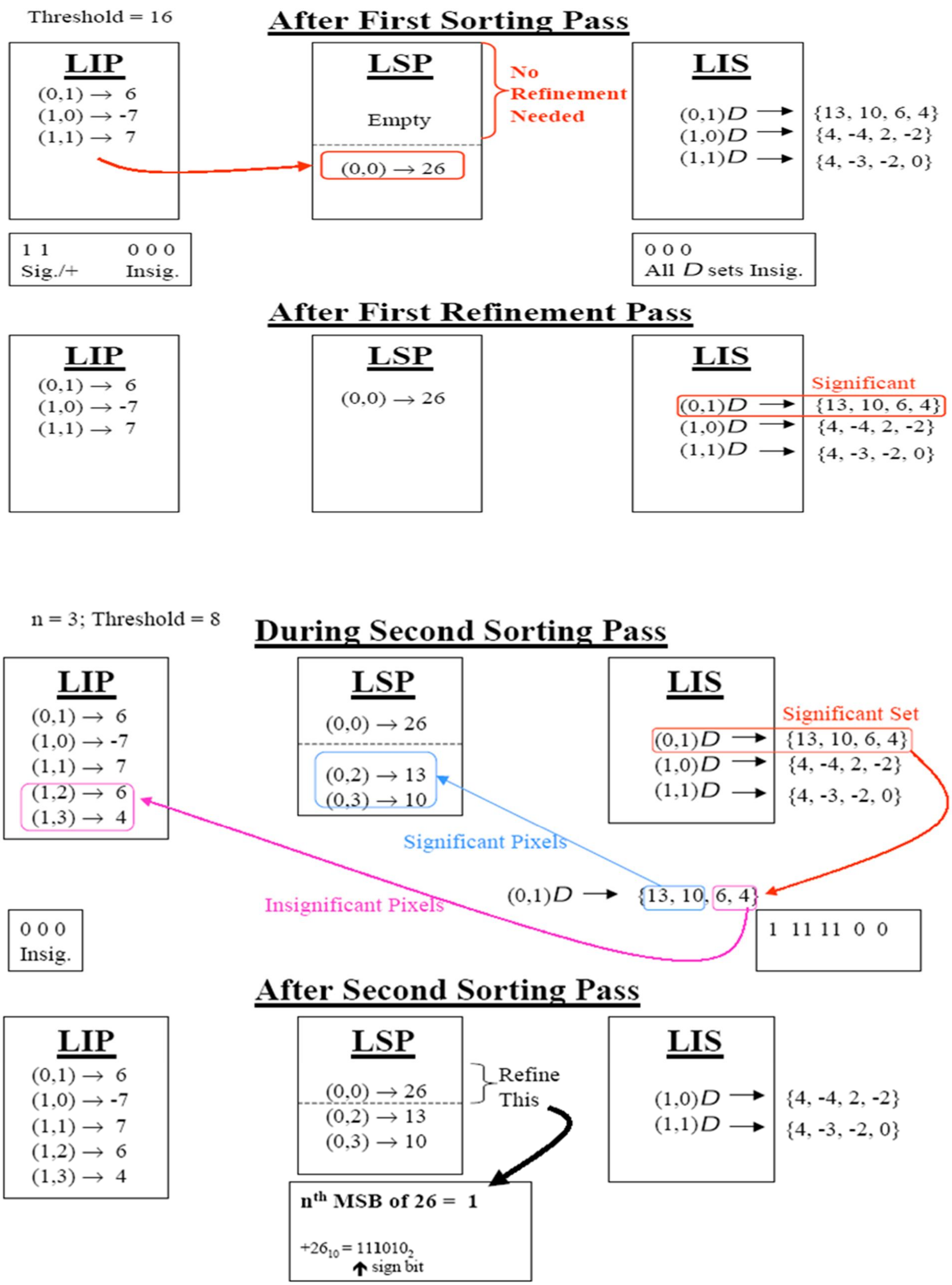
Fig:3.4 SPIHT refinement pass

Example of SPIHT

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

Initialization

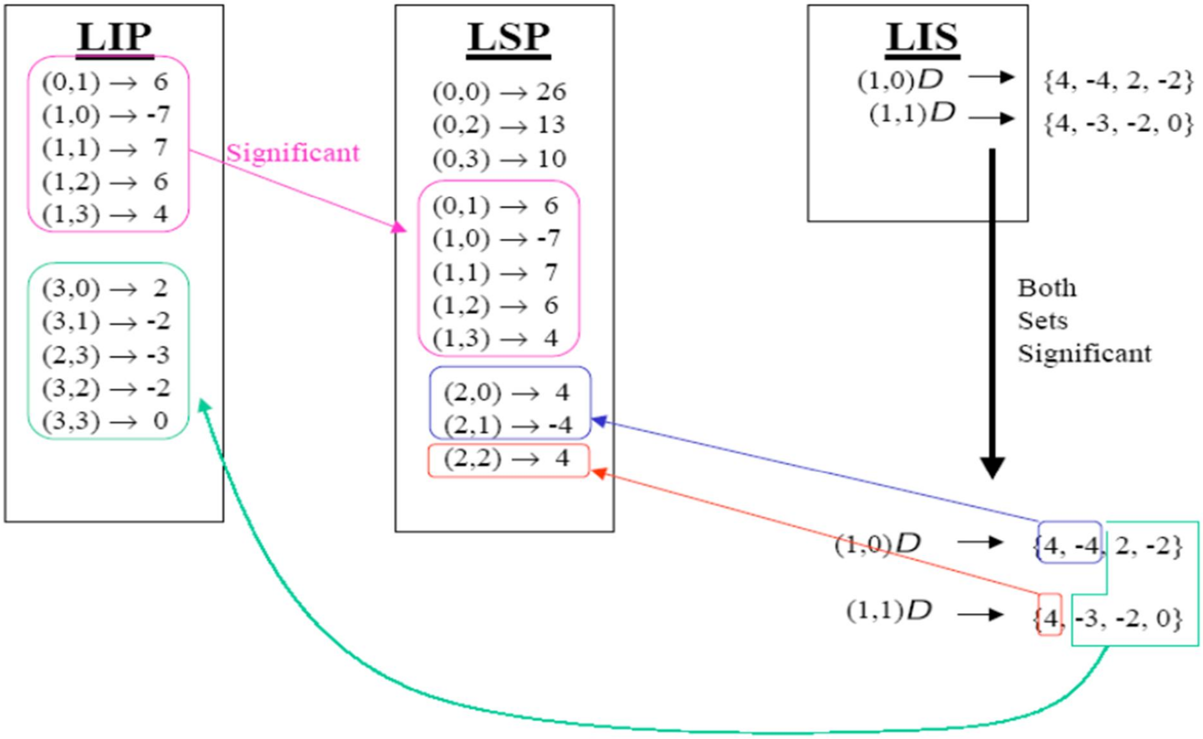




Threshold = 16
After First Sorting Pass
LIP
(0,1) → 6
(1,0) → -7
(1,1) → 7
LSP
Empty
(0,0) → 26
No Refinement Needed
LIS
(0,1)D → {13, 10, 6, 4}
(1,0)D → {4, -4, 2, -2}
(1,1)D → {4, -3, -2, 0}
1 1 Sig./+ 0 0 0 Insig.
0 0 0 All D sets Insig.
After First Refinement Pass
LIP
(0,1) → 6
(1,0) → -7
(1,1) → 7
LSP
(0,0) → 26
LIS
Significant
(0,1)D → {13, 10, 6, 4}
(1,0)D → {4, -4, 2, -2}
(1,1)D → {4, -3, -2, 0}
n = 3; Threshold = 8
During Second Sorting Pass
LIP
(0,1) → 6
(1,0) → -7
(1,1) → 7
(1,2) → 6
(1,3) → 4
LSP
(0,0) → 26
(0,2) → 13
(0,3) → 10
LIS
Significant Set
(0,1)D → {13, 10, 6, 4}
(1,0)D → {4, -4, 2, -2}
(1,1)D → {4, -3, -2, 0}
Significant Pixels
Insignificant Pixels
(0,1)D → {13, 10, 6, 4}
0 0 0 Insig.
1 11 11 0 0
After Second Sorting Pass
LIP
(0,1) → 6
(1,0) → -7
(1,1) → 7
(1,2) → 6
(1,3) → 4
LSP
(0,0) → 26
(0,2) → 13
(0,3) → 10
Refine This
LIS
(1,0)D → {4, -4, 2, -2}
(1,1)D → {4, -3, -2, 0}
n^th MSB of 26 = 1
+26_10 = 111010_2
sign bit

