SIGNAL COMPRESSION USING REDUNDANT 1-D DISCRETE WAVELET TRANSFORMS

by

NARSIMHA BHARAT KOTIKANYADANAM, B.E.

A THESIS

IN

ELECTRICAL ENGINEERING

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

Approved

Tanja Nicolette Karp
Chairperson of the Committee

Sunanda Mitra

Brian Nutter

Accepted

John Borrelli
Dean of the Graduate School
December 2005

ACKNOWLEDGEMENTS

I would like to thank Dr. Tanja Karp, whose constant guidance and support helped me write this thesis. I also thank Dr. Sunanda Mitra and Dr Brian Nutter for providing valuable suggestions and service as committee members. I express gratitude towards friends and relatives for their help and companionship.

I also thank University Writing Center and Vietnam Archives at Texas Tech University for making my years as a graduate student more rewarding. I lastly thank the United States Government for granting me the opportunity to pursue research oriented graduate studies at Texas Tech University. Finally, I am grateful for my parents and sister for their unfathomable love, as always.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

ABSTRACT v

LIST OF FIGURES vi

CHAPTERS

1. INTRODUCTION 1

1.1 Motivation 1

1.2 Problem Statement 2

1.3 Outline of Thesis 4

II. BACKGROUND 6

2.1 Discrete Wavelet Transform (DWT) and Inverse Transform 6

2.1.1 Why Wavelets over Fourier analysis? 6

2.1.2 Multiresolution Analysis 9

2.1.3 DWT Implementation and its Inverse 7

2.1.4 Orthogonal vs. Biorthogonal Wavelet filters 10

2.2 Signal Compression with DWT 12

2.2.1 Signal Compression for efficient storage and transmission 12

2.2.2 Quantization of DWT coefficients 13

2.3 Trellis Diagrams 13

III. OVERCOMPLETE DISCRETE WAVELET TRANSFORMS FOR SIGNAL COMPRESSION 14

IV. COMPRESSED DOMAIN PROCESSING OF 1-D SIGNALS METHODOLOGY AND RESULTS 19

4.1 General Description 20

4.1.1 Quantization Schemes 20

4.1.2 Addition of Sub-bits to improve resolution 23

4.1.3 Algorithm for encoder 27

4.1.4 Algorithm for decoder 28

4.2 Haar Wavelet Implementation 29

Experimental Results 29
4.3 DB2 Wavelet Implementation 36
Experimental Results 39
V. CONCLUSION AND FUTURE WORK 49
REFERENCES 51
APPENDIX 53

ABSTRACT

Methods of compressing data prior to storage and/or transmission are of significant practical and commercial interest. Signal compression using wavelet expansion-based transform coding techniques has become popular. In addition to providing maximum compression, signal compression algorithms should be designed to provide good signal reconstruction with minimum computational overhead. To facilitate compression and storage, the sub-band coefficients are quantized. The quantization process is irreversible and introduces distortion. The Discrete Wavelet Transform (DWT) filter bank perfect reconstruction property no longer holds. Further, aliasing of quantization error leads to large distortion in the reconstructed signal.

The concept of the Overcomplete Discrete Wavelet Transform (ODWT) is introduced. The general idea is to exploit the phase shift relationship between ODWT quantized member sets and develop an iterative process wherein one ODWT quantized member set is used to improve the quality of the other. This idea has been implemented with Haar wavelets and Daubechies DB2 wavelets. The results are used to explore the feasibility of adding this ODWT functionality to existing signal compression algorithms like SPIHT, which uses the Daubechies 9, 7 biorthogonal wavelet.

LIST OF FIGURES

1.1	Low pass & high pass frequency response of DWT filter	2
1.2	Effect of quantization of ODWT coefficients	3
2.1	Difference between a wave function and a wavelet function	6
2.2	STFT and Wavelet Multiresolution Analysis	7
2.3	Relationship between scaling and wavelet function spaces	8
2.4	2-channel filter bank used in sub-band coding	10
2.5	Orthogonal and Biorthogonal Wavelet Filters	12
3.1	Overcomplete Discrete Wavelet Transform	14
3.2	Straightforward Phase shifting network using 2-channel Filter Bank	15
3.3	Direct Phase-Shifting in Wavelet Space	16
4.1	Discrete Wavelet Sub-band encoder with quantization	20
4.2	A Scalar Quantizer for Scheme 1	21
4.3	Decision block criteria depiction	22
4.4	Scalar Quantizer for Scheme 2	23
4.5	Addition of 2 bits to improve resolution	24
4.6	LSB combinations and transition matrices (T_{MN}) for DB2 case	26
4.7	Trellis diagram for LSB transitions at decode	27
4.8	Encoder	28
4.9	Decoder Block Diagram	28
4.10	Quantized zeroth and first ODWT polyphase set	30
4.11	Quantization errors for all the 4 ODWT members after 1st iteration.	32
4.12	Quantization errors for all the 4 ODWT members after 2nd iteration.	34
4.13	Quantization error range for first LSB bit.	37
4.14	The location of a particular wavelet coefficient estimate.	38
4.15	ODWT estimate lies within its maximum and minimum limits	39
4.16	Quantization errors for all the 4 ODWT members after 1 LSB addition	40
4.17	Quantization errors for all the 4 ODWT members after 1 LSB addition	42

4.18 Quantization errors for all the 4 ODWT members after 2nd LSB addition 45
4.19 First polyphase, low pass, after addition of third LSB 47

CHAPTER I
INTRODUCTION

1.1 Motivation

The sustained growth of the Internet and wireless networks over the last few years has resulted in the emergence of a new family of communication services for packet based communication, which involve the delivery of audio and image/video data over error-prone channels. Much of this on-line information is graphical or pictorial in nature. Methods of compressing the data prior to storage and/or transmission are of significant practical and commercial interest. The compression of signals, one of the main applications of digital signal processing, uses signal expansions as a major component. "*That which shrinks must first expand*" – Lao-Tzu, Tao Te Ching [1].

An expansion can be viewed as a set of functions that represent a given signal in vector space. Signal compression using wavelet expansion-based transform coding techniques has become popular. Due to the fact that in the transform domain, the wavelet coefficients are de-correlated and can be more efficiently coded than the original data in temporal form. Furthermore, wavelet transforms are both computationally efficient and inherently local (i.e. their basis functions are limited in duration) [2]. By "efficient" it is meant that for a given complexity of the encoder, better compression is achieved.

JPEG2000 standard uses wavelet coding techniques by passing the input image through a set of filter banks. Another recent algorithm is Backward Coding of Wavelet Trees (BCWT) known to possess resolution scalability, significantly lower memory usage, faster coding speed and lower complexity than other wavelet-tree-based codecs [4]. Source coding aims to remove redundancy to produce an efficient representation of a signal. Channel coding involves the addition of redundancy to provide robustness in a noisy transmission environment. Multiple Description Coding (MDC) [3] is a joint-source channel-coding scheme, which has proven to provide effective error resilience. It assumes the existence of multiple independent channels between the transmitter and the receiver. With MDC, several coded streams of a media source, called descriptions, are

generated and transmitted over different channels. At the destination, if all streams are received error free, a maximal signal reconstruction is possible.

1.2 Problem Statement

In sub-band coding, the input signal is decomposed into a set of band-limited components, called sub-bands, which can be reassembled to reconstruct the original signal without error. Each sub-band is generated by passing the input signal through a set of lowpass and highpass filters followed by a decimator (downsampling). Reconstruction of the original signal is accomplished by upsampling, filtering and summing the individual sub-bands.

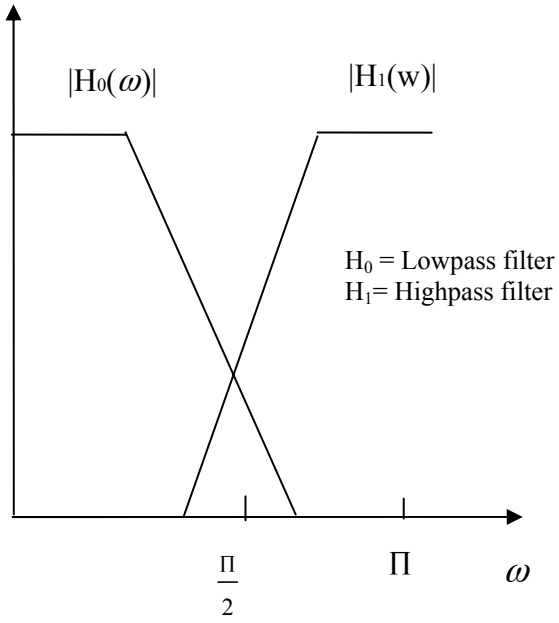


Figure 1.1 Lowpass & highpass frequency response of DWT filters.

Figure 1.1 shows a sketch of frequency response of such DWT filters (analysis and synthesis in the discrete wavelet transform, DWT) [5]. The brick wall response is not practically realizable, hence when the signal is decimated, the Nyquist criterion is violated [5] and aliasing occurs in each band. DWT filters ensure perfect reconstruction based on the fact that aliasing introduced by analysis filters gets cancelled by the synthesis filters during the reconstruction of signal. This perfect reconstruction property

holds true if no signal processing is applied to the DWT coefficients (sub-bands). In the Overcomplete Discrete Wavelet Transform (ODWT), there is no decimation of DWT sub-bands in the analysis section, and we end up with twice the number of sub-bands as compared to DWT. The advantage is that one set of ODWT sub-bands can be obtained from the corresponding set using the phase shift property of the ODWT, although the memory required to store the signal is doubled.

To facilitate compression and storage, the sub-band coefficients are quantized. The quantization process is irreversible and introduces distortion. The DWT filter bank perfect reconstruction property no longer holds. Figure 1.2 illustrates the effects of quantization of wavelet coefficients on the reconstructed signal quality.

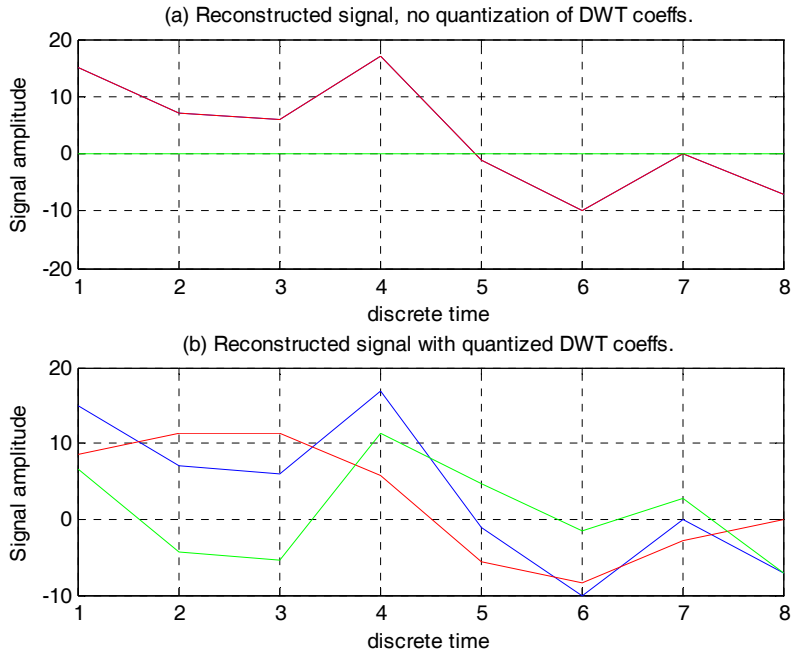


Figure 1.2 Effect of quantization of DWT coefficients. (a) No quantization, (b) With quantization of DWT coefficients. Blue curve is original signal, red curve is reconstructed signal and green curve is the difference.

In the example of figure 1.2, the DWT sub-band coefficients were quantized with a step size of 8. Figure 1.2(a) shows perfect reconstruction of input signal occurs when DWT coefficients are not quantized (Haar wavelet filters used). Figure 1.2 (b) shows that the signal cannot be reconstructed perfectly when DWT coefficients are quantized. In

some cases the error can be larger than the quantization step interval. This increase is due to the aliasing of quantization errors. The objective of this thesis is to exploit the phase shift relationship between ODWT quantized member sets and develop an iterative process wherein one ODWT quantized member set is used to improve the quality of the other. This idea has been implemented with Haar wavelets and Daubechies DB2 wavelets in order to demonstrate the feasibility of adding this ODWT functionality to existing signal compression algorithms like MDC.

1.3 Outline of Thesis

The Discrete Wavelet Transform (DWT), multiresolution expansion and the implementation of the 1-D DWT with filter banks are explained in Chapter 2. The difference between orthogonal and bi-orthogonal wavelets is discussed. This is followed by discussion on effects of quantization of DWT coefficients. Various signal compression techniques for efficient storage and transmission are presented. Finally, the concept of Trellis diagram representation is discussed.

An introduction to overcomplete discrete wavelet transform (ODWT), its design, and implementation are discussed in Chapter 3.

Chapter 4 deals with compressed domain processing of 1-D signals. The methodology, algorithms and implementation details for encoder and decoder developed for each of Haar DWT and DB2 DWT implementation are presented. The two different methods used for quantization of ODWT coefficients from Haar wavelet transform and DB2 wavelet transform are explained. The iterative process through which the decoder reduces quantization errors is explained in detail. The chapter ends with a discussion on experimental results.

Chapter 5 provides conclusions of the two different implementations leading to the feasibility of ODWT for compression and robust transmission with joint source and channel coding techniques. Future work with other advanced wavelets like DB (9, 7) is explained.

CHAPTER II
BACKGROUND

2.1 Discrete Wavelet Transform (DWT) and Inverse Transform

2.1.1 Why Wavelets over Fourier analysis?

Wavelets are localized waves and they extend not from $-\infty$ to $+\infty$ but only for finite time duration, as shown in figure 2.1.

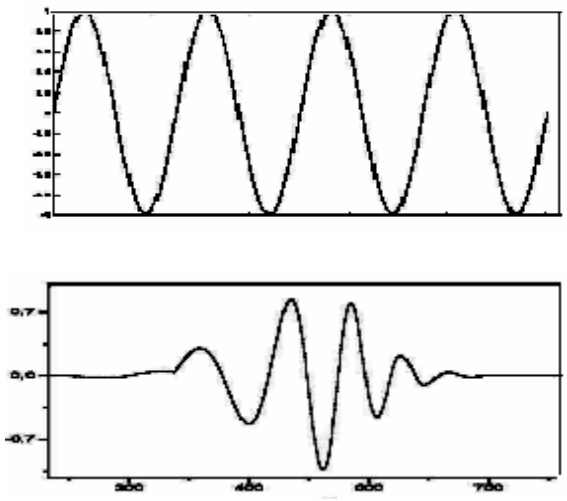


Figure 2.1 Difference between a wave function (top) and a wavelet function h(t) (bottom) which has nonzero values for a finite time window

A general equation for wavelet function h(t), which conveys the idea is:

$$h(t) = \frac{1}{\sqrt{a}}\, h\left(\frac{t-b}{a}\right) \quad . \tag{2.1}$$

We see that the wavelet transform has 2-D transformation properties with the dimensions being a, the scale or inverse frequency parameter, and b, the shift parameter. In case of waves which extend over the entire space, they do not need any shift parameter. Thus, the Fourier transform, although a useful and popular frequency analysis tool, maps 1-D time signals to 1-D frequency signals $(t \rightarrow \omega)$, whereas the wavelet transforms maps 1-D time signals to a 2-D transform domain $(t \rightarrow a, b)$ [6].

