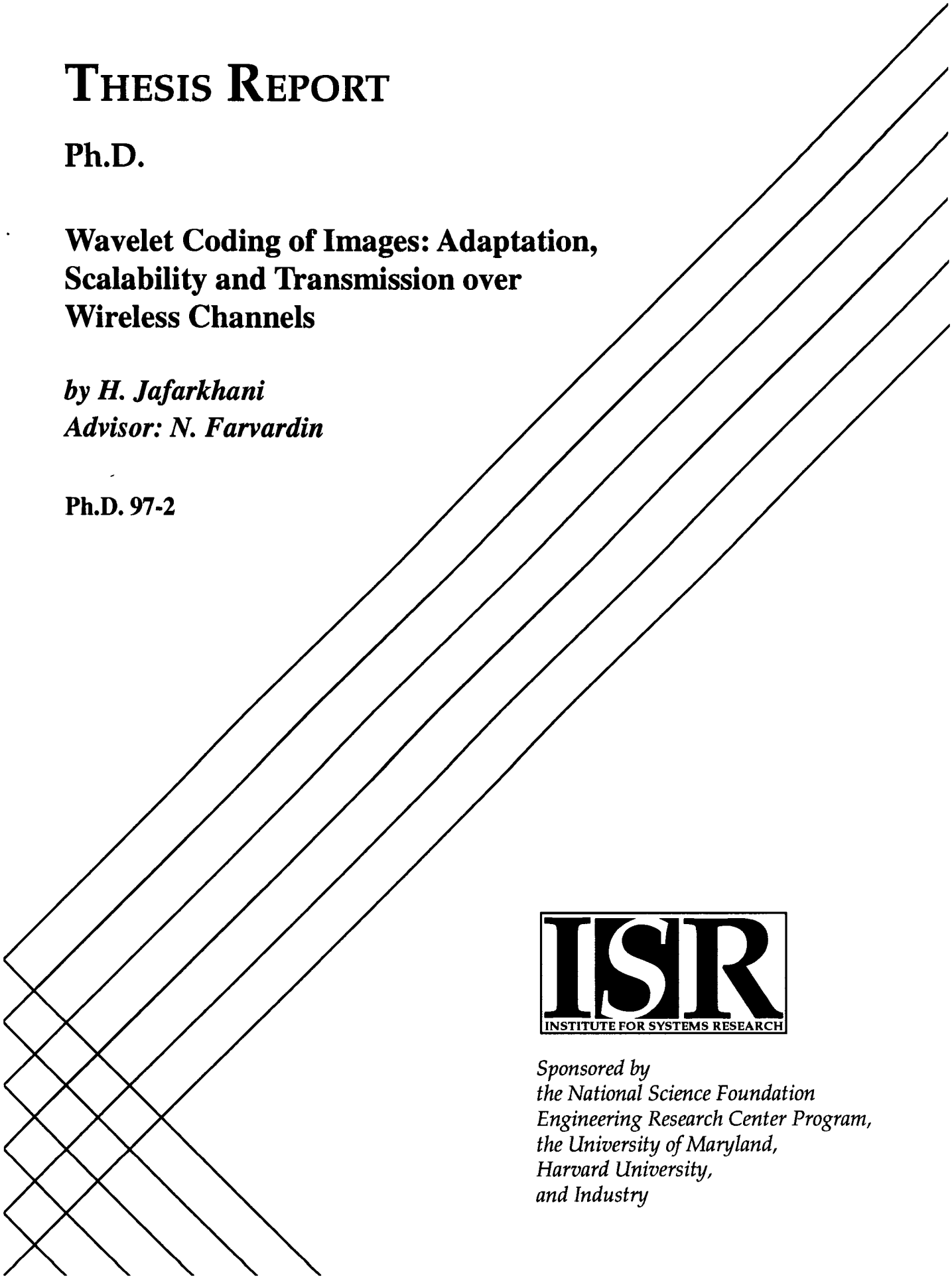
THESIS REPORT

Ph.D.

Wavelet Coding of Images: Adaptation, Scalability and Transmission over Wireless Channels

by H. Jafarkhani
Advisor: N. Farvardin

Ph.D. 97-2



Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Abstract

Title of Dissertation: Wavelet Coding of Images:
Adaptation, Scalability,
and Transmission over Wireless Channels

Hamid Jafarkhani, Doctor of Philosophy, 1997

Dissertation directed by: Professor Nariman Farvardin
Department of Electrical Engineering

In this dissertation, we study the problem of image compression for storage and transmission applications separately. In addition to proposing new image coding systems, we consider different design constraints such as complexity and scalability.

We propose a new classification scheme, dubbed spectral classification, which uses the spectral characteristics of the image blocks to classify them into one of a finite number of classes. The spectral classifier is used in adaptive image coding based on the discrete wavelet transform and shown to outperform gain-based classifiers while requiring a lower computational complexity. The resulting image coding system provides one of the best available rate-distortion performances in the literature. Also, we introduce a family of multi-resolution image coding systems with different constraints on the complexity. For the class of rate-scalable image coding systems, we address the problem of progressive transmission and propose a method for fast reconstruction of a subband-decomposed progressively-transmitted image.

Another important problem studied in this dissertation is the transmission of images over noisy channels, especially for the wireless channels in which the characteristics of the channel is time-varying. We propose an adaptive rate allocation scheme to optimally choose the rates of the source coder and channel coder pair in a tandem source-channel coding framework. Also, we suggest two adaptive coding systems for quantization and transmission over a finite-state channel using a combined source and channel coding scheme. Finally, we develop simple table-lookup encoders to reduce the complexity of channel-optimized quantizers while providing a slightly inferior performance. We propose the use of lookup tables for transcoding in heterogeneous networks.

Wavelet Coding of Images: Adaptation, Scalability, and Transmission over Wireless Channels

by

Hamid Jafarkhani

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland at College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
1997

Advisory Committee:

Professor Nariman Farvardin, Chairman/Advisor
Professor Carlos A. Berenstein
Professor Rama Chellappa
Professor Thomas E. Fuja
Professor Steve Tretter

© Copyright by

Hamid Jafarkhani

1997

Preface

I am twice condemned to torture:

to live so,

and to live so

amongst you

with you

whom I have loved for so long.

A. Bamdad

Dedication

To $\mathcal{M}$.

Acknowledgements

I would like to thank my advisor, Professor Nariman Farvardin, for his support and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Image coding is concerned with minimizing the number of bits required to represent a digital image for a given quality. Given a fidelity criterion, rate-distortion theory provides theoretical limits for compressing stationary sources [Sha59]; however, there is not such a solid known bound for images which are apparently nonstationary. This absence of a theoretical lower bound makes the problem of image compression more challenging; it is not clear how further we can compress an image. Also, there is no perceptually acceptable distortion measure which can be used as a standard to compare the quality of different images. Peak signal-to-noise ratio (PSNR) [1] has been used for the purpose of comparison because it is easy to calculate and is mathematically tractable. A typical digital monochrome image with 8 bits per pixel is considered as the **original** quality although this original image is obtained by sampling and quantizing the corresponding analog image. For a spatial resolution of 512×512 pixels, the amount of data associated with the original still image (or one frame of video) is 256 Kbytes. The

[1] PSNR=$10 \log_{10}[255^2 / \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x}_i)^2]$

huge amount of memory requirements for image storage, or channel capacity for image transmission, suggests the use of data compression. Image storage is required for medical images, documents, motion pictures, satellite images, weather maps, geographical surveys, and so on [Jai89]. Image transmission is applicable in broadcast television, remote sensing via satellite, radar and sonar, video teleconferencing, computer communications, facsimile transmission, and so on [Jai89]. In this dissertation, we address the problems of image storage and image transmission individually in separate chapters.

Image coding for storage applications is a mature field with existing standards like JPEG [Pen93]. However, there are several open issues and new applications which require further investigation. One of the important issues is the capability of providing different resolutions and rates without sacrificing performance. The multi-resolution property of the system, resolution-scalability, refers to the ability of reconstructing different display resolutions at the decoder while providing a single embedded bit stream at the encoder. The multi-rate property of the system, rate-scalability, is defined by the ability of using a subset of the embedded bit stream to reconstruct a lower quality version of the image or adding new bits to reconstruct a better quality version of the image. Based on these characteristics, we can categorize image coding systems into two general classes of scalable and non-scalable systems. For each category, another important issue is the complexity of the encoder and the decoder. The complexity can be considered as a third dimension, besides rate and distortion, for comparing different image coding schemes.

We introduce a family of multi-resolution image coding systems with different constraints on the complexity. First, we assume that there is no constraint on

rate-scalability and the complexity of the system. We design a system which, to the best of our knowledge, outperforms all existing image coding systems. Then, we develop a modified version of the system which involves less complexity. For the class of rate-scalable systems, since there exist schemes with very good performances, e.g. [Sai96] and [Tau94], our focus will be on developing efficient and low-complexity scalable image coding systems. Also, we address the problem of progressive transmission for the class of rate-scalable image coding systems and propose a method for fast reconstruction at the decoder.

Another important problem studied in this dissertation is the transmission of images over noisy channels. A new challenge arises when a system is to be used for transmission over noisy channels, especially for the wireless channels in which the characteristics of the channel is time-varying. One way to overcome this problem is to use channel coding techniques. Shannon has shown that, theoretically and under some conditions, source coding and channel coding can be done separately without loss of optimality [Sha48]. It is a well-known fact that in the case of transmission over noisy channels under complexity or delay constraints, a joint design of source coders and channel coders provides a better performance than the performance of a separate design [Tan92, Far90]. For the special case of time-varying channels (e.g. wireless channels), the separate design of the source coder and the channel coder is not even asymptotically optimal [Vem95].

Transmission of images over wireless channels is an interesting application with many open problems. In this dissertation, we address some of these problems and propose some preliminary solutions. Especially, we concentrate on stationary (or, in the worst case, very slowly moving) users, and as a result, slow

fading or fading caused by the movement of other objects in the surrounding. There are some useful applications corresponding to this assumption: (i) two-way, real-time teleconferencing with a portable computer on at least one end, (ii) one-way or two-way, real-time video communication in search and rescue type operations or in tele-medicine applications, and (iii) one-way, non-real-time wireless access to image data.

In a joint source-channel coding scheme, one of the important problems is how to allocate the bit rate (bandwidth) between the source coder and the channel coder. The time-varying characteristics of the wireless channels suggest that the source coder and the channel coder must be able to adapt themselves to the state of the channel. One way to provide such an adaptation is to change the rate of the coders. We propose an adaptive rate allocation scheme to optimally choose the rates of the source coder and channel coder pair.

We also propose a channel-matched hierarchical table-lookup vector quantizer which simultaneously provides low encoding complexity and robustness against transmission noise. An important application is the use of lookup tables for transcoding in heterogeneous networks. We use a finite-state channel to model slow fading channels and propose two adaptive coding schemes to encode a source for transmission over wireless channels: (i) a fully-adaptive system in which the channel state information (CSI) is available at the encoder and decoder and (ii) a decoder-adaptive system in which the CSI is only available at the decoder.

Chapter 2

Basic Building Blocks

2.1 Discrete Wavelet Transform and Subband Decomposition

Image coding based on subband decomposition or discrete wavelet transform (DWT) ideas has received much attention in recent years [Woo86, Mal89, Ant92]. In addition to giving good compression results (in a rate-distortion sense), these systems are suitable for progressive transmission and provide a multi-resolution capability – a feature that is desirable in some practical situations. The basic idea behind the DWT is to decompose the input signal into two components: (i) a **low-resolution approximation** and (ii) a **detail signal**. This results in decomposing the input signal into low-pass and high-pass versions, generally referred to as subbands. Each of the resulting subbands can be further decomposed using the same approach. In this manner, the DWT decomposes a given input signal into a number of frequency bands. Two-dimensional (2-D) extensions of the DWT can be obtained by a separable decomposition in the horizontal and

vertical directions [Woo86, Mal89].

The idea of subband coding is to decompose the source into its different frequency subbands and encode each subband using a suitable coding system [Cro76a]. Because there is little correlation between the decomposed subbands, they can be coded separately [Mal89, Ant92, Tan92]. The separate coding of different subbands provides three desirable features. First, by allocating the available bits for encoding among the subbands and using an appropriate quantizer for each of them, the encoding process can be tailored to the statistics of each subband. Second, spectral shaping of the quantization noise is possible. This feature can be used to take advantage of the noise perception of the human auditory system for speech or the human visual system for images. Third, the subband decomposition of the signal spectrum leads naturally to multiresolution signal decomposition.

Using a bank of n band-pass filters, each associated with a separate frequency channel, n different subbands of the source will be created. A dual set of filters is needed to reconstruct the original source from its subbands. Quadrature Mirror Filters (QMF's) can be used to have a perfect reconstruction [Cro76b]. A multiresolution representation using wavelet orthonormal [Mal89] or biorthogonal [Ant92] bases is another approach which is the subject of this section.

In a subband coding system, after decomposition, the subbands are quantized and encoded. Therefore, it is not clear that the subband filters need to be perfect reconstruction filters. However, since no definitive techniques for combining the filter design and quantization are known, the filters used for decomposition are usually perfect reconstruction filters. Good coding results have been reported based on these filters.

2.1.1 Multiresolution Signal Decomposition

Defining $L^2(\mathbb{R})$ as the vector space of measurable and square-integrable one-dimensional (1-D) functions, wavelets are functions $\psi(x)$ whose translations and dilations ($\psi_{m,n}(x) = 2^{-m/2}\psi(2^{-m}x - n)$) can be used for the expansion of functions in $L^2(\mathbb{R})$ [Mey85]. The class of functions $\psi(x) \in L^2(\mathbb{R})$ that generate an orthonormal basis for $L^2(\mathbb{R})$ can be described using the multiresolution approach to wavelets [Mal89]. In what follows, the model is first described for 1-D functions and then extended to two dimensions for image processing applications.

Let A_m be a linear operator used to approximate a signal at resolution m. Clearly, if $A_m f(x)$ is the approximation of $f(x) \in L^2(\mathbb{R})$ at resolution m, then $A_m(A_m f(x)) = A_m f(x)$. Thus, A_m is a projection operator on a particular vector space $V_m \subset L^2(\mathbb{R})$. The vector space V_m can be interpreted as the set of all possible approximations at resolution m of functions in $L^2(\mathbb{R})$. Any set of vector spaces $\{V_m\}$, for all m in $\mathbb{Z}$, which satisfies the following properties is called a multiresolution approximation of $L^2(\mathbb{R})$ [Mal89]:

$$V_m \subset V_{m-1}, \quad \forall m \in \mathbb{Z}, \tag{2.1}$$

$$f(x) \in V_m \Leftrightarrow f(2x) \in V_{m-1}, \quad \forall m \in \mathbb{Z}, \tag{2.2}$$

$$\exists \phi \in V_0 \text{ such that } \phi(x-n) \in V_0, \quad \forall n, \tag{2.3}$$

$$\bigcap_m V_m = \{0\}, \tag{2.4}$$

$$\bigcup_m V_m = L^2(\mathbb{R}). \tag{2.5}$$

Mallat has shown that for a multiresolution approximation of $L^2(\mathbb{R})$, there exists a unique function $\phi(x) \in L^2(\mathbb{R})$, called a **scaling function**, such that $\phi_{m,n}(x) = 2^{-m/2}\phi(2^{-m}x - n)$, $n \in \mathbb{Z}$ forms an orthonormal basis for V_m [Mal89].

The orthogonal projection of $f(x)$ on V_m can be computed by decomposing the signal $f(x)$ with the above orthonormal basis:

$$A_m f(x) = \sum_{n=-\infty}^{\infty} < f, \phi_{m,n} > \phi_{m,n}(x). \tag{2.6}$$

A discrete approximation of $f(x)$ at resolution m can be defined as

$$a_{m,n}(f) = A_m^n f =< f, \phi_{m,n} >, \quad \forall n \in \mathbb{Z}. \tag{2.7}$$

Equation (2.7) can be interpreted as a convolution product evaluated at point $2^m n$:

$$a_{m,n}(f) = A_m^n f = (f(u) * \phi_{m,0}(-u))(2^m n), \quad \forall n \in \mathbb{Z}. \tag{2.8}$$

The above equation is also equivalent to low-pass filtering of $f(x)$ followed by a uniform sampling at the rate 2^m. To follow rapid changes of the signal in space domain as well as the frequency domain using finite number of coefficients, the transform used must accept nonstationarity and be well localized in both space and frequency domains [Ant92].

Let $\tilde{H}$ be a discrete filter with impulse response

$$\tilde{h}(n) =< \phi_{1,0}, \phi_{0,n} >, \tag{2.9}$$

and let H be the mirror filter with impulse response $h(n) = \tilde{h}(-n)$. It can be shown that [Mal89]

$$a_{m,n}(f) = \sum_{k=-\infty}^{\infty} h(2n-k) a_{m-1,k}(f). \tag{2.10}$$

Equation (2.10) implies that $a_{m,n}(f)$ can be computed by convolving $a_{m-1,k}(f)$ with H and sub-sampling by 2. All the discrete approximations $a_{m,n}(f)$, for $m > 0$ can be computed from $a_{1,n}(f)$ by repeating this process. Usually a

regularity condition is imposed on the scaling function requiring that $\phi(x)$ be continuously differentiable and that the asymptotic decay of $\phi(x)$ and $\phi\prime(x)$ satisfy

$$| \phi(x) | = O(x^{-2}), \tag{2.11}$$

and

$$| \phi\prime(x) | = O(x^{-2}). \tag{2.12}$$

Having the above conditions on $\phi(x)$, the filter H will satisfy the following properties:

$$h(n) = O(n^{-2}), \tag{2.13}$$

$$| H(0) | = 1, \tag{2.14}$$

and

$$| H(\omega) |^2 + | H(\omega + \pi) |^2 = 1. \tag{2.15}$$

The orthogonal complement subspace of V_m in V_{m-1} is denoted by W_m, i.e.

$$W_m \oplus V_m = V_{m-1} \text{ and } W_m \perp V_m. \tag{2.16}$$

The orthogonal projection of a signal on W_m is called the detail signal at resolution m.

Denoting the Fourier transform of $f(x)$ by $\hat{f}(\omega)$ and the conjugate of a complex number x by $\overline{x}$, $\psi(x)$ is a function defined by

$$\hat{\psi}(\omega) = \tilde{G}(\omega/2)\hat{\phi}(\omega/2), \tag{2.17}$$

where

$$\tilde{G}(\omega) = e^{-j\omega}\overline{\tilde{H}(\omega + \pi)}. \tag{2.18}$$

Then $\psi_{m,n}(x)$, for all n in $\mathbb{Z}$, is an orthonormal basis of W_m and $\psi_{m,n}(x)$, for all (m,n) in $\mathbb{Z}^2$, is an orthonormal basis of $L^2(\mathbb{R})$. Here, $\psi(x)$ is called an orthogonal mother wavelet. The orthogonal projection of $f(x)$ on W_m (detail signal) can be characterized by the following set of inner products:

$$c_{m,n}(f) = D_m^n f =< f, \psi_{m,n} >, \quad \forall n \in \mathbb{Z}. \tag{2.19}$$

Upon defining a filter $\tilde{G}$ with impulse response $\tilde{g}(n)$ given by

$$\tilde{g}(n) =< \psi_{1,0}, \phi_{0,n} >, \quad \forall n \in \mathbb{Z}, \tag{2.20}$$

it can be shown that

$$c_{m,n}(f) = \sum_{k=-\infty}^{\infty} g(2n-k) a_{m-1,k}(f), \tag{2.21}$$

where $g(n) = \tilde{g}(-n)$. The detail signal $c_{m,n}(f)$ can be computed by convolving $a_{m-1,n}(f)$ with the filter g and sub-sampling by 2. Using (2.18), $\tilde{g}(n)$ and $\tilde{h}(n)$ are related by the following equation:

$$\tilde{g}(n) = (-1)^{1-n} \tilde{h}(1-n). \tag{2.22}$$

The two filters $\tilde{G}$ and $\tilde{H}$ are referred to as QMF's and are high-pass and low-pass filters respectively.

In practice, a physical measuring device can only measure a signal at a finite resolution. For normalization purposes, we suppose that this resolution is equal to 1. The original discrete $a_{1,n}(f)$ measured at resolution 1 is represented by $a_{M,n}(f)$, $\{c_{m,n}(f)\}$, $1 < m \leq M$. This set of discrete signals is called an orthogonal wavelet representation. If the discrete approximation of the original signal at resolution 1 has N samples, then the discrete signals $c_{m,n}$ and $a_{m,n}$ each will have $2^{-m}N$ samples. Therefore, the wavelet representation has the same total number of samples as the original one. This is because of orthogonality.

Reconstruction can be achieved by upsampling (placing zeros between neighboring samples) $a_{m,n}(f)$ and $c_{m,n}(f)$ and passing them through $\tilde{H}$ and $\tilde{G}$ respectively:

$$a_{m-1,l}(f) = \sum_{n=-\infty}^{\infty} [\tilde{h}(l-2n)a_{m,n}(f) + \tilde{g}(l-2n)c_{m,n}(f)]. \tag{2.23}$$

2.1.2 Biorthogonal Wavelet Bases

Most of the orthonormal wavelet bases have infinitely supported ψ, corresponding to filters h and g with infinitely many taps. Having finite number of taps (using FIR filters) is required in practice and can be achieved when the support of ψ is finite. It is desirable that the FIR filters used be linear-phase, since such filters can be easily cascaded in a pyramidal filter structure without the need for phase compensation [Ant92].

Unfortunately, there are no nontrivial orthonormal linear-phase FIR filters with the exact reconstruction property. The only symmetric exact reconstruction filters are those corresponding to the Haar basis, i.e. $h_0 = h_1 = 2^{1/2}$ and $g_0 = -g_1 = 2^{1/2}$, with all other $g_n, h_n = 0$ [Mal89].