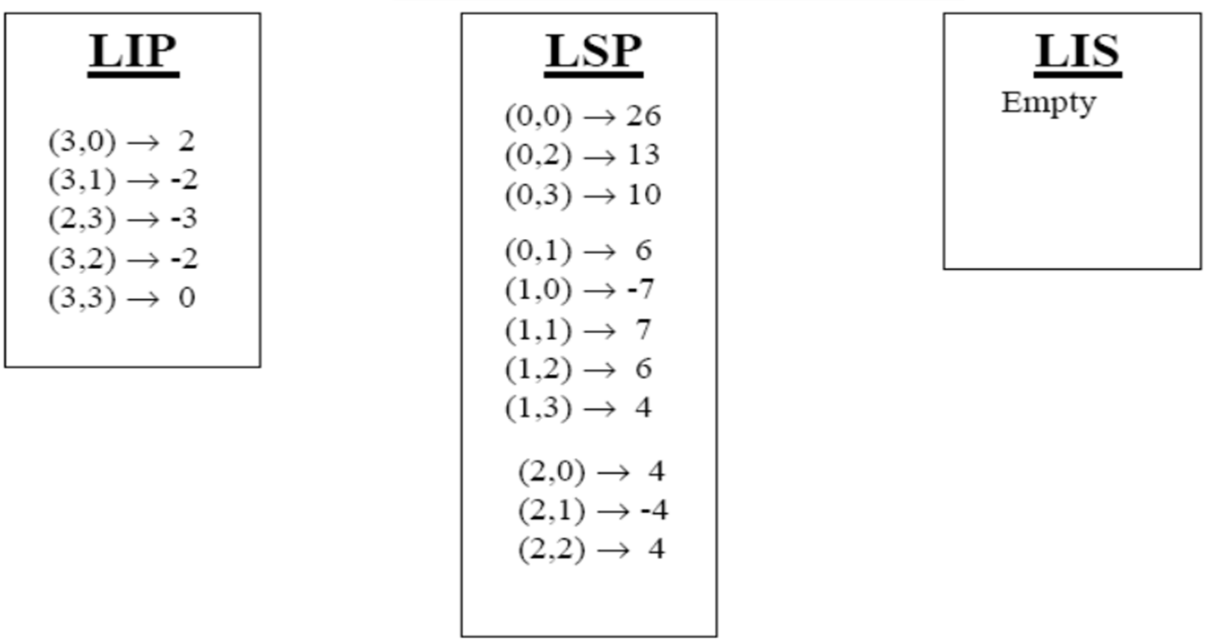
Threshold = 4

During Third Sorting Pass



Threshold = 4

After Third Sorting Pass



3.3.2.1. Advantages of SPIHT:

The powerful wavelet-based image compression method called Set Partitioning in Hierarchical Trees (SPIHT). The SPIHT method is not a simple extension of traditional methods for image compression, and represents an important advance in the field. The method deserves special attention because it provides the following:

1) Highest Image Quality
2) Progressive image transmission
3) Fully embedded coded file
4) Simple quantization algorithm
5) Fast coding/decoding
6) Completely adaptive
7) Lossless compression
8) Exact bit rate coding
9) Error protection

Each of these properties is discussed below. Note that different compression methods were developed specifically to achieve at least one of those objectives [11]. What makes SPIHT really outstanding is that it yields all those qualities simultaneously**.** So, if in the future you find one method that claims to be superior to SPIHT in one evaluation parameter (like PSNR), remember to see who wins in the remaining criteria.

- **Image Quality:** Extensive research has shown that the images obtained with wavelet-based methods yield very good visual quality [5][18]. At first it was shown that even simple coding methods produced good results when combined with wavelets and is the basis for the most recently JPEG2000 standard. However, SPIHT belongs to the next generation of wavelet encoders, employing more sophisticated coding. In fact, SPIHT exploits the properties of the wavelet-transformed images to increase its efficiency. Many researchers now believe that encoders that use wavelets are superior to those that use DCT or fractals. The SPIHT advantage is even more pronounced in encoding colour images, because the bits are allocated automatically for local optimality among the colour components, unlike other algorithms that encode the colour components separately based on global statistics of the individual components. You will be amazed to see that visually lossless color compression is obtained with some images at compression ratios from 100-200:1.

- **Progressive Image Transmission:** In some systems with progressive image transmission the quality of the displayed images follows the sequence:

(a) weird abstract art;

(b) you begin to believe that it is an image of something;

(c) CGA-like quality;

(d) Lossless recovery.

With very fast links the transition from (a) to (d) can be so fast that you will never notice. With slow links (how "slow" depends on the image size, colors, etc.) the time from one stage to the next grows exponentially, and it may take hours to download a large image. Considering that it may be possible to recover an excellent-quality image using 10-20 times less bits, it is easy to see the inefficiency. Furthermore, the mentioned systems are not efficient even for lossless transmission. The problem is that such widely used schemes employ a very primitive progressive image transmission method. On the other extreme, SPIHT is a state-of-the-art method that *was* designed *for* optimal progressive transmission (and still beats most non progressive methods!). It does so by producing a fully embedded coded file (see below), in a manner that at any moment the quality of the displayed image is the best available for the number of bits received up to that moment. So, SPIHT can be very useful for applications where the user can quickly inspect the image and decide if it should be really downloaded, or is good enough to be saved, or need refinement.

- **Optimized Embedded Coding:** A strict definition of the embedded coding scheme is: if two files produced by the encoder have size M and N bits, with $M > N$, then the file with size N is *identical* to the first N bits of the file with size M. Let's see how this abstract definition is used in practice. Suppose you need to compress an image for three remote users. Each one have different needs of image reproduction quality, and you find that those qualities can be obtained with the image compressed to at least 8 Kb, 30 Kb, and 80 Kb, respectively. If you use a non-embedded encoder (like JPEG), to save in transmission costs (or time) you must prepare one file for each user. On the other hand, if you use an embedded encoder (like SPIHT) then you can compress the image to a single 80 Kb file, and then send the first 8 Kb of the file to the first user, the first 30 Kb to the second user, and the whole file to the third user. SPIHT *all* three users would get an image quality comparable or superior to the most sophisticated non-embedded encoders available today. SPIHT achieves this feat by optimizing the embedded coding process and always coding the most important information first. An even more important application is for progressive image transmission, where the user can decide at which point the image quality satisfies his needs, or abort the transmission after a quick inspection, etc.

- **Compression Algorithm:** The following is a comparison of image quality and artifacts at high compression ratios versus JPEG. SPIHT represents a small "revolution" in image compression because it broke the trend to more complex (in both the theoretical and the computational senses) compression schemes. While researchers had been trying to improve previous schemes for image coding using very sophisticated vector quantization, SPIHT achieved superior results using the *simplest* method: uniform scalar quantization. Thus, it is much easier to design fast SPIHT code.
- **Decoding Speed:** The SPIHT process represents a very effective form of entropy coding. This is shown by the demo programs using two forms of coding: binary uncoded (extremely simple) and context-based adaptive arithmetic coded (sophisticated). Surprisingly, the difference in compression is small, showing that it is not necessary to use slow methods (and also pay royalties for them!). A fast version using Huffman codes was also successfully tested, but it is not publicly available. A straightforward consequence of the compression simplicity is the greater coding/decoding speed [5]. The SPIHT algorithm is nearly symmetric, i.e., the time to encode is nearly equal to the time to decode. (Complex compression algorithms tend to have encoding times much larger than the decoding times.) Some of our demo programs use floating-point operations extensively, and can be slower in some CPUs (floating points are better when people want to test you programs with strange 16 bpp images). However, this problem can be easily solved: try the lossless version to see an example. Similarly, the use for progressive transmission requires a somewhat more complex and slower algorithm. Some shortcuts can be used if progressive transmission is not necessary. When measuring speed please remember that these demo programs were written for academic studies only, and were not fully optimized as are the commercial versions.

3.3.2.2.Applications:

SPIHT exploits properties that are present in a wide variety of images. It had been successfully tested in natural (portraits, landscape, weddings, etc.) and medical (X-ray, CT, etc) images. Furthermore, its embedded coding process proved to be effective in a broad range of reconstruction qualities. For instance, it can code fair quality portraits and high-quality medical images equally well (as compared with other methods in the same conditions). SPIHT has also been tested for some less usual purposes, like the compression of elevation maps, scientific data, and others.

Lossless Compression: SPIHT codes the individual bits of the image wavelet transform coefficients following a bit-plane sequence [7][17]. Thus, it is capable of recovering the image perfectly (every single bit of it) by coding all bits of the transform. However, the wavelet transform yields perfect reconstruction only if its numbers are stored as infinite-precision numbers. In practice it is frequently possible to recover the image perfectly using rounding after recovery, but this is not the most efficient approach. For lossless compression an integer multi resolution transformation is proposed, similar to the wavelet transform, which is called S+P transform. It solves the finite-precision problem by carefully truncating the transform coefficients *during* the transformation (instead of after). A codec that uses this transformation to yield efficient progressive transmission up to lossless recovery is among the SPIHT demo programs. A surprising result obtained with this codec is that for lossless compression it is as efficient as the most effective lossless encoders (lossless JPEG is definitely not among them). In other words, the property that SPIHT yields progressive transmission with practically no penalty in compression efficiency applies to lossless compression too. Below are examples of Lossless and lossy (200:1) images decoded from the same file.

