Université d'Ottawa • University of Ottawa

Image Coding for Transmission over Wireless CDMA Channels

Xiaodong Huang

A thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of
Master of Applied Science
in Electrical Engineering

April, 2002

Ottawa-Carleton Institute of Electrical and Computer Engineering
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada

Abstract

This thesis describes an image coding scheme based on the wavelet transform and lattice vector quantization (LVQ) for transmission over CDMA channels. For the purpose of providing reliable transmission, a joint source-channel coding method — rate compatible convolutional coding (RCPC) — is used for channels which have Rayleigh fading effects to protect the coded bitstream from the noise of CDMA channels. Three different kinds of wavelets — Daubechies, biorthogonal and B-spline wavelets — were used for the source coding of images, and the statistics of subband coefficients of these wavelet transformed images was analysed. For the CDMA channel model, multipath Rayleigh fading effects and the multiple access interference (MAI) were considered, and two different receivers were used to see their performance in handling those different types of interference. One is the Rake receiver used together with RCPC for channels having Rayleigh fading effects as well as MAI. Another is a recursive-least-square (RLS) minimum-mean-square-error (MMSE) receiver, which is based on the theory of blind multiuser detection, to be used for the cases of fixed CDMA wireless applications which do not have Rayleigh fading effects but still have MAI. The overall performance of the coding scheme based on Daubechies, biorthogonal and B-spline wavelets respectively is also shown .

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor Dr. Eric Dubois, for his willingness to support my study and research, as well as his patience and invaluable guidance that helped me finish this thesis.

I also want to thank the working colleagues of VIVA Lab who gave me various help.

Last but not least, my thanks to my family who gave me precious moral support.

Xiaodong Huang

Ottawa, Canada, April 2002

Contents

List of Figures

List of Tables

List of Acronyms

CDMA	Code Division Multiple Access
AWGN	Additive White Gaussian Noise
DCT	Discrete Cosine Transform
PSNR	Peak Signal to Noise Ratio
MAI	Multiple Access Interference
BPSK	Binary Phase Shift Keying
RCPC	Rate Compatible Punctured Convolutional Code
SMI	Self Multipath Interference
MMSE	Minimum Mean Square Error
EZW	Embedded Zerotree Wavelet
SPIHT	Set Partitioning in Hierarchical Trees
PR	Perfect Reconstruction
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
LVQ	Lattice Vector Quantization

RLS Recursive Least-Squares Algorithm

PN Pseudo-Noise

Chapter 1

Introduction

With the rapid development of wireless communications, there are more and more plans to transmit multimedia content (image/video) over wireless channels. Typical wireless channels are noisy and of narrow bandwidth. Usually there is a 9.6kbps bandwidth for a customer using a digital wireless device like a code-division multiple-access (CDMA) cell phone. Even if the bandwidth increases up to 2Mbps for the coming 3G wireless, it is still not comparable to the bandwidth of broadband optical communication systems like ATM, which could allocate dozens of Mbps to end users. Thus for wireless multimedia applications, higher compression is required for both image and video signals.

For image compression, the JPEG standard has already been widely used. It uses discrete cosine transform (DCT) and Huffman coding techniques. Its main disadvantage is that when the coded bit rate is lower than a certain value (about 0.25 bits/pixel), there are blocking effects in the decoded image, due to the 8×8 block two-dimensional (2-D) DCT. Also, if the image is directly transmitted over noisy channels — which is usually the case for wireless applications — it is easy to lose blocks, because Huffman coding is a variable-length-code (VLC). The noisier the channel is, the more blocks are lost. Recently, the wavelet decomposition has been proved to be a better tool for image compression. Especially for very high compression ratios, it performs better than DCT based JPEG. Thus, the new JPEG2000 standards adopt wavelet subband coding.

Fig. 1.1 shows a generic block diagram for transmitting images through CDMA channels.

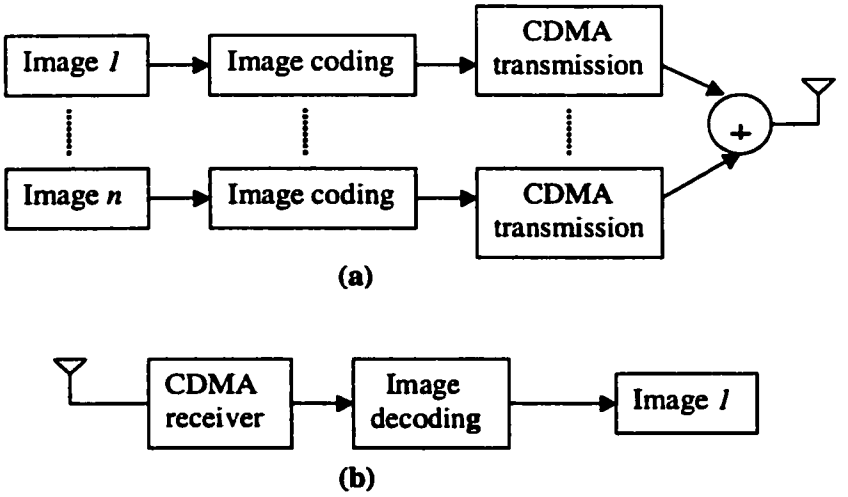

Figure 1.1: Block diagram of an image communication system using CDMA: (a)tramsmitter; (b)receiver

Part (a) of the figure shows the block structure for transmission, in which different images from different users are coded separately and then transmitted using the same bandwidth of the CDMA channel. Part (b) of the figure shows the block structure of a receiver in which the coded image signal for a particular user is detected from the mixture of signals of all the users using CDMA receiving techniques and then restored as an image for the user.

For a long time, researchers have been studying how to compress and transmit an image over wireless channels [2] - [12]. Among them, most did not specify the features of wireless channels, such as if the channel is CDMA or not, noise characteristics, etc.. Aïssa and Dubois [9] use CDMA as wireless channel model, however, a key factor in CDMA systems — multiple access interference (MAI) which is caused by the fact that all the users use the same wireless bandwidth — is modeled as AWGN, which is true when the number of mobile users is low. MAI is typical in CDMA uplink channels. When the number of users is high, MAI will become impulsive in nature. MAI is an important factor that affects the capacity of CDMA systems. Iun and Khandani [10] also use CDMA wireless channel model, but the related image coding scheme is based on 2-D DCT. Among other papers, most image coders are based on EZW[13] or SPIHT[14] using channel protection for bitstreams of different

importance. For example, in [2] and [3], a concatenated channel coding scheme was applied to the SPIHT. In [4], an effective coding procedure that builds upon the method in [2] and [3] was developed for transmission channels that allow feedback. In [5] and [12], a class of modified EZW algorithms was presented which limits error propagation by reducing the amount of variable-length codes in the transmitted bitstream. Only [6] provides a different coding scheme using lattice vector quantization (LVQ) for subband coefficients. This coding scheme can provide more fixed-length bitstreams than SPIHT or EZW, which is crucial for the decoder when the channel is very noisy.

1.1 Thesis Objectives

The objectives of this thesis are: (1) to develop a wavelet based image coding scheme which utilizes LVQ, so that most of the coded bit streams could have fixed length; (2) to investigate the energy concentration properties of Daubechies, biorthogonal and spline wavelets by analyzing their transformed coefficients of different subbands; (3) to combine source and channel coding together by using rate compatible punctured convolutional coding (RCPC) and assigning different protection to different bit streams with different importance; (4) to present the performance of image coder using different wavelets (Daubechies, biorthogonal and spline wavelets) under a CDMA channel model which involves MAI, and using standard CDMA Rake receiver as well as an advanced adaptive receiver, in order to find out which wavelets perform best in a noisy CDMA channel.

1.2 Thesis Organization

In this chapter we give a brief introduction of the thesis contents, objectives and organization.

Chapter 2 provides basic concepts of image coding with wavelets as well as some state-of-the-art image coders like EZW and SPIHT, the principles of channel coding, especially convolutional coding and RCPC, and the main components of CDMA systems with emphasis on spreading/despreading and MAI.

In Chapter 3, the following are included: (1) The features of Daubechies, biorthogonal and spline wavelets. In particular, we emphasize the principle of high order (n=3) B-spline and cardinal B-spline wavelets, their properties and implementation details. (2) Quantization and coding methods used in this thesis which utilize LVQ. (3) Some simulation results for these wavelets in noiseless condition, i.e. without channel coding.

Chapter 4 focuses on the channel model and receiving techniques of CDMA systems. Two different channel models for Rake receiver and adaptive receiver as well as their different signal detection principles will be shown respectively. Some simulation results based on 1-D signals will also be presented for the two receivers.

Chapter 5 presents the final simulation results of transmitting coded images through CDMA channels using RCPC and with different receiving techniques. The performance of Daubechies, biorthogonal and spline wavelets under different channel conditions and receiving techniques will be supplied.

In Chapter 6, we summarize the conclusions derived from this work, and suggest some topics for future work.

Chapter 2

Image Coding, Channel Coding and CDMA Systems

In this chapter, we first present the basic concepts of wavelets and filter banks and their application in image coding. The discrete wavelet transform (DWT) techniques that will be applied on images in this thesis are based on these concepts. Some existing compression methods based on the wavelet filter-bank will then follow with emphasis on *embedded zerotree wavelet* (EZW) algorithm as well as *set partitioning in hierarchical trees* (SPIHT) algorithm. By introducing these two well-known algorithms we could show why they are susceptible to channel noise and the rationale for some other compression methods which are more robust under the noise of wireless channels. Then the principle of convolutional coding will be shown plus the basic idea of a joint source-channel coding, namely, rate compatible punctured convolutional coding (RCPC). RCPC will be used in this thesis for adding different redundant bits to different parts of the encoded bitstream for channel protection. Finally the CDMA system will be introduced with the focus on its channel model and spreading/despreading schemes. By showing these schemes we could explain why there is *multiple access interference* (MAI) in the CDMA system and how the MAI will be induced in this thesis.

2.1 Wavelets and Subband Coding

Wavelets are functions generated from one single function ψ by dilations and translations

$$\psi^{a,b}(t) = a^{-1/2}\psi\left(\frac{t-b}{a}\right). \tag{2.1}$$

The *mother wavelet* ψ has to satisfy $\int \psi(x)dx = 0$, which implies at least some oscillations. The definition of wavelets as dilates of one function means that high frequency wavelets correspond to $a < 1$ or narrow width, while low frequency wavelets have $a > 1$ or wider width.

The basic idea of the wavelet transform is to represent any arbitrary function f as a superposition of wavelets. For engineering purposes, because we are dealing with discrete signals, in most cases we only use discrete wavelet transform (DWT). In general, the DWT is generated by samplings of a corresponding continuous wavelet transform.

In a multiresolution analysis, one really has *two* functions: the mother wavelet ψ and a *scaling function* ϕ. The transform with ϕ results in coarse information, while the transform with ψ results in detail information. In the DWT, usually the scaling function ϕ corresponds to a lowpass filter, while the wavelet ψ corresponds to a highpass filter. Besides the analysis, we also need to reconstruct the original signals and this requires another pair of synthesis filters. Mallat [15] invented a mathematical tool which is particularly well adapted to the use of wavelet bases in image analysis, and which gives rise to a fast computational algorithm.

Fig. 2.1 depicts a basic two-channel filter bank structure, where $H(z)$ and $G(z)$ are lowpass and highpass analysis (decomposition) filters, and $\tilde{H}(z)$ and $\tilde{G}(z)$ are lowpass and highpass synthesis (reconstruction) filters. The input signal is filtered by $H(z)$ and $G(z)$ respectively and both filter outputs are downsampled by two so that the total number of samples in the output is the same as in the input. Conversely, these output samples are filtered by $\tilde{H}(z)$ and $\tilde{G}(z)$ after being upsampled by two respectively, and the filter outputs from $\tilde{H}(z)$ and $\tilde{G}(z)$ are summed together to get a final output signal. The choice of $H(z), G(z), \tilde{H}(z)$ and $\tilde{G}(z)$ depends on the features of the wavelet basis, i.e. orthonormal, biorthogonal or spline. Most of the orthonormal wavelet bases have infinitely supported ψ, corresponding to filters h and g with infinitely many taps. The construction in [16] gives ψ with finite

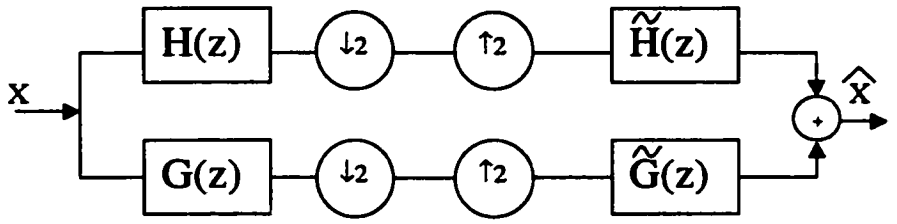

Figure 2.1: Basic structure of a two-channel filter bank

support, and therefore, corresponds to FIR filters. It follows that the orthonormal bases in [16] correspond to a subband coding scheme with exact reconstruction property, using the same FIR filters for reconstruction as for decomposition. We will give detailed information about these different kinds of wavelet filters in Chapter 3.

From the theory of multirate analysis, in the z-transform domain, we have at the output of Fig. 2.1:

$$\widehat{X}(z) = \frac{1}{2}[H(z)\tilde{H}(z) + G(z)\tilde{G}(z)]X(z) + \frac{1}{2}[H(-z)\tilde{H}(z) + G(-z)\tilde{G}(z)]X(-z). \qquad (2.2)$$

In order to obtain perfect reconstruction (PR), $X(z) = \widehat{X}(z)$, we need the following conditions to be satisfied [17]:

$$\begin{aligned} H(z)\tilde{H}(z) + G(z)\tilde{G}(z) &= 2 \\ H(-z)\tilde{H}(z) + G(-z)\tilde{G}(z) &= 0 \end{aligned} \qquad (2.3)$$

Whichever kind of wavelet basis, it has to satisfy (2.3) for perfect reconstruction.

For the above analysis-synthesis structure, if the filters are one-dimensional, then for a two-dimensional image signal, we need to treat the rows and columns of pixels separately. Fig. 2.2 and Fig. 2.3 show one stage in an image decomposition and reconstruction respectively. The four subband images after the decomposition can be labeled as LL, LH, HL and HH for the branch coming through filters of $H(z_1)H(z_2), H(z_1)G(z_2), G(z_1)H(z_2)$ and $G(z_1)G(z_2)$ respectively. Usually we submit subband LL to more stages of decomposition, resulting in subbands of LLL, LLH, LHL and LHH, and so on. The compression can be achieved by making use of different energy concentration in different subbands. The energy of an image is mostly concentrated in the lowest subband.

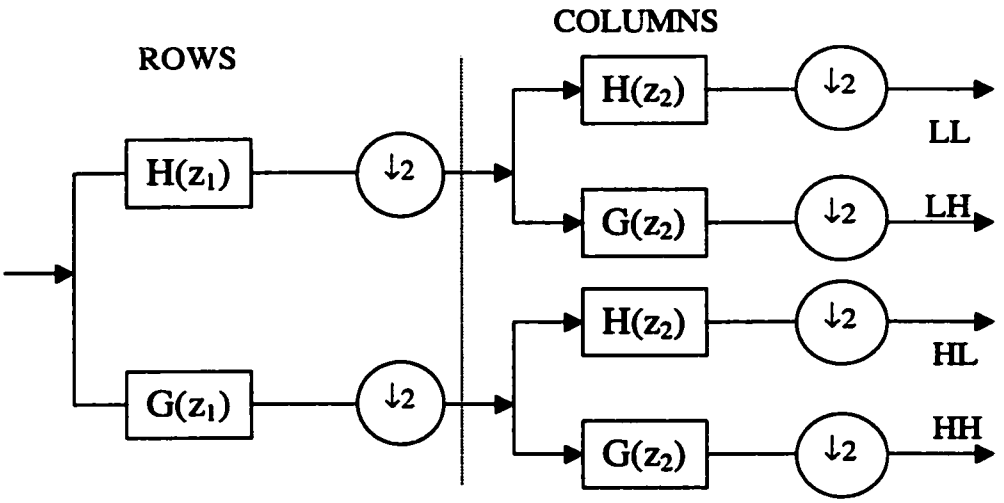

Figure 2.2: One stage in a multiscale image decomposition

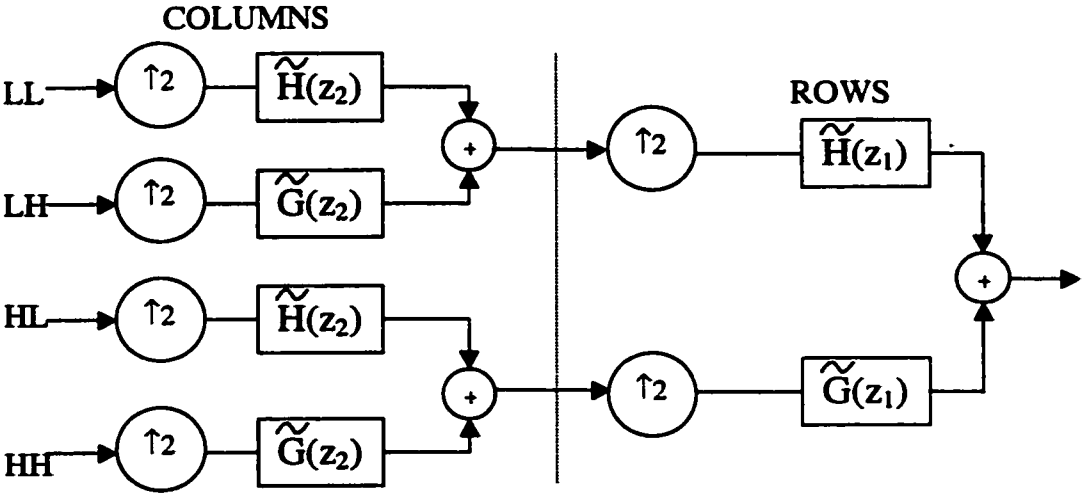

Figure 2.3: One stage in a multiscale image reconstruction

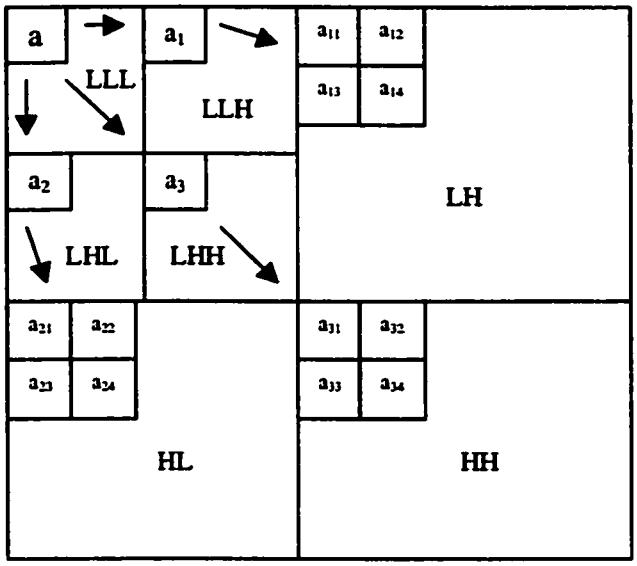

Figure 2.4: A 7-band wavelet decomposition

2.1.1 Embedded Zerotree Coder

The embedded zerotree wavelet (EZW) coder was introduced by Shapiro [13]. It is a quantization and coding strategy that incorporates some characteristics of the wavelet decomposition. The particular characteristic used by the EZW algorithm is that there are wavelet coefficients in different subbands that represent the same spatial location in the image. If the decomposition is such that the size of the different subbands is different, then a single coefficient in the smaller subband may represent the same spatial location as multiple coefficients in the other subbands.

Let's consider the 7-band decomposition shown in Fig. 2.4. The coefficient a in the upper-left corner of band LLL represents the same spatial location as coefficients a_1 in band LLH, a_2 in band LHL, and a_3 in band LHH. In turn, the coefficient a_1 represents the same spatial location as coefficients a_{11}, a_{12}, a_{13}, and a_{14} in band LH, and so on. In fact, we can visualize the relationships of these coefficients in the form of a tree. The coefficient a forms the root of the tree with three descendants a_1, a_2, and a_3. The coefficient a_1 has descendants a_{11}, a_{12}, a_{13}, and a_{14}. The coefficient a_2 has descendants a_{21}, a_{22}, a_{23}, and a_{24} and the coefficient a_3 has

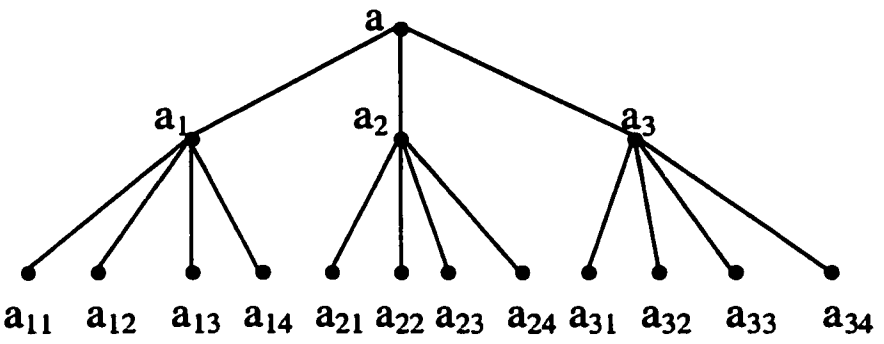

Figure 2.5: Data structure used in the EZW coder

descendants a_{31}, a_{32}, a_{33}, and a_{34}. A pictorial representation of the tree is shown in Fig. 2.5.

Due to the fact that when natural images are decomposed in this manner most of the energy is compacted into the lower bands, in many cases the coefficients closer to the root of the tree have higher magnitudes than coefficients further away from the root. This means that often if a coefficient has a magnitude less than a given threshold, all its descendants will have magnitudes less than that threshold; this property can be exploited for the purpose of compression.

Before we describe the EZW algorithm, we need to introduce some terminology. Given a threshold T, if a given coefficient has a magnitude greater than T, it is called a *significant* coefficient at level T. If the magnitude of the coefficient is less than T (it is insignificant), and all its descendants have magnitudes less than T, then the coefficient is called a *zerotree root.* Finally, it might happen that the coefficient itself is less than T but some of its descendants have a value greater than T. Such a coefficient is called an *isolated zero.*

The EZW algorithm is a multiple-pass algorithm, with each pass consisting of two steps: *significance map encoding* or the *dominant pass*, and *refinement* or the *subordinate pass.* If c_{max} is the value of the largest coefficient, the initial value of the threshold T_0 is given by

$$T_0 = 2^{\lfloor \log_2 c_{max} \rfloor} \tag{2.4}$$

In each pass, the threshold T_i is reduced to half the value it had in the previous pass:

$$T_i = \frac{1}{2} T_{i-1} \tag{2.5}$$

For a given value of T_i, we assign one of four possible labels to the coefficients: *significant positive (sp), significant negative (sn), zerotree root (zr)* and *isolated zero (iz)*. If we used a fixed-length code, we would need 2 bits to represent each of the labels. When a coefficient has been labeled a zerotree root, we do not need to label its descendants. This assignment is referred to as *significance map coding*. The coefficients labeled *significant* are assigned an initial reconstruction value of $1.5T_i$ or $-1.5T_i$, depending on whether the coefficient is positive or negative. The selection of T_i according to (2.4) and (2.5) guarantees that the significant coefficients will lie in the interval $[T, 2T)$. Once a determination of significance has been made, the significant coefficients are included in a list for further refinement in the refinement or subordinate passes. In the refinement pass, we determine whether the coefficient lies in the upper or lower half of the interval $[T, 2T)$. In successive refinement passes, as the value of T is reduced, the interval containing the significant coefficient is narrowed still further and the reconstruction is updated accordingly. An easy way to perform the refinement is to take the difference between the coefficient value and its reconstruction and quantize it using a two-level quantizer with reconstruction values $\pm T/4$. This quantized value is then added to the current reconstruction value as a correction term.

The wavelet coefficients that have not been previously determined to be significant are scanned in a manner similar to the above. This process can be repeated and the threshold reduced accordingly until the bit budget is used up or until some other criterion is satisfied.

2.1.2 Set Partitioning in Hierarchical Trees

The SPIHT (Set Partitioning in Hierarchical Trees) algorithm is a generalization of the EZW algorithm which is proposed in [14]. It uses a partitioning of the trees (which in SPIHT are called *spatial orientation trees*) in a manner that tends to keep insignificant coefficients together in larger subsets. The partitioning decisions are binary decisions that are transmitted to the decoder, providing a significance map encoding that is more efficient than EZW. The thresholds used for checking significance are powers of two, so in essence the SPIHT algorithm sends the binary representation of the integer value of the wavelet

coefficients. As in EZW, the significance map encoding, or set partitioning and ordering step, is followed by a refinement step in which the representations of the significant coefficients are refined.