By preserving the linear-phase property of the FIR filters and relaxing the orthonormality requirement, and using biorthogonal bases, it is possible to have perfect reconstruction and arbitrarily high regularity [Ant92]. In such a scheme (Figure 2.2), the decomposition is the same as before

$$c_{m,n}(f) = \sum_{k=-\infty}^{\infty} g(2n-k)a_{m-1,k}(f), \tag{2.24}$$

$$a_{m,n}(f) = \sum_{k=-\infty}^{\infty} h(2n-k)a_{m-1,k}(f). \tag{2.25}$$

However, reconstruction becomes

$$a_{m-1,l}(f) = \sum_{n=-\infty}^{\infty} [\tilde{h}(2n-l)a_{m,n}(f) + \tilde{g}(2n-l)c_{m,n}(f)], \tag{2.26}$$

where the filters $\tilde{h}$ and $\tilde{g}$ may be different from h and g. Perfect reconstruction is possible when

$$\tilde{g}(n) = (-1)^n h(1-n), \tag{2.27}$$

$$g(n) = (-1)^n \tilde{h}(1-n), \tag{2.28}$$

and

$$\sum_{n=-\infty}^{\infty} h(n)\tilde{h}(n+2k) = \delta_{k,0}. \tag{2.29}$$

The interpretation of the biorthogonal scheme in terms of the bases can be done as follows. Define the functions ϕ and $\tilde{\phi}$ by

$$\phi(x) = \sum_{n=-\infty}^{\infty} h(n)\phi(2x-n), \tag{2.30}$$

$$\tilde{\phi}(x) = \sum_{n=-\infty}^{\infty} \tilde{h}(n)\tilde{\phi}(2x-n). \tag{2.31}$$

Also define

$$\psi(x) = \sum_{n=-\infty}^{\infty} g(n)\phi(2x-n), \tag{2.32}$$

$$\tilde{\psi}(x) = \sum_{n=-\infty}^{\infty} \tilde{g}(n)\tilde{\phi}(2x-n). \tag{2.33}$$

Then $a_{m,n}(f)$ and $c_{m,n}(f)$ can be rewritten as

$$a_{m,n}(f) = < \phi_{m,n}, f >, \tag{2.34}$$

$$c_{m,n}(f) = < \psi_{m,n}, f > . \tag{2.35}$$

Reconstruction is done by

$$f = \sum_{m,n} < \psi_{m,n}, f > \tilde{\psi}_{m,n}. \tag{2.36}$$

Equation (2.36) is very similar to the orthonormal case. The only difference is that the expansion of f with respect to the basis $\tilde{\psi}_{m,n}$ uses the coefficients computed via the dual basis $\psi_{m,n}$.

2.1.3 Extension to Two-Dimensional Signals

A simple way for extending the 1-D transform of the previous section to a 2-D case is by separating the horizontal and vertical orientations. A scaling function can be defined by

$$\phi(x,y) = \phi(x)\phi(y), \tag{2.37}$$

where $\phi(x)$ is a 1-D scaling function. Let $\psi(x)$ be the wavelet associated with the scaling function $\phi(x)$. Then, the three 2-D wavelets are defined as

$$\psi^H(x,y) = \phi(x)\psi(y), \tag{2.38}$$

$$\psi^V(x,y) = \psi(x)\phi(y), \tag{2.39}$$

$$\psi^D(x,y) = \psi(x)\psi(y). \tag{2.40}$$

The implementation is the same as before. Practically, rows will be decomposed by using the 1-D decomposition filters and then the columns of the output will be decomposed using the same system.

When an $M \times N$ image is processed, after the first stage of decomposition, two $M \times \frac{N}{2}$ images will be created. Each of them will go through the same process for the columns and resulting in four $\frac{M}{2} \times \frac{N}{2}$ images. Clearly, the total number of samples is the same as the original one. Thus, to prevent using extra memory, these four sub-images can be placed in the memory location of the original image as in Figure 2.3. The original image can be replicated by first

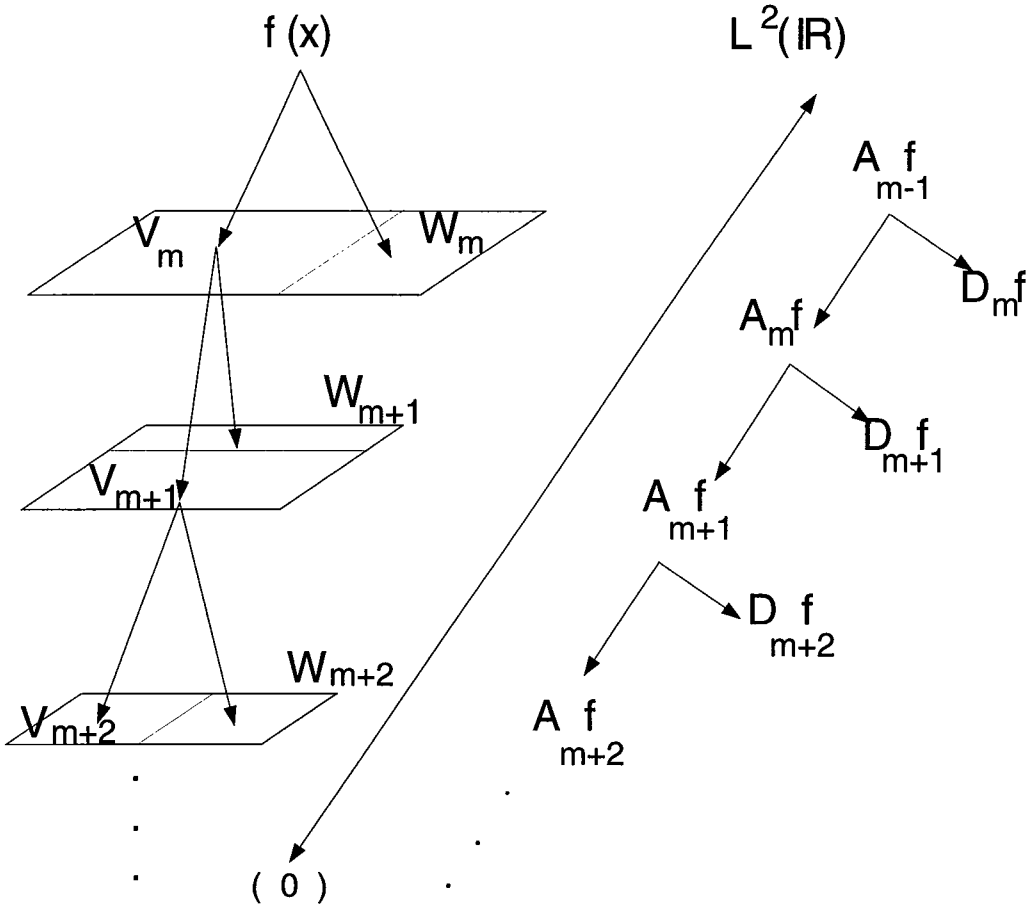


Figure 2.1: Multiresolution Approximation of $L^2(\mathbb{R})$

performing the reconstruction operations on columns and then doing the same process on rows.

Each sub-image can be decomposed into new subbands using the same set of filters. The result for two levels of decomposition for the 512×512 monochrome Lenna image using the 9-7 spline filters designated No. 2 in [Ant92] is shown in Figure 2.5.

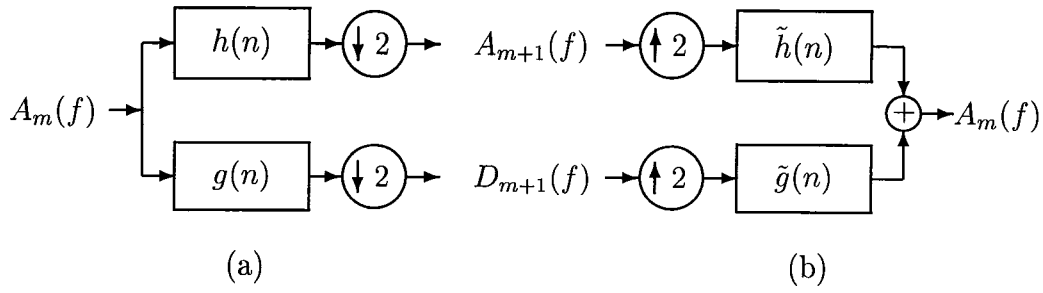


Figure 2.2: Implementation of the Biorthogonal Wavelet Scheme: (a) Wavelet Decomposition and (b) Wavelet Reconstruction

H for Rows H for Columns	H for Rows G for Columns
G for Rows H for Columns	G for Rows G for Columns

Figure 2.3: Decomposing an Image into Four Subbands.

2.2 Gain-Based Classification

In general, **classification** is a mapping from a high-dimensional event space onto a low-dimensional feature space. Some form of classification is often used in compression of real-world signals. Classifying blocks of signal samples into a number of classes with similar properties (statistical, spectral, perceptual, etc.), makes it possible to design quantizers that are better suited for coding the overall signal. In a sense, signal compression with classification is some type of adaptive

Figure 2.4: 512×512 Lenna

Figure 2.5: Decomposed 512×512 Lenna

signal compression.

A classified quantizer uses separate codebooks for different classes [Ram86]. Due to the apparent nonstationary behavior of the image blocks, classifying these blocks into more homogeneous groups provides the opportunity for using stationary probabilistic models for each group. Additionally, using a nonuniform bit allocation among the different groups, more bits can be assigned to the more important parts of the image. In this manner, the coding system captures the nonstationary behavior of the source and results in better performance.

A block classification algorithm is characterized by (i) the feature which is used for classification and (ii) the mapping which assigns a class to each block. Of course, different choices of features or mappings result in different classification schemes. A suitable parameter for block classification is the gain (square root of the block ac-energy) of each image block. In gain-based classification schemes, blocks are classified into a prescribed number of classes according to their gain values. Thus, by and large, blocks with close gain values are assigned to the same class.

To compute the gains, the image is divided into $L \times L$ nonoverlapping blocks of pixels. If the $L \times L$ matrix which contains the pixels in the k^{th} block is denoted by A_k, its gain, g_k, is given by

$$g_k = \sqrt{\frac{1}{L^2} \sum_{i=1}^{L} \sum_{j=1}^{L} [A_k(i,j) - m_k]^2}, \tag{2.41}$$

where

$$m_k = \frac{1}{L^2} \sum_{i=1}^{L} \sum_{j=1}^{L} A_k(i,j). \tag{2.42}$$

This gain is a good measure of the level of activity in the block. For example, if the block only consists of one grey level, the corresponding gain is zero. On

the other hand, when grey levels change rapidly in a block, its gain is relatively high. The extreme case is when half of the pixels are white (grey level 255) and the other half are black (grey level 0) and the corresponding gain is 127.5.

In gain-based classification schemes, blocks are classified into a prescribed number of classes according to their gain values. In this manner, blocks with close gain values are gathered in the same class. The classification procedure therefore assigns a class index to each block. A matrix whose entries are these class indices is referred to as the **classification table**. Needless to say, different criteria for classification result in different classification tables. In this section, we consider different criteria for classifying the blocks of an image based on their gain values.

2.2.1 Chen-Smith Approach

Chen and Smith [Che77] suggested a simple method to perform the classification, or, equivalently, to choose thresholds between the gain values of different classes. In this approach, thresholds are chosen such that all classes have the same number of blocks. If there are N blocks to be classified to K classes, the $\frac{N}{K}$ blocks with highest gain values fall into the first class. The $\frac{N}{K}$ blocks with highest gain values among the remaining blocks fall into the second class and so on. These class indices specify the classification table and can be coded using $\log_2 K$ bits/block. The classification table using four classes for the 512×512 Lenna image and blocks of size 16×16 is shown in Figure 2.6 in the form of an image. This image is constructed by assigning a different grey level to each class in order to visualize the classification table. Class 0 (low activity class) is represented by a grey value equal to 0. Classes 1, 2, and 3 are represented by

grey levels 100, 150, and 255, respectively.

The Chen-Smith classification procedure is an easy way to classify blocks into different regions of activity. Although we have not defined a measurable quantity to compare classification schemes with each other, intuitively, there are situations in which this simple approach does not work well. For example, an image containing a large area with constant grey levels has a lot of blocks with zero (or very small) gains. If the number of such blocks exceeds $\frac{N}{K}$, the Chen-Smith procedure will put some of them into other classes although it is expected to have all of them in one class. The capability of allowing a different number of blocks in each class can potentially improve the classification. Also, as a gain-based classification method, Chen-Smith scheme pays no attention to the spectral shape of the blocks. This issue is considered in depth in Chapter 3.

2.2.2 Equal Mean-Normalized Standard Deviation (EMNSD) Approach

In what follows we propose a different method in which after sorting the gain values, they are split into the given number of classes such that the mean-normalized standard deviation of the resulting classes are the same. The idea behind this approach is to allow the possibility of having a different number of blocks in each class and to have similar statistical properties within each class such that the representation of the blocks as one class is meaningful from a coding stand point. For a stationary source, standard deviation is a measure of dispersion of samples and the smaller is the standard deviation of a source, the denser will be the samples about the mean. When one of the classes has a higher dispersion than others, the blocks in that particular class do not have the same level of

activity. It is difficult to compare dispersions in sets with different means. The "coefficient of variation" (defined by standard deviation divided by mean) is a good measure for dispersion [Stu87].

In EMNSD classification scheme, we try to make the mean-normalized standard deviation of the gains in the resulting classes as close to each other as possible. Because of the discrete nature of the problem an exact solution for the problem may not be possible and there is no guarantee that a solution, if one exists, is unique. For simplicity, for the moment, we consider the case with two classes. When there are N blocks sorted in an increasing order of their gain values g_i, $i = 1, 2, \ldots, N$, we look for an integer N' such that blocks 1 to N' belong to the first class and the remaining blocks belong to the second class. The mean m and standard deviation σ of each class is defined by

$$\begin{aligned} m_1 &= \tfrac{1}{N'} \textstyle\sum_{n=1}^{N'} g_n, \\ m_2 &= \tfrac{1}{N-N'} \textstyle\sum_{n=N'+1}^{N} g_n, \\ \sigma_1^2 &= \tfrac{1}{N'} \textstyle\sum_{n=1}^{N'} (g_n - m_1)^2, \\ \sigma_2^2 &= \tfrac{1}{N-N'} \textstyle\sum_{n=N'+1}^{N} (g_n - m_2)^2. \end{aligned} \tag{2.43}$$

Here, N' is chosen such that

$$q_1 = \frac{\sigma_1}{m_1} = \frac{\sigma_2}{m_2} = q_2. \tag{2.44}$$

An iterative algorithm to find N' satisfying (2.44) is provided below. If there is no integer N' which solves (2.44), the algorithm finds the N' which minimizes $\mid q_1 - q_2 \mid$.

- **Algorithm**:

  1. Choose an initial value for N' (e.g. $N' = N/2$) and set the iteration

number $i = 0$. Also choose i_{max} as an upper limit on the number of iterations.

2. Compute q_1 and q_2 using (2.44) and set $i = i + 1$.

3. If $\frac{|q_1 - q_2|}{q_1} < \delta$ or $i > i_{max}$ stop. Otherwise,

$$\text{if } q_1 < q_2, \text{ set } N' = N' + \Delta N'$$

$$\text{if } q_1 > q_2, \text{ set } N' = N' - \Delta N'$$

and go to (2).

For fast convergence, a large $\Delta N'$ can be chosen at the beginning of the algorithm and as the iteration number increases $\Delta N'$ must be gradually decreased to one.

The same algorithm can be generalized for a larger number of classes. For the case of K classes, K ratios $q_i = \frac{\sigma_i}{m_i}$, $i = 1, 2, \ldots, K$, and $(K-1)$ thresholds are needed. The algorithm is stopped when $\frac{\max_i q_i - \min_i q_i}{\min_i q_i} < \delta$ or when the number of iteration exceeds its maximum. At each step of the algorithm, the thresholds corresponding to the class with the maximum and minimum q_i are alternatively adjusted so as to make q_i's as close to one another as possible. The resulting classification image using four classes for the 512×512 Lenna ($\delta = 0.01$) is illustrated in Figure 2.7. In Figure 2.7, we have used the same grey levels as in Figure 2.6. Although it is not conclusive by itself, comparing Figures 2.6 and 2.7 shows that the EMNSD classification is more successful than the Chen-Smith classification in separating high activity regions from other parts of the image. For example, using the Chen-Smith scheme results in putting the blocks corresponding to the strong edges and most of the blocks corresponding to the texture in the feather in the same class. On the other hand, using EMNSD approach has placed these blocks into different classes.

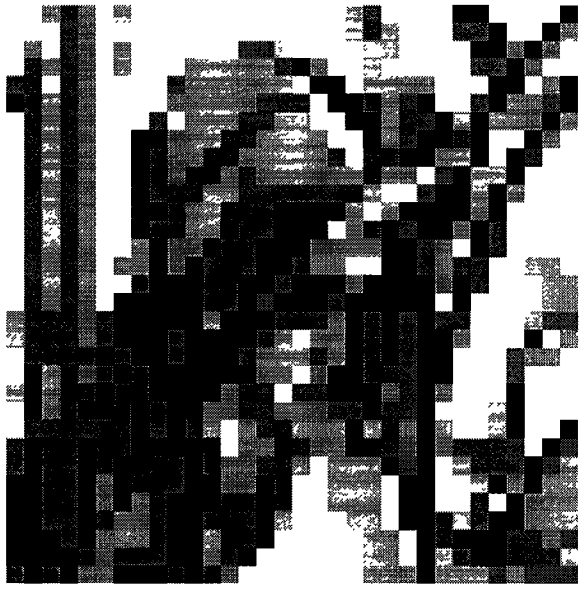


Figure 2.6: Chen-Smith Classification Image; 512 × 512 Lenna.

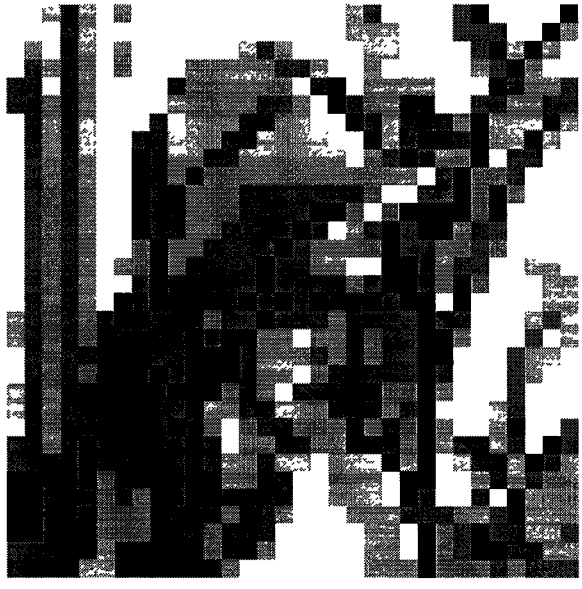


Figure 2.7: EMNSD Classification Image; 512 × 512 Lenna.

2.3 Hierarchical Table–Lookup Vector Quantization

Vector quantization, a powerful technique for source coding and signal classification, has been the subject of intense research since early 1980's when Linde, Buzo, and Gray [Lin80] suggested a practical method for vector quantizer (VQ) design. A comprehensive treatment of vector quantization, including discussion of applications to speech and image coding can be found in [Ger92].

The codebook of an m-dimensional, rate R bits/sample VQ consists of $M = 2^{mR}$ codevectors. In general, a full-searched VQ encoder computes the distortion between the input vector and each codevector in the codebook to find the codevector which results in minimum distortion. This involves M distortion computations per input vector. The decoder however is a simple lookup table. The encoding complexity of the VQ, which grows exponentially with the product of dimension and rate, constitutes a bottleneck in many practical applications in which the VQ encoder is to be implemented in real time. This problem has led to the development of a variety of different methods for reducing the VQ encoding complexity. Examples are tree-structured vector quantization [Buz80], multi-stage vector quantization [Jua82], fine-coarse vector quantization [Moa91], lattice-based vector quantization [Con82, Fis86], and gain-shape vector quantization [Buz80]. Other examples can be found in [Ger92]. Invariably, these methods offer a tradeoff between performance, encoding complexity, and memory requirements.

Another technique for reducing the encoding complexity, developed by Chang, May, and Gray [Cha85], is the hierarchical table-lookup VQ (HTVQ) in which

the input vector is successively quantized in multiple stages. Each quantization stage is implemented using a lookup table which stores, for every possible input vector, the codeword (index) of the nearest codevector. The entire idea is predicated on the assumption that there are a finite number of input vectors, e.g., each input sample is already quantized – a reasonable assumption in most practical situations. When applied to speech coding, the HTVQ system proposed in [Cha85] is shown to suffer no more than 1 dB performance degradation compared with a full-searched VQ. What is particularly attractive about the HTVQ is that it entirely eliminates the need for any additions or multiplications, thus making HTVQ a viable candidate for real-time, software-based source coding applications.

More recently, Vishwanath and Chou have used the same concept for video coding where they also develop a system combining the HTVQ idea with a DWT for improved performance [Vis94]. The potential applications of HTVQ encoders for real-time video compression over Internet-like networks, has resulted in a proliferation of research on various extensions of the HTVQ in the past two years [Cha96b, Cha96a, Meh96, Jos96].

Let us briefly review the design and operation of an HTVQ. Consider an m-dimensional VQ which performs the encoding operation using a lookup table. Assuming n-bit input samples, encoding is performed by a lookup table which stores the index of the nearest VQ codevector for every possible source vector (2^{mn} possibilities). Note that this table-lookup encoding does not lead to any performance loss as compared to a full-searched VQ.

The size of such a table (in bits) is 2^{mn} times the number of output bits. Clearly, for large values of mn this table will be prohibitively large. To keep

Table 2.1: Lookup Table for One Stage of an HTVQ.

Input 1	0	0	$\cdots$	255
Input 2	0	1	$\cdots$	255
Output	i_0	i_1	$\cdots$	i_{65535}

the table sizes within manageable limits for large dimension VQ's, a hierarchical structure, as shown in Figure 2.8, was proposed in [Cha85]. In the most general case, m_{i+1} outputs of Table i constitute the input of Table $i+1$ in Figure 2.8. A practical configuration, which has been used throughout this dissertation, corresponds to $m_i = 2\ \forall i$. Figure 2.9 shows a 3-stage HTVQ with $n = 8$, $m_i = 2,\ i = 1, 2, 3$, and each stage VQ containing 256 codevectors (thus 8-bit outputs). To construct the lookup tables for each stage, first three VQ's must be designed for the source: a 2-dimensional VQ for the first stage, a 4-dimensional VQ for the second stage, and an 8-dimensional VQ for the third stage. These VQ's operate at 4, 2, and 1 bit/sample, respectively. Table 1 is constructed by considering each pair of input samples (2^{16} possibilities), and storing the index (codeword) of the nearest codevector of the first stage VQ as the output of the table (see Table 2.1). Likewise, to construct Table 2, all possible 8-bit pairs at the input of Table 2 are considered and the 4-dimensional vector corresponding to each pair is encoded using the 4-dimensional VQ of the second stage; the resulting 8-bit codeword is stored as the output of Table 2. A similar procedure can be used to design Table 3.

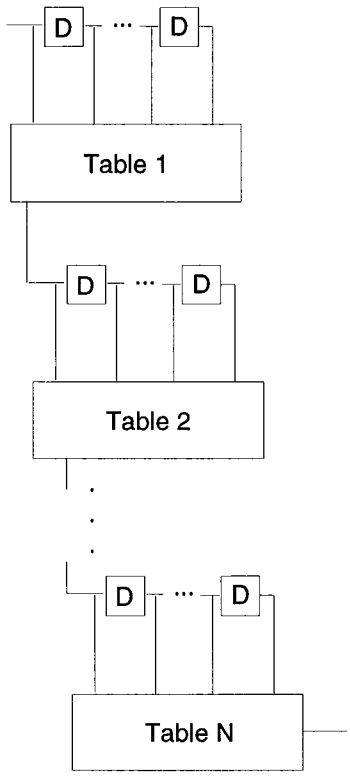


Figure 2.8: An N-Stage HTVQ Encoder.