Short Time Fourier Transform (STFT) uses a windowing technique to provide a time-frequency representation of a signal using the Fourier Transform itself. STFT gives a constant resolution at all frequencies; the wavelet transform uses a multi-resolution technique by which different frequencies are analyzed with different resolutions. Figure 2.2 below illustrates this point.

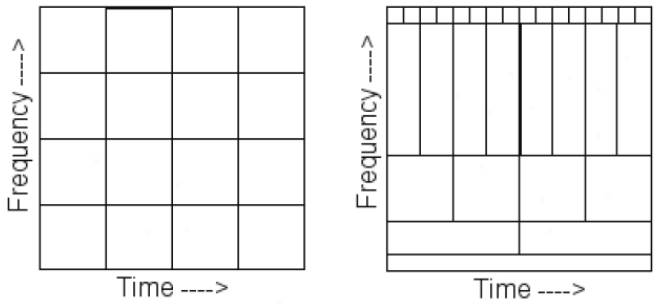


Figure 2.2 (a) STFT (b) Wavelet Multiresolution Analysis

As illustrated in figure 2.2 (b), at low frequencies, the height of boxes are shorter [7] (which corresponds to better frequency resolutions, since there is less ambiguity regarding the value of the exact frequency), but their widths are larger (implying poor time resolution). At higher frequencies the width of the boxes decreases, i.e., the time resolution gets better, and the heights of the boxes increase, providing a poorer frequency resolution. In following section, multiresolution analysis is used to explain how a wavelet transform expands a signal into the wavelet domain.

2.1.2 Multiresolution Expansion

An expansion set is a set of functions, $\Phi_k(t), k \in Z,$ which represent any signal in a vector space, V, as

$$f(t) = \sum_k a_k \Phi_k(t), k \in Z,\ f(t) \in V. \tag{2.2}$$

$\Phi_k(t)$ spans the vector space V, and is said to form a basis set if the a_k are unique for any given f(t) [5].

The wavelet expansion set includes a set of two functions, the low-pass scaling function $\Phi_k(t)$, the band-pass wavelet function $\Psi(t)$, and their translations. These functions form the basis set and divide the original subspace, V, into two different subspaces V_1 and W_1, where $V_1 \perp W_1$ and $V_1 \oplus W_1 \subset V$ as illustrated in Figure 2.3.

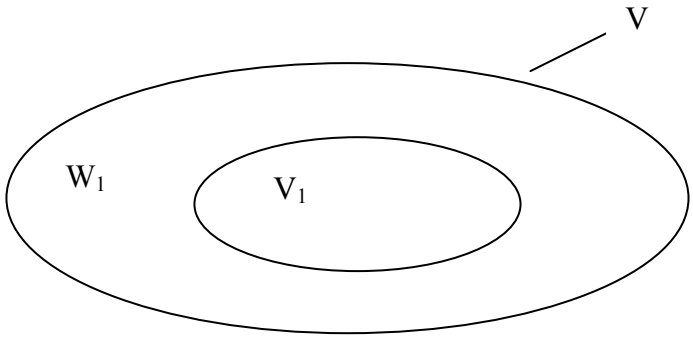


Figure 2.3 Relationship between scaling and wavelet function spaces [5].

Rather than representing the signal with one set of coefficients, the signal is now represented by two, c_k and d_k [5]. The subscript k represents each translation of the signal. Equations 2.3 and 2.4 show this new dual expansion set:

$$f(t) = \sum_k c_k \Phi_k(t) + \sum_k d_k \psi_k(t) \ . \tag{2.3}$$

or

$$f(t) = \sum_k c_k \Phi_k(t-k) + \sum_k d(k) \psi(t-k) \ . \tag{2.4}$$

The wavelet transform can expand the signal into multiple scales. This is done by dilating the scaling and wavelet functions according to the appropriate scale. Dilation of the functions is represented by a subscript *m*. In equations 2.5 through 2.8, the coefficients c_k are then expanded in the same way as the original signal.

Scale-1:

$$f(t) = \sum_k c_{1,k}\Phi_{1,k}(t) + \sum_k d_{1,k}\Psi_{1,k}(t) . \quad (2.5)$$

Scale-2:

$$c_{1,k} = \sum_k c_{2,k}\Phi_{2,k}(t) + \sum_k d_{2,k}\Psi_{2,k}(t) . \quad (2.6)$$

This process is iterated to the desired number of scales.

$$\text{Scale-m: } c_{m,k} = \sum_k c_{m+1,k}\Phi_{m+1,k}(t) + \sum_k d_{m+1,k}\Psi_{m+1,k}(t) . \quad (2.7)$$

$$f(t)\textit{Expansion}: f(t) = \sum_k c_{m,k}\Phi_{m,k}(t) + \sum_k \sum_{j=m}^{\infty} d_{j,k}\Psi_{j,k}(t) . \quad (2.8)$$

In the discrete time domain, application of the wavelet transform to a signal is typically implemented with a filter bank. This is referred to as *discrete wavelet transform* (DWT) and is discussed in the next section.

2.1.3 DWT Implementation and its Inverse

The wavelet series expansion discussed in the previous section maps a function of a continuous variable into a sequence of coefficients. If the function being expanded is a sequence of numbers, like samples of a continuous function *f(x)*, the resulting coefficients are called the *discrete wavelet transform* (DWT) [2].

The discrete wavelet transform uses lowpass and highpass filters h(n) and g(n), to expand a digital signal. These are referred to as analysis filters. These filters correspond to $\Phi(t)$ and $\Psi(t)$ of the previous section. The dilation performed for each scale is now achieved by a decimator. The coefficients c_k and d_k are produced by convolving the digital signal, with each filter, and then decimating the output. The c_k coefficients are produced by the lowpass filter, h(n), and called coarse coefficients. The d_k coefficients are produced by the highpass filter, g(n), and called detail coefficients. Coarse coefficients provide information about low frequencies, and detail coefficients provide information about the high frequencies.

Inverse discrete wavelet transform

After analyzing or processing the signal in the wavelet domain, it is often necessary to return the signal back to its original domain. This is achieved using synthesis filters

and up sampling. The wavelet coefficients are first up sampled and applied to a synthesis filter bank to restore the signal.

In the 2-channel filter bank of figure 2.4, the downsampling operators are decimators and upsampling operators are expanders [5].

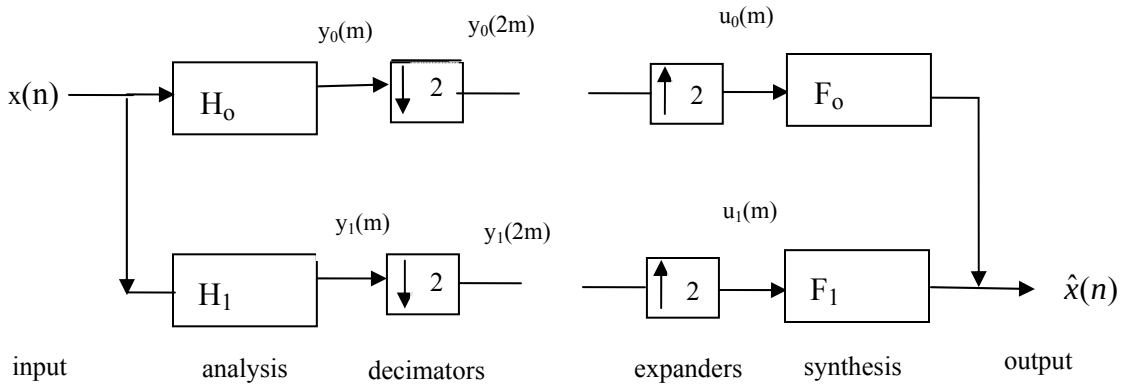


Figure 2.4 2-channel filter bank used in sub-band coding [5].

With respect to Haar and Daubechies DWT, the synthesis filters F0 and F1 are specially adapted to the analysis filters H0 and H1 in order to cancel the errors (aliasing and distortion) introduced in this analysis bank [5]. This feature makes the filter bank structure of figure 2.4 a perfect reconstruction (PR) filter bank. Furthermore, the synthesis bank consisting of F0 and F1 and the expander, is the inverse of the analysis bank. Inverse matrices automatically involve biorthogonality [5].

2.1.4 Orthogonal vs. Biorthogonal Wavelet filters

Certain properties of the Wavelet Transform depend upon which types of filters are used for analysis and synthesis. Filters used for analysis and synthesis fall into two categories: orthogonal and biorthogonal.

Classical wavelet systems use orthogonal filters, which require the filters to be orthogonal across both translation and scale. Taking the scaling and wavelet functions from Section 2.1.2, this means that

$$\left\langle \Phi_{m,k}, \Phi_{n,l} \right\rangle = \delta(k-l, m-n), \text{ and} \tag{2.9}$$

$$\left\langle \Psi_{m,k}, \Psi_{n,l} \right\rangle = \delta(k-l, m-n). \tag{2.10}$$

These requirements give a clean, robust system in which energy is preserved in

accordance with Parseval's Theorem. However, the requirements also place large limitations on the possibilities of the system. The wavelet and scaling functions must have the same length and the length must be even. These restrictions prevent linear phase analysis, except with the Haar wavelet. It is desirable, and in some cases necessary, to relax the restrictions of orthogonality. Biorthogonal filters are the result of this relaxation.

Biorthogonal filters are not required to be orthogonal across translation and scale. Instead of having each element in a basis set be orthogonal to each other, they are required to be orthogonal to the elements of a dual basis, $\{\Phi_{n,l}\}$, so that

$$\left\langle \Phi_{m,k}, \widetilde{\Phi_{n,l}} \right\rangle = \delta(k-l, m-n), \text{ and} \tag{2.11}$$

$$\left\langle \Psi_{m,k}, \widetilde{\Psi_{n,l}} \right\rangle = \delta(k-l, m-n). \tag{2.12}$$

Figure 2.5 shows an example of orthogonal and biorthogonal wavelet filters. The filters can be designed with different lengths, to include odd length filters. Perhaps most importantly, the filters can be designed symmetrically with linear phase. The only symmetric linear phase orthogonal filter is the Haar, which has a length of only two. Biorthogonal filters can be much longer than this. Longer filters generally correspond to smoother wavelet functions, and a compact signal representation. Biorthogonal filters have been shown to give better space and spatial frequency localization.

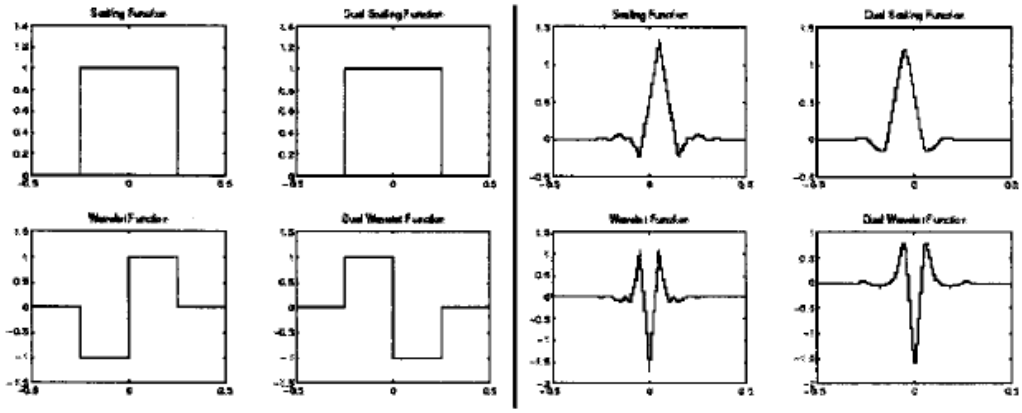


Figure 2.5 Orthogonal and Biorthogonal Wavelet Filters. (a) The orthogonal linear phase Haar filters (b) The biorthogonal linear phase Daubechies 9, 7 filters [8].

2.2 Signal Compression with DWT

2.2.1 Signal Compression for efficient storage and transmission

Most good compression schemes, particularly for images, are based on the wavelet transform. They quickly decorrelate data, which can lead to a more compact representation than the original data. The compression and translation allow the wavelet functions to represent signals in multiple levels of time and frequency resolutions. The latest image compression standard, JPEG2000 is based on the wavelet transform.

The concept of zerotree provides an efficient and embedded representation of quantized wavelet coefficients called Embedded Zerotree Wavelet Coding (EZW) [9]. A zerotree is used to represent a particular group of wavelet coefficients across different wavelet sub-bands. The zerotree approach exploits the multiresolution nature of the wavelet decomposition and has lead to several other low complexity and extremely efficient image compression schemes. SPIHT (Set Partitioning in Hierarchical Trees) improves upon EZW with better management of the zerotrees [3]. These encoding algorithms and others implement various quantization schemes (scalar, vector and hybrid quantization), and their performance is dependent on it. The objective of signal compression algorithms is to provide fast, efficient and low complexity encoding for signals (esp. images) at extremely low bit rates making them suitable for transmission across communication channels.

2.2.2 Quantization of DWT coefficients

In this thesis work, the scalar quantizer was implemented with two different methods. In the first method, the quantizer rounds the wavelet coefficients towards zero and was used for Haar wavelet implementation. The second method uses truncation as a rounding scheme and was used for DB2 wavelet implementation. Both these are discussed in detail in chapter 4.

2.4 Trellis Diagrams

Different characterization schemes are available for representing the operation of a decoder. Popular ones are the tree diagram, the state diagram and the trellis diagram

[10]. The tree diagram represents the encoder in the form of a tree, where each branch represents a different decoder state and corresponding output. A state diagram is a graph showing the different states of the decoder and the possible state transitions and corresponding outputs. A Trellis diagram uses the fact that the tree representation repeats itself after the initial stage processing is over and steady state is reached. A Trellis diagram provides a more manageable decoder description by merging nodes in the tree diagram corresponding to the same decoder state [11]. We found this compact means of representation by Trellis-like structures very helpful in describing the decoding of ODWT coefficients, and it is presented in chapter 4.

CHAPTER III
OVERCOMPLETE DISCRETE WAVELET TRANSFORMS FOR SIGNAL COMPRESSION

In this chapter, we will consider expansions that are overcomplete; that is, the set of functions used in expansion is larger than actually needed. We consider implementation with filter banks. In filter bank terminology, overcomplete means we have a noncritically sampled filter bank. The ODWT has a long history, discovered independently a number of times and given different names: undecimated DWT (UDWT) [12], the shift-invariant DWT (SIDWT) [12] and redundant DWT (RDWT) [12]. It will be referred to as ODWT in this document. There are several ways to implement the ODWT, and several ways to represent the resulting overcomplete set of coefficients. The following sections discuss the way it has been implemented for this thesis work.

Section 2.1.3 of chapter 2 discussed sub-band coding using critically sampled two-channel filter banks with analysis filters $H_0(z)$ and $H_1(z)$ and synthesis filters $F_0(z)$ and $F_1(z)$. The simplest way to obtain ODWT coefficients is not to down-sample at all, producing an overcomplete expansion [1]. Figure 4.1 shows a 2-channel filter bank with no down-sampling.

