IMAGE COMPRESSION IMPLEMENTATION USING MODIFIED BIORTHOGONAL WAVELET FILTERS

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE AWARD OF DEGREE OF

MASTER OF ENGINEERING

(COMPUTER TECHNOLOGY AND APPLICATION)

By

SANJAY KUMAR YADAV

Under the guidance of

Dr. S.K. SAXENA
Delhi College of Engineering

DEPARTMENT OF COMPUTER ENGINEERING
DELHI COLLEGE OF ENGINEERING
DELHI UNIVERSITY, DELHI-110042
2006

<u>Certificate</u>

This is to certify that the Dissertation entitled "**Image Compression Implementation using** Modified **Biorthogonal Wavelet Filters** " submitted by Sanjay Kumar Yadav Roll No.8531 in the partial fulfillment of the requirement for the award of degree of Master of Engineering in Computer Technology and Application, Delhi College of Engineering is an account of his work carried out under my guidance and supervision.

Dr. S. K. Saxena
Project Guide
Department Of Computer Engineering
Delhi College of Engineering
(University Of Delhi)

Acknowledgement

I am delighted to express my heartily and sincere gratitude and indebtedness to Dr. S. K. Saxena, Computer Engineering Department, Delhi College of Engineering, Delhi for his invaluable guidance and wholehearted cooperation. His continuous inspiration only has made me complete this major project.

I am greatly thankful to Prof. D. Roy Choudhary, Dr.(Mrs.) Goldi Gabarani, for his support and encouragement . My heartily thanks to all professors for their expertise and all rounded personality they have imparted to me.

I would also like to thanks to my wife and sons (Divyanshu & Amritanshu) for their support and patience, which has helped me in a great sense to complete this work.

Sanjay Kumar Yadav
Roll no. 8531
M.E.(CTA)

Abstract

Image compression is now essential to reduce data volume. In Internet browsing the **progressive transmission scheme** is efficient. Here the user sees small size coarse image reconstructed from very few bits, then more bits received the image quality is successively refined. This scheme also provides the user the option to quit any time if image found is irrelevant.

In this thesis I am implementing image compression in MATLAB using different wavelet filters and encoding the decomposed coefficients using **Embedded Zero Tree** Wavelet coding[1,3,9]. As Biorthogonal filters exhibits linear phase but not energy preserving[1,3] so I have proposed a **modification to Biorthogonal Wavelet Filter** [1,4,5] Filter which will become closer to orthogonal Wavelet Filters which are energy preserving.

CONTENTS

Certificate

This is to certify that the Dissertation entitled "**Image Compression Implementation using Modified Biorthogonal Wavelet Filters** " submitted by Sanjay Kumar Yadav Roll No.8531 in the partial fulfillment of the requirement for the award of degree of Master of Engineering in Computer Technology and Application, Delhi College of Engineering is an account of his work carried out under my guidance and supervision.

Dr.(Mrs.) Goldi Gabrani
Head (Computer Engineering)
Delhi College of Engineering
(University Of Delhi)

Dr. S. K. Saxena
Project Guide
Delhi College of Engineering
(University Of Delhi)

CHAPTER1

INTRODUCTION

1.1 **Introduction**

Image compression is essential for application such as representing high quality of images for transmission, reception, storage and display. For example, a nearly photographic image requires approximately 1,280 rows of 800 pixels each, with 24 bits for color information per pixel; that is a total of 3,072,000 bytes (3.07MB). The large data associated with images thus drive the need for extremely high compression ratios to make storage practical. Without compression, a CD with storage capacity 700 MB would only be able to store approximately 225 pictures like that above.

The information contained in the images must, therefore be compressed by extracting only visible elements, which are then encoded. The quantity of data involved is thus reduced substantially. But at the same time, the quality of image should not be degraded beyond a certain limit. Thus, the fundamental goal of image compression is to reduce the transmission bit rate or storage while maintaining the acceptable fidelity or image quality.

For some application like **progressive** transmission and image browsing, image compression is not the complete solution. There is also a growing need of scalability. The term scalability refers to methods to which allow partial decodability by the decoder in order to meet certain requirements. In progressive transmission such as image / video database browsing, the user first sees a coarse version of an image reconstructed from few bits, and as the more bits are received, the image quality is successively refined until the end of bit stream reached. This allows fast retrieval of an intelligent image, and more important, it gives user the option to terminate the transmission at any time if image is found to be irrelevant. On the other hand non-progressive transmission will require the entire bit stream to be received before the image is viewable.

Usual methods of compression are **loss less** coding techniques**.** With loss less coding, we restore every detail of original data after decoding. Obviously it is necessity for numerical, financial documents. Our tolerance of image approximation and need for high compression opens the opportunity to exploit a new for of coding **lossy** coding. Lossy coding can be applied to a data such as images and audio for which humans will tolerate some loss of fidelity (faithfulness of our reproduction of an image after compression and decompression with the original image) .

In images the neighboring pixels are correlated and therefore contain redundant information. Before we compress an image, we first find out the pixels, which are correlated. The fundamental components of compression are redundancy and irrelevancy reduction. Redundancy means duplication and Irrelevancy means the parts of signal that will not be noticed by the signal receiver, which is the Human Visual System (HVS).

There are three types of redundancy can be identified:

1. Spatial Redundancy i.e. correlation between neighboring pixel values.
2. Spectral Redundancy i.e. correlation between different color planes or spectral bands.
3. Temporal Redundancy i.e. correlation between adjacent frames in a sequence of images (in video applications).

Image compression focuses on reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies. Since this project is about still image compression, therefore temporal redundancy is not relevant.

1.2 **Objective**

The objective of lossy image compression is to store image data efficiently by reducing the redundancy of image content and discarding unimportant information while keeping the quality of image acceptable. Thus, the trade of lossy image compression is the number of bits required to represent an image and the quality of compressed image. This is usually known as rate of distortion tradeoff. However, The" closeness" between compressed and the original image is not a pure Objective measure, since human perception always plays an important role in determining the quality of the compressed image. At present, the most widely used objective distortion measure is MSE and related PSNR . It is very well known that MSE does not correlate very well with the visual quality perceived by human being . This can easily can be explained by the fact that MSE is computed by adding the square difference of individual pixels without considering the visual interaction between adjacent pixel .

In this thesis I am analyzing Biorthogonal wavelets Filters and its possible modification so that it can become more energy preserving, close to orthogonal wavelet filters. Further I am implementing Embedded Zero Tree coding and decoding of decomposed wavelet coefficients using MATLAB7.0(Release 14).

1.3 Organization of thesis

The thesis is organized as follows :

- Chapter 2 performs the study of various transform techniques , brief introduction advantages and drawbacks and performance metrics.
- Chapter 3 discusses wavelets, continuous and discrete forms and multiresolution properties ,family of wavelets. Finally literature survey regarding image compression.
- Chapter 4 explains concept of biorthogonal wavelets , multirate filters and perfect reconstruction filter design concepts.
- Chapter 5 Introduces the problems of Biorthogonal wavelets its possible solution using weighting concepts.
- Chapter 6 Explains the embedded zero tree wavelet coding , successive approximation entropy quantization ,encoding and decoding using example.
- Chapter 7 Explains implementation test condition, flow charts, Result, conclusion and future scope.

CHAPTER 2

VARIOUS TRANSFORM TECHNIQUES

2.1 **FOURIER TRANSFORM**

For a continuous function of one variable f(t), the Fourier Transform F(f) will be defined as:

$$F(f) = \int_{-\infty}^{\infty} f(t)\ e^{-j2\pi ft}\ dt$$

And the inverse transform as

$$f(t) = \int_{-\infty}^{\infty} F(f)\ e^{j2\pi ft}\ df$$

Where j is the square root of -1 and e denotes the natural exponent

$$e^{j\phi} = \cos(\phi) + j\sin(\phi).$$

DISCRETE

Consider a complex series x(k) with N samples of the form

$$x_0, x_1, x_2, x_3 \ldots x_k \ldots x_{N-1}$$

Where x is a complex number

$$x_i = x_{real} + j\, x_{imag}$$

Further, assume that that the series outside the range 0, N-1 is extended N-periodic, that is, $x_k = x_{k+N}$ for all k. The FT of this series will be denoted X(k), it will also have N samples. The forward transform will be defined as

$$X(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(k)\ e^{-jk2\pi n/N} \quad \text{for } n=0..N-1$$

The inverse transform will be defined as

$$x(n) = \sum_{k=0}^{N-1} X(k)\ e^{jk2\pi n/N} \quad \text{for } n=0..N-1$$

Of course although the functions here are described as complex series, real valued series can be represented by setting the imaginary part to 0 . In general, the transform into the frequency domain will be a complex valued function, that is, with magnitude and phase.

$$magnitude = \|X(n)\| = (x_{real} * x_{real} + x_{imag} * x_{imag})^{0.5}$$

$$phase = \tan^{-1}\left(\frac{x_{imag}}{x_{real}}\right)$$

Fourier transforms are very useful at providing frequency information that cannot be seen easily in the time domain. However they do not suit brief signals, signals that change suddenly, or in fact any non-stationary signals. The reason is that they show only what frequencies occur, not when these frequencies occur, so they are not much help when both time and frequency information is required simultaneously. In stationary signals, all frequency components occur at all times, so Fourier Transforms are very useful.

SHORT-TIME FOURIER TRANSFORM

In an effort to correct deficiency of Fourier transform, Dennis Gabor (1946) adapted the Fourier transform to analyze only a small section of the signal at a time a technique called *windowing* the signal (see Fig2.1). Gabor's adaptation, called the Short-Time Fourier

Transform (STFT), maps a signal into a two-dimensional function of time and frequency.

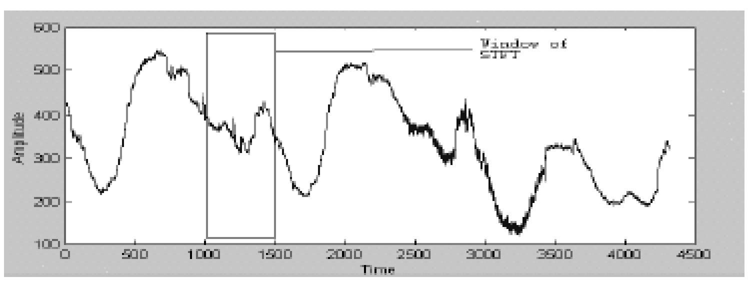


Fig.2.1 Showing Short-Time Fourier Transform

The following equation can be used to compute a STFT. It is different to the FT as it is computed for particular windows in time individually, rather than computing overall time (which can be alternatively thought of as an infinitely large window). x is the signal, and w is the window.

$$STFT_x^w(t,f) = \int [x(t).w^*(t-t')].e^{-j2\Pi ft}\,\delta t$$

This is an improvement as a time domain signal can be mapped onto a function of time and frequency, providing some information about what frequencies occur when. However using windows introduces a new problem; according to Heisenberg's Uncertainty principle it is impossible to know exactly what frequencies occur at what time, only a range of frequencies can be found. This means that trying to gain more detailed frequency information causes the time information to become less specific and visa versa. Therefore when using the STFT, there has to be a sacrifice of either time or frequency information. Having a big window gives good frequency resolution but poor time resolution small windows provide better time information, but poorer frequency information.

2.2 THE KARHUNEN-LOEVE TRANSFORM (KLT)

Originated from the series expansions for random processes developed by karhunen and loeve in 1947 and 1949 based on the work of Hoteling in 1933 (the discrete version of the kl transform). Also known as Hoteling transform or method of principal component. It packs the maximum energy in first few samples. It minimizes the mean square error for any truncated series expansions. Errors vanishes in case there is no truncation. The idea is to transform a signal into a set of uncorrelated coefficients.

K.L. TRANSFORM OF IMAGES

An N×N image is represented by a two dimensional random sequence v(m, m). it can be represented by matrix of order N×N .alternatively , a given N×N image can be viewed as an $N^2 \times 1$ column vector v. now just as one dimensional signal can be represented by an orthogonal series of basis function, an image can also be generated by unitary matrices. A general orthogonal series expansion for an n×n image v (m, n) is given as,

General form:

$$v(m,n)=\sum_{K=0}^{N-1} \sum_{1=0}^{N-1} u(k, 1)\Psi(k,1,m,n) \quad m,n=0,\dots,n-1$$

where the kernel $\Psi(k,1,m,n)$ is given by the orthonormalized eigenvectors of the correlation matrix ,i.e. it satisfies

$\lambda i \Psi i = r \Psi i$ $\quad\quad$ $i=0,..,n^2-1$

R is the $(N^2 \times N^2)$ covariance matrix of image mapped into an $(N^2 \times 1)$ vector and Ψi is the ith column of Ψ

If r is separable, i.e.,

$R= R1 \otimes R2$

Then the KL kernel is also separable,i.e.,

$\Psi(k,l;m,n)=\Psi 1(m,k)\Psi 2(n,l)$

Or

$\Psi=\Psi 1 \otimes \Psi 2$

For images, the eigen matrix of auto-correlation matrix r can be obtained using the separable property of auto correlation matrix R. in which we separate the $n^3 \times n^3$matrix into three $n \times n$ matrix and then find the eigen matrix of each. After that by taking the kronecker product of these eigen matrix we get the eigen matrix of auto correlation matrix R.

Advantages of separability

Reduce the computational complexity from $o(n^6)$to $o(n^3)$

Recall that an $n \times n$ eigen value problem requires $o(n^{3)}$ computations

PROPERTIES OF THE KL TRANSFORM

1. Decorrelation:

The kl transform coefficients are uncorrelated and have zero mean, i.e.,

$e[v(k,1)]=0$ for all k,1

2. It minimize the mse for any truncated series expansion Error vanishes in case there is no truncation.

3. 3. Among all unitary transformations, KL packs the maximum average energy in the first few samples of v.

DRAWBACKS

a) Unlike other transforms, the KL is image dependent.

b) It is computationally very intensive.

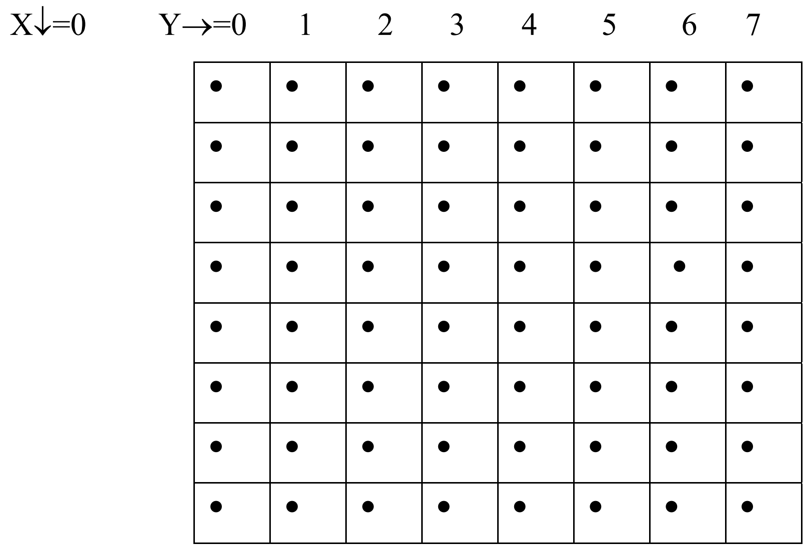
2.3 **DISCREATE COSINE TRANSFORM (DCT)**

The discrete cosine transform (DCT) separates the image into spectral sub-bands of differing frequency. The DCT is similar to the discrete Fourier transform: it transforms a signal or image from the time domain to the frequency domain. DCT, in image compression can be defined as a block of 8x8 array, or 64 pixels. In general, most of the importance of signal lies at low frequencies; these appear in the upper left corner of the DCT. Calculated using following expression

$$F(I,J)=1/4\,C(I)C(J)\sum_{I=0}^{7}\sum_{J=0}^{7} P(X,Y)\cos(2X+1)I\pi/16\,\cos(2Y+1)J\pi/16$$

WHERE $C(I)=C(J)=1/\sqrt{2}$ FOR I,J=0

= 1 FOR OTHER VALUES OF I,J



The square tiles are representing the pixels and the center of each represents the location of the sample that carries the value of pixel intensity. In DCT we apply horizontal analysis followed by vertical analysis. We can form a block of 64 intensity values into 64 coefficients. As wee move right the coefficients represent higher horizontal frequency, and as wee move down the coefficients represent higher vertical frequency.

The inverse DCT is given by following expression

$$P(X,Y)=1/4 \sum_{I=0}^{7} \sum_{J=0}^{7} C(I)C(J)F(I,J)\mathrm{Cos}(2X+1)I\pi/16\ \mathrm{Cos}(2Y+1)J\pi/16$$

Where $C(I)=C(J)=1/\sqrt{2}$ for I,J=0
=1 for all other values of I,J

Disadvantages

It creates blocking artifacts at higher compression ratios.

2.4 **WAVELET TRASFORM**

The wavelet transform (WT) is multiresolution description of an image: the image decoding can be processed sequentially from a very low resolution, corresponding to a very compact code, to the highest resolution. Further more wavelet transform offers perfect reconstruction.

The wavelet transform is closer to the human visual system than DCT. The artifacts introduced by WT coding with high compression ratio and adequate perceptual quantization are less annoying than those introduced at the same bit rate by the DCT.

The WT of an image generates a data structure known as scale-space representation. In this representation, the high frequency signals are precisely located pixel domain, while the low frequency signals are precisely located in frequency domain. The spatial resolution of WT increases linearly with frequency while that of DCT is constant. Sharp edges, which are well localized spatially and have significantly very high frequency component, can be represented more compactly in WT than with the DCT.

Advantages of wavelet based compression

1. Wavelet coding schemes at higher compression avoid blocking artifacts.
2. Wavelet-based coding is more robust under transmission and decoding errors, and also facilitates progressive transmission of images.
3. They are better matched to the HVS (Human Visual System**)** characteristics.
4. Compression with wavelets is scalable as the transform process can be applied to an image as many times as wanted and hence very high compression ratios can be achieved.

5. They provide an efficient decomposition of signals prior to compression.
6. Wavelet based compression allow parametric gain control for image softening and sharpening.
7. Wavelet compression is very efficient at low bit rates.