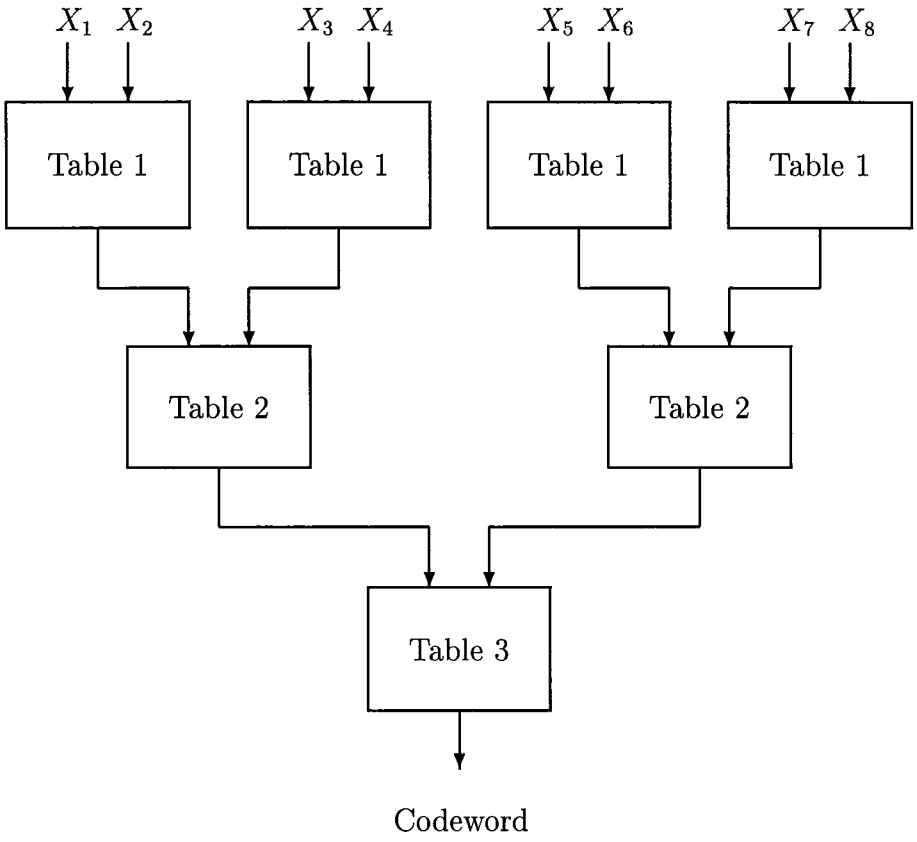


Figure 2.9: A 3-Stage HTVQ Encoder.

Chapter 3

Adaptive Wavelet Coding Using Spectral Classification

3.1 Introduction

Ever since the pioneering work of Chen and Smith in adaptive discrete cosine transform (ADCT) coding [Che77], it has been well-known that the gain of a spatial block is a useful quantity for measuring the level of activity within a block. By first classifying the image blocks into four classes based on their ac-energies, Chen and Smith were able to tailor the parameters of the quantizers to suit the needs of each class and therefore obtain coding gains.

However, this classification suffers from the unnecessary constraint that all classes contain an equal number of blocks. Because most natural images contain fewer high activity blocks than other blocks, this requirement forces some low activity blocks to be assigned to high activity classes, thus wasting some of the available bit rate and reducing the efficacy of the classification scheme. By al-

lowing the number of blocks in each class to vary according to the characteristics of the image at hand, one can avoid this inefficient use of rate. To do this, however, it is necessary to develop a criterion for determining where the thresholds for each class should be placed so that all blocks with similar activity levels are placed in the same class. Recently, some solutions to this problem and applications of gain classification in subband coding have been explored in [Jos95]. Two new gain classification schemes, maximum classification gain and EMNSD classification, which allow an unequal number of blocks in each class and provide almost the same performance have been introduced in [Jos95]. It has been shown that in a subband coding system, use of these classification schemes results in an improvement over Chen-Smith classification (0.1-0.5 dB for the 512×512 Lenna [Jos95]).

Proposing adaptive schemes which use features other than the gain for classification is not new. Pearlman [Pea90] proposed an ADCT image coding system in which a constant target distortion is assigned to each block.

In this chapter, the spectral characteristics of the image blocks are used as a feature for classification. A VQ with an appropriate distortion measure is designed to split the spectral space into a prespecified number of classes. The performance and complexity of the proposed classification scheme are investigated for an adaptive wavelet transform coding system and compared with those of gain classification schemes, specifically EMNSD classification. It is shown that spectral classification improves the rate-distortion performance at a lower (sometimes significantly) complexity cost.

The organization of the chapter is as follows. Section 3.2 introduces spectral classification and Section 3.3 provides a complexity analysis. Section 3.4 presents

the application of spectral classification in a wavelet-based image coding system. Section 3.5 shows how to design an optimal 2-D spectral classifier which is used in a 2-D DPCM framework.

3.2 Spectral Classification

In image coding systems, gain-based classification techniques use the gain of each block as a measure of block activity [Che77]. Based on this measure, the classifier assigns to each block one of a finite number of activity classes. The image blocks are then quantized using quantizers that are appropriately designed for the activity class to which they correspond. Although gain-based classifiers perform reasonably well in terms of improving the coder's overall rate-distortion performance, their capability in separating dissimilar blocks into different classes is limited. For example, Figure 3.1 shows three blocks consisting of only black and white pixels. Since the gain values of these blocks are exactly the same, any gain-based classifier assigns them to the same class. However, from a spectral point of view these three blocks are significantly different and therefore should be quantized differently.

Indeed, the local spectrum of the signal is another feature which can, and perhaps should, be used for classification. In this chapter, we propose a method for classification which uses the block spectral content (including the block ac-energy) to perform the classification task.

Specifically, the 2-D image blocks to be classified are first converted into a 1-D sequence by zigzag scanning the block [Pea90]. Then, the linear prediction (LP) coefficients of the resulting 1-D signal (represented by $X(z)$) are used to

characterize the spectral behavior of the signal. In other words, the signal $X(z)$ is modeled by an all-pole filter (M^{th}-order autoregressive (AR) model) $H(z) = \sigma/A(z)$, where $A(z) = \sum_{i=0}^{M} a_i z^{-i}$ and $a_0 = 1$. Well-known techniques can be used to compute the LP coefficients ($a_i,\ i = 1, \cdots, M$) corresponding to the optimal filter (minimizing the mean squared prediction error) [Buz80].

Spectral classification is performed by vector quantizing these LP coefficients. To quantize the LP coefficients, we need to use a distortion measure that ultimately minimizes the mean squared prediction error in the image domain. The Itakura-Saito distortion measure, first developed for quantization of speech spectral parameters, is such a measure [Buz80, Ita68]. For each block, the VQ designed based on the Itakura-Saito distortion measure selects the best AR model from among a finite number of AR models in the codebook (all having a fixed and predefined order). The codebook contains a set of codevectors each described by a vector of LP coefficients; each vector defines one class. The spectral classifier finds the best match for each block and uses the corresponding codeword as the classification index (class) associated with the block.

The algorithm used for the design of the VQ is the generalized Lloyd algorithm [Lin80] in which the basic idea is to start from an initial codebook and iteratively improve it. The algorithm consists of two steps: (i) Finding the best partition of the space for a given codebook (generalized nearest neighbor rule) and (ii) determining the best codebook for a given partition of the space (generalized centroid rule). Because the average distortions at successive steps of the algorithm form a non-increasing sequence, convergence is guaranteed. The process is terminated when the rate of reduction in average distortion falls below a prescribed stopping threshold. The main steps needed for designing such a VQ

are summarized below; details can be found in [Buz80].

- Generalized nearest neighbor distortion calculations

  For a given input sequence, we wish to select the codevector (all-pole filter) that minimizes the Itakura-Saito distortion measure. Buzo et al. [Buz80] have shown that this is equivalent to finding the filter that minimizes $\frac{\alpha}{\sigma^2} + \ln\ \sigma^2$, where α is the residual energy and can be computed as follows:

$$\alpha = \sum_{n=-\infty}^{\infty} r_a(n) r_x(n) = r_a(0) r_x(0) + 2 \sum_{n=1}^{M} r_a(n) r_x(n), \tag{3.1}$$

  where

$$r_a(n) = \sum_{k=0}^{M-n} a_k a_{k+n},\ n = 0, 1, \cdots, M, \tag{3.2}$$

  and $r_x(n)$ is the autocorrelation of the input sequence.

- Generalized centroid calculations

  For each class, we wish to choose the best representative LP vector. The autocorrelation sequences of all blocks in the same class can be averaged to find an average autocorrelation sequence and then by solving the Wiener-Hopf equations for the average autocorrelation sequence the coefficients of $H(z) = \sigma / A(z)$ can be calculated. These coefficients represent the generalized centroid of the LP vectors [Buz80].

Using a large training sequence of monochrome images, a classification VQ is designed based on the Itakura-Saito distortion measure. The resulting VQ is used to classify images outside the training sequence. We refer to this classification method as spectral classification.

Classification image for the 512×512 Lenna using 16×16 blocks and a 4^{th}-order filter is shown in Figure 3.2. As can be seen in Figure 3.2, spectral classification is successful in separating blocks with different levels of activity.

3.3 Computational Complexity

In this section we compare the complexity of the proposed spectral classification scheme with that of EMNSD classification. Table 3.1 provides the number of operations needed to classify N blocks of size $L \times L$ into K classes. Clearly, the computational complexity of spectral classification is $\mathcal{O}(N)$. In the gain-based classification schemes of Chen-Smith [Che77] and those proposed in [Jos95], the block gain values need to be computed and sorted. Using quick sort, the complexity is at least $\mathcal{O}(N \log N)$ ($\mathcal{O}(N^2)$ in the worst case) [Pre92]. So, spectral classification is less complex than gain classification.

Another very important observation is that the spectral classifier can be implemented as an HTVQ which performs the encoding operation using a lookup table, thus eliminating the need for additions and multiplications altogether. We have actually implemented the spectral classifier as an HTVQ and will report the results in the next section.

Finally, note that spectral classification is a one-pass process as each block can be classified independently. However, gain-based classification is a two-pass process: The gain of all blocks must be computed in the first pass and sorted in the second pass. This fact, makes spectral classification more desirable from an implementation point of view [Che96].

In the following section, we will make more definitive statements on the

efficacy of the proposed spectral classification scheme in the context of a wavelet-based image coding system.

3.4 Adaptive DWT Coding of Images

Now we consider an adaptive image coding system based on the DWT. Here, our focus is on comparing spectral classification and EMNSD in a simple-to-understand DWT encoder and not on developing a very high performance image coding system.

In the encoder considered, the image is decomposed into 22 subbands (see Figure 3.3) using a 2-D separable DWT based on the 9-7 biorthogonal spline filters of [Ant92]. Each of the 22 subbands is then encoded by an appropriately designed encoder and transmitted. For a given design rate, the rate-distortion performances of the quantizers are used to perform an optimal bit assignment [Sho88, Woo92]. At the receiver, the received signal is decoded and a replica of the original image is reconstructed using the inverse filters.

Our classification strategy in this section is similar to Method 1 in [Jos95]. The low frequency subband (LFS) is not classified; the block size for classification in high frequency subbands (HFS's) is adjusted according to the decimation factor, so that each block corresponds to a 16×16 block in the original image, i.e., block size of 1×1 for subbands 1-3, 2×2 for subbands 4-6, and 4×4 for other subbands in Figure 3.3. The HFS's are split into three groups with different frequency orientations, i.e., subbands 1, 4, 7, 10, 11, 12 , and 13 in the vertical group, subbands 2, 5, 8, 14, 15, 16, and 17 in the horizontal group, and subbands 3, 6, 9, 18, 19, 20, and 21 in the diagonal group. A single classification map is

used for each group, so that blocks corresponding to the same spatial location have the same classification index. The three classification tables are obtained by classifying subbands 7, 8, and 9 into two classes using spectral or EMNSD classification. To compare the two classification schemes in a meaningful way, the same quantizers are used in both. All quantizers used are entropy-constrained trellis-coded quantizers (ECTCQ's) designed for the so-called generalized Gaussian distribution (GGD) [Far84] with parameter 0.6. The ECTCQ scheme used in this work is a modified version [Lee93] of the scheme first proposed in [Fis92] and then improved in [Mar94]. The ECTCQ proposed in [Lee93] places a symmetry constraint on the reproduction codebook. This constraint, while essentially costing no performance loss, simplifies the implementation of the entropy coder used to encode the ECTCQ output. The entropy coder used here is an arithmetic coder.

For 512×512 Barbara and Lenna images, Tables 3.2 and 3.3 summarize the PSNR's of the proposed system using different classification methods: Spectral classification, HTVQ-implemented spectral classification, EMNSD classification and no classification. The reconstructed Lenna images using the spectral classification method are shown in Figures 3.5 and 3.6. Table 3.4 summarizes the simulation results for other images at a design rate of $r = 1$ bit/pixel.

The superiority of spectral classification is established once again in Tables 3.2, 3.3, and 3.4. Spectral classification consistently outperforms EMNSD classification by 0.1-0.4 dB and leads to 0.5-1.1 dB improvement compared to the case where no classification is employed. It is important to note that even the HTVQ-implemented spectral classification scheme outperforms EMNSD classification. This is particularly important from a real-time implementation point

of view as HTVQ has minimal computational complexity.

3.5 2-D Spectral Classification

In this section, first we generalize the idea of spectral classification to a 2-D spectral classification. Then, the 2-D spectral classification is used in a 2-D DPCM structure to encode the LFS.

To have a causal 2-D linear prediction, we may predict each pixel using three neighboring pixels as in Figure 3.7. To be more specific, the predicted pixel, $\hat{x}(m,n)$, can be written as:

$$\hat{x}(m,n) = \rho_h x(m-1,n) + \rho_v x(m,n-1) + \rho_d x(m-1,n-1). \tag{3.3}$$

The corresponding prediction error for each pixel is defined as:

$$e(m,n) = x(m,n) - \hat{x}(m,n). \tag{3.4}$$

The objective is finding the LP coefficients that minimize the resulting mean-squared prediction error $E[\mathrm{E}^2(m,n)]$.[1] It can be easily shown that the optimal coefficients $\hat{\boldsymbol{\rho}}=[\hat{\rho}_h\ \hat{\rho}_v\ \hat{\rho}_d]^T$ are the solutions of the following equation:

$$\mathbf{A}\hat{\boldsymbol{\rho}} = \mathbf{r}, \tag{3.5}$$

where

$$\mathbf{A} = \begin{bmatrix} r(0,0) & r(1,-1) & r(0,-1) \\ r(-1,1) & r(0,0) & r(-1,0) \\ r(0,1) & r(1,0) & r(0,0) \end{bmatrix}, \tag{3.6}$$

[1] we use capital letters for random variables and small letters for specific values of the random variable.

and

$$\mathbf{r} = \begin{bmatrix} r(1,0) \\ r(0,1) \\ r(1,1) \end{bmatrix}, \tag{3.7}$$

where r is the corresponding autocorrelation function computed based on ensemble averages.

Using a set of LP coefficients, the corresponding mean-squared prediction error is given by:

$$\begin{aligned} E[\mathrm{E}^2(m,n)] &= r(0,0)[1+\rho_h^2+\rho_v^2+\rho_d^2] + 2r(1,0)[\rho_v\rho_d - \rho_h] \\ &\quad +2r(0,1)[\rho_h\rho_d - \rho_v] + 2r(1,-1)[\rho_h\rho_v] + 2r(1,1)[-\rho_d]. \end{aligned} \tag{3.8}$$

This error is minimized when $\boldsymbol{\rho} = \hat{\boldsymbol{\rho}}$ given by (3.5).

In a block-based 2-D DPCM image coding system, the image is segmented into $L \times L$ blocks and a suitable set of LP coefficients can be used to model each block. However, the quantization of the corresponding coefficients results in a large amount of overhead information. This suggests the use of a VQ for block classification and quantization of the corresponding LP coefficients. The direct use of conventional VQ design procedures for prediction coefficients minimizes the mean-squared error in the LP coefficient space domain. This minimization is not satisfactory for two reasons : (i) It does not guarantee the minimization of the mean-squared prediction error in (3.8), and therefore is not the best solution and (ii) there is no reason to guarantee that the resulting coefficients correspond to a stable filter. In what follows, we propose a method to design a VQ which does not suffer from the aforementioned weaknesses and optimally performs the classification and quantization operations in one step. This is a 2-D extension of the spectral classification method.

To extend the idea to the 2-D case, the mean-squared prediction error of (3.8) is used as the distortion measure for designing the VQ. Using a set of LP coefficients ($\boldsymbol{\rho}$), the distortion can be defined as:

$$d(X, \hat{X}(\boldsymbol{\rho})) = E[(X - \hat{X}(\boldsymbol{\rho}))^2] = E[\mathrm{E}^2(m, n)]. \tag{3.9}$$

The generalized Lloyd algorithm can be adapted to achieve a local minimum for this distortion measure by iteratively applying the following steps in the design procedure. Each Voronoi region of the VQ forms a class for the corresponding blocks.

In the generalized centroid step of the algorithm, having a group of blocks belonging to a given class C, we determine the best LP coefficient vector for predicting the pixels within that class. This problem is the same as the original 2-D linear prediction problem and the solution is similar to the solution of (3.5). The only difference is that in this case all expectations are conditional expectations and, using a training sequence to design the VQ, we should only consider those blocks which belong to class C. In other words, the autocorrelation function in (3.5) must be replaced by the following conditional autocorrelation function:

$$r'(m', n') = E[X(m+m', n+n')X(m, n) \mid (m, n), (m+m', n+n') \in C]. \tag{3.10}$$

The optimal codevector is the solution of (3.5) when r is replaced by r'.

In the generalized nearest neighbor step, we decide which LP coefficient vector in the codebook is the best representative of an input block. This can be done by computing the conditional autocorrelation function of each block and finding the codevector which minimizes the distortion defined by (3.9).

Since the distortion is decreased at each step of the algorithm, the convergence of the algorithm to a local minimum is guaranteed. The algorithm will be

terminated when the rate of improvement is less than a prescribed small value.

Now, we propose an adaptive encoding structure similar to that of Section 3.4 and try to reduce the complexity of the system. This reduction in complexity can be achieved by replacing the wavelet packet used in Section 3.4 with a simpler one. The proposed image coding system contains the following parts. First, the input image is decomposed into 7 subbands using two levels of a 2-D separable DWT (Figure 3.8). Each of the subbands is then encoded separately. After segmenting the LFS into 8×8 blocks, each block is encoded using a 2-D predictive ECTCQ. Each LFS block is first normalized by removing its mean and dividing the pixel values by its standard deviation. Then, the normalized blocks are classified into one of four classes by the proposed 2-D spectral classifier designed using the aforementioned generalized Lloyd algorithm. The process of classification is the same as the generalized nearest neighbor part of the VQ design algorithm. Based on the resulting codeword, a 2-D DPCM encoder using the corresponding LP coefficients for each class encodes the normalized LFS coefficients.

Each HFS neighboring the LFS is segmented into 4×4 blocks and is classified into one of two classes using spectral classification proposed in Section 3.2. The resulting classification tables are used to classify 8×8 blocks in subbands 4, 5, and 6 as it is shown in Figure 3.8.

The overhead information needed to be sent to the receiver includes: (i) the mean and variance of each LFS block, (ii) the variance of each class in each HFS, (iii) the classification tables, (iv) the size of the image, and (v) the design rate. We have used 8 bits to quantize each mean and variance parameter. The variance of those blocks which have been assigned zero bits by the optimal bit

allocation procedure need not to be transmitted. The simulation results and the reconstructed images for the 512×512 Lenna are shown in Figures 3.9, 3.10, and 3.11. The classification gain is about 0.7 dB at rate $r = 1$ bit/pixel. The performance of the 7-band system is up to 0.6 dB worse than the performance of the 22-band system. However, the encoding and decoding complexity of the 7-band system is quite small, thus making it an interesting candidate for situations where the complexity is a serious practical constraint.

3.6 Comparison with Existing Image Compression Systems

In this section, we compare our results with, to the best of our knowledge, some of the best results reported in the literature. In previous sections, our emphasis was on developing a simple image coding system as a framework to compare different classification schemes. Although the system presented in Section 3.4 exhibits a rate-distortion performance comparable to or better than, to the best of our knowledge, all existing image coding systems in the literature, it is possible to improve the performance, specially at high bit rates, albeit with some increase in complexity. For a complete survey of different possibilities, the interested reader is referred to [Jos95].

In this section, we propose following modifications to improve the performance:

- Classifying all subbands into 4 classes.
- Compressing the classification tables.

- Using a different quantizer for each class in different subbands.

Now, let us explain the characteristics of the image coding system. We use the 22-band decomposition of Figure 3.3. Each subband is classified into 4 classes using EMNSD classification scheme. We use 2×2 blocks for subbands 0-6 and 4×4 blocks in the remaining subbands. For each subband, samples from blocks having the same classification index are grouped together into one class. Each class is modeled as having a GGD , with the shape parameter chosen from the set $\{0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0\}$. The ECTCQ designed for the corresponding GGD is used to encode the coefficients. (The bit rate is defined by an optimal bit allocation scheme.)

To compress the classification tables, we should exploit the dependence between the classification indices. In addition to the dependence between the classification maps of different subbands, there is also some dependence between the classification indices of spatially adjacent blocks from the same subband. Any one or both of these dependencies can be exploited to reduce the side information required for sending the classification maps.

Consider a subband coding system for images where each subband is classified into 4 classes and the classification map for each subband is sent as side information. Assume that the side information for a subband is being arithmetic coded. In the absence of any dependencies, a single probability table, where each entry corresponds to the probability of a class, is adequate. However, the intra-band and inter-band dependence of classification maps can be exploited as follows. Multiple conditional probability tables are maintained, one for each state. The state depends on the classification index of the previous block as well as the block from the lower frequency subband corresponding to the same spatial location.

For example, let $C_8(i,j)$ denote the classification index for block (i,j) from subband 8 in Figure 3.3. Then the choice of probability table for encoding $C_8(i,j)$ is dependent on $C_8(i,j-1)$ as well as $C_5(i,j)$. The conditional probability tables have to be known at the decoder. Thus, if subbands 5 and 8 have 4 classes, then 16 tables have to be sent to the decoder. If all classes have nonzero rates, the side information for sending all the probability tables can be prohibitive. But if some classes have zero rates, all the classes having zero rates can be combined into a single class. For example, suppose that subbands 5 and 8 have only 1 class having nonzero rate, then we can modify the classification map for each subband, so that it contains 2 classes. In that case the number of probability tables is reduced to 4. Since higher frequency subbands typically tend to have many classes having zero rates, it is possible to exploit both intra-band and inter-band dependencies. For low frequency bands, it is advantageous to exploit only one of the dependencies. Thus for the 22-band decomposition, for bands 10-21, both dependencies are exploited. For bands 1-9, only inter-band dependence is exploited whereas for band 0 only intra-band dependence is exploited. The entries in the probability tables are quantized to 5 bits and adaptive arithmetic coding [Wit87] is used for encoding the classification maps. A side rate reduction of $15-20\%$ can be obtained using this method.