- **Rate or Distortion Specification:** Almost all image compression methods developed so far do not have precise rate control. For some methods you specify a target rate, and the program tries to give something that is not too far from what you wanted. For others you specify a "quality factor" and wait to see if the size of the file fits your needs. (If not, just keep trying...) The embedded coding property of SPIHT allows *exact* bit rate control, without any penalty in performance (no bits wasted with padding or whatever). The same property also allows exact mean squared-error (MSE) distortion control. Even though the MSE is not the best measure of image quality, it is far superior to other criteria used for quality specification.
- **Error Protection:** Errors in the compressed file cause havoc for practically all important image compression methods. This is not exactly related to variable length entropy-coding, but to the necessity of using context generation for efficient compression. For instance, Huffman codes have the ability to quickly recover after an error. However, if it is used to code run-lengths, then that property is useless because all runs after an error would be shifted [6][7][18]. SPIHT is not an exception for this rule. One difference, however, is that due to SPIHT's embedded coding property, it is much easier to design efficient error-resilient schemes. This happens because with embedded coding the information is sorted according to its importance, and the requirement for powerful error correction codes decreases from the beginning to the end of the compressed file. If an error is detected, but not corrected, the decoder can discard the data

after that point and still display the image obtained with the bits received before the error. Also, with bit-plane coding the error effects are limited to below the previously coded planes. Another reason is that SPIHT generates two types of data. The first is sorting information, which needs error protection as explained above. The second consists of uncompressed sign and refinement bits, which do not need special protection because they affect only one pixel. While SPIHT can yield gains like 3 dB PSNR over methods like JPEG, its use in noisy channels, combined with error protection as explained above, leads to much larger gains, like 6-12 dB. (Such high coding gains are frequently viewed with skepticism, but they do make sense for combined source-channel coding schemes.)

3.3.3. Improved Set Partitioning in Hierarchical Trees (ISPIHT):

ISPIHT is the wavelet based image compression method it provides the Highest Image Quality [1]. The improved SPIHT algorithm mainly makes the following changes. SPIHT codes four coefficients and then shifts to the next four ones. Therefore, views the four coefficients as a block. The maximum of them regarded as the compared threshold will decrease number of comparison, which is relate with the distribution of coefficient matrix. Even more, when the maximum in the block is smaller than the current threshold or equal to it, the block will be coded with only one bit instead of four zeros. Therefore, this proposed method can reduce redundancy to a certain extend. When computing the maximum threshold, the improved algorithm can initialize the maximum of every block. So, it can obviously reduce number of comparison when scanning and coding zero trees. The coefficients in nonimportant block will be coded in next scanning process or later, rather than be coded in the present scanning process. This method can implement the coefficients coded earlier to the non-important ones more adequately. Generally, wavelet transform coding for still image using SPIHT [8] algorithm can be modelled as Fig. Firstly, original image matrix goes through wavelet transform. The output wavelet coefficients are then quantized and encoded by SPIHT coder. After that, bit streams are obtained. Figure I. Wavelet transform image coding using SPIHT Traditional SPIHT has the advantages of embedded code stream structure, high compression rate, low complexity and easy to implement [9]. However, for it, there still exist several defects.

1) When scanning the list of insignificant pixels (LIP), list of insignificant sets (LIS), or list of significant pixels (LSP), the repeated coefficient comparison can increase complexity of the algorithm.

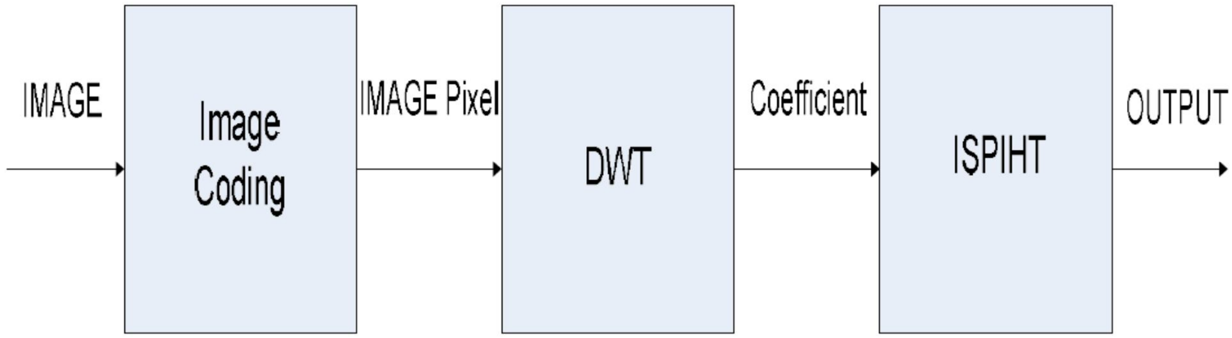


Fig 3.7: Block diagram of ISPIHT

2) The coefficients put into LIP at last scanning procedure which are smaller than the current threshold will result in redundancy.

3) Early coding for non-important coefficients in SPIHT will affect the performance of channel coding, especially unequal error protection (UEP).

Therefore, the improved SPIHT algorithm mainly makes the following changes.

- SPIHT codes four coefficients and then shifts to the next four ones. Therefore, views the four coefficients as a block [1][4]. The maximum of them regarded as the compared threshold will decrease number of comparison, which is relate with the distribution of coefficient matrix. Even more, when the maximum in the block is smaller than the current threshold or equal to it, the block will be coded with only one bit instead of four zeros. Therefore, this proposed method can reduce redundancy to a certain extend.
- When computing the maximum threshold, the improved algorithm can initialize the maximum of every block. So, it can obviously reduce number of comparison when scanning and coding zero trees.
- The coefficients in non-important block will be coded in next scanning process or later, rather than be coded in the present scanning process [1]. This method can implement the coefficients coded earlier to the non-important ones more adequately. On the basis of above-mentioned ideas for algorithm improvement, an improved algorithm is proposed and briefly describe it in the following paragraphs. In order to comprehend conveniently, symbols are given firstly. B(i,j) which represents a wavelet coefficient block with coordinate (i, j) includes four coefficients (i, j), (i+ 1, j), (i, j+l) and (i+l, j+l), like SPIHT described in detail in and will be divided into its four offsprings with coordinates (2i, 2j), (2i+2,2 j), (2i, 2j+2) and (2i+2, 2j+2) .
    - O(i, j): set of coordinates of all off springs of B(i,j).

- D(i, j): set of coordinates of all descendants of B(i,j) .
- L(i, j)=D(i, j)-O(i, j).
- LSP=*{(i, j)| (i, j) ∈H}* and LIS have the same definitions as in [5]. But the set in LIS represents either *D(i, j)* or *L(i, j).* to distinguish them, we say that type D represents *D(i, j)* and type *L(i, j).* Define list of insignificant block as LIB= *{B(i, j)|(i, j) ∈H}* instead of LIP. It stores the first coordinate of a group of 2x2 adjacent pixels which are regarded as a block. H stands for the wavelet coefficient matrix. Our algorithm encodes the sub band pixels by performing initialization and a sequence of sorting pass, refinement pass and quantization-step updating. However, differences of initialization and sorting pass still exist between the modified SPIHT[3] and traditional SPIHT.

4) **Initialization:** *LIB={B(0, 0), B(0,2), B(2,0), B(2,2)}, LIS={D(0, 2), D(2,0), D(2,2)}, T=2n, Ci,j* is wavelet matrix coefficient and LSP is empty. n is expressed in below..

$$n = \left[\log_2 \left(\max_{(i,j)} \left| c_{i,j} \right| \right) \right]$$

Ci,j is the matrix coefficient after DWT and (i, j) is the coordinate of Ci,j

5) **Sorting pass:** The sorting pass consists of two tests: the LIB test (LIBT) and LIS test (LIST). The LIBT will code the block or coefficients in blocks, while the LIST mainly disposes the sets in LIS. In each LIBT, if the maximum value of the coefficient block is smaller than the current threshold, the block is insignificant and ° is the coded bit. Otherwise, 1 will be output and represented the significance of the block. Then, the four coefficients will be respectively compared to the current threshold. When the coefficient has not been put into LSP, if it is insignificant. Otherwise, 10 or 11 represent significant negative sign or significant positive sign, respectively. After that, it will be removed from the block and added to the tail of LSP. While the test is finished, the block will be removed from LIB if all the four coefficients have been put into LSP. Otherwise, the block will be tested again in next LIBT. While in LIST, the set in LIS will be tested and coded according to its type. For type D, if the maximum coefficient in D(i,j) is smaller than the current threshold, the set is insignificant. Otherwise, the significant bit I will be coded and D(i, j) will be divided into its children tree and four blocks with coordinate (m, n) E Q(i, j) rather than four adjacent coefficients. The four blocks will be coded with the style as in LIBT, but we should add them to the tail of LIB corresponding to their significances. After coding the four blocks, our algorithm will later *D(i, j) to L(i, j)* and add *L(i, j)* to the tail of *LIS if D(i, j)* has grandson coefficients. Then, set *D(i, j)* will be removed from LIS.