Disadvantages of wavelet based compression

Wavelet compression does require more computational power than compression based on other techniques such as Discrete Cosine Transform (DCT).

Practical uses of wavelet transforms

- Progressive image compression (useful for low bit rate)
- ECG (electrical activity of the heart, electrocardiograph)
- EEG (electrical activity of the brain, electroencephalograph)
- EMG (electrical activity of the muscles, electromyogram)

2.5 Performance Metrics

The performance of Compression achieved is measured by two significant ratios. These ratios serve as an important variable for comparison of various techniques.

1) Mean Square Error (MSE)

It is the Cumulative squared error between the compressed and the original image.

$$MSE = 1/MN \sum_{Y=1}^{M} \sum_{X=1}^{N} [I(x, y) - I'(x, y)]^2$$

I (x, y) = original image,

I' (x, y)= approximation of decompressed image

M, N = dimensions of the images

MSE **lower** the better, means lesser error.

2**)** Peak Signal to Noise Ratio (PSNR)

It is a measure of the peak error.

PSNR = 20 * log10 (255 / sqrt (MSE))

PSNR higher the better means that the ratio of Signal to Noise is higher**.**

'Signal' is the original image,

'Noise' is the error in reconstruction.

CHAPTER 3

WAVELET THEORY

3.1 **Wavelets**

Wavelets (small waves) are functions defined over a finite interval and having an average value of zero see fig 3.1. The basic idea of the wavelet transform is to represent any arbitrary function *f* (t) as a superposition of a set of such wavelets or basis functions. These basis functions are obtained from a single wave, by dilations or contractions (scaling) see figure 3.2 & 3.3and translations (shifts). For pattern recognition continuous wavelet transform is better.

Consider a real or complex-valued continuos time function ψ(t) with following properties.

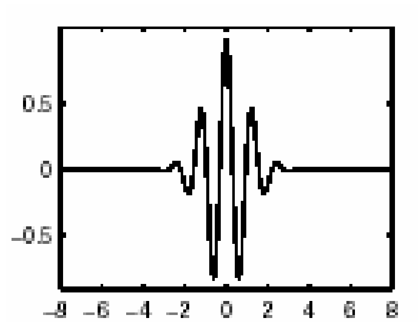
1. The function integrates to zero:

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

2. It is square integrable or , equivalently have a finite energy:

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

Fig. 3.1 Following figure shows Morlet wavelet ψ(t):



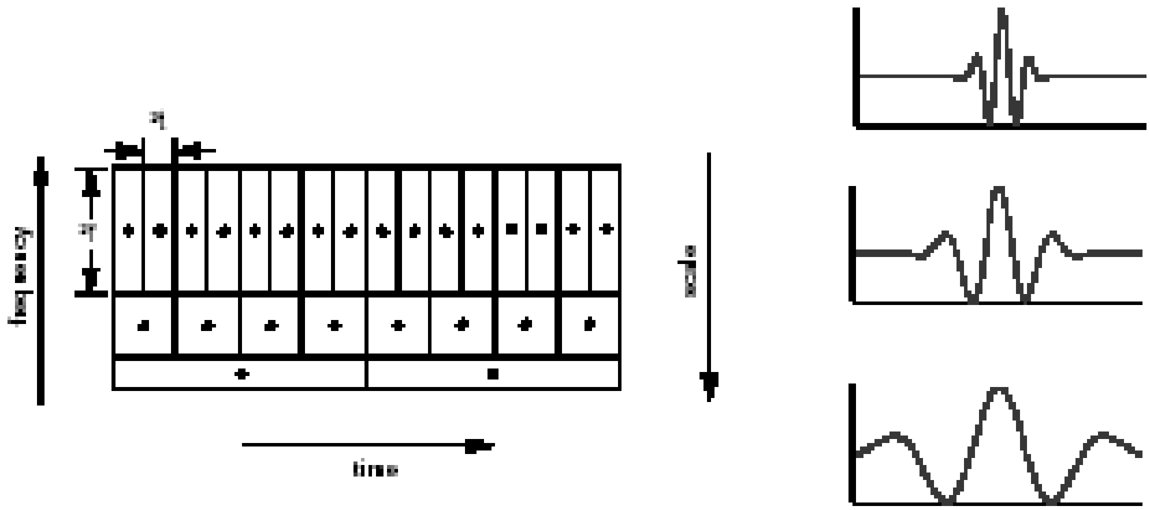


Fig 3.2 Showing scaling of wavelet Bases at three levels

Fig 3.3 showing shifting of wavelet

3.2 **Continuos Wavelet Transform**

The continuous wavelet transform was developed as an alternative approach to the short time Fourier transform to overcome the resolution problem. The wavelet analysis is done in a similar way to the STFT analysis, in the sense that the signal is f(t) multiplied with a function $\psi(t)$ {it the wavelet}, similar to the window function in the STFT, and the transform is computed separately for different segments of the time-domain signal. However, there are two main differences between the STFT and the CWT:

1. The Fourier transforms of the windowed signals are not taken, and therefore single peak will be seen corresponding to a sinusoid, i.e., negative frequencies are not computed.
2. The width of the window is changed as the transform is computed for every single spectral component, which is probably the most significant characteristic of the wavelet transform.

The continuous wavelet transform is defined as follows:

$$W(a, b) \equiv \int_{-\infty}^{\infty} f(t)\ 1/\sqrt{|a|}\ \psi^*((t-b)/a)\, dt$$

The wavelet $\psi(t)$ must satisfy the admissibility condition for inverse wavelet transform

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

The wavelets are functions generated from one single function $\psi(t)$ by dilations and translations :

$$\psi_{a,b}(t) = |a|^{1/2}\ \psi((t-b)/a)$$

Where a and b are dilation and translation parameters respectively.

3.3 **Multiresolution:**

The multiresolution property is inherent in the wavelet transform. The Wavelet transform decomposes a given image into a coarse approximation and details. In Multilevel decomposition, this approximation is decomposed further into lower resolution sub images and detail sub images .The simultaneous appearance of multiple scales is known as multiresolution.

The goal of multiresolution in continuos time is decomposition of the whole function space into subspaces. By decomposing a function f(t) , a piece wise of f(t) is present in each subspace .The signal is resolved at scales $\Delta t = 1, ½ , \ldots\ldots(½)^j$. For audio signals these scales are essentially octaves. They represent higher and higher frequencies. The simultaneous appearance of multiple scales is known as **multiresolution.**

Multiresolution is described for the scaling spaces V_j and the wavelet space W_j. The wavelet space W_j is difference between V_j and V_{j+1}. The sum of V_j and W_j is V_{j-1} .

Scaling spaces:

The scaling space V_j are decreasing. Each V_j is contend in the next subspace V_{j-1}. A function in one subspace is in all the higher (finer) subspaces.

$$V_\infty \ldots\ldots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots\ldots V_{-j} \subset V_{-j-1} \subset \ldots\ldots$$

Detail Subspace:

W_j contains a new information $\Delta f_j (t) = f_{j-1} (t)- f_j (t)$. This is detail at level j. The space W_j are differences between the V_j .

$$V_j \oplus W_j = V_{j-1}$$

The reconstruction of f(t) from its details Δf_j can start at $j =\infty$.

$$f(t) = \sum_{-\infty}^{\infty} \Delta f_j (t) .$$

3.4 **Quadrature Mirror filter Bank**

QMF bank applied to two band system, the analysis filters are linear phase filters and exact reconstruction is represented by the equation:

$$| H_l(e^{jw}) |^2 - |H_u(e^{jw})|^2 = 2$$

where $H_l(e^{jw})$ is low pass frequency response and is the $H_u(e^{jw})$ high frequency response . The response of $H_u(e^{jw})$is mirror image of $H_l(e^{jw})$ with respect to π/2, which is quarter of sampling frequency . The name Quadrature Mirror filter is derived from this fact. Frequency response of $H_l(e^{jw})$ and $H_u(e^{jw})$is overlapping so that no frequency range is left out. To avoid distortion at π/2 , there should be 3 dB attenuation at π/2.See fig 3.4.

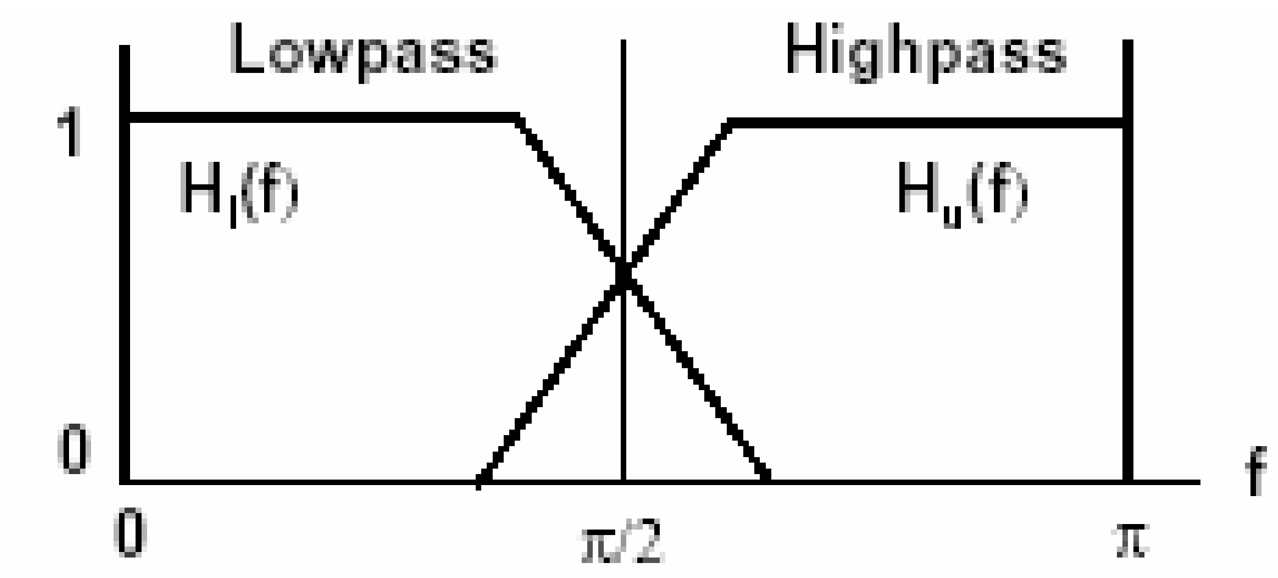


Fig3.4 Typical Two Band Magnitude Response

3.5 **The Discrete Wavelet Transform**

Although the discretized continuous wavelet transform enables the computation of the continuous wavelet transform by computers, it is not a true discrete transform. As a matter of fact, the wavelet series is simply a sampled version of the CWT, and the information it provides is highly redundant as far as the reconstruction of the signal is concerned. This redundancy, on the other hand, requires a significant amount of computation time and resources. The discrete wavelet transform (DWT), on the other hand, provides sufficient information both for analysis and synthesis of the original signal, with a significant reduction in the computation time.

The DWT is considerably easier to implement when compared to the CWT. The basic concepts of the DWT will be introduced in this section

The continuous wavelet transform was computed by changing the scale of the analysis window, shifting the window in time, multiplying by the signal, and integrating over all times. In the discrete case, filters of different cutoff frequencies are used to analyze the signal at different scales. The signal is passed through a series of high pass filters to analyze the high frequencies, and it is passed through a series of low pass filters to analyze the low frequencies

The resolution of the signal, which is a measure of the amount of detail information in the signal, is changed by the filtering operations, and the scale is changed by upsampling and downsampling (subsampling) operations. Subsampling a signal corresponds to reducing the sampling rate, or removing some of the samples of the signal. For example, subsampling by two refers to dropping every other sample of the signal. Subsampling by a factor n reduces the number of samples in the signal n times.

Upsampling a signal corresponds to increasing the sampling rate of a signal by adding new samples to the signal. For example, upsampling by two refers to adding a new sample, usually a zero or an interpolated value, between every two samples of the signal. Upsampling a signal by a factor of *n* increases the number of samples in the signal by a factor of *n*.

The procedure starts with passing this signal (sequence) through a half band digital lowpass filter with impulse response h[n]. Filtering a signal corresponds to the mathematical operation of convolution of the signal with the impulse response of the filter. The convolution operation in discrete time is defined as follows

$$x[n]*h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$

A half band lowpass filter removes all frequencies that are above half of the highest frequency in the signal. For example, if a signal has a maximum of 1000 Hz component, then half band lowpass filtering removes all the frequencies above 500 Hz. The unit of frequency is of particular importance at this time. In discrete signals, frequency is expressed in terms of radians. Accordingly, the sampling frequency of the signal is equal to 2p radians in terms of radial frequency. Therefore, the highest frequency component that exists in a signal will be p radians, if the signal is sampled at Nyquist's rate (which is twice the maximum frequency that exists in the signal); that is, the Nyquist's rate corresponds to p rad/s in the discrete frequency domain. Therefore using Hz is not appropriate for discrete signals. However, Hz is used whenever it is needed to clarify a discussion, since it is very common to think of frequency in terms of Hz. It should always be remembered that the unit of frequency for discrete time signals is radians.

` After passing the signal through a half band lowpass filter, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of p/2 radians instead of p radians. Simply discarding every other sample will **subsample** the signal by two, and the signal will then have half the number of points. The scale of the signal is now doubled. Note that the lowpass filtering removes the high frequency information, but leaves the scale unchanged. Only the subsampling process changes the scale. Resolution, on the other hand, is related to the amount of information in the signal, and therefore, it is affected by the filtering operations. Half band lowpass filtering removes half of the frequencies, which can be interpreted as losing half of the information. Therefore, the resolution is halved after the filtering operation. Note, however, the subsampling operation after filtering does not affect the resolution, since removing half of the spectral components from the signal makes half the number of samples redundant anyway. Half the samples can be discarded without any loss of information. In summary, the lowpass filtering halves the resolution, but leaves the scale unchanged. The signal is then subsampled by 2 since half of the number of samples are redundant. This doubles the scale.

This procedure can mathematically be expressed as

$$y[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[2n-k]$$

Having said that, we now look how the DWT is actually computed: The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information. DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with low pass and highpass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive highpass and lowpass filtering of the time domain signal. The original signal x[n] is first passed through a halfband highpass filter g[n] and a lowpass filter h[n]. After the filtering, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of p /2 radians instead of p. The signal can therefore be subsampled by 2, simply by discarding every other sample. This constitutes one level of decomposition and can mathematically be expressed as follows:

$$y_{high}[k] = \sum_{n} x[n] \cdot g[2k-n]$$

$$y_{low}[k] = \sum_{n} x[n] \cdot h[2k-n]$$

Where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the highpass and lowpass filters, respectively, after subsampling by 2.