The algorithm makes use of three lists: the *list of insignificant pixels* (LIP), the *list of significant pixels* (LSP), and the *list of insignificant sets* (LIS). The LSP and LIS lists will contain the coordinates of coefficients, while the LIS will contain the coordinates of descendant coefficients. The LIP is initialized with the set of all root nodes, while the LIS is initialized with all the descendants of the root nodes. The LSP is initially empty. We start by determining the initial value of the threshold by calculating:

$$n = \lfloor \log_2 c_{max} \rfloor \tag{2.6}$$

In each pass, we will first process the members of the LIP, then the members of LIS. This is essentially the significance map encoding step. We then process the elements of LSP in the refinement step.

We begin by examining each coordinate contained in LIP. If the coefficient at that coordinate is significant, we transmit a 1 followed by a bit representing the sign of the coefficient (e.g. 1 for positive, 0 for negative). We then move that coefficient to the LSP. If the coefficient at that coordinate is not significant, we transmit a 0. After examining each coordinate in the LIP, we begin examining the sets in the LIS. If the set at coordinate (i, j) is not significant, we transmit a 0. Otherwise we transmit a 1 plus a sign bit. Once we have processed each of the sets in the LIS, we proceed to the refinement step. In the refinement step we examine each coefficient in the LSP *prior to the current pass* and output the nth MSB of $\mid c_{i,j} \mid$.

This completes one pass. If we still have more bits available, we can start another pass after decreasing the threshold by subtracting 1 from n.

2.1.3 Other Coding Schemes

EZW and SPIHT are most popular image coders based on wavelet decomposition. However, their excellent performance is under one precondition: there are no errors in transmitting the bit stream of the coded image. From the description above, we can see that the length of

the coded bit stream is variable, and the meaning of a bit might be related to other bits in the stream. So for the decoder, once a decoded bit is wrong, the error might not only lead to the wrong value for that pixel, but also propagate to the following decoding decision. This is mainly due to the built-in tree structure of the general zerotree approach and the implicit dependence of the significance map coding strategy. A single bit error can easily cause loss of synchronization between the encoder and decoder execution paths. For example, if zr is decoded as iz in EZW, then it would not be difficult to imagine how disastrous it will be for the subsequent decoding. Therefore, for the transmission over noisy wireless channels which are characterized by unknown and highly varying channel conditions, it is not appropriate to put the bitstream of EZW and SPIHT directly to the channels.

Several papers with different methods have addressed this problem [2]-[8],[11], [12]. Most notably [5], [12] where a modified EZW is proposed, and in [11] where a packetization of SPIHT bitstream and forward error control (FEC) are proposed. A common point in [5], [11] and [12] is to make the bitstream as fixed-length (FL) as possible, so that even if an error occurs it will not affect the following decoding. In [6] a completely new coding scheme for wavelet transformed image is proposed. This scheme makes use of lattice vector quantization (LVQ) to enable faster implementation of the coding procedure, and also make an effort to prevent the decoding errors from propagating by using more fixed-length code streams. This thesis will also use LVQ for most of the higher band wavelet coefficients, rather than using modified EZW or SPIHT.

Except for the difference in coding scheme, there are two focus points of this thesis on which we think more work is needed, and which are not specified by the above papers:

(1) The wireless channels used in above papers are either not for CDMA, but only artifically induced bit errors by assuming that the channel has additive white Gaussian noise (AWGN), or do not take into account all the effects of a CDMA model which should have multipath Rayleigh fading effects as well as multiple access interference (MAI). This thesis will fully consider all the effects of a typical CDMA channel, and specifically, will consider MAI as impulsive interference which only happens when the user number is very high.

(2) In the above papers, only one kind of wavelet filters is used in each paper, either

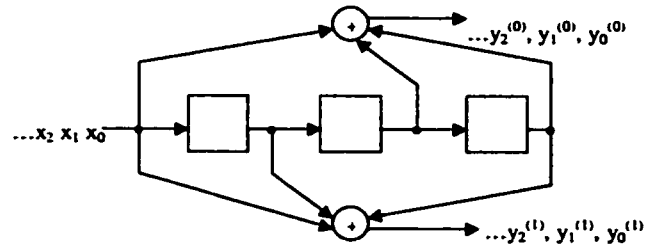

Figure 2.6: A rate-1/2 linear convolutional encoder

Daubechies or biorthogonal. No one uses spline wavelets, and there is no comparison of performance for different wavelets under the same coding scheme and channel conditions. This thesis will give the different performance of Daubechies, biorthogonal and spline wavelets to see which performs better in a standard CDMA channel with the same coding scheme.

2.2 Channel Coding

This thesis will use convolutional codes for channel protection, similar to the approaches of [18]. The following subsections will give the description of the encoding and decoding algorithms of convolutional codes, as well as the principles of a joint source-channel coding scheme, RCPC (rate compatible convolutional coding), which is based on convolutional codes.

2.2.1 Linear Convolutional Encoders

A convolutional encoder converts the entire data stream, regardless of its length, into a single code word. The redundancy of protection information can be introduced into a data stream through the use of a linear shift register. Fig. 2.6 shows a typical rate-1/2 linear convolutional encoder. The rate of this encoder is established by the fact that the encoder outputs two bits for every input bit. In general, an encoder with k inputs and n outputs is said to have rate k/n.

In Fig. 2.6 the binary data stream $\mathbf{x} = (x_0, x_1, x_2, ...)$ is fed into a shift register circuit

consisting of a series of memory elements. With each successive input to the shift register, the values of the memory elements are tapped off and mod-2 added according to a fixed pattern, creating a pair of binary output coded data streams $\mathbf{y}^{(0)} = (y_0^{(0)}, y_1^{(0)}, y_2^{(0)}, \ldots)$ and $\mathbf{y}^{(1)} = (y_0^{(1)}, y_1^{(1)}, y_2^{(1)}, \ldots)$. These output streams can be multiplexed to create a single coded data stream $\mathbf{y} = (y_0^{(0)}y_0^{(1)}, y_1^{(0)}y_1^{(1)}, y_2^{(0)}y_2^{(1)}, \ldots)$. Each element in $\mathbf{y}$ is a linear combination of the elements in the input streams $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k-1)}$. It is assumed that the shift-register contents are initialized to zero before the encoding process begins.

In a convolutional encoder, an important concept - *constraint length* K - is defined as following:

$$K = 1 + \max_i m_i \tag{2.7}$$

where m_i is the number of shift registers of ith branch.

2.2.2 Viterbi Decoding

As information sequence $\mathbf{x}$ is encoded to form a convolutional code word $\mathbf{y}$, then $\mathbf{y}$ is transmitted across a noisy channel. The convolutional decoder takes the received vector $\mathbf{r}$ and generates an estimate $\mathbf{y'}$ of the transmitted code word.

The *maximum likelihood (ML) decoder* selects the estimate $\mathbf{y'}$ that maximizes the probability $p(\mathbf{r}|\mathbf{y'})$. A rate $1/n$ convolutional encoder takes 1 input bit and generates n output bits with each shift of its internal registers. Suppose that we have an input sequence $\mathbf{x}$ composed of L bits:

$$\mathbf{x} = (x_0, \ldots, x_{L-1})$$

The output sequence $\mathbf{y}$ will consist of L n-bit blocks as well as m additional blocks, where m is the length of the shift register in the encoder.

$$\mathbf{y} = (y_0^{(0)}, y_0^{(1)}, \cdots, y_0^{(n-1)}, y_1^{(0)}, y_1^{(1)}, \cdots, y_1^{(n-1)}, y_{L+m-1}^{(0)}, y_{L+m-1}^{(1)}, \cdots, y_{L+m-1}^{(n-1)})$$

A noise-corrupted version $\mathbf{r}$ of the transmitted code word arrives at the receiver, where the decoder generates a maximum likelihood estimate $\mathbf{y'}$ of the transmitted sequence. $\mathbf{r}$ and $\mathbf{y'}$

have the following form:

$$\begin{aligned}\mathbf{r} &= (r_0^{(0)}, r_0^{(1)}, \ldots, r_0^{(n-1)}, r_1^{(0)}, r_1^{(1)}, \ldots, r_1^{(n-1)}, r_{L+m-1}^{(0)}, r_{L+m-1}^{(1)}, \ldots, r_{L+m-1}^{(n-1)}) \\ \mathbf{y'} &= (y_0'^{(0)}, y_0'^{(1)}, \ldots, y_0'^{(n-1)}, y_1'^{(0)}, y_1'^{(1)}, \ldots, y_1'^{(n-1)}, y_{L+m-1}'^{(0)}, y_{L+m-1}'^{(1)}, \ldots, y_{L+m-1}'^{(n-1)})\end{aligned} \tag{2.8}$$

We assume the channel is memoryless. Since the probability of joint, independent events is simply the product of the possibilities of the individual events, it follows that:

$$p(\mathbf{r}|\mathbf{y'}) = \prod_{i=0}^{L+m-1} \left(\prod_{j=0}^{n-1} p\left(r_i^{(j)}|y_i'^{(j)}\right) \right) \tag{2.9}$$

Equation (2.9) is called the *likelihood function* for $\mathbf{y'}$. People usually use its logarithm form:

$$\log p(\mathbf{r}|\mathbf{y'}) = \sum_{i=0}^{L+m-1} \left(\sum_{j=0}^{n-1} \log p\left(r_i^{(j)}|y_i'^{(j)}\right) \right) \tag{2.10}$$

In hardware implementation of the Viterbi decoder, the summands in (2.10) are usually converted to the form called *bit metrics*:

$$M\left(r_i^{(j)}|y_i'^{(j)}\right) = a\left[\log p\left(r_i^{(j)}|y_i'^{(j)}\right) + b\right] \tag{2.11}$$

a and b are chosen such that the bit metrics are small positive integers that can be easily manipulated by digital logic circuits. The *path metric* for a code word $\mathbf{y'}$ is then computed as follows:

$$M(\mathbf{r}|\mathbf{y'}) = \sum_{i=0}^{L+m-1} \left(\sum_{j=0}^{n-1} M\left(r_i^{(j)}|y_i'^{(j)}\right) \right) \tag{2.12}$$

Thus the code word $\mathbf{y'}$ that maximizes $p(\mathbf{r}|\mathbf{y'})$ also maximizes $M(\mathbf{r}|\mathbf{y'})$. Thus the problem now is reduced to finding a code $\mathbf{y'}$ that minimizes the path metric.

The Viterbi algorithm uses trellis diagrams for the computation of path metrics. It has been proven that the path selected by the Viterbi decoder is the maximum likelihood path.

2.2.3 Rate-Compatible Punctured Convolutional Codes

Rate-compatible punctured convolutional (RCPC) codes were introduced in [19]. A family of RCPC codes is described by the mother code of rate-1/n and memory M having the

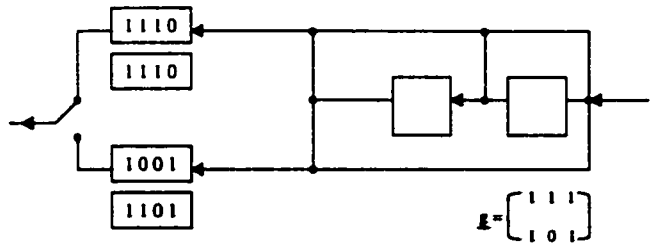

Figure 2.7: RCPC with two rate compatible puncturing tables

generator tap matrix of dimension $n \times (M+1)$. Together with n, the puncturing period P determines the range of code rates

$$R = \frac{P}{P+l} \qquad l = 1, \ldots, (n-1)P \tag{2.13}$$

between $P/(P+1)$ and $1/n$. The RCPC codes are punctured codes of the mother code with puncturing matrices of dimension $n \times P$

$$\mathbf{a}(l) = (a_{ij}(l)) \tag{2.14}$$

with $a_{ij}(l) \in (0,1)$ where 0 implies puncturing. The puncturing matrix can be viewed as a modulo P rule for multiplexing the two streams of code bits. Instead of transmitting $2 \cdot P = 2 \cdot 4$ only $P+l$ bits are transmitted per $P = 4$ information bits, where $P+l$ is the number of "1" in a puncturing matrix.

Fig. 2.7 shows an example where a rate-1/2 code with memory M=2 is punctured periodically with period P=4. Its generator tap matrix is:

$$\mathbf{g} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

And the puncturing matrix

$$\mathbf{a}(1) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

generates a code rate of 4/5, while the puncturing matrix

$$\mathbf{a}(2) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

generates a code rate of 4/6.

Thus with RCPC codes, bitstreams of different importance can be assigned with different puncturing matrice and so different bit rates, so that the most important bitstream is most protected, while less important ones use lower bit budgets.

2.3 CDMA Systems

Code-division multiple-access (CDMA) is a multi-user access scheme that uses different long pseudo-noise (PN) binary sequences to either multiply each user's information sequence which is called direct sequence CDMA (DS-CDMA), or control each user's carrier frequency hopping pattern which is called frequency hopped CDMA (FH-CDMA). In either case, the overall frequency bandwidth used by each user is much larger than the user's information bandwidth so CDMA channel is a kind of spread spectrum (SS) channel. Another aspect of CDMA is that multiple users are using the same frequency band meaning that users are co-channel interference to each other. But this co-channel is still separated by a firewall of PN code which makes the interference noise like. By using a long PN sequence, it achieves the advantages of anti-jamming and user privacy. Also by transmitting all users' signal in the same frequency-time domain, DS-CDMA achieves flexibilities such as soft service degradation with the increase of the number of users, antenna sectorization to improve capacity, soft handoff, etc.

2.3.1 CDMA Signals and Channel

We first give a mathematical formula description of DS-CDMA BPSK (binary phase shift keying) signals and channel. Let's assume that the kth user's signal comes out from the transmitter with initial transmit power A_k, carrier frequency of ω_c and initial phase of θ_k, and can be described as:

$$d_k(t) = \sqrt{2A_k}s_k(t)b_k(t)\cos(\omega_c t + \theta_k) \tag{2.15}$$

where $b_k(t)$ and $s_k(t)$ are the data sequence and spreading sequence respectively with

$$b_k(t) = \sum_{i=-\infty}^{+\infty} b_k[i] P_{T_b}(t - iT_b) \tag{2.16}$$

$$s_k(t) = \sum_{i=-\infty}^{+\infty} s_k[i] P_{T_c}(t - iT_c) \tag{2.17}$$

and $P_T(t)$ is a pulse waveform of duration T, T_c and T_b are spreading bit duration and data bit duration respectively, $s_k[i]$ and $b_k[i]$ are spreading bit and data bit respectively with value of either 1 or -1. A bit of the spreading sequence is usually called a *chip*, and the *chip period* T_c is much smaller than T_b so that, after spreaded by $s_k(t)$, the spectrum of the product $s_k(t)b_k(t)$ is much wider than the spectrum of the original data sequence $b_k(t)$.

Assuming there are K users and each user's signal goes through L different paths, then the received signal at the cellular base station will be:

$$r_c(t) = \sum_{k=1}^{K} \sum_{l=1}^{L} \sqrt{2A_k} c_{kl}(t) s_k(t - \tau_{kl}) b_k(t - \tau_{kl}) \cos(\omega_c t + \varphi_{kl}) + n_c(t) \tag{2.18}$$

where c_{kl} is the fading coefficient for the lth path of kth user, τ_{kl} is the path time delay and φ_{kl} is the composite phase which is:

$$\varphi_{kl} = \theta_k - \omega_c \tau_{kl} - \phi_{kl} \tag{2.19}$$

where ϕ_{kl} is the path phase delay, and $n_c(t)$ is the white noise in the bandwidth of the wireless channel.

2.3.2 Pseudonoise Generators

Pseudonoise (PN) sequences are used as spectrum-spreading modulations for direct sequence SS designs. Ideally, most of the PN sequence design problems would be solved if it were possible to produce a sample of a sequence of independent variables, uniformly distributed, for use at the SS transmitter, and an identical sample sequence at the receiver for use in the detection process. Unfortunately, the generation, recording, and distribution of these sample sequences at a rate equal to the processing gain is generally not feasible.

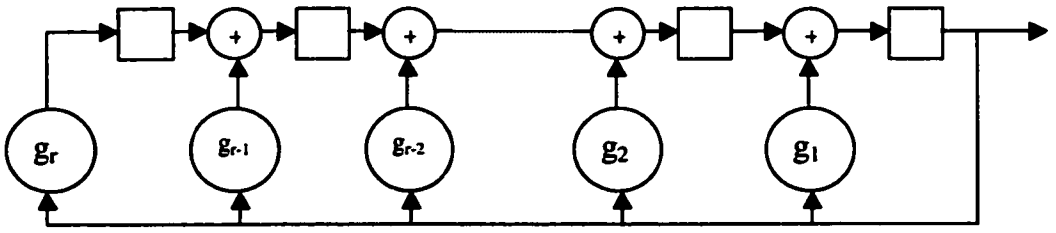

Figure 2.8: Linear feedback shift-register generator

In CDMA systems, PN codes are generated by linear feedback shift-registers having the form of Fig. 2.8 with primitive polynomial $g(D)$. Shift-register sequences having the maximum possible period for an r-stage shift register are called *maximal-length sequences* or *m-sequences*. Since the shift register passes through all possible states, each different initial condition results in a different phase of the same m-sequence. Ideally, a received signal that has been spread using a different PN code will only cause minimal interference in the desired signal after despread by the desired user's PN code. The specific amount of interference from a user employing a different spreading code is related to the cross-correlation between the two spreading codes.

The full-period discrete cross-correlation between two spreading codes of period N is defined as follows [20]:

$$\theta_{ss'}(k) = \frac{1}{N}\sum_{n=0}^{N-1} s(n)s'(n+k) \tag{2.20}$$

Although the detailed correlation could be evaluated for multiple-access code sets, in many cases adequate information for system analysis can be obtained from the *cross-correlation spectrum*. The *cross-correlation spectrum* is a list of all possible values of $\theta_{ss'}(k)$ and the number of values of k which yield that particular cross-correlation. When $s = s'$ the cross-correlation spectrum becomes the *autocorrelation spectrum*. The autocorrelation spectrum for an m-sequence is

1.0	occurs 1 time
$-\frac{1}{N}$	occurs $N-1$ times

In practical CDMA systems, the PN codes used are called *Gold codes* which are generated

from m-sequences. Consider an m-sequence s of length N, and a second sequence s' obtained by sampling every qth symbol of s. The second sequence is said to be a *decimation* of the first. The decimation of an m-sequence may or may not yield another m-sequence. It has been proven [20] that the decimation does yield another m-sequence if and only if gcd(N, q)=1, where "gcd" denotes the greatest common divider.

The cross-correlation spectrum of pairs of m-sequences can be three-valued, four-valued, or possibly many-valued. Certain special pairs of m-sequences whose cross-correlation spectrum is three-valued, where the three values are:

$$\begin{matrix} -\frac{1}{N}t(n) \\ -\frac{1}{N} \\ \frac{1}{N}[t(n)-2] \end{matrix} \tag{2.21}$$

where

$$t(n) = \begin{cases} 1+2^{0.5(n+1)} & \text{for } n \text{ odd} \\ 1+2^{0.5(n+2)} & \text{for } n \text{ even} \end{cases}$$

where the code period $N = 2^n-1$, are called preferred pairs of m-sequences. Finding preferred pairs of m-sequences is necessary in defining sets of Gold codes. The following conditions are sufficient to define a preferred pair s and s' of m-sequences:

1. $n \neq 0$ mod 4; that is, n is odd or n=2 mod 4
2. s'=s[q] where q is odd and there is an integer k such that either

$$q = 2^k + 1$$

or

$$q = 2^{2k} - 2^k + 1$$

3. $\gcd(n,k) = \begin{cases} 1 & \text{for } n \text{ odd} \\ 2 & \text{for } n\text{=2 mod 4} \end{cases}$

The length of the PN codes used in this thesis is 31. They are generated by the polynomial:

$$g(D) = 1 + D^2 + D^5 \tag{2.22}$$

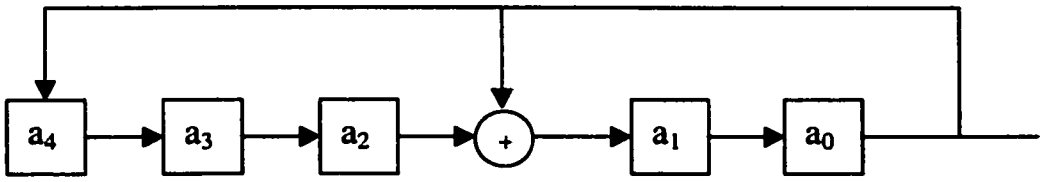

Figure 2.9: Gold code generator of length-31

Fig. 2.9 shows the structure of the PN code generator. The initial values in shift registers are: $a1 \sim a4 \rightarrow 0, a0 \rightarrow 1$. Five PN codes are generated according to the above decimation rules. Their cross-correlation values are listed in Table 2.1.

Table 2.1: Cross-correlation of 5 PN codes

PN	s1	s2	s3	s4	s5
s1	1	-1/31	-1/31	-1/31	-1/31
s2	-1/31	1	-1/31	-1/31	-1/31
s3	-1/31	-1/31	1	-1/31	-1/31
s4	-1/31	-1/31	-1/31	1	-1/31
s5	-1/31	-1/31	-1/31	-1/31	1

According to the theory of PN codes, for any phase shift, the cross-correlation of any pair of PN codes (length 31) takes one of three values: -9/31, -1/31, or 7/31. While auto-correlation of any PN code is 1, the cross-correlation of any pair in this table is obtained when the pair of codes are orthogonal. If asynchronous (there is phase shift between the 2 PN codes), then the cross-correlation will become larger, i.e. be 7/31 or -9/31.

2.3.3 Interference in CDMA that Limits its Capacity

From (2.18) and the description of PN codes above, we can see that when the receiver in the cellular base station despreads a user's signal with its PN codes, there will be interference caused by the cross-correlation with other users' PN codes. This kind of interference is

called multiple access interference (MAI). It happens in the up link of CDMA systems, where mobile users transmit their signal to the base station, and there is no way to control their synchronization so that their PN codes are orthogonal.

When the user number is low, MAI could be approximated as AWGN. However, if the number of users is increasing, MAI needs to be considered separately [21] and could be approximated as being impulsive if the number of users is very high [22], and this is a key factor that limits the capacity of CDMA systems.

This thesis will assume MAI is impulsive by arbitrarily inducing the maximum cross-correlation of different users' PN codes.

Chapter 3

Image Coding Using Wavelets

In chapter 2, we have already introduced the basic structure of wavelet filter banks. In this chapter, we will begin by giving some further details of filters based on different kind of wavelets, namely orthogonal (Daubechies) wavelets, biorthogonal wavelets and spline wavelets. Actually, the spline wavelets that we will use in this thesis are also biorthogonal, but their structure and features are different with the general biorthogonal filters. So we will use a separate section to introduce filters based on spline wavelets.

After that, the principle of lattice vector quantization (LVQ) and its application in image coding using the above wavelets will be shown.

3.1 Orthogonal (Daubechies) Wavelets

The class of Daubechies wavelets is the most commonly used orthogonal wavelets for image coding due to its compact support feature. For the four FIR filters in Fig. 2.1, orthogonality imposes the following important properties:

$$\bar{h}[n] = h[2K-1-n], \qquad \tilde{g}[n] = g[2K-1-n] \tag{3.1}$$

$$g[n] = (-1)^{n+1}h[2K-1-n], \qquad \tilde{g}[n] = (-1)^{n}\tilde{h}[2K-1-n] \tag{3.2}$$

where $2K$ represents the length of those FIR filters. This means that once one filter is determined, the other three filters can also be determined according to (3.1) and (3.2).

Overall, the orthogonal Daubechies wavelet filters have the following properties in the time and frequency domain [17]:

$(a)\langle h[n], h[n+2m]\rangle = \delta[m]$, and $\langle g[n], g[n+2m]\rangle = \delta[m]$.

$(b)H(z)H(z^{-1}) + H(-z)H(-z^{-1}) = 2z^{-2K}$

where $< \cdot, \cdot >$ represents *inner product* for two general vectors $\mathbf{x} = \{\ldots, x[-1], x[0], x[1], \ldots\}$ and $\mathbf{y} = \{\ldots, y[-1], y[0], y[1], \ldots\}$ with

$$< \mathbf{x}, \mathbf{y} >= \sum_{n=-\infty}^{\infty} x^*[n]y[n].$$

Note that (b) could be obtained by applying (3.1) and (3.2) to the perfect reconstruction (PR) conditions (2.3) with the only difference of a delay factor $2K$ in the final output of those filters, in order to make those filters causal. And the filter coefficients are obtained by factoring ([16][23])

$$P(z) = H(z)H(-z) \tag{3.3}$$

where $P(z)$ is

$$P(z) = 2\left(\frac{1+z}{2}\right)^p \left(\frac{1+z^{-1}}{2}\right)^p \sum_{k=0}^{p-1} \binom{p+k-1}{k} \left(\frac{1-z}{2}\right)^k \left(\frac{1-z^{-1}}{2}\right)^k \tag{3.4}$$

and p is said to be the regularity of the related wavelet function. The set of Daubechies filters that will be used in this thesis is called *db8*

with $p = 8$, and the filter length is 16. The coefficients for the four *db8* filters are shown in Table 3.1.