For type L, if the maximum coefficient in *L(i, j)* is smaller than the current threshold, 0 will be output and represented the insignificance of the set. Otherwise, the significant bit 1 will be coded and L(i, j) will be divided into four sets *D(m, n), (m, n) ∈Q(i, j)* which will be added to the tail of LIS. Then, set *L(i, j)* will be removed from LIS.

After completing LIPT and LIBT tests, we perform the same refinement pass and updating the threshold as in traditional SPIHT.

For the Modified SPIHT, when the maximum value of a coefficient block put into LIB is small enough, only one bit will used to represent it and the four coefficients of it will not be coded until the current threshold is smaller than the maximum value. Therefore, this algorithm can better avoid repeat coding and early coding for non-important coefficients better. Moreover, this algorithm has the same scanning order and method to determine importance of wavelet coefficients as SPIHT [2][3]. Consequently, it will inherit many advantages of SPIHT.

Algorithm: ISPIHT Coding

Output: Bit stream

Input: Wavelet co-efficient or data matrix to be coded, A, the number of threshold levels, N.

Assign LIP={ A(1,1), A(1,2), A(2,1), A(2,2)}

Assign LIS with VTs for coordinates (1,3), (3,1) and (3,3) as type-0 descendent trees.

Compute Vm, M, Threshold (as described in the Initialization).

Assign Lp1=0;

Comment: Sorting Pass

For I=1 to N

Comment: LIP Testing

For each pixel in the LIP

    If a pixel is significant

        Send a 1, followed by sign bit.

        Delete the pixel form LIP and append its absolute value to LSP.

    Else

        Send a 0

    End **if**

    End **For each**

Comment: LIS Testing

For each VT in LIS

If the type of VT is 0

If VT is significant

Send a 1

For each of the four pixels associated with the node of the VT.

If a pixel is significant,

Send a 1 followed by sign bit.

Append the absolute value of the pixel to LSP.

Else

Send a 0

Append the pixel to LIP

End **If**

End **For each**

If VT has more than 1 element

Neglect the first element, change its stype to 1 and append to LIS.

End **If**

Else

Send a 0

End **If**

Else

If VT (of type-1) is significant

Send a 1

Generate VT corresponding to the four top leftmost pixel co-ordinates of the four 2x2 sub-matrices associated with current node.

Delete the VT from the LIS.

Else

Send a 0

End **If**

End **If**

End **For each**

<u>Comment: Refinement Pass</u>

For r=1 to Lp1

```
    If (LSP(r) –Threshold)>=Threshold/2
        Send a 1
    Else
        Send a 0
    End if
    End For
Lp1=No. of pixels in the LSP
Threshold=Threshold/2;
End For.
```

3.3.4.HUFFMAN ENCODING:

Huffman coding is an entropy encoding algorithm mainly used for lossless data compression. The term refers to encoding a source symbol (such as a character in a file) use a variable length code table, where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It uses a specific method for choosing the representation for each symbol, resulting in a prefix code that expresses the most common source symbols using shorter strings of bits than are used for less common source symbols.

The Huffman algorithm is mainly based on statistical coding, which means that the probability of a symbol has a direct bearing on the length of its representation. More probable of occurrence of a symbol have shorter will be its bit size representation. In any file, certain characters are used more than the others. Using binary representation, the number of bits required to represent each character depends upon the no.of characters that have to be represented. Using one bit we can represent two characters, i.e., '0' represents the first character and '1' represents the second character. Using two bits we can represent four characters. And so on[21].

Unlike ASCII code, which is a fixed length code using seven bits per character, Huffman encoding compression is a variable length coding system that assigns smaller codes for more frequently used characters and larger codes for less frequently used characters in order to reduce the size of files being compressed and transferred[26].

The decoding algorithm is just the reverse of the coding algorithm. But a difference with coding algorithm is that the LIP and LSP are stored as co-ordinates [1] and the LIS stores only the pixel co-ordinates of the topmost modes of the descendent trees and does not store VTs. Because during the decoding, testing whether a descendent tree is significant or not requires only whether the corresponding bit is 1 or zero, it does not require exhaustive searching as in the case of coding. In other words decoding remains the same as in SPIHT except the way of representation of the tree structure.

3.4. Terms used in Image Compression:

There are various types of terms that are used in calculation of image compression. Some are listed below:

3.4.1. Peak signal to noise ratio:

The phrase peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation [14][18]. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

The PSNR is most commonly used as a measure of quality of reconstruction in image compression etc. It is most easily defined via the mean squared error (MSE) which for two $m \times n$ monochrome images I and K where one of the images is considered a noisy approximation of the other is defined as:

$$MSE = \frac{1}{MK}\sum_{i=0}^{K-1}\sum_{j=0}^{M-1}\left| f(i,j) - f_1(i,j)\right|^2$$

The PSNR is defined as:

$$PSNR = 10 * \log_{10}((MAX_1^2)^2 / MSE) = 20 * \log_{10}\left(\frac{MAX_1}{\sqrt{MSE}}\right)$$

Here, *MAXi* is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using linear PCM with *B* bits per sample, *MAXI* is $2^B - 1$.

For color images with three RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all squared value differences divided by image size and by three. An identical image to the original will yield an undefined PSNR as the MSE will become equal to zero due to no error. In this case the PSNR value can be thought of as approaching infinity as the MSE approaches zero; this shows that a higher PSNR value provides a higher image quality. At the other end of the scale an image that comes out with all zero value pixels (black) compared to an original does not provide a PSNR of zero [17][19]. This can be seen by observing the form, once again, of the MSE equation. Not all the original values will be a long distance from the zero value thus the PSNR of the image with all pixels at a value of zero is not the worst possible case.

3.4.2. Signal-to-noise ratio:

It is an electrical engineering concept, also used in other fields (such as scientific measurements, biological cell signaling), defined as the ratio of a signal power to the noise power corrupting the signal. In less technical terms, signal-to-noise ratio compares the level of a desired signal (such as music) to the level of background noise [3]. The higher the ratio, the less obtrusive the background noise is.

In engineering, signal-to-noise ratio is a term for the power ratio between a signal (meaningful information) and the background noise:

$$SNR = \frac{P_{Signal}}{P_{Noise}} = \left(\frac{A_{Signal}}{A_{Noise}}\right)^2$$

Where *P* is average power and *A* is RMS amplitude. Both signal and noise power (or amplitude) must be measured at the same or equivalent points in a system, and within the same system bandwidth.

Because many signals have a very wide dynamic range, SNRs are usually expressed in terms of the logarithmic decibel scale. In decibels, the SNR is, by definition, 10 times the logarithm of the power ratio. If the signal and the noise is measured across the same impedance then the SNR can be obtained by calculating 20 times the base-10 logarithm of the amplitude ratio:

$$SNR(db) = 10 * \log_{10}\left(\frac{P_{Signal}}{P_{Noise}}\right) = 20 * \log_{10}\left(\frac{A_{Signal}}{A_{Noise}}\right)^2$$

In image processing, the SNR of an image is usually defined as the ratio of the mean pixel value to the standard deviation of the pixel values. Related measures are the "contrast ratio" and the "contrast-to-noise ratio". The connection between optical power and voltage in an imaging system is linear. This usually means that the SNR of the electrical signal is calculated by the 10 log rule[8][12][32]. With an interferometric system, however, where interest lies in the signal from one arm only, the field of the electromagnetic wave is proportional to the voltage (assuming that the intensity in the second, the reference arm is constant). Therefore the optical power of the measurement arm is directly proportional to the electrical power and electrical signals from optical interferometer are following the 20 log rule. The Rose criterion (named after Albert Rose) states that an SNR of at least 5 is needed to be able to distinguish image features at 100% certainty. An SNR less than 5 means less than 100% certainty in identifying image details.