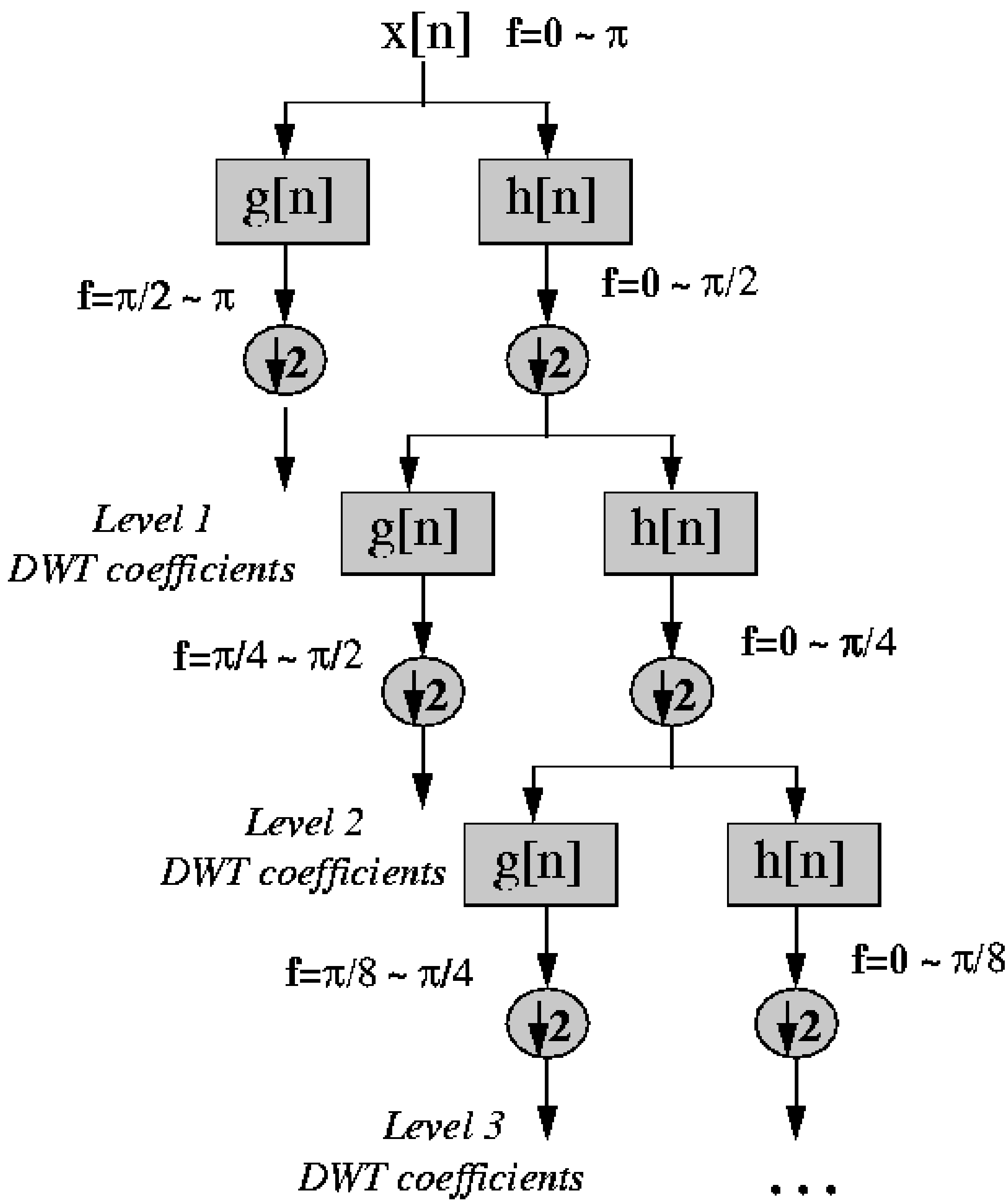


Fig 3.5 Showing Discrete Wavelet transform at three levels

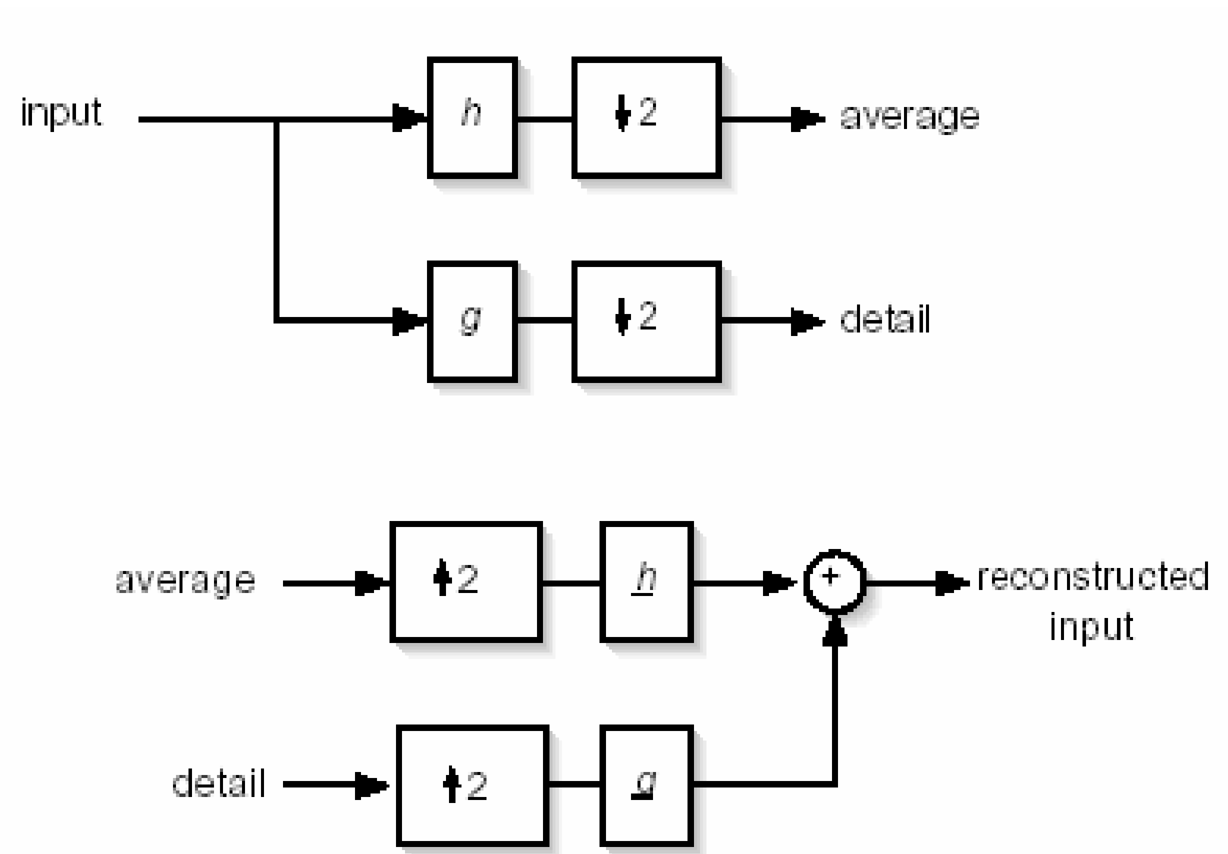
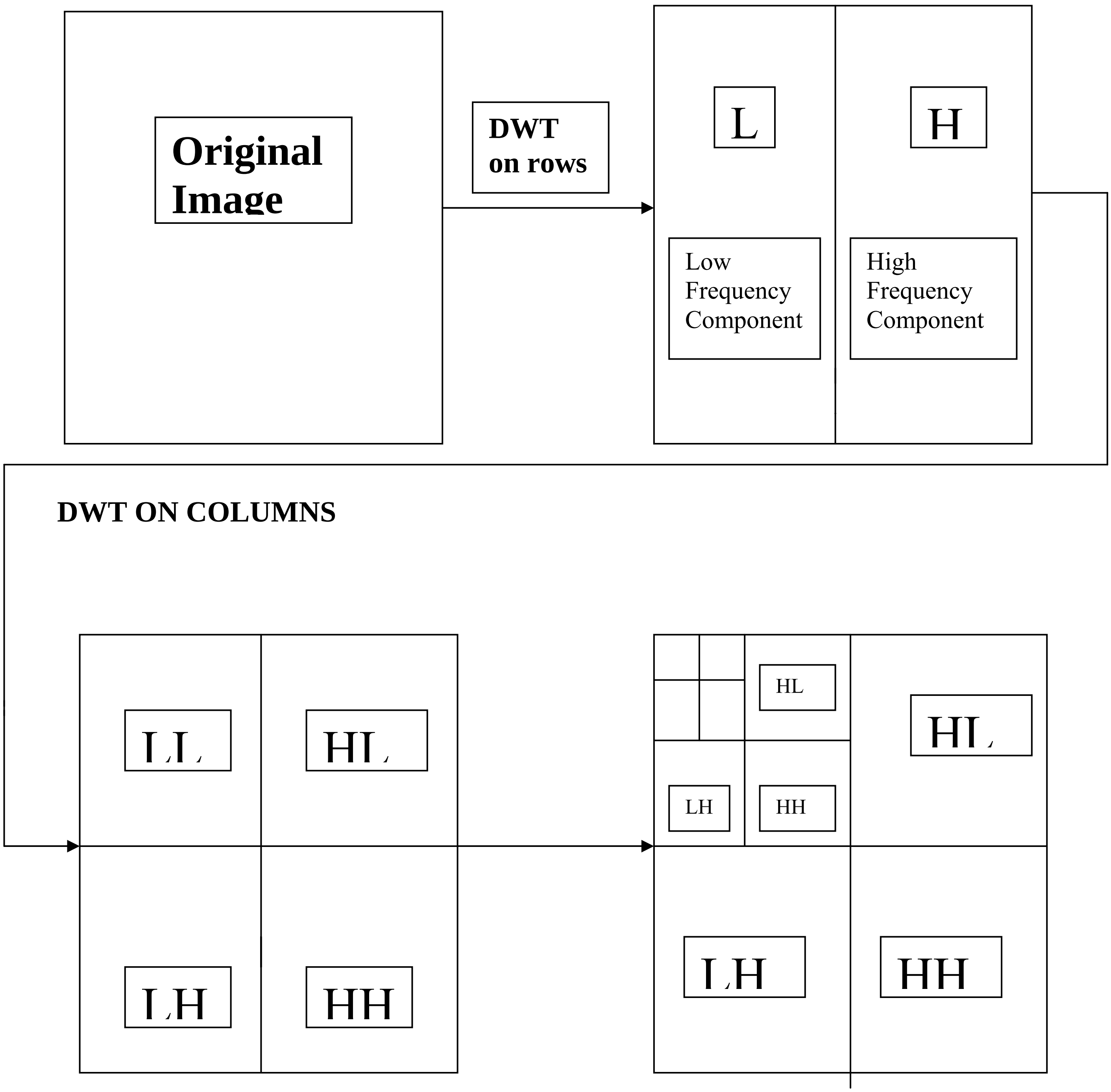


Fig 3.6 A general two band structure



Original Image
DWT on rows
L
H
Low Frequency Component
High Frequency Component
DWT ON COLUMNS
LL
HL
LH
HH
HL
LH
HH
HL
LH
HH

Fig 3.7 Showing Image decomposition into 3levels

3.6 Wavelet families

Haar

Any discussion of wavelets begins with Haar wavelet, the first and simplest.Haar wavelet is discontinuous, and resembles a step function. It represents the same wavelet as Daubechies db1.

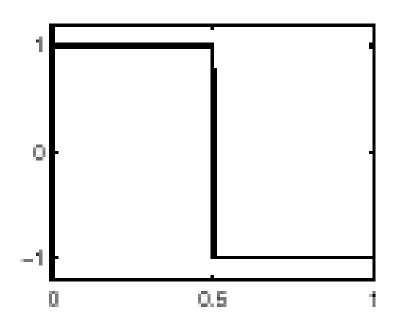


Fig. 3.8 Haar Wavelet.

Daubechies

Ingrid Daubechies, one of the brightest stars in the world of wavelet research, invented what are called compactly supported orthonormal wavelets — thus making discrete wavelet analysis practicable. The names of the Daubechies family wavelets are written dbN, where N is the order, and db the "surname" of the wavelet. The db1 wavelet, as mentioned above, is the same as Haar wavelet. Here are the wavelet functions psi of the next nine members of the family:

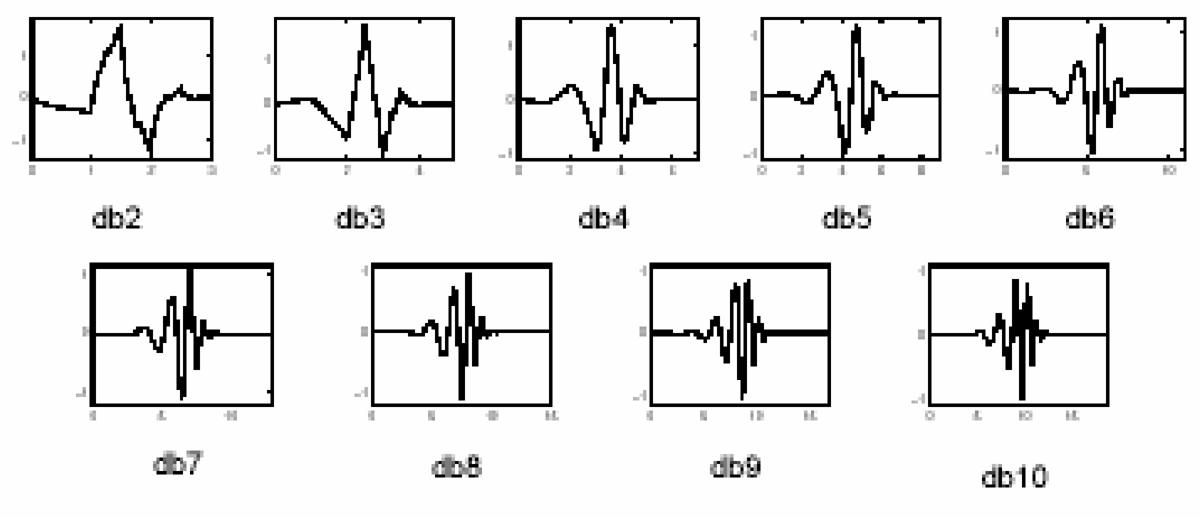


Fig3.9 Daubechies family wavelets.

Biorthogonal

This family of wavelets exhibits the property of linear phase, which is needed for signal and image reconstruction. By using two wavelets, one for decomposition (on the left side) and the other for reconstruction (on the right side) instead of the same single one, interesting properties are derived.

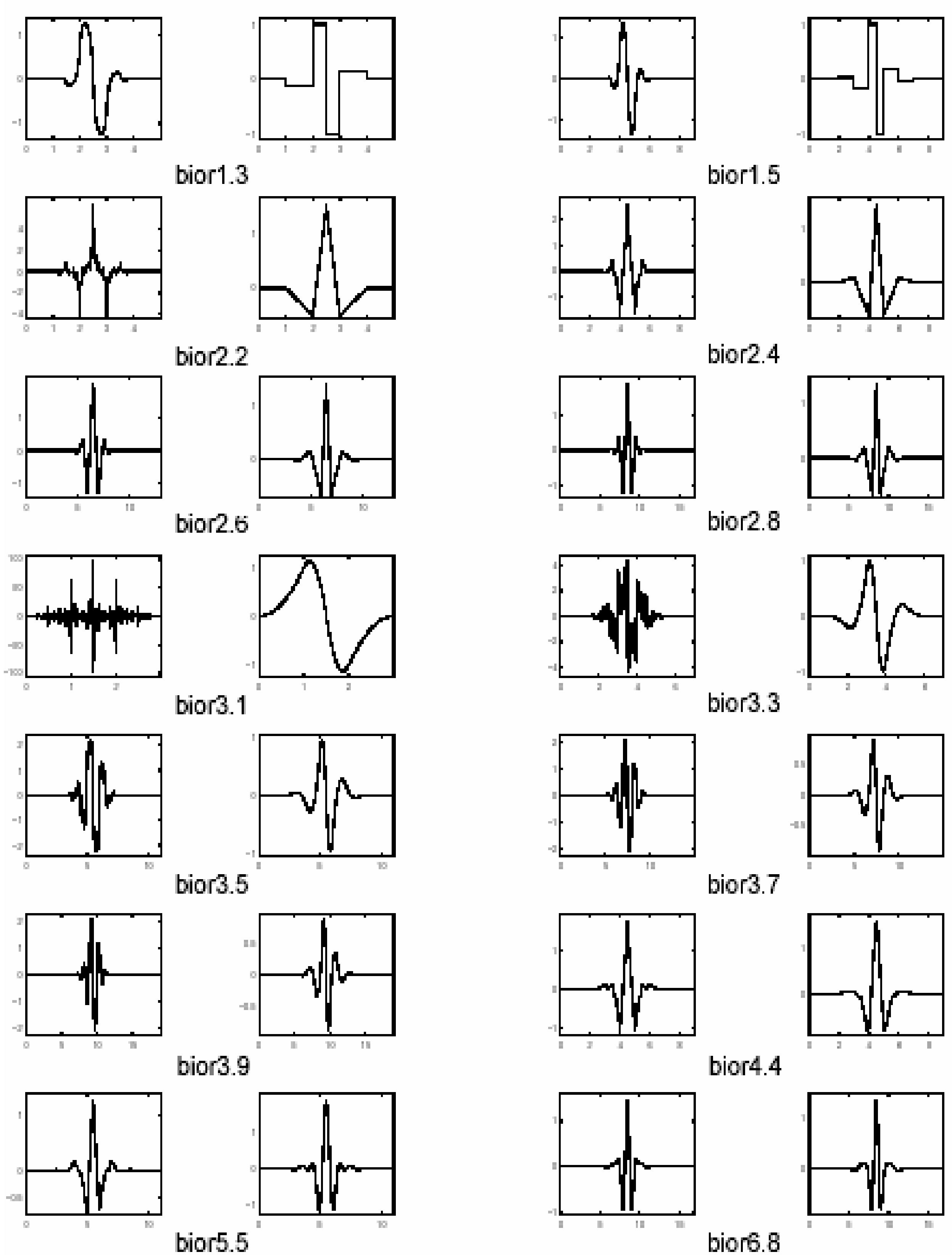


Fig 3.10 Biorthogonal Wavelet Families

Coiflets

Built by I. Daubechies at the request of R. Coifman. The wavelet function has $2N$ moments equal to 0 and the scaling function has $2N$-1 moments equal to 0. The two functions have a support of length $6N$-1.

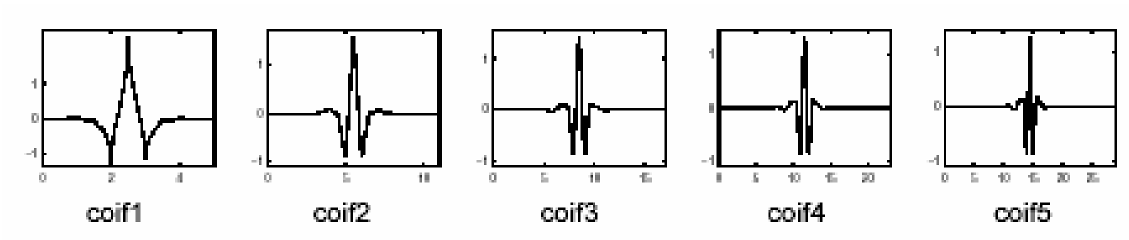


Fig 3.11 Coiflets Wavelets.

Symlets

The symlets are nearly symmetrical wavelets proposed by Daubechies as modifications to the db family. The properties of the two wavelet families are similar. Here are the wavelet functions psi.

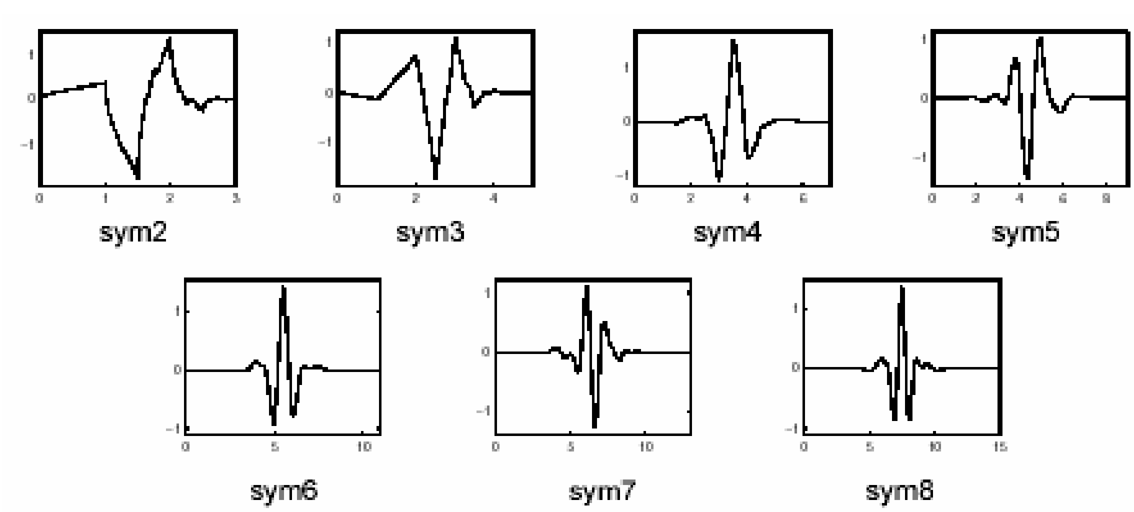


Fig 3.12 Symlets

3.7 Literature Survey

There has been a particular interest in the past decade in the field of image compression. A lot of work has been reported on image compression technique using wavelet transform coding.