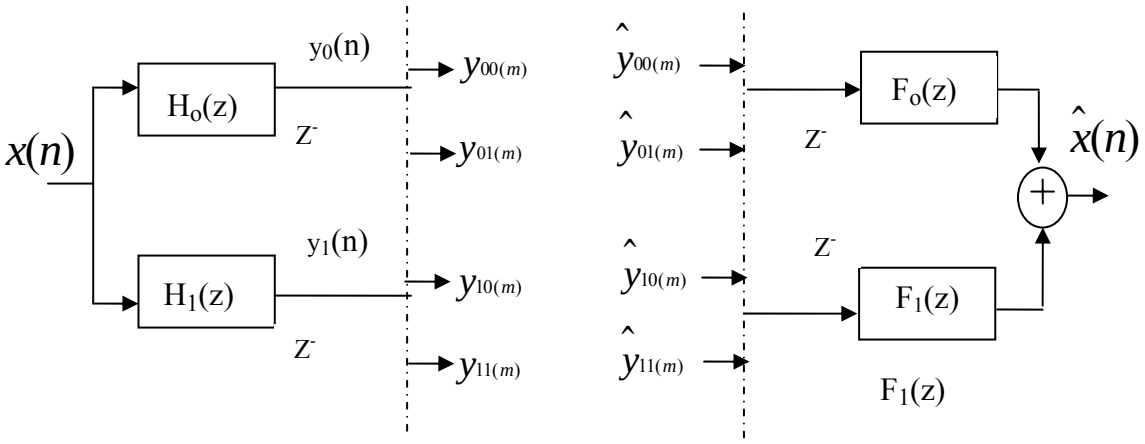


Fig 3.1 Overcomplete Discrete Wavelet Transform

The set of even-indexed coefficients is called the *zeroth polyphase* component (or set), denoted by $y_{00}(m) = y_0(2m)$ (Lowpass) and $y_{10}(m) = y_1(2m)$ (Highpass). The other set of odd-indexed coefficients is called the *first polyphase* component (or set), denoted by $y_{01}(m) = y_0(2m+1)$ (Lowpass) and $y_{11}(m) = y_1(2m+1)$ (Highpass).

The two sets of DWT coefficients of input x(n), described above, form the redundant ODWT member sets. These two ODWT member sets of coefficients, $[y_{k,0}(n)\ y_{k,1}(n)]$, k = 0, 1 have a different set of values each, since the DWT is shift-variant. However, they are redundant in the sense that an inverse DWT (IDWT) can be applied to any one of them to recover the original input signal. Furthermore, each member belonging to one set of ODWT coefficients can be calculated from the corresponding member of the other set. The simplest way to do this is to use a direct implementation of the two-channel filter bank. This is done by passing the output from the filter bank structure of figure 2.4 through the *analysis section* followed by delay blocks (to change phase), and, finally, downsampling to obtain the corresponding ODWT member of the other polyphase set.

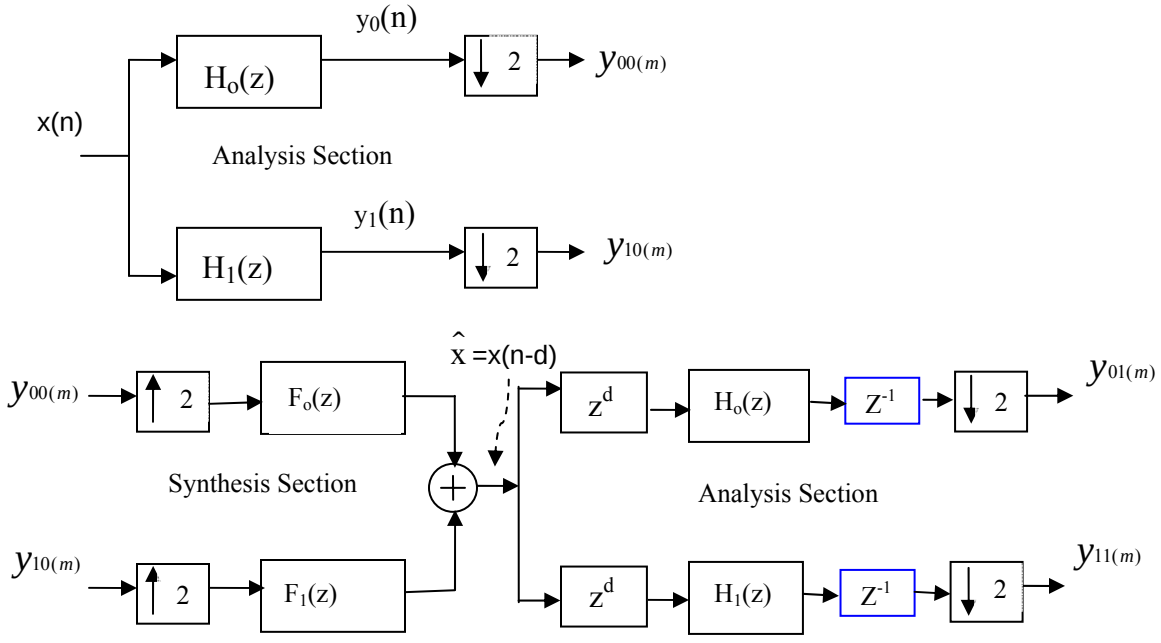


Figure 3.2 Straightforward Phase Shifting network using 2-channel Filter Bank. The "z^d" blocks undo the delay introduced by previous sections.

Figure 3.2 depicts this straightforward approach, which is basically a concatenation of an inverse transform, and a forward transform with band shifting (delay block). In this type of straightforward implementation, the analysis filter computes an output sample for each value of input, but then only one of every 2 (M=2) output points is retained. An efficient way would be not to compute the samples that get discarded at the downsampling stage. Furthermore, the operations of linear filtering and downsampling or upsampling can be interchanged if the linear filters (in analysis and synthesis blocks) are modified [13]. Such a filter bank structure which achieves the same result as the one in figure 3.2 above but in a computationally efficient manner is shown in figure 3.3.

Either the 0th polyphase set or the 1st polyphase set belonging to the ODWT member sets (obtained by passing the input data through the analysis wavelet filters) is given as input. The output obtained is the corresponding member from the other polyphase set.

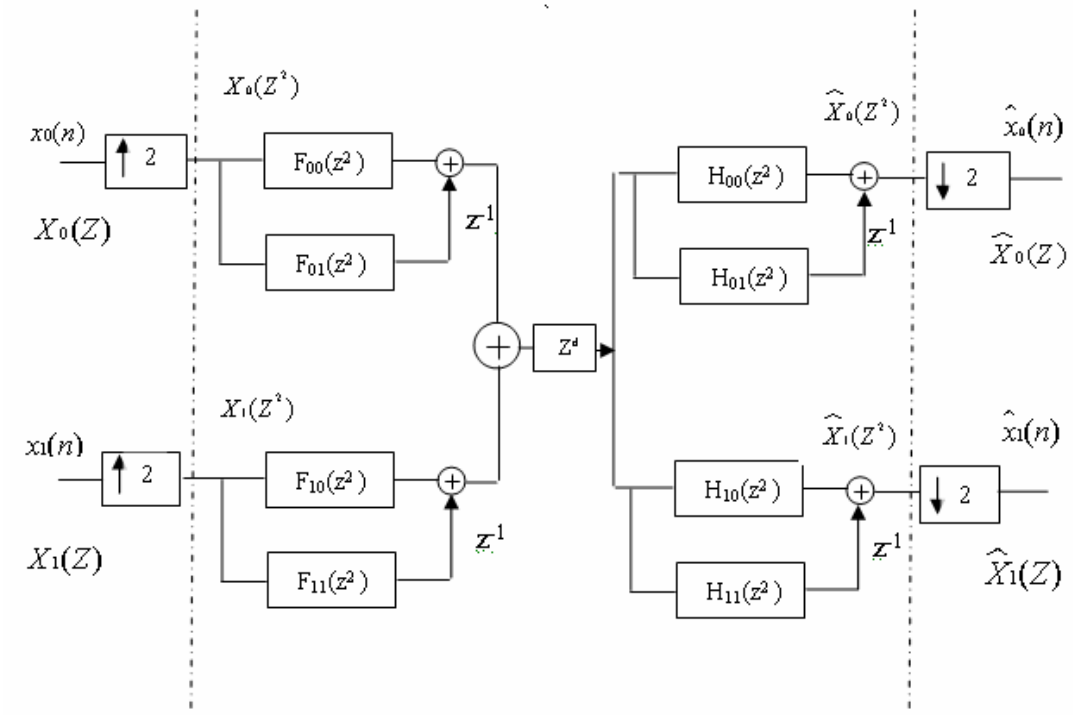


Figure 3.3 Direct Phase-Shifting in Wavelet Space

The filters used in figure 3.3 above are the down sampled versions of the corresponding filters in the analysis and synthesis sections of the 2-channel filter bank of figure 3.1. It can be expressed in a general form as:

$$H_{j,k}(n) = H_j(2n+k)\ ;\ k = 0,1. \tag{3.1}$$

If j = 0, gives low pass filters and j = 1 gives high pass filter coefficients. For instance, when j = 0, we get the filter coefficients for the top left branch of figure 3.3 as follows.

$$H_{00}(n) = H_0(2n)\ ;\ H_{01}(n) = H_0(2n+1)\ . \tag{3.2}$$

It is to be noted that the 2 new polyphase filters will be about half the size of the original filter.

Using the structure of figure 3.3, a unique phase shifting matrix can be derived and is in the form shown below.

$$\begin{bmatrix} \widehat{X}_0(z) \\ \widehat{X}_1(z) \end{bmatrix} = \begin{bmatrix} T_{00}(z)\ T_{10}(z) \\ T_{01}(z)\ T_{11}(z) \end{bmatrix} \cdot \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix}\ ;\ T = \begin{bmatrix} T_{00}\ T_{10} \\ T_{01}\ T_{11} \end{bmatrix}\ . \tag{3.3}$$

Phase Shifting Matrix T links the even phase coefficients to the odd phase coefficients and makes the phase shifting easy to implement in software. Matrix T is derived as shown below. It should be noted that the equations are in the Z domain, and when implementing in time domain, the convolution operation is used in place of multiplication in the Z domain.

With reference to figure 3.3, $\widehat{X}_j(z)$ is the output for input $X_j(z)$, j = 0,1. We can write the equation for output before final downsampling as:

$$\begin{aligned} \widehat{X}_0(z^2) = {} & H_{00}(z^2).F_{00}(z^2).X_0(z^2) + H_{00}(z^2).F_{01}(z^2).X_0(z^2).\ z^{-1} \\ & + H_{01}(z^2).F_{00}(z^2).X_0(z^2).z^{-1} + H_{01}(z^2).F_{01}(z^2).X_0(z^2).z^{-2} \\ & + H_{10}(z^2).F_{00}(z^2).X_1(z^2) + H_{10}(z^2).F_{01}(z^2).X_1(z^2).z^{-1} \\ & + H_{11}(z^2).F_{00}(z^2).X_1(z^2).z^{-1} + H_{11}(z^2).F_{01}(z^2).X_1(z^2).z^{-2}. \end{aligned} \tag{3.4}$$

After downsampling, the above equation for low pass branch output can be written as:

$$\widehat{X}_0(z) = \left\{H_{00}(z).F_{00}(z) + H_{01}(z).F_{01}(z).z^{-1}\right\}.X_0(z) + \left\{H_{10}(z).F_{00}(z) + H_{11}(z).F_{01}(z).z^{-1}\right\}.X_1(z). \tag{3.5}$$

In a similar fashion, the equation for high pass branch output is arrived at, as below:

$$\hat{X}_1(z) = \left\{H_{00}(z).F_{10}(z) + H_{01}(z).F_{11}(z).z^{-1}\right\}.X_0(z) + \left\{H_{10}(z).F_{10}(z) + H_{11}(z).F_{11}(z).z^{-1}\right\}.X_1(z). \quad (3.6)$$

With these above two equations, members of phase-shift matrix T are as follows:

$$\begin{aligned}
T_{00} &= H_{00}(z).F_{00}(z) + H_{01}(z).F_{01}(z).z^{-1}\ , \\
T_{10} &= H_{10}(z).F_{00}(z) + H_{11}(z).F_{01}(z).z^{-1}\ , \\
T_{01} &= H_{00}(z).F_{10}(z) + H_{01}(z).F_{11}(z).z^{-1}\ , \text{ and} \\
T_{11} &= H_{10}(z).F_{10}(z) + H_{11}(z).F_{11}(z).z^{-1}, \text{ where}
\end{aligned} \quad (3.7)$$

$$T = \begin{bmatrix} T_{00} & T_{10} \\ T_{01} & T_{11} \end{bmatrix}.$$

When computed, the numerical value of the T matrix is dependent on the type of wavelets used. The values of T matrix for Haar wavelet and DB2 wavelet are provided in chapter 4.

Intuitively, using a phase shift matrix based on the filter bank structure of figure 3.3 is more efficient than the straightforward approach of figure 3.2, because the repeated decimation (downsampling) operations are not performed and intermediate results are skipped.

In the straightforward approach of figure 3.2, suppose the length of the filter H(z) (lowpass or highpass) is N. The input x(n) is clocked at a rate of 1 sample per unit time. Calculating for the upper low pass branch, 3N multiplications and (3N-1) additions are required per unit time. For the phase-shifting matrix approach, the filter lengths would be N/2, and the input x(n) would be clocked at a rate of 1 per 2 units of time. Again calculating for either of the low pass or high pass branches, 3N/2 multiplications and (1/2 (2N-1)) additions are required per unit time. Furthermore, it has been shown experimentally that as much as 56% computational savings can be achieved with the direct phase shifting approach over the straightforward one in the case of the four-level 9-7 wavelet [14].

CHAPTER IV
COMPRESSED DOMAIN PROCESSING OF 1-D SIGNALS METHODOLOGY AND RESULTS

In this chapter, the algorithm used to implement the joint-source channel encoder and decoder is discussed. First, implementation with Haar wavelets is explained. Second, implementation with Daubechies family of wavelets is discussed. The Overcomplete DWT members are quantized at the encoder, and the quantized wavelet coefficients are stored and transmitted as data. At the receiver, the decoder takes one polyphase member set and estimates its counterpart polyphase wavelet coefficients using the phase shift matrix method described in chapter 3. With reference to chapter 2, perfect reconstruction of data is not possible from quantized wavelet coefficients due to aliasing of quantization error. It implies that when the decoder reconstructs the polyphase sets, the quantization error in some of the estimated values may be greater than the permissible error limit (determined by the quantization step). We have attempted to address this issue by using a decision algorithm that classifies these estimated coefficients as '*good estimates*' and '*bad estimates*'. The quantization error limits used for such classification purposes is dependent on the type of quantization scheme used. Two scalar quantization schemes were used; one rounds towards zero and the other rounds by 'truncation'. Former worked well with Haar ODWT implementation, and the latter worked with DB2 ODWT.

Summarizing, the decoder first reduces quantization errors in the zeroth polyphase ODWT set, then uses it to reduce the error on first polyphase set. This process is reiterated with update carried out alternately on both the ODWT sets till no further improvements can be attained. The encoding scheme exploits the phase shift relationship between ODWT quantized member sets using one ODWT quantized member set to improve the quality of the other.

4.1 General Description

4.1.1 Quantization Schemes

Figure 4.1 shows the encoder with quantizer blocks for scalar quantization of the wavelet coefficients after filtering. Superscript 'q' is used with polyphase ODWT member sets to indicate that they are quantized values.

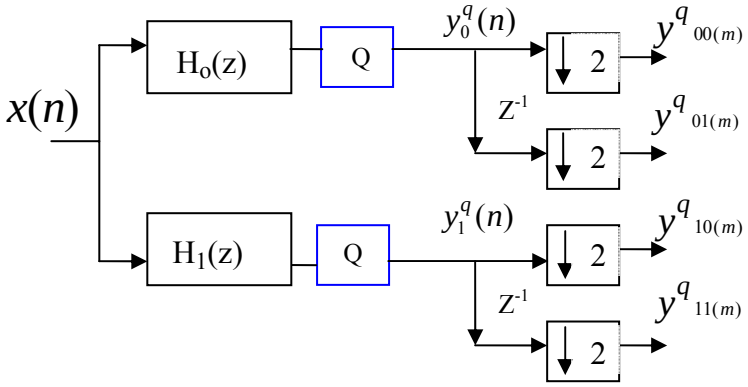


Figure 4.1 Discrete Wavelet Sub-band encoder with quantization.