3.4.3. Mean Square Error:

In statistics, the mean square error or MSE of an estimator is one of many ways to quantify the amount by which an estimator differs from the true value of the quantity being estimated. As a loss function, MSE is called squared error loss. MSE measures the average of the square of the "error" [9][10]. The error is the amount by which the estimator differs from the quantity to be estimated. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate [22]. The MSE is the second moment (about the origin) of the error, and thus incorporates both the variance of the estimator and its bias. For an unbiased estimator, the MSE is the variance. Like the variance, MSE has the same unit of measurement as the square of the quantity being estimated. In an analogy to standard deviation, taking the square root of MSE yields the root mean square error or RMSE, which has the same units as the quantity being estimated; for an unbiased estimator, the RMSE is the square root of the variance, known as the standard error. Definition and basic properties

The MSE of an estimator θ^1 with respect to the estimated parameter θ is defined as

$$MSE(\theta^1) = E((\theta^1 - \theta)^2)$$

The MSE can be written as the sum of the variance and the squared bias of the estimator

$$MSE(\theta^1) = Var(\theta^1) + (Bias(\theta^1, \theta))^2$$

The MSE thus assesses the quality of an estimator in terms of its variation

In a statistical model where the estimate is unknown, the MSE is a random variable whose value must be estimated. This is usually done by the sample mean

$$MSE(\theta^1) = \frac{1}{n}\sum_{j=1}^{n}(\theta_j - \theta)^2$$

Chapter- 4

Implementation

Implementation of EZW algorithm with Huffman encoding

Implementation of SPIHT Algorithm with Huffman encoding

Implementation of Modified SPIHT algorithm

4. IMPLEMENTATION

4.1. Implementation of EZW algorithm with Huffman encoding:

Coding the wavelet coefficients is performed by determining two lists of coefficients[15]:

1. The dominant list D contains information concerning significance of coefficients, which will be coded using Huffman encoding.

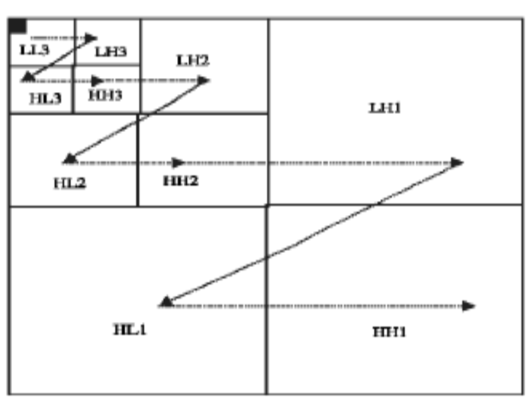
2. The significant list S contains the amplitude values of the significant coefficients, which will undergo uniform scalar quantization followed by Huffman encoding.

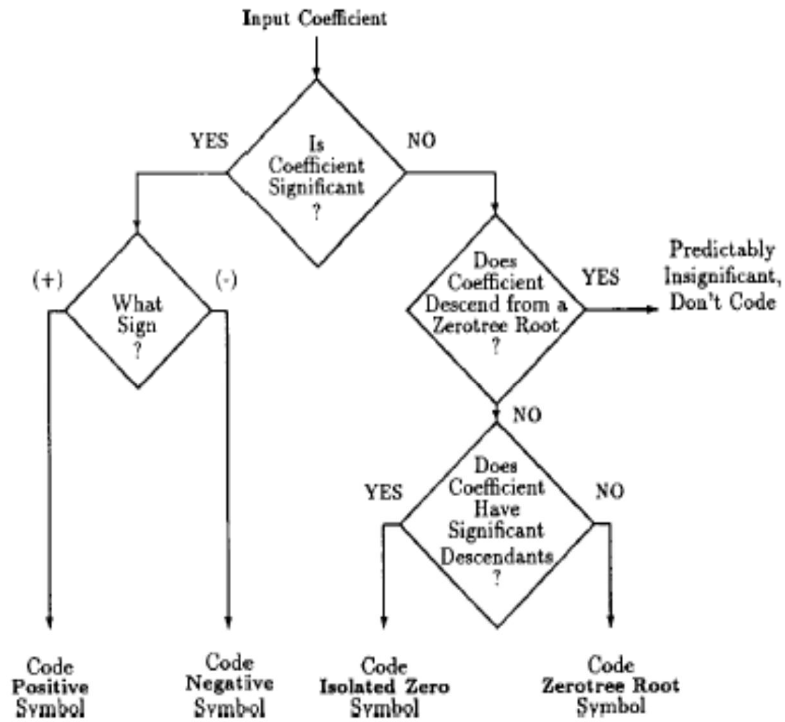
63	-34	49	10	7	-13	12	7
-31	23	-14	-13	3	4	6	1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	9
-5	9	-1	47	4	-6	-2	2
3	0	-3	2	2	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

Fig.4.1 Example of decomposition to three resolutions for an 8*8 matrix

Significance test :

The wavelet transform coefficients are scanned for the path as shown in the fig below. In our implemented method, we have used Mortan scan, which is more accurate and produces standard results





Flow chart for encoding a co-efficient of the significance map

EZW coding algorithm:

Each coefficient is assigned a significance symbols (P, N, Z, T), by comparing with the actual threshold.

1. P (significance and positive): if the absolute value of the coefficient is higher than the threshold T and is positive.

2. N (significance and positive): if the absolute value of the coefficient is higher than the threshold T and is negative.

3. T (zerotree): if the value of the coefficient is lower than the threshold T and has only insignificant descendants

4. Z (isolated zero): if the absolute value of the coefficient is lower than the threshold T and has one or more significant descendents.

The insignificant coefficients of the last sub bands, which do not accept descendents and are not themselves descendents of a zerotree, are also considered to be zero tree.

The significance symbols are then placed in a list D which is subjected to Huffman encoding.

The dominant list and the significance list are shown below:

D: P N Z T P T T T T Z T T T T T T T P T T

S: 1 0 1 0

Huffman coding algorithm:

The steps involved in encoding dominant list D is as follows:

5. In the dominant list since the probability of occurrence of the symbol T is more when compared to others, this symbol should be coded with the less number of bits.

6. The other symbols probability of occurrence are less when compared to the symbol T, they should be coded with more number of bits.

7. After encoding all the symbols with binary digits, a separator is appended to the end of the encoded bit stream to indicate the end of the stream.

For eg:

In our algorithm to encode the symbols P, N, Z and T we use the binary bits as follows:

P is encoded as 1110

N is encoded as 110

Z is encoded as 10

since the probability of occurrence is less when compared to T.

T is encoded as 0 (since the probability of occurrence is more when compared to other bits)

Then we insert a separator bits i.e. a stream of 1‟s .Here we used 11111 to indicate the end of the bit stream.

For decoding from the encoded bit stream the dominant list symbols are found and from these symbols the pixels values of the image are predicted.

4.2. Implementation of the SPIHT Algorithm with Huffman encoding:

Initialization:

1) The wavelet-transformed image is searched for the largest magnitude which defines the bit plane k with the highest significance.

2) The lists are initialized: The empty set is assigned to the LSC since no coefficient is significant yet. The tree roots 7H are added to the LIC and those with descendants to the LIS.

Sorting Pass:

1) Coefficients of coordinates in the LIC are tested for significance according to (1) and correspondingly moved to the LSC list.

2) Sub trees defined by the entries in the LIS are searched for significant coefficients. Those found to be significant including their siblings and ancestors but not the roots are added either to LSC or LIC.

3) New sub trees are defined where significant coefficient were found.

Refinement Pass:

1) Only the coordinates of the k-th bit plane which are significant are transmitted.

2) In case it has not been the least significant bit plane, k is decremented by one and the algorithm starts with another sorting pass.

After getting the bitstream of spiht encoding that bit stream can applied to huffman encoding(same as above EZW with Huffman encoding).

4.3. Implementation of Modified SPIHT algorithm:

On the basis of above mentioned ideas for algorithm improvement, we propose a modified algorithm and briefly describe it in the following paragraphs.

In order to comprehend conveniently, symbols are given firstly.

B(i,j) which represents a wavelet coefficient block with coordinate *(i, j)* includes four coefficients *(i, j), (i+1, j), (i, j+1), (i+1,j+1),* like SPIHT described in detail in [3] and will be divided into its four offsprings with coordinates *(2i, 2j), (2i+2, 2j), (2i, 2j+2) and (2i+2, 2j+2).*

O(i, j): set of coordinates of all offsprings of *B(i, j).*

D(i, j): set of coordinates of all descendants of *B(i, j).*

L(i, j)=D(i, j)-O(i, j).

LSP=*{(i, j)| (i, j) ∈H}* and LIS have the same definitions as in [5]. But the set in LIS represents either *D(i, j)* or *L(i, j).* to distinguish them, we say that type D represents *D(i, j)* and type *L(i, j).* Define list of insignificant block as LIB= *{B(i, j)|(i, j) ∈H}* instead of LIP. It stores the first coordinate of a group of 2x2 adjacent pixels which are regarded as a block. H stands for the wavelet coefficient matrix.

Our algorithm encodes the sub band pixels by performing initialization and a sequence of sorting pass, refinement pass and quantization-step updating. However, differences of initialization and sorting pass still exist between the modified SPIHT and traditional SPIHT.

Sequentially, we will describe the initialization and sorting pass of the modified SPIHT.