3.2 Biorthogonal Wavelets

Daubechies wavelet analysis is a highly useful orthogonal multiresolution analysis when the signal's characteristics are sought at different scales. For image analysis, since images are mostly smooth (except for occasional edges) it seems appropriate that an exact reconstruction subband coding scheme for image analysis should correspond to an orthogonal basis with a reasonably smooth mother wavelet. On the other hand it is desirable that the FIR filters used be linear phase, because the phase component of an image is more important than its

Table 3.1: Filter coefficients of db8

i	h[n]	g[n]	$\tilde{h}[n]$	$\tilde{g}[n]$
0	-0.00011747678400	-0.05441584224308	0.05441584224308	-0.00011747678400
1	0.00067544940600	0.31287159091447	0.31287159091447	-0.00067544940600
2	-0.00039174037300	-0.67563073629801	0.67563073629801	-0.00039174037300
3	-0.00487035299301	0.58535468365487	0.58535468365487	0.00487035299301
4	0.00874609404702	0.01582910525602	-0.01582910525602	0.00874609404702
5	0.01398102791702	-0.28401554296243	-0.28401554296243	-0.01398102791702
6	-0.04408825393106	-0.00047248457400	0.00047248457400	-0.04408825393106
7	-0.01736930100202	0.12874742662019	0.12874742662019	0.01736930100202
8	0.12874742662019	0.01736930100202	-0.01736930100202	0.12874742662019
9	0.00047248457400	-0.04408825393106	-0.04408825393106	-0.00047248457400
10	-0.28401554296243	-0.01398102791702	0.01398102791702	-0.28401554296243
11	-0.01582910525602	0.00874609404702	0.00874609404702	0.01582910525602
12	0.58535468365487	0.00487035299301	-0.00487035299301	0.58535468365487
13	0.67563073629801	-0.00039174037300	-0.00039174037300	-0.67563073629801
14	0.31287159091447	-0.00067544940600	0.00067544940600	0.31287159091447
15	0.05441584224308	-0.00011747678400	-0.00011747678400	-0.05441584224308

amplitude component. Unfortunately, not all of these conditions can be satisfied simultaneously since there are no orthonormal linear phase FIR filters enabling exact reconstruction for two-channel filter banks except for the Haar basis.

One can preserve linear phase (corresponding to symmetry for the wavelet) by relaxing the orthonormality requirement, and using biorthogonal bases. In such a scheme, the filters h, $\tilde{h}$, and $g, \tilde{g}$ are introduced which are related respectively to the scaling functions ϕ, $\tilde{\phi}$ and to the wavelets ψ, $\tilde{\psi}$. To ensure exact reconstruction the following relationship must hold between the two filters:

$$\begin{aligned} \tilde{g}[n] &= (-1)^n h[-n] \\ g[n] &= (-1)^n \tilde{h}[-n] \end{aligned} \tag{3.5}$$

and, since $\phi_{0,k}$ and $\tilde{\phi}_{0,k}$ are orthogonal,

$$\sum_n h[n]\tilde{h}[n+2k] = \delta_{0,k} \tag{3.6}$$

For the reason of image compression, it is desirable that the regularity of ψ and $\tilde{\psi}$ are sufficiently high. In [24] it is shown that the highly regular wavelets ψ and $\tilde{\psi}$ can be achieved with sufficiently long filters. In particular, if the functions ψ and $\tilde{\psi}$ are respectively $(k-1)$ and $(\tilde{k}-1)$ times continuously differentiable, then the trigonometric polynomials $H(\omega)$ and $\tilde{H}(\omega)$ must be divisible by $(1+e^{-i\omega})^k$ and $(1+e^{-i\omega})^{\tilde{k}}$ respectively.

Applying (3.5) to the PR conditions (2.3), we get a PR condition for biorthogonal cases which is usually the starting point for designing scaling functions ϕ and $\tilde{\phi}$ (in the frequency domain):

$$H(\omega)\tilde{H}(\omega) + H(\omega+\pi)\tilde{H}(\omega+\pi) = 2 \tag{3.7}$$

This condition, along with the divisibility of $H(\omega)$ and $\tilde{H}(\omega)$ by $(1+e^{-i\omega})^k$ and $(1+e^{-i\omega})^{\tilde{k}}$ respectively, leads to [24]

$$H(\omega)\tilde{H}(\omega) = 2\cos^{2l}(\omega/2)\left[\sum_{p=0}^{l-1}\binom{l-1+p}{p}\sin^{2p}(\omega/2) + \sin^{2l}(\omega/2)R(\omega)\right] \tag{3.8}$$

where $R(\omega)$ is an odd polynomial in $\cos(\omega)$ and $2l = k + \tilde{k}$. The symmetry of filters h and $\tilde{h}$ means that $(k+\tilde{k})$ is even. This relationship enables us to compute the dual filter $\tilde{H}(\omega)$ if the filter $H(\omega)$ is imposed.

A broad range of biorthogonal wavelet bases can be constructed. For image compression applications, the regularity of the synthesis functions $\tilde{\psi}_{m,n}$, which is related to the number of zero moments in the analysis functions $\psi_{m,n}$, is more important than the regularity of the $\psi_{m,n}$, or the number of zero moments in the $\tilde{\psi}_{m,n}$. With this consideration, and the limit imposed on filter length, we choose the highest possible values for $\tilde{k}$.

One can choose $R \equiv 0$ with $\tilde{H}(\omega) = \sqrt{2}\cos(\omega/2)^{\tilde{k}} e^{-j\kappa\omega/2}$ where $\kappa = 0$ if $\tilde{k}$ is even, $\kappa = 1$ if $\tilde{k}$ is odd. This corresponds to the filters called *spline filters* in [24] or *binomial filters* because the $\tilde{h}$ are simply binomial coefficients. It then follows that:

$$H(\omega) = \sqrt{2}\cos(\omega/2)^{2l-\tilde{k}} e^{j\kappa\omega/2} \left[\sum_{p=0}^{l-1} \binom{l-1+p}{p} \sin^{2p}(\omega/2) \right] \tag{3.9}$$

Another way in choosing $H(\omega)$ and $\tilde{H}(\omega)$ is still using $R \equiv 0$, but factoring the right-hand side of (3.8), breaking up the polynomial of degree $l-1$ in $\sin(\omega/2)$ into a product of two polynomials in $\sin(\omega/2)$ with real coefficients, one to be allocated to H, the other to $\tilde{H}$. so as to make the lengths of h and $\tilde{h}$ as close as possible. These filters are called *Cohen-Daubechies-Feauveau (CDF) biorthogonal filters.* In this thesis, we will use biorthogonal filter 6/8 (*bior6.8*) which belong to this kind of filters. We choose this kind of filter for biorthogonal case because the filter length is *17* for h and $\tilde{g}$ and *11* for $\tilde{h}$ and g, similar to the length of *db8* for orthogonal case, and the number of zero moments is *5* for $\psi_{m,n}$, which is the highest one available in Matlab toolbox for biorthogonal filter banks. The filter coefficients are listed in Table 3.2.

Table 3.2: Filter coefficients for *bior6.8* filter bank

n	0	±1	±2	±3	±4	±5
h_n	0.8259	0.4208	-0.0941	-0.0773	0.0497	0.0119
$\tilde{h}_n$	0.7589	0.4178	-0.0404	-0.0787	0.0145	0.0144
n	±6	±7	±8			
h_n	-0.0170	-0.0019	0.0019			
$\tilde{h}_n$	0	0	0			

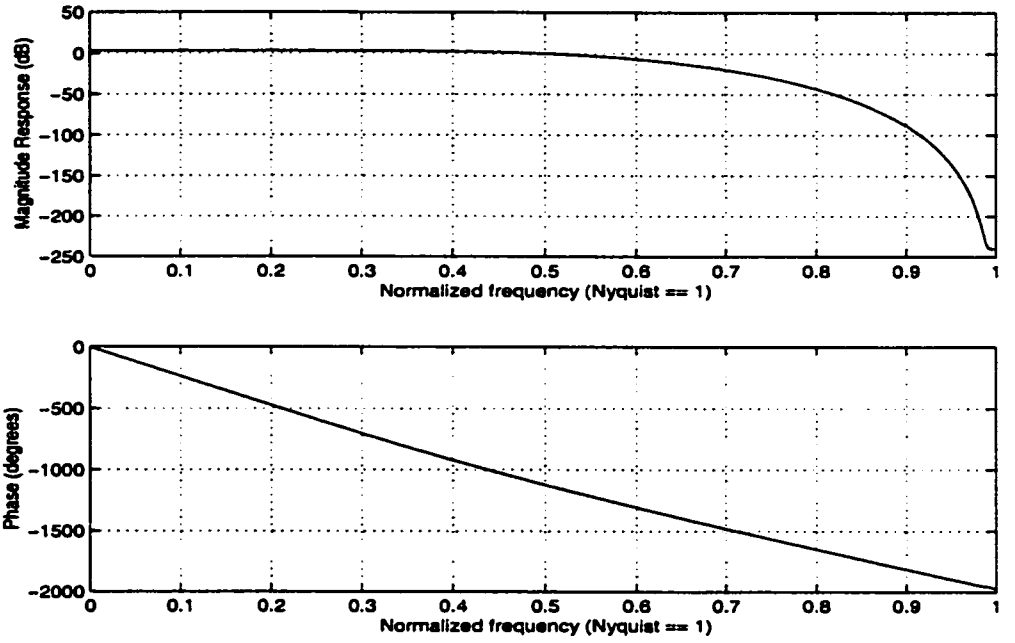

Figure 3.1: Frequency response of low-pass analysis filter h of *db8*

Overall, a two-channel biorthogonal filter bank has the following properties [17]:

$(a)\langle h[-n], \tilde{h}[n-2m]\rangle = \delta[m]$, and $\langle g[-n], \tilde{g}[n-2m]\rangle = \delta[m]$.

$(b) H(z)\tilde{H}(z) + G(z)\tilde{G}(z) = 2$, and $H(-z)\tilde{H}(z) + G(-z)\tilde{G}(z) = 2$.

The frequency response and phase property of filters *db8* and *bior6.8* (low-pass analysis h) are shown in Fig. 3.1 and Fig. 3.2 respectively, from which we can see the improvement of phase linearity of *bior6.8* compared with *db8*. Due to their symmetric nature, biorthogonal filters are usually preferred in dealing with edge effects of wavelet coded images.

3.3 B-Spline Wavelets

Splines are piecewise polynomials, with a smooth fit between the pieces. In the last section we used a B-spline function for one of the scaling functions $\tilde{\phi}$, then determined another scaling function ϕ by the PR condition (3.7). In this section, we will introduce wavelet analysis with ϕ, $\tilde{\phi}$ and ψ, $\tilde{\psi}$ all constructed by B-spline functions.

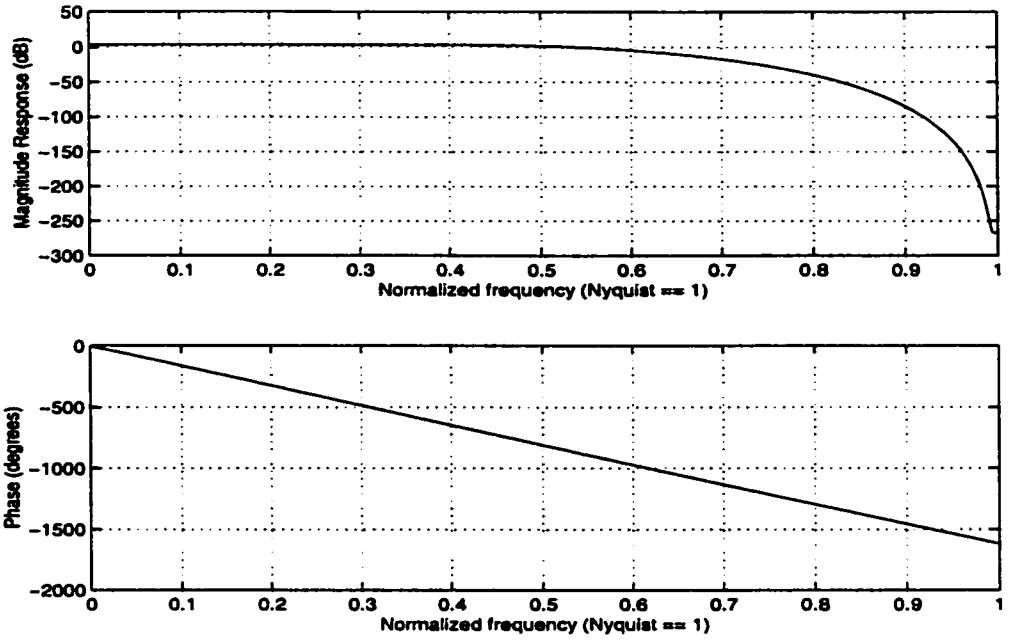

Figure 3.2: Frequency response of low-pass analysis filter h of *bior6.8*

3.3.1 Cubic Spline Wavelets and Cardinal Cubic Spline Wavelets

The B-splines of order n (n odd) satisfy the two scale relation

$$\beta^n(x/2) = \sum_{k=-\infty}^{+\infty} u_2^n(k)\beta^n(x-k), \tag{3.10}$$

where u_2^n is the binomial kernel of order n,

$$u_2^n(k) = \begin{cases} \frac{1}{2^n}\binom{n+1}{k+(n+1)/2}, |k| \leq (n+1)/2, \\ 0, \text{otherwise} \end{cases}$$

$$\longleftrightarrow U_2^n(f) = 2\cos^{n+1}(\pi f) \tag{3.11}$$

and $\beta^n(x)$ can be constructed by repeated convolution of a B-spline of order 0:

$$\beta^n(x) = \underbrace{\beta^0 * \beta^0 * \cdots * \beta^0}_{n+1 \text{ times}}(x), \tag{3.12}$$

where $\beta^0(x)$ is the characteristic function in the interval $[-\frac{1}{2}, \frac{1}{2})$. The shifted version of $\beta^0(x)$ to the interval $[0, 1)$ is just the scaling function of the Haar wavelet. This two-scale relation

can be used to construct a multiresolution analysis [25]:

$$f_{(i)} = 2^{-i} \sum_{k \in \mathbf{Z}} < f, \phi_{i,k} > \tilde{\phi}_{i,k} \tag{3.13}$$

$$f_{(i)} = 2^{-i} \sum_{k \in \mathbf{Z}} < f, \psi_{i,k} > \tilde{\psi}_{i,k} \tag{3.14}$$

where $\phi_{i,k} = \phi^n(2^{-i} - k)$ and $\tilde{\phi}_{i,k} = \tilde{\phi}^n(2^{-i} - k)$, $\psi_{i,k} = \psi^n(2^{-i} - k)$ and $\tilde{\psi}_{i,k} = \tilde{\psi}^n(2^{-i} - k)$ are biorthogonal and defined as follows ([25][26]):

$$\begin{cases} \phi^n(x) = \sum_{k=-\infty}^{+\infty} p(k)\beta^n(x-k) \\ \tilde{\phi}^n(x) = \sum_{k=-\infty}^{+\infty} \left(p' * b^{2n+1}\right)^{-1}(k)\beta^n(x-k) \end{cases} \tag{3.15}$$

where $(a * b)^{-1}$ means calculating the convolution of a and b first, then inversing the convolution result. And

$$\begin{cases} \psi^n(x) = \sum_{k=-\infty}^{+\infty} q(k)\psi_b^n(x-k) \\ \tilde{\psi}^n(x) = \sum_{k=-\infty}^{+\infty} \left(q' * b^{2n+1} * \left[\hat{b}^{2n+1} * b^{2n+1}\right]_{\downarrow 2}\right)^{-1}(k))\psi_b^n(x-k) \end{cases} \tag{3.16}$$

where

$$\psi_b(x/2) = \sum_{k=-\infty}^{+\infty} \hat{u}_2^n * \hat{b}^{2n+1}(k+1)\beta^n(x-k), \tag{3.17}$$

The sequence b^n denotes the discrete spline of order n, which is obtained by sampling the B-splines at the integers:

$$b^n(k) = \beta^n(x)|_{x=k} \longleftrightarrow B_1^n(z) \tag{3.18}$$

and the notation $\hat{a}(k) := (-1)^k a(k)$ refers to a modulation by $(-1)^k$, and $p'(k) = p(-k)$. The weighting sequences p and q depend on which kind of spline wavelets we are using. For general B-spline functions, we have $p = q = \delta_0$, where $\delta_0 = \delta(0)$ is the *impulse function* defined by:

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Another kind of spline filters that we will use in this thesis is called *cardinal spline filters* [27]. It allows a function or a signal to be represented by its sample values. The weighting sequences for it are

$$\begin{cases} p(k) = (b_n)^{-1}(k) \\ q = \left(\left[b^n * \hat{u}_2^n * \hat{b}^{2n+1}\right]_{\downarrow 2}\right)^{-1} \end{cases} \tag{3.19}$$

The cardinal spline wavelet decomposition allows a better visualization of the underlying continuous signals because the expansion coefficients are the sample values of the coarse resolution spline and wavelet signal components. More important is, the cardinal spline has better energy concentration than that of the general B-spline. We can see that in Fig. 3.3 in which the basic functions of cubic spline and cardinal cubic spline and their Fourier transforms are shown. As the order n increases, the cardinal spline becomes more and more similar to a *sinc* function (we also draw the *sinc* function in dashed line in Fig. 3.3(c) for comparison) that corresponds to the ideal interpolator for a bandlimited function [28]. The B-spline functions, on the other hand, become more and more Gaussian-like, as a consequence of the central limit theorem.

Equations (3.15) and (3.16) are the general scaling and wavelet functions for spline wavelets. Their associated filter banks are determined as follows [26]:

$$\begin{cases} h(k) = \frac{1}{2}\left[(p * b^{2n+1})^{-1}\right]_{\uparrow 2} * p * b^{2n+1} * u_2^n(k) \\ g(k+1) = \frac{1}{2}\left[(q * b^{2n+1})^{-1}\right]_{\uparrow 2} * p * \hat{u}_2^n(k) \\ \tilde{h}(k) = [p]_{\uparrow 2} * (p)^{-1} * u_2^n(k) \\ \tilde{g}(k-1) = [q]_{\uparrow 2} * (p)^{-1} * \hat{u}_2^n * \hat{b}^{2n+1}(k) \end{cases} \tag{3.20}$$

In next section we will describe how to implement these filters.

3.3.2 Implementation of Spline Filters

For the spline filter banks in (3.20), we will use basic B-spline and cardinal B-spline wavelets in the simulation.The choices of weighting coefficients p and q for them are listed in Table 3.3. And the expressions for $B^n(z)$ - the z-transform of b^n - of order up to $n = 7$ with poles of their reciprocals are listed in Table 3.4.

From the expressions for B-spline filters in (3.20), we can see that part of the filter implementation will involve symmetric anti-causal IIR filters of the form $[H^{-1}(z)]$ and $[H^{-1}(z)]_{\uparrow 2}$, where $H^{-1}(z)$ can be expressed as

$$H^{-1}(z) = \frac{c_0}{[z^N + z^{-N}] + \left(\sum_{k=1}^{N-1} d_k[z^k + z^{-k}]\right) + d_0} \tag{3.21}$$

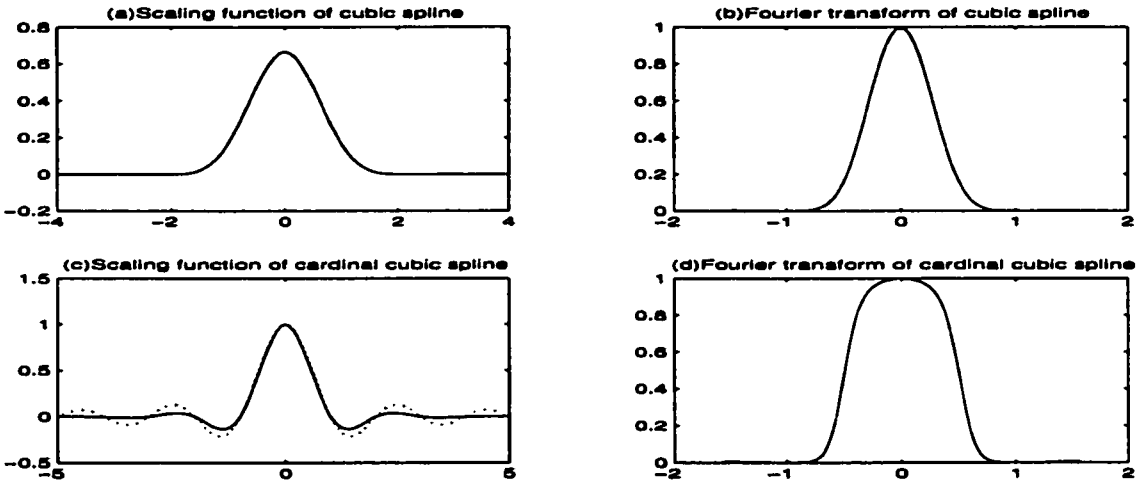

Figure 3.3: Signals associated with the cubic spline interpolator. (a)Cubic B-spline, (b)Fourier transform of the cubic B-spline, (c)cardinal cubic spline, the dashed line corresponds to *sinc* function, (d)Fourier transform of the cardinal cubic spline.

Table 3.3: Parameters for basic and cardinal B-spline wavelets

Representations	p	q
basic (B-splines)	δ_0(identical)	δ_0(identical)
cardinal (B-spline)	$(b^n)^{-1}$	$\left([b^n * \hat{u}_2^n * \hat{b}^{2n+1}]_{\downarrow 2}\right)^{-1}$

Now we present a general procedure for implementing the IIR filters in general form of (3.21) [29][30]. We use the fact that a generic symmetrical recursive filter may be decomposed into a cascade of elementary symmetrical exponential filters which themselves can be separated into two complementary causal and anticausal components as follows:

$$H^{-1}(z) = c_0 \prod_{i=1}^{N} H_0(z; z_i) \tag{3.22}$$

where $H_0(z; z_i)$ is defined as follows:

$$H_0(z; z_i) := \frac{-z_i}{(1 - z_i z^{-1})(1 - z_i z)} = \frac{-z_i}{(1 - z_i^2)} \left(\frac{1}{1 - z_i z^{-1}} + \frac{1}{1 - z_i z} - 1 \right) \tag{3.23}$$

It is not difficult to show that $H_0(z; z_i)$ is the transfer function of a symmetrical exponential filter whose impulse response is

$$h_0(k; z_i) = \frac{z_i}{(1 - z_i^2)} z_i^{|k|} \tag{3.24}$$

Starting from (3.24), assuming that the input signal is $x(k)_{k=0,\cdots,K-1}$,we can derive the fol-

Table 3.4: Transfer functions and poles of b^n

n	$B^n(z)$	Poles of $B^n(z)^{-1}$
0	1	-
1	1	-
2	$\frac{z+6+z^{-1}}{8}$	$z_1 = -3 + 2\sqrt{2} = -0.171573$
3	$\frac{z+4+z^{-1}}{6}$	$z_1 = -2 + \sqrt{3} = -0.267949$
4	$\frac{z^2+76z+230+76z^{-1}+z^{-2}}{384}$	$z_1 = -0.361341$, $z_2 = -0.0137254$
5	$\frac{z^2+26z+66+26z^{-1}+z^{-2}}{120}$	$z_1 = -0.430575$, $z_2 = -0.0430963$
6	$\frac{z^3+722z^2+10543z+23548+10543z^{-1}+722z^{-2}+z^{-3}}{46080}$	$z_1 = -0.488295$, $z_2 = -0.0816793$, $z_3 = -0.00141415$
7	$\frac{z^3+120z^2+1191z+2416+1191z^{-1}+120z^{-2}+z^{-3}}{5040}$	$z_1 = -0.53528$, $z_2 = -0.122555$, $z_3 = -0.00914869$

lowing recursive filter equations [29]:

$$\begin{cases} y^+(k) = x(k) + z_i y^+(k-1), \quad (k = 1, \cdots, K-1) \\ y(K-1) = c_i(2y^+(K-1) - x(K-1)) \\ y(k) = z_i(y(k+1) - y^+(k)), \quad (k = K-2, \cdots, 0) \end{cases} \tag{3.25}$$

where $x(k)$ and $y(k)$ are the input and output signals respectively, and where $c_i = -z_i/(1-z_i^2)$ is a scaling constant. For boundary conditions we use symmetric extension. The recursion begins with

$$y^+(1) = \sum_{k=1}^{k_0} z_i^{|k-1|} x(k) \tag{3.26}$$

where k_0 is chosen to ensure that $z_i^{|k_0|}$ is smaller than some prescribed level of precision. In our simulation, we set $k_0 = \log(10^{-8})/log(|z_i|)$. The second equation in (3.25) is borrowed from a sum decomposition (right-hand side of (3.23)) and is required to obtain a correct initialization of the backward recursion. Thus the symmetric anti-causal IIR filter of the style $[H^{-1}(z)]$ can be implementated according to (3.22)-(3.25).