We first review the history of wavelet theory briefly. It was in 1807 Josef Fourier came up with a revolutionary idea that stirred pundits of signal analysis. The idea expressing the function as a weighted sum of sinusoids was a paradigm shift from the existing framework of signal analysis and it opened new vistas leading to evaluation of harmonic analysis. Wavelets, on the other hand, provide an elegant alternative to sinusoidal representation of local details of signal . Although techniques like Gabor Transform (1940) or its generalization in form of Short Fourier transform (STFT) can also be used to analysis of local details, wavelets attract attention of researchers because of their inherent simplicity and flexibility. Development of wavelet is result of the merging of ideas from different fields. Actual consolidation of wavelet theory began in late 1970's when J. Morlet ,a geophysical engineer , proposed an alternative to shot time Fourier transform by using a narrow time window for low frequency analysis Daubechies formalized the construction of compactly support orthonormal wavelet bases .

Subband image coding has recently been shown to be an effective technique for high quality coding. Mark J. Smith proposed analysis/synthesis techniques for Subband coding based on FIR Quardrature Mirror Filters (QMF's). The Subband coding system may be viewed as having two basic components: the Subband analysis/synthesis subsystem which is composed of filter banks; and the coding system may employ some form of quantization and entropy coding. Vetterli treated 2D subbands analysis/synthesis system using both separable and non-separable filter banks. The most computationally efficient approach is splitting and merging Subband images, which results from using separable filters. With separable filters the 2D filtering can be implemented as a set of 1D filtering operations. Filtering is performed on each row and then on each column of image.

A new theory introduced by Mark Antonini [3] was for analyzing image compression methods that are based on compression of wavelet decompositions. It was shown that if picture can be characterized by their membership in smoothness then wavelet-based methods are near optimal within the larger class of stable, transform based nonlinear methods of image compression.

Bryan E. Usevitch [1] explained Subband and wavelet coding as currently used in image processing standard. He gave the mathematical background and discussed practical implications. The article leads naturally to the standard that is originated from these ideas. He also explained the use of biorthogonal wave filters rather than using orthogonal filters in the modern wavelet coders; since wavelet transform can use essentially a infinite number of possible linear phase biorthogonal filters where there is only one linear phase orthogonal filter that is Haar filter.

JPEG 2000 is covered by A.Skodras, C.Christopoulos and T. Ebrahimi [2]. It is a comparative review of the standard, which was recently introduced and includes wavelet decomposition as a key ingredient. Beyond just compression performance, the article points out several features that are enabled by wavelet decomposition. The JPEG committee has recently released its new image coding standard JPEG 2000, which serve as a supplement for the original JPEG standard introduced in 1992. Rather than incrementally improving on the original standard, JPEG 2000 implements an entirely new way of compressing images based on wavelet transform, in contrast to Discrete Cosine Transform (DCT) used in the original JPEG standard. The state of wavelet based coding has improved significantly since the introduction of the original JPEG standard.

A noble breakthrough was introduction of EZW coding algorithm, which is designed by J. Shapiro [9]. The EZW coding algorithm is able to exploit the multi-resolution property of the wavelet transform to give a computationally simple algorithm with outstanding performance relative to block transform coders. As the result, wavelet based coding has been adopted as the underlying method to implement the JPEG 2000 standard.

CHAPTER 4

BIOTHOGONALWAVELET FILTER & PR

4.1 WHY BIORTHGONAL WAVELETS?

The use of symmetric extension and linear phase wavelets filter would seem to solve the problem of border effects in the wavelet transform. However, there is still one technical difficulty to overcome, which is illustrated by the following:

Fact : for real valued , compactly supported orthogonal wavelets, there is only one set of linear phase filter, and that set is the trivial haar filters , h=(1,1), g=(1,-1)

Biorthogonal wavelets would allow for large number of linear phase filters. As the name implies, biorthogonal wavelets have same orthogonality relationship between there filter.

Biorthogonal wavelet differs from orthogonal in that the forward wavelet transform is equivalent to projecting the input signal on to non orthogonal basis function. The orthogonal and biorthogonal wavelets transform are analogous to orthogonal and non singular matrix transforms respectively. Both the orthogonal and nonsingular matrix transforms are invertible, but only the orthogonal matrix transform is energy preserving.

The main advantage in using the biorthogonal wavelets transform is that it permits the use of a much broader class of filters and this class includes symmetric filters. But the disadvantage that it is not energy preserving. The fact that biorthogonal wavelets are not energy preserving does not turn out to be a big problem, since there are linear phase biorthogonal filter coefficient which are “close” to being orthogonal.

4.2 Multirate filter

A Multirate filter is one which splits signal into multiple parts. One of the most important multirate filters is the one which splits a signal into two equal halves where one half has the lower frequency components. This is fundamental to the subband concept and leads to wavelets, multiresolution, and most of the other applications. Of course it is obvious that if we can split the band into two halves efficiently, we can continue doing so to any resolution we desire.

4.3 Two-channel filter bank

The two-channel filter bank is shown in Figure 4.1, where H_0**(z)** is the low-pass analysis filter H_1 (z) is the high – pass analysis filter. As the output signals $\upsilon_0(n)$ and $\upsilon_1(n)$ have half the bandwidth of the original input signal, we can use decimation by 2 and still lose no information. The synthesis filter bank is also shown in Figure 4..1 and consist of two up-samplers and the low-pass synthesis filter $G_0(z)$ and the high-pass one $G_1(z)$.

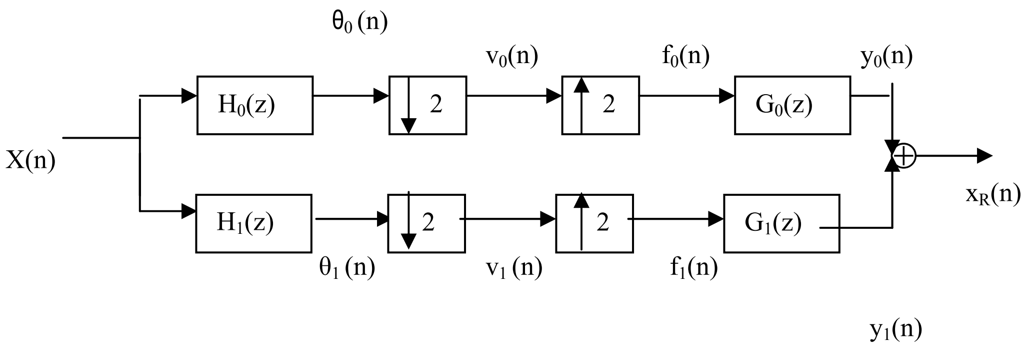


Figure 4.1 Two channel filter-bank structure showing analysis and synthesis stages.

$$\theta_0(z) = H_0(z) \;.\; X(z)$$
$$Y_0(z) = G_0(z) \;.\; F_0(z)$$
$$V_0(z) = \tfrac{1}{2}\,[\theta_0(z^{1/2}) + \theta_0(-z^{1/2})] \qquad (4..1)$$
$$F_0(z) = V_0(z^2)$$

Substituting for $F_0(z)$ and V_0 (z), we obtain

$$Y_0(z) = \tfrac{1}{2}\, G_0(z)\, [H_0(Z)\, X(z) + H_0(-z)\, X(-z)] \qquad (4..2)$$

And

$$Y_1(z) = \tfrac{1}{2}\, G_1(z)\, [H_1(z)\, X(z) + H_1(-z)\, X(-z)] \qquad (4..3)$$

Finally, we obtain for the output X (z),

$$X(z) = \tfrac{1}{2}\, X(z)\, [H_0(z)\, G_0(z) + H_1(z)\, G_1(z)]$$

$$+ \ \tfrac{1}{2} \ X(-z) \ [H_0 (-z) \, G_0(z) \ + \ H_1(-z) \, G_1(z)] \qquad (4..4)$$

Equation 4..4 can be simplified to

$$X(z) \ = \ T(z)X(z) \ + \ S(z) \, X(-z) \qquad (4..5)$$

Where we have defined

$$T(z) \ = \ \tfrac{1}{2} \ [H_0(z)G_0(z) \ + \ H_1(z)G_1(z)]$$

$$S(z) \ = \ \tfrac{1}{2} \ [H_0(-z)G_0(z) \ + \ H_1(-z)G_1(z) \,] \qquad (4..6)$$

In order to have perfect reconstruction (PR) at the synthesis, we must impose that

$$X(z) \ = \ c \ X(z) \ z^{-no} \qquad (4..7)$$

Where c is a constant and n_o is a fixed delay.

Thus, the conditions for PR are

$$S(z) \ = \ 0 \ = \ H_0 (-z) \, G_0(z) \ + \ H_1 (-z) \, G_1(z) \qquad (4..8)$$

and

$$T(z) \ = \ H_0(z)G_0(z) \ + \ H_1(z) \, G_1(z) \ = \ cz^{-n}{}_0 \qquad (4..9)$$

From the above equations, we obtain

$$G_0(z)/G_1(z) \ = \ -H_1(-z)/H_0(z) \qquad (4..10)$$

Before we find the PR solution, it is of interest to note that T(z) represents distortion and S(z) represents aliasing.

4.4 Biorthogonal Filters

To avoid distortion, we must satisfy Eq4.9

$$T(z) = cz^{-n_0}$$

$$H_0(z)G_0(z) + H_1(z)G_1(z) = cz^{-n_0}$$

To avoid aliasing errors, we must impose that :

$$S(z) = 0 \qquad 4.11$$

$$H_0(-z)G_0(z) + H_1(-z)G_1(z)=0 \qquad 4.12$$

The orthogonal case can be shown to give solutions which can never be linear phase. However that is not true in biorthogonal case.

For this let us choose

$$G_0(z) = H_1(-z)$$

$$G_1(z)= -H_0(-z)$$

Which are the same conditions we imposed for orthogonal case except of sign changes. For this case T(z) becomes

$$T(z)= G_0(z)\ H_0(z) - G_0(-z)\ H_0(-z)=P_0(z)- P_0(-z) \qquad 4.13$$

Where $P_0(z)= G_0(z)\ H_0(z)$

And $P_0(-z)= G_0(-z)\ H_0(-z)$

PR condition is given by

$$P_0(z)- P_0(-z)=z^{-l}$$

If we define $P(z)= z^{l}\ P_0(z)$, then PR condition becomes

$$P(z)+P(-z)=2$$

This is the same condition for half band filter.

CHAPTER 5

Analysis of Biorthogonal Wavelet Filter and proposed Modification

5.1 Introduction

The biorthogonal wavelet transform is not an energy preserving transform. As a result, the MSF changes made in the wavelet coefficient will cause weighted MSE changes in the reconstructed output . Certain applications require knowledge of this weighted relationship between wavelet coefficient and reconstructed output. One such application is weighted based coding, which perform bit application and quantization in the wavelet domain, but measure performance based on the reconstructed output.

This correspondence details a method for computing the weights for both 1-dimensional and 2-diomensional biorthogonal wavelet transform. The method involves only interpolating filtering, and taking the norms of wavelet filter coefficient , example weights are computed for two biorthogonal wavelet sets to illustrate the method. Weight Computation

5.2 1D Weight Computation

To introduce the biorthogonal weighting problem, first consider the 2-level biorthogonal wavelet transform shown in fig.-1 . In this system the analyses and synthesis filters are denoted by h_{ij} and g_{ij} respectively, and the subscript I and j denote the transform level and band respectively. The problem to be solved can be stated as follows. Given the subband variance σ_r^2 of the reconstructed output r(n). this question arises in coding application where the $\sigma_{ij}^{\ 2}$ represent the quantizer error variance, and σ_r^2 represent the mean squared reconstructed error.

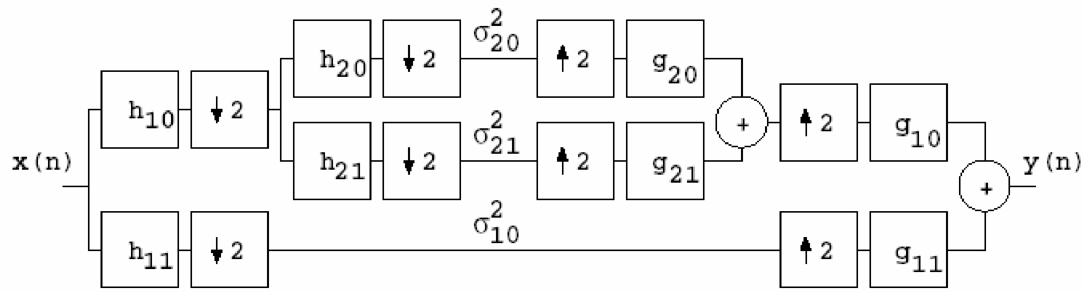


Figure 1: A two level biorthogonal wavelet transform system.

For orthogonal system, the answer to the problem is simple, since input to output energy is conserved. In biorthogonal system, the answer is more difficult because of the weighting introduced by the biorthogonal filters.

To analyze the weighting effect of the biorthogonal filters, consider an input x(n) upsampled by M to give v(n) then filtered by h(n) to give y(n). the input/output relation between x(n) and y(n) is given by following :

$$Y(n) = \sum_{k=-\infty}^{\infty} v(k)h(n-k)$$

$$= \sum_{\substack{k=-\infty \\ k \text{ mode } M=0}}^{\infty} x(k/M)h(n-k)$$

$$= \sum_{t=-\infty}^{\infty} x(l)h(n-Ml) \qquad (1)$$

Assuming x(n) to be a zero mean stationary process, equation (1) can be used to compute the variance of y, $\sigma_y^2 = E[y^2(n)]$. Because of the time varying nature of the interpolation operator, $E[y^2(n)]$ is cyclostationary, with M values corresponding to n mode M. averaging over these M values given the following result

$$E[y^2(n)] = 1/M \sum h^2(n) \quad (2)$$

Reconstruction from multiple levels of transform requires the coefficients to pass through multiple stages of interpolation and filtering. To characterize the effect of these cascaded interpolation and filtering blocks, consider an input x(n) upsampled by 2 and filter by h(n) to give y(n), then upsampled by 2 and filtered by g(n) to give z(n). Using 1, the output z (n) can written in terms of the input x(n)and Filters h and g as follows:

$$Z(n) = \sum_{k=-\infty}^{\infty} y(k)g(n-2k)$$

$$= \sum_{k=-\infty}^{\infty} \left(\sum_{j=-\infty}^{\infty} x(j)h(k-2j)\right) g(n-2k)$$

$$= \sum_{j=-\infty}^{\infty} x(j) \sum_{t=-\infty}^{\infty} h(l)g(n-4j-2l) \quad (3)$$

Where the last step comes by substituting l=k-2j. Careful inspection of equation (3), while using (1) Shows that the two level cascade of filter blocks is equivalent to interpolating x (n) by 4 and filtering with the following equivalent filter

$$F(n) = \sum_{l=-\infty}^{\infty} h(l)g(n-2l) \quad (4)$$

Generalizing the previous results shows that processing an input with L stages of interpolation by and filtering blocks is equivalent to interpolating by 2^L and filtering by an equivalent filter. Equation (4) shows that this equivalent filter is computed by interpolation and convolving the filters in the chain of interpolator /filter blocks. For examples, referring to figure 1 and assuming 3 levels of reconstruction from the lowest frequency subband, the equivalent filter is given as

$$G_{eq} = \text{interp2 (interp2 (g30)* g20)* g10}$$

Where interp2 denotes the interpolation by 2 operator and * denotes convolution
Using the previous results, the total reconstruction error from all the sub band,
for an L level wavelet recomposition is

$$\sigma_r^2=1/2^L W_{LO}\sigma_{LO}{}^2+\sum_{i=1}^{L}1/2^i W_{ij}\sigma_{i1}{}^2 \qquad (5)$$

Where σ_{ij}^2 represents the variance in subband ij and W_{ij} represents the weighting introduced by wavelet recomposition. From equation(2) the weights can be computed as

$W_{ij}=\sum_n g_{eq,ij}{}^2(n)$, where $g_{eq,ij}{}^2(n)$ is equivalent filter from subband ij to the output.

For orthogonal filters, all the W_{ij} equal to one, and the error energy in the output equals the error energy in the subtends. As the weights differ from one, the error energy differs according to Equation (5). Thus, the weights can be used as a measure of the deviation of biorthogonal wavelets from being orthogonal. Table 1 shows an example of the 1-dimentional weights computed for two examples biorthogonal filter sets (3,4).The table includes all possible weights for up to four levels of the re-composition and assumes the same filters at each level. These weight values show that the 9/7 biorthogonal wavelet filters are reasonable close to being orthogonal.

5.3 2D Weight Computation

Deriving the weighting for the 2-Dimensional case from the 1-dimensional case is straight-forward. This paper will only illustrate the derivation for the more common separable filter case. Using the subband weight structure shown in figure 2, the two-dimensional reconstruction error formulae, equivalent to Equation (5) , can be shown to be

$$\sigma_r^2=1/4^L S_{LO}\sigma_{LO}{}^2+\sum_{j=1}^{3}\sum_{i=1}^{L}1/4^i s_{ij}\sigma_{ij}{}^2$$

Note in particular that subband index j is valid in the {0,…,3}for the 2-dimensional case. The weights s_{ij} can be computed as a product of 2 “one-dimensional” weights

$$S_{ij}=t_{ij}^{H}t_{ij}^{V},\ \ j\in\{0,\ldots,3\},i\in\{1,\ldots,L\}$$

Where t_{ij}^{H} represents the weight resulting from all the horizontal filters from subband ij to the output, and t_{ij}^{V} represents all the vertical filters from subband ij to the output. Both t_{ij}^{H} and t_{ij}^{V} are computed exactly the same as in the 1-dimentional case.