1) ***Initialization*** *:*
   LIB={B(0, 0), B(0,2), B(2,0), B(2,2)}, LIS={D(0, 2), D(2,0), D(2,2)}, T=2n, Ci,j is wavelet matrix coefficient and LSP is empty. N is expressed in below..

$$n=\left[\log_2\left(\max_{(i,j)}\left|c_{i,j}\right|\right)\right]$$

2) ***Sorting pass****:*
   The sorting pass consists of two tests: the LIB test (LIBT) and LIS test (LIST). The LIBT will code the block or coefficients in blocks, while the LIST mainly disposes the sets in LIS.

In each LIBT, if the maximum value of the coefficient block is smaller than the current threshold, the block is insignificant and '0' is the coded bit. Otherwise, '1' will be output and represented the significance of the block. Then, four coefficients will be respectively compared to the current threshold. When the coefficients have not been put into LSP, '0' is emitted if it is insignificant. Otherwise, 10 or 11 represent significant negative sign or significant positive sign, respectively. After that, it will be removed from the block and added to the tail of LSP. While the test is finished, the block will be removed from LIB if all the four coefficients have been put into LSP. Otherwise, the block will be tested again in next LIBT.

While in LIST, the set in LIS will be tested and coded according to its type.

For type D, if the maximum coefficients in *D(i, j)* is smaller than the current threshold, the set is insignificant and 0 is emitted. Otherwise, the significant bit 1 will be coded and *D(i, j)* will be divided into its children tree and four blocks with coordinate *(m, n) ∈Q(i, j)* rather than four adjacent coefficients.

The four blocks will be coded with the style as in LIBT, but we should add them to the tail of LIB corresponding to their significances. After coding the four blocks, our algorithm will later *D(i, j) to L(i, j)* and add *L(i, j)* to the tail of *LIS if D(i, j)* has grandson coefficients. Then, set *D(i, j)* will be removed from LIS.

For type L, if the maximum coefficient in *L(i, j)* is smaller than the current threshold, 0 will be output and represented the insignificance of the set. Otherwise, the significant bit 1 will be coded and L(i, j) will be divided into four sets *D(m, n), (m, n) ∈Q(i, j)* which will be added to the tail of LIS. Then, set *L(i, j)* will be removed from LIS.

After completing LIPT and LIBT tests, we perform the same refinement pass and updating the threshold as in traditional SPIHT. For the Modified SPIHT, when the maximum value of a coefficient block put into LIB is small enough, only one bit will used to represent it and the four coefficients of it Will not be coded until the current threshold is smaller than the maximum value. Therefore, this algorithm can better avoid repeat coding and early coding for non-important coefficients better. Moreover, this algorithm has the same scanning order and method to determine importance of wavelet coefficients as SPIHT. Consequently, it will inherit many advantages of SPIHT, such as embedded bit stream, excellent rate-distortion performance and so on.

Chapter- 5

Experimental Results

5. EXPERIMENTAL RESULTS

A. PSNR comparison:

Computational formula of PSNR and mean square error(MSE).

$$PSNR = 10 * \lg\left[\frac{(2^n - 1)^2}{MSE}\right]$$

$$MSE = \sum_{i=0}^{K-1}\sum_{j=0}^{M-1}\frac{\left(f(i,j) - f1(i,j)\right)^2}{M * K}$$

Firstly, original image is applied to the compression program, EZW encoded image is obtained. To reconstruct compressed image, compressed image is applied to decompression program, by which EZW decoded image is obtained. Compression Ratio (CR) and Peak-Signal-to-Noise Ratio (PSNR) are obtained for the original and reconstructed images. In the experiment the original image 'Cameraman.tif' having size 256 x 256 (65,536 Bytes). The different statistical values of the image cameraman.tif for Various Thresholds are summarized in the table. Thus, it can be concluded that EZW encoding gives excellent results. By choosing suitable threshold value compression ratio as high as 8 can be achieved.

Results for the image Cameraman.tif for various thresholds given below:

Figures and Tables:

5.1:Tabular form of direct apply of ezw encoder and without huffman encoder:

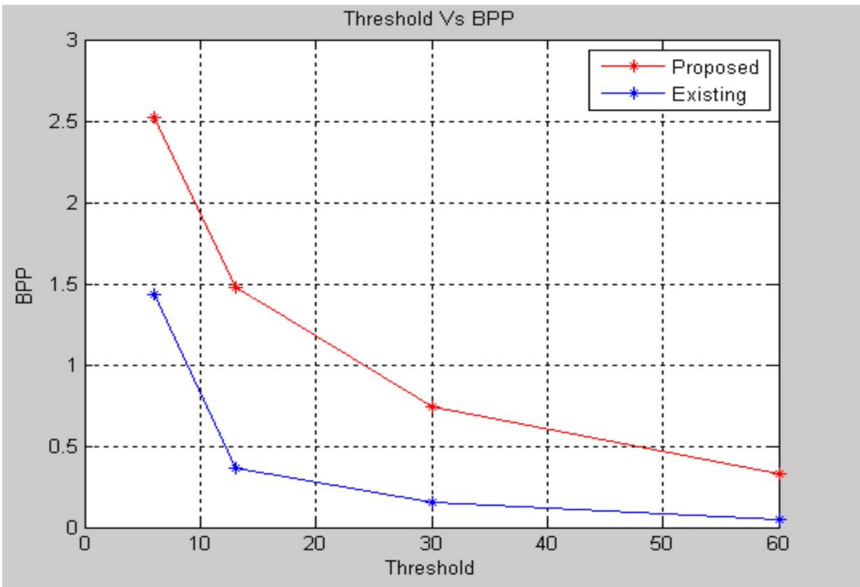
Parameter	Th=6	Th=10	Th=30	Th=60
Compression Ratio	5.57	10.37	52.19	151.36
PSNR	35.55	30.75	23.18	20.30
Bpp	1.43	0.77	0.15	0.05
Encoding time	182.62	102.21	43.34	35.47
Deccoing time	448.86	272.37	78.54	37.18
Original size	65240	65240	65240	65240
Compressed size	11697	6287	1250	431

5.2:Tabular form of by combining both ezw encoder and huffman encoder:

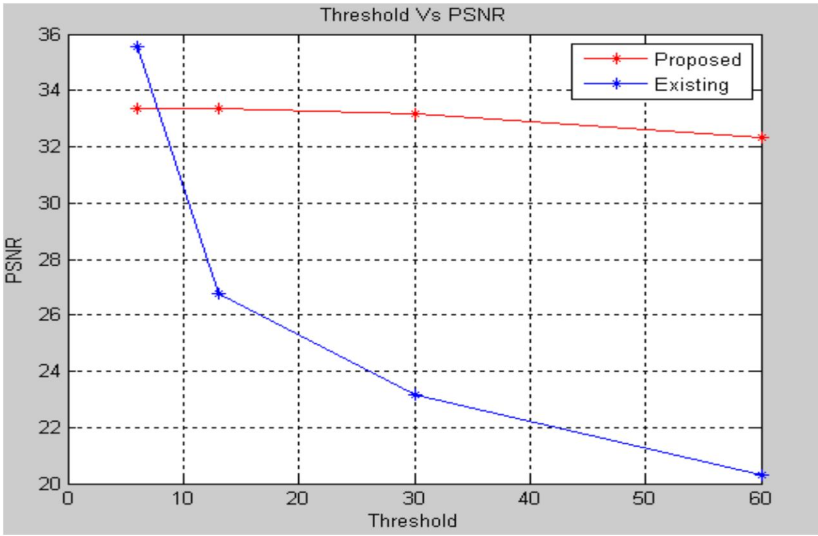
Parameter	**TH=6**	**TH=10**	**TH=30**	TH=6**0**
Compression Ratio	5.83	6.00	6.56	7.73
PSNR	33.3617	33.3704	33.1611	32.33
Bpp	2.52	1.48	0.74	0.33
Encoding Time (Sec)	295.09	175.29	99.36	56.87
Decoding Time (Sec)	867.62	611.16	359.83	202.98
Original File Size(bytes)	65240	65240	65240	65240
Compressed Size	11186	10870	9944	8437

5.1: Comparison plots:

THR Vs BPP



THR Vs PSNR



5.3: Reconstructed Images by combining of EZW algorithm with Huffman encoding:

Original Image

Threshold=6

Threshold=10

Threshold=30

Threshold=60

Here below tabular forms shows the comparison of PNSRs of SPIHT and EZW algorithms in different wavelet families at different bit rates, and the input image is **boat.png.**

Wherein, n is the number of every pixel bit, M and K are the length and width of the matrix.