For the anti-causal IIR filter of the style $[H^{-1}(z)]_{\uparrow m}$, its z-transform can be written in the standard form [30]:

$$H^{-1}(z^m) = \frac{c_0}{[z^{mN} + z^{-mN}] + \left(\sum_{k=1}^{N-1} d_k[z^{mk} + z^{-mk}]\right) + d_0} \tag{3.27}$$

The filter is then expressed as a cascade of simple first-order causal/anti-causal components

$$H^{-1}(z^m) = c_0 \prod_{i=1}^{N} H_0(z^m; z_i) \tag{3.28}$$

where $H_0(z^m; z_i)$ is defined as:

$$H_0(z^m; z_i) := \frac{-z_i}{(1 - z_i z^{-m})(1 - z_i z^m)} = \left(\frac{1}{1 - z_i z^{-m}}\right)\left(\frac{-z_i}{1 - z_i z^m}\right) \tag{3.29}$$

Then the recursive equations to calculate the output $y(k)$ of a component filter $H_0(z^m; z_i)$ is as follows:

$$\begin{cases} y^+(k) = x(k) + z_i y^+(k-m), \quad k = m, \cdots, K-1 \\ y(k) = -z_i y^+(k) + z_i y(k+m) = z_i(y(k+m) - y^+(k)), \quad k = K-1-m, \cdots, 0 \end{cases} \tag{3.30}$$

In order to calculate $y^+(k)$ recursively using (3.30), we need to know $y^+(k)$ for $k = 0, \cdots, m-1$. These initial values are computed using

$$y^+(k) = (h_0^+ * x)(k) = \sum_{j\geq 0} x(k-mj)z_i^j = \sum_{j=0}^{k_0} x(k-mj)z_i^j, \quad k = 0, \cdots, m-1 \tag{3.31}$$

The recursive computation of (3.30) also requires $y(k)$ to be known for $k = K-m, \cdots, K-1$. In order to obtain these values, we rewrite (3.29) as

$$H_0(z^m; z_i) := \frac{-z_i}{(1-z_i^2)}\left(\frac{1}{1-z_i z^{-m}} + \frac{1}{1-z_i z^m} - 1\right) = \frac{-z_i}{(1-z_i^2)}\left(H_0^+(z^m) + H_0^-(z^m) - 1\right) \tag{3.32}$$

which suggests the following computation:

$$y(k) = \frac{-z_i}{(1-z_i^2)}(y^+(k) + y^-(k) - x(k)), \quad k = K-m, \cdots, K-1 \tag{3.33}$$

where

$$y^-(k) = \sum_{j=0}^{k_0} x(k+mj)z_i^j \tag{3.34}$$

Specifically, we first calculate $y^-(k)$ for $k = K-m, \cdots, K-1$, and then using (3.33) to compute the m initial values for $y(k)$. Note that (3.33) is only for calculating the initial values of $y(k)$; for the other filter outputs we should use (3.30).

Thus, for the two B-spline filters we will use, we have the basic cubic B-spline (n=3) filters as:

$$\begin{cases} h(k) = \frac{1}{2}\left[(b^7)^{-1}\right]_{\uparrow 2} * b^7 * u_2^3(k) \\ \quad = \frac{1}{16}\left[\frac{1}{z^3+120z^2+1191z+2416+1191z^{-1}+120z^{-2}+z^{-3}}\right]_{\uparrow 2} \\ \quad * (z^5 + 124z^4 + 1677z^3 + 7904z^2 + 18482z + 24264 + 18482z^{-1} + 7904z^{-2} + 1677z^{-3} + 124z^{-4} + z^{-5}) \\ g(k+1) = \frac{1}{2}\left[(b^7)^{-1}\right]_{\uparrow 2} * \hat{u}_2^3(k) \\ \quad = 315\left[\frac{1}{z^3+120z^2+1191z+2416+1191z^{-1}+120z^{-2}+z^{-3}}\right]_{\uparrow 2} * (z^2 - 4z + 6 - 4z^{-1} + z^{-2}) \\ \tilde{h}(k) = u_2^3(k) = \frac{1}{8}(z^2 + 4z + 6 + 4z^{-1} + z^{-2}) \\ \tilde{g}(k-1) = \hat{u}_2^3 * \hat{b}^7(k) = \frac{1}{40320} \\ \quad * (-z^5 + 124z^4 - 1677z^3 + 7904z^2 - 18482z + 24264 - 18482z^{-1} + 7904z^{-2} - 1677z^{-3} + 124z^{-4} - z^{-5}) \end{cases} \tag{3.35}$$

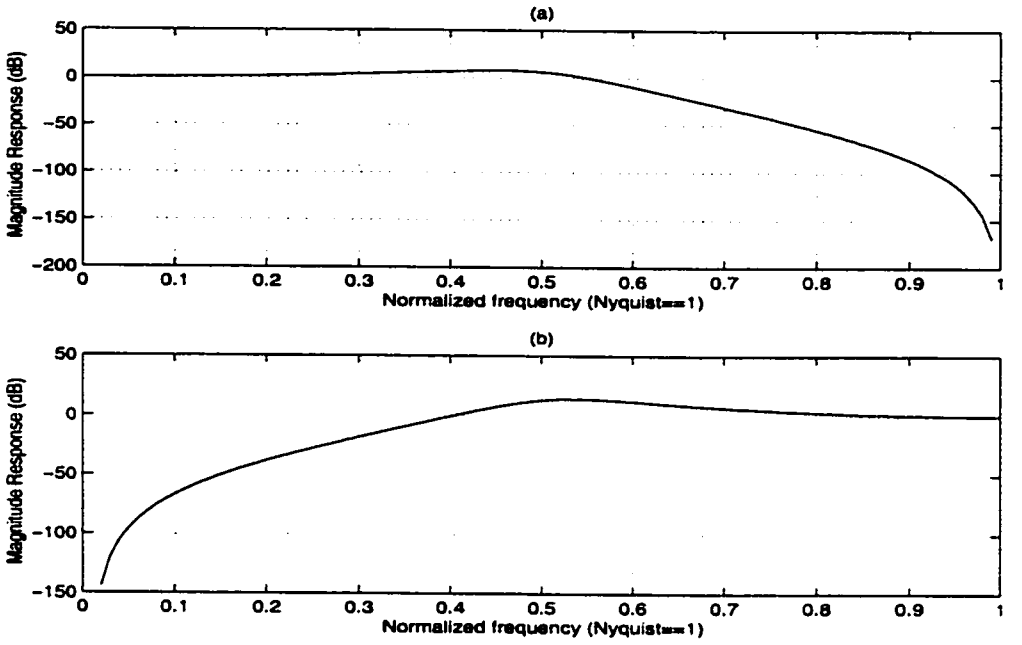

Figure 3.4: Magnitude frequency response of analysis filters of basic cubic B-spline: (a)lowpass filter h; (b)highpass filter g.

Fig. 3.4 and Fig. 3.5 show the magnitude frequency response of these cubic B-spline filters. And due to the symmetric nature of these filters, they also have the feature of linear phase. Fig. 3.6 shows the original image of *Lena* which we will use in this thesis. And Fig. 3.7 shows the *Lena* image after two-stage wavelet transforms by analysis filters h and g. The reconstructed image using synthesis filters $\tilde{h}$ and $\tilde{g}$ is exactly the same as the original image.

For the cardinal cubic B-spline, we have

$$\begin{cases} p = (b^3)^{-1} = \frac{6}{z+4+z^{-1}} \\ q = \left(\left[b^3 * \hat{u}_2^3 * \hat{b}^7 \right]_{\downarrow 2} \right)^{-1} \\ \quad = \frac{241920}{-z^3-1182z^2+11457z+60092+11457z^{-1}-1182z^{-2}-z^{-3}} \end{cases}$$

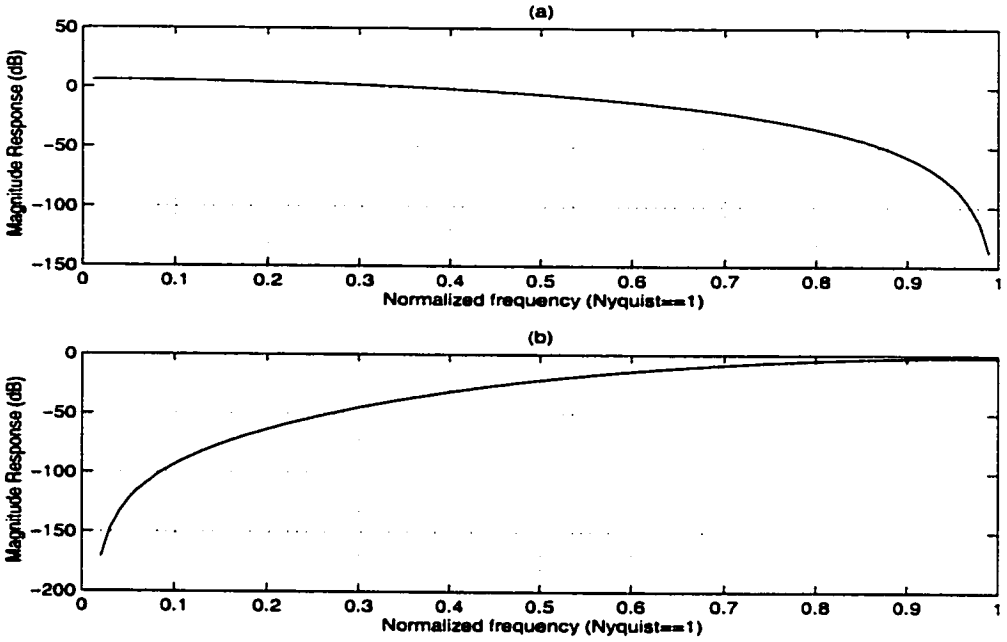

Figure 3.5: Magnitude frequency response of synthesis filters of basic cubic B-spline: (a)lowpass filter $\tilde{h}$; (b)highpass filter $\tilde{g}$.

Figure 3.6: Original image *Lena*

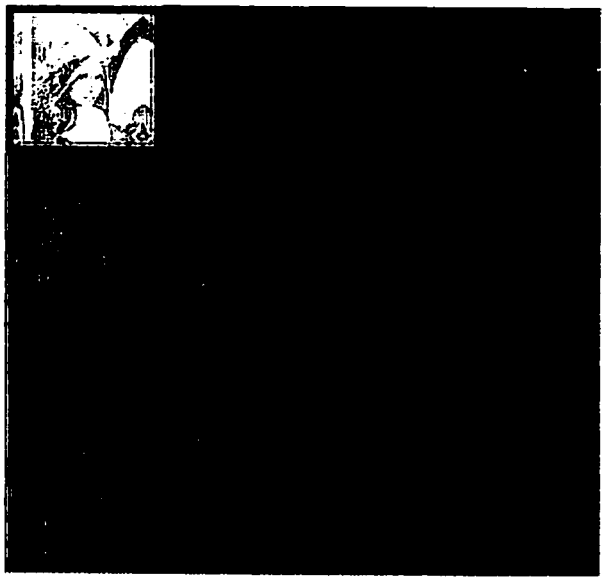

Figure 3.7: Two stage wavelet transforms by basic cubic B-spline filters

And its related filters are:

$$
\begin{cases}
h(k) = \frac{1}{2}\left[\left((b^3)^{-1} * b^7\right)^{-1}\right]_{\uparrow 2} * (b^3)^{-1} * b^7 * u_2^3(k) \\
\quad = \frac{1}{16}\left[\frac{z+4+z^{-1}}{z^3+120z^2+1191z+2416+1191z^{-1}+120z^{-2}+z^{-3}}\right]_{\uparrow 2} * \left(\frac{1}{z+4+z^{-1}}\right) \\
\quad * \left(z^5 + 124z^4 + 1677z^3 + 7904z^2 + 18482z + 24264 + 18482z^{-1} + 7904z^{-2} + 1677z^{-3} + 124z^{-4} + z^{-5}\right) \\
g(k+1) = \frac{1}{2}\left[(q * b^7)^{-1}\right]_{\uparrow 2} * (b^3)^{-1} * \hat{u}_2^3(k) \\
\quad = \frac{1}{128}\left[\frac{-z^3-1182z^2+11457z+60092+11457z^{-1}-1182z^{-2}-z^{-3}}{z^3+120z^2+1191z+2416+1191z^{-1}+120z^{-2}+z^{-3}}\right]_{\uparrow 2} * \left(\frac{1}{z+4+z^{-1}}\right) * \left(z^2 - 4z + 6 - 4z^{-1} + z^{-2}\right) \\
\tilde{h}(k) = \left[(b^3)^{-1}\right]_{\uparrow 2} * b^3 * u_2^3(k) = \frac{1}{8}\left(\frac{1}{z+4+z^{-1}}\right)_{\uparrow 2} * \left(z^3 + 8z^2 + 23z + 32 + 23z^{-1} + 8z^{-2} + z^{-3}\right) \\
\tilde{g}(k-1) = [q]_{\uparrow 2} * b^3 * \hat{u}_2^3 * \hat{b}^7(k) = \left(\frac{1}{z^3+1182z^2-11457z-60092-11457z^{-1}+1182z^{-2}+z^{-3}}\right)_{\uparrow 2} \\
\quad * \left(z^6 - 120z^5 + 1182z^4 - 1320z^3 - 11457z^2 + 41760z - 60092 + 41760z^{-1} - 11457z^{-2} - 1320z^{-3} + 1182z^{-4} - 120z^{-5} + z^{-6}\right)
\end{cases}
\tag{3.36}
$$

Fig. 3.8 and Fig. 3.9 show the magnitude frequency response of these cardinal cubic B-spline filters. As in the case of basic cubic B-spline, these filters also have the feature of linear phase. And Fig. 3.10 shows the image after two-stage wavelet transforms by the analysis filters of cardinal cubic B-spline h and g. The reconstructed image using synthesis filters $\tilde{h}$

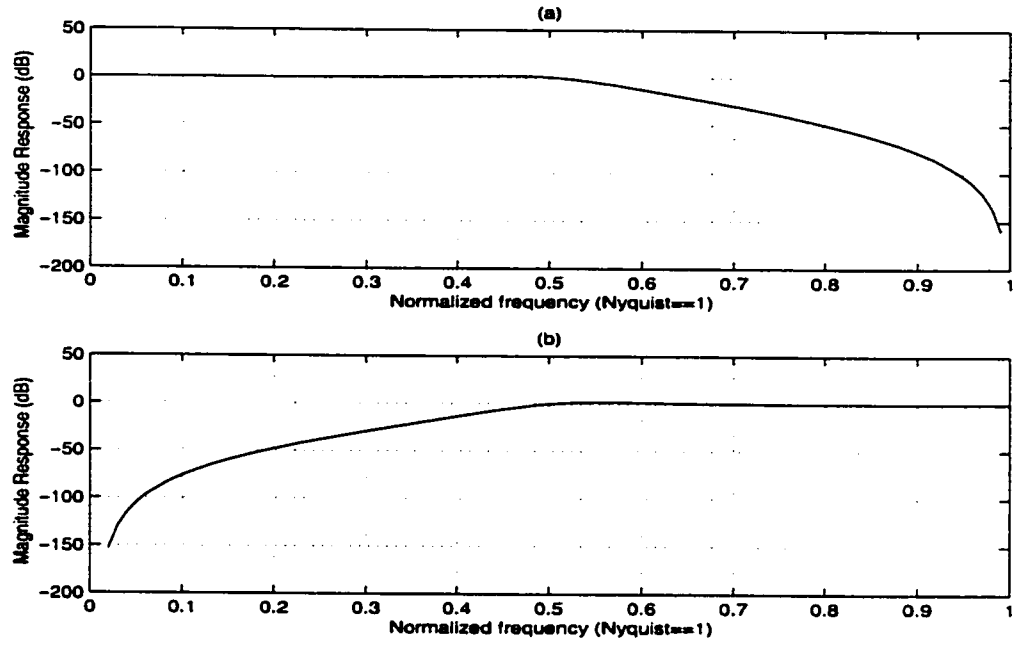

Figure 3.8: Magnitude frequency response of analysis filters of cardinal cubic B-spline: (a)lowpass filter h; (b)highpass filter g.

and $\tilde{g}$ is exactly the same as the original image.

3.3.3 Some Statistics of Subband Coefficients

Before we apply any compression techniques to the transformed images, we need to do some investigation of the statistical properties of the subband coefficients of transformed images. First we show the histogram of the original image of *Lena* in Fig. 3.11, then we put the histogram of the lowest subband LLL for the filters of *db8, bior6.8*, basic cubic spline as well as cardinal cubic spline in Fig. 3.12. By comparing Fig. 3.12 with Fig. 3.11, we can find that *db8, bior6.8* and cardinal spline maintained the similar shape of the original histogram (except for cardinal spline where the peak around zero is caused by edge effect). The range of pixel values for *db8, bior6.8* are spreaded from [0,1] to about [0,4], but this could be solved by scaling down the whole pixel values while doing compression, and scale it back when doing decompression.

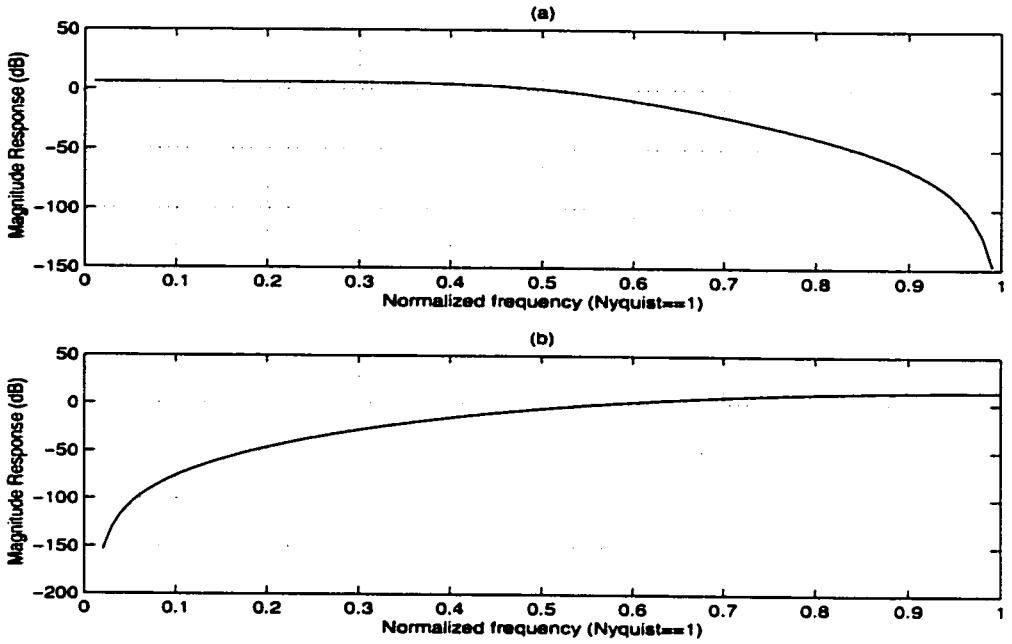

Figure 3.9: Magnitude frequency response of synthesis filters of cardinal cubic B-spline: (a)lowpass filter $\tilde{h}$; (b)highpass filter $\tilde{g}$.

Figure 3.10: Two stage wavelet transforms by cardinal cubic B-spline filters

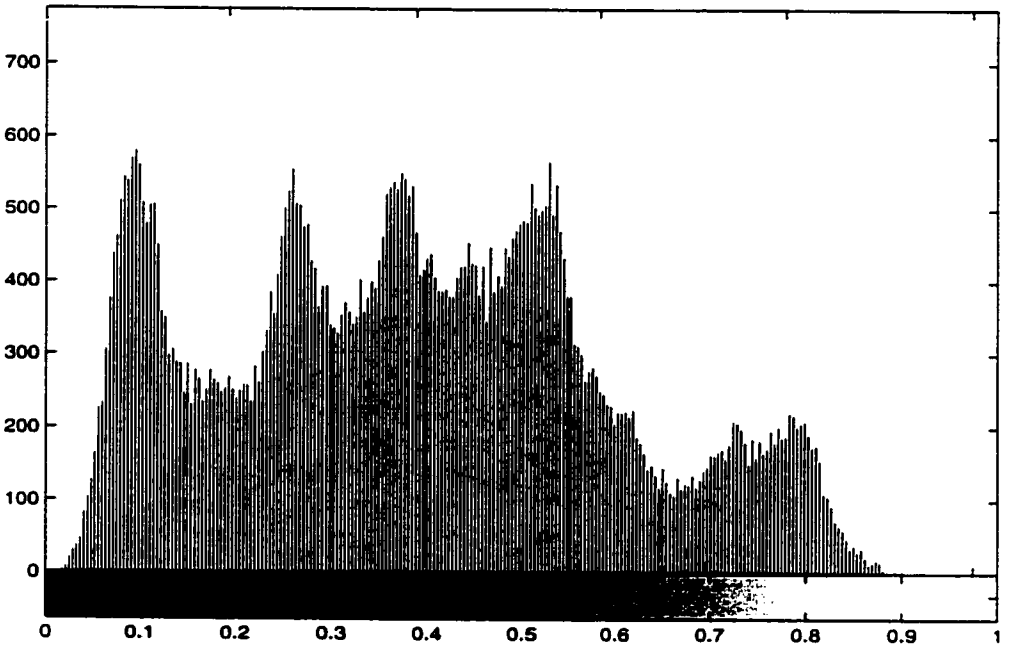

Figure 3.11: Histogram of original image *Lena*

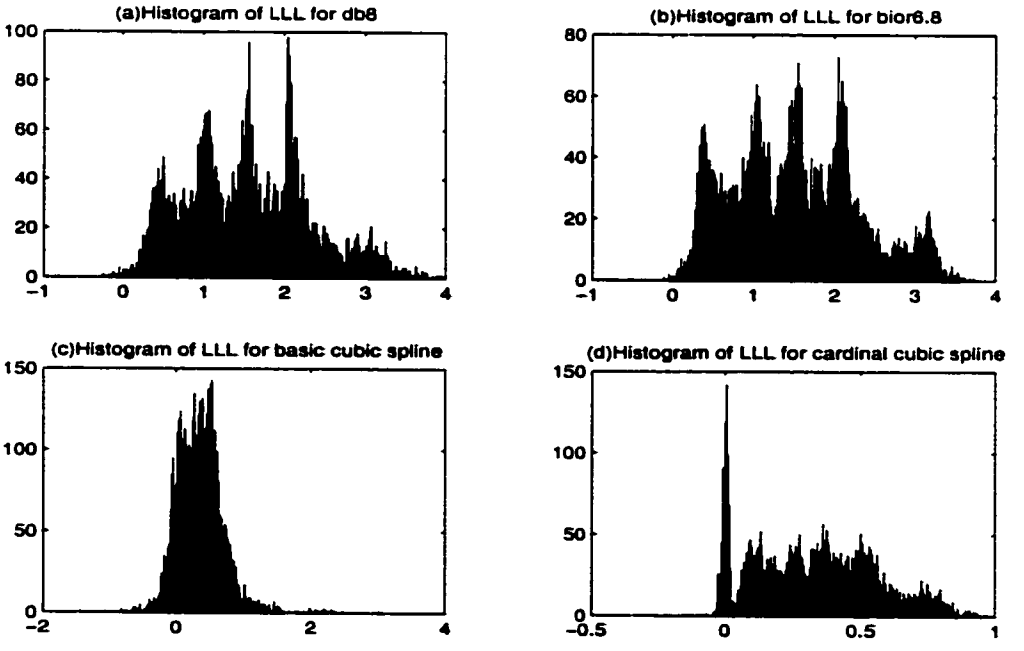

Figure 3.12: Histogram of subband *LLL*

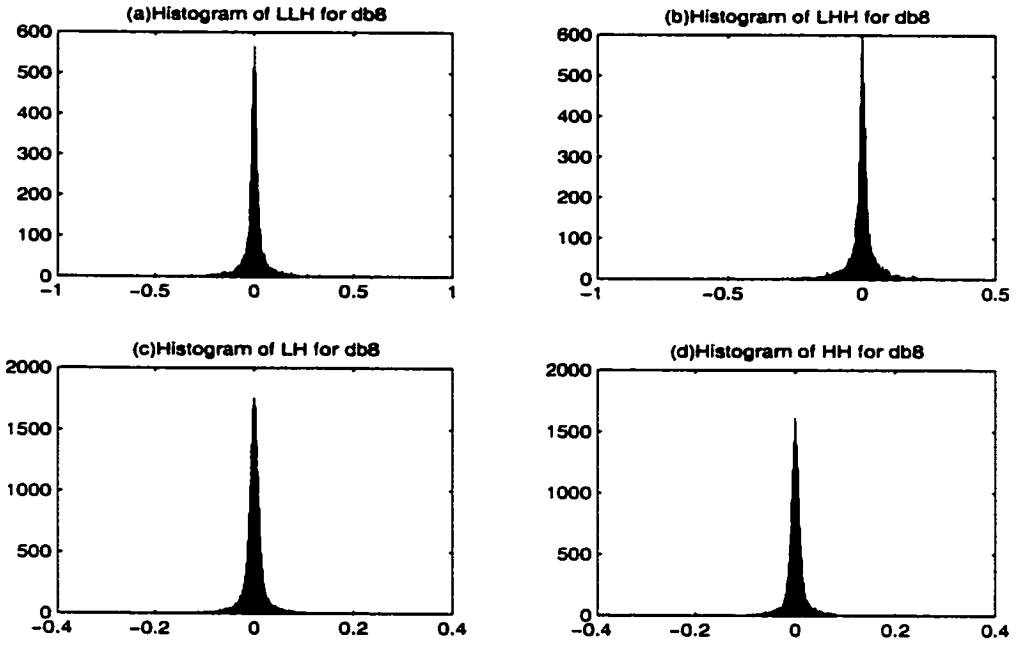

Figure 3.13: Histogram of higher subbands for *db8*

For higher subbands, we give the histogram of subbands LLH, LHH, LH and HH For the four kinds of filters in Fig. 3.13—3.16. From these figures we can see that, cardinal cubic spline filters have the narrowest coefficient distribution compared with the distribution of any subband coefficients of the other filters' related subband. And the basic cubic spline filters have the widest distribution. For the quantization of these coefficients, a narrower distribution has more advantage. Therefore, from the distribution point of view, cardinal cubic spline wavelets should have more privileges in quantization than that of the other wavelets. And basic cubic spline is the least privileged one among the four kinds of wavelets. Thus, we will not use basic cubic spline wavelets further in this thesis.