As an example, consider computing the weight $\boldsymbol{s_{21}}$**.** In the horizontal direction, the $\boldsymbol{s_{21}}$ is reconstructed through a highpass filter followed by lowpass filter. Thus $\boldsymbol{t_{21}^{H} = w_{21}}$ where $\boldsymbol{w_{21}}$ is the one-dimensional weight. In the vertical direction, the $\boldsymbol{s_{21}}$ passes through two lowpass filters and $\boldsymbol{t_{21}^{V} = w_{20}}$. The resulting weight is $\boldsymbol{s_{21}} = \boldsymbol{t_{21}^{H}\ t_{21}^{V}} = \boldsymbol{w_{21}\ w_{20}}$.

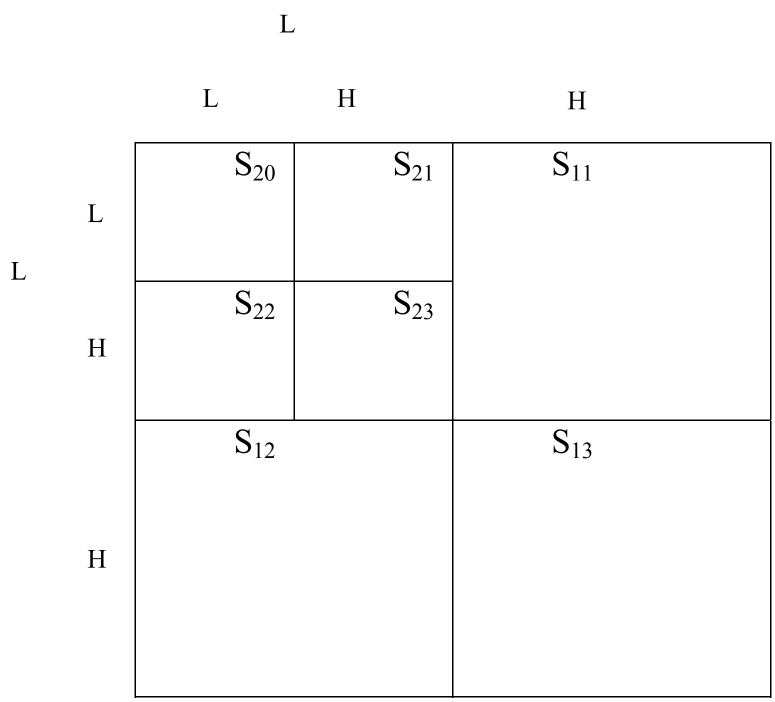


Figure 2;Wieghts representation for two dimensional, two level wavelet transform. The axes also show the lowpass / highpass frequency designation for horizontal and vertical orientations.

Table 1: All possible weights for up to 4 levels of the one-dimensional transform, tabulated for two wavelet coefficient sets shown in the right half of the table.

weight	9/7	5/3	9/7 filter coerf.		5/3 filter coef.		filter index
			ho	*go*	*ho*	90	
W_{10}	0.98295	0.75000	0.852699	0.788486	1.060660	0.707107	0
W_{11}	1.04043	1.43750	0.377402	0.418092	0.353553	0.353553	-1, 1
W_{20}	1.03060	0.68750	*-0.110624*	-0.040689	-0.176777		-2, 2
W_{21}	0.96721	0.92187	-0.023849	-0.064539			-3, 3
W_{30}	1.05209	0.67187	0.037828				-4,4
W_{31}	1.03963	0.79297					
W_{40}	1.05848	0.66797					
W_{41}	1.07512	0.76074					

5.4 Proposed Modification

Tree structured subband coding is an increasing and flexible alternative to other subband coding techniques based on block orthogonal transform ,which exhibits annoying blocky artifacts at low bit rates. The main building block of tree structured subband coder is the two channel subband coder. Given a number of levels of decomposition , L, the corresponding subband transform is a function of the analysis low pass and high pass filters , $H_o(z)$and $H_1(z)$,and the synthesis low pass filters $G_0(z)$and $G_1(z)$(see figure).

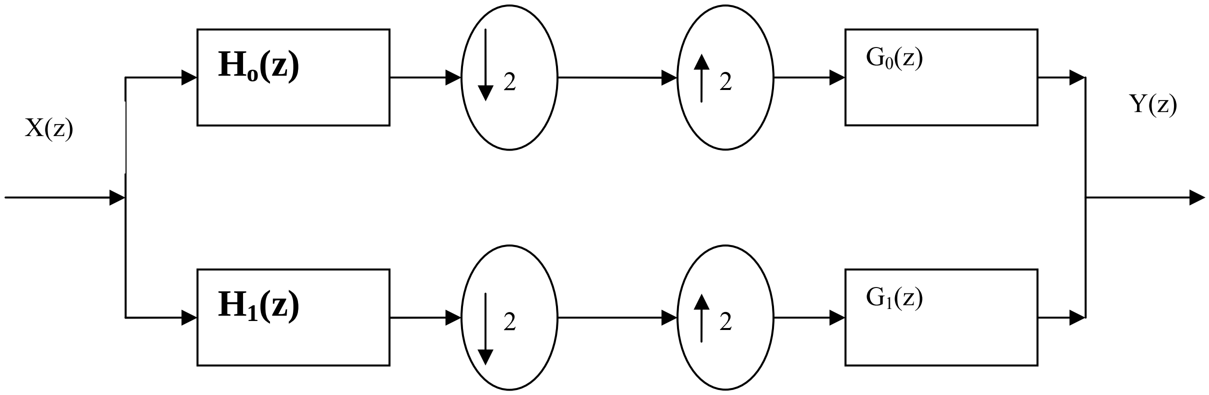


Figure 3 Analysis and Synthesis section for a maximally decimated two band filter bank

The following theorem as well as proof from [ ];it parameterizes all filters $H'_1(z)$ that are complementary to a prototype filter $H_0(z)$.

Y(z)

Theorem: If the length of $|h_0|$ and $|h_1|$ of two complementary filters $H_0(z)$ and $H_1(z)$are odd two satisfy $|h_0|=|h_1|+2$ and if $H_0(z)=0$, then all high pass analysis filters $H'_1(z)$ complementary of $H_0(z)$are of the form

:

$$H'_1(z) = z^{-2m} H_1(z)+E(z^2)\, H_0(z)$$

Where

$$E(z^2)=\sum_{i=1}^{m} \alpha_i \,(z^{-2(i-1)}+z^{-2(2m-i)})$$

The length of $H'_1(z)$ is clearly $|h'_1|=|h_1|+4m$.It some time appropriate to impose zeroes at π for low pass filter ($H_1(e^{-j\pi})= H_0(-1)=0$)zeros at dc frequency for high pass filter ($H_1(e^{=j\pi}) = H_1(1)=0$) . Since

$E(1)=2\sum_{i=1}^{m} \alpha_i$ is nonzero in general, $H_0(-1)=0$ is the only way to ensure that

$H'_1(-1)= H_1(-1)=\sqrt{2}$. Finally the requirement that $H_1(1)=0$(which will find important later) translates into a constraint on coefficients α_i :

$$H'_1(1)= H_1(1)+E(1)\, H_0(1)= H_1(1)+2\, H_0(1) \sum_{i=1}^{m} \alpha_i$$

Therefore to ensure that $H'_1(1)$ has zero at dc frequency, we must have

$$\sum_{i=1}^{m} \alpha_i = -H_1(1)/2\, H_0(1)$$

If $H_1(1)=0$,then we must have $\sum_{i=1}^{m} \alpha_i = 0.$

FIR filter banks will be referred as $|h_0|/|h_1|$. In this thesis I am trying to construct Biorthogonal filter with PR linear phase filter pair in such away that resulting filters are more close to orthogonal by choosing the coefficient of filter such away that weights generated are close to 1(as in orthogonal case).

Consider an arbitrary linear phase low pass filter of length of length 9 $H_0(z)$=[a b c d e d c b a] and$H_1(z)$=[x y z w z y x] satisfying all biorthogonal PR filter conditions .Here m=2,so

$$H'_1(z)=z^{-4}\, H_1(z)+E(z^2)\, H_0(z).$$

Where

$$E(z^2) =\sum_{i=1}^{m} \alpha_i \,(z^{-2(i-1)}+z^{-2(2m-i)}) \quad \& \quad m=2$$

Here $H_1(z)=0$, so $\sum_{i=1}^{2} \alpha_i = 0$, $\alpha_1+\alpha_2=0$ hence

$$E(z^2)= \alpha_1(1- z^{-2}- z^{-4}+ z^{-6})$$

Putting values of $E(z^2)$, $H_0(z)$and $H_1(z)$.

$$H'_1(z)= \alpha_1 a+ \alpha_1 b\, z^{-1}+\alpha_1 (c-a)z^{-2} +\alpha_1 (d-b)z^{-3}+(x+ \alpha_1 (e-c-a))\, z^{-4} +$$
$$(y-\alpha_1 b)\, z^{-5}+(z+\alpha_1(a-e))\, z^{-6}+(w+2\, \alpha_1 (b-d))z^{-7}+(z+\alpha_1 (a-e))z^{-8}+$$
$$(y-b\alpha_1)z^{-9}+ (x+\alpha_1(e-c-a))\, z^{-10}+\alpha_1 (d-b)\, z^{-11}+\alpha_1 (c-a)\, z^{-12}+\alpha_1 b z^{-13}+$$
$$\alpha_1 a z^{-14}$$

Now we can choose the values of a , b, c , d ,e, x, y, z ,w and α_1 such that weights of new 9/15 linear phase biorthogonal filter close to orthogonal.

CHAPTER 6

EMBEDDED ZEROTREE WAVELET ALGORITHM

6.1 EMBEDDED CODING :

An embedded code represents a sequence of binary decisions that distinguish an image from an image from the 'null' or 'all gray' image. Since the embedded code contains all lower rate codes "embedded" at the beginning of the bit stream, effectively, the bits are "ordered in importance". Using an embedded code, an encoder can terminate the encoding at any point thereby allowing a target rate or distortion metric to be met exactly. Some target parameter, such as bit count, is monitored in the encoding process. When the target is met, the encoding stops. Similarly, given a bit stream, the decoder can cease decoding at any point and can produce reconstructions corresponding to all lower-rate encodings.

FEATURES OF THE EMBEDDED CODER:

The EZW algorithm contains the following features:

- A discrete Wavelet transform which provides a compact multiresolution representation of the image.
- Zerotree coding which provides the compact multiresolution representation of significance maps, which are binary maps indicating the position of significant coefficients. Zerotree successfully predicts insignificant coefficients across scales to be efficiently represented as part of growing trees.
- Successive Approximation which provides a compact multiprecision representation of the significant coefficients and facilitates the embedded algorithm.
- A prioritization protocol whereby the ordering of importance is determined, in order by the precision, magnitude, scale and spatial location of the wavelet coefficients. The larger coefficients are deemed more important than smaller coefficients.
- Adaptive multilevel arithmetic coding which provide a fast and efficient method for entropy coding string of symbols and requires no training or prestored tables

- The algorithm run sequentially and stops whenever a target bit rate or distortion metric is met.

6.2 ZEROTREES OF WAVELET COFFICIENTS:

The important aspect of low bit rate image coding is coding of the positions of those coefficients that will be transmitted as nonzero values. A large fraction of the bit budget must be spent on encoding the significance map, or the binary decision as to whether a coefficient of a 2-D discrete wavelet transform, has a zero or nonzero quantized values. This encoding of significance map results a significant improvement in compression ratio.

Compression of significance map can be improved by a data structure called zerotree. A wavelet coefficient x is said to be insignificant with respect to a given threshold T if $|x| < T$.The zerotree is based on the hypothesis that if a wavelet coefficient at a coarse scale is in significant with respect to a given threshold T, then all wavelet coefficients of the same orientation in the same spatial location in finer scales are likely to be insignificant with respect to T.

In hierarchical subband system, with the exception highest frequency subbands, every coefficient at a given scale can be related to a set of coefficients at the next finer scale of similar orientation. The coefficient at the coarse scale is called the parent, and all coefficients corresponding to same spatial location at the next finer scale of similar orientation are called children. For a given parent, the set of all coefficients at all finer scales of similar orientation corresponding to the same location are called descendants. Similarly for given child, the set of coefficient at all coarse scales of similar orientation corresponding to the same location are called ancestor. With the exception of the lowest frequency subband, all parents have four children. For a QMS-pyramid subband decomposition, the parent-child dependencies are shown in figure 6.1.

A scanning of the coefficients is performed in such a way that no child node is scanned before its parent. For an N-scale transform, the scan begins at the lowest frequency subband, denoted as LL_N, and scans subbans HL_N, LN_N, HH_N, at which point it moves on to scale an N-1 scale, etc. scanning pattern is shown in fig.6.2.

A coefficient x is said to be an element of a zerotree for threshold T if it self and of its descendents are insignificant with respect to T. an element of a zerotree for threshold T is zerotree root if it is not the descendant of a previously found zerotree root. All the descendents of zerotree root are predictably insignificant .The significance map can be efficiently represented as string of symbol from a 4-symbol alphabet. The three alphabets are following.

1) Zerotree root: Coefficient is the root of zerotree.
2) Isolated zero: Coefficient itself is insignificant but has significant descendant.
3) Positive significant: Coefficient is significant having positive sign.
4) Negative significant : Coefficient is significant having negative sign.

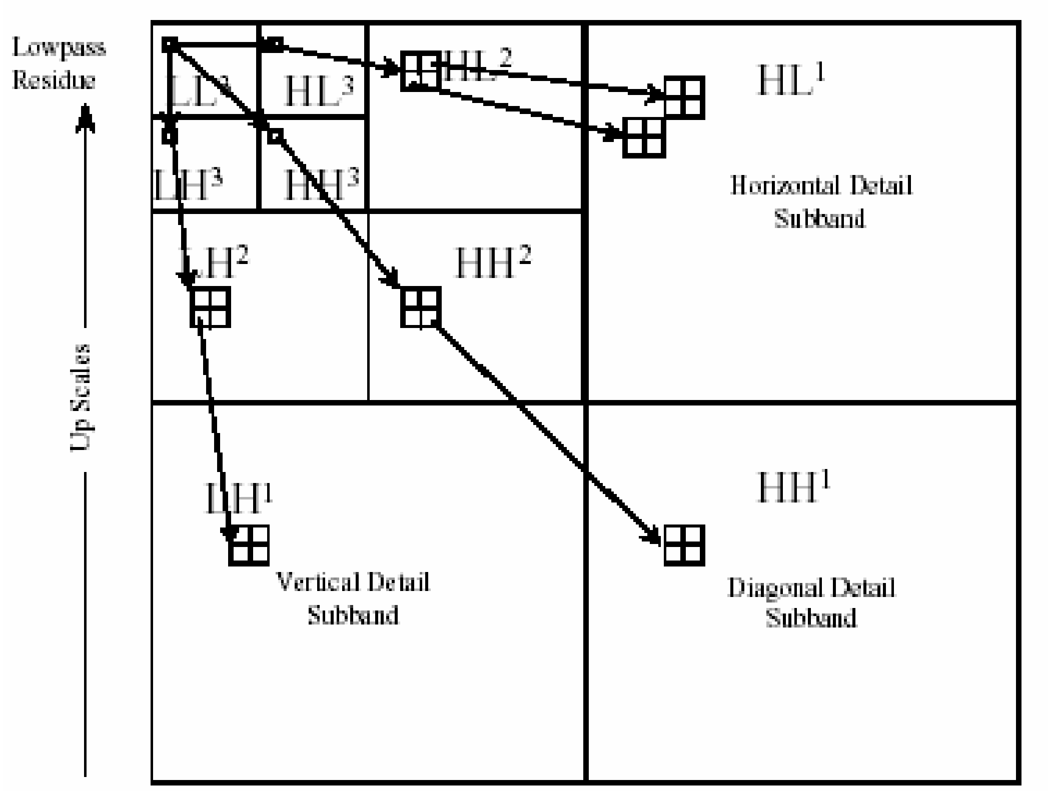
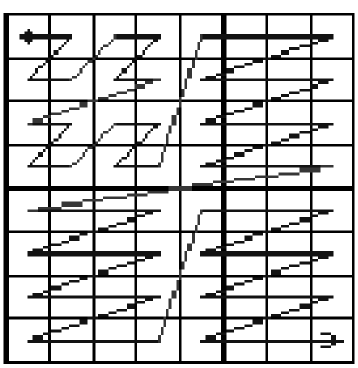
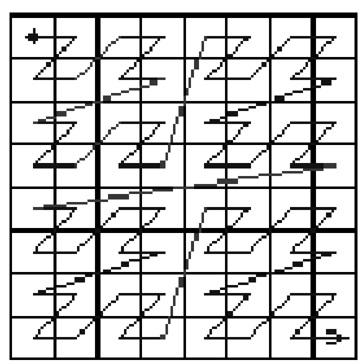


Fig 6.1



(a)



(b)

Fig6.2 (a) Raster scan order (b) Morton scan order

6.3 SUCCESSIVE – APPROXIMATION ENTROPY – CODED QUANTIZATION:

The successive –approximation (SAQ) sequentially applies a sequence of thresholds T_0, T_1,….T_{N-1} to determine significance, where the thresholds are chosen so that T_i= $T_{i-1}/2$. The initials thresholds is chosen so that $| x |< 2T_0$ for all transform coefficients x_j.

During the encoding, two separate lists of Wavelet coefficients are maintained.

1) dominant list
2) subordinate list

A dominant list contains the coordinates of those coefficients that have not yet been found to be significant. The subordinate list contains the magnitude of those coefficients that have been found to be significant. For each thresholds each list is scanned once.

During a dominant pass, coefficients in the dominant list are compared to the thresholds T_1 to determine their significance, and if significant, their sign. This significance map is then zerotree coded. If the coefficient is significant, its magnitude is appended to the subordinate list, and the coefficient in the Wavelets transform array is sit to zero so that it does not prevent the occurrence of a zerotree on future dominant passes at smaller thresholds.

In a subordinate pass, all coefficients in the subordinate list are scanned and the magnitude available to the decoder are refined to an additional bit of precision. The width of the effective quantizer step size. Which defines an uncertainty interval is cut in half. This refinement is encoded with a “1”indicating that the true value falls in the upper half of the old uncertainty level, and a “0” indicating the lower half. The string of symbols from this binary alphabet that is generated during the subordinate on the subordinate list sorted in decreasing, to the extent that the decoder has the information to perform the same sort.