5.3:Tabular form for PSNRS of various Wavelet Families applied to SPIHT Image Compression Algorithm:

Bit rate (bpp)	PSNR's (dB) of different wavelet families											
	Db1	*Db2*	*Db4*	*Db8*	*Db10*	*Bior1.1*	*Bior2.2*	*Bior4.4*	*Bior6.8*	*Coif1*	*Coif 4*	*Coif 5*
0.1	24.88	25.57	25.92	25.76	25.76	24.88	25.92	26.20	26.21	25.64	26.01	25.99
0.2	27.01	27.89	28.27	28.11	28.07	27.01	28.23	28.50	28.57	27.95	28.40	28.41
0.3	28.34	29.37	29.81	29.63	29.57	28.34	29.82	30.22	30.23	29.43	29.96	29.96
0.4	29.64	30.72	31.24	31.11	31.03	29.64	31.14	31.55	31.63	30.77	31.44	31.42
0.5	30.66	31.71	32.19	32.14	32.08	30.66	32.09	32.56	32.61	31.76	32.42	32.39
0.6	31.43	32.55	33.06	33	32.94	31.43	33.13	33.40	33.51	32.62	33.29	33.26
0.7	32.19	33.29	33.92	33.88	33.82	32.19	33.84	34.27	34.36	33.36	34.17	34.16
0.8	32.86	34.08	34.65	34.63	34.58	32.86	34.44	34.92	35.03	34.15	34.89	34.90
0.9	33.63	34.71	35.20	35.19	35.17	33.63	34.98	35.44	35.55	34.77	35.41	35.42
1	34.23	35.20	35.69	35.68	35.67	34.23	35.46	35.97	36.07	35.26	35.90	35.91

5.4:Tabular form for PSNRS of various Wavelet Families applied to EZW Image Compression Algorithm:

Threshold	Bior4.4		Bior6.8		Db4		Db10		Coif4		Coif5	
	Bitrate	*PSNR*	*Bitrate*	*PSNR*	*Bitrate*	*PSNR*	*Bitrate*	*PSNR*	*Bitrate*	*PSNR*	*Bitrate*	*PSNR*
Th=100	0.08	24.38	0.09	24.73	0.09	24.41	0.09	24.26	0.09	24.49	0.09	24.59
Th=80	0.08	24.38	0.09	24.73	0.09	24.41	0.09	24.26	0.09	24.49	0.09	24.59
Th=50	0.22	26.56	0.24	26.89	0.24	26.48	0.25	27.07	0.24	26.70	0.24	27.03
Th=30	0.52	27.83	0.56	28.05	0.58	27.71	0.61	28.71	0.58	27.92	0.57	28.45
Th=20	0.52	27.83	0.56	28.05	0.58	27.71	0.61	28.71	0.58	27.92	0.57	28.45
Th=10	1.1	28.25	1.16	28.40	1.22	28.14	1.27	29.26	1.19	28.35	1.20	28.96

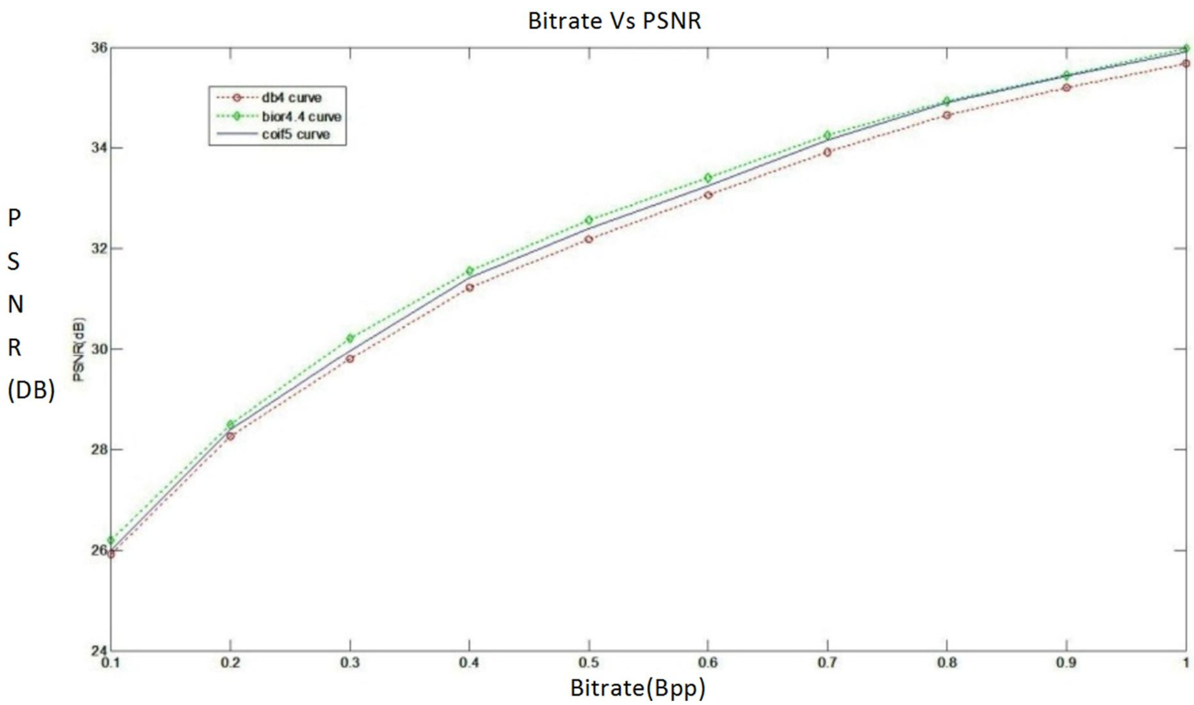


Fig.5.4: Comparative evolution of different wavelet families using SPIHT algorithm with Boat.png image.

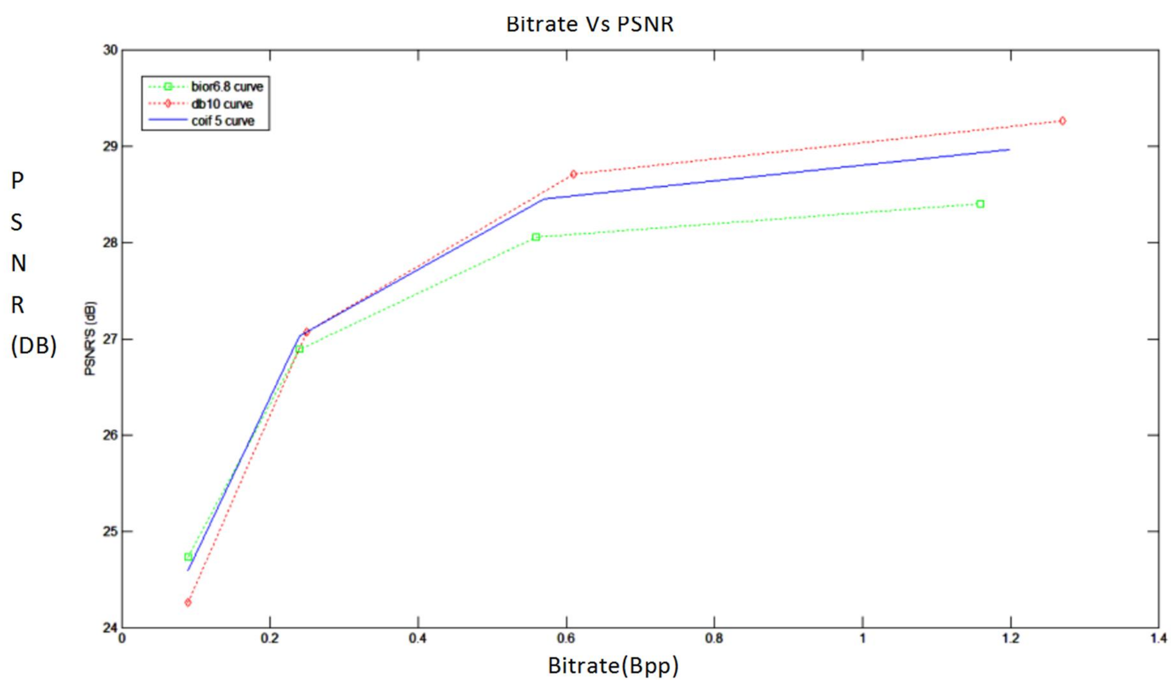


Fig.5.5: Comparative evolution of different wavelet families using EZW algorithm with boat.png image

5.6: Reconstructed images by combining of spiht algorithm with Huffman encoding:

Original Image

Db 4 Reconstructed Image:

PSNR=35.69 at bitrate=1

Bior6.8 Reconstructed Image:

PSNR=36.07 at bitrate=1

Coif 5 Reconstructed Image:

PSNR=35.91 at bitrate=1

Chapter- 6

Conclusion and Future Scope

6. Conclusion and Future scope:

The EZW having the embedding and adaptive arithmetic property that contains all the lower rate encoding of the same algorithms . This technique is flexible yet it is not essential to know the range of number being compressed. An encoder can stop its encoding process at any point and decoder can cease decoding at any point. Though it allow a final rate or final distortion metric to be met exactly, same for decoder still it's produce the same image that is encoded. It will constantly produce the same compression ratio in comparison with all algorithms on test image.