Figure 3.12 compares the rate-distortion performance of the above system with some of the best results reported in the literature [Sai96, Sha93, Tau94, Xio96]. As can be seen in Figure 3.12, to the best of our knowledge, the obtained PSNR's are equal to or better than any other results in the literature. Note that unlike the systems presented in previous sections, the performance of this system is not very sensitive to the choice of classification method (less than

Table 3.1: Comparison of the Complexity of Spectral Classification with EMNSD Classification. Number of Blocks=N, Block Size=$L \times L$, Number of Classes=K, Number of Iterations for EMNSD=I (20–30), Order of AR Model=M.

	EMNSD	Spectral
Multiplications	$NL^2 + IN$	$[(M+1)L^2 - \frac{M(M+1)}{2} + (M+1)\log_2 K]N$
Additions	$3NL^2 + 3N$	$[(M+3)L^2 - \frac{M(M+1)}{2} + (M+2)\log_2 K]N$
Divisions	$N + 3K + 2$	N
Comparisons	$2K$	K
Sorting	$CN\log_2 N$ to N^2	–
Order	$N\log_2 N$ to N^2	N

0.1 dB). This is because of the fact that first we over-classify the subbands by classifying them into 4 classes and then we reduce the overhead by compressing the classification tables. Over-classification reduces the performance degradation due to the limitations of the classification scheme. For example, consider a subband where 75% of the blocks have high activity and 25% of the blocks have low activity. If the blocks are divided into two equally populated classes, then one of the classes will contain a mixture of high and low activity blocks in equal proportion. This is clearly not desirable. Using a smart classification scheme which allows unequally populated classes, results in having high and low activity blocks separated. In this example, classifying the subband into 4 equally-populated classes will also solve the problem, albeit at the cost of an increase in side information if we do not compress the classification tables.

Table 3.2: Simulation Results (PSNR in dB); 512 × 512 Barbara.

Design Rate	Spectral		Spectral(HTVQ)		EMNSD		No Classification	
	Actual Rate	PSNR (dB)	Actual Rate	PSNR (dB)	Actual Rate	PSNR (dB)	Actual Rate	PSNR (dB)
0.25	0.267	28.03	0.267	28.01	0.269	28.08	0.259	27.61
0.50	0.500	31.58	0.493	31.46	0.489	31.21	0.471	30.56
1.00	1.024	36.64	1.015	36.37	1.002	36.04	1.035	35.76

Table 3.3: Simulation Results (PSNR in dB); 512 × 512 Lenna.

Design Rate	Spectral		Spectral(HTVQ)		EMNSD		No Classification	
	Actual Rate	PSNR (dB)	Actual Rate	PSNR (dB)	Actual Rate	PSNR (dB)	Actual Rate	PSNR (dB)
0.25	0.264	34.48	0.263	34.44	0.265	34.46	0.241	33.31
0.50	0.505	37.43	0.501	37.33	0.507	37.39	0.454	36.15
1.00	1.030	40.81	1.018	40.68	1.017	40.68	0.990	39.89

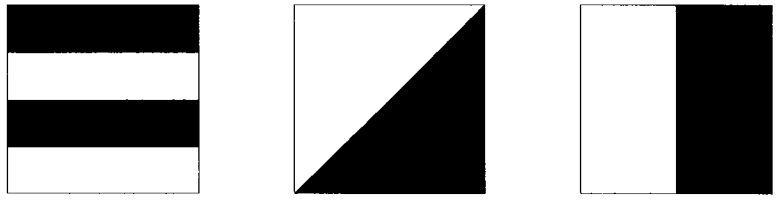


Figure 3.1: An Example of Three Blocks with the Same Ac-Energy and Different Spectrum.

Table 3.4: Simulation Results (PSNR in dB); $r = 1$ bit/pixel.

Image	Spectral		Spectral(HTVQ)		EMNSD		No Classification	
	Actual Rate	PSNR (dB)	Actual Rate	PSNR (dB)	Actual Rate	PSNR (dB)	Actual Rate	PSNR (dB)
Barbara	1.024	36.64	1.015	36.37	1.002	36.04	1.035	35.76
Goldhill	1.069	37.34	1.074	37.34	1.077	37.29	1.009	36.71
Girl	1.058	42.26	1.051	42.17	1.049	42.12	1.000	41.74
Airplane	0.999	41.42	0.994	41.30	0.974	40.87	0.985	39.78
Boat	1.076	35.54	1.079	35.53	1.052	35.17	0.980	34.47
Lenna	1.030	40.81	1.018	40.68	1.017	40.68	0.990	39.89

Figure 3.2: Spectral Classification Image; 512×512 Lenna.

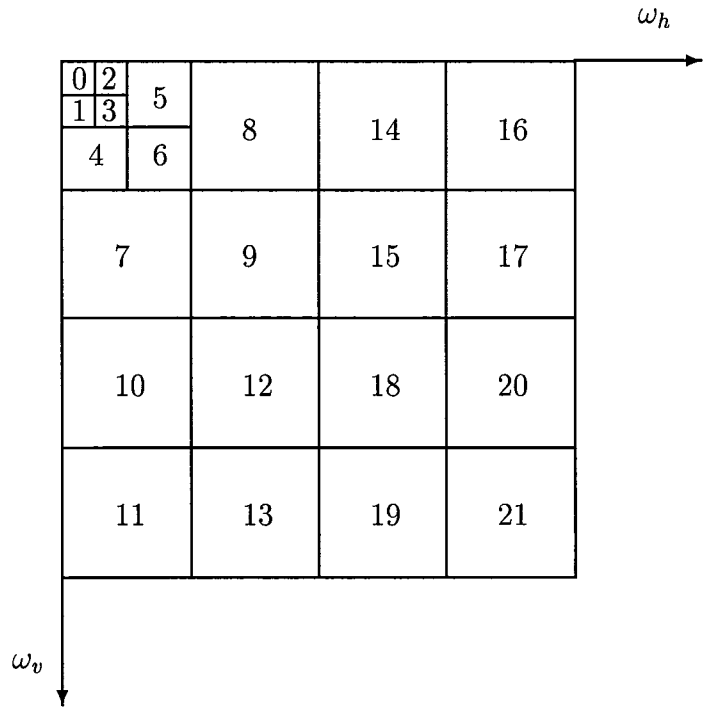


Figure 3.3: The 22-Band Decomposition.

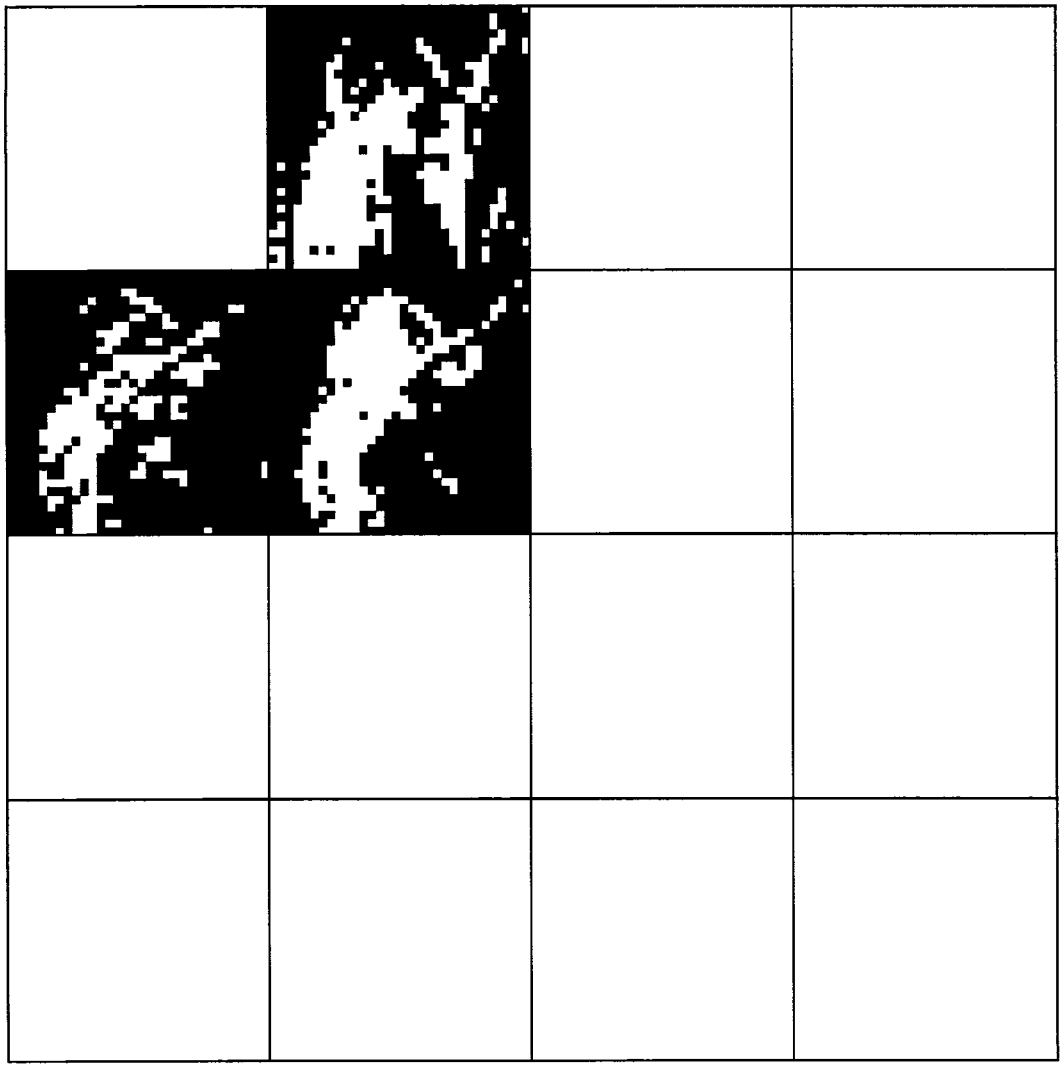


Figure 3.4: Classification Image for Subbands; 512×512 Lenna.

Figure 3.5: Reconstructed Image for the 22-Band Wavelet Coding System; 512×512 Lenna; $r = 0.25$ bits/pixel.

Figure 3.6: Reconstructed Image for the 22-Band Wavelet Coding System; 512×512 Lenna; $r = 0.5$ bits/pixel.

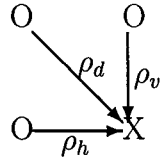


Figure 3.7: Two-Dimensional Linear Prediction.

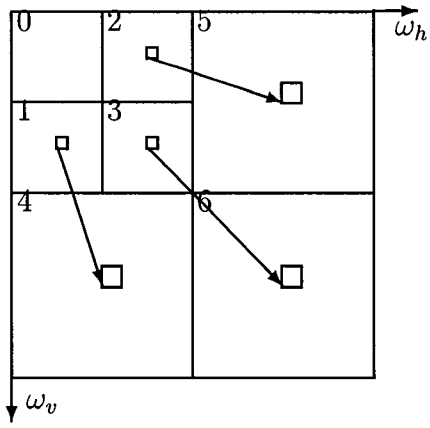


Figure 3.8: Classification Map and Enumerating Subbands.

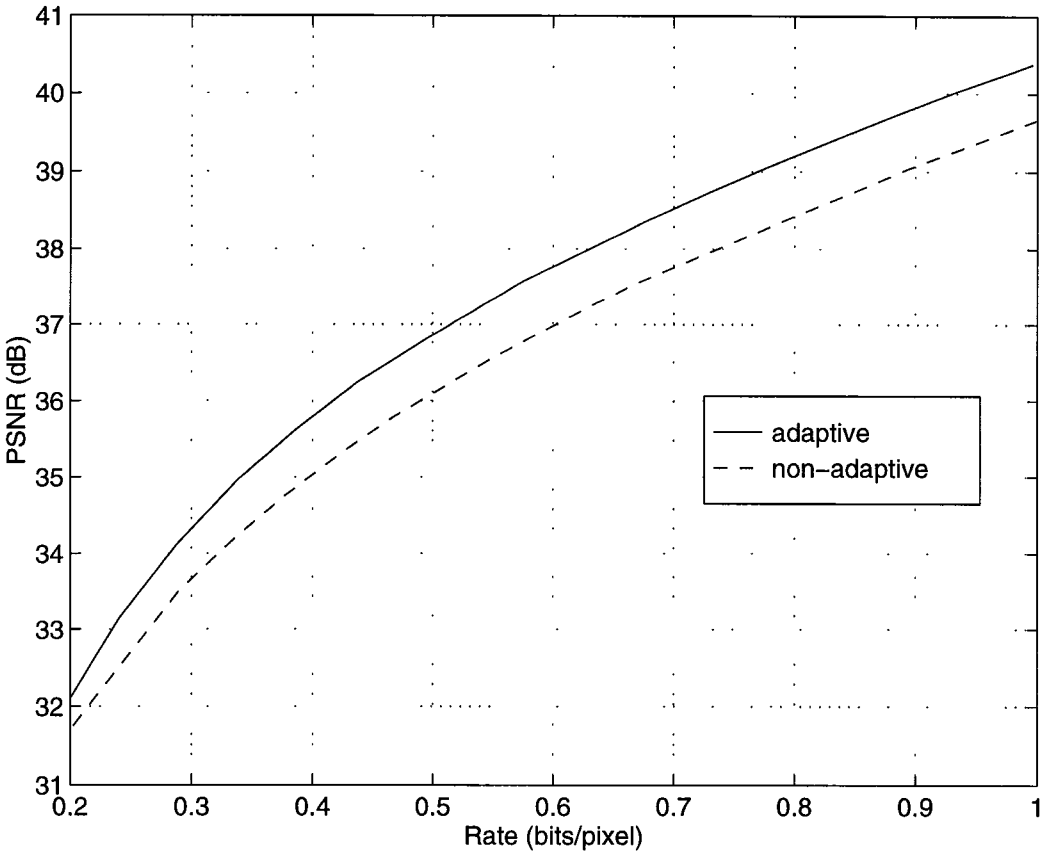


Figure 3.9: Performance of the Image Coding System Using 7-Band Decomposition and 2-D DPCM; 512×512 Lenna.

Figure 3.10: Reconstructed Image for the 7-Band Wavelet Coding System; 512×512 Lenna; $r = 0.25$ bits/pixel.

Figure 3.11: Reconstructed Image for the 7-Band Wavelet Coding System; 512×512 Lenna; $r = 0.5$ bits/pixel.

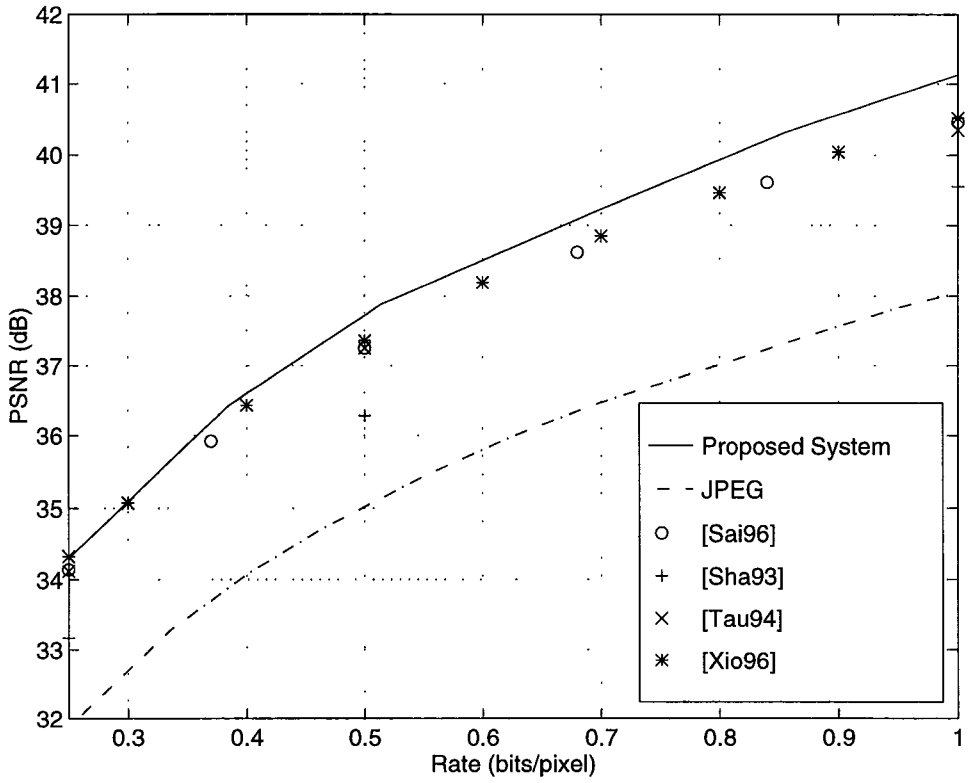


Figure 3.12: Performance of Different Image Coding Systems; 512×512 Lenna.

Chapter 4

A Scalable Wavelet Image Coding System

4.1 Introduction

Progressive transmission of signals is a mechanism by which the encoder's output is transmitted in groups of bits and the decoder produces a higher quality replica of the signal based on receiving each new group of bits. Figure 4.1 illustrates a simple progressive transmission system based on a uniform quantizer. A 128-point source signal, shown in Figure 4.1 (a), is progressively transmitted and reconstructed with 128, 256, and 384 bits (1, 2, and 3 bits/sample), as shown in Figures 4.1 (b), (c), and (d), respectively. Upon receiving each group of 128 bits, the receiver updates the reconstructed signal and provides a more accurate representation of the signal.

Progressive transmission is a desirable feature in many practical signal transmission situations such as telebrowsing and database retrieval. Progressive trans-

mission also provides the opportunity for interrupting transmission when the quality of the received signal has reached an acceptable level or when the receiver decides that the received signal is of no interest. Likewise, in applications where the receiver is more interested in specific parts of the signal rather than the entire signal (e.g., content-based image transmission), a valid question is how to transmit and reconstruct a signal with different levels of quality (distortion) in different temporal or spatial regions. For example, after receiving a rough reproduction of a medical image in a telemedicine application, the radiologist, at the receiver, may want to highlight a part of the image and request a higher fidelity (or even lossless) replica of only the highlighted area.

Although the image coding systems proposed in Chapter 3 provide a very good rate-distortion performance, they are not rate-scalable and hence not suitable for progressive transmission and image browsing in which a higher quality image is gradually constructed by progressively transmitting new bits. The good performance of these systems is partly due to the use of ECTCQ's followed by arithmetic coding which amounts to a non-scalable quantization structure and relatively high coding complexity. The focus of this chapter is on developing an efficient, low-complexity, and scalable image coding system **without** using arithmetic coding.

Pruned Tree-Structured Vector Quantizer (PTSVQ) is a low-complexity quantizer scheme which offers a good rate-distortion performance while providing successive approximation of the source (thus, rate-scalability) due to the fact that optimal trees are nested [Cho89]. However, there are practical limitations on designing good PTSVQ's at high rates. In particular, designing a VQ with a good out-of-the-training-sequence performance requires a sufficiently large num-

ber of training vectors so as to present a valid empirical approximation of the statistics of the source. The robustness of PTSVQ to distribution mismatch is less than that of VQ. To investigate this fact, we design 16-dimensional VQ's and PTSVQ's operating at different bit rates. We use 4×4 blocks of the 512×512 PEPPERS as our training sequence. Figure 4.2 shows the rate-distortion performance of the VQ's and PTSVQ's for the training sequence and a test sequence (4×4 blocks of the 512×512 Lenna). As can be seen from Figure 4.2, the performance degradation of a PTSVQ due to distribution mismatch is more than that of a VQ operating at the same bit rate.

In subband image coding, after decomposing an image into its subbands, encoding the LFS requires a high bit rate and the number of samples in the LFS is much less than the number of pixels in the original image. Even for low vector dimensions, it is practically impossible to design a VQ with a high enough bit rate which does not suffer from the distribution mismatch. This problem also arises when the HFS's are classified into two or more classes. One of the basic ideas behind the classification is to separate high and low activity regions of a subband and to spend more bits for the high activity part. Usually, the number of blocks in the high activity class is a small portion of the total number of blocks in each subband. Thus, not only does classification reduce the number of training vectors, but also it amplifies the need for a higher bit rate quantizer.

In what follows, we suggest a solution for the above-mentioned problems and propose an image coding system which is simple and scalable (both in resolution and in rate). Section 4.2 introduces a hierarchical PTSVQ in which a multistage PTSVQ (MS-PTSVQ) is used. Section 4.3 suggests an appropriate way to choose 2-D blocks. Section 4.4 includes simulation results.

4.2 Multi-Stage PTSVQ

To overcome the problems in designing a robust PTSVQ using a relatively short training sequence, we use a tree-structured encoder in which the first stage is a gain-based classifier. Here, we use the EMNSD classification scheme to classify the vectors into two classes based on their gain values. The second stage of the encoder is a PTSVQ with a maximum bit rate of R_{max}. If a rate higher than R_{max} is required, the quantization error of the last stage is computed and another PTSVQ is used to encode the error vectors. Depending on the maximum desirable encoding rate, additional stages can be used.

We must emphasize that use of such an MS-PTSVQ would not be wise if we were able to design a good high bit rate PTSVQ. However, practical limitations on the size of available training data and the corresponding distribution mismatch are the primary motivation for resorting to this multi-stage structure. It is important to recognize that only one stage of PTSVQ is used when the encoding rate is sufficiently low and, as it is shown in Figure 4.3, an ordinary PTSVQ is enough for coding the low activity regions. Another important issue is the memory requirement of the PTSVQ. Using MS-PTSVQ instead of PTSVQ reduces the required memory of the system. The number of nodes in a PTSVQ depends on the source (in addition to its dependency on the rate); however, this number is source independent for complete trees. The memory requirement of tree-structured vector quantizers (TSVQ's) increases exponentially by rate and this will cause practical limitations on the rate of the quantizers. For example, a 4-dimensional TSVQ with rate $r = 5$ bits/sample contains 2,097,151 nodes and storing the codebook would be impractical in most applications. A two-stage TSVQ for the same rate, with equal rates for the two stages, needs only 4,094

nodes to be saved.

To improve the performance of the MS-PTSVQ at high bit rates, we may normalize the vectors at each stage of the MS-PTSVQ. This normalization is usually done by removing the mean of vectors and dividing the mean-removed vectors by their standard deviation. Since HFS's have zero means, there is no need for removing the mean of HFS's. The effects of the normalization in improving the performance and increasing the complexity are discussed in Section 4.4.