In general, [$y^q_{k,0}(n)$] denotes the quantized 0th polyphase set and [$y^q_{k,1}(n)$] denotes quantized 1st polyphase set; k = 0 describes the low pass component of the respective polyphase set, and k = 1 describes the high pass part. For instance, if the input is the quantized zeroth polyphase ODWT set, the estimated output (1st polyphase set) is given by the equations:

$$\begin{bmatrix} \hat{y}_{01}(m) \\ \hat{y}_{11}(m) \end{bmatrix} = \begin{bmatrix} t_{00}(m)\ t_{10}(m) \\ t_{01}(m)\ t_{11}(m) \end{bmatrix} * \begin{bmatrix} y^q_{00}(m) \\ y^q_{10}(m) \end{bmatrix}. \qquad (4.1)$$

Note that lower case't (m)' denotes the impulse response.

Scheme 1

The quantizer rounds the wavelet coefficients towards zero and it is represented as shown in figure 4.2:

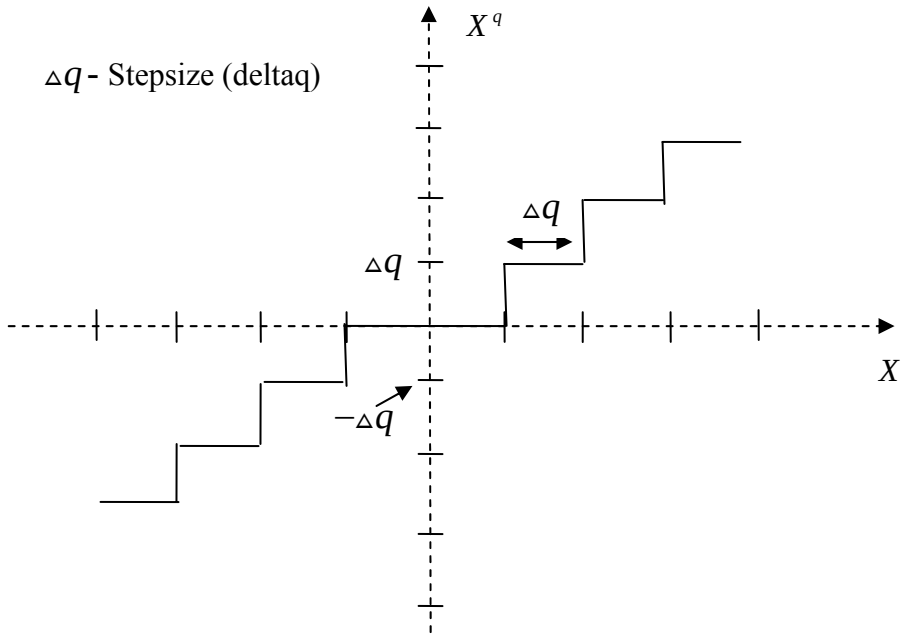


Figure 4.2 a Scalar Quantizer used in Scheme 1.

In figure 4.2, X is the input shown on horizontal axis and X^q is the quantized output on vertical axis. Then quantization error (Q_{err}) is the difference between the two.

$$Q_{err} = X - X^q. \tag{4.2}$$

The value of Q_{err} can be of 3 types, depending on the quantized value X^q:

- Q_{err} is positive if X^q is positive.
- Q_{err} is negative if X^q is negative.
- Q_{err} is unknown (i.e. either positive or negative) if X^q is zero.

Based on the above assumptions, we derive conditions for a given ODWT member, say for the 1st polyphase low pass set:

- $y_{01}^q(m)$ is positive implies that $y_{01}(m)$ (ground truth or true value) lies between $y_{01}^q(m)$ and ($y_{01}^q(m) + \triangle q$) (figure 4.3),
- $y_{01}^q(m)$ is negative implies that $y_{01}(m)$ (ground truth or true value) lies between $y_{01}^q(m)$ and ($y_{01}^q(m) - \triangle q$).

Based on these observations, a way of deciding the estimated values as either a *good* or *a bad* estimate is derived and is illustrated in figure 4.3, for positive the Q_{err} case.

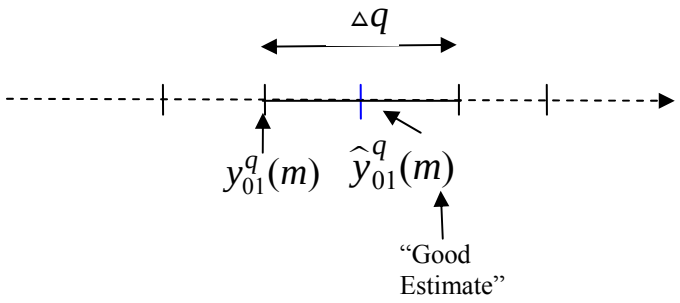


Figure 4.3 Decision block criteria depiction (positive Q_{err} case)

Any estimated value found lying outside the permissible region is classified as a *bad estimate,* and a suitable correction value based on the quantization step size can be applied. Haar wavelet implementation is carried out using this scheme.

Scheme 2 (Truncation as rounding scheme)

If we use truncation as the rounding scheme, the assumptions made for Q_{err} as having 3 types of values no longer hold. For any given X^{q}, Q_{err} is unknown (i.e. either positive or negative). Figure 4.4 shows such a quantizer. Note the step width around the origin is same as elsewhere, in contrast to the quantizer of figure 4.2.

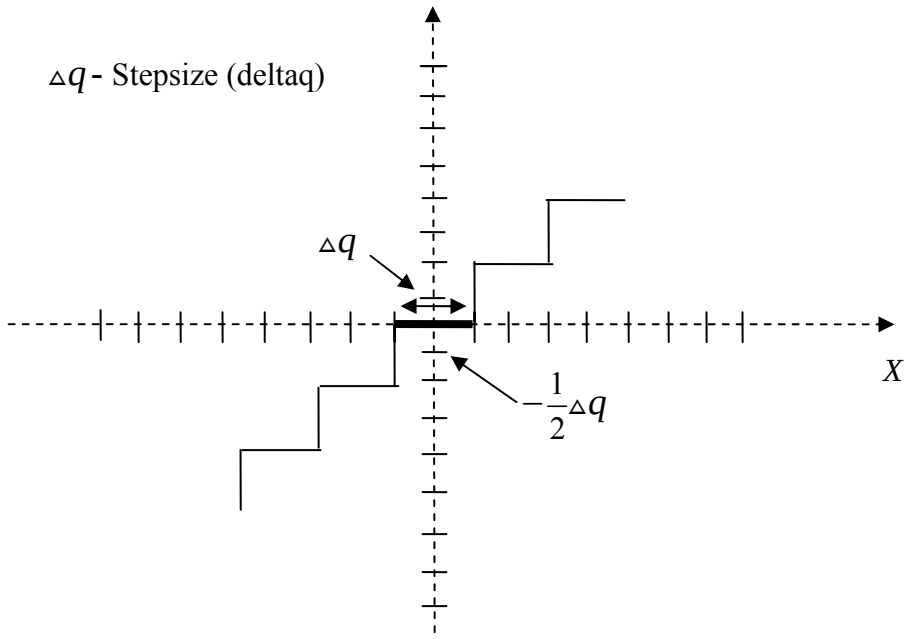


Figure 4.4 Scalar Quantizer used in scheme 2.

The quantizer now "rounds" the wavelet coefficient to the nearest quantization step level rather than "fixing" to a level towards zero. This scheme helped us to develop a more feasible decision algorithm for DB2 DWT implementation. DB2 DWT requires a memory depth of 3 taps (1 in Haar). This leads to more LSB combinations during validation and requires more looping structures as compared to the Haar DWT. Higher coding flexibility is possible in scheme 2, as the 3 types of values that could be taken by Q_{err} in scheme 1 now reduce to one type of value irrespective of X^q. Section 4.3 has a more detailed discussion on this scheme.

4.1.2 Addition of sub-bits to improve resolution

In this section, the design for adding sub-bits (LSB) to the quantized wavelet coefficients at the decoder is described. Furthermore, the case of adding 2 sub-bits is discussed in detail, as a generic example. Corresponding arguments and calculations can be derived when N (N= 1, 3 …) sub-bits are to be added.

Consider equation 4.3 and 4.4, which give the estimated 1st polyphase ODWT set. It can be expanded as follows:

$$\hat{y}_{01}[m] = t_{00}(0)\, y^{q}_{00}[m] + t_{00}(1)\, y^{q}_{00}[m-1] + t_{10}(0)\, y^{q}_{10}[m] + t_{11}(1) y^{q}_{10}[m-1], \text{ and} \qquad (4.3)$$

$$\hat{y}_{11}[m] = t_{01}(0)\, y^{q}_{00}[m] + t_{01}(1)\, y^{q}_{00}[m-1] + t_{11}(0)\, y^{q}_{10}[m] + t_{11}(1) y^{q}_{10}[m-1]. \qquad (4.4)$$

Note that a one-tap memory is required for every sample of the polyphase set that is being estimated ($\hat{y}_{01}[m]$, $\hat{y}_{11}[m]$ in this example). In above equation, [m] denotes the current sample and [m-1] denotes the previous sample (memory element).

We reduce the quantization error on this 1st polyphase set ($\hat{y}_{01}[m]$, $\hat{y}_{11}[m]$) by adding 2 sub-bits to each sample of y^{q}_{00} and y^{q}_{01} as shown below in figure 4.5. The right half of figure 4.5 illustrates the physical meaning of adding 2 sub-bits to a particular wavelet coefficient ($y^{q}_{00}[m]$ positive case).

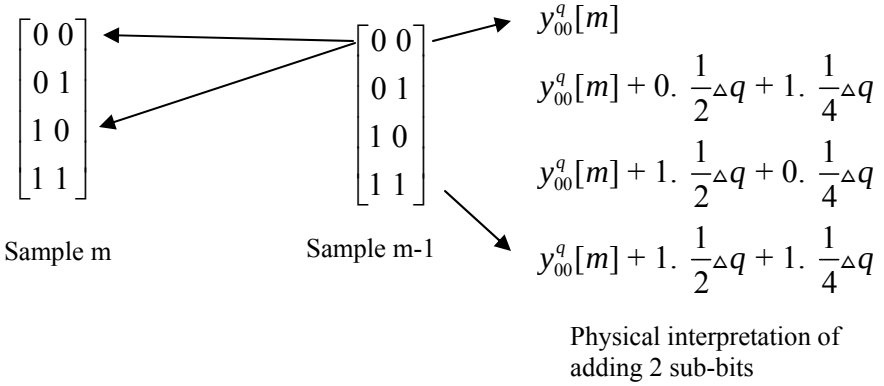


Figure 4.5 Addition of 2 bits to improve resolution (assume $y^{q}_{00}[m]$ positive).

The *update function,* in the Matlab program, adds 2 binary bits for every m[th] sample in the combination shown above. For every combination of binary bits for the m[th] sample, it checks for valid solutions with every other combination of binary bits for the (m-1) [th] sample. By "checking for valid solutions" it is meant that the values of $y^{q}_{00}[m]$ and $y^{q}_{00}[m-1]$ (in this example) are improved by adding in corresponding fractional values of step size (Δq), then the 1st polyphase (LP and HP) values are re-estimated using equations 4.3 and 4.4. These new estimates of $\hat{y}_{01}[m]$ and $\hat{y}_{11}[m]$ are quantized and checked against the true values of $y^{q}_{01}[m]$ and $y^{q}_{11}[m]$. They will be equal only when the

re-estimated values have a quantization error within half the step size. Hence, only those sub-bit combinations are kept (in matrix form) that produce such a reduction in quantization errors, and the rest of them are discarded. Note that bit combination "00" implies no change done to the coefficients. Furthermore, storage of solutions in matrix form is carried out in such a manner that discarded or "not working" bit combinations are not checked again when evaluating for subsequent samples (in the case of Haar Wavelet filters, only current and previous samples' solution matrices are needed because of the one-tap memory).

This idea is now expanded to add single sub-bit combinations for DB2 DWT implementation, which requires three-tap memory. Figure 4.6 expands the "Sub-bit combination tree" for the DB2 case (single sub-bit). Note that for set of matrices in the upper half of figure 4.6, column 1 designates the sub-bit combination for the Lowpass Polyphase set and column 2 for the Highpass.

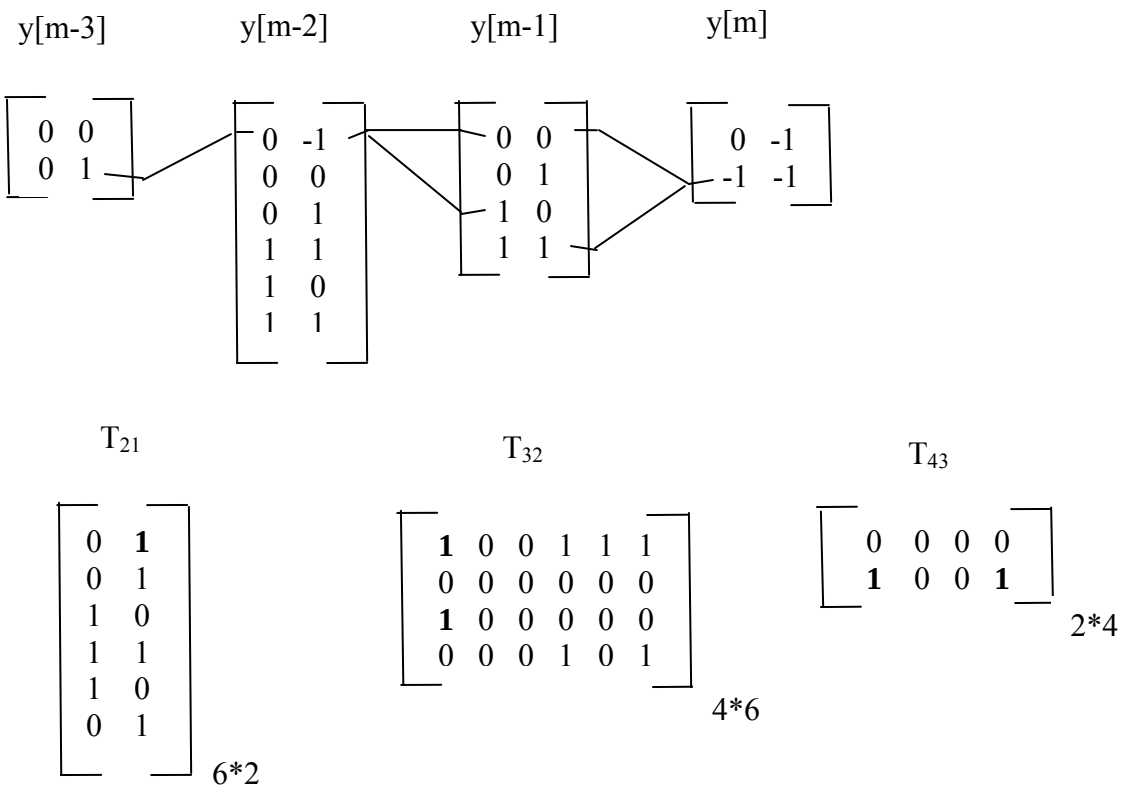


Figure 4.6 Sub-bit combinations and transition matrices (T_{MN}) for DB2 case.