The process continues to alternate between dominant passes and subordinate passes where the threshold is halved before each dominant pass. The encoding stops when the bit budget is exhausted or a target distortion metric is met.

6.4 DECODING:

In the decoding operation, each symbol, both during dominant pass and a subordinate pass, refines and reduces the width of uncertainty interval in which the true of the coefficient(s) may occur. The reconstruction value can be anywhere in the uncertainty interval. To minimize the mean square error centroid of the uncertainty interval should be used using some model of probability density function of the coefficients. For practical purpose center of uncertainty interval can also be used.

Encoding can cease at any time and the resulting bit stream contains all lower rate encoding. Similarly decoding can also be terminated at any point. Terminating the decoding of an embedded bit stream at a specific point in bit stream produces the exactly same image that would have resulted had that point been that initial target rate. This ability to cease encoding and anywhere is extremely useful in system that are either rate-constrained or distortion –constrained.

Coding and decoding can be explained by following example. Consider the simple 3- scale wavelet transform of an 8x8 image. The array of values in shown in table6.1. Here the largest coefficient is 63, so initial threshold can be chosen as 32. Table I shows the processing on the first dominant pass. The following comments refer to the Table 6.1.

- The coefficient has magnitude 63 which is greater than the threshold 32, and is positive so a positive symbol is generated. After decoding decoder will decode it as the center value of the interval [32,64], which is 48.
- The coefficient –31is significant with respect to 32, and has a significant descendant two generation down in subband LH_1 with magnitude 47. Thus, isolated zero symbol is generation for this coefficient.
- The coefficient 23 is insignificant with respect to 32 having all insignificant descendants (3, -12, -14, 8). A zerotree symbol is generated for this coefficient.
- The magnitude 10 is less than 32 and all descendants (-12, 7, -3, 2) also have magnitude less than 32. Thus a zerotree symbol is generated for it.
- The coefficient 14 is significant with respect to 32. Its children are (-1, 37, -3, 2). Since its child 47 is significant, an isolated zero is generated for this coefficient.

- No symbols are generated from subband HH_2 because all coefficients in this band are members of zerotree.
- The magnitude is significant with respect to32 and is positive so a positive symbol is generated for it.

During the first dominant pass, four significant coefficients are generated. These coefficients are refined in the first subordinate pass. Prior to this subordinate pass, the uncertainty interval for the magnitudes of all of the significant coefficient is the interval [32,64]. The first subordinate pass refine these magnitudes and identify them as being either in the interval [32,48] which will encode with the "0", or in the interval [48,32]. Which will be encoded with the symbol "1". Table 6.2 shows the symbols generated during first subordinate pass. The coefficient which fall in the interval [32,48] are decoded 40, which is center of the interval [32,48] and the coefficient which fall in the interval [48,63]are decoded as 56, which is the center of the interval [48,63].

The process continues on to the second dominant pass at the new threshold of 16.during this pass, only those coefficient not yet found to be significant are scanned. Additionally those coefficient previously found to be significant are treated as zero. Thus, the second dominant pass consists of encoding the coeffints –31 in subband LH_3 as negative coefficient, the coefficient 23 in subband HH_3 as positive coefficient, the coefficient in the subband HL_2 that have not being previously found to be significant (10, 14, -13) are encoded as zerotree roots, as are all four coefficients in subband LH_2 and all four coefficient in subband HH_2.

The subordinate list contains the magnitude (63, 34, 49, 47, 31, 23). Prior to the second subordinate pass. There are three uncertainty intervals ([16, 32], [32, 48] and [48,64]), each having width 16. The processing will refine each magnitude by creating two new uncertainty intervals for each of the previous intervals. Using the center of the uncertainty interval as the reconstructed(seeTable3).

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

Table 6.1 Wavelet coefficient of a 8*8 image

Coefficient Magnitude	Symbol	Reconstruction Magnitude
63	1	56
34	0	40
49	1	56
47	0	40

Table 6.2 Processing of subordinate pass

Table 6.3 Encoded output

D 1 : p n z t p t t t t z t t t t t t t p t t

S 1 : 1 0 1 0

D 2 : z t n p t t t t t t t t

S 2 : 1 0 0 1 1 0

D 3 : z z z z z p p n p p n t t n n p t p t t n t t t t t t t t p t t t p t t t t t t t t t p t t t t t t t t t t t

S 3 : 1 0 0 1 1 1 0 1 1 1 1 0 1 1 0 1 1 0 0 0

D 4 :

z z z z z z z t z t z n z z z z p t t p t p p t p n p t n t t t t t p t p n p p p p t t t t t p t p t t t p n p

S 4 : 1 1 0 1 1 1 1 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 1 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 0

D 5 : z z z z z t z z z z z z t p z z z t t p t t t t n p t p p t t p t t t n p p n t t t t p n n p t t p t t p p t t t

S 5 :

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

0 1 1 0 0 0 1 1 1

D 6 : z z z t t z t t t z t t t t t n n t t t

CHAPTER 7

IMPLEMENTATION, RESULTS AND CONLCUSION

7.1 Implementation

The image compression technique using discrete wavelet transform is implemented using MATLAB (version 7release 14). Different wavelet filters used for decomposition and reconstruction purpose.

1. The source image is decomposed into approximation coefficients and detail Coefficients using multiple levels of DWT.

2. The image is reconstructed using approximation, horizontal detail, vertical detail and diagonal details.

3. The image compression is analyzed for different wavelet filter viz. 'Haar', 'Daubechies', 'Biorthogonal', 'Symlets'.

4. The subband coefficients are quantized and coded using Embedded Zerotree Wavelet coder.
5. Decoder is the reverse of encoder .EZW decoder first decodes the coded stream, de-quantized and inverse discrete transform is applied to get the reconstructed image.

7.2 Flow charts

Flow charts for the major steps involved for decomposition , re-composition, EZW coding(including dominant pass and subordinate pass) and decoding algorithm is shown in following figures(7.1,7.2,7.3,7.4).

Fig. 7.1 Flow chart for coding

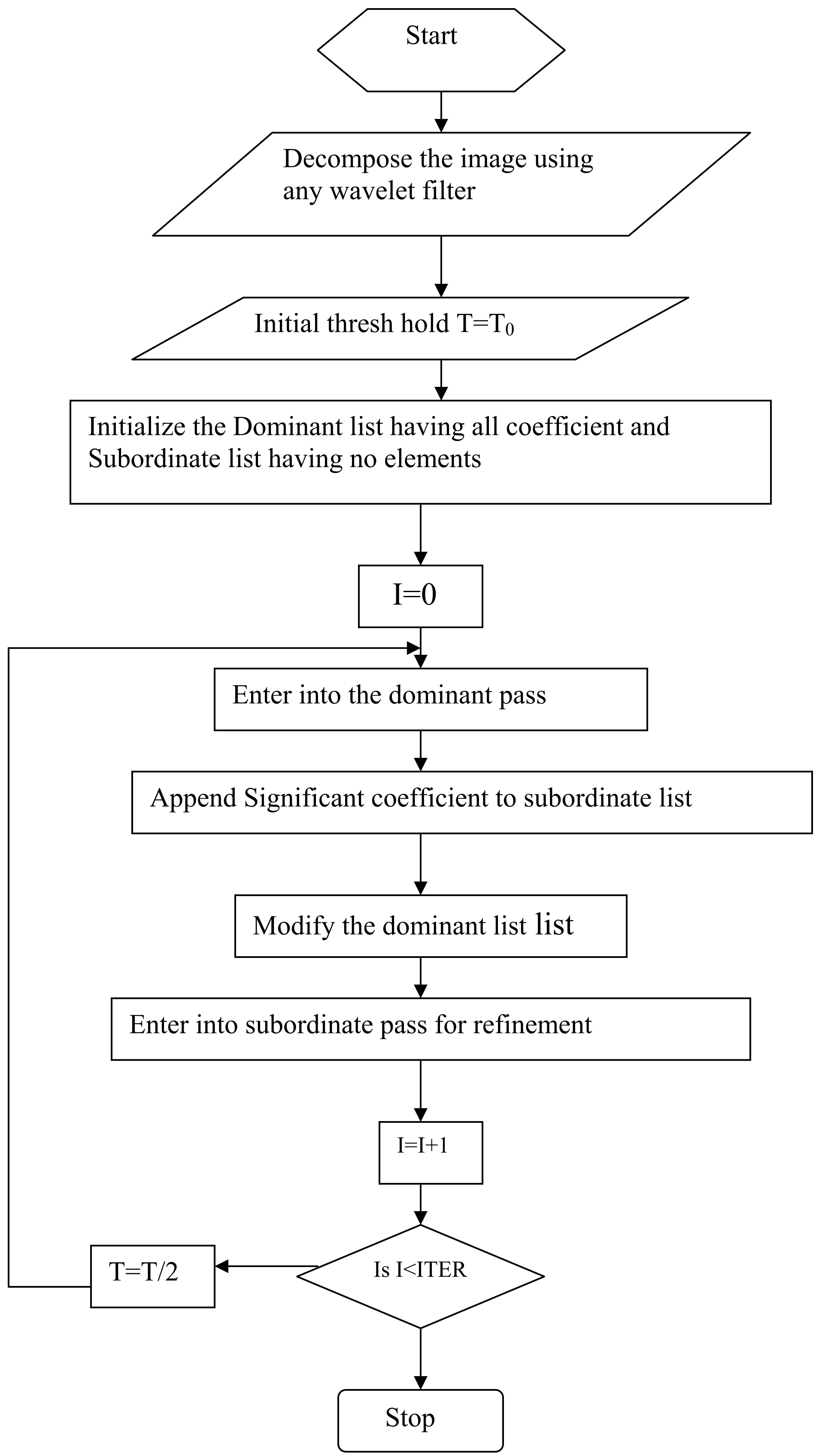


Fig 7.2Flowchart for Dominance pass

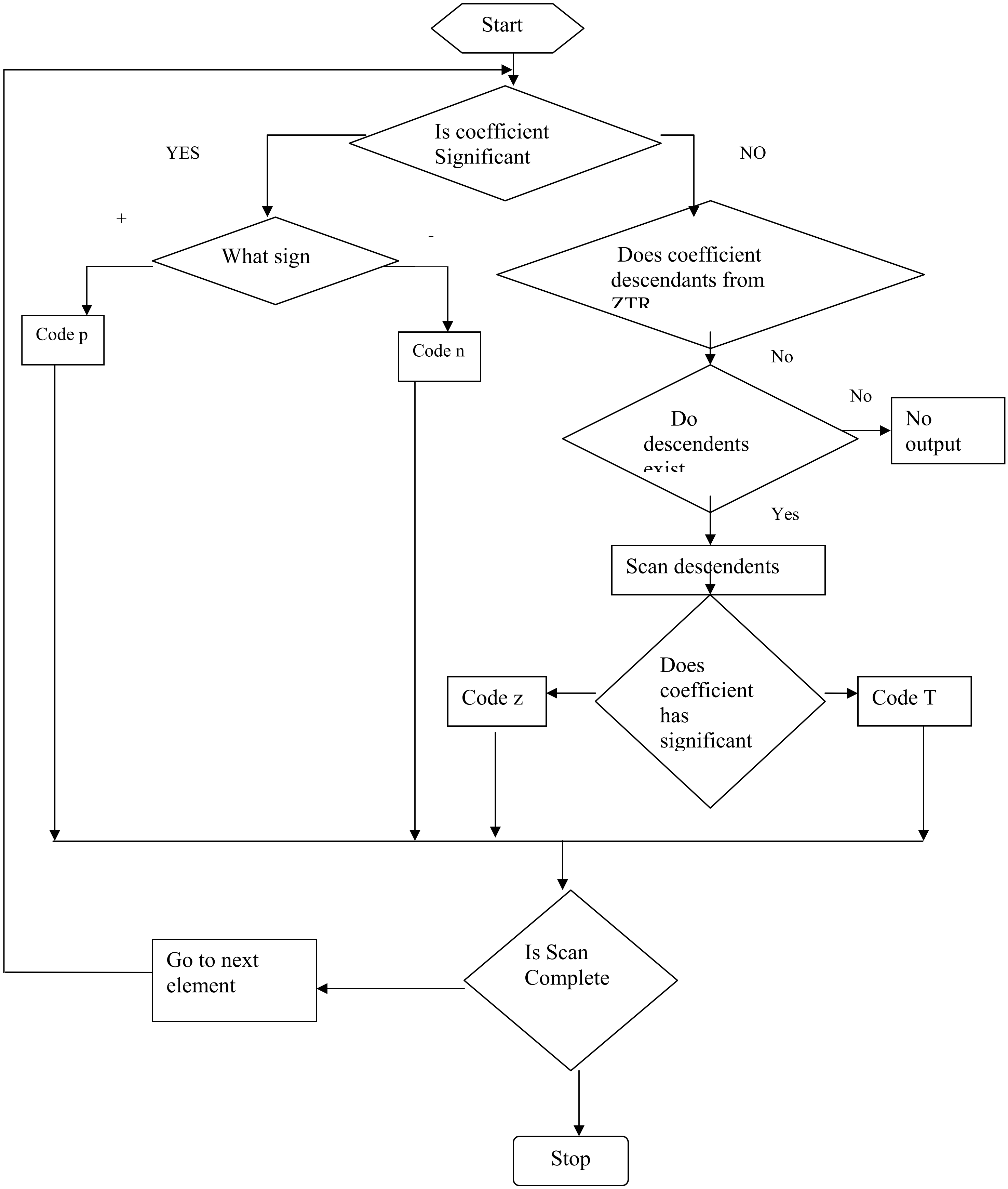


Fig. 7.3 Flow chart for subordinate pass

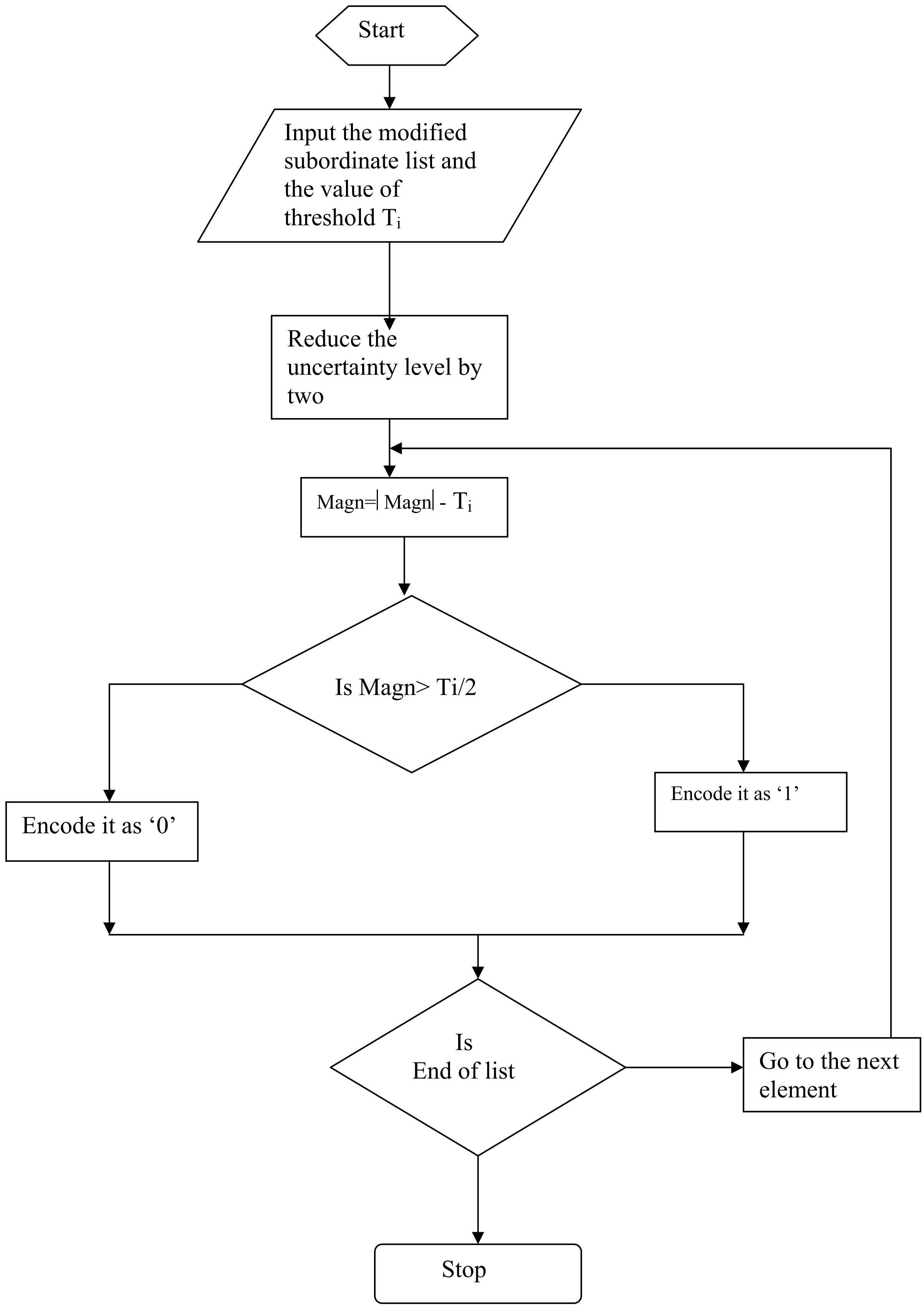
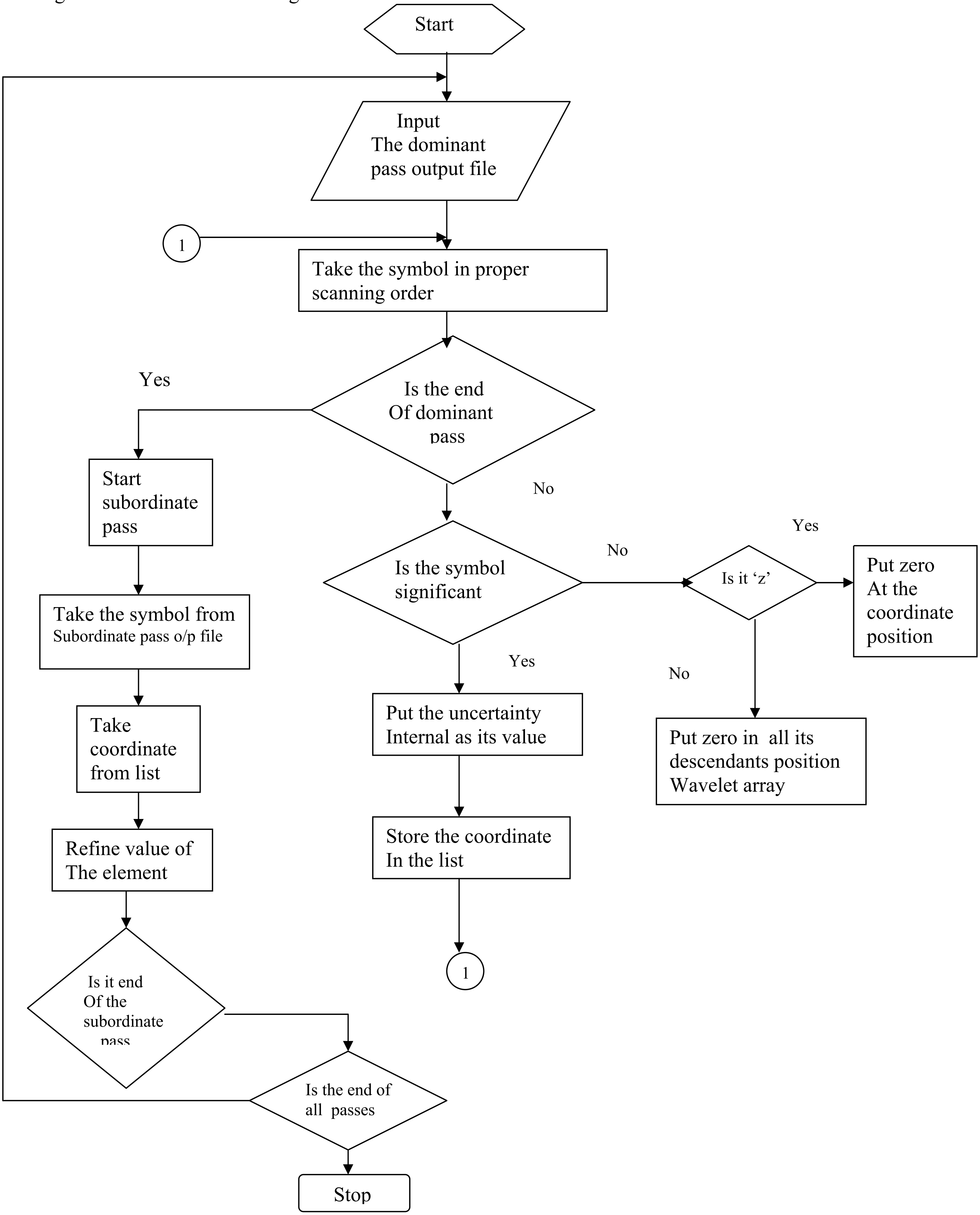