4.3 Choosing the Block Sizes

After three levels of subband decomposition, we enumerate the image subbands as in Figure 4.4. Statistical characteristics of the subbands indicate that subbands 2, 5, and 8 have higher row correlations than column correlations; the opposite is true for subbands 1, 4, and 7. Using non-square blocks in the classification procedure and vector quantization exploits this higher correlation and provides a better performance. To show the effect of non-square blocks, we consider two different systems. In System A, we use 2×2 blocks to classify subbands 1, 2, and 3. The corresponding block sizes for higher subbands in the hierarchy are 4×4 for subbands 4, 5, and 6 and 8×8 for subbands 7, 8, and 9. The size of the vector quantizer blocks is tabulated in Table 4.1. In System B, to exploit the directional correlations in different subbands, we use non-square block sizes as described in Table 4.2. In both systems, 2×2 blocks are used for vector quantization of subband 0 and no classification is employed. The effects of changing the block sizes on the final results are investigated in the next section.

4.4 Simulation Results

Now, let us describe the overall image coding system. After three levels of wavelet decomposition based on the 9-7 spline filters designated No. 2 in [Ant92], the subbands neighboring the LFS are classified into 2 classes using the EMNSD classification algorithm. Other HFS's are classified based on the result of the classification in the corresponding lower resolution subbands as it is shown in Figure 4.4. This selection of classification map is because of the spatial dependency of the HFS's and is similar to the scheme presented in Chapter 3 to save in overhead. MS-PTSVQ's are used for the LFS and the high activity regions of the HFS's. PTSVQ's are designed for the low activity portions of the HFS's. An optimal bit allocation procedure is used to assign bits among different quantizers [Ris91].

To compare the performance of our system with some of the existing image coding results, first we choose four systems. The first system is a modified version of the system presented in Chapter 3 and provides one of the best available rate-distortion performances. This modified system, hereafter referred to as System I, uses three levels of decomposition and a scheme similar to Figure 4.4 for classification. Other specifications of System I, like using ECTCQ's, are the same as the system presented in Chapter 3. Note that System I is not rate-scalable (not suitable for progressive transmission) and is more complex than the proposed system. We use System I as a benchmark to measure the rate-distortion loss incurred by demanding low complexity and rate-scalability. Another system which has been used to evaluate the performance of the suggested system is the one reported in [Per94] (System II). System II is a low-complexity PTSVQ-based system providing scalability. System III is the same as the proposed

system (System B) but uses PTSVQ instead of MS-PTSVQ. The only difference between Systems III and IV is in the choice of vector dimensions. In System IV, we use pruned tree-structured scalar quantizers instead of PTSVQ's. Since the dimension of a scalar quantizer is one, we can design high bit rate quantizers and there is no need for a multi-stage structure. Note that theoretical results show that asymptotically a VQ has space-filling advantage, shape advantage, and memory advantage over a scalar quantizer [Loo89]. We include the results for System IV to study the tradeoff between the VQ advantages and the practical limitations on designing VQ's at high rates.

The codebooks of System II are designed based on 10 computerized tomography (CT) images [Per94]. The codebooks of Systems B, III, and IV are designed based on a set of 26 CT images including the 10 used for System II. In Figure 4.5 we compare the signal-to-noise ratio (SNR)[1] results of the aforementioned systems for the same 12-bit 512×512 CT images used in [Per94]. The results, using only two stages of PTSVQ without employing normalization, are the average SNR's of two images outside the training sequence. As it is shown in Figure 4.5, the proposed system (System B) provides excellent results for very low bit rates which at some points are even better than the results of System I. As the rate is increased, the performance results fall short of those of System I but rate-scalability, progressive transmission, and simplicity are preserved. On the other hand, the performance is always superior to that of System II although this comes at the cost of a higher encoding complexity. The proposed system provides more than 3 dB improvement over System III at high bit rates. Also, note that the memory requirement for the MS-PTSVQ is much less than that of

[1] SNR=$10 \log_{10} \sum_i (x_i - \overline{x})^2 / \sum_i (x_i - y_i)^2$

the PTSVQ. An interesting observation is that System IV outperforms System III at high bit rates. This is due to the practical limitations on designing VQ's which has more effects than the VQ advantages at high bit rates. As it is expected the performance of System III is superior to that of System IV at low bit rates. Also, note that not only does the proposed system, System B, overcome the practical design problems, but also System B utilizes the VQ advantages and outperforms System IV. Figures 4.6 and 4.7 show the original image and reconstructed CT image for System B at $r = 0.3$ bits/pixel, respectively.

Our simulation results show that choosing non-square block sizes, System B, will improve the results by up to 1.2 dB (compared to System A). This is because of more efficient use of the horizontal and vertical correlations in different subbands. Normalizing the vectors results in about 0.3 dB improvement at rate 0.35 bits/pixel. As we increase the bit rate, the gap becomes wider. Increasing the bit rate results in using a larger portion of the rate in the second stage of the MS-PTSVQ's. Also, the second stage is not used for some subbands when the rate is low and as a result normalization is not beneficial at low bit rates. Using normalization adds one division per pixel per stage to the complexity of the system.

We also compare our results with those of [Cos95] for the 512×512 Lenna in Figure 4.8. In [Cos95], embedded Zerotrees wavelet (EZW) coding approach of [Sha93] is combined with PTSVQ. The complexity of this system is comparable with that of System B. As it is shown in Figure 4.8, the performance of System B is about 0.5 dB better than that of [Cos95] at $r = 0.2$ bits/pixel.

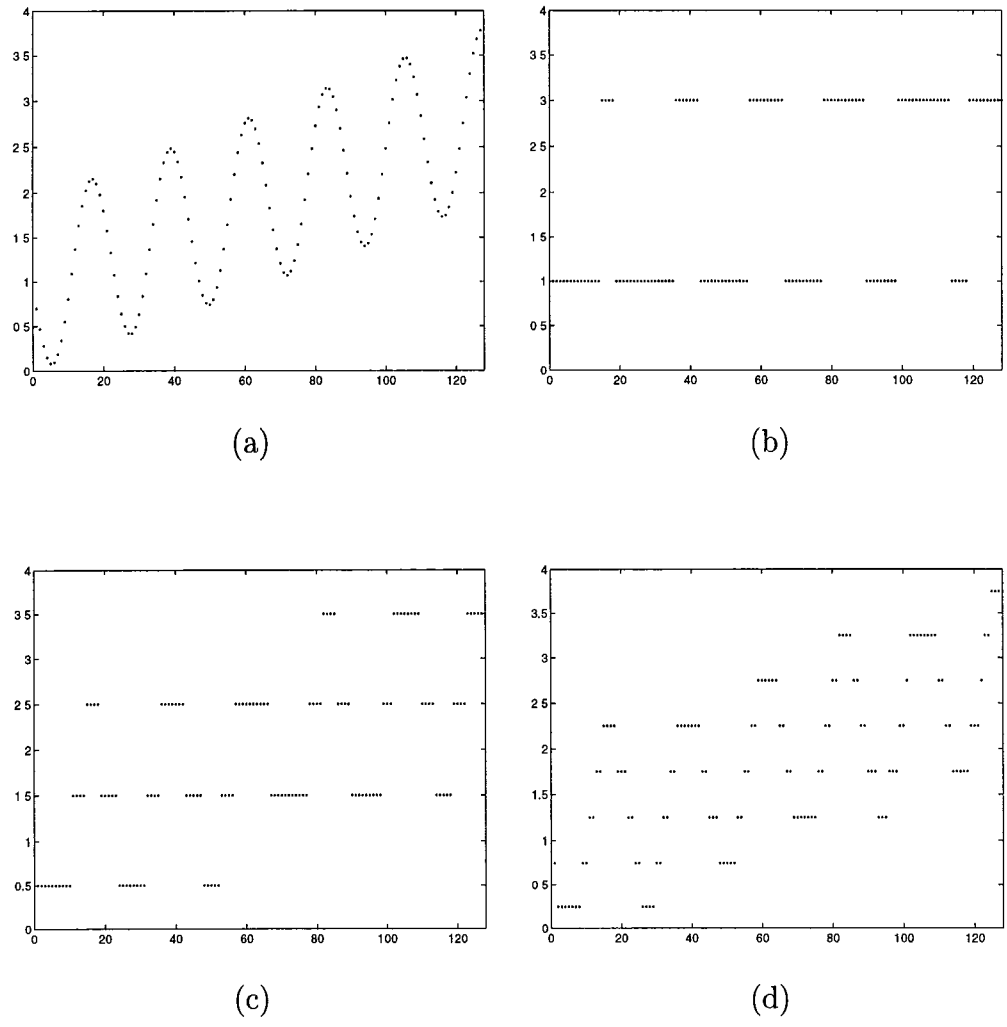


Figure 4.1: Illustration of Reconstructed Signal Quality in Progressive Transmission: (a) Original Signal, (b) First Refinement at 1 bit/sample, (c) Second Refinement at 2 bits/sample, and (d) Third Refinement at 3 bits/sample.

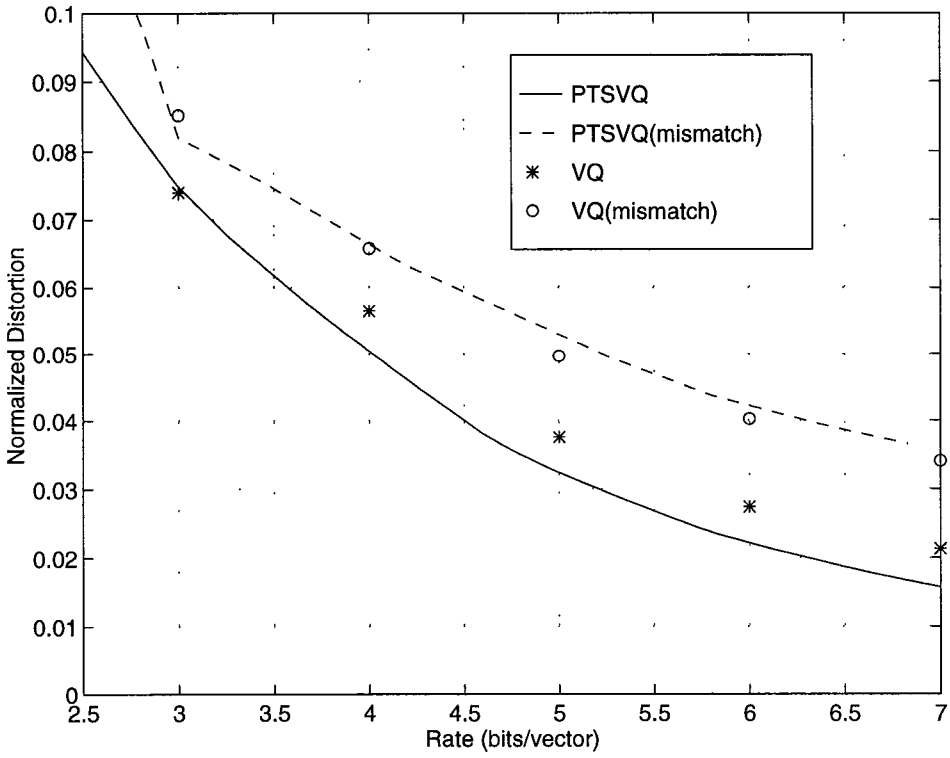


Figure 4.2: Effects of Mismatch on Full-Searched VQ and PTSVQ.

Table 4.1: Selection of Block Sizes in System A.

Subband	Classification	Vector Quantizer
0	–	2×2
1	2×2	2×2
2	2×2	2×2
3	2×2	2×2
4	4×4	2×2
5	4×4	2×2
6	4×4	2×2
7	8×8	4×4
8	8×8	4×4
9	8×8	4×4

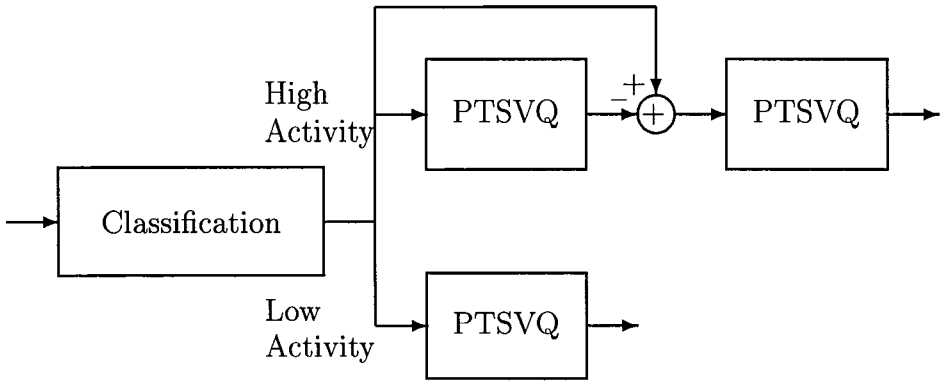


Figure 4.3: Encoder Block Diagram.

Table 4.2: Selection of Block Sizes in System B.

Subband	Classification	Vector Quantizer
0	–	2×2
1	4×1	4×1
2	1×4	1×4
3	2×2	2×2
4	8×2	4×1
5	2×8	1×4
6	4×4	2×2
7	16×4	8×2
8	4×16	2×8
9	8×8	4×4

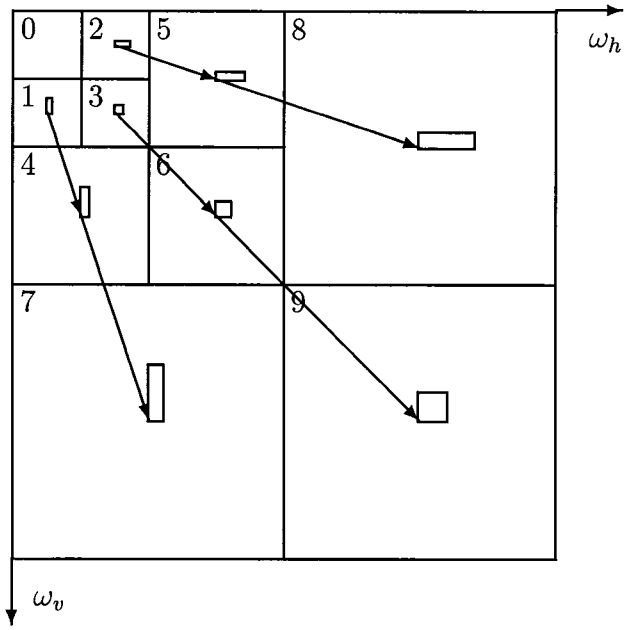


Figure 4.4: Classification Map and Enumerating Subbands.

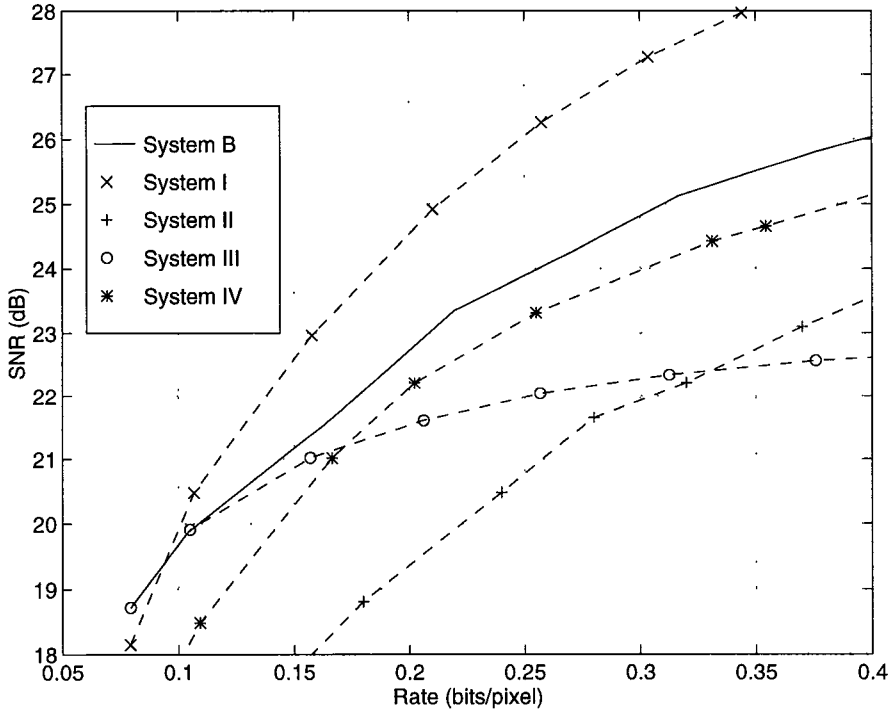


Figure 4.5: SNR Performance; CT Images.

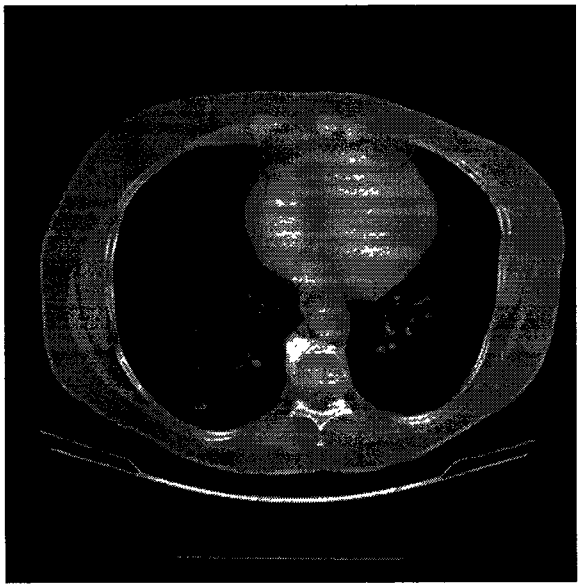


Figure 4.6: Original 512×512 CT Image.

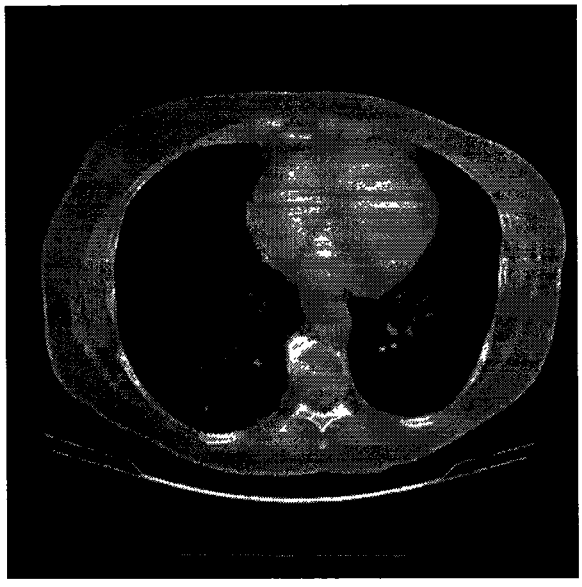


Figure 4.7: Reconstructed CT Image for System B; $r = 0.3$ bits/pixel.

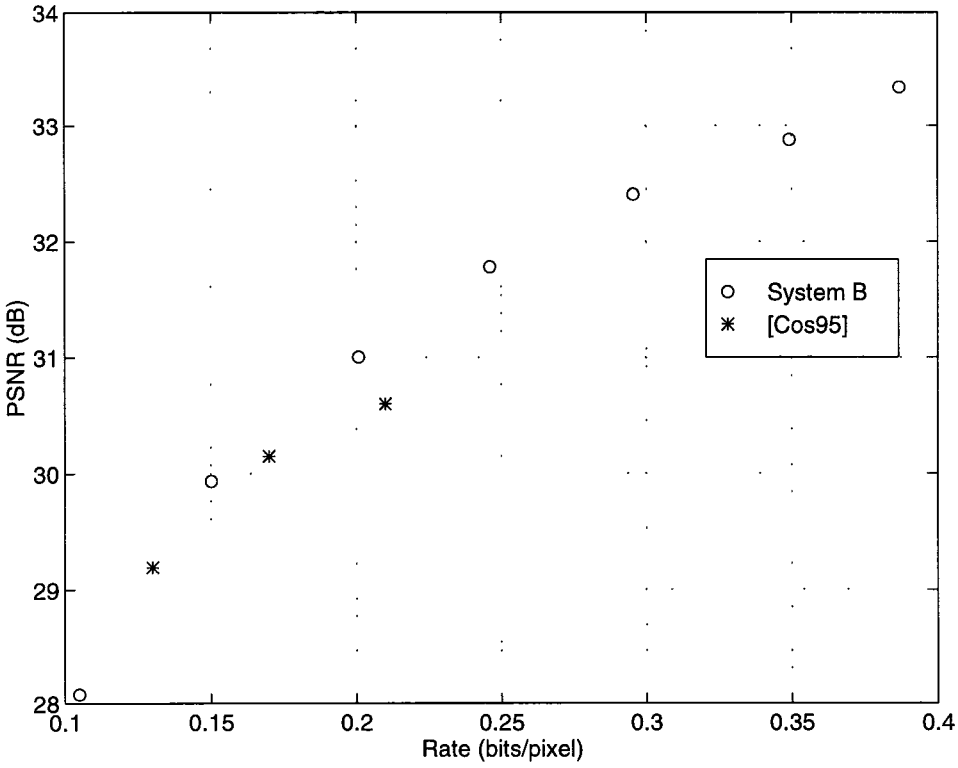


Figure 4.8: PSNR Performance; 512×512 Lenna.

Chapter 5

Fast Reconstruction of Subband-Decomposed Signals for Progressive Transmission

5.1 Introduction

We proposed a simple and rate-scalable subband image coding system in Chapter 4. Also, several rate-scalable subband image coding systems have been proposed in the literature which provide very good performances [Sha93, Sai96, Tau94]. These coding systems are strong candidates for multicasting and progressive transmission due to the fact that they offer resolution-scalability, rate-scalability, and good rate-distortion performance.

One of the problems in a DWT-based progressive transmission scheme is that the decoder, upon receiving a new group of bits, has to perform the inverse filtering operation to reconstruct the image. The conventional approach for doing

this is to apply the inverse filters on the decoded versions of all DWT coefficients to reconstruct a replica of the image. In this approach, even if only one DWT coefficient is refined, the reconstruction complexity will be the same as in the case when all coefficients are refined.

In a realistic progressive transmission scenario however, at each step of progression, only some of the DWT coefficients are refined and the other coefficients remain unchanged. Therefore, only a portion of the image pixels need to be updated after receiving the new bits. To reduce the complexity of reconstruction, we propose a method through which one can recompute only those pixels of the image that need to be updated. Not only does this method provide a fast reconstruction of the output, but also it provides the capability to update the output, without any increase in complexity, as the refinement of each DWT coefficient arrives. This feature of the proposed reconstruction scheme is not limited to progressive transmission systems and can be used in any packetized bit stream for an on-line reconstruction of the output.

In Section 5.2, we propose a new method for signal reconstruction in progressive transmission which allows for updating only the necessary portions of the signal instead of reconstructing the whole thing. The complexity of this new reconstruction approach is proportional to the number of refined coefficients. Section 5.3 generalizes the approach of Section 5.2 to a general filter bank structure. In Section 5.4, we provide two specific examples to illustrate the efficacy of the proposed fast reconstruction method.