Matrix T_{21} gives the sub-bit combinations that are valid (meet the quantization error limit requirement) between wavelet coefficients y[m-2] and y[m-3]. A "1" denotes a working combination and "0" a not working combination in the transition matrix T_{21}. Also note that the number of rows in the sub-bit matrix of y[m-3] gives the number of columns in T_{21}, and the number of rows of sub-bit matrix of y[m-2] determines the number of rows of $T_{21.}$ Corresponding arguments apply for transition matrices T_{32} and $T_{43.}$

In the upper half of figure 4.6, the lines drawn between sub-bit matrices give an example of valid sub-bit for a wavelet coefficient y[m] in relation to the previous three coefficients and their sub-bit combination set. The manner in which the transitions are

stored for this example is illustrated by means of highlighted "1"s in the transition matrices of figure 4.6.

We further develop this idea to come up with a Trellis-like diagram for the decoder to store the sub-bit transitions. Figure 4.7 illustrates the low pass case. The dark and dotted lines give the paths for the example explained in the previous paragraph.

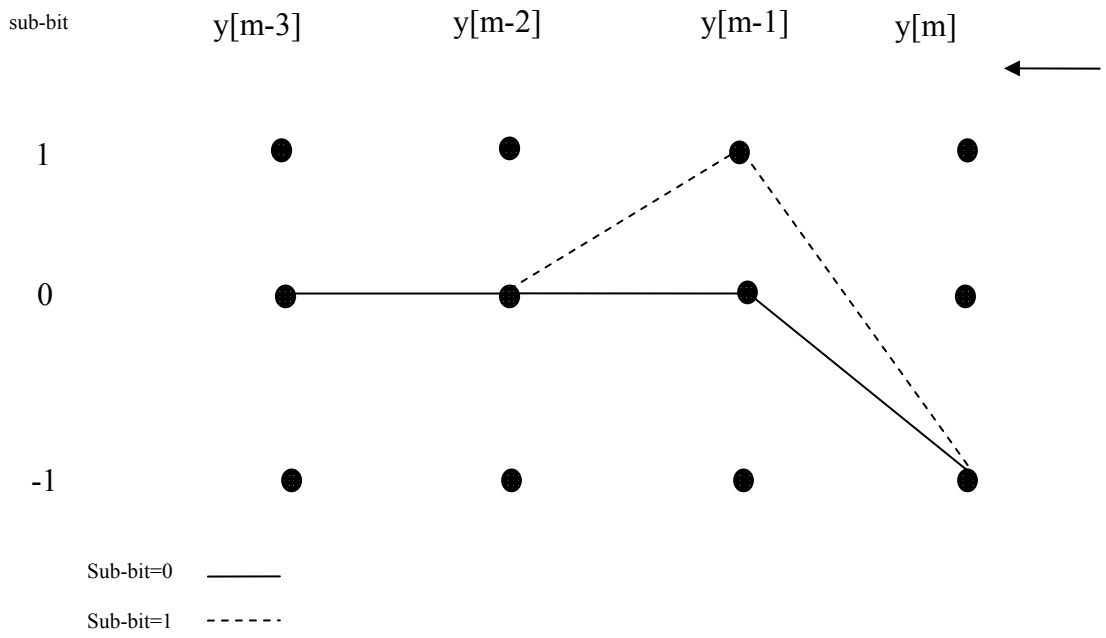


Figure 4.7 Trellis-like diagram for sub-bit transitions at decoder.

Furthermore, this helps us to relate the decoder operation to that of a state machine.

4.1.3 Algorithm for Encoder

Figure 4.8 gives the block diagram for a generic ODWT-based encoder.

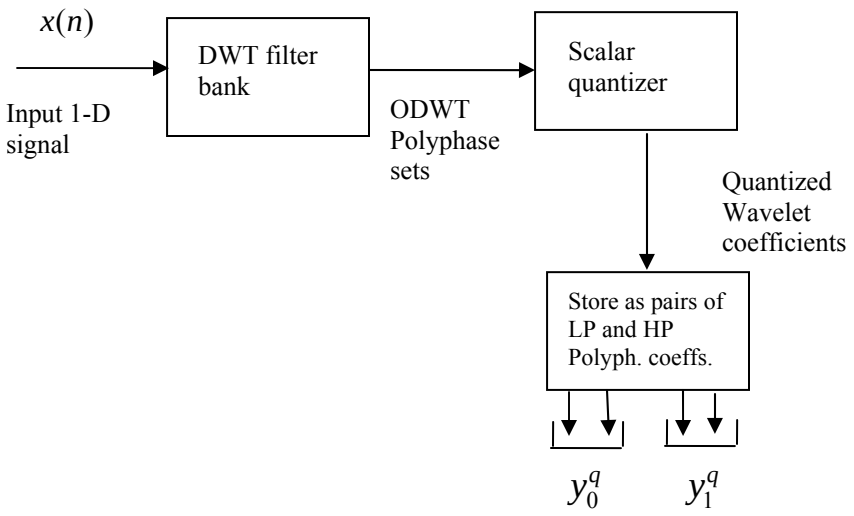


Figure 4.8 Encoder

The lowpass and highpass wavelet coefficients belonging to the respective polyphase sets are grouped and stored together to achieve an efficient decoder implementation.

4.1.4 Algorithm for Decoder

Figure 4.9 shows a block diagram outlining the decoder that can be implemented for an encoder of the type shown in figure 4.4.

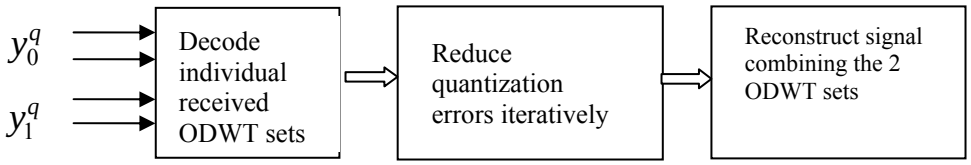


Figure 4.9 Decoder block diagram

- Input the quantized ODWT polyphase coefficient sets.
- Reduce quantization error on y_{0q} by estimating y1.
    - Add at least 1 sub-bit to LP and HP values of y_{0q} and predict the 1st polyphase component.
    - Check which values of y_{0q} result in quantization errors smaller than q_{1max} (the max quantization error limit on the positive side).

    - Refine y_{0q} by finding those samples for which all possible LSB solutions are the same for the first bit.
- Reduce quantization error on y_{1q} by estimating y_0.
    - Refine y_{1q} by finding those samples for which all possible LSB solutions are the same for the first bit.
- Reiterate by carrying out validation of solution estimates and refinements for y_{0q} and y_{1q}, alternately, till no further improvements can be achieved.

4.2 Haar Wavelet Implementation

To begin with, we add 2 sub-bits for every wavelet coefficient for either ODWT set and then carry out the iterative process of validation and refinements as described in section 4.1. From the quantizer of Figure 4.2, the permissible error limit turns out to be +8 in the positive direction and -8 in the negative one. The decoder reduces this error range (to +/-4 or even +/-2) for specific data points during the refinement process carried out after update of corresponding wavelet coefficients with the addition of an appropriate value (a fraction of $\triangle q$). The criterion for selection of such points is based on the type of LSB solutions that are retained after the validation process. For the Haar wavelet case, these points were chosen when all possible sub-bit solutions are found to be same for the first bit. This would indicate the region where the quantization error lies, thereby allowing the decoder to narrow the error margin (shown as red line in the plots).

Experimental Results

One-dimensional data input of length 256 samples is used. The nature of this input stream resembles a row of image data (MatLab code snippet provided in Appendix). The quantization step is $\triangle q$=8. The phase-shift matrix, T coefficients for Haar DWT are as follows:

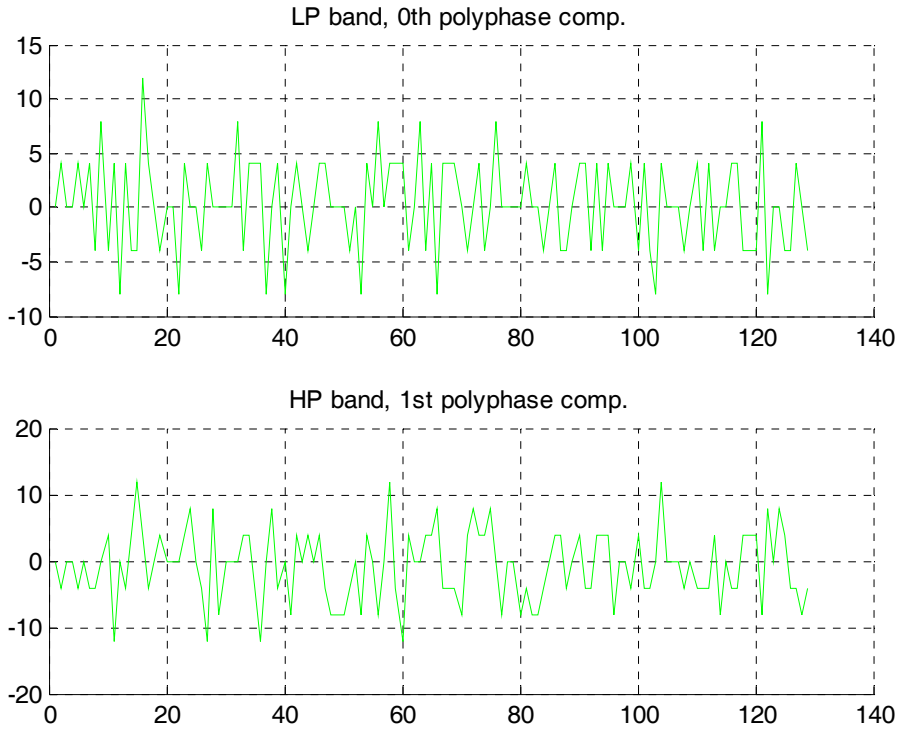
T_{00} = [0.5, 0.5];

T_{10} = [0.5, -0.5];

T_{01} = [-0.5, 0.5]; and

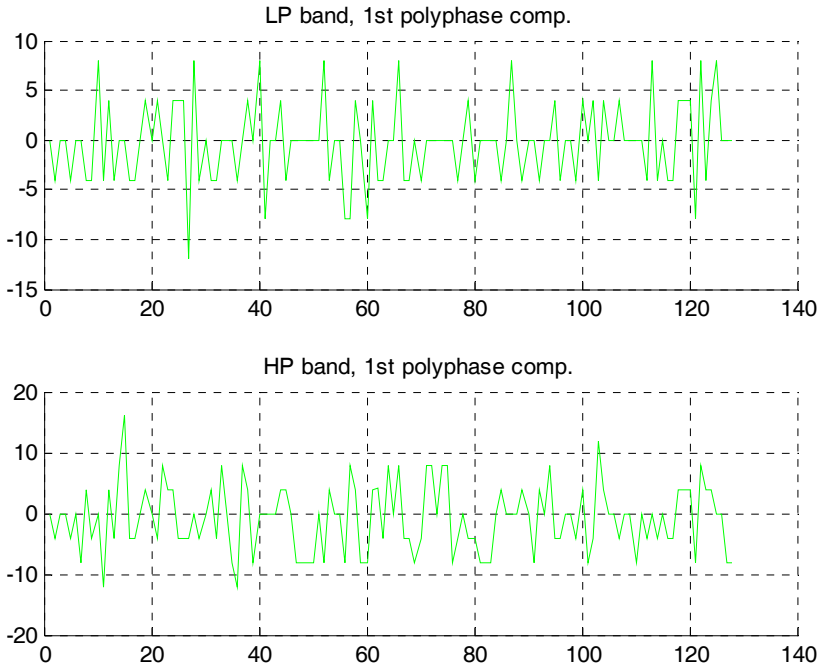
$T_{11} = [-0.5, -0.5]$. (4.5)

Figure 4.10 (a) and (b) show the difference between the quantized wavelet coefficients and the corresponding predicted values using the phase-shift matrix, for both the ODWT polyphase sets (green curve).



(a)

Figure 4.10 (a) [$y_{k,0}^{q}(n)$], error in quantized 0^{th} ODWT polyphase set



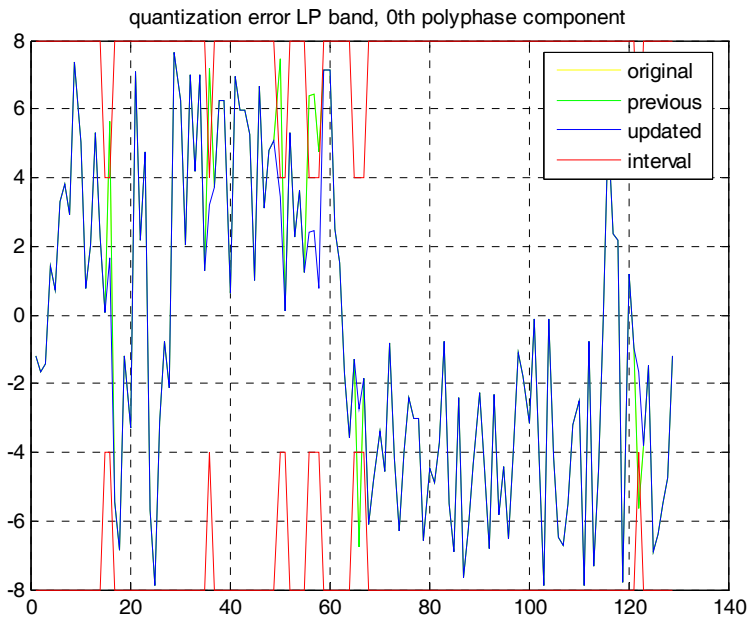
(b)

Figure 4.10 continued. (b) $[\, y_{k,1}^{q}(n)\,]$, error in quantized 1st ODWT polyphase set. (Shown as green curve)

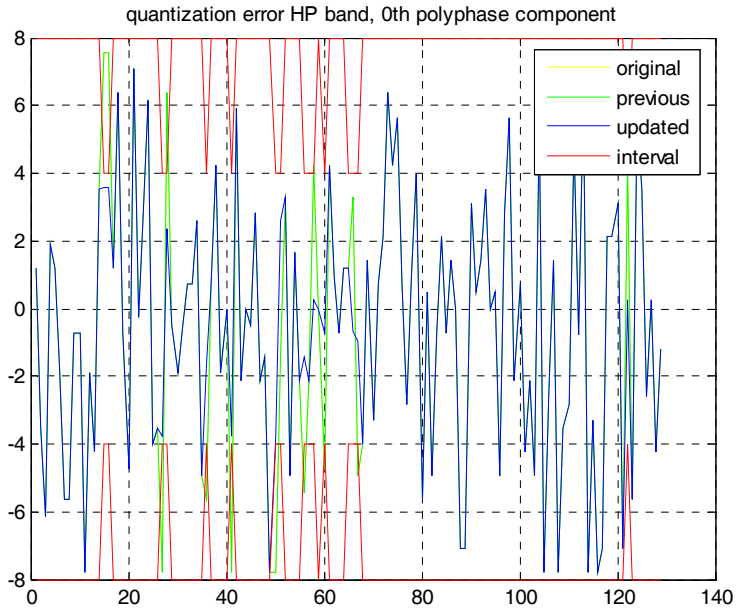
For the 0th polyphase set, lowpass case, only for the data point 16 the quantization error is larger than $\triangle q$. While in the highpass case, the significant data points are: [11, 15, 27, 36, 58, 60, and 104], since they have quantization errors larger than $\triangle q$.

For the 1st polyphase set, lowpass case, data point 27 has a large quantization error. For the highpass case, the points with significant errors are at [11, 15, 36, and 122].

Figure 4.11 (a-d) shows the output of the decoder after iteration 1. First, the 0th polyphase set, $[\, y_{k,0}^{q}(n)\,]$, is updated with the addition of 2 sub-bits and validation of all solutions. Then, a similar operation is carried out on the 1st polyphase component, $[\, y_{k,1}^{q}(n)\,]$.