Fig 7.4 Flow chart for decoding



7.3 Result

The images used are 256 by 256 and 128 by 128 black and white images each pixel is coded up to 256 gray levels(8bpp). The evaluation has been done by computing PSNR and MSE between original and the coded images. Table 7.1and 7.2 gives the comparison. An image decomposition up to two levels has been shown in fig 7.1 and 7.2. Further the same image is compressed using different wavelet filters and best result at level 4and 5 has been shown in fig. 7.3 and 7.4.

To see the EZW encoder and decoder performances an image 128 by 128 has been taken and first decomposed at one level and its approximation coefficients are encoded using encoder and decoded by decoder . Reconstructed directly from single coefficient

by taking inverse DWT. The different pass reconstruction image is shown in fig7.5-7.9.

A table showing the symbols generated during dominant pass and subordinate pass is listed in table 7.3.

Fig 7.1One level decoposition

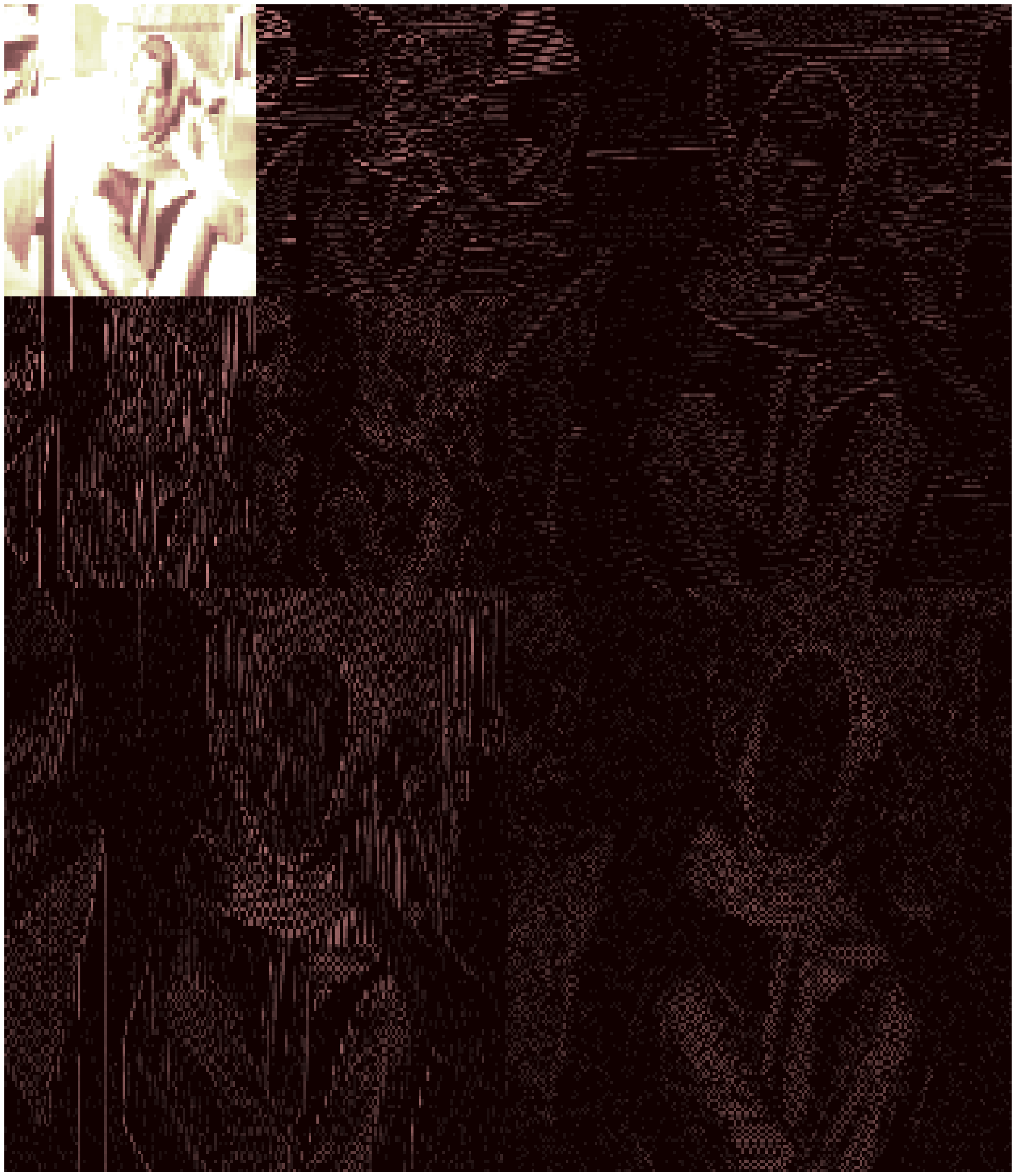


Fig 7.2 Two level decomposition

Table 7.1 Showing the compression scores, Norm. Recovery, PSNR of Wbarb by different wavelet filters on global threshold=45.

	Norm Recovery	Compression score	PSNR(db)

Wavelet type	Haar		
Level			
1	99.064	73.9731	28.706
2	98.6945	91.4749	27.2606
3	98.568	94.9493	26.8612
4	98.541	95.3964	26.7792
5	98.5383	95.4422	26.7697
	db2		
1	99.19	74.19	29.338
2	98.9117	91.7093	27.9108
3	98.8433	95.4389	27.4267
4	98.927	95.956	27.3206
5	99.101	95.9679	27.3045
	bior4.4		
1	99.27	74.1936	29.6532
2	99.0643	91.5169	28.2967
3	99.11	95.1654	27.7636
4	99.352	95.6947	27.6214
5	99.6054	95.6507	27.6088
	sym4		
1	99.2551	74.1857	29.6803
2	99.04	91.4416	28.3156
3	99.068	95.0435	27.8139
4	99.2518	95.4896	27.7036
5	99.5063	95.4337	27.6928
6	99.7406	95.31	27.691

Table 7.2 Showing the compression scores, Norm. Recovery, PSNR of Wbarb by different wavelet filters on level threshold.

	Norm Recovery	Compression score	PSNR(db)
Wavelet type	Haar		
Level			
1	100	2.99	318.6247
2	99.645	67.9047	32.9248
3	98.818	90.79	27.695
4	97.5118	97.4867	24.4595
	db2		
1	100	0	263.987
2	99.7516	66.659	34.2835
3	99.188	89.9046	28.9187
4	98.4772	96.9128	25.8068
	bior4.4		
1	100	0	259.97
2	99.827	64.857	35.4034
3	99.467	87.706	29.829
4	99.2661	94.9756	27.0336
	sym4		
1	100	0	264.88
2	99.81	65.202	35.32
3	99.4235	88.08	29.732
4	99.0534	95.6715	26.665

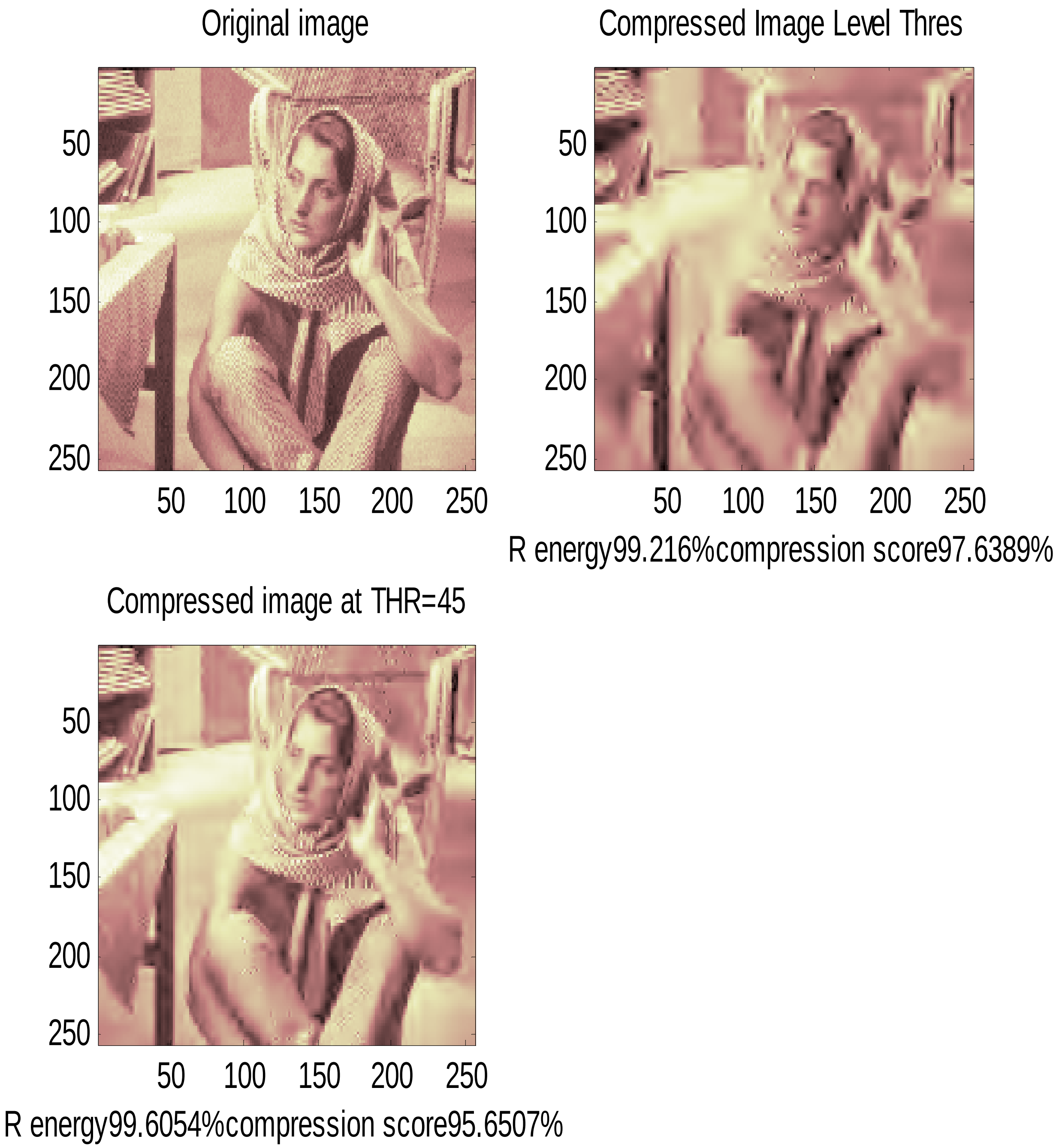


Fig 7.3 Showing Original and reconstructed images using level threshold and global threshold at level 5 with PSNR 24.6013 and 27.6088 respectively.

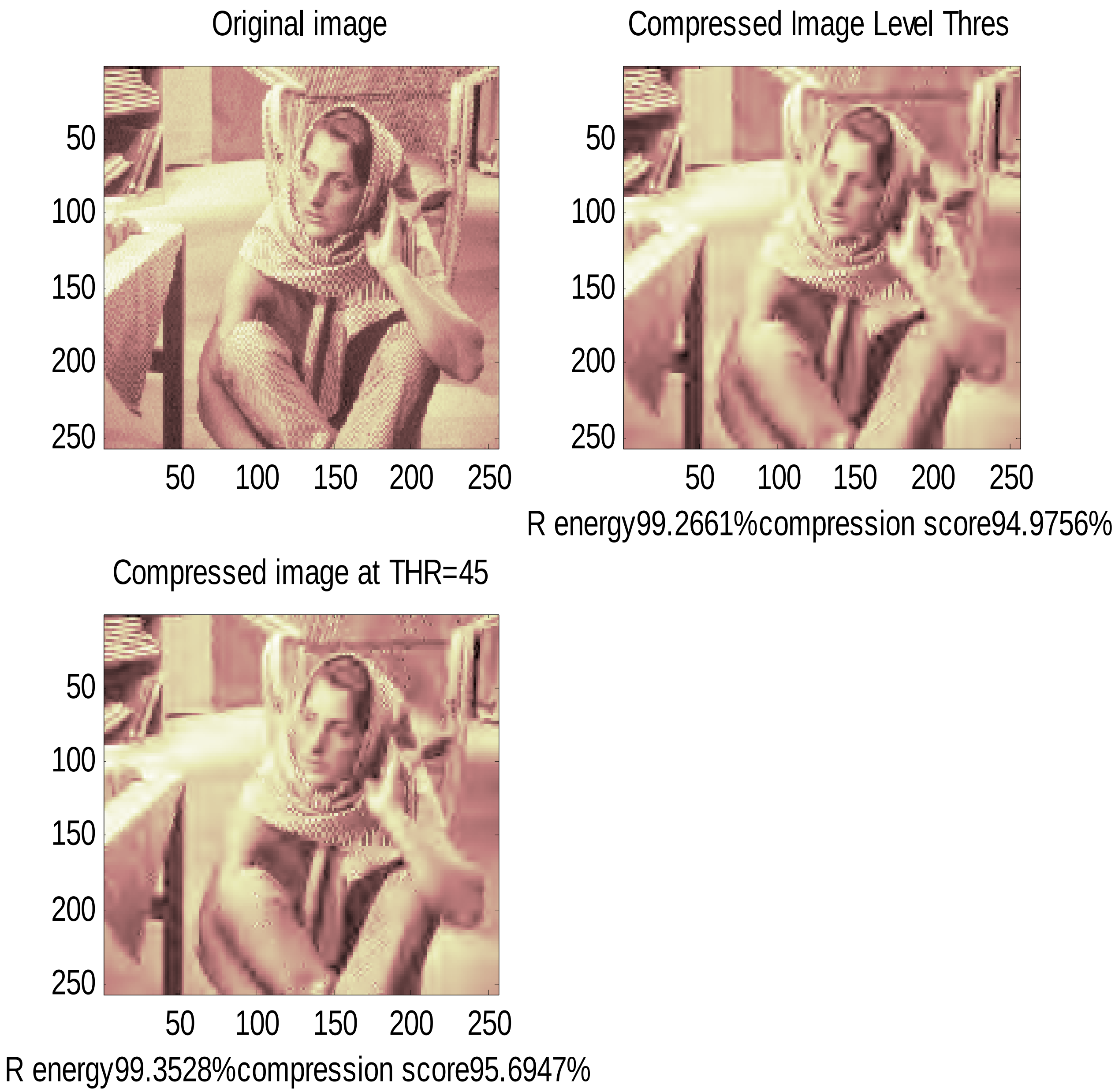


Fig. 7.4 Showing Original and reconstructed images using level threshold and global threshold at level 4 with PSNR 27.033and 27.6214 respectively.