5.2 One Level of Decomposition

In signal coding based on the DWT, the input signal is typically decomposed into two components: (i) a low-resolution approximation and (ii) a detail signal. This results in decomposing the input signal into low-pass and high-pass versions, generally referred to as subbands. Each of the resulting subbands can be further decomposed using the same approach. In this manner, the DWT decomposes a given input signal into a number of frequency bands [Mal89]. At the receiver, the signal can be reconstructed using appropriate inverse DWT filters. Figure 5.1 shows one level of a two-band decomposition and reconstruction process. In this chapter the filters in Figure 5.1 are assumed to be linear-phase, FIR filters.

In this section, to convey the basic idea behind the proposed fast reconstruction approach, we focus our attention to a one-level decomposition as in Figure 5.1 where the reconstructed output signal can be written as:

$$y(n) = \sum_{m=-N_h}^{N_h} x'_h(n-m)h(m) + \sum_{m=-N_g}^{N_g} x'_g(n-m)g(m), \tag{5.1}$$

where

$$x'_h(n) = \begin{cases} x_h(\frac{n}{2}) & n \text{ even} \\ 0 & n \text{ odd} \end{cases}, \; X'_h(z) = X_h(z^2),$$

$$x'_g(n) = \begin{cases} x_g(\frac{n}{2}) & n \text{ even} \\ 0 & n \text{ odd} \end{cases}, \; X'_g(z) = X_g(z^2), \tag{5.2}$$

and $2N_h + 1$ and $2N_g + 1$ are the lengths of the h and g filters, respectively.

To simplify the notation, we define $x_h(a) = x_g(a) = 0$ if $a \notin \mathbb{Z}$. So,

$$x'_i(n) = x_i(\frac{n}{2}), \; i = h, g, \text{ and } \; n \in \mathbb{Z}. \tag{5.3}$$

Using the above notation, the output can be written as:

$$y(n) = \sum_{m=-N_h}^{N_h} x_h(\frac{n-m}{2})h(m) + \sum_{m=-N_g}^{N_g} x_g(\frac{n-m}{2})g(m). \tag{5.4}$$

5.2.1 Fast Reconstruction for Progressive Transmission Systems

Now, let us consider a coding system in which the DWT coefficients are progressively encoded in K levels of refinement. At the l^{th} level of refinement, $l = 1, 2, \ldots, K$, the DWT coefficient $x_i(n)$ is quantized and encoded by $b_l^i(n)$ bits/sample, $i = h, g$. The values of $b_l^i(n)$ are defined by a bit allocation algorithm or are otherwise dictated by the encoding algorithm. In general, $b_l^i(n) \leq b_{l+1}^i(n)$ and it is possible – and crucial to note – that we might have $b_l^i(n) = b_{l+1}^i(n)$, for some i and n. That is, at each level of refinement, only **some** DWT coefficients are refined. Let us use $x_i(n, b_1^i), x_i(n, b_2^i), \ldots, x_i(n, b_K^i)$ to represent the decoded versions of the DWT coefficients $x_i(n)$, at different levels of refinement ($x_i(n, b_1^i)$ is the coarsest representation of $x_i(n)$ using $b_1^i(n)$ bits for quantization and $x_i(n, b_K^i)$ is the finest representation using $b_K^i(n)$ bits for quantization). Also, we use $y_l(n)$ to denote the reconstructed signal (output of the reconstruction process) corresponding to the l^{th} level of refinement.

Using the **conventional** method of reconstruction in subband coding systems, for each $l = 1, 2, \ldots, K$, upon obtaining x_h and x_g, the decoder reconstructs the l^{th} refinement of the output, y_l, according to:

$$y_l(n) = \sum_{m=-N_h}^{N_h} x_h(\frac{n-m}{2}, b_l^h)h(m) + \sum_{m=-N_g}^{N_g} x_g(\frac{n-m}{2}, b_l^g)g(m). \tag{5.5}$$

In other words, given that y_l is reconstructed at the decoder, upon receiving additional bits from which $x_h(n, b_{l+1}^h)$ and $x_g(n, b_{l+1}^g)$ – the $(l+1)^{\text{st}}$ refinements of x_h and x_g – are obtained, the inverse DWT filters are used **anew** to compute an updated replica of the signal – the $(l+1)$st refinement of y. Therefore, every time that the output is updated, the entire computational complexity of the reconstruction process is repeated.

In this chapter, we propose an alternative method in which we define $\Delta y_l(n) = y_{l+1}(n) - y_l(n)$ and $\Delta x_i(n, b_l^i) = x_i(n, b_{l+1}^i) - x_i(n, b_l^i)$, $i = h, g$. Clearly, Δy_l, $l = 1, 2, \ldots, K-1$, can be computed as follows:

$$\Delta y_l(n) = \sum_{m=-N_h}^{N_h} \Delta x_h(\frac{n-m}{2}, b_l^h)h(m) + \sum_{m=-N_g}^{N_g} \Delta x_g(\frac{n-m}{2}, b_l^g)g(m). \quad (5.6)$$

Then, to reconstruct a new refinement of the signal, y_{l+1}, it suffices to use

$$y_{l+1}(n) = y_l(n) + \Delta y_l(n). \quad (5.7)$$

Now let us consider the complexity (number of multiplications) of the conventional approach using (5.5) and compare it against the complexity of the proposed fast reconstruction method. If, at each level of refinement l, we use (5.5) to compute y_l, it costs $L = \frac{N}{2}(2N_h + 1) + \frac{N}{2}(2N_g + 1)$ multiplications – a quantity which is independent of the number of refined DWT coefficients. Therefore, for K levels of refinement, the computational complexity is KL multiplications. By contrast, using (5.6) and (5.7) to update the output signal costs $2N_i + 1$ multiplications per each **nonzero** Δx_i, $i = h, g$ (number of nonzero $\Delta x_i \leq \frac{N}{2}$). If $\alpha_{il}\frac{N}{2}$ DWT coefficients (x_i, $i = h, g$) are refined at the l^{th} level of refinement, then using (5.6) and (5.7) to update the output signal, the total number of multiplications would be $\frac{N}{2}[(2N_h + 1)\sum_{l=1}^{K} \alpha_{hl} + (2N_g + 1)\sum_{l=1}^{K} \alpha_{gl}]$. Note that for large values of K, $\sum_{l=1}^{K} \alpha_{il} << K$ for the rate-scalable image cod-

ing systems reported in the literature [Sha93, Sai96, Tau94] (also the system of Chapter 4), and therefore the computational complexity of the proposed approach is much less than that of the conventional approach. For example, in the EZW coding approach of [Sha93], not only are there many "insignificant" coefficients at low bit rates which are not transmitted and are assumed to be zero, but also the "significant" coefficients are refined in dominant and subordinate passes (at each pass some of the significant coefficients remain unchanged). In Section 5.4, we provide two specific examples to demonstrate the amount of complexity reduction obtained by the proposed method.

5.2.2 On-line Updating of the Output

In addition to reducing the reconstruction complexity in a progressive transmission system, using (5.6) and (5.7) to reconstruct the output provides the capability to update the output signal upon receiving the new refinement of **each** DWT coefficient, thus allowing for an **on-line** update of the output. For many signal coding systems the output bit stream is transmitted in the form of packets. Consider a general signal coding system (or one level of refinement of a progressive coding system) and assume that the output is transmitted using M packets. The size of each packet and thus the value of M depends on the chosen protocol. Each packet contains the refinement bits corresponding to one or more DWT coefficients. At the receiver, there are two general alternatives for reconstruction:

- **Alternative 1.** Wait until all M packets are received and then reconstruct the output, or

- **Alternative 2.** Reconstruct M intermediate versions of the output – one for each newly received packet.

If the conventional reconstruction formula, (5.5), is used, Alternative 1 results in a delay and Alternative 2 results in an increase in computational complexity by a factor of M. Using the proposed approach (described by (5.6) and (5.7)) to reconstruct the output makes it possible to provide M intermediate versions of the output without increasing the complexity or delay, i.e., Alternative 2 without the added complexity. This is possible because, in the proposed method, upon receiving a refinement of a given DWT coefficient, say $\Delta x_h(k)$, we can determine exactly which set of output samples need to be updated. Let us refer to this set as $\mathcal{A}_k$. The output samples are obtained by multiplying $\Delta x_h(k)$ by $h(m)$ to update $y(2k+m)$, $2k+m \in \mathcal{A}_k$ (see (5.6)). Taking into account the boundary effects,[1] it is easy to show that $\mathcal{A}_k = \{2k+m : m \in \mathcal{B}_k\}$, where

$$\mathcal{B}_k = \begin{cases} 0, 1, \ldots, N_h + 2k, & 0 \le k \le \frac{N_h}{2} \\ -N_h, -(N_h-1), \ldots, N_h, & \frac{N_h}{2} < k < \frac{N-N_h}{2} \\ -N_h, \ldots, N-2k, & \frac{N-N_h}{2} \le k < \frac{N}{2} \end{cases} . \tag{5.8}$$

The same type of argument holds for updating the output when a refinement $\Delta x_g(k)$ is received.

Also, note that if after receiving a rough reproduction of the signal, the receiver requests an update of **only a portion of the signal** (region of interest), it suffices to send a refinement of the DWT coefficients corresponding to the spatial location of the region of interest. In this case, the computational cost of using (5.6) and (5.7) to update the output is proportional to the number

[1] To have a smooth boundary after reconstruction, the signal is symmetrically extended near the boundaries [Mal89].

of refined coefficients and obviously less than the computational cost of the conventional approach. Such **interactive** coding systems are particularly useful in medical applications in which after receiving a rough version of the image, the user identifies a region of interest (usually small compared to the entire image) and requests additional information only for this region.

The same procedure can be applied to encoding systems which utilize more than one-level of decomposition. The two-dimensional (2-D) extension of the proposed approach is also straightforward for a separable 2-D DWT.

5.3 General Filter Banks

In this section, we extend the idea for complexity reduction developed in the previous section to general filter banks. Figure 5.2 illustrates a general decomposition and reconstruction structure using a set of passband filters called **filter banks**. This structure can be used to perform linear transformations like the discrete cosine transform, lapped orthogonal transform, Laplacian pyramid, Gabor transform, QMF's, and DWT [Vet91]. Although there exist "fast implementations" for many of these transformations, the study of this general structure is worthy of consideration as it unifies the application of our approach for different transformations and provides a general methodology for progressive transmission situations. Furthermore, our approach does not depend on the specific transformation used. In fact, the ratio of the computational complexity of the proposed approach to that of the conventional approach for reconstruction is fixed for any transformation as long as the number of refined coefficients are the same. So, the choice of the transformation structure affects the complexity of both approaches

in the same manner.

To make the argument more precise, first, let us consider the example illustrated in Figure 5.3 consisting of three levels of the two-band structure in Figure 5.1. Later, we generalize our fast reconstruction approach to the filter bank shown in Figure 5.2. The example in Figure 5.3 is equivalent to the example in Figure 5.2 when $k_1 = 2$, $k_2 = 4$, $k_3 = k_4 = 8$ and filters g_i and h_i, $i = 1, 2, 3, 4$, are chosen appropriately. Given filters h and g in Figure 5.3, the corresponding reconstruction filters in Figure 5.2 can be found. The main equation governing the relationship between g_i, $i = 1, 2, 3, 4$ and g and h is the equivalence of the two structures in Figure 5.4 [Vet91]. Using Figure 5.4, one can establish the following equations relating different filters in Figures 5.2 and 5.3:

$$G_1(z) = G(z), \tag{5.9}$$

$$G_2(z) = G(z^2)H(z), \tag{5.10}$$

$$G_3(z) = G(z^4)H(z^2)H(z), \tag{5.11}$$

and

$$G_4(z) = H(z^4)H(z^2)H(z). \tag{5.12}$$

Although the reconstruction operations in Figures 5.2 and 5.3 are the same, the number of multiplications in Figure 5.2 is more than the number of multiplications in Figure 5.3. Specifically the output signal in Figure 5.2 can be written as:

$$y(n) = \sum_{m=-N_1}^{N_1} x_1'(n-m)g_1(m) + \sum_{m=-N_2}^{N_2} x_2'(n-m)g_2(m) + \sum_{m=-N_3}^{N_3} x_3'(n-m)g_3(m) + \sum_{m=-N_4}^{N_4} x_4'(n-m)g_4(m), \tag{5.13}$$

where

$$x_1'(n) = \begin{cases} x_1(\frac{n}{2}) & n \text{ even} \\ 0 & n \text{ odd} \end{cases}, \; X_1'(z) = X_1(z^2),$$

$$x_2'(n) = \begin{cases} x_2(\frac{n}{4}) & n = 4k \\ 0 & \text{otherwise} \end{cases}, \; X_2'(z) = X_2(z^4),$$

$$x_3'(n) = \begin{cases} x_3(\frac{n}{8}) & n = 8k \\ 0 & \text{otherwise} \end{cases}, \; X_3'(z) = X_3(z^8),$$

$$x_4'(n) = \begin{cases} x_4(\frac{n}{8}) & n = 8k \\ 0 & \text{otherwise} \end{cases}, \; X_4'(z) = X_4(z^8), \tag{5.14}$$

$2N_i + 1$ is the length of g_i and $N_1 = N_g$, $N_2 = 2N_g + N_h$, $N_3 = 4N_g + 3N_h$, and $N_4 = 7N_h$. So, using all coefficients to reconstruct the output, we need $L_1 = N[\frac{2N_1+1}{2} + \frac{2N_2+1}{4} + \frac{2N_3+1}{8} + \frac{2N_4+1}{8}]$ multiplications in Figure 5.2 and $L_2 = N[(2N_h + 1)(\frac{1}{8} + \frac{1}{4} + \frac{1}{2}) + (2N_g + 1)(\frac{1}{8} + \frac{1}{4} + \frac{1}{2})]$ multiplications in Figure 5.3. After computing L_1 in terms of N_h and N_g, we can show that

$$L_1 = N[1 + 3N_g + 3N_h] > L_2 = N[\frac{7}{4}(N_h + N_g + 1)], \tag{5.15}$$

establishing that the number of multiplications corresponding to the implementation of Figure 5.3 is less than that of Figure 5.2.

Now, we concentrate on the main issue of this section and show that a fast reconstruction approach similar to what was proposed in Section 5.2 can be used for the filter bank shown in Figure 5.2. To simplify the notation, we define $x_i(a) = 0$ if $a \notin \mathbb{Z}$. So,

$$x_i'(n) = x_i(\frac{n}{2^i}), \; i = 1, 2, 3, \text{ and } x_4'(n) = x_4(\frac{n}{8}). \tag{5.16}$$

Using the above notation, the output can be written as:

$$y(n) = \sum_{m=-N_1}^{N_1} x_1(\frac{n-m}{2})g_1(m) + \sum_{m=-N_2}^{N_2} x_2(\frac{n-m}{4})g_2(m) + \sum_{m=-N_3}^{N_3} x_3(\frac{n-m}{8})g_3(m) + \sum_{m=-N_4}^{N_4} x_4(\frac{n-m}{8})g_4(m). \quad (5.17)$$

In the conventional approach for reconstructing the output of a progressive transmission scheme, the different updates, $y_1, y_2, \ldots, y_K$, are computed after receiving $x_i(n, b_1^i), x_i(n, b_2^i), \ldots, x_i(n, b_K^i)$ as follows:

$$y_l(n) = \sum_{m=-N_1}^{N_1} x_1(\frac{n-m}{2}, b_l^1)g_1(m) + \sum_{m=-N_2}^{N_2} x_2(\frac{n-m}{4}, b_l^2)g_2(m) + \sum_{m=-N_3}^{N_3} x_3(\frac{n-m}{8}, b_l^3)g_3(m) + \sum_{m=-N_4}^{N_4} x_4(\frac{n-m}{8}, b_l^4)g_4(m). \quad (5.18)$$

As before, we define $\Delta y_l(n) = y_{l+1}(n) - y_l(n)$ and $\Delta x_i(n, b_l^i) = x_i(n, b_{l+1}^i) - x_i(n, b_l^i)$, where Δy_l, $l = 1, 2, \ldots, K-1$, is given by:

$$\Delta y_l(n) = \sum_{m=-N_1}^{N_1} \Delta x_1(\frac{n-m}{2}, b_l^1)g_1(m) + \sum_{m=-N_2}^{N_2} \Delta x_2(\frac{n-m}{4}, b_l^2)g_2(m) + \sum_{m=-N_3}^{N_3} \Delta x_3(\frac{n-m}{8}, b_l^3)g_3(m) + \sum_{m=-N_4}^{N_4} \Delta x_4(\frac{n-m}{8}, b_l^4)g_4(m). \quad (5.19)$$

Then, given that $y_l(n)$ is available at the decoder, $y_{l+1}(n)$ can be computed by adding $\Delta y_l(n)$ to $y_l(n)$. Equation (5.19) is a generalization of (5.6) and therefore provides a fast reconstruction scheme as was discussed in Section 5.2. As before, complexity reduction is achieved due to the fact that many Δx_i's are zero.

This approach also provides the capability to update the reconstructed signal upon obtaining the refinement of each coefficient. Table 5.1 shows the required updates for the coefficients that are sufficiently far from the boundaries: For each coefficient $x_i(k)$, the first row contains the filter coefficients needed for updating the output and the second row contains the output samples that are affected by the refinement of $x_i(k)$.

5.4 Examples

Although our scheme works for any rate-scalable transform coding system, we pick two particular coding systems to show the practical usefulness of the proposed fast reconstruction method. We use the EZW approach of [Sha93] and the scalable image coding system of Chapter 4 for this purpose.

5.4.1 Example 1

In [Sha93], an 8×8 DWT decomposed image has been used to present the details of the EZW algorithm. We use the same example (Table 5.2) and assume that refining each coefficient, using a decomposition structure like the 2-D version of the one in Figure 5.2, affects all output pixels in the same row or column (a reasonable assumption on the length of filters for an 8×8 image). The conventional reconstruction approach takes 1,024 multiplications to create each update of the output. Without going through the details of the EZW algorithm, as it is shown in [Sha93], the first dominant pass results in refining coefficients in locations 11, 12, 13, and 54 (the numbers indicate the row and column indices, respectively). Reconstructing the image using the proposed approach takes 160 multiplications instead of 1,024. The first subordinate pass (the second level of refinement) refines the same four coefficients [Sha93] and thus results in the same amount of computational complexity reduction. The second dominant pass (the third level of refinement) changes coefficients in locations 21 and 22. Our approach requires 80 multiplications to reconstruct the third image. The second subordinate pass (the fourth level of refinement) refines the aforementioned six DWT coefficients. To create the fourth image, our approach requires 240 multiplications. There-

fore, to reconstruct the first four images, the fast scheme needs a total of 640 multiplications compared to 4,096 multiplications for the conventional method – a six-fold reduction. Also, note that since we have had 16 coefficient refinements in the first four passes, we could have reconstructed 16 intermediate images using the fast reconstruction method without any additional computational complexity. To construct these 16 images using the conventional approach would cost 16,384 multiplications.

5.4.2 Example 2

As our second example, we consider the image coding system described in Chapter 4. Let us consider the progressive transmission of an image using System B in Chapter 4. For this example, we assume that at each level of refinement, 0.025 bits/pixel (6,554 bits for a 512×512 image) are transmitted. The conventional reconstruction approach takes 5,505,024 multiplications to create each update of the output. Table 5.3 provides the exact number of multiplications needed in the proposed fast reconstruction approach. As it is shown in Table 5.3, the computational complexity of the proposed approach, for this example, is almost one third of that of the conventional approach. Furthermore, note that if we want to packetize the bit stream and transmit it over a network, e.g. an asynchronous transfer mode (ATM) network, we should wait to receive all 6,554 bits before starting to reconstruct each refinement. In ATM, each packet consists of 48 bytes of information. So, we need about $M = 17$ packets to transmit the bit stream corresponding to each level of refinement. Using the proposed method, we could generate 17 intermediate images (one image after receiving each packet). Creating these intermediate images using the conventional approach would increase

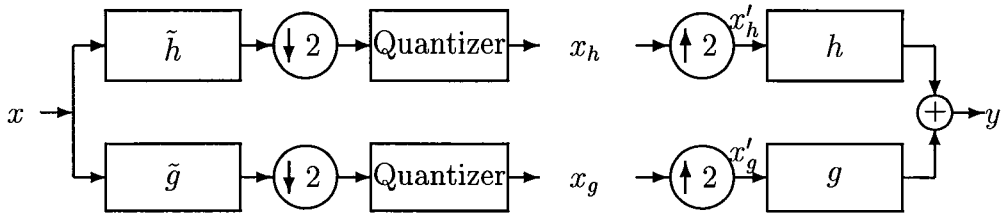


Figure 5.1: Two-Band Subband Analysis and Synthesis Structure.

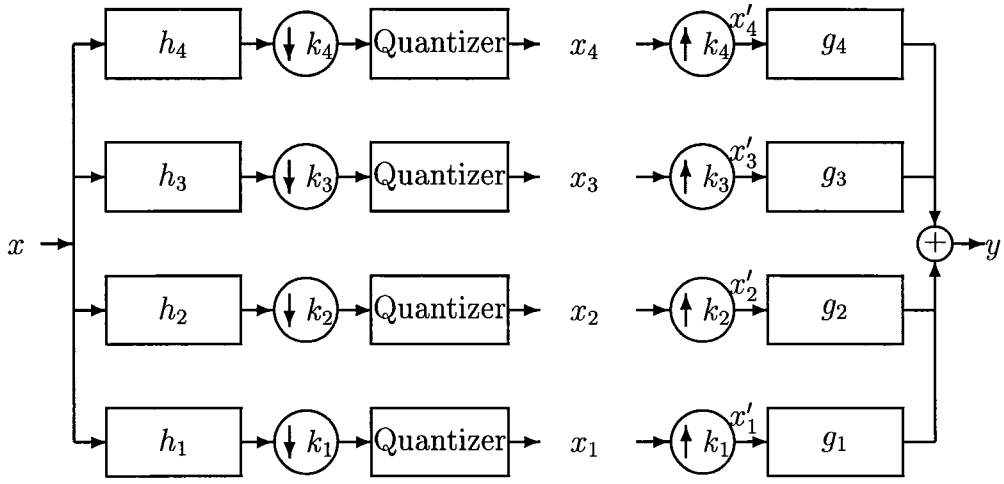


Figure 5.2: Subband Analysis and Synthesis Filter Banks.

the computational complexity by a factor of 17. Avoiding this increase in complexity in the conventional approach is only possible by accepting a delay and waiting to receive all 17 packets.