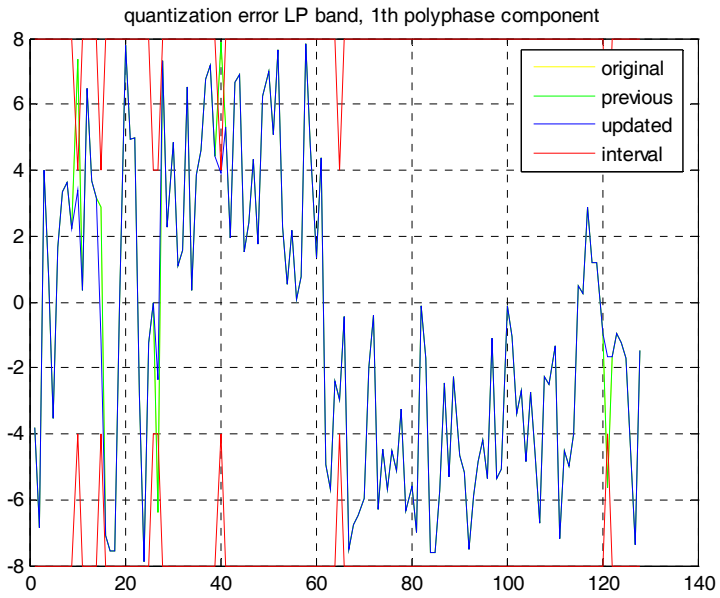


(a)

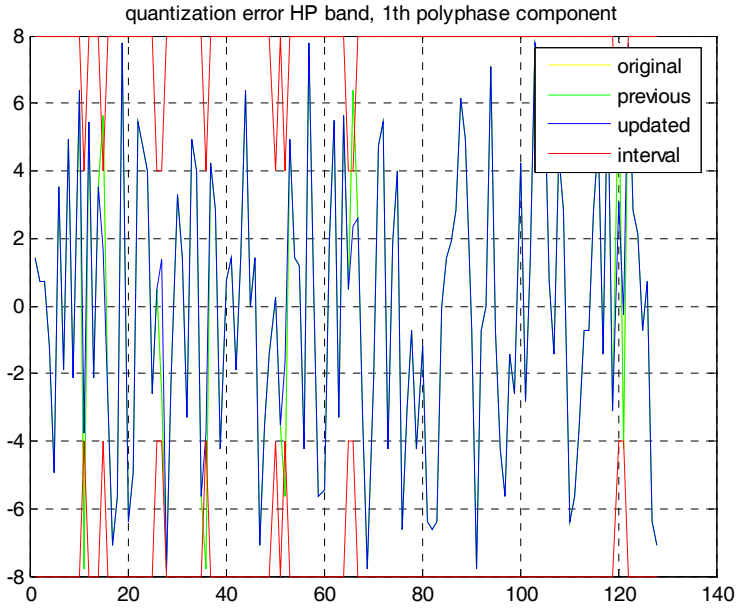


(b)

Figure 4.11 Quantization errors for all the 4 ODWT members after 1st iteration.



(c)



(d)

Figure 4.11 continued. Green curve is initial error value, blue curve is current and red curve is the error interval.
(a) 0^{th} Lowpass (b) 0^{th} Highpass (c) 1^{st} Lowpass (d) 1^{st} Highpass

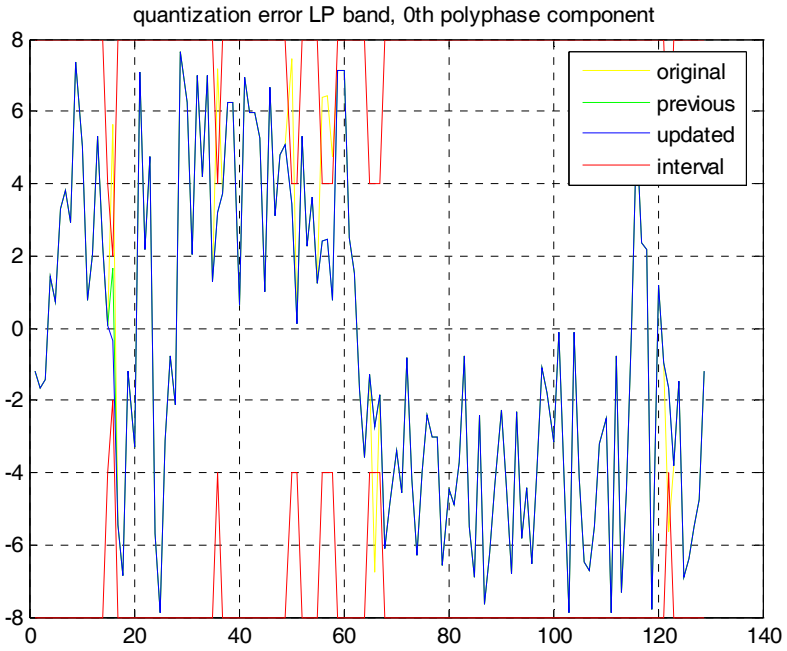
Figure 4.11 (a) shows data points [16, 36, 50, 57, 66 and 122] have a reduction in quantization error as observed from the comparison of the blue curve (current error value) to the green curve (initial error value). Note that in the region where the two plots overlap, only the blue line is seen. Data point 16 is of significance. With reference to figure 4.10 (a), for the lowpass case, the quantization error for the 16^{th} data point is greatly reduced from being over $\triangle q$ to under $\frac{1}{4}\triangle q$. The error range (red curve) is correspondingly reduced to between +/-4.

Figure 4.11 (b) shows data points [15, 16, 28, 27, 41, 50, 56, 60, 67 and 122] have a reduction in quantization error. With reference to figure 4.10 (a), for the highpass case, reduction in quantization error achieved for data points 15, 27 and 60 is noteworthy.

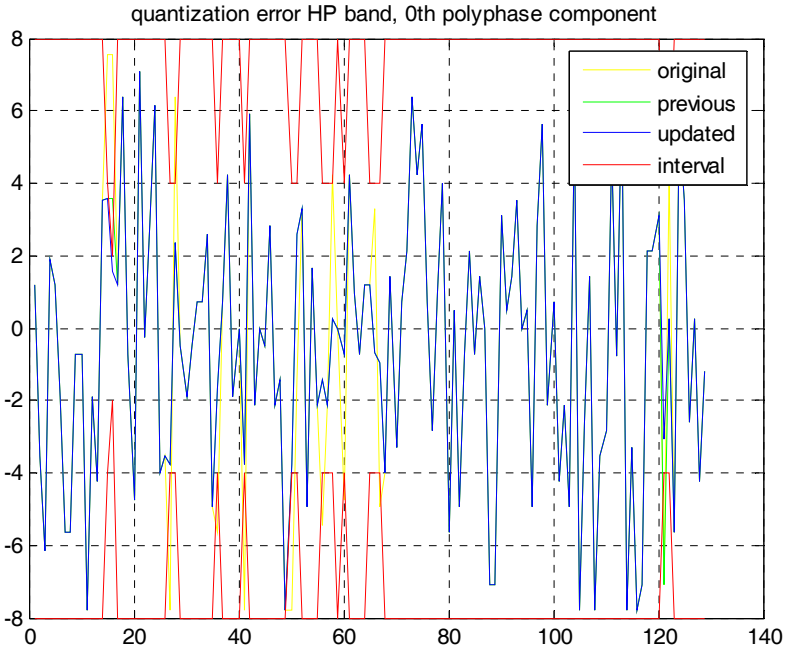
Figure 4.11 (c) shows data points [10, 15, 27, 40, and 121] have reduction in errors. Of these, data point 27 is noteworthy since the initial error was greater than $\triangle q$.

Figure 4.11 (d) shows data points [11, 15, 36, 52, 66, 120, and 121] have their quantization errors reduced. Of these, the most significant reduction is achieved for the data point 15, as the error reduces from twice the $\triangle q$ (figure 4.7 (b) high pass case) to one-fourth $\triangle q$.

Figure 4.12 (a-d) shows the output of decoder after iteration 2.

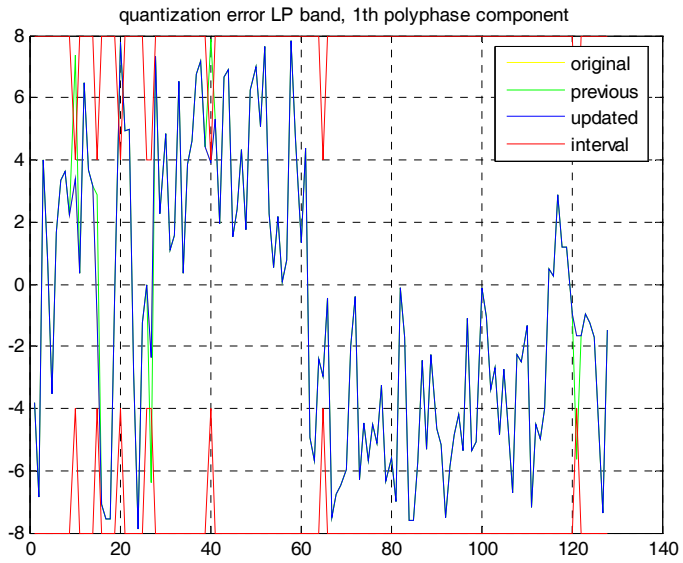


(a)

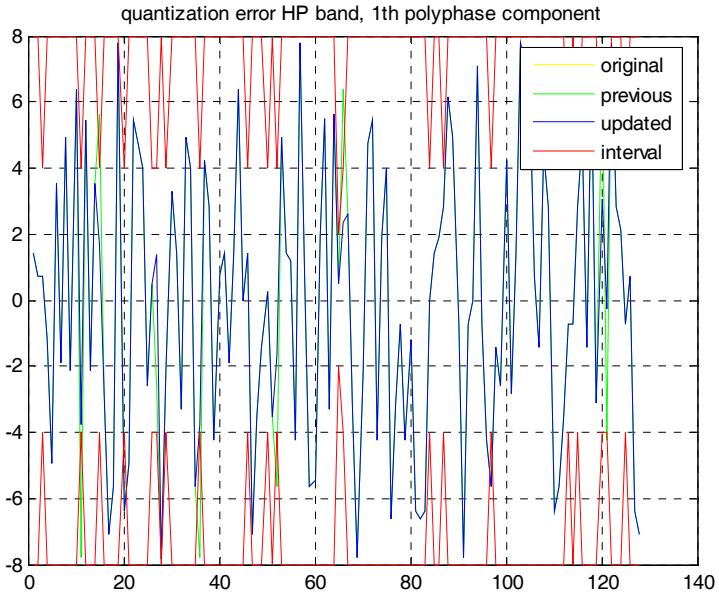


(b)

Figure 4.12 Quantization errors for all the 4 ODWT members after 2nd iteration



(c)



(d)

Figure 4.12 continued. Green curve is previous iteration error value, blue curve is current, yellow curve is original error value and red curve is the error interval.
(a) 0^{th} Lowpass (b) 0^{th} Highpass (c) 1^{st} Lowpass (d) 1^{st} Highpass

Figure 4.12 (a) shows only data point 16 undergoes further reduction in quantization error. In (b), data point 16 and 121 get refined. For the 1st polyphase set as shown in (c) and (d), there are no changes, and hence no green lines are observed, as they are overlapped by red lines.

No further improvements were found in any of the 4 ODWT polyphase sets after the 3rd iteration, and hence the process is stopped.

4.3 DB2 Wavelet Implementation

DB2 wavelets filters are of length four, and hence a memory depth of 3 wavelet coefficients is required for every sample of the polyphase set that is being estimated. A logical extension of equation 4.3 and 4.4 would provide the working equations for Daubechies family-DB2 DWT implementation. A quantizer of scheme 2 (figure 4.4) is used here. To begin, we add 1 sub-bit to every wavelet coefficient of the polyphase set that is being updated. Figure 4.13 shows the quantization error to be centered at 0 (zero) and maximum error limit to be 4 in either direction when the 1st sub-bit is added. The sub-bit states can be [-1 0 or +1]. The corresponding quantization error limits to which the estimated value corresponds when any one of these sub-bit states is added is also shown in the figure 4.13. For example, “0” state represents the error range of +/-2. If after the validation process, only solutions containing “0” state are kept, then the quantization error lies between +2 and -2; and the decoder would reduce the error range to +/-2 during the refinement process (see red curve in plots). If solutions containing “0” states and “-1” states are kept then it means that the quantization error lies between zero and +4. If the all of “0”, “-1” and “+1” states are kept, then it means the quantization error could lie anywhere in the entire range of +/-4, and the decoder cannot attain further refinement from what was already provided.

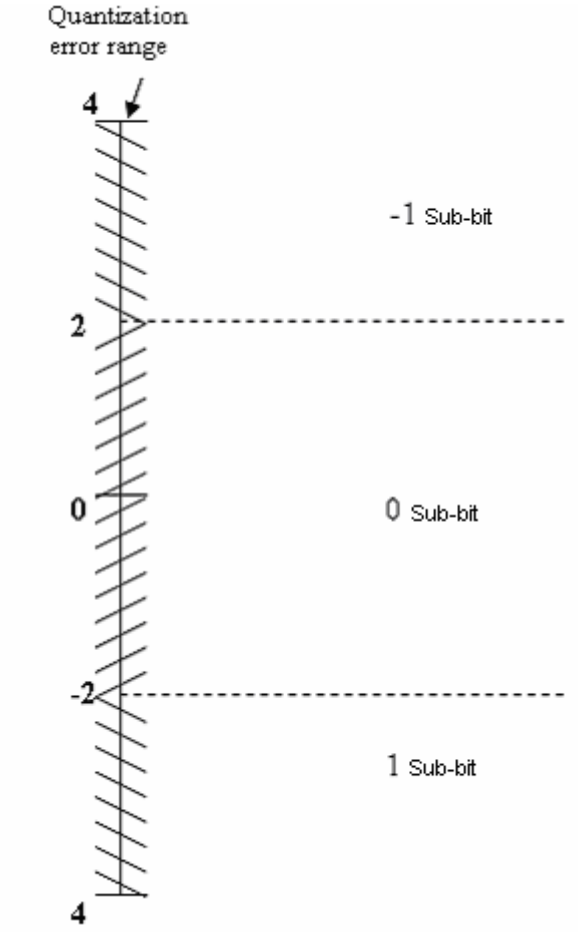


Figure 4.13 Quantization error range for 1st Sub-bit.

First, we reduce the quantization error on y_{1q} by estimating y_{0q}, and hence the 1st sub-bit is added to the y_{1q} set. After update, the decoder reduces the quantization error range (to +/-2) for specific data points during the refinement process. Then an identical set of operations is carried out for the y_{0q} polyphase set (note that the refined y_{1q} from the previous step is used here). The set of sub-bit solutions obtained for either polyphase set is observed. Only for those ODWT coefficients for which the sub-bit solutions kept were monotonic (i.e. only state "0" or "+1"), a second sub-bit can be added and the process repeated.

During the validation process where we look for valid solutions, the following assumption is made regarding the location of a particular wavelet coefficient estimate

($\hat{y}_1$) with respect to the maximum ($\hat{y}_{1\max}$) and minimum ($\hat{y}_{1\min}$) possible values it can take. This idea is illustrated in figure 4.14.