An important property of wavelet filters is the energy concentration of subband coefficients. The more the energy of coefficients concentrates in one or several subbands, the better for quantization and compression. To compare the energy concentration property of cardinal cubic spline, *bior*6.8 and *db*8, we list the mean-square-errors of the images reconstructed by using only the subband LLL, as well as using LLL plus LLH, LHL and LHH subbands of

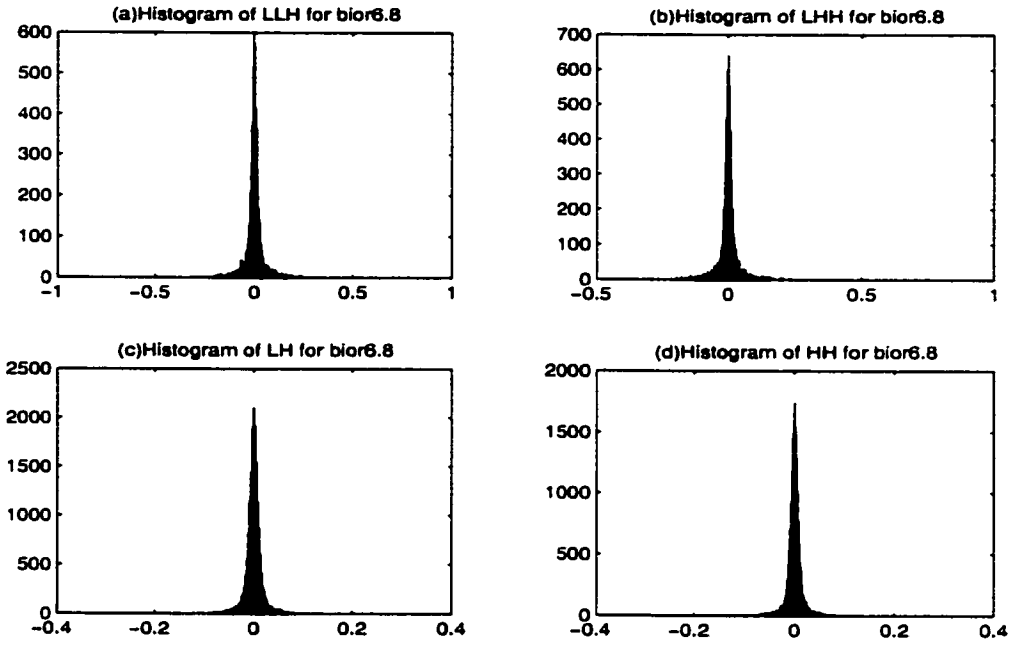

Figure 3.14: Histogram of higher subbands for *bior6.8*

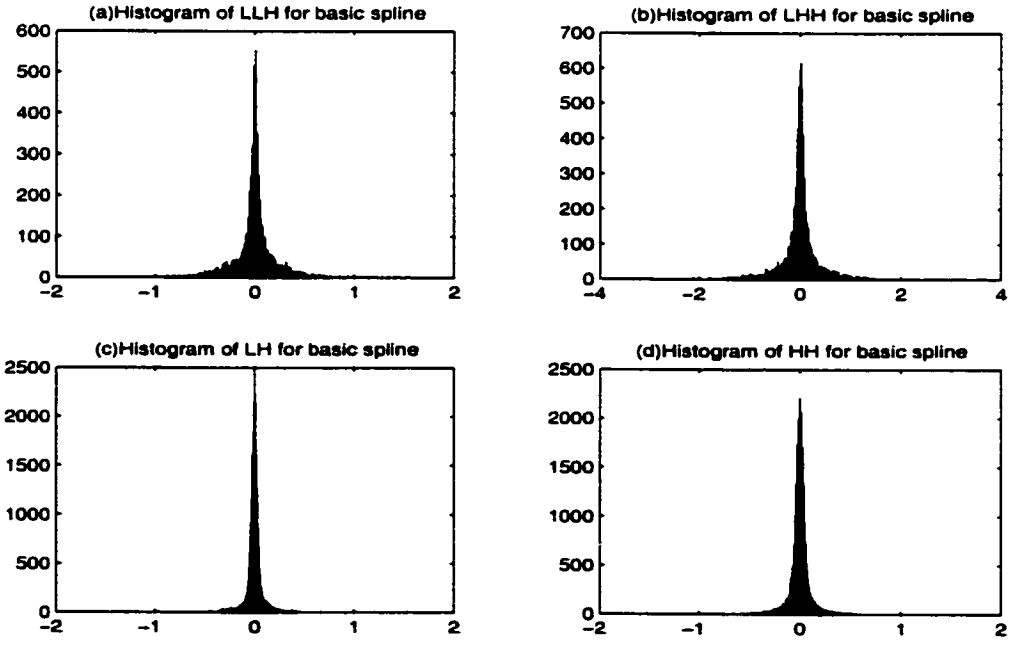

Figure 3.15: Histogram of higher subbands for basic cubic spline

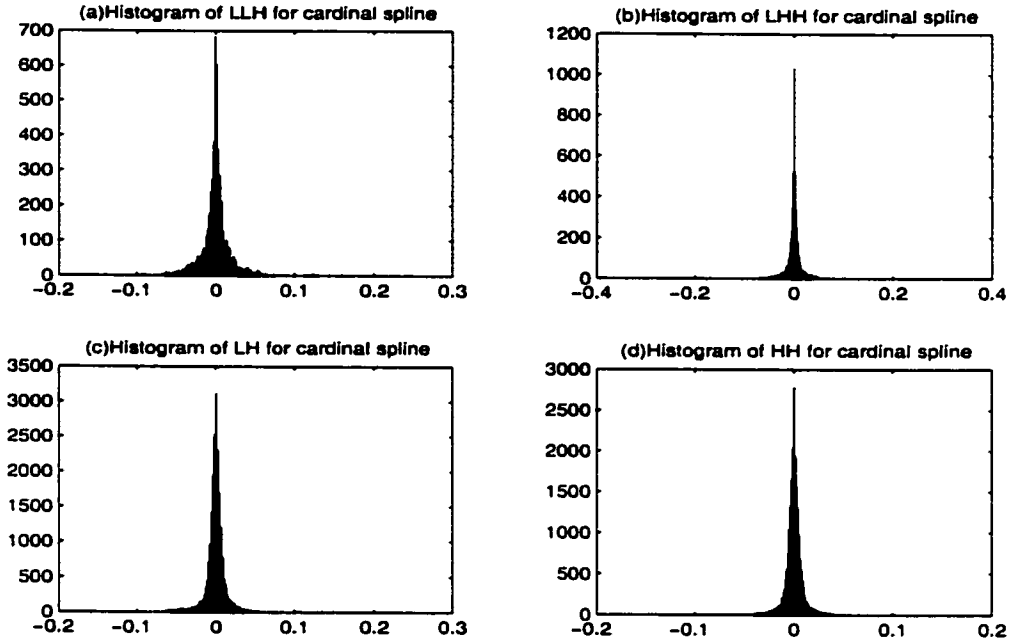

Figure 3.16: Histogram of higher subbands for cardinal cubic spline

the three kinds of wavelet filters in Table 3.5. From the table we can see that *bior*6.8 and *db*8 have the most energy concentration in subband LLL, while cardinal cubic spline has the most energy concentration if considering subbands LLL, LLH, LHL and LHH together.

Table 3.5: Mean-square-errors of the images reconstructed by using only one or several subbands

Wavelet	Subband	MSE
Cardinal spline	LLL	0.0027
	LLL,LLH,LHL,LHH	8.12×10^{-4}
*bior*6.8	LLL	0.0026
	LLL,LLH,LHL,LHH	8.21×10^{-4}
*db*8	LLL	0.0026
	LLL,LLH,LHL,LHH	8.50×10^{-4}

To show that in another way, we give the characteristic curves of these distributions of

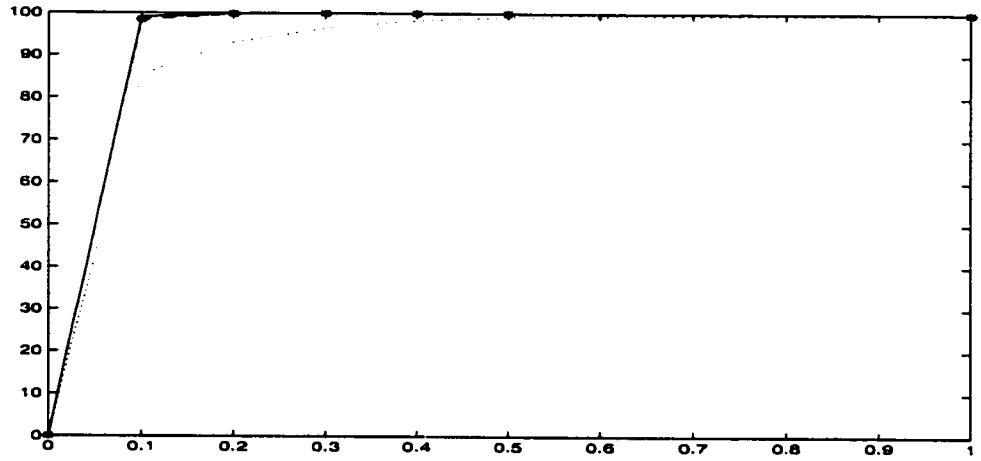

Figure 3.17: Percentage of distribution characteristics of subband LH. Solid line —cardinal spline: star line ** *bior6.8*; dashed line - - *db8*; dotted line · · ·basic cubic spline

subband coefficients in Fig. 3.17—3.20. The points of every curve is the percentage of the number of pixels whose values are below the value of that point. We can see that cardinal spline is indeed a little superior than the others, especially in subbands of LLH and LHH; *bior6.8* and *db8* perform similarly; while basic cubic spline gives the worst (most spreaded) distribution.

3.4 Quantization

After obtaining the wavelet-transformed image, we will apply quantization and compression techniques by utilizing the statistical properties of different subband coefficients.

3.4.1 Quantization of Lowest Subband LLL

For the lowest subband, we will not apply any further compression techniques in this thesis. We just use their original 8-bit data, due to the fact that we need fixed-length bitstreams as much as possible for the noisy channel. If we have the noiseless condition, then we can use a variable-length coding scheme like DCT for LLL subband, in order to achieve much higher

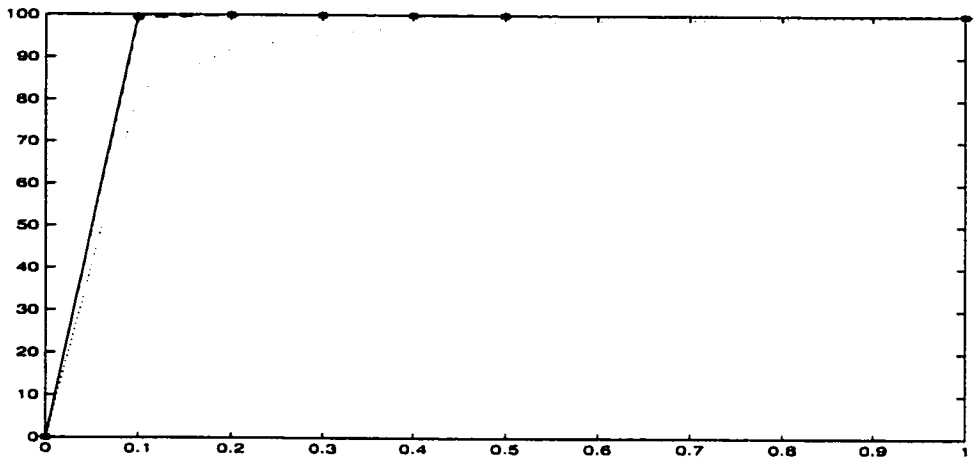

Figure 3.18: Percentage of distribution characteristics of subband HH. Solidline – cardinal spline; star line ** *bior6.8*; dashed line - - *db8*; dotted line ···basic cubic spline

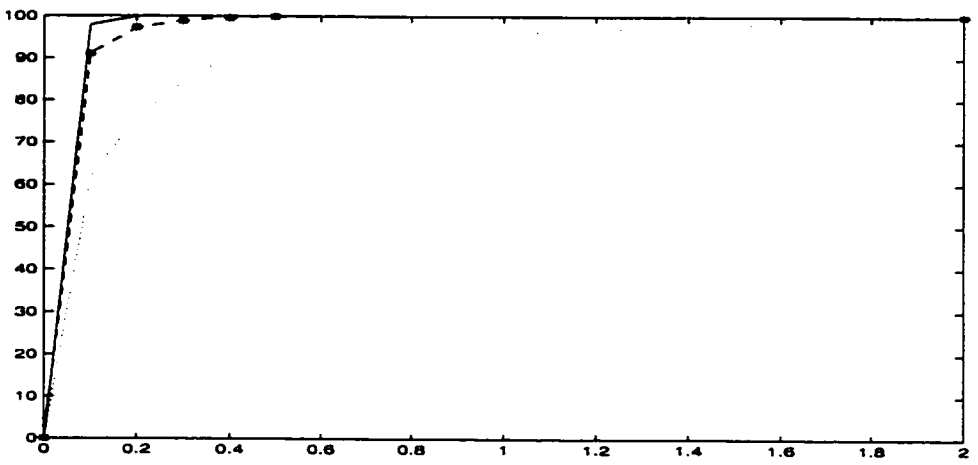

Figure 3.19: Percentage of distribution characteristics of subband LLH. Solid line –cardinal spline; star line ** *bior6.8*; dashed line - -*db8*; dotted line ··· basic cubic spline

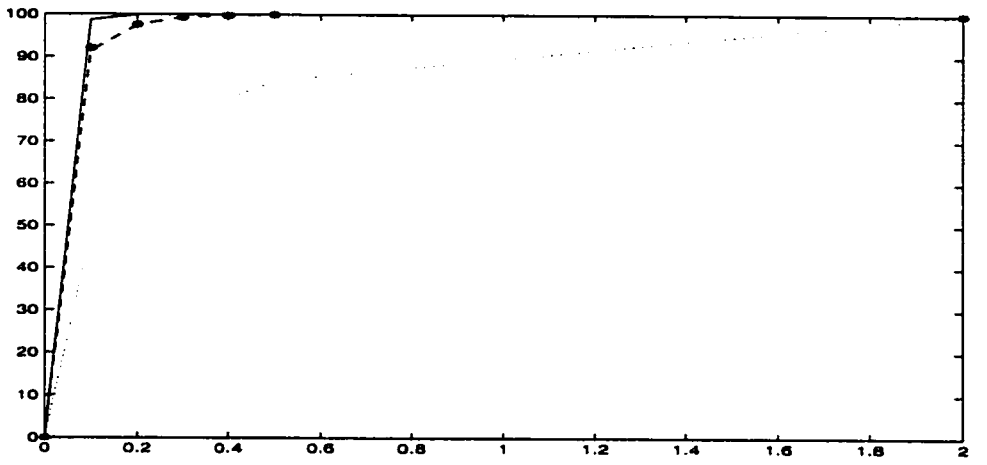

Figure 3.20: Percentage of distribution characteristics of subband LHH. Solid line – cardinal spline; star line ** *bior6.8*; dashed line – –*db8*; dotted line · · ·basic cubic spline

compression ratio.

3.4.2 Lattice Vector Quantization

Let $\mathbf{X}$ be an N-dimensional source vector with joint pdf $f(\mathbf{X}) = f(x_1, x_2, \ldots, x_N)$. An N-dimensional vector quantizer (VQ) is a function $Q(\mathbf{X})$ that maps $\mathbf{X} \in \mathbf{R}^N$ into one of L output points with each output point corresponding to an output vector $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_L$, belonging to $\mathbf{R}^N$. The performance of a VQ is completely determined by two quantities: the average distortion D and the required rate R. The original motivation for VQ is that as the block length N of the vector $\mathbf{X}$ increases, the performance of the source coder approaches the best performance possible, namely the *rate distortion* $R(D)$ which specifies the lowest output rate while keeping the distortion less than or equal to D, and $D(R)$ which specifies the minimum distortion at a rate of R or less, respectively when N tends to infinity.

The design of vector quantizers is generally based on one of two approaches, an iterative design procedure often called the LBG algorithm or the specification of uniform quantizers by using lattices. The former approach generates a locally optimal vector quantizer design, but the quantization/encoding problem may be formidable. The lattice-based approach can

greatly simplify the quantization operation, but the resulting quantizers are only optimal for uniformly distributed sources or asymptotically optimal as the number of output points becomes large [31].

An N-dimensional lattice is defined as a set of vectors

$$\Lambda = \{\mathbf{x} : \mathbf{x} = u_1\mathbf{a}_1 + u_2\mathbf{a}_2 + \cdots + u_N\mathbf{a}_N, u_i \in \mathbf{Z}\} \tag{3.37}$$

where $\mathbf{a}_i, i = 1, 2, \ldots, N$, are the basis vectors of the lattice and the u_i are integers. An N-dimensional VQ with outputs $\mathbf{Y}_1, \cdots, \mathbf{Y}_L \in \Lambda$ is the function $Q(\mathbf{X})$ which sends each point $\mathbf{X} \in \mathbf{R}^N$ into $Q(\mathbf{X})$ = closest codepoint $\mathbf{Y}_i$. Around each lattice point $\mathbf{Y}_i$ is its *Voronoi region* $V(\mathbf{Y}_i)$ defined as

$$V(\mathbf{Y}_i) = \{\mathbf{X} \in \mathbf{R}^N : \|\mathbf{X} - \mathbf{Y}_i\| \leq \|\mathbf{X} - \mathbf{Y}_j\| \text{for all } j \neq i\}$$

where $\|\mathbf{X}\| = (\mathbf{X} \cdot \mathbf{X})^{1/2}$. Thus if a source vector $\mathbf{X}$ belongs to ith Voronoi region, then $Q(\mathbf{X}) = \mathbf{Y}_i$. It has been shown that a quantizer will perform well if its Voronoi region approaches the shape of a hypersphere.

In lattice quantization, the following lattices are frequently used:

- A_n Lattice [32]

  A_n is the n-dimensional lattice consisting of the points having integer coordinates that sum to zero.

- Z_n Lattice [33]

  The n-dimensional Z_n lattice is the simplest of all lattices. It is comprised of all points in $\mathbf{R}^n$ whose coordinates are integers.

- D_n Lattice [33]

  The n-dimensional D_n lattice with $n \geq 2$ corresponds to those points in the Z_n lattice for which the sum of their coordinates is even.

- E_8 Lattice [33]

The eight-dimensional E_8 lattice is the union of two D_8 lattices with the following relationship:

$$E_8 = D_8 \cup \left(\frac{1}{2}[\mathbf{1}] + D_8\right)$$

where $[\mathbf{1}]$ is the vector whose components are all equal to 1.

- Λ_{16} Lattice

The sixteen-dimensional Λ_{16} lattice is the union of 32 D_{16} Lattices shifted by the translation vectors r_i:

$$\Lambda_{16} = \bigcup_{i=0}^{31} (r_i + 2D_{16})$$

These lattices and their duals give rise to the densest known sphere packing and they can be used as the basis for efficient block quantizers for uniformly distributed inputs and to construct codes for band-limited channel with Gaussian noise.

According to [34], D_4 is the best lattice in four dimensions. It has the densest sphere packing capability in $\mathbf{R}^4$. Therefore, for this thesis, we select the D_4 lattice VQ for the quantization of wavelet transformed coefficients.

[32] gives the fast algorithm for finding the closest point of the integer lattice $\mathbf{Z}^n$ to an arbitrary point $\mathbf{x} \in \mathbf{R}^n$. For a real number x, let

$$f(x) = \text{closest integer to } x$$

In the case of a tie, choose the integer with the smallest absolute value. For $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbf{R}^n$, let

$$f(\mathbf{x}) = [f(x_1), \ldots, f(x_n)]$$

We also define $g(\mathbf{x})$, which is the same as $f(\mathbf{x})$ except that the *worst* component of $\mathbf{x}$ - that furthest from an integer - is rounded the *wrong way*. In the case of a tie, the component with the lowest subscript is rounded the wrong way. More generally, for $x \in \mathbf{R}$ which is a component of $\mathbf{x}$, we have $f(x)$ which is the closest integer of x and $w(x)$ which rounds the

wrong way, then the procedure to find $f(x)$ and $w(x)$ is listed as follows (m is an integer):

$$\begin{aligned}
&\text{If } x = 0, && \text{then } f(x) = 0,\\
& && w(x) = 1.\\
&\text{If } 0 < m \le x \le m + \tfrac{1}{2}, && \text{then } f(x) = m,\\
& && w(x) = m + 1.\\
&\text{If } 0 < m + \tfrac{1}{2} < x < m + 1, && \text{then } f(x) = m + 1,\\
& && w(x) = m.\\
&\text{If } -m - \tfrac{1}{2} \le x \le -m < 0, && \text{then } f(x) = -m,\\
& && w(x) = -m - 1.\\
&\text{If } -m - 1 < x < -m - \tfrac{1}{2}, && \text{then } f(x) = -m - 1,\\
& && w(x) = -m.
\end{aligned} \tag{3.38}$$

Thus, from $\mathbf{x} = [x_1, \ldots, x_n]$ we got $f(\mathbf{x}) = [f(x_1), \ldots, f(x_n)]$ and $g(\mathbf{x}) = [f(x_1), \ldots, w(x_j), \ldots . f(x_n)]$ where the index j is determined by

$$j = \arg\max_i \mid x_i - f(x_i) \mid$$

With the above general algorithm, if given a vector $\mathbf{x} \in \mathbf{R}^4$, the closest point of D_4 is whichever of $f(\mathbf{x})$ and $g(\mathbf{x})$ has an even sum of components (one will have an even sum, the other an odd sum). If $\mathbf{x}$ is equidistant from two or more points of D_4 this algorithm produces a nearest point having the smallest norm.