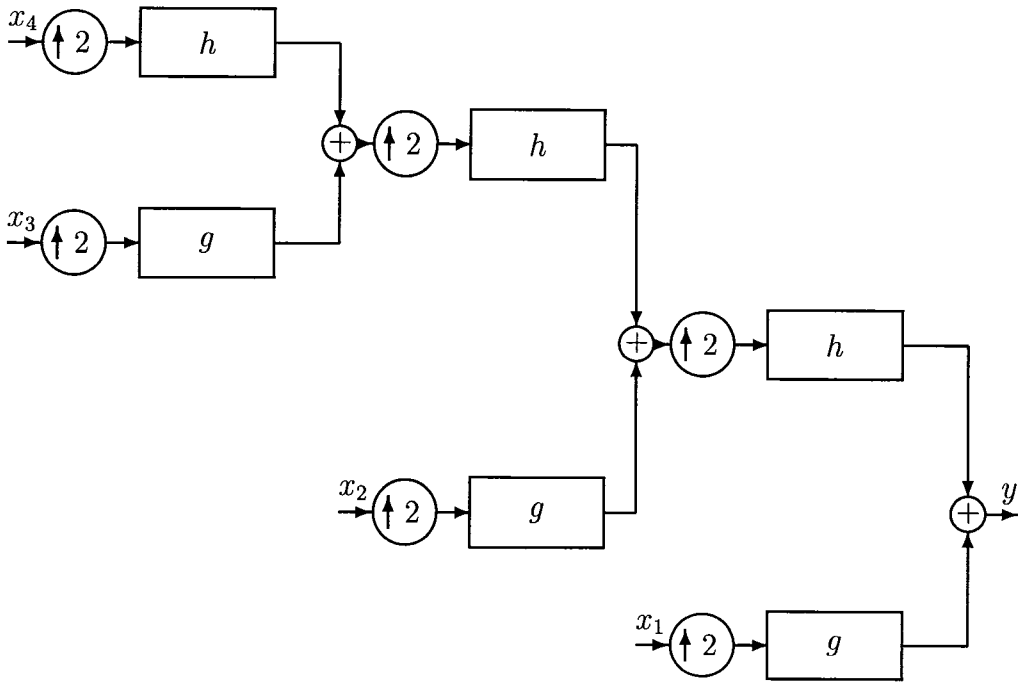


Figure 5.3: Subband Synthesis Filter Bank.

Figure 5.4: Two Equivalent Structures.

Table 5.1: Update Table for Coefficients Sufficiently Far from the Boundaries.

$x_1(k)$	$g_1(-N_1)$	$\cdots$	$g_1(0)$	$\cdots$	$g_1(N_1)$
$\frac{N_1}{2} < k < \frac{N-N_1}{2}$	$y(2k-N_1)$	$\cdots$	$y(2k)$	$\cdots$	$y(2k+N_1)$
$x_2(k)$	$g_2(-N_2)$	$\cdots$	$g_2(0)$	$\cdots$	$g_2(N_2)$
$\frac{N_2}{4} < k < \frac{N-N_2}{4}$	$y(4k-N_2)$	$\cdots$	$y(4k)$	$\cdots$	$y(4k+N_2)$
$x_3(k)$	$g_3(-N_3)$	$\cdots$	$g_3(0)$	$\cdots$	$g_3(N_3)$
$\frac{N_3}{8} < k < \frac{N-N_3}{8}$	$y(8k-N_3)$	$\cdots$	$y(8k)$	$\cdots$	$y(8k+N_3)$
$x_4(k)$	$g_4(-N_4)$	$\cdots$	$g_4(0)$	$\cdots$	$g_4(N_4)$
$\frac{N_4}{8} < k < \frac{N-N_4}{8}$	$y(8k-N_4)$	$\cdots$	$y(8k)$	$\cdots$	$y(8k+N_4)$

Table 5.2: Example of 3-level Wavelet Transform of an Image [Sha93].

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

Table 5.3: Number of Multiplications Required to Build Each Level of Refinement.

Refinement	Our approach	Normal approach	Ratio
1	1,806,336	5,505,024	33%
2	1,855,270	5,505,024	34%
3	1,855,270	5,505,024	34%
4	1,855,270	5,505,024	34%
5	2,051,006	5,505,024	37%
6	2,051,006	5,505,024	37%
7	2,051,006	5,505,024	37%

Chapter 6

Joint Source-Channel Coding for Wireless Channels

6.1 Introduction

In a joint source-channel coding scheme (Figure 6.1), one of the important problems is how to allocate the bit rate (bandwidth) between the source coder and the channel coder. Consider a class of source coders consisting of fixed-length coders operating at different bit rates $r_s \in \mathcal{R}_s$ and a class of channel coders consisting of coders with rates $r_c \in \mathcal{R}_c$. Then, the overall transmission rate for a particular combination of source and channel coders is $r = r_s/r_c$. The number of possible (r_s, r_c) pairs is $|\mathcal{R}_s| \times |\mathcal{R}_c|$ where $|\mathcal{R}|$ is the cardinality of the set $\mathcal{R}$. For a channel with maximum transmission rate r_t, any pair which satisfies the condition $r = r_s/r_c \leq r_t$ can be used. So, a valid and important question is which pair provides the best performance. Tanabe and Farvardin use a packetization scheme to prevent infinite error propagation in a variable-length entropy-coded

quantization scheme [Tan92]. They propose an optimal algorithm to assign bits between the source and channel encoders of different subbands of an image for a memoryless binary symmetric channel (BSC).

In this chapter, we propose an optimal rate allocation policy which minimizes the end-to-end average squared-error distortion. First, we consider a BSC and then we extend our approach to wireless channels with slow fading. The wireless channel can be modeled as a finite-state channel which is a varying BSC with crossover probabilities determined by a multi-state Markov process [Wan95b]. The organization of the remainder of the chapter is as follows: In Section 6.2, we express the total distortion as the sum of the source coder distortion and the channel distortion. Then, we propose a model to compute the channel distortion and minimize the end-to-end distortion for a BSC. In Section 6.3, an adaptive scheme is developed to minimize the distortion for finite-state channels (FSC's). Section 6.4 provides the simulation results.

6.2 Minimizing End-to-End Distortion for Binary Symmetric Channels

Let us first assume that the channel is a memoryless BSC. Later we will generalize the solution to the case of finite-state channels. Our approach is to express the expected total average distortion D_{tot}, as the sum of two terms: (i) the source coder distortion D_s and (ii) the channel distortion D_c which is caused by channel errors. With reference to Figure 6.1, we can write the total average distortion as

$$D_{tot} = E\|X - Y_J\|^2 = E\|X - Y_I\|^2 + 2E[(X - Y_I)^t(Y_I - Y_J)] + E\|Y_I - Y_J\|^2. \quad (6.1)$$

Assuming the channel errors are independent from the source, and the source encoder is designed to meet the centroid condition [Far90], we conclude that

$$E[(X - Y_I)^t(Y_I - Y_J)] = E\{E[(X - Y_I)^t|I]E[(Y_I - Y_J)|I]\} = 0. \tag{6.2}$$

Therefore,

$$D_{tot} = D_s(r_s) + D_c(r_c), \tag{6.3}$$

where $D_s(r_s) \triangleq E\|X - Y_I\|^2$ and $D_c(r_c) \triangleq E\|Y_I - Y_J\|^2$.

Now, the problem can be formulated as the following minimization problem:

$$\min_{r_s/r_c \leq r_t} D_{tot}. \tag{6.4}$$

To compute D_{tot}, we need to compute $D_s(r_s)$ and $D_c(r_c)$. Here, $D_s(r_s)$ is the distortion-rate performance of the source coder and is known or precomputed for the chosen class of source encoders. To compute $D_c(r_c)$, we need to model the combination of the channel encoder, channel, and channel decoder, hereafter referred to as the equivalent channel. For the class of algebraic codes, since each error only affects one block of data, the equivalent channel can be modeled as a discrete memoryless channel. For the more interesting case of the class of convolutional codes, the decoding errors are known to occur in bursts, and therefore the equivalent channel has memory. A modified Gilbert noise channel [Gil60] is used to model the equivalent channel. For each convolutional code in the given class of channel codes, the parameters of the model (b, g, p_b, and p_g in Figure 6.2) are chosen so as to match the characteristics of the equivalent channel as measured by the actual simulation of a convolutional encoder-decoder and a memoryless BSC.

The model parameters are extracted in the following fashion. The combination of a convolutional encoder, a memoryless BSC, and a convolutional decoder

yields a channel of bursty nature. We identify as error burst any cluster of consecutive bits with a high (typically higher than 0.5) probability of bit error. In order to extract the start and end points of each burst we morphologically process the bit error sequence obtained by simulation. The morphological filtering consists of a dilation operation followed by an erosion operation, which connect any two error bits that are no more than d positions apart. We have determined experimentally that the choice of $d = 4$ gives an intuitively satisfying identification of bursts. Moreover, the clusters identified as burst will not change unless d is increased significantly. Then, we compute the average burst length l_b taking into account all bursts of length greater than or equal to two. The rest are considered random errors and are used to determine p_g. The model parameters b, g, and p_b are computed from the total number of bursts n_b, the average burst length l_b, and the average bit error rate p_e of the simulated bit-error sequence. Direct calculations yield

$$\begin{aligned} b &= 1 - \frac{1}{l_b}, \\ g &= 1 - \frac{(1-b)q_b}{1-q_b}, \\ p_g &= \frac{n_g}{q_g N}, \\ p_b &= \frac{p_e - q_g p_g}{q_b}, \end{aligned} \tag{6.5}$$

where n_g is the number of random errors, N is the total number of bits in good state, and $q_b = (1-g)/(2-b-g)$ and $q_g = (1-b)/(2-b-g)$ are the steady state probabilities of the "bad" and "good" states, respectively. Assuming a uniformly distributed binary sequence as the input of the convolutional encoder, we compute the model parameters (b, g, p_b, and p_g) using the aforementioned

procedure. The reason for assuming a uniform input is that we want to model the equivalent channel independently of the statistics of the source encoder output.

After computing the parameters of the equivalent channel model, we can calculate the channel distortion by

$$D_c(r_c) = E\|Y_I - Y_J\|^2 = E\{E[\|Y_I - Y_J\|^2|I]\} = \sum_i \sum_j \|y_i - y_j\|^2 P_{eq}(j|i)P(i), \tag{6.6}$$

where $i, j \in \{0,1\}^K$, K is the length of the source encoder codewords, and $P_{eq}(j|i)$ denotes the equivalent channel transition probability which can be computed using the parameters of the model in Figure 6.2. Let s be the integer with binary representation $s_0 s_1 \ldots s_{K-1}$, where s_k is determined by the equivalent channel state during the transmission of the k^{th} bit of a source encoder codeword, i.e. $s_k = 0$ if the channel state is good and $s_k = 1$ if the channel state is bad. Let us define π_{s_0} as the probability of starting in state s_0 and $\pi_{s_k s_{k-1}}$ as the probability of transition from state s_{k-1} to s_k, $\pi_{s_k s_{k-1}} \in \{b, 1-b, g, 1-g\}$. Then,

$$P_{eq}(j|i) = \sum_{s=0}^{2^K-1} (\pi_{s_0} \prod_{k=1}^{K-1} \pi_{s_k s_{k-1}}) p_g^{W[(j\oplus i)\wedge \overline{s}]} (1-p_g)^{W[\overline{(j\oplus i)}\wedge \overline{s}]} p_b^{W[(j\oplus i)\wedge s]} (1-p_b)^{W[\overline{(j\oplus i)}\wedge s]}, \tag{6.7}$$

where $\oplus$, $\wedge$, $\overline{(\cdot)}$, and $W(\cdot)$ denote modulo 2 bitwise addition, bitwise AND operation, binary compliment, and the Hamming weight function, respectively.

We compute $P_{eq}(j|i)$ for each convolutional code and each value of r_c using (6.7) and the parameters of the model. To compute $D_c(r_c)$, using (6.6), we need to know the source encoder codebook, the convolutional code, and the BSC probability of error (ϵ) (no knowledge about the source is needed). Having $D_c(r_c)$ and $D_s(r_s)$, minimizing D_{tot} in (6.4) is straightforward.

The procedure described above yields the optimal allocation of the available transmission rate between the source and channel codes for a memoryless BSC. In next section, we solve the same problem for an FSC.

6.3 Optimal Rate Allocation for a Finite-State Channel

To minimize the end-to-end distortion for an FSC, we assume that the receiver and transmitter (through a feedback channel) have access to the channel state information (CSI). Repeating the procedure of Section 6.2 for each of the constituent channels of the FSC, leads to an adaptive joint source-channel coding scheme which assigns bit rates to the source and channel coders according to the current state of the channel, while maintaining the overall transmission rate fixed. Note that recent developments in convolutional coding theory, allow for fast and easily implementable switching between convolutional codes of different rates [Hag88, Hag90, Cox91]. So, based on the results achieved by the model, we pick the best (r_s, r_c) pair for each memoryless BSC and save the results in a table. This procedure is done off-line and takes no time during the transmission. Then, at each time instant, the best (r_s, r_c) pair is chosen from the corresponding table, based on the CSI, and is used to quantize and transmit the source. Whenever the state of the channel is changed, the source encoder and the channel encoder are adapted correspondingly. The advantage of the adaptive rate allocation procedure described above over a fixed rate allocation scheme will be discussed in Section 6.4. We consider FSC's representing Rayleigh and log-normal fading channels arising in practical mobile communication scenarios.

6.3.1 Selection of the FSC

In this section, we describe the procedure used for modeling a slow fading channel by an FSC. In this approach, we divide the range of received SNR's into a finite

number of intervals, each representing one state of the FSC. Figure 6.3 illustrates a log-normal p.d.f. which is a good model for the received SNR of a slow fading channel [Rap96]. In this figure, the average received SNR is 10 dB. The vertical lines in Figure 6.3 represent the thresholds which divide the range of received SNR's into four intervals. For an uncoded binary phase-shift keying (BPSK) signaling system, the bit error rate (BER) is given by

$$\text{BER} = \frac{1}{2}\text{erfc}(\sqrt{\gamma_b}), \tag{6.8}$$

where γ_b is the received SNR per bit and erfc is the error function. For this signaling system, we assume that each interval, representing a state of the FSC, is modeled by a BSC. For a K-state model, let us denote the $K-1$ SNR thresholds by T_k, $k = 1, \ldots, K-1$, and assume $T_0 = 0$ and $T_K = \infty$. Then, the BER, say ϵ_k, associated with the BSC of state k is taken as the average error rate of the BPSK system given that $\gamma_b \in [T_{k-1}, T_k)$, i.e.,

$$\epsilon_k = \frac{\int_{T_{k-1}}^{T_k} \frac{1}{2}\text{erfc}(\sqrt{x})p(x)dx}{P_k}, \quad k = 1, \ldots, K, \tag{6.9}$$

where P_k, the probability of being at state k, is given by

$$P_k = \int_{T_{k-1}}^{T_k} p(x)dx, \tag{6.10}$$

and $p(.)$ is a log-normal p.d.f.

To select the SNR thresholds, we adopt an ad-hoc approach. This approach places a constraint on the thresholds by restricting the resulting BER values to belong to the set $\mathcal{A} = \{0, 0.01, 0.03, 0.05, 0.1, 0.15\}$. This reduces the number of distinct BER's and therefore, the number of required simulations. The thresholds are initially placed so as to get K equiprobable intervals, $P_k = \frac{1}{K}$, $\forall k$. If $\epsilon_1 \notin \mathcal{A}$, T_1 is adjusted until $\epsilon_1 \in \mathcal{A}$. Changing T_1 will affect ϵ_2. If the new value of ϵ_2

does not belong to $\mathcal{A}$, T_2 is adjusted until $\epsilon_2 \in \mathcal{A}$, and so on. Note that in this process, some of the intervals may be merged in order to achieve the constraint on the BER values. Therefore, the final number of states might be less than K.

The resulting FSC models for log-normal fading channels with average received SNR's equal to 10 and 15 dB are tabulated in Table 6.1.

6.4 Simulation Results

In this section, we consider the class of TSVQ's [Ger92] as the source coder and the class of rate-compatible punctured convolutional (RCPC) codes [Hag88] as the channel coder. Although our scheme works for any class of source and channel coders, we choose TSVQ and RCPC codes in this section due to the fact that changing their rates is very simple. So, not only does this example show the usefulness of our proposed rate allocation scheme, but also it provides a practical encoding system which is easily implementable.

First, let us investigate the accuracy of the proposed equivalent channel model in estimating $D_c(r_c)$ using (6.6). We use the RCPC code in Table I of [Hag88] and a 10-stage TSVQ with dimension 4 designed for a zero-mean, unit-variance memoryless Gaussian source. For each stage of the TSVQ, the end-to-end distortion of the best (r_s, r_c) pair for a memoryless BSC with $\epsilon = 0.05$ is shown in Figure 6.4. The channel transmission rate $r_t = 2.5$ bits per sample. As it is clear from Figure 6.4, the simulation- and model-based results coincide for the first 5 stages. In other cases, the estimated distortion is not identical to the distortion computed by simulation; however, the minimum distortion point of the two sets of results coincide and therefore the model is adequate for choosing the optimal

(r_s, r_c) pair. Similar results are obtained for $\epsilon = 0.01,\ 0.03,\ 0.1,$ and 0.15 and for other TSVQ's and transmission rates. In all cases investigated, the model selects the optimal (r_s, r_c) pair.

Table 6.1 shows the different FSC models representing log-normal fading channels. The FSC's for the Rayleigh channel were very similar to those for the log-normal channels and therefore we chose to present results for the latter case only. This is also the most relevant case for the problem at hand since Rayleigh fading is usually faster than log-normal and does not lend itself easily to adaptive transmission. For each case in Table 6.1, we investigate the performance of the system for memoryless Gaussian sources as well as Gauss-Markov sources with correlation coefficient $\rho = 0.9$. We consider different dimensions for the TSVQ and different transmission rates for the channel as summarized in Tables 6.2-6.5. In obtaining the results for System I, the performance of the system corresponding to each (r_s, r_c) pair is obtained by simulation for each state of the channel and the system resulting in minimum distortion is identified by considering all possible cases. System II characterizes the system in which the optimal (r_s, r_c) pair is obtained from our model for each state of the channel. The results reported in these tables for System II are also SNR simulation results. In all investigated cases, the model picks the best (r_s, r_c) pair and therefore provides the same performance as the optimal performance (System I). System III reports the best simulation results attainable using a non-adaptive rate allocation. Using adaptation provides up to 1.9 dB for the memoryless Gaussian source and up to 4.7 dB for the Gauss-Markov source. A higher transmission rate results in a higher "adaptation gain." The adaptation gain increases as the SNR of the channel decreases. So, adaptation is more useful for channels with higher band-

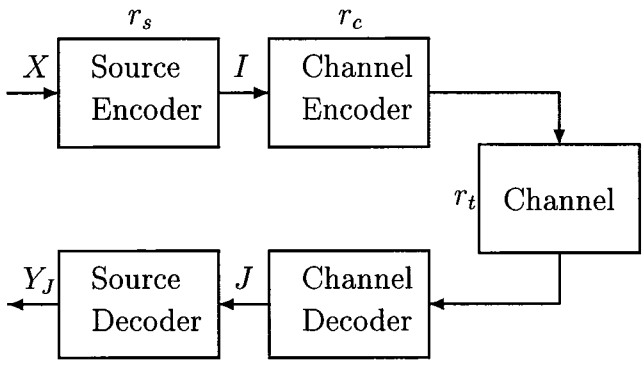


Figure 6.1: System Block Diagram.

width and lower SNR. We also report theoretical bounds on the performance of a system with no complexity or delay constraints in Tables 6.2-6.5.

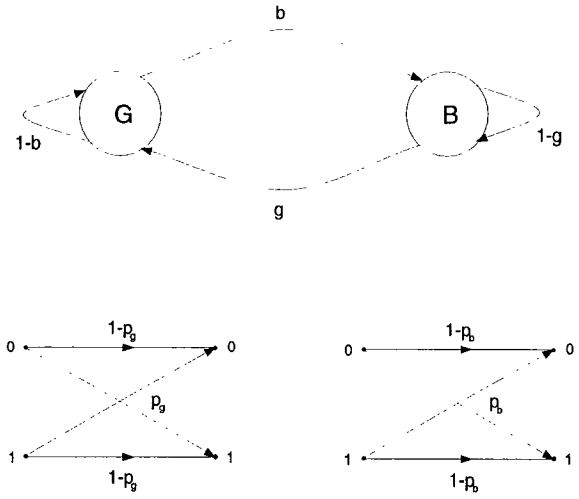


Figure 6.2: Gilbert Channel Model.

Table 6.1: FSC Models Representing Log-Normal Fading Channels.

Ref.	Fading Channel	State Prob.	State BER
1	SNR=10 dB, 2 states	0.238	0.1
		0.762	0.0
2	SNR=10 dB, 3 states	0.133	0.15
		0.129	0.03
		0.738	0.00
3	SNR=15 dB, 2 states	0.237	0.03
		0.763	0.00
4	SNR=15 dB, 3 states	0.064	0.10
		0.072	0.01
		0.864	0.00

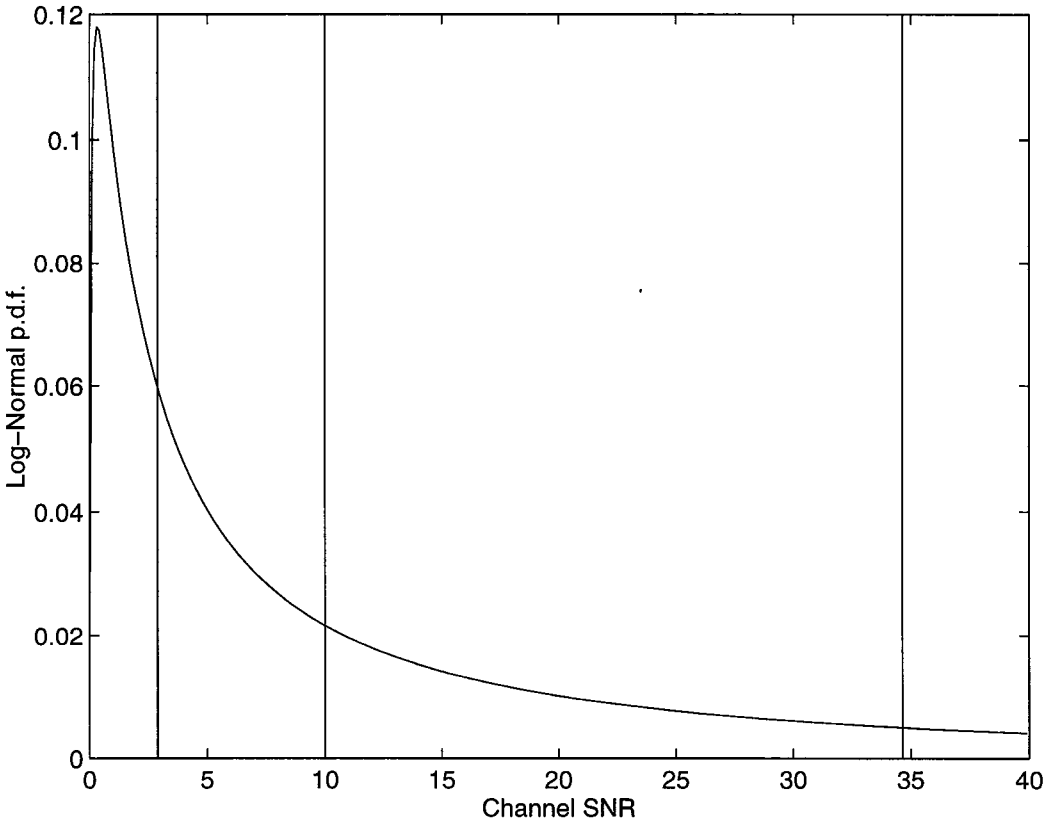


Figure 6.3: Log-Normal Distribution.