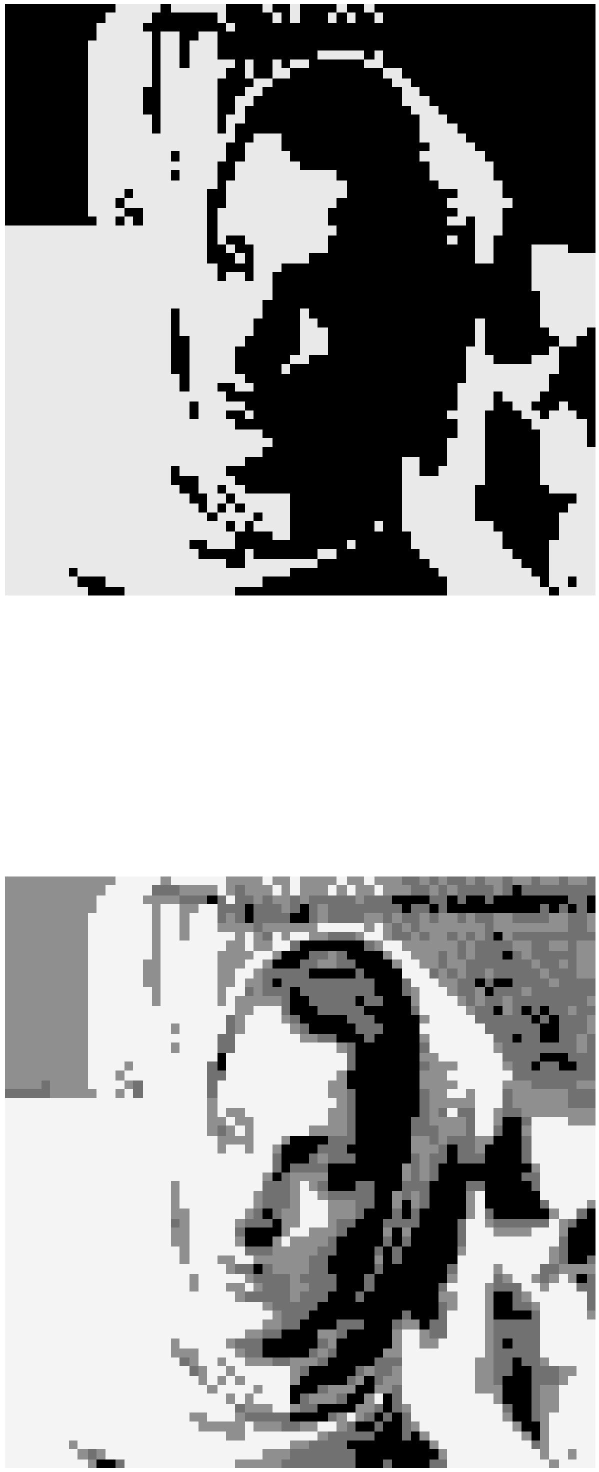


Fig 7.5 Reconstructed image (Approximation coefficient only) at decoder on threshold 256 and 128 respectively.

Fig 7.6 Reconstructed images (Approximation coefficient only) at decoder on threshold 64and 32 respectively.

Fig 7.7 Reconstructed image (Approximation coefficient only) at decoder on threshold 16 and 8 respectively.

Fig 7.8 Reconstructed image (Approximation coefficient only) at decoder on threshold 4 and 2 respectively.

Fig 7.9 Reconstructed image (Approximation coefficient only) at decoder on threshold 1.

Table7.3 Showing No. of symbols generated during dominant pass and subordinate pass

No of passes	Threshold	Dominant pass Symbol	Subordinate pass Symbol
1	256	3616	1949
2	128	3164	3462
3	64	1484	3889
4	32	608	4041
5	16	220	4096
6	8	4	4096
7	4	4	4096
8	2	4	4096
9	1	4	4096

7.4 CONCLUSION:

In this thesis I have implemented image compression using MATLAB (7.0, REL.14) and developed software for Embedded Zero Tree wavelet coding and decoding. EZW is easier to implement and achieve good performance with relatively simple algorithm. EZW does not require any prior knowledge and complicated bit allocation scheme like JPEG and vector quantization does.

The importance of biorthogonal filters has been shown. I have proposed a new wavelet filter derived from bior9.7 , the coefficient may be chosen to keep the wavelet filter closer to orthogonal , which will become more energy preserving.

EZW coding algorithm has a problem is that it performs poorly when error introduced into coded data. This is because of embedded nature of coding causes errors to propagate from the point they introduced to the end of data. So, it will not be advisable to use it where error rates are quite high.

FUTURE SCOPE OF WORK:

The work presented in this thesis opens up varies new possibilities .Here is the list of few of them:

- Arithmetic coding phase can be introduced to code the symbols generated by the EZW encoder for better performance.
- EZW coding may be extended for selective spatial decoding to increase resolution in certain portion of image.
- Color image compression can be implemented using EZW coding.
- We can find different sets of coefficients to make the biorthogonal Wavelet filters better energy preserving.

REFERECES

1) Bryan E. Usevirch , " A tutorial on modern lossy wavelet image compression" IEEE Signal Processing Magazine,vol.18 pp 22-35,sept 2001.

2) A. Skodars,C. Cristopoulos ,T. Ebrahimi , "The JPEG 2000 Still Image standard", IEEE Signal Processing Magazine,vol.-18 pp 36-58,sept 2001.

3) M. Antonini , " Image coding using Wavelet Transform", IEEE transaction on Image Processing, pp205-220,1992.

4) B. Usevitch," Optimal bit allocation for Biorthogonal Wavelet Coding" Proc.Data Compression Conf.Snowbird,UT,pp.387-395,March1996.

5) M. Vetterli and C. Harley, "Wavelet and Filter Banks", IEEE Signal Processing MagazineVol 40,pp.2207-2231,Sep. 1992.

6) R.M.Rao and Ajit S. Borpardikar, "Wavelete Transform :Introduction to theory and applications", Pearson Education,India.

7) Stephan Mallat , " A Wavelet Tour Of Signal Processing", Academic Press

8) A. Abbate ,C.M. DeCusatis, Pankaj K. Das " Wavelet and Subbands Fundamental and application" , Birkhäuser.

9) J. M. Shapiro, "Embedded Image Coding Using Zerotree of wavelet coefficient", IEEE Signal Processing Magazine,vol.41,No.-12,sept 1993

10) Z. Xiong, K. Ramchandran and M.T. Orchard ,Space frequency quantization for wavelet image coding." IEEE transaction on Image Processing,Vol-6, pp677-793,1997.

11) I. Daubechies, " Orthonormal bases of compactly supported wavelets",Comm. Pure Appl. Math., vol-41,pp. 909-996,1988.

Appendix

```
%  The following program shows the decomposition of an image

clear;

close;

% loading the image

load wbarb;

% the following command shows the image variable loaded

whos

% a single-level wavelet decomposition using Haar.

[ca1,ch1,cv1,cd1] = dwt2(X,'haar');

% The variable  ca1  is approximation coefficient

%The variable  ch1,cv1,cd1 is detail  coefficients   at level 1

% following 2nd level coefficients

[ca2,ch2,cv2,cd2]=dwt2(ca1,'haar');

%  following 3rd level coefficients

[ca3,ch3,cv3,cd3]=dwt2(ca2,'haar');

% level-one approximation and details (A1, H1, V1, and D1) from

% the coefficients ca1, ch1, cv1, and cd1, type

A1 = upcoef2('a',ca1,'haar',1);

H1 = upcoef2('h',ca1,'haar',1);

V1 = upcoef2('v',cv1,'haar',1);
```

```
D1 = upcoef2('d',cd1,'haar',1);

A=upcoef2('a',ca2,'haar',1);

H2=upcoef2('h',ch2,'haar',1);

V2=upcoef2('v',cv2,'haar',1);

D2=upcoef2('d',cd2,'haar',1);

A3=upcoef2('a',ca3,'haar',1);

H3=upcoef2('a',ch3,'haar',1);

V3=upcoef2('a',cv3,'haar',1);

D3=upcoef2('a',cd3,'haar',1);

r=[A3 H3;V3 D3];

q=[r H2; V2 D2];

p=[q H1; V1 D1];

figure

imshow(p,map);

q=[A2 H2;V2 D2];

p=[q H1;V1 D1];

figure

imshow(p, map);

p=[A1 H1; V1 D1];

figure

imshow(p,map);
```

% This is a **EZW encoder**

% This function calls following functios

% dominantpass.m

% subordinatepass.m

% checkdescents1.m

% checkchildren.m

% mapping.m

```
clear;

close all;

load woman2

[a,h,v,d]=dwt2(X,'db1');

Xdec=a;

X0=Xdec;

Y0=max(X0);

 Y1=max(Y0);

 for i=0:20;


 if 2^i<=Y1 & 2^i>0.5*Y1;


 threshold=2^i;   % get initial threshold T0;


  initialthreshold=threshold; % get initial threshold T0;


   laststeplevel=i+1;% last step level

    break;

     end;

     end;
```

```
 sublist=[];

 sub_list=[];

 [xx,yy]=size(Xdec);

A=mapping(xx);

[m,n]=size(A);

global N;      % Let Morton scanorder vector as a global variable

 N=zeros(m*n,2);

     for i=1:m,

             for j=1:n,

                          N(A(i,j),1)=i;

                                     N(A(i,j),2)=j;



               end

     end

 order=1;

while threshold ~= 0.5,  % if threshold~=1, do dominantpass and subordinatepass.

 threshold
```

```
%Dominant Pass

[D,Xdec,sublist,sub_list] = dominantpass(Xdec,threshold,sublist,sub_list);

DD{order}=D
significantlist{order}=sub_list;
 %Subordinate pass
 threshold=threshold/2;
 if threshold ==0.5,
  break;
  end
S = subordinatepass(sublist,threshold);
SS{order}=S
order=order+1;
 end
```

% EZW decoder

```
global N;
 [m,n]=size(N);% the size of initial image
              % m is the pixels of initial image
 XX=zeros(sqrt(m)); % initialize the reconstructured image to zero;
 threshold=initialthreshold; % initial theshold ;

 sublist=[]; % sublist is the new position matrix
            % for all significant coefficients 'p' and 'n';

Xezw=zeros(xx,xx,laststeplevel);
for level=1:laststeplevel,
            RR=zeros(size(XX)); % reference matrix RR;
    [a,b]=size(DD{level}); % ?
```

```
% dominant pass
     i=1; j=1;
   while j<=b,
      if RR(N(i,1),N(i,2))==0

          if DD{level}(j)=='p'
             if threshold==1
               XX(N(i,1),N(i,2))=threshold;
            else
               XX(N(i,1),N(i,2))=1.5*threshold;
            end
         end

            if DD{level}(j)=='n'
            if threshold==1
               XX(N(i,1),N(i,2))=-threshold;
            else
               XX(N(i,1),N(i,2))=-1.5*threshold;
            end
            end
```

```
        if DD{level}(j)=='t'& A(N(i,1),N(i,2))<=m/4
          RR=checkchildren(i,RR);% all zerotree's descendants are set to 1.
        end
        RR(N(i,1),N(i,2))=1; %reference matrix =1;
        i=i+1;
        j=j+1;
        else i=i+1;
    end

end
```

```
% subordinate pass

[xx,yy]=size(significantlist{level});

threshold=threshold/2;

  for i=1:xx,

      if level==laststeplevel|threshold==0.5

          break
      end

     if SS{level}(i)==1

        if XX(sub_list(i,1),sub_list(i,2))>0;
         XX(sub_list(i,1),sub_list(i,2))= fix(XX(sub_list(i,1),sub_list(i,2))+ threshold/2);

else
```

```
        XX(sub_list(i,1),sub_list(i,2))= fix(XX(sub_list(i,1),sub_list(i,2))-threshold/2);
      end
    end

    if SS{level}(i)==0

      if XX(sub_list(i,1),sub_list(i,2))>0;

        XX(sub_list(i,1),sub_list(i,2))= fix(XX(sub_list(i,1),sub_list(i,2))-threshold/2);

      else

        XX(sub_list(i,1),sub_list(i,2))= fix(XX(sub_list(i,1),sub_list(i,2))+threshold/2);
      end

    end

  end
```

```
      threshold

      level

  Xezw(:,:,level)=XX;


end


 for p=1:laststeplevel,

R = upcoef2('a',Xezw(:,:,p),'haar',1);

 figure

 imshow(R,map);

 end
```

```
function [D,X,sublist,sub_list] = dominantpass(X,threshold,sublist,sub_list)

% Dominant pass function

D=[];

global N;

[m,n]=size(X);

% X is the coefficients matrix

R=zeros(m); % matrix R is a reference matrix, same size as X; '0' means

  %this coefficient is not a descendant from zerotree root;

[a,b]=size(N);

   if abs(X(1,1))>=threshold % X(1,1) is DC coefficient
```

```
   sublist=[sublist, abs(X(1,1))]; % put significant coefficients's value to sublist

    sub_list=[sub_list;N(1,1),N(1,2)]

% put the significant coefficients' position in sub_list

                if X(1,1)>0;

                   D=[D,'p'];

                else D=[D,'n'];

                end

                X(1,1)=0;

             else D=[D,'z'];

            end

                      for k=2:4,
```

```
if abs(X(N(k,1),N(k,2)))>=threshold,
    sublist=[sublist, abs(X(N(k,1),N(k,2)))];
  % append this significant coefficient to the subordinate list;
    sub_list=[sub_list;N(k,1),N(k,2)];
            if X(N(k,1),N(k,2))>0 % determine the sign
              D=[D,'p']; % >0,assign a "p"
            else D=[D,'n'];% <0,assign a "n"
            end
            X(N(k,1),N(k,2))=0;
% the significent coefficients is replaced by a '0' in the coefficients matrix
```

```
else

  % 2,3,4 has no parents,just check its descendants.

            result = checkdescendants1( k,X,threshold,0);

            if result==1

               D=[D,'z'];

            else

               D=[D,'t'];

               R(N(k,1),N(k,2))=1;   % Zerotree, make all its descendants

               R=checkchildren(k,R); % refference matrix component to 1.
            end

end
```

```
end

for k=5:a,

    if abs(X(N(k,1),N(k,2)))>=threshold,

       sublist=[sublist, abs(X(N(k,1),N(k,2)))];

       sub_list=[sub_list;N(k,1),N(k,2)];

            if X(N(k,1),N(k,2))>0, % determine the sign

               D=[D,'p']; % >0,assign a "p"

            else D=[D,'n'];% <0,assign a "n"

            end

            X(N(k,1),N(k,2))=0;
```

```
elseif R(N(k,1),N(k,2))==0

          result = checkdescendants1( k,X,threshold,0);

  % Check its has significant descendants?

          if result==1,

             D=[D,'z']; % isolated zero

          else D=[D,'t'];% zerotree

             R(N(k,1),N(k,2))=1;

             R=checkchildren(k,R);

          % if zerotree, reference matrix coefficients=1

          end

end
```

```
          end

function S = subordinatepass(sublist,threshold)

S=[];

 [m,n]=size(sublist);

 for i=1:n;

          if bitand(uint8(sublist(1,i)),threshold)==threshold,

%      if sublist(1,i)>=threshold,
         S=[S,1];
         %sublist(1,i)=sublist(1,i)-threshold;
     else S=[S,0];
     end

end
```

```
function   RR=checkchildren(j,RR)

% if a symbol 't' is encounted, then make all its descendants in reference

% matrix RR's components equal 1---ZEROTREES

global N

[m,n]=size(N);

for i=(4*j-3):4*j;

      if  i<=m,

         RR(N(i,1),N(i,2))=1;

         RR=checkchildren(i,RR);
          end

end;
```

```
function   result = checkdescendants1(j,X,threshold,result)

% initial set result=0

% if the result=1, means that a coefficient has at least

% 1 significant descendant.

global N

[m,n]=size(N);

for i=(4*j-3):4*j;

    if result==1 | i > m

       break;

    end;
```

```
        if abs(X(N(i,1),N(i,2)))>=threshold
    result=1;
    break;
  else
  result=checkdescendants1(i,X,threshold,result);
  end;
end;
```

```
function A = mapping(n)

if n == 2

  A = [1 2; 3 4];

else

  B = mapping(n/2);

  A = [B B+(n/2)^2; B+(n/2)^2*2 B+(n/2)^2*3];

end
```

```
function PSNR= psnr2(cimg, oimg)

%psnr2 Comput the PSNR of the compressed image cimg and original oimg


[m,n]=size(cimg);


%calculate MSE

eimg=cimg-oimg;

mse=sum(sum(eimg.^2))/(m*n);


%calculate RMSE

RMSE=sqrt(mse);


%calculate PSNR

PSNR=20*log10(255/RMSE)
```

```
function MSE= mse2(cimg, oimg)

%  compressed image cimg and original oimg


[m,n]=size(cimg);


%calculate MSE

eimg=cimg-oimg;

MSE=sum(sum(eimg.^2))/(m*n)
```

```
clear;
close all;
% Load original image.
load wbarb;
nbc = size(map,1);
% Perform a wavelet decomposition of the image
wname = 'bior4.4'; lev =5;
[c,s] = wavedec2(X,lev,wname);
% for image compression using the adviced parameters.
alpha = 1.5; m = 6*prod(s(1,:));
```

```
[thr,nkeep] = wdcbm2(c,s,alpha,m);

%nkeep is Level of coeficient at level i
% thresholds with hard thresholding.

[xd,cxd,sxd,perf0,perfl2] = ...
        wdencmp('lvd',c,s,wname,lev,thr,'h');

%xd compressed matrix

%cxd  vector norm coeficient

% Plot original and compressed images.

colormap(pink(nbc));

subplot(221), image(wcodemat(X,nbc)),

title('Original image')
```

```
subplot(222), image(wcodemat(xd,nbc)),

title('Compressed Image Level Thres')

%cal of PSNR and MSE
psnr2(xd,X);

mse2(xd,X);

% xlabel display

xlab1 = ['R energy',num2str(perfl2),'%'];

xlab2 = ['compression score',num2str(perf0),'%'];

xlabel([xlab1          xlab2 ]);

% user given global threshold

th=45;
```

```
[xc,cxd,sxd,perf0,perfl2] = ...

        wdencmp('gbl',c,s,wname,lev,th,'h',1);

subplot(223), image(wcodemat(xc,nbc)),

title(['Compressed image at THR=',num2str(th)])

xlab1 = ['R energy',num2str(perfl2),'%'];

xlab2=['compression score',num2str(perf0),'%'];

xlabel([xlab1       xlab2 ]);

%cal of PSNR and MSE

psnr2(xc,X);

mse2(xc,X);
```