In [1], it has been shown that pdf of subband coefficients can be approximated by a generalized Gaussian law:

$$p(x) = a \exp(-|bx|^r) \tag{3.39}$$

where a, b and r are subband related constants, with $r = 2$ leads to the Gaussian pdf. and $r = 1$ leads to a Laplacian pdf. And it is also shown that based on this pdf as well as LBG VQ, the optimal bit allocation scheme (for biorthogonal wavelet transformed subband coefficients) should be like that presented in Fig. 3.21. Although this bit allocation is derived from LBG VQ, we can also make use of its idea, namely, putting more bits to subbands LLH, LHL than that of subbands LHH, LH and HL, as well as ignore the wavelet coefficients in subband

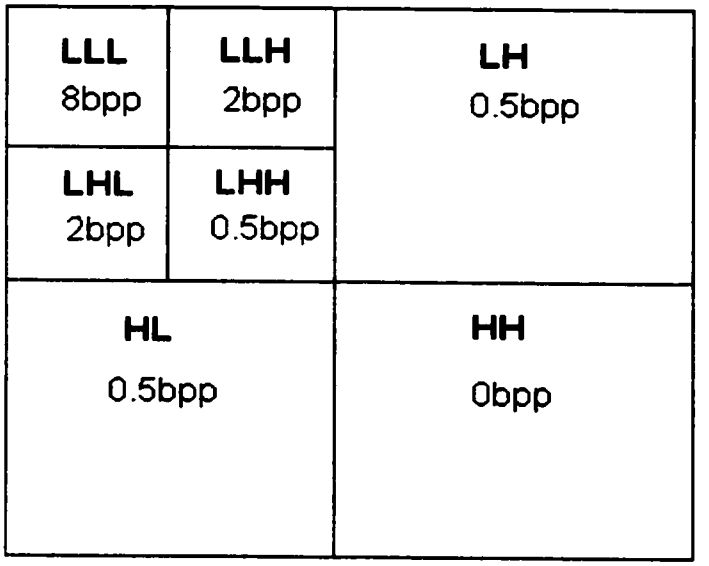

Figure 3.21: Subband bit allocation of [1].

HH, because the pdf of the same subband is similar for that of *bior6.8*, *db8* and cardinal cubic spline.

Before the quantization, we will scale and shift the subband coefficients to the value range of [0,3] so that after finding the nearest integer for every element in a 2×2 vector, we could use 2 bits to represent this integer. Thus after LVQ for a 2×2 subband coefficient vector, we get 8-bit codes which means 2bpp for every coefficient in one LVQ stage; if a second LVQ for the residue is applied, then there will be 4bpp for every pixel.

The bit allocation to different subbands for this thesis is given in Table 3.6. For the purpose of getting a more fixed length bitstream, we will not apply any optimization scheme which would lead to variable length codes, but just use the bit number listed in Table 3.6 for a subband coefficient selected to go through LVQ.

3.5 Simulation for Image Source Coding

In the simulation, assuming the dimension of subbands HL, LH and HH is $2m \times 2n$ and $m \times n$ for LLH, LHL and LHH, we scan the subbands as follows: (1)scan the 2×2 blocks of LLH and LHL in zig-zag format, whenever a pixel in the 2×2 block is above the threshold, then put all the four pixels in this block through LVQ and put the 8-bit LVQ result to a *code* stream, and add a "1" to a *coordinate* stream denoting that there are four pixels in the related coordinate quantized, otherwise add a "0" to the *coordinate* stream, thus generating two *coordinate* streams of length $m \times n$ for LLH and LHL respectively, plus a *code* bitstream of variable length; (2) scan the 2×2 block of LHH and its correspondent 8×8 blocks in LH and HL, if any pixels in these three blocks is above threshold, then put all these blocks through LVQ and the LVQ result to another *code* stream, and add a "1" to a third *coordinate* stream, otherwise add a "0" to the third *coordinate* stream, thus generating a third fixed length *coordinate* stream of length $m \times n$ and another variable length *code* bitstream.

We use this scanning and coding scheme to generate fixed length code as much as possible. Although ambiguity and error will still occur if the coordinate bits are decoded in a wrong way, the error will not propagate and stop the decoding process.

The performance for the three kinds of wavelets using the above coding scheme are shown in Fig. 3.22, in which the horizontal axis represents the *bit rate* in units of *bits-per-pixel (bpp)*, and the vertical axis represents the *peak signal to noise ratio (PSNR)* between the original image $x(m, n)$ and the decoded image $\hat{x}(m, n)$. If pixel values are in the range of $[0, 1]$, then

Table 3.6: Bit number for a coefficient in different subband

Subband	Bit Rates (bpp)
LLL	8
LLH, LHL	2 or 4
LHH, LH and HL	2 or 0
HH	0

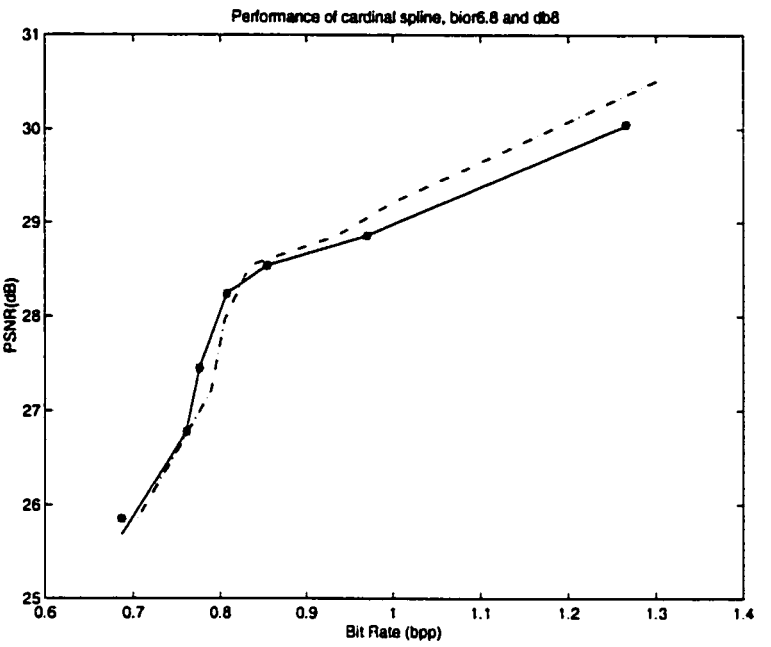

Figure 3.22: Performance of cardinal spline, *biorb.8* and *db8*. solid line - cardinal spline; dash dot —· *biorb.8*; star line * *db8*

the $PSNR$ is defined as:

$$PSNR = 10 \log_{10} \frac{1}{(M \times N)^{-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (x(m,n) - \hat{x}(m,n))^2} \tag{3.40}$$

where M and N are image dimensions. From the figure, we find that both cardinal spline and *db8* perform better than *bior6.8* in the low bit rate region (below 0.8bpp), and *bior6.8* outperforms the cardinal spline and *db8* when the bit rate is higher than 0.8bpp.

Chapter 4

Channel Model of CDMA Systems and Receiving Techniques

For a CDMA system, there are two kinds of transceiving links: one is the *uplink* (or *reverse link*) in which users transmit their respective signals to the base station using the same wireless frequency with the bandwidth equal to the bandwidth of the PN (pseudo-noise) sequence; another is the *downlink* (or *forward link*) in which the base station transmits the signals to different users using another wireless frequency with the bandwidth of the PN sequence. Because the bandwidth of the PN sequence is the same for each user, therefore, in the *uplink* all users' transmitting signals are mixed together in one wireless frequency bandwidth, and in the *downlink* there is another mixture of signals which the base station transmits to the users using another wireless frequency bandwidth. In this thesis we will use the model of the *uplink* for which the receiving technique is more complex than that of the *downlink*.

4.1 Channel Model

As we mentioned in section 2.3.1, the signal that a receiver of a CDMA system can get is a mixture of the users' signals from different paths with different fading coefficients plus the

channel noise which we rewrite as follows:

$$r_c(t) = \sum_{k=1}^{K} \sum_{l=1}^{L} \sqrt{2A_k} c_{kl}(t) s_k(t - \tau_{kl}) b_k(t - \tau_{kl}) \cos(\omega_c t + \varphi_{kl}) + n_c(t) \tag{4.1}$$

From this formula we find that after the demodulation with carrier frequency ω_c this mixture of different signals will be composed of four terms:

$$r(t) = e^{j\varphi_{11}} \sqrt{2A_1} c_{11}(t) s_1(t - \tau_{11}) b_1(t - \tau_{11}) + SMI + MAI + AWGN \tag{4.2}$$

where the first term can be regarded as the desired signal of user-1, SMI represents the *self multipath interference* which also comes from user-1 but through different paths:

$$SMI = \sqrt{2A_1} \sum_{l=2}^{L} e^{j\varphi_{1l}} c_{1l}(t) s_1(t - \tau_{1l}) b_1(t - \tau_{1l}) \tag{4.3}$$

and MAI is *multiple-access-interference* from other users:

$$MAI = \sum_{k=2}^{K} \sum_{l=1}^{L} e^{j\varphi_{kl}} \sqrt{2A_k} c_{kl}(t) s_k(t - \tau_{kl}) b_k(t - \tau_{kl}) \tag{4.4}$$

and $AWGN$ (additive white Gaussian noise) is just the white noise resulting from the wireless channel white noise $n_c(t)$ passing through the demodulator.

For the *downlink* of a CDMA system, MAI is minor and can be neglected because the base station can control the synchronization of different users' spreading waveforms (PN codes) so that the cross-correlation with other users' PN codes induced when a user is receiving the signal is negligible. However, for the *uplink* it is impossible for different users to send their own signals with their respective PN codes synchronized, so there exists MAI when the base station processes the signals from the *uplink*.

The MAI in the *uplink* can be regarded as $AWGN$ when the number of users that use their CDMA devices at the same time is not large. This is often the case for most of the CDMA applications now. However, in some metropolitan areas, sometimes there is a large number of users transmitting signals to the same base station. In this case, MAI will become more or less impulsive in nature [22].

In this thesis we will only consider the *uplink* and assume the worst case for MAI by deliberately introducing the maximum cross-correlation between the PN codes of different users.

4.2 Rake Receiver

The Rake receiver is most commonly used in commercial CDMA systems currently. Before going through its principles, we first give some definitions:

(1)T_m denotes the total multipath delay of the channel.

(2)B_d denotes the Doppler spread of the channel.

(3)*Coherent bandwidth* W_c: Two sinusoids with frequency separation greater than W_c are affected differently by channel, where

$$W_c \approx \frac{1}{T_m} \tag{4.5}$$

(4)*Coherent time* T_c: Two identical signals separated by a time delay larger than T_c will be affected independently, where

$$T_c \approx \frac{1}{B_d} \tag{4.6}$$

(5)*Transmission bandwidth* W is the chip rate $1/T_s$ of the spreading waveform or PN sequence.

(6)*Information bandwidth* W_i is $1/T_b$ where T_b is the data bit time interval.

For the uncoded direct sequence BPSK (DS/BPSK) signal one data bit is transmitted every T_b seconds. Assume that

$$T_b \gg T_m \tag{4.7}$$

so that intersymbol interference between data bits can be ignored. Also assume that the channel is slowly varying so that

$$T_b \ll T_c. \tag{4.8}$$

Thus the channel disturbance is almost constant during a data bit time T_b. Finally, since our signal is a wideband signal of bandwidth W, assume

$$W \gg W_c. \tag{4.9}$$

Thus, overall, we have $W \gg W_c \gg W_i$. Since the received signal $r_c(t)$ has bandwidth W centered at carrier frequency ω_c, the demodulated signal $r(t)$ can be modelled as [35]:

$$r(t) = \sum_{n=1}^{L} \alpha_n e^{j\varphi_n} x\left(t - \frac{n}{W}\right) + n(t) \tag{4.10}$$

for the Rake receiver, where

$$L = WT_m + 1 \tag{4.11}$$

is the number of delayed versions of the signal sampled at time intervals $\{n/W : n = \ldots, -1, 0, 1, \ldots\}$, and $x(t)$ is simply the data bit signal of a user spread by its own PN sequence

The parameters α_n and φ_n are amplitude and phase coefficients of nth delayed path.

Comparing (4.10) with (4.2) we find that the model of the received signal is simplified by deleting the MAI term from (4.2). That is because the Rake receiver is based on the assumption that the cross-correlation with other users' PN sequence is trival and thus MAI can be reduced to $AWGN$. This assumption makes for an efficient implementation structure for the Rake receiver, but it also means that it can not effectively handle impulsive MAI when the number of users is very large.

Because the delayed signals in $r(t)$ go through different propagation paths, we can treat their fading as independent. The Rake receiver utilizes this inherent diversity to estimate the data bits $b(t)$ being transmitted.

The ideal Rake receiver is shown in Fig. 4.1. Here we assume the receiver can have complete knowledge of the amplitude and phase terms $\alpha_n e^{-j\varphi_n}$. When the fading is slow this estimate is quite good. The nth branch's output is

$$r(t + \frac{n}{W})s(t)\alpha_n e^{-j\varphi_n} \tag{4.12}$$

and at the end the Rake receiver compares

$$\int_0^{T_b} \left[\sum_{n=1}^{L} r(t + \frac{n}{W})s(t)\alpha_n e^{-j\varphi_n} \right] dt \tag{4.13}$$

with zero to make binary decision for data bit $\hat{b}_0$.

However, it is impossible for the Rake receiver to know the value of $\alpha_n e^{-j\varphi_n}$ in advance. Thus we have to estimate $\alpha_n e^{-j\varphi_n}$ which is required in a practical Rake receiver. Note that

$$\begin{aligned} r\left(t + \tfrac{l}{W}\right) s(t) &= s(t) \textstyle\sum_{n=1}^{L} \alpha_n e^{j\varphi_n} s\left(t - \tfrac{n-l}{W}\right) b\left(t - \tfrac{n-l}{W}\right) + s(t)n(t + \tfrac{l}{W}) \\ &= \alpha_l e^{j\varphi_l} b(t) + s(t)n(t + \tfrac{l}{W}) \\ &+ s(t) \textstyle\sum_{\substack{n=1 \\ n \neq l}}^{L} \alpha_n e^{j\varphi_n} s\left(t - \tfrac{n-l}{W}\right) b\left(t - \tfrac{n-l}{W}\right) \end{aligned} \tag{4.14}$$

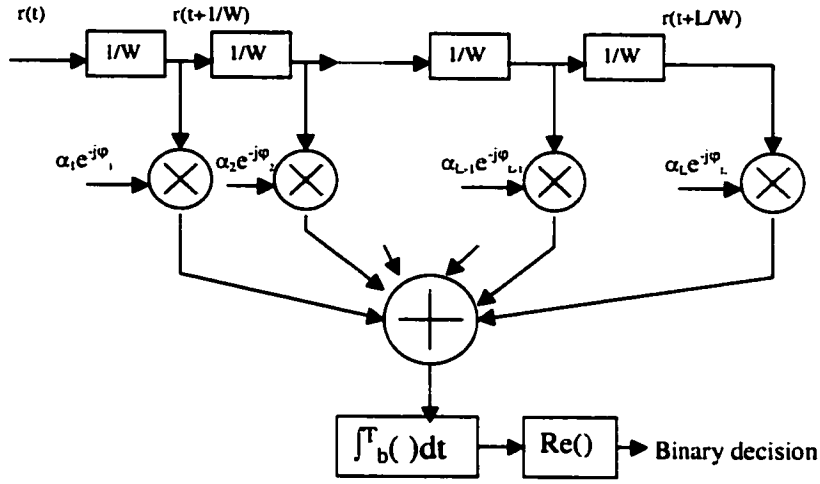

Figure 4.1: Ideal Rake receiver

and because $s(t)$ is independent of $s\left(t-\frac{n-l}{W}\right)$ for each value of t when $n \neq l$

$$\int_0^{T_b} r\left(t+\frac{l}{W}\right) s(t)dt = \alpha_l e^{j\varphi_l} b_0 \cdot T_b + n_l \tag{4.15}$$

where b_0 is the data bit in $(0, T_b)$ and n_l is Gaussian noise. This suggests that the estimate for $\alpha_n e^{-j\varphi_n}$ be given by the conjugate of (4.15). This estimate, however, includes the data bit b_0. Assuming $\alpha_n e^{-j\varphi_n}$ remains unchanged over $2T_b$ seconds, an estimate can be based on the previous T_b-second channel output signal. This estimate is shown in the complex form of the Rake receiver illustrated in Fig. 4.2.

4.2.1 1-D Simulation for the Rake Receiver

We use two original 1-D signals as two user signals to test the performance of the Rake receiver. The two signals are shown in Fig. 4.3. Every sample of the signal is quantized with eight bits, forming the binary data signal $b(i)$, and then every data bit is spread by a PN sequence of length 31. The cross-correlation of the PN sequences for two users is 7/31. At the receiving side, we will try to recover the signal of user-1.

For the transmission, we assume there are three different paths for every user, and the delay between the first path with the second (third) path is one chip (two chips) of the

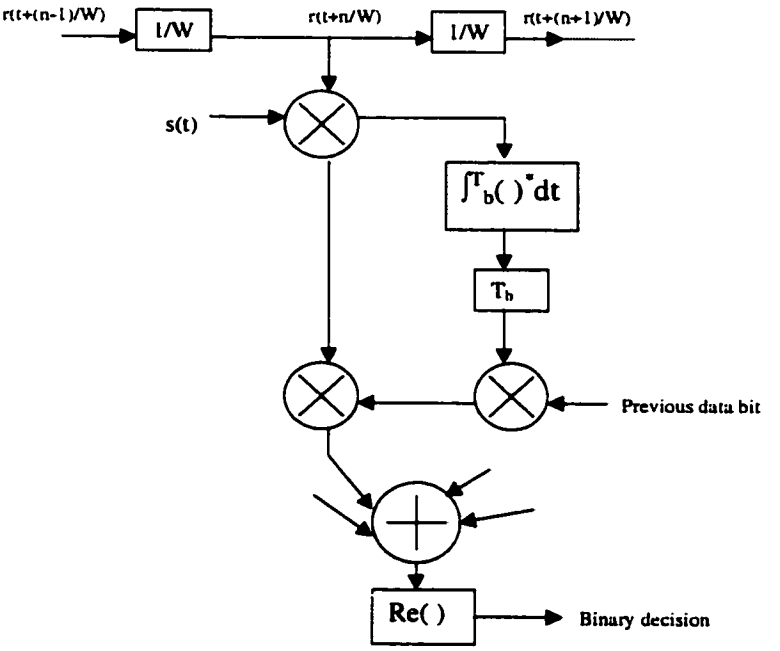

Figure 4.2: One branch of practical Rake receiver

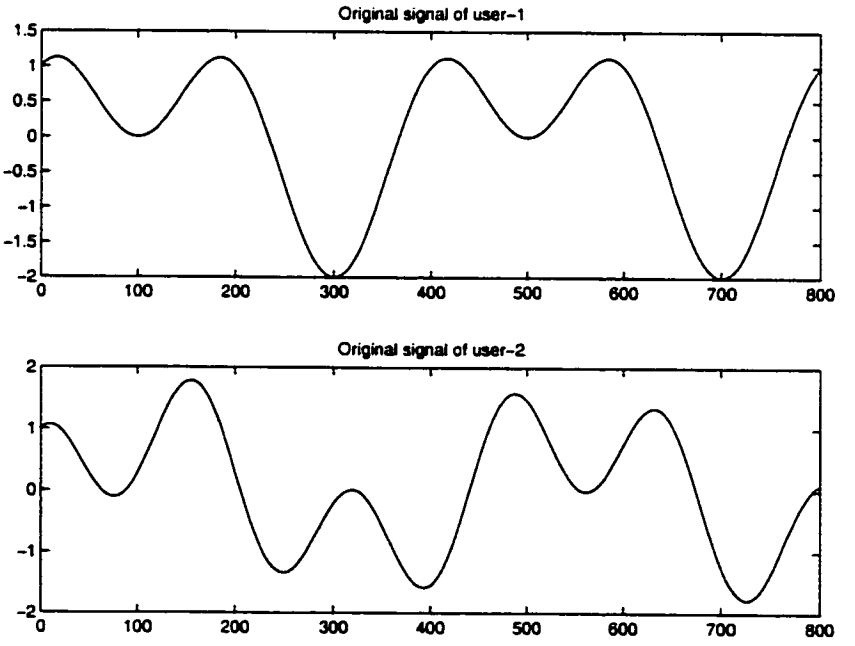

Figure 4.3: Two original 1-D user signals

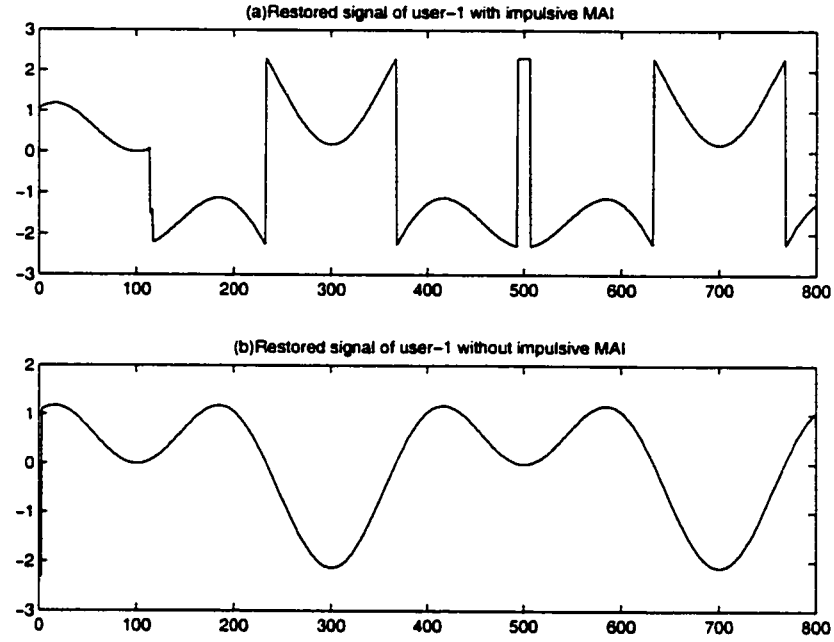

Figure 4.4: Restored signal of user-1

PN sequence. Since we assume BPSK, there should be two independent fading coefficients for each path. The channel coefficients that approximate the statistical properties of the Rayleigh fading model can be generated by feeding two independent real iid (independently identically distributed) Gaussian processes to two identical 3rd-order Butterworth filters [36]. The 3-dB bandwidth f_D of the filter normalized to the symbol rate $1/T_b$ is used as a measure of the fading rate. We assume the carrier frequency is 1800 MHz and the symbol rate $1/T_b$ is 64 Kb/s. f_D can be calculated as

$$f_D = v/\lambda \tag{4.16}$$

where v is the mobile speed and λ is the wavelength of the carrier frequency. We assume the velocity $v = 50Km/h$, which corresponds to f_D of 75 Hz.

Fig. 4.4(a) shows the restored signal of user-1, we can see that there is indeed impulsive interference. To compare, we put in Fig. 4.4(b) the same signal restored but with another user's PN sequence having minimum cross-correlation "-1/31". We can see that, without impulsive MAI, the Rake receiver works very well.

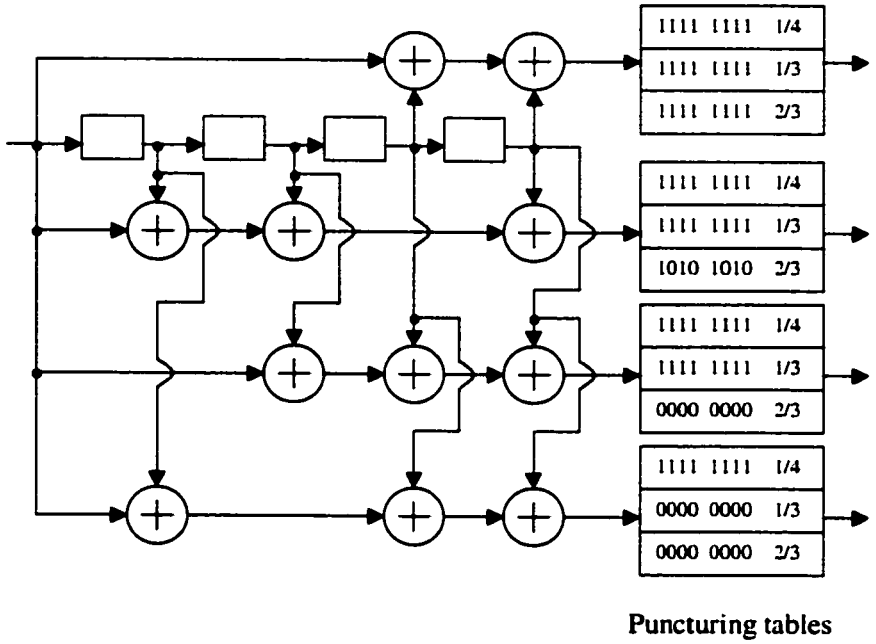

Figure 4.5: RCPC encoder

4.3 Application of RCPC with the Rake Receiver

Now we will apply rate compatible convolutional codes (RCPC) to the coding/decoding of 1-D signals using the Rake receiver. The RCPC we will use in this thesis is based on a rate-1/4 convolutional encoder, and the two subrates we are using are 1/3 and 2/3. The structure of RCPC and the related puncturing matrix are shown in Fig. 4.5.