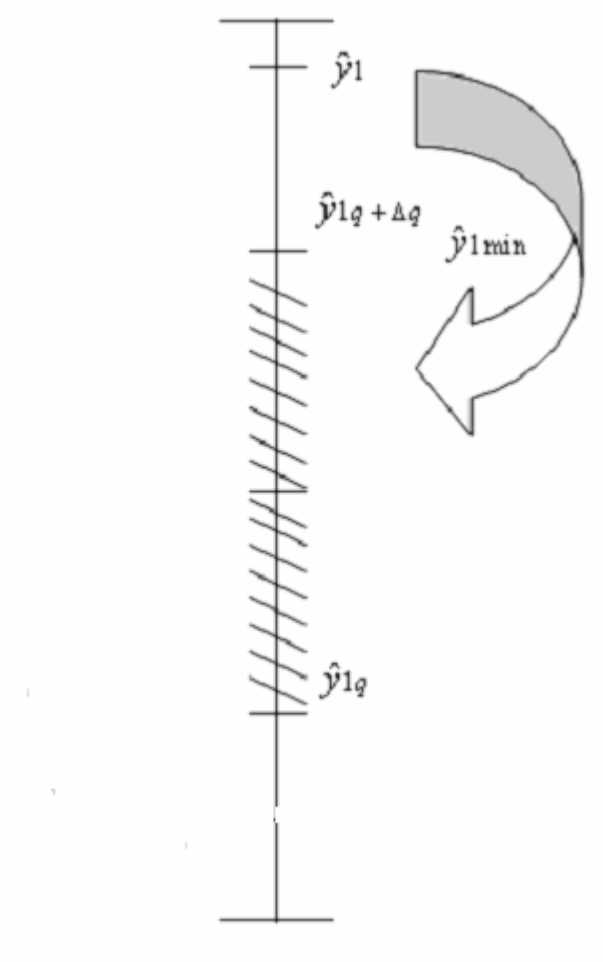


Figure 4.14 the location of a particular wavelet coefficient estimate.

The $\hat{y}_{1\max}$ and $\hat{y}_{1\min}$ values are calculated for every wavelet coefficient based on the quantization error limits set (initialized to $\triangle q$) for that data point. Then the estimated coefficient value $\hat{y}_1$ (after addition of the 1st sub-bit to its ODWT counterpart) is compared to see if it lies in the valid region shown *hatched* in figure 4.14. Any $\hat{y}_1$ is considered to be valid if ($\hat{y}_1+\hat{y}_{1\max}$) and ($\hat{y}_1+\hat{y}_{1\min}$) lie in the *hatched* region. Then the sub-bit combinations (including those of memory samples) are retained and others that did not satisfy this criterion are discarded. This method of validating was determined

after making the assumption that any given $\hat{y}_{1}$ will lie between its $\hat{y}_{1\max}$ and $\hat{y}_{1\min}$ values, as illustrated in figure 4.15.

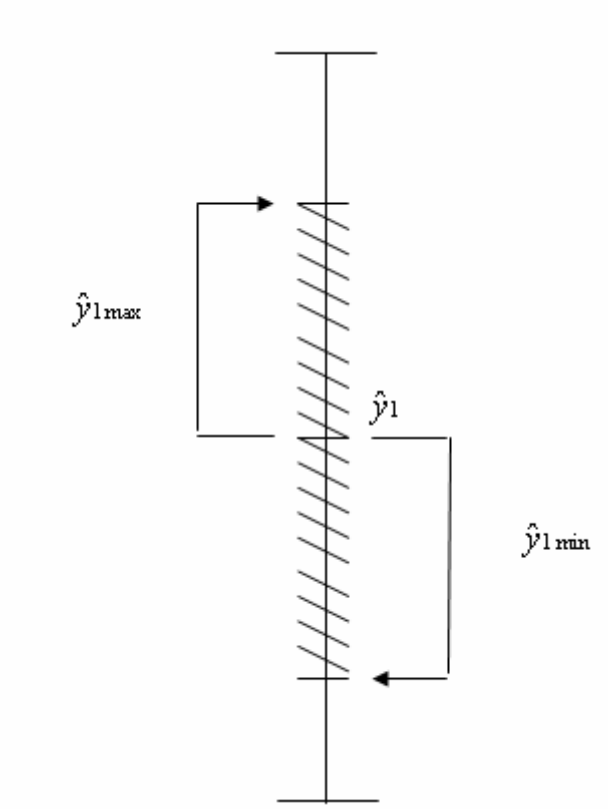


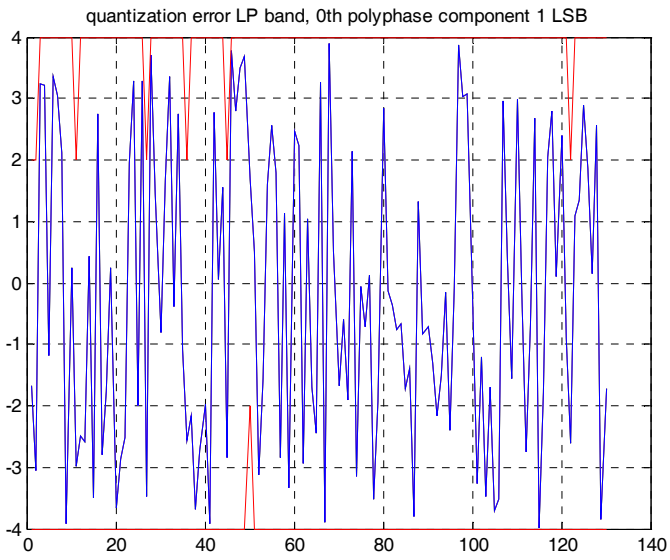
Figure 4.15 ODWT estimate lies within its maximum and minimum limits.

Experimental Results

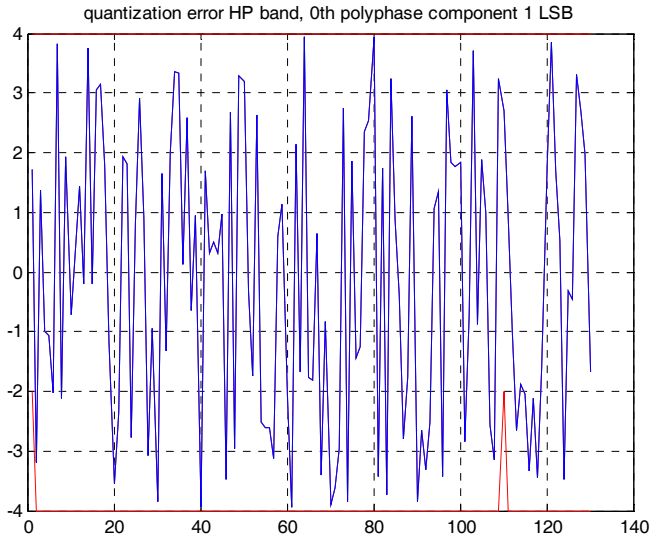
One-dimensional data input of length 256 samples is used (the same as that used for Haar implementation). The quantization step size, Δq=8. The phase-shift matrix, T, has coefficients for DB2 DWT as follows:

T_{00} = [-0.0625 0.5625 0.5625 -0.0625];
T_{10} = [0.0167 -0.2667 0.4833 -0.2333]; (4.6)
T_{01} = [-0.2333 0.4833 -0.2667 0.0167] and
T_{11} = [0.0625 -0.5625 -0.5625 0.0625].

Figure 4.16 (a-d) shows the output of the decoder after addition of the 1st sub-bit bit to the 1st polyphase component, $[\, y_{k,1}^{q}(n)\,]$. Then a similar operation is carried out on the 0th polyphase component, $[\, y_{k,0}^{q}(n)\,]$, and results are shown in Figure 4.17 (a-d). Color code for interpreting these plots is the same as the Haar wavelet implementation.

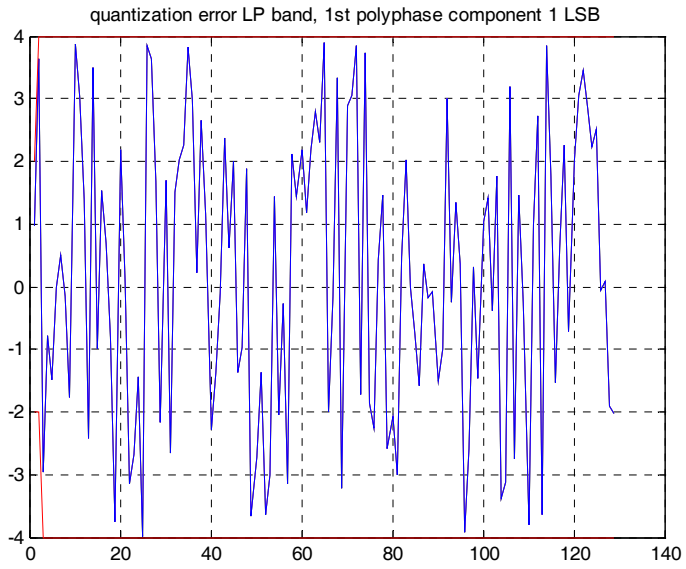


(a)

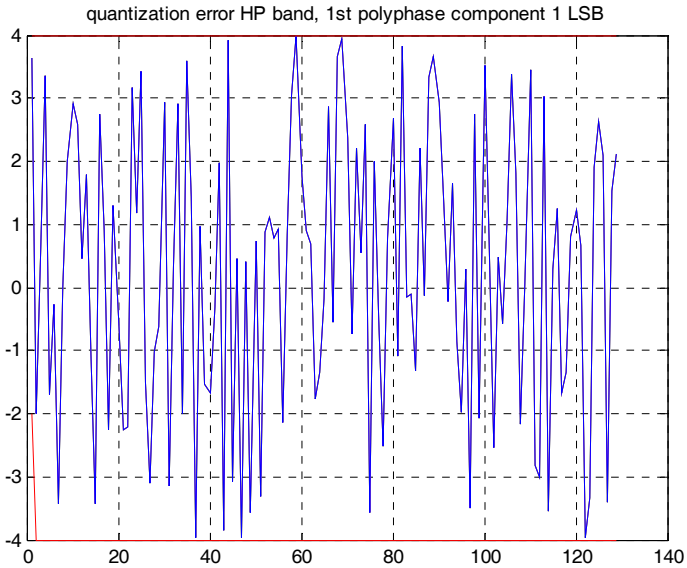


(b)

Figure 4.16 Quantization errors for all the 4 ODWT members after 1 sub-bit addition to $y_{k,1}^{q}(n)$.

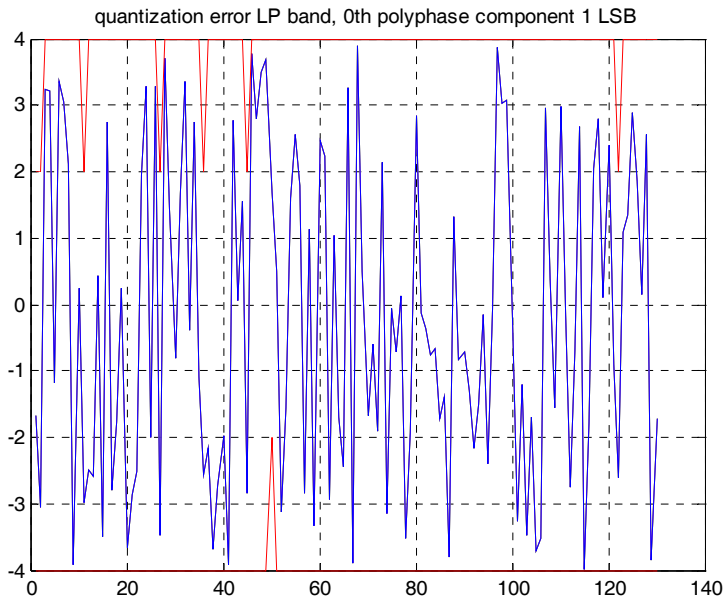


(c)

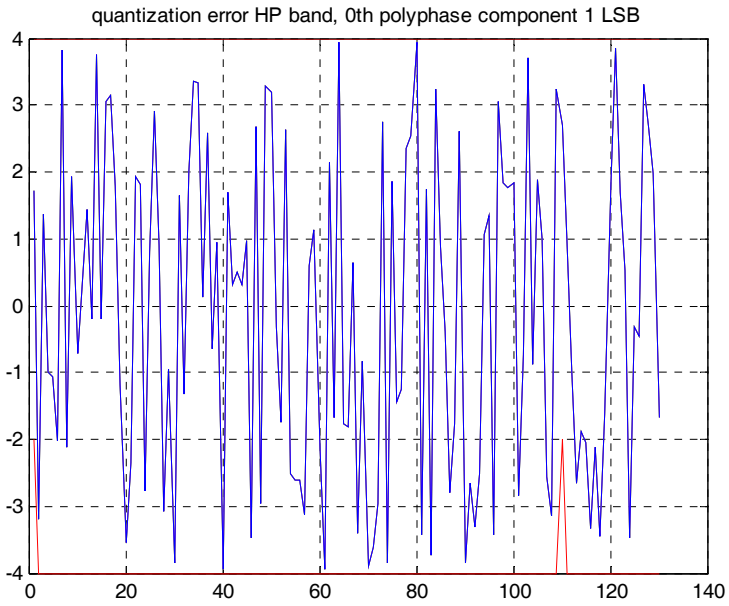


(d)

Figure 4.16 continued. Green curve is previous iteration error value, blue curve is current, and red curve is the error interval.
(a) 0^{th} Lowpass (b) 0^{th} Highpass (c) 1^{st} Lowpass (d) 1^{st} Highpass

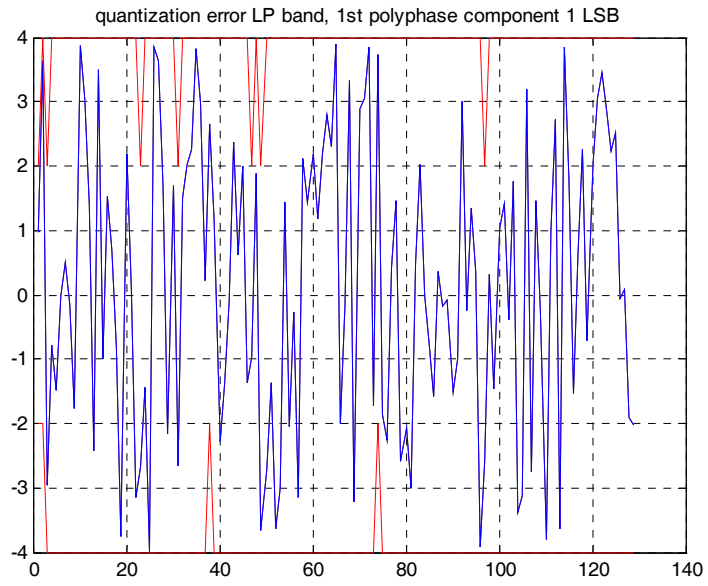


(a)

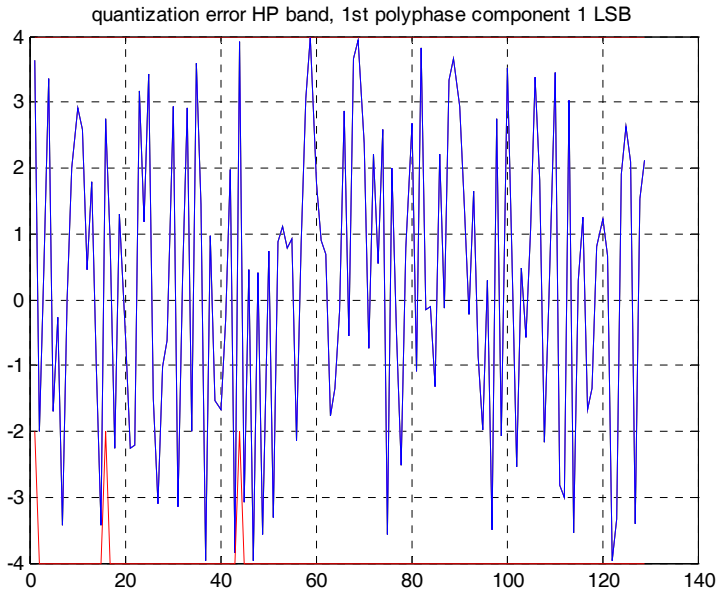


(b)

Figure 4.17 Quantization errors for all the 4 ODWT members after 1st sub-bit addition to $y_{k,0}^{q}(n)$.