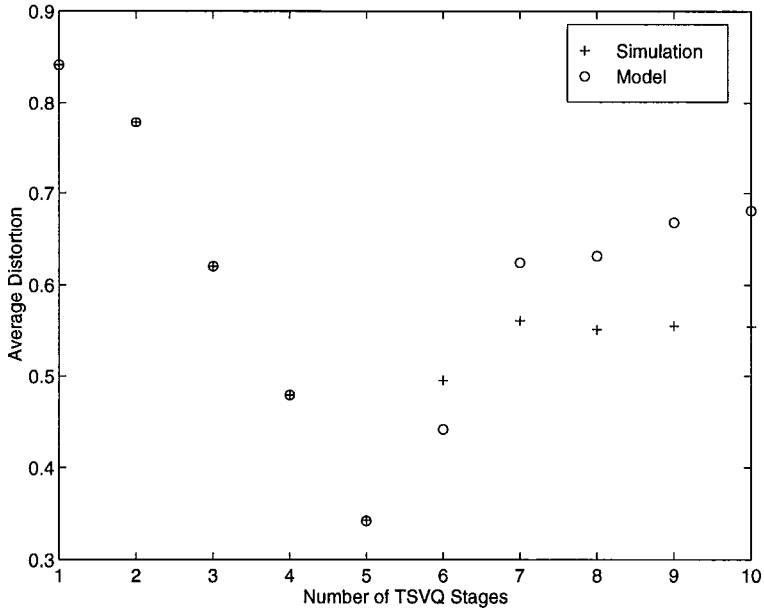


Figure 6.4: End-to-End Distortion.

Table 6.2: Simulation Results (SNR in dB) for Channel 1 in Table 6.1.

Memoryless Gaussian Source						
Dim.	Rate	I	II	III	$\sum_i p_i D(RC_i)$	$D(RC)$
8	1.25	3.81	3.81	3.22	6.39	6.69
4	2.00	6.03	6.03	5.11	9.90	10.70
4	2.50	7.49	7.49	5.60	12.10	13.37
Gauss-Markov Source ($\rho = 0.9$)						
Dim.	Rate	I	II	III	$\sum_i p_i D(RC_i)$	$D(RC)$
8	1.25	9.65	9.65	6.59	13.57	13.90
4	2.00	11.13	11.13	7.63	17.12	17.91
4	2.50	12.80	12.80	8.34	19.32	20.58

Table 6.3: Simulation Results (SNR in dB) for Channel 2 in Table 6.1.

Memoryless Gaussian Source						
Dim.	Rate	I	II	III	$\sum_i p_i D(RC_i)$	$D(RC)$
8	1.25	3.85	3.85	3.34	6.38	6.72
4	2.00	6.13	6.13	5.24	9.79	10.76
4	2.50	7.41	7.41	5.78	11.85	13.45
Gauss-Markov Source ($\rho = 0.9$)						
Dim.	Rate	I	II	III	$\sum_i p_i D(RC_i)$	$D(RC)$
8	1.25	9.56	9.56	6.62	13.33	13.94
4	2.00	11.32	11.32	7.22	17.00	17.98
4	2.50	12.30	12.30	7.63	19.06	20.67

Table 6.4: Simulation Results (SNR in dB) for Channel 3 in Table 6.1.

Memoryless Gaussian Source						
Dim.	Rate	I	II	III	$\sum_i p_i D(RC_i)$	$D(RC)$
8	1.25	4.34	4.34	4.34	7.13	7.18
4	2.00	7.38	7.38	7.31	11.36	11.49
4	2.50	9.04	9.04	8.52	14.16	14.36
Gauss-Markov Source ($\rho = 0.9$)						
Dim.	Rate	I	II	III	$\sum_i p_i D(RC_i)$	$D(RC)$
8	1.25	10.90	10.90	9.56	14.34	14.39
4	2.00	12.74	12.74	10.95	18.57	18.70
4	2.50	14.72	14.72	11.50	21.37	21.57

Table 6.5: Simulation Results (SNR in dB) for Channel 4 in Table 6.1.

Memoryless Gaussian Source						
Dim.	Rate	I	II	III	$\sum_i p_i D(RC_i)$	$D(RC)$
8	1.25	4.65	4.65	4.45	7.15	7.26
4	2.00	7.88	7.88	7.48	11.29	11.61
4	2.50	9.79	9.79	8.80	13.96	14.51
Gauss-Markov Source ($\rho = 0.9$)						
Dim.	Rate	I	II	III	$\sum_i p_i D(RC_i)$	$D(RC)$
8	1.25	11.09	11.09	9.65	14.35	14.46
4	2.00	13.23	13.23	11.00	18.50	18.82
4	2.50	15.14	15.14	11.70	21.18	21.72

Chapter 7

Channel–Matched Hierarchical Table–Lookup Vector Quantization for Finite–State Channels

7.1 Introduction

Source coding applications which involve transmission over noisy channels, have been the main motivation for studying the sensitivity of the VQ to channel noise. These studies have led to the development of techniques for making a VQ robust with respect to channel noise, either by an appropriate binary codeword assignment, [Far90, Zeg90] or by a complete redesign of the VQ partition and codebook, resulting in the so-called channel-optimized vector quantizer (COVQ) [Kum84, Far91].

In this chapter, we consider an encoding system which borrows ideas from channel-optimized vector quantization and hierarchical table-lookup vector quantization. Specifically, we study the performance of the HTVQ over noisy channels and propose a channel-matched HTVQ (CM-HTVQ) which simultaneously provides a low-complexity encoding (table-lookup encoding and decoding) and robustness against channel noise. For this scheme, we present performance results for a memoryless Gaussian source, a Gauss-Markov source, and still images over a memoryless binary symmetric channel. Further, we develop adaptive extensions of the CM-HTVQ for wireless communication applications where the transmission medium is modeled by a finite-state channel.

Section 7.2 provides the design procedure for the CM-HTVQ for a discrete memoryless channel and presents simulation results. In Section 7.3, two adaptive versions of the CM-HTVQ system are developed for finite-state channels. In the first version, it is assumed that both the decoder and encoder have access to the channel state; the second version assumes that the channel state is available only at the decoder. Section 7.4 studies the effects of channel mismatch on the performance of CM-HTVQ's. In Section 7.5, a comparison between the proposed CM-HTVQ's and a jointly-optimized tandem source-channel coding scheme is presented. Section 7.6 focuses on the applications of lookup tables for transcoding. In this section, an example motivated by multicasting applications over a heterogeneous network is used to illustrate the use of lookup tables for transcoding.

7.2 Table-Lookup Vector Quantization for a Discrete Memoryless Channel

7.2.1 Channel-Optimized VQ

The basic idea behind channel-optimized vector quantization is to incorporate the effect of transmission noise in the design of the quantizer. In these schemes the encoder and decoder are designed so that the end-to-end average distortion after encoding, transmission over the channel, and decoding is minimized [Kum84, Far91]. The optimization is done for a given source, a given discrete memoryless noisy channel, a fixed dimension m, and a fixed codebook size M.

Let us denote the i^{th} encoding cell and codevector by S_i and $\mathbf{c}_i$, respectively. Then, it is straightforward to show that for the squared-error distortion measure, the design algorithm for the channel-optimized vector quantizer (COVQ) is a modified generalized Lloyd algorithm [Lin80], with the following expressions for the optimal partition, $\mathcal{P} = \{S_0, S_1, \ldots, S_{M-1}\}$, and the optimal codebook $\mathcal{C} = \{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{M-1}\}$, [Far91]:

$$S_i = \{\mathbf{x} : \sum_{j=0}^{M-1} P(j|i) \parallel \mathbf{x} - \mathbf{c}_j \parallel^2 \leq \sum_{j=0}^{M-1} P(j|l) \parallel \mathbf{x} - \mathbf{c}_j \parallel^2, \ \forall l\}, \tag{7.1}$$

and

$$\mathbf{c}_j = \frac{\sum_{i=0}^{M-1} P(j|i) \int_{S_i} \mathbf{x} p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}}{\sum_{i=0}^{M-1} P(j|i) \int_{S_i} p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}}, \tag{7.2}$$

where $P(j|i)$ denotes the probability that j is received given that i is transmitted, $p_{\mathbf{x}}(\mathbf{x})$ is the m-fold probability density function (p.d.f.) of the source, and $i, j \in \{0, 1, \cdots, M-1\}$. The design algorithm is based on an iterative application of Equations (7.1) and (7.2).

When applied to the noisy channel it is designed for, a COVQ performs better than a VQ (designed for a noiseless channel) [Far91]. The performance improvement depends on the encoding rate, the level of noise in the channel, and the statistics of the source. We will have more to say on this later in this section.

Like an ordinary VQ, a COVQ suffers from a high encoding complexity. The encoding complexity (number of additions and multiplications) of a COVQ is proportional to the number of non-empty encoding cells [Far91]. We now proceed to describe a COVQ-type scheme in which the hierarchical table-lookup structure of HTVQ is used to reduce the encoding complexity. First, we consider the case of a discrete memoryless channel with known transition probabilities.

7.2.2 Channel-Matched HTVQ

To design an N-stage HTVQ for a noisy channel, only the lookup table of the last (N^{th}) stage and the codebook at the decoder need to be modified. The role of the first $N-1$ stages of an HTVQ is to provide the best pair of addresses (a good approximation of the input vector) to the N^{th} stage and therefore the design procedure for these stages remains unchanged. The last stage, which describes the actual encoder partition, must be adapted to the characteristics of the channel. We therefore propose an HTVQ-like quantizer in which the first $N-1$ stages are exactly the same as those in the HTVQ described in [Cha85], and the last stage is based on a COVQ with partition $\mathcal{P}^* = \{S_0^*, S_1^*, \ldots, S_{M-1}^*\}$ and codebook $\mathcal{C}^* = \{\mathbf{c}_0^*, \mathbf{c}_1^*, \ldots, \mathbf{c}_{M-1}^*\}$. Each pair of indices at the output of the $(N-1)^{st}$ stage, say (i_1, i_2), corresponds to an m-dimensional vector $\mathbf{x}_{N-1}(i_1, i_2)$ – an approximation to the m-dimensional vector $\mathbf{x}$ at the input of the first stage.

The lookup table of the N^{th} stage (Table N) is constructed in such a way that each possible pair of indices (i_1, i_2) at the output of the $(N-1)^{st}$ stage (input of Table N), corresponds to an index i at the output of the N^{th} stage such that $\mathbf{x}_{N-1}(i_1, i_2) \in S_i^*$, i.e.,

$$\sum_{j=0}^{M-1} P(j|i) \parallel \mathbf{x}_{N-1}(i_1, i_2) - \mathbf{c}_j^* \parallel^2 \leq \sum_{j=0}^{M-1} P(j|l) \parallel \mathbf{x}_{N-1}(i_1, i_2) - \mathbf{c}_j^* \parallel^2, \ \forall l. \quad (7.3)$$

We refer to this structure as the Channel-Matched HTVQ (CM-HTVQ). Note that the resulting encoder is not optimal since the design of the intermediate stages, even for a noiseless channel, is not optimal [Cha85] – hence the use of the term "channel-matched" instead of "channel-optimized."

7.2.3 Results and Comparisons

We now present simulation results on the performance of VQ, COVQ, HTVQ, and CM-HTVQ over a memoryless BSC with probability of error ϵ ($P(j|i) = \epsilon^{h(i,j)}(1-\epsilon)^{log_2 M - h(i,j)}$, where $h(i,j)$ = Hamming distance between i and j).

Tables 7.1 and 7.2 summarize the SNR performance of VQ, COVQ, HTVQ, and CM-HTVQ for a memoryless Gaussian source and a Gauss-Markov source, respectively. Both sources have zero mean and unit variance; the Gauss-Markov source has a correlation coefficient $\rho = 0.9$. Likewise, the PSNR performance results for the 512×512 monochrome Lenna are tabulated in Table 7.3. The Gaussian and Gauss-Markov sources consist of 1,200,000 samples from a memoryless Gaussian and a Gauss-Markov source quantized using an 8-bit Lloyd-Max scalar quantizer, respectively. The quantizers for still images are designed using a training sequence which consists of five images from the USC database (Cou-

ple, Crowd, Man, Woman1, and Woman2) and the PSNR results are obtained by averaging the mean squared error over 10 runs of channel simulation. Simulated annealing is used to assign binary codewords [Far90] to the codevectors of VQ and HTVQ in all cases. This provides some degree of robustness against channel noise [Far90], thus making the comparison between different schemes meaningful. For visual examination, a graphical presentation of the results in Table 7.3 at rate 1 bit/sample is provided in Figure 7.1.

It is clear from the tables that, a CM-HTVQ – like a COVQ – exhibits some robustness against transmission noise (compared with HTVQ and VQ). A very interesting observation is that the difference between the performance of COVQ and CM-HTVQ is less than the gap between VQ and HTVQ for a noiseless channel ($\epsilon = 0.0$). The gap shrinks as the channel becomes noisier. This fact can be seen more easily in Figure 7.1. Notice that for large values of ϵ, the COVQ and CM-HTVQ have practically the same performance. Thus, for a memoryless BSC, not only does CM-HTVQ provide a low-complexity table-lookup encoding, but also it achieves almost the same performance as COVQ. An intuitive explanation for this phenomenon follows.

Let us first isolate what is responsible for the loss in performance of an HTVQ compared to a VQ of the same rate and dimension. To this end, let us divide the process of HTVQ encoding into two steps: The quantization of the first $N-1$ stages and that of the last stage. The first step provides an approximation of the input vector to the last stage; the second step quantizes this approximate vector. Note that the HTVQ codebook is identical to the codebook of the VQ used for the design of the last stage. Consequently, the only difference between an HTVQ and a full-searched VQ (which is also used for the last stage of the

HTVQ) is the encoder partition. If the HTVQ encoder could implement an optimal nearest neighbor rule, it would give rise to the same partition as that of the VQ encoder and, as a result, a performance identical to that of the VQ. Unfortunately, the HTVQ encoder implements a nearest neighbor rule on the output of the $(N-1)^{st}$ stage, not on the input vector, and therefore its encoding cells are not optimal (they only approximate the encoding cells of a full-searched VQ). This means that, due to this approximation error, some input vectors are suboptimally encoded – hence, the suboptimal performance of the HTVQ.

It is shown in [Far91] that the number of encoding regions in a COVQ is less than that of a VQ operating at the same rate (some of the COVQ encoding regions are empty). Since there are fewer encoding regions in a COVQ compared with a VQ, in general, the COVQ encoding regions are larger and, therefore, the effect of the approximation error of encoding cells is less noticeable in a CM-HTVQ. The noisier is the channel, the smaller will be the number of COVQ encoding regions and the less pronounced will be the suboptimality of CM-HTVQ, an observation which is corroborated by our simulation results in Tables 7.1, 7.2, and 7.3.

7.3 VQ Design for Finite-State Channels

In this section, we extend the COVQ and CM-HTVQ results to FSC's. We consider FSC's in which there is a memoryless BSC associated with each state and a Markov chain governs the transition between the states [Gil60, Ell63, Mus89]. Such an FSC is an appropriate model for the channel in many wireless communication scenarios [Wan95a].

In what follows, we propose two adaptive schemes to encode a source for transmission over an FSC: (i) A system in which the channel state information (CSI) is available at the decoder and encoder and (ii) a system in which the CSI is only available at the decoder.

7.3.1 CSI Available at the Decoder and Encoder

Here, we assume that the state of the FSC is available both in the receiver and transmitter. Since in practice the state of the channel is estimated in the receiver (based on the received signal power), this assumption is hinged on the availability of a feedback channel which can be used to send the estimated channel state back to the transmitter. In addition, this assumption is only realistic in situations where the variations in the fade depth are much slower than the round trip transmission time.

Assuming a K-state FSC, our adaptive coding system consists of a bank of K encoder-decoder pairs, one for each channel state. The encoder-decoder pairs may be COVQ's (full-searched encoding) or CM-HTVQ's (table-lookup encoding). In either case, for each state, the COVQ or CM-HTVQ encoder-decoder pair is designed based on the BER of the BSC associated with that state. To encode each input vector, the appropriate encoder is used based on the available CSI. Since the receiver has access to the CSI, the received codeword is decoded using the corresponding decoder. For ease of referencing, we call this encoding scheme a **fully-adaptive** (adaptive at both ends) system. We use the acronym FA-COVQ for a fully-adaptive system in which a COVQ is the basic encoding unit. Similarly, we use the acronym FA-CM-HTVQ for a fully-adaptive system in which a CM-HTVQ is the basic encoding unit. A useful set of

simulation results for these fully-adaptive coding systems along with appropriate comparisons with other schemes are presented in Section 7.3.3.

7.3.2 CSI Available Only at the Decoder

In this section, we maintain the assumption that the channel is an FSC but focus on the case that the CSI is only available at the decoder. This corresponds to the case where the feedback channel is not available or when the variations of the fade depth are too fast. The resulting coding system has a fixed encoder and an adaptive decoder. We refer to such a scheme as a **decoder-adaptive** system.

In what follows, we consider a decoder-adaptive coding system in which the basic coding unit is a COVQ (DA-COVQ). This is a generalization of the scalar case studied by Wang and Moayeri [Wan90]. Then, we extend these results to a CM-HTVQ based decoder-adaptive system (DA-CM-HTVQ).

A. DA-COVQ

Let us consider an m-dimensional VQ with a codebook size M and an FSC with K states, where the probability of being in state k is denoted by P_k, $k = 1, \cdots, K$. We assume the CSI is only available at the decoder and during the transmission of a codeword the state does not change. Using a single encoder and K sets of codevectors at the decoder, the end-to-end average squared-error distortion, D, can be expressed as

$$D = \sum_{i,j,k} P(j|i,k) P_k \int_{S_i} p_{\mathbf{X}}(\mathbf{x}) \parallel \mathbf{x} - \mathbf{c}_{j,k} \parallel^2 d\mathbf{x}, \tag{7.4}$$

where $P(j|i,k)$, $i, j \in \{0, 1, \cdots, M-1\}$, denotes the probability that j is received given that i is transmitted and the channel is in state k, $p_{\mathbf{X}}(\mathbf{x})$ is the m-fold p.d.f.

of the source, S_i is the encoding region corresponding to codeword i, and $\mathbf{c}_{j,k}$ is the reconstruction vector associated with the received index j when the channel is in state k. The fixed encoder and the K decoders can be designed using a generalized Lloyd algorithm consisting of two steps:

- **Computation of the generalized centroid**

  For each received index, j, and CSI, k, we wish to choose the best codevector, $\mathbf{c}_{j,k}$, when the encoding regions, S_i's, are fixed. It can be shown that the optimal codevectors are given by

$$\mathbf{c}_{j,k} = \frac{\sum_i P(j|i,k) \int_{S_i} \mathbf{x} p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}}{\sum_i P(j|i,k) \int_{S_i} p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}}. \tag{7.5}$$

- **Computation of the generalized nearest neighbor**

  For each input vector, $\mathbf{x}$, we wish to select the codeword i that minimizes the distortion for a fixed set of codevectors, $\mathbf{c}_{j,k}$'s. It can be shown that the optimal encoding region corresponding to codeword i is given by

$$S_i = \{\mathbf{x} : \sum_{j,k} P(j|i,k) P_k \parallel \mathbf{x} - \mathbf{c}_{j,k} \parallel^2 \leq \sum_{j,k} P(j|l,k) P_k \parallel \mathbf{x} - \mathbf{c}_{j,k} \parallel^2; \forall l \neq i\}. \tag{7.6}$$

Iterating between the above two steps results in reducing the end-to-end average distortion. The iteration is terminated when the relative change in D in two successive steps of iteration is less than a predefined small number.

Encoding an m-dimensional vector, $\mathbf{x}$, corresponds to finding the codeword i such that $\mathbf{x} \in S_i$. To simplify the process of encoding, let us define $S_i = \cap_{l \neq i} S_{il}$, where

$$\begin{aligned} S_{il} = \{\mathbf{x} : 2 \textstyle\sum_{j,k} [P(j|i,k) - P(j|l,k)] P_k < \mathbf{x}, \mathbf{c}_{j,k} > \geq \\ \textstyle\sum_{j,k} [P(j|i,k) - P(j|l,k)] P_k \parallel \mathbf{c}_{j,k} \parallel^2\}. \end{aligned} \tag{7.7}$$

Also, define

$$\mathbf{c}_i = \sum_{j,k} P(j|i,k) P_k \mathbf{c}_{j,k}, \tag{7.8}$$

and

$$\alpha_i = \sum_{j,k} P(j|i,k) P_k \parallel \mathbf{c}_{j,k} \parallel^2 . \tag{7.9}$$

Then, it can be shown that

$$S_{il} = \{\mathbf{x} : 2 < \mathbf{x}, \mathbf{c}_i - \mathbf{c}_l > \geq \alpha_i - \alpha_l\}. \tag{7.10}$$

Therefore, by precomputing $\mathbf{c}_i$ and α_i, we can render the encoding process identical to that of a COVQ. There are MK codevectors, $\mathbf{c}_{j,k}$'s, at the decoder. To reduce the memory requirements at the decoder, we can use the following formulas to compute the $\mathbf{c}_{j,k}$'s instead of storing them. First, define

$$\mathbf{C}_i = \frac{\int_{S_i} \mathbf{x} p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}}{\int_{S_i} p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}}, \tag{7.11}$$

and

$$p_i = \int_{S_i} p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \tag{7.12}$$

Then, assuming that $\mathbf{C}_i$'s and p_i's are precomputed and stored at the decoder, we can compute $\mathbf{c}_{j,k}$ using the following formula:

$$\mathbf{c}_{j,k} = \frac{\sum_i P(j|i,k) p_i \mathbf{C}_i}{\sum_i P(j|i,k) p_i}. \tag{7.13}$$

B. DA-CM-HTVQ

It is straightforward to extend the above design procedure to an encoder which uses only lookup tables. Such a system consists of N tables at the encoder (one table per stage) and K decoding tables at the decoder (one per channel state). The design of the first $N-1$ stages of an N-stage DA-CM-HTVQ encoder is the same as that of an N-stage HTVQ. The lookup table of the last stage is based on the DA-COVQ proposed before.

7.3.3 Simulation Results

In this section, we investigate the performance of different coding systems when used over an FSC. To study the relative merits of using the CSI in the design of the coding system, we consider four different schemes: (i) a fully-adaptive system which uses the CSI to adapt both the decoder and encoder, (ii) a decoder-adaptive system which uses the CSI only to adapt the decoder, (iii) a non-adaptive system in which the channel is assumed to be a binary symmetric channel with a BER equal to the statistical average of the BER's of the constituent BSC's of the FSC, and