For a 1-D signal quantized with eight bits, we will allocate rate-1/4 to $1st$ and $2nd$ MSBs, rate-1/3 to $3rd$ - $5th$ MSBs, and rate-2/3 to the last three LSBs. Because the RCPC in Fig. 4.5 has four memories, the constraint length is 5. The block length we are using is 64, and the *decoding depth* is 20 in the related Viterbi decoder.

The restored 1-D signal of user-1 is shown in Fig. 4.6. Comparing to the original signal in Fig. 4.3 and the restored signal without RCPC in Fig. 4.4(a), we can see that RCPC can help to recover the approximate shape of the original 1-D signal from the impulsive interference, but at the cost of destroying most of the "*detailed*" shape information of the waveform. This is because convolutional coding is mainly used for Gaussian white noise. If

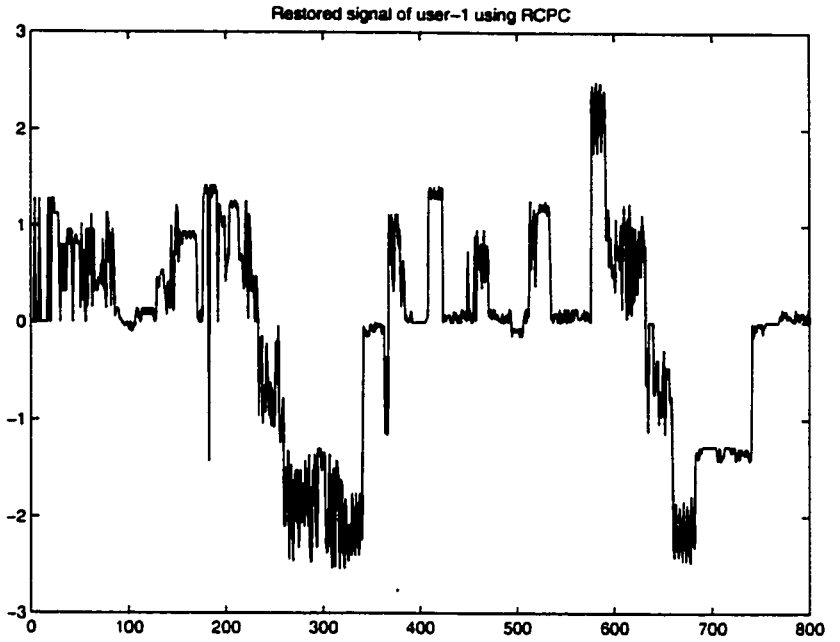

Figure 4.6: Restored signal of user-1 using RCPC

used for an impulsive noise environment, it should be combined with bit interleaving, as used in CDMA voice communication now, so as to disperse the impulsive interference to become more or less AWGN like.

4.4 Adaptive Multiuser Detection

Along with the development of adaptive signal processing, some new algorithms have been proposed for the receiving of CDMA signals using the complete channel model of (4.2) including MAI, rather than assuming the MAI as part of AWGN as in Rake receiver.

We will try a recently developed algorithm from [36] and [37] in which the channel model used is

$$r(t) = \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{i=0}^{M-1} e^{j\varphi_{kl}} \sqrt{2A_k} b_k[i] P_{T_b}(t - iT_b - \tau_{kl}) c_{kl}(t) s_k(t - \tau_k - \frac{l-1}{W}) + n(t) \qquad (4.17)$$

where M is the number of data bits transmitted. We can see that, compared to the channel model in (4.10) for the Rake receiver, as well as the original channel model in (4.2), the

algorithm in [36] and [37] considered all the interference including MAI and SMI.

Suppose the carrier phase φ_{11} for the first path of user-1 is known, then after removing this phase from the received signal, the resulting signal during the ith signaling interval is $\tilde{r}(t) = r(t)e^{-j\varphi_{11}}$, where for $iT_b \leq t - \tau_{11} < (i+1)T_b$,

$$\begin{aligned}\tilde{r}(t) = &\sqrt{2A_1}b_1[i]\sum_{l=1}^{L} e^{j(\varphi_{1l}-\varphi_{11})}c_{1l}(t)s_{1l}(t-iT_b) \\ &+\sum_{k=2}^{K}\sqrt{2A_k}b_k[i]\sum_{l=1}^{L} e^{j(\varphi_{kl}-\varphi_{11})}c_{kl}(t)s_{kl}(t-iT_b) + n(t)e^{-j\varphi_{11}}\end{aligned} \tag{4.18}$$

and where

$$s_{kl}(t-iT_b) = s_k\left(t - \tau_k - \frac{l-1}{W} - iT_b\right) \tag{4.19}$$

Writing the real and imaginary components of (4.18) separately, we get

$$\begin{bmatrix}\tilde{r}^R(t)\\ \tilde{r}^I(t)\end{bmatrix} = \sqrt{2A_1}b_1[i]\sum_{l=1}^{L}\begin{bmatrix}c_{1l}^R(t)s_{1l}(t-iT_b)\\ c_{1l}^I(t)s_{1l}(t-iT_b)\end{bmatrix} + \sum_{k=2}^{K}\sum_{l=1}^{L}\sqrt{2A_k}b_k[i]\begin{bmatrix}c_{kl}^R(t)s_{kl}(t-iT_b)\\ c_{kl}^I(t)s_{kl}(t-iT_b)\end{bmatrix} + \begin{bmatrix}\tilde{n}^R(t)\\ \tilde{n}^I(t)\end{bmatrix} \tag{4.20}$$

In what follows, we consider using the minimum-mean-square-error MMSE filter to process the real component of $\tilde{r}(t)$. The application to its imaginary component follows in exactly the same way. Before we go through the details, we need to briefly introduce the theory of *blind adaptive multiuser detection* [38] on which the new adaptive algorithm is based.

The blind multiuser detector for CDMA systems can be decomposed into two orthogonal components. One component is equal to the *signature waveform* or PN codes s_1 of the desired user which is assumed known and fixed; another component is a *canonical component* x_1 where x_1 is orthogonal to s_1

$$\langle s_1, x_1\rangle = \int_0^{T_b} s_1(t)x_1(t)dt = 0. \tag{4.21}$$

The idea of MMSE filter for blind multiuser detection is to get the filter output $\langle r, s_1 + x_1\rangle$ which minimizes

$$E\Big[(b_1[i] - \langle r, s_1 + x_1\rangle)^2\Big] \tag{4.22}$$

This can be reduced to the minimization of

$$E\Big[(\langle r, s_1 + x_1\rangle)^2\Big] \tag{4.23}$$

because

$$E\Big[(b_1[i] - \langle r, s_1 + x_1\rangle)^2\Big] = E\Big[(b_1[i])^2\Big] + E\Big[(\langle r, s_1 + x_1\rangle)^2\Big] - 2E\Big[(b_1[i])^2\langle s_1, s_1 + x_1\rangle\Big]$$
$$= E\Big[(\langle r, s_1 + x_1\rangle)^2\Big] - E\Big[(b_1[i])^2\Big] \tag{4.24}$$

Thus, for (4.20), the MMSE filter corresponding to the real components of the lth path of the first user has the canonical form as $s_{1l} + x_{1l}^R$, and

$$\begin{aligned}\langle \tilde{r}^R, s_{1l} + x_{1l}^R\rangle &= \sqrt{2A_1}b_1[i]c_{1l}^R(t) \\ &+\sqrt{2A_1}b_1[i]\textstyle\sum_{j\neq l}^{L} c_{1j}^R(t)\Big[\rho_{(1,l)(1,j)} + \langle x_{1l}^R, s_{1j}\rangle\Big] \\ &+\textstyle\sum_{k=2}^{K}\sum_{j=1}^{L}\sqrt{2A_k}b_k[i]c_{kl}^R(t)\Big[\langle s_{1l}, s_{kj}\rangle + \langle x_{1l}^R, s_{kj}\rangle\Big] \\ &+\langle s_{1l} + x_{1l}^R, \tilde{n}^R\rangle\end{aligned} \tag{4.25}$$

The right hand side of (4.25) consists of four terms: the first term is the channel gain of the lth path; the second term comprises the SMI at the output of the correlation; the third term comprises the MAI at the output of the correlation; and the last is the ambient noise. The multipath MMSE filter chooses x_{1l}^R to eliminate the SMI and minimize the mean-square value of the MAI and ambient noise, i.e.:

$$\text{minimize} \qquad E\Big\{\Big[b_1[i] - \langle \tilde{r}^R, s_{1l} + x_{1l}^R\rangle\Big]^2\Big\} \tag{4.26}$$

In [36] and [37], the exponentially windowed RLS algorithm was used to select the weight vectors $\mathbf{W}^R(n) \equiv [w_1^R \cdots w_L^R]$ to minimize the sum of exponentially weighted output energy:

$$\text{minimize} \qquad \sum_{i=1}^{n}\lambda^{n-i} \parallel \mathbf{W}^R(n)^T\mathbf{r}^R(i) \parallel^2 \tag{4.27}$$

where $w_l^R(i) \equiv s_1 + x_l^R(i)$ is the tap weight vector at time i of the MMSE filter for the lth path, and $0 < \lambda < 1$ is the forgetting factor $(1 - \lambda \ll 1)$.

4.4.1 Simulation Results for RLS MMSE Receiver

The simulation result with the same fading and MAI conditions of Rake receiver (correlation of PN codes is -7/13) is shown in Fig. 4.7(a). We could find that, compared with the result

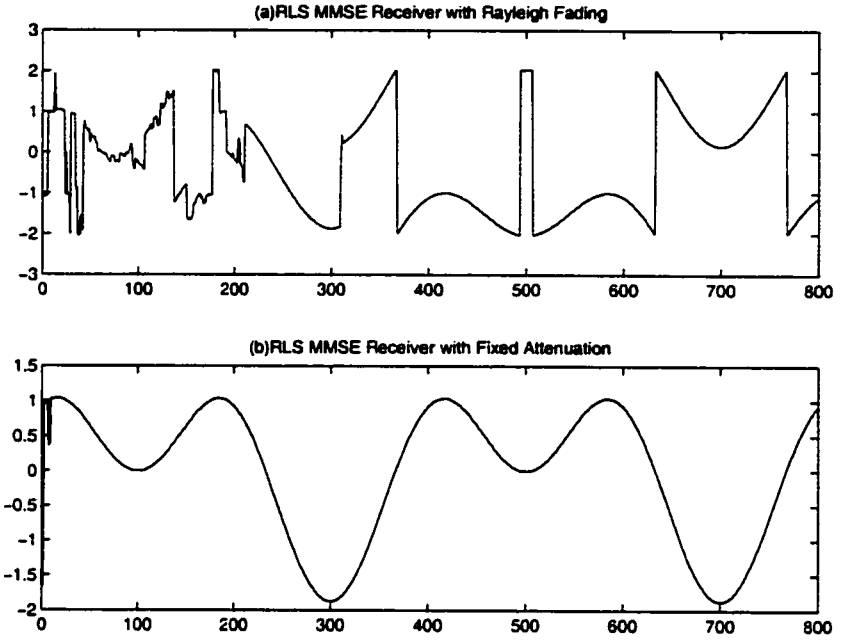

Figure 4.7: Restored signal of user-1 using RLS MMSE receiver under impulsive MAI

of the Rake receiver in Fig. 4.4(a), it has not much improvement. However, when the fading coefficients used in the simulation are fixed (fixed attenuation), and still with impulsive MAI, the RLS MMSE receiver can recover the original signal almost perfectly except for some convergence period at the beginning for RLS adaptation, as is shown in Fig. 4.7(b). This feature of the RLS MMSE receiver outperforms that of the Rake receiver, which could not work well in impulsive MAI even with fixed attenuation. The fixed attenuation can be applied to some fixed wireless applications. Thus we will use RLS MMSE receiver with fixed attenuation and impulsive MAI in the final simulation of this thesis, as the case of fixed CDMA wireless applications. We will not use RCPC for this case because the RLS MMSE receiver can recover the original signal perfectly in this situation with either no white noise or moderate white noise.

Chapter 5

Simulation Schemes and Final Results

In this chapter we describe how to put the coded images through CDMA channels, and use the Rake receiver as well as RLS MMSE receiver to restore the coded image.

For the transmitter (users) side, we consider two scenarios: one is shown in Fig. 5.1 in which two images - *Lena* and *cameraman* - are wavelet transformed, coded using LVQ, transmitted with RCPC and spread by respective PN codes, and finally mixed together to go into the channel; another is shown in Fig. 5.2 which is the same as in Fig. 5.1 except that RCPC is not used. We will use this transmitter scenario for the Rake receiver with non-impulsive MAI, or the RLS MMSE receiver with impulsive MAI.

For the receiver on the wireless base-station side, we will try to restore the image *Lena*. Related to the two transmitter scenarios, there are two receiving scenarios. One (scenario A)

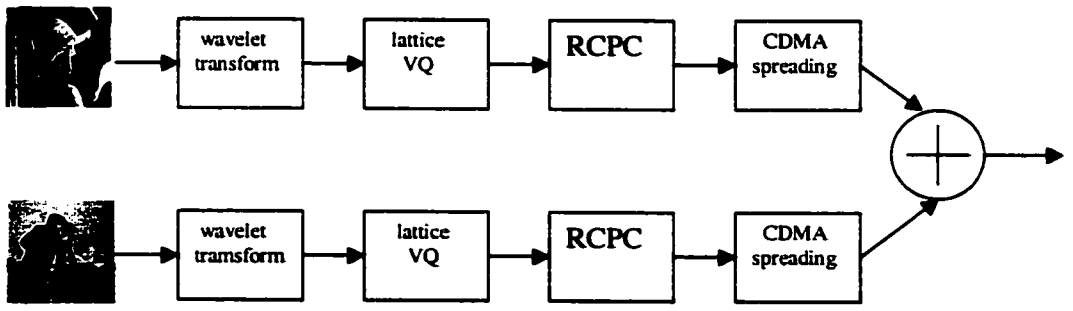

Figure 5.1: Transmitter scenario A with impulsive MAI

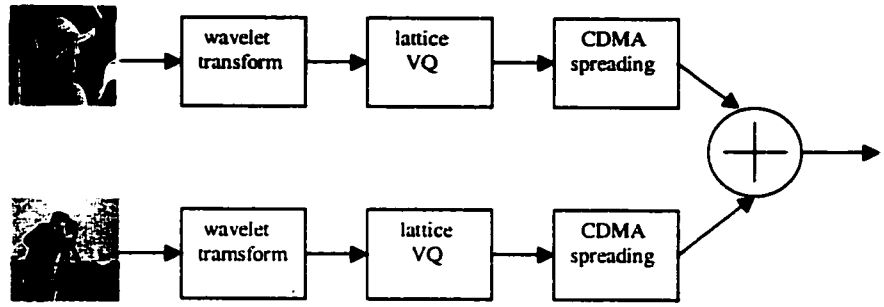

Figure 5.2: Transmitter scenario B with non-impulsive MAI

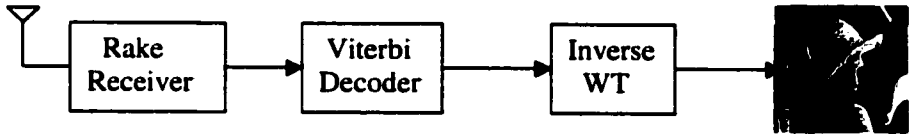

Figure 5.3: Receiving scheme for transmitter scenario A

is shown in Fig. 5.3 with the Rake receiver, Viterbi decoder and inverse wavelet transform; this scenario is for transmitter with RCPC. Another (scenario B) is shown in Fig. 5.4 with the Rake or RLS MMSE receiver without Viterbi decoder; this scenario is for the transmitter with non-impulsive MAI using the Rake receiver, or for the transmitter with impulsive MAI but the channel attenuation coefficients are fixed (which means no Rayleigh fading effects) using the RLS MMSE receiver. Both cases in scenario B do not use RCPC.

For every coded image, we have two fixed length bitstreams: one is the original 8-*bpp* bitstream of LLL subband; another is the coordinate scanning bitstream representing whether the related 2×2 or 4×4 blocks in the scanned subband have pixel values above threshold. We also have a variable-length bitstream containing the LVQ results of any blocks which

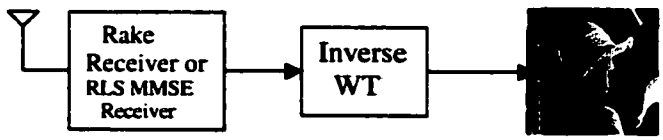

Figure 5.4: Receiving scheme for transmitter scenario B

have pixel values above threshold. If we are using RCPC, the first fixed-length bitstream will be assigned to *rate-1/4*, while the second one — the coordinate scan — will be assigned to *rate-1/3*, and the variable-length bitstream will be assigned to *rate-2/3*.

For the CDMA wireless channels, the signal-to-noise ratio is defined as [36]

$$SNR = 10\log_{10}\left\{\frac{2A_1}{|\sigma|^2}\sum_{l=1}^{L} E|c_{1l}(i)|^2\right\} \tag{5.1}$$

In the thesis, this value will be 2.93 dB. We assume that there are three different paths with one chip (two chips) delay between first path and second (third) path for each user, and each path has independent Rayleigh fading effect.

All the simulation were implemented on a PC with Pentium II 333MHz processor. The wavelet transforms of images were implemented by Matlab, other parts of the simulation program including RCPC, Rake receiver, Viterbi decoder as well as RLS MMSE receiver were implemented by C++.

5.1 Simulation Results for the Rake Receiver with Non-Impulsive MAI

The recovered LLL subband for the three kinds of wavelet transforms after transmission with non-impulsive MAI and Rake receiving is shown in Fig. 5.5. To compare their performance, we first show the finally restored image using these LLL subbands plus their respective *original* higher subband coefficients in Fig. 5.6-5.8. These images can be seen as the upper bound of performance for the restored images based on restored LLL subbands and transmitted coordinate and LVQ bitstreams. From these figures, we could see that there are interference which look like a white-bar in similar locations of each of the three images. The reason for it should be that the channel fading coefficients used in transmitting each image are the same for the three kinds of wavelets, while for each transmission there are twelve groups of fading coefficients for the three paths of each of the two users. These coefficients were generated by filtering random numbers of Gaussian distribution with a low-pass filter, thus for certain segments of the transmission the overall interference caused by the fading effect might be more

Figure 5.5: Received LLL subband using Rake with non-impulsive MAI

severe than for other segments. And the LLL subband was transmitted column-by-column. Therefore there are white-bar like interference around the similar area of each of the three images.

Next we show the performance diagram of three wavelets in Fig. 5.9. The image was coded by scanning the three subbands of LLH, LHL and LHH, then putting the LVQ results together with the *coordinate bitstream* and LLL subband directly to the channel, and using Rake as receiver. We can see that, without RCPC and impulsive MAI, *bior6.8* has the best performance.

The simulation program (C++) for scenario B runs for 3 seconds.

5.2 Simulation Results of Rake Receiver with Impulsive MAI

Once we deliberately introduced impulsive MAI, the image restored was severely damaged. To show the effect, we put a restored image *Lena* using transmitting scenario B with Rake receiver in Fig. 5.10, in which we only transmitted LLL subband, and using the original higher subbands coefficients for restore. Compared with the 1-D case in Fig. 4.4(a), we could find that they have similar phenomena: certain segments of the signal might be "*shifted*" vertically while keeping the same shape.

Figure 5.6: *bior6.8*, 14.44 dB

Figure 5.7: *db8*, 12.56 dB

Figure 5.8: cardinal spline, 11.78 dB

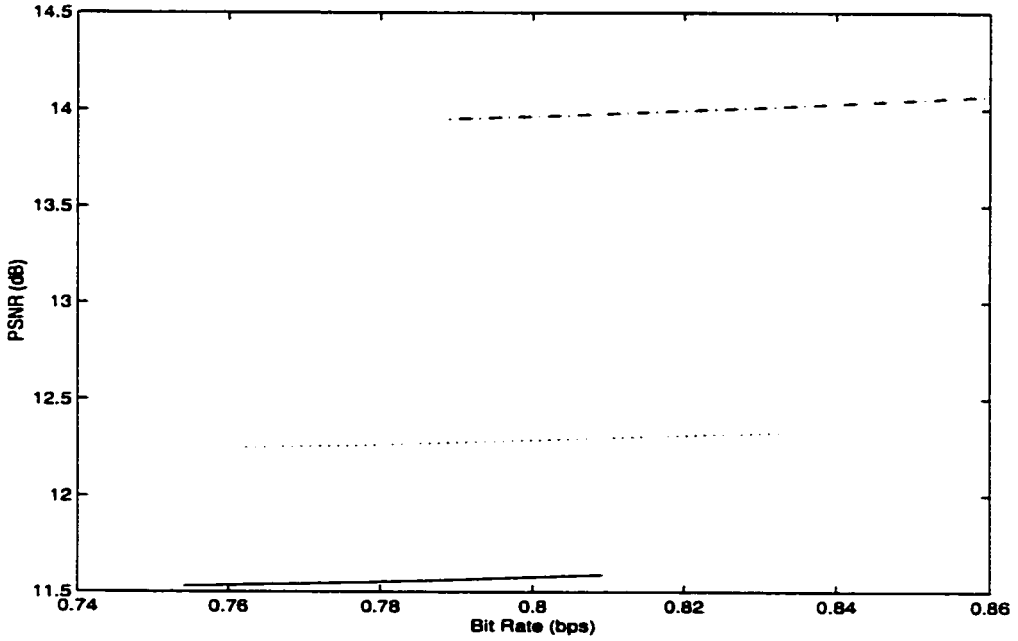

Figure 5.9: Performance of cardinal spline, *bior6.8* and *db8* for scenario B with Rake receiving.solid line - cardinal spline; dash dot —· *bior6.8*; dashed line —— *db8*

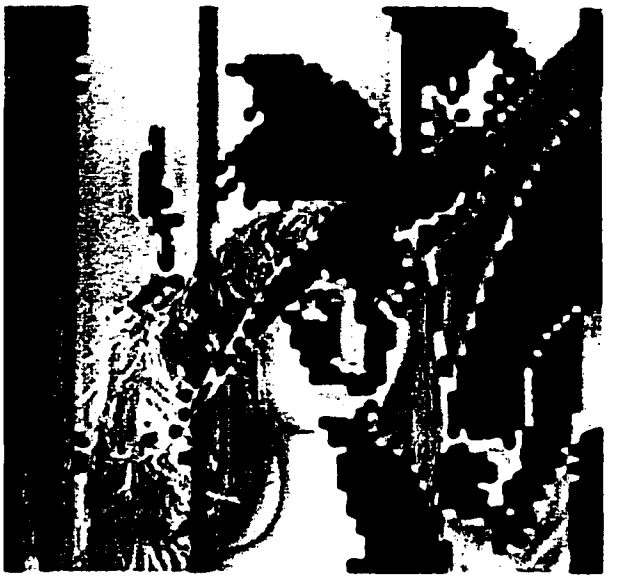

Figure 5.10: Restored *Lena* from transmitted LLL subband (*bior6.8*), 7.36 dB

Therefore, to handle this severe interference, we use RCPC as in scenario A before transmission, and use Viterbi decoding after Rake receiving. We show the finally restored images using these restored LLL subbands plus their respective *original* higher subbands coefficients in Fig. 5.11-5.13. From these figures, we could see that *db8* has the best performance, while *bior6.8* performs the worst. These images can be seen as the upper bound of performance for the restored images based on restored LLL subbands and transmitted coordinate and LVQ bitstreams.

Next we show the performance data of three wavelets using RCPC in Table 5.1-5.3. These data of performance were obtained by putting LLL subband as well as *coordinate* and *LVQ coded* bitstreams through RCPC using their respective coding rates, and using Rake receiver and Viterbi decoder at the receiving side. For comparison and to show the improvement, the performance data of the three wavelets without RCPC and Viterbi decoder but with impulsive MAI are also shown in each table respectively.

Comparing these data, we could also find that *db8* has the best performance, while *bior6.8* performs the worst.

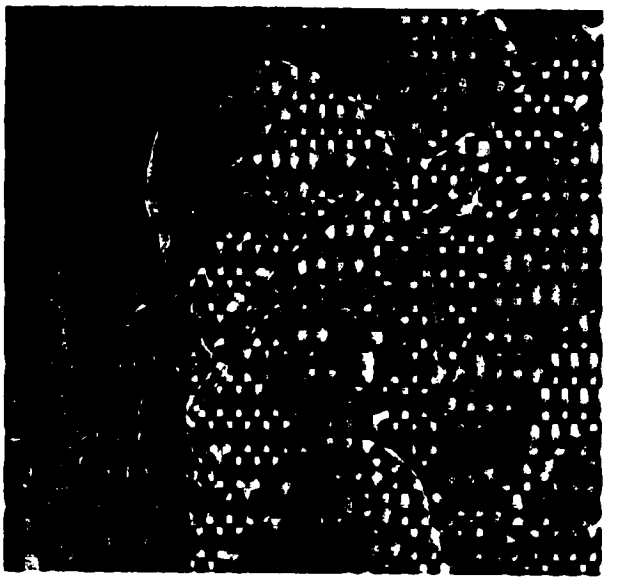

Figure 5.11: *bior6.8*, 13.46 dB

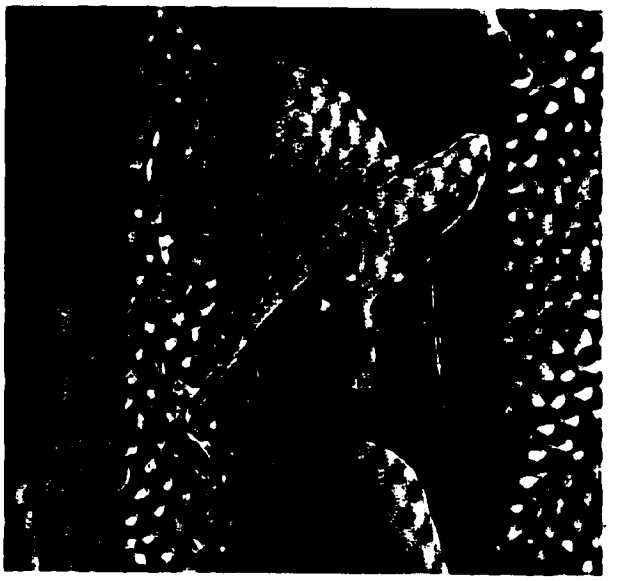

Figure 5.12: *db8*, 15.14 dB

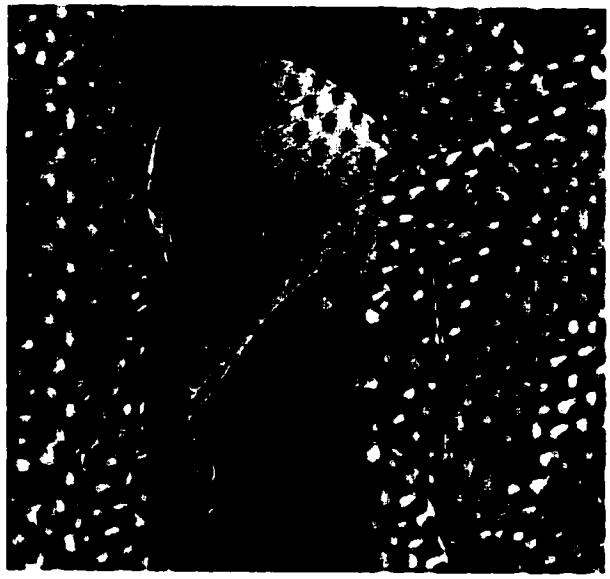

Figure 5.13: cardinal spline, 14.21 dB

Table 5.1: Performance of *bior6.8*

Bit Rate	0.789	0.805	0.835	0.86
PSNR(with RCPC)	12.98	13.03	13.12	13.18
PSNR(without RCPC)	7.29	7.28	7.33	7.33

Table 5.2: Performance of *db8*

Bit Rate	0.761	0.776	0.808	0.834
PSNR(with RCPC)	14.46	14.51	14.65	14.74
PSNR(without RCPC)	7.27	7.26	7.25	7.23

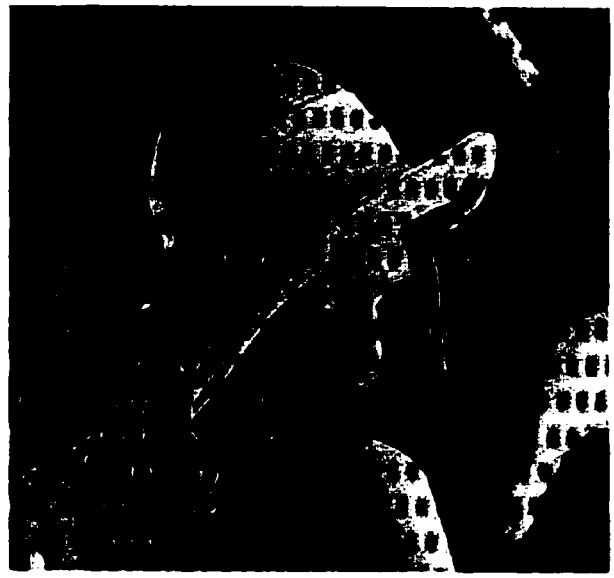

Figure 5.14: *bior6.8* with only one path per user, 19.43 dB

In order to find out how severe interference the MAI will bring to the finally restored images, we reduce the path number for each user in the transmission channel to only one path per user, and ran the simulation again using the same scheme as that of Fig. 5.11-5.13 for scenario A. The results are shown in Fig. 5.14-5.16. We can see that, although the PSNR performance of the restored images are improved, the severe impulsive-like noise still exists in each image. This is because convolutional coding and Viterbi decoding are mainly used to deal with white noise. For the communication channel having white noise as well as impulsive noise, Reed-Solomon coding or the concatenation of Reed-Solomon and convolutional coding are preferred.

Table 5.3: Performance of cardinal spline

Bit Rate	0.754	0.777	0.786	0.809
PSNR(with RCPC)	13.63	13.7	13.73	13.84
PSNR(without RCPC)	5.83	5.86	5.86	5.8

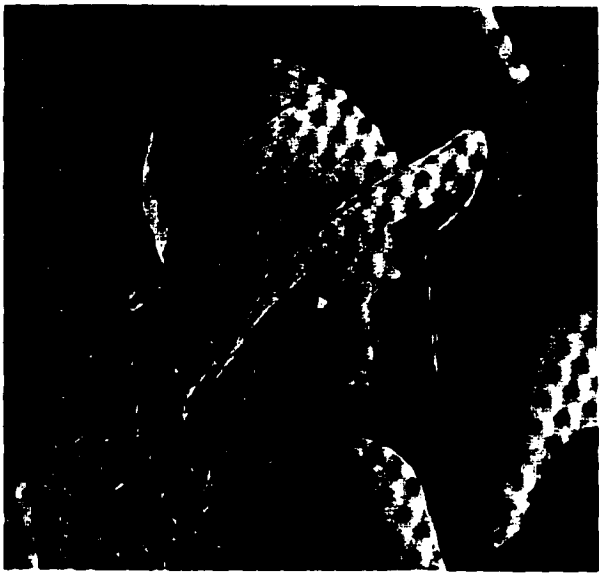

Figure 5.15: *db8* with only one path per user, 20.13 dB

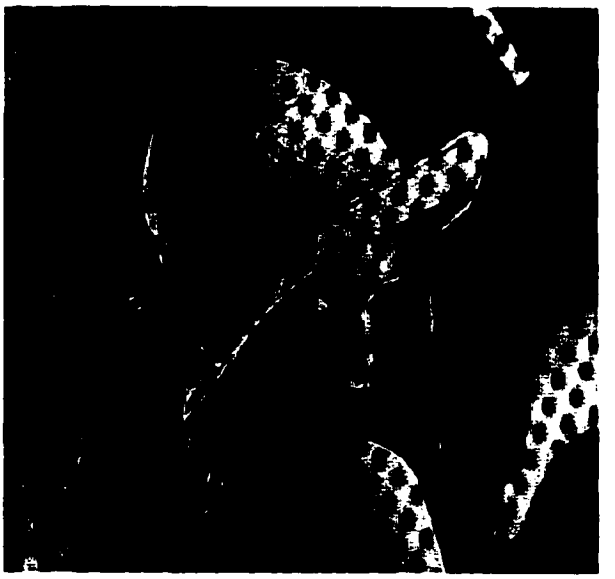

Figure 5.16: cardinal spline with only one path per user, 20.46 dB

The simulation program (C++) for scenario A runs for 26 seconds.

5.3 Simulation Results for RLS MMSE Receiver

For RLS MMSE receiver, when we put the coded bitstreams through channel with impulsive MAI but with fixed fading coefficients (no Doppler effects), just as we did in 1-D case, we could get perfect reconstruction at the beginning. However, after long iterations the filter output always converges to zero, no matter what parameters used or how big energy the input data has. We show this in Fig. 5.17, from which we could find that the first half of image is reconstructed perfectly, but the second half is severely damaged. This phenomenon was not shown in the 1-D case where the bitstream was much shorter than image bitstreams. The reason for this phenomenon is because the basic idea of the *blind multiuser detection*, as shown in (4.22), is to minimize the mean square error between the original data bit and the filter output, and this can be further reduced to (4.23) which is to minimize the filter output. As for the case of RLS MMSE receiver, [36] and [37] use QR-RLS algorithm to update the weight vectors $\mathbf{W}^R(n) \equiv [w_1^R \cdots w_L^R]$ for the adaptive filter coefficients to minimize the filter outputs as in (4.27) which is the counterpart of (4.23) in CDMA applications. This makes the filter outputs of the RLS MMSE receiver gradually approach zero after long iterations and causes the bad effects for the second half of the restored image in Fig. 5.17.

Due to its excellent ability in eliminating the impulsive MAI for a certain length of bits from the beginning of the input data, we make use of the RLS MMSE receiver by segmentation. We can divide the input data to several segments of fixed length, and after the RLS MMSE receiver runs the iterations of a certain number which equals to the length of each segment, we just reset the receiver. As shown in Fig. 5.18, every segment has a fixed length prefix to enable the receiver to get adaptation again, followed by a fixed length segment of data bits. For example, we used three segments to transfer the 45000 bits ($75 \times 75 \times 8$) of LLL subband of *db8*, i.e., 15000 bits per segment, with each segment a prefix of 2500 bits. The receiver recovered the original data for LLL subband perfectly, and the reconstructed image using this recovered LLL subband as well as the original other higher subbands is

Figure 5.17: Half restored image using RLS MMSE receiver

Figure 5.18: Segmentation scheme for RLS MMSE receiver

shown in Fig. 5.19.

By making use of *segment+reset* for the RLS MMSE receiver, we can restore an image perfectly after transmission in the fixed CDMA systems with impulsive MAI. The performance will be similar to that of the performance in noiseless environments, except for a small number of increase in the transmitted data bits. Fig. 5.20 shows an image received by the RLS MMSE receiver which is originally coded at 0.85 bpp using *db8*.

The simulation program (C++) using the RLS MMSE receiver for scenario B runs for 28 seconds.

Figure 5.19: *Lena* using *LLL* subband recovered by RLS MMSE receiver

Figure 5.20: Restored *Lena* using *db8* and RLS MMSE receiver, 0.85 bpp, 28.54dB

5.4 Summary of the Simulation Results

In this chapter, we presented the simulation results for three conditions: (1)scenario A with the impulsive MAI and using RCPC and the Rake receiver; (2)scenario B with non-impulsive MAI using the Rake receiver; (3)scenario B with the impulsive MAI using the RLS MMSE receiver.

In condition (1), we showed that using RCPC can improve the performance of the restored image. Under this condition, *db8* has the best performance, while *bior6.8* performs the worst. However, to obtain a usable image, we still need Reed-Solomon coding which is dedicated for the impulsive noise.

In condition (2), because there is no impulsive MAI, *bior6.8* performs the best. However, without the channel protection, this scheme is very vulnerable to the noise.

In condition (3), we found a best receiving solution for fixed wireless CDMA systems. By using the RLS MMSE receiver, we can transmit the data through the impulsive MAI environment and recover the data with high accuracy without using the channel protection. If combined with equalization techniques to trace the channel coefficients, this method might also have good performance in mobile environments.

Chapter 6

Conclusion

In this chapter, based on all the simulation results, we give a summary of the main conclusions we can draw, a list of the original contributions of this thesis, and discuss what needs to be done in future work.

6.1 Conclusions

- With the main goal of finding an optimal solution to transmit images using wavelets in CDMA systems with impulsive MAI, we found that the orthogonal or spline wavelets are preferred. For fixed wireless CDMA systems with impulsive MAI, the RLS MMSE receiver is much better than the Rake receiver from the performance point of view as well as coding efficiency.

- Cardinal spline wavelets have an interesting feature of excellent energy concentration. The third order ($n = 3$) cardinal spline is a little better than *db8* and *bior6.8* from the energy concentration point of view, which already have very high filter lengths among the commonly used orthogonal and biorthogonal wavelets. The cardinal spline filter banks also have the feature of linear phase.

- Having the feature of time-frequency localization, the general B-spline wavelet transform appears to be the representation of choice if the determining factor is the time-

frequency localization of the basis functions, while the cardinal spline wavelets have good bandpass characteristics and are well suited for coding and compression [39].

- A main disadvantage of spline wavelets is that anti-causal symmetric IIR filters have to be implemented as part of the filter implementation. However, this can be overcome by using its FIR approximation [40] if a certain degree of error is allowed. The error degree can be controlled by increasing the length of FIR approximations.

- The performance of three wavelets — cardinal cubic spline, *bior6.8* and *db8* — in compression and source coding of images is approximately the same. If we only scan and quantize LLH, LHL and LHH subbands, the cardinal spline performed a little better than the other two; if LH, HL and HH subbands are also considered for LVQ, then *bior6.8* performed a little better than the other two. However, there is no big difference in their overall performance in compression.

- The performance diagram for compression shown in Chapter 3 is based on transmitting all of the data in the LLL subband, i.e., no compression was done to the LLL subband. A much higher compression ratio can be achieved if some other existing compression techniques are applied to the LLL subband, like DCT and Huffman coding, but this will lead to more variable-length codes.

- If a higher compression ratio is needed and we do not want to introduce many variable-length codes in the compression of the LLL subband, then more levels of wavelet transforms can be applied, at the cost of increasing complexity. Another way to achieve this purpose is to apply LVQ to all the pixels in LLL subband, but assigning more bits to represent a pixel, like 4-6 bits rather than 2-3 bits used in other subbands.

- Lattice vector quantization (LVQ) is very efficient in the quantization of groups of data. It is easy to pre-process the data before LVQ by just scaling and shifting the data values to "fit" them into the range of LVQ. LVQ can lead to fixed-length output. The only drawback of LVQ is that it does not make use of the statistical properties of the original data. However, this is acceptable in some circumstances like the conditions

in this thesis where more fixed-length codes are needed. Besides, LVQ also allows us to put more complexity in the implementation of wavelet transforms if overall complexity of an image coder is considered.

- The Rake receiver has the properties of less complexity and fast implementation compared to other adaptive algorithms, but could not handle impulsive MAI which would severely damage the received data if no measures are taken. This will more likely happen in some metropolitan areas of big cities, or in some office-concentrated buildings where many computers use wireless LAN cards to inter-connect. Thus, in designing these wireless devices, the designers need to consider measures to handle impulsive MAI, either by using some adaptive algorithm in receiving, or using channel coding.

- The RLS MMSE receiver tested in this thesis can handle impulsive MAI under the condition of no Doppler effect in fading. However, this has a drawback which we did not find in 1-D case: its filter output tends to zero gradually. This can be overcome by initializing the filter tap weights and re-run the algorithm periodically. If it can be combined with some equalization techniques to estimate the channel coefficients, it will be a promising receiving techniques for fixed as well as mobile CDMA systems.

- Convolutional codes as well as RCPC are very useful in joint source-channel coding, and can improve the bit-error-rates under impulsive MAI. However, this technique was originally designed for AWGN rather than impulsive noise. That is why it can coarsely recover the signals contaminated by impulsive MAI to their original shape, but still leave much Gaussian-like noise in the recovered signal that blurred the signal details. We could see this both in 1-D and 2-D cases. For the commercial CDMA systems now, convolutional coding is used in connection with interleaving techniques which will convert the impulsive interference to a more Gaussian-like noise, plus the final low-pass filtering of audio signals, it could effectively recover the original audio signals. However, due to the nature of image signals, we could not simply add a low-pass filter at the final receiving stage, since this would blur the edge detail of the image. Thus Reed-Solomon coding, which is very effective for the impulsive noise, should be used in connection

with convolutional coding for 2-D image cases [9]. Because the focus of this thesis is in implementing spline related wavelets and comparing their features and performance with other commonly used wavelets, Reed-Solomon coding and interleaving techniques were not used in this thesis.

- *db8* has the best overall performance in impulsive MAI environments, while *bior6.8* performs the worst. This might be because for coding systems under impulsive MAI, i.e., under severe multi-user crosstalk, it is important for basis functions of the transform to be orthogonal. The orthogonal feature of basis functions plus channel coding can effectively handle impulsive MAI, compared to biorthogonal basis functions. This is another important result of this thesis in addition to implementing the spline wavelet transform and investigating their properties.

6.2 Contributions of this Thesis

- The spline wavelet transforms of order three ($n = 3$) based on symmetric spline functions and hence anti-causal filter structures were implemented for cubic spline and cardinal cubic spline, and their properties were also investigated for the applications of image coding compared with other conventional wavelet transforms. These spline wavelet transforms are not available in Matlab toolbox.

- A LVQ based on D_4 lattice was implemented, and a coding scheme based on LVQ for the subband coefficients of the wavelet transforms was also achieved for the image coding. This coding scheme is completely different from the state-of-the-art SPIHT as well as EZW coding schemes, with the emphasis on efficiency and robustness to noise. Although the performance of this coding scheme in noiseless conditions is not that high compared with the SPIHT and EZW, it still has some aspects which could be improved. From its existing performance, we could predict that it would perform similarly to the SPIHT and EZW if those improvements are achieved.

- A CDMA receiving scheme using the Rake receiver and Viterbi decoder was imple-

mented to test the overall performance of the different wavelet coded images under MAI, and we found that the wavelet transform with orthogonal basis functions performs better than biorthogonal wavelets under the existance of MAI.

- An adaptive CDMA receiver using RLS MMSE algorithm was successfully implemented, and an attractive feature of this kind of receiver is found that it can effectively handle MAI even without channel coding.

6.3 Future Work

- Considering the good energy-concentration properties of cardinal spline wavelets, and the need for orthogonal basis functions, it would be interesting to see the performance of orthogonal spline wavelets specified in [39], although its implementation is more complex.

- Reed-Solomon coding and interleaving techniques should be added to the coding scheme in which RCPC and the Rake receiver are used.

- Use LVQ with high dimensions, like Λ_{16}, in the quantization of subband coefficients. Because with the increase of the lattice dimensions, we can approach the optimal performance of the *rate distortion*.

Bibliography

[1] M.Antonini, M.Barlaud, P.Mathieu, and I.Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.*, vol. 1, pp. 205–220, 1992.

[2] P.G.Sherwood and K.Zeger, "Progressive image coding on noisy channels," in *Proc. DCC'97*, pp. 72–81, J.A.Storer and M.Cohn., Eds., 1997.

[3] P.G.Sherwood and K.Zeger, "Progressive image coding for noisy channels," in *IEEE Signal Processing Lett.*, vol. 4, pp. 189–191, July 1997.

[4] V.Chande, H.Jafarkhani, and N.Farvardin, "Joint source-channel coding of images for channels with feedback," in *Proc. Information Theory Workshop*, pp. 50–51, Feb. 1998.

[5] H.Man, F.Kossentini, and M.J.Smith, "A class of EZW image coders for noisy channels," in *Proc. ICIP 97*, vol. 3, pp. 90–93, 1997.

[6] H.Man, F.Kossentini, and M.J.Smith, "A family of efficient and channel error resilient wavelet/subband image coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 95–108, 1999.

[7] P.G.Sherwood and K.Zeger, "Error protection for progressive image transmission over memoryless and fading channels," *IEEE Trans. Commun.*, vol. 46, pp. 1555–1559, 1998.

[8] P.G.Sherwood and K.Zeger, "Macroscopic multistage image compression for robust transmission over noisy channels," in *Proc. SPIE*, vol. 3653, pp. 73–83, 1999.

[9] S. Aïssa and E. Dubois, "Robust VB 2D-CELP image transmission over CDMA Rayleigh fading channels," *Signal Process., Image Commun.*, vol. 16, pp. 931–948, 2001.

[10] E.V.H.Iun and A.K.Khandani, "Combined source-channel coding for the transmission of still images over a code division multiple access (CDMA) channel," in *Proc. IEEE Int. Conf. Commun.*, June 1997.

[11] P.Cosman, J.Rogers, P.Sherwood, and K.Zeger, "Combined forward error control and packetized zerotree wavelet encoding for transmission of image over varying channels," *IEEE Trans. Image Process.*, vol. 9, pp. 982–993, 2000.

[12] H. Man, F.Kossentini, and M.Smith, "Robust EZW image coding for noisy channels." *IEEE Signal Processing Letters*, vol. 4, pp. 227–229, 1997.

[13] J.M.Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, pp. 3445–3462, 1993.

[14] A.Said and W.A.Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, 1996.

[15] S.Mallat. "A theory of multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 674–693, 1989.

[16] I.Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 41, pp. 906–996, 1988.

[17] M.Vetterli and J.Kovacevic, *Wavelets and Subband Coding.* New Jersey: Prentice-Hall, 1995.

[18] S.B.Wicker, *Error Control Systems for Digital Communication and Storage.* New Jersey: Prentice-Hall, 1995.

[19] J.Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, pp. 389–400, 1988.

[20] D.Sarwate and M.Pursley, "Crosscorrelation properties of pseudorandum and related sequences," in *Proc. IEEE*, vol. 68, pp. 593–619, 1980.

[21] R.D.Gaudenzi, F.Giannetti, and M.Luise, "Design of a low-complexity adaptive interference-mitigating detector for DS/SS receivers in CDMA radio networks," *IEEE Trans. Commun.*, vol. 46, pp. 125–134, 1998.

[22] J.Ilow and A.N.Venetsanopoulos, "Performance of FH/SS radio networks with interference modeled as a mixture of Gaussian and Alpha-stable noise," *IEEE Trans. Commun.*, vol. 46, pp. 509–520, 1998.

[23] I.Daubechies, *Ten Lectures on Wavelets.* Philadelphia,PA: SIAM, 1992.

[24] A.Cohen, I.Daubechies, and J.C.Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. on Pure and Appl. Math.*, vol. 45, pp. 485–560, 1992.

[25] M.Unser, A.Aldroubi, and M.Eden, "A family of polynomial spline wavelet transforms," *Signal Process.*, vol. 30, pp. 141–162, 1993.

[26] M.Unser and A.Aldroubi, "Spline multiresolutions and wavelet transforms," in *Proc. IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 315–318, 1992.

[27] M.Unser, A.Aldroubi, and M.Eden, "Polynomial spline signal approximations: filter design and asymptotic equivalence with shannon's sampling theorem," *IEEE Trans. Inf. Theory*, vol. 38, pp. 95–103, 1992.

[28] M.Unser, A.Aldroubi, and M.Eden, "Cardinal spline filters: Stability and convergence to the ideal sinc interpolator," *Signal Process.*, vol. 28, pp. 127–138, 1992.

[29] M.Unser, A.Aldroubi, and M.Eden, "B-spline signal processing: Part II-efficient design and applications," *IEEE Trans. Signal Process.*, vol. 41, pp. 834–848, 1993.

[30] M.J.Vrhel and M.Unser, "Fast continuous wavelet transform: A least-squares formulation," *Signal Process.*, vol. 57, pp. 103–119, 1997.

[31] J.D.Gibson and K.Sayood, "Lattice quantization," in *Advances in Electronics and Electron Physics*, vol. 72, pp. 259–330, 1988.

[32] J.H.Conway and N.J.A.Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inf. Theory*, vol. 28, pp. 227–232, 1982.

[33] M.Barlaud, ed., *Wavelets in Image Communication.* ELSEVIER, 1994.

[34] J.H.Conway and N.J.A.Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inf. Theory*, vol. 29, pp. 820–824, 1983.

[35] M. K. Simon, ed., *Spread Spectrum Communications Handbook.* New York: McGraw-Hill, 1994.

[36] X.Wang and H. Poor, "Adaptive joint multiuser detection and channel estimation in multipath fading cdma channels," *Wireless Networks*, vol. 4, pp. 453–470, 1998.

[37] X.Wang and H. Poor, "Space-Time multiuser detection in multipath CDMA channels," *IEEE Trans. Signal Process.*, vol. 47, pp. 2356–2374, 1999.

[38] M.Honig, U.Madhow, and S.Verdú, "Blind adaptive multiuser detection," *IEEE Trans. Inf. Theory*, vol. 41, pp. 944–960, 1995.

[39] M.Unser and A.Aldroubi, "Spline multiresolutions and wavelet transforms," *Proc. IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis.* pp. 315–318, 1992.

[40] M.Unser and M.Eden, "FIR approximations of inverse filters and perfect reconstruction filter banks," *Signal Process.*, vol. 36, pp. 163–174, 1994.