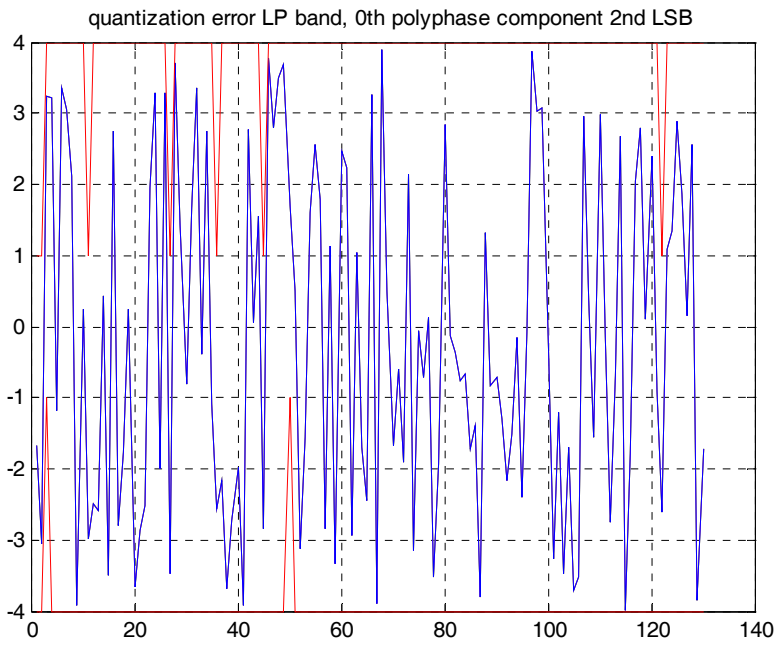
(c)



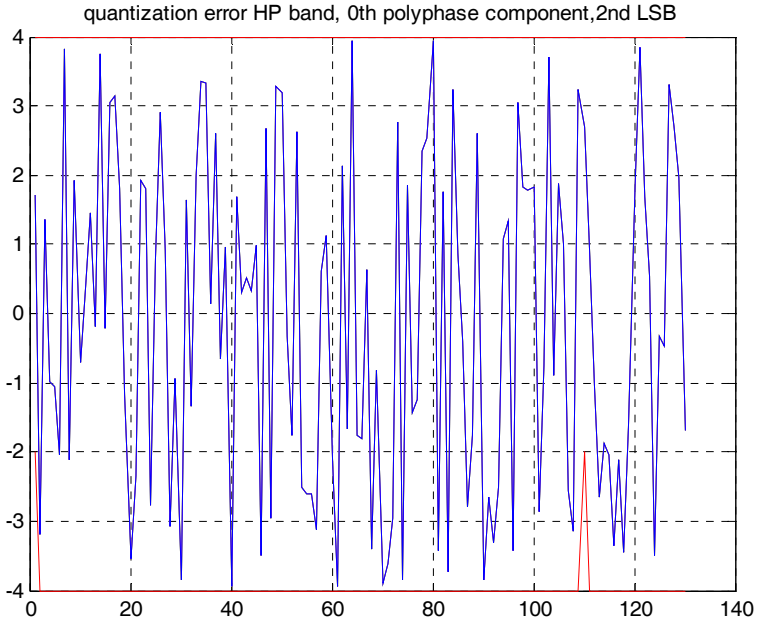
(d)

Figure 4.17 continued. Blue curve is current error value, and red curve is the error interval. (a) 0^{th} Lowpass (b) 0^{th} Highpass (c) 1^{st} Lowpass (d) 1^{st} Highpass

From these two figures, it is observed that the decoder refines the error limits in almost as many cases it did for the Haar wavelet implementation. Figure 4.16 (c) shows for the 1st data point (index zero on x axis of plot), the quantization error lies between +/-2 and is indicated by the reduction of error limit (red line) to +/-2, implying that only state "0" sub-bit solutions are kept. Hence, only to this wavelet coefficient we add a 2nd sub-bit, and later we update the y_{1q} value. Figure 4.18 (a-d) gives the plots after addition of the 2nd sub-bit.

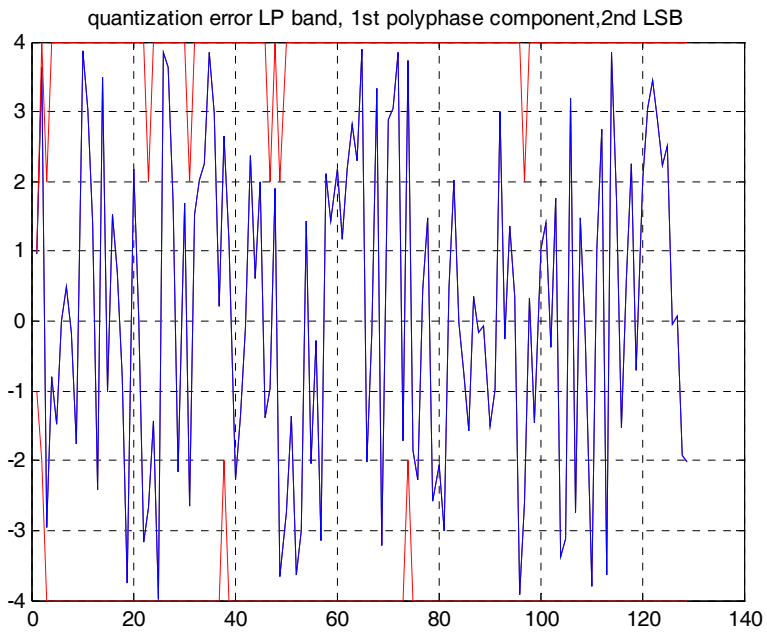


(a)

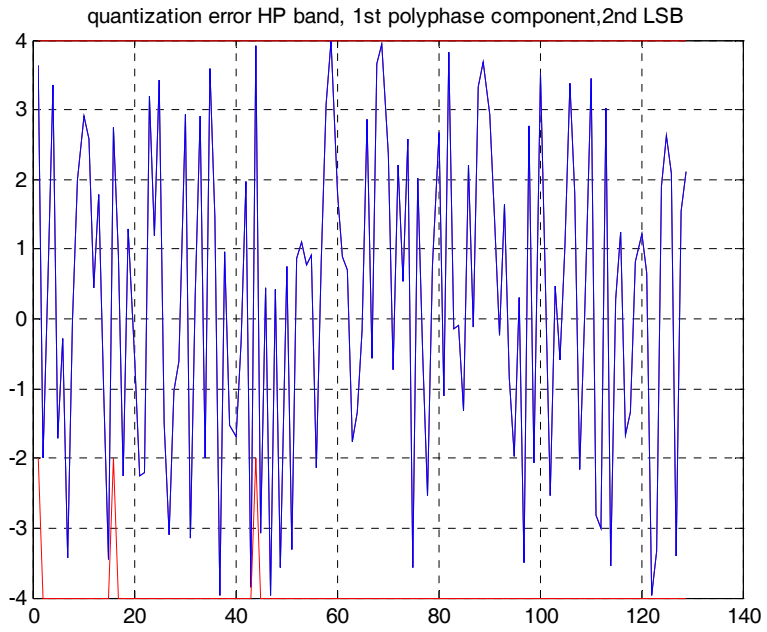


(b)

Figure 4.18 Quantization errors for all the 4 ODWT members after 2nd sub-bit addition to $y_{k,1}^{q}(1)$.



(c)



(d)

Figure 4.18 continued. Blue curve is current iteration error value, and red curve is the error interval. (a) 0^{th} Lowpass (b) 0^{th} Highpass (c) 1^{st} Lowpass (d) 1^{st} Highpass

Figure 4.18 (c) shows again for the 1st data point (index zero on x axis of plot), the error limit is reduced to +/- 1, indicating that a 3rd sub-bit can be added. Figure 4.19 gives the result for the 1st polyphase, lowpass, set after addition of a 3rd bit. Other plots are not shown as they remain unchanged.

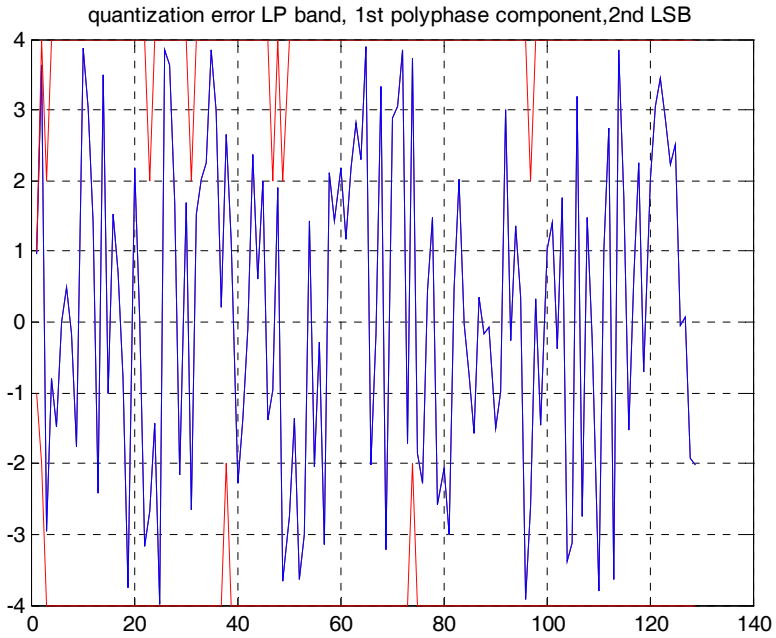


Figure 4.19 1st polyphase, lowpass, after addition of 3rd sub-bit.

It is observed that the error limit for the 1st data point remains between +/-1. This means that addition of a 3rd sub-bit could not further improve the quantization error.

Furthermore, from figure 4.18 (c), for the 2nd data point, the red line error limit shows that the quantization error is in the +2 to -4 range, indicating that only "+1" state sub-bit solutions are retained after 2nd bit processing. Similarly, from figure 4.18 (a) we see that the 2nd data point has quantization error in the -2 to +4 range, thereby indicating that only "-1" state sub-bit solutions are retained. To both these cases, we can add a 2nd sub-bit and iterate the complete ODWT sets.

It was found that addition of a 2nd sub-bit to 2nd data points of lowpass components of both polyphase sets (figure 4.15 (a) and (c)) did not produce any further refinements in the error limits (red line) and quantization error continued to remain

between [+2,-4] and [-2,+4], respectively. Furthermore, no useful changes were observed for subsequent polyphase components that would enable the decoder to add in the next additional LSB. The plots remain the same as figure 4.18.

CHAPTER V
CONCLUSION AND FUTURE WORK

Conclusions

The Overcomplete Discrete Wavelet Transform to reduce quantization error was found to be effective in Haar wavelet implementation. Coding and decoding schemes which used the Haar wavelet were developed and tested. The decoder could identify many data points (ODWT coefficients) with high quantization errors and reduce them to a good extent. In some cases, the error was reduced to one-fourth of the original value.

The quantization scheme and the decoder developed for the DB2 wavelet implementation was not effective in reducing the quantization errors. The reason for this is that the decision criteria used in the decoder could not successfully identify data points for which the ODWT coefficient value could be updated and result in a reduction of quantization errors. Although, the decoder did find some data points for which the error range could be reduced this was not true for the error itself. Furthermore, only in few cases a second and third sub-bits could be added and even that did not help in reducing the quantization errors.

The results obtained from Haar wavelet-based ODWT coding scheme is encouraging. Decoders that use multiple description-coding concepts would find the ODWT method of enhancing signal quality attractive, as it makes optimal use of the additional/redundant descriptions. The tradeoff in eliminating the downsampling operation when converting the signal to transform domain is well compensated by its benefits.

Future Work

DB (9, 7) is the wavelet used in popular signal compression algorithms. But we need to develop a better algorithm and a decision criterion that would work effectively for the DB2 wavelet-based ODWT before it can be adapted for the DB (9, 7) wavelet-based ODWT. This would make the ODWT concept feasible for signal compression applications. Furthermore, the asymmetric delays and phase shifts that would occur in the

ODWT sets due to the unequal lengths of analysis and synthesis filters in the DB (9, 7) wavelet, needs to be investigated.

Incorporation of ODWT into a Multiple Description Coding scheme and measurement of its performance with respect to signal quality and quantization error resilience need to be performed in future work.

REFERENCES

[1] Martin Vetterli, Jelena Kovacevic , *Wavelets and Subband Coding*, Prentice Hall PTR, 1995.

[2] Rafael C Gonzalez, Richard E Woods, *Digital Image Processing*, Pearson Education, 2nd Edition, 2003.

[3] V. K Goyal, "Compression Meets the Network", IEEE Signal Processing Magazine, September 2001.

[4] Jiangling Guo, "A Fast and Low Complexity Image Codec based on Backward Coding of Wavelet Trees", Doctoral Dissertation, Texas Tech University, October 2005.

[5] G Strang, T Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.

[6] Agostino Abbate, Casimer M.DeCusatis, Pankaj K.Das , *Wavelets and Subbands*, Birkhauser, 2002.

[7] Robi Polikar, "The engineers' ultimate guide to wavelet analysis-the wavelet tutorial", http://users.rowan.edu/~polikar/WAVELETS /WTtutorial.html.

[8] Andrew P.Bradley, "Shift-invariance in the Discrete Wavelet Transform", *Digital Image Computing: Techniques and Applications* (DICTA'03), pp 29-38, Sydney, Australia, December 2003.

[9] Jerome M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients.*IEEE Trans. on Signal Processing*, 41(12):3445–3462, December 1993.

[10] Andrea Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.

[11] G.Kabatiansky,E. Krouk ,S.Semenov,*Error Correcting Coding and Security For Data Networks*, John Wiley & Sons, Ltd, 2005.

[12] James E Fowler, "The Redundant Discrete Wavelet Transform and Additive Noise", *IEEE Signal Processing Letters*, Vol 12, No. 9, September 2005.

[13] Alan V. Oppenheim,Ronald W. Schafer with John R.Buck,*Discrete -Time Signal Processing*,Pearson Education,Second Edition,2003.

[14] Xin Li, "New Results of Phase Shifting in the Wavelet Space", *IEEE Signal Processing Letters*, Vol 10, No. 7, July 2003.

[15] The Math works Inc.,MATLAB®,Release 13.

APPENDIX

1. Mat Lab 7.0 code for generating the input signal

```
% create AR-1 lowpass signal
randn('state',5);
x=randn(1,256);
x=filter(0.05,[1 -0.95], x);
% round to finite resolution (optional, but makes it more similar to
% images)
x=round(256*x);
```

PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Texas Tech University or Texas Tech University Health Sciences Center, I agree that the Library and my major department shall make it freely available for research purposes. Permission to copy this thesis for scholarly purposes may be granted by the Director of the Library or my major professor. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my further written permission and that any user may be liable for copyright infringement.

Agree (Permission is granted.)

Narsimha Bharat Kotikanyadanam — 11/30/05
Student Signature — Date

Disagree (Permission is not granted.)

_____ _____
Student Signature — Date