SPIHT is the improved version of EZW. It improves the coding performance by reducing the redundant coefficients across the sub bands. It has many advantages, such as good image quality, high PSNR and good progressive image transmission. Hence, it also has wider application in the compression of images. A typical successful example was that an improvement to SPIHT has to be used to compress the images. Although the improvement made the memory space requirement to be optimized by some additional means.

In order to solve the memory space problem, a new algorithm using ISPIHT and ROI has been used in this research work. The size and execution time of this new algorithm is much less than that of the original one.

In this thesis implemented the SPIHT and EZW algorithms with huffman encoding using different wavelet families and then compare the PSNRs and bitrates of these families. Among these different wavelet families, in the biorthogonal wavelet family 'bior4.4 & bior 6.8' wavelet types, in the daubechies wavelet family 'db4 & db10' wavelet types, and in the coiflet wavelet family 'coif5' wavelet types having good PSNR at low bitrates. At lower bit rates, the PSNR is almost identical to the original and modified versions but at higher bit rates, the PSNR is higher for the modified algorithm than the original one. These algorithms were tested on different images, and it is seen that the results obtained by these algorithms have good quality and high compression ratio as compared to the previous lossless image compression techniques.

Future work :

This proposed work in this thesis having a lot potential for further research in the area of image compression using embedded wavet based codings. Embedded wavelet based coding techniques have good quality and high compression ratio at low bit rates. The research can be extended to compress the colour images, using these proposed wavelet based algorithms with high PSNR at low bitrates.

Bibliography:

[1] Yumnam Kirani Singh "ISPIHT-Improved SPIHT "A simplified and efficient subband coding scheme" Center for Development of Advanced Computing Plot: E-2/1, Block GP, Sector V, Salt Lake Electronics Complex.

[2] Hualiang Zhu, Chundi Xiu and Dongkai Yang " An Improved SPIHT Algorithm Based on Wavelet Coefficient Blocks for Image Coding " Beijing University of Aeronautics and Astronautics Beijing, P.R.China

[3] A.Said and W.APeariman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. Circuits and Systems for Video Technology, vol. 6, no.3, pp.243-250, Jun.1996.

[4] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Signal Proc., Vol. 41, No. 12, pp. 3445-3462, 1993.

[5] 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.

[6] 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.

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

[8] F. W. Wheeler, and W. A. Pearlman, "SPIHT Image Compression without Lists," IEEE Int. Conf on Acoustics, Speech and Signal Processing(ICASSP 2000). Istanbul: IEEE, 2000.2047-2050.

[9] Jianxiong Wang "Study of the Image Compression based on SPIHT Algorithm" college of water resources & hydropower and architecture, yunnan agriculture university Kunming.

[10] Min HU, Changjiang Zhang , Juan LU, Bo Zhou "A Multi-ROIs Medical Image Compression Algorithm with Edge Feature Preserving" Zhejiang normal university.

[11] Jia ZhiGang Guo XiaoDong Li LinSheng "A Fast Image Compression Algorithm Based on SPIHT " College of Electronic and Information Engineering TaiYuan University of Science and Technology TaiYuan, ShanXi, China.

[12] Jianjun Wang "Modified SPIHT Based Image Compression Algorithm for Hardware Implementation" Xi'an Institute of Optics and Precision Mechanics Chinese Academy of Sciences Xi'an.

[13] LIU Wei "Research on Image Compression Algorithm Based on SPHIT" School of Equipment and Engineering ShenYang Ligong University Shenyang, P.R.China.

[14] Chunlei Jiang "A Hybrid Image Compression Algorithm Based on Human Visual System" Electrical and Information Engineering College Northeast Petroleum University Daqing, Heilongjiang Province.

[15] Janaki.R, Dr.Tamilarasi.A, "Still Image Compression by Combining EZW Encoding with Huffman Encoder", International Journal of Computer Applications, Vol.13-No.7,January 2011.

[16] Macarena Boix , Begoña Cantó "Wavelet Transform application to the compression of images" a Departamento de Matemática Aplicada, Universidad Politécnica de Valencia, Escuela Politécnica Superior de Alcoy, Plaza Ferrándiz y Carbonell 2, 03801 Alcoy (Alicante), Spain b Instituto de Matemática Multidisciplinar, Universidad Politécnica de Valencia, 46022 Valencia, Spain.

[17] Stephan Rein , Martin Reisslein," Performance evaluation of the fractional wavelet filter: A low-memory image wavelet transform for multimedia sensor networks" a Telecommunication Networks Group, Technical University Berlin, Berlin b School of Electrical, Computer, and Energy Eng., Goldwater Center, MC 5706, Arizona State University, Tempe, AZ 85287-5706, United States.

[18] J. Jyotheswar, Sudipta Mahapatra "Efficient FPGA implementation of DWT and modified SPIHT for lossless image compression" Department of Electronics and Electrical Communication Engineering, IIT Kharagpur, Kharagpur 721 302, West Bengal, India.

[19] Peter Schelkens, Adrian Munteanu, Jan Cornelis "Wavelet-based compression of medical images: Protocols to improve resolution and quality scalability and region-of-interest coding" Department of Electronics and Information Processing (ETRO), Vrije Universiteit Brussel, Pleinlaan.

[20] 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.

[21] Tripatjot Singh, Sanjeev Chopra, Harmanpreet Kaur, Amandeep Kaur,Image Compression Using Wavelet and Wavelet Packet Transformation,IJCST Vol.1, Issue 1, September 2010.

[22] Nicholas Kolokotronis, Aliki Vassilarakou, 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 .

[23] Lalitha Y. S, M.V.Latte "Lossless and Lossy Compression of DICOM images With Scalable ROI" Appa Institute of Engineering & Technology, Gulbarga, Karnataka, India. JSS Institute & Academy, Bangalore, India

[24] K.V.Sridhar, Prof. K.S.R.Krishna Prasa " MEDICAL IMAGE COMPRESSION USING ADVANCED CODING TECHNIQUE" National Institute of Technology, Warangal (AP)- India.

[25] A New Fast and Efficient "Image Codec Based on Set Partitioning in Hierarchical Trees "Amir Said Faculty of Electrical Engineering P.O. Box. State University of Campinas Brazil William A. Pearlman Department of Electrical Computer and Systems Engineering Rensselaer Polytechnic Institute.

[26] Kharate G. K., Patil V. H., "Color Image Compression Based On Wavelet Packet Best Tree," IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 3, March 2010.

[27] Nikola Sprljan, Sonja Grgic, Mislav Grgic "Modified SPIHT algorithm for wavelet packet image coding" Multimedia and Vision Lab, Department of Electronic Engineering, Queen Mary, University of London, London E1 4NS, UK Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3/XII, HR-10000 Zagreb, Croatia.

[28] Dr.S.Shenbaga Devi "Development of Medical Image Compression Techniques" Assistant Professor, College of Engineering Anna University Anna University Guindy, Chennai.

[29] Amir Said, W.A. Pearlman, "An image multiresolution representation for lossless and lossy compression", IEEE Trans. Image Processing, Vol. 5, pp. 1303-1310, 1996.

[30] Chun-Liang Tung, Tung-Shou Chen, Wei-Hua Wang and Shiow-Tyng Yeh, "A new improvement of SPIHT progressive image transmission," Proceedings of the IEEE Fifth International Symposium on Multimedia Software Engineering, IEEE Press,Taichung, Taiwan ,Dec, 2003, pp ISO-IS7.

[31] Proc.of International conference on Advances in computer engineering 2011,Image Compression using WDR and ASWDR Technique with Different Wavelets Codes.

[32] Sprljan, S. Grgic and M. Grgic, , Aug. 2005,Modified SPIHT Algorithm for Wavelets Packet Image Coding, Real-Time Imaging, Elsevier, vol. 11, pp. 378-388

[33] To Appear in Proc ICIP 2000 Sep. 10-13, Vancouver Canada,Adaptive Scanning Methods for wavelets difference Reduction in lossy image compression.

[34] James S Walker, A low power, low memory system for wavelets based image coding.

[35] ICGST –GVIP Journal, Volume (5), Issue (6), June 2005,Image Compression using coding of wavelets coefficients-A survey.

[36] Resnikoff H. L and Wells R O, NY 1998 ,Wavelets analysis the Scalable Structure of Information Springer, New York.

Dissemination

Conference:

1. **Sure Srikanth** and Sukadev Meher, "Compression Efficiency for Combining Different Embedded Image Compression Techniques with Huffman Encoding", Proceedings of *IEEE International Conference on Communication and Signal Processing*, April 2013, APEC Chennai, India (**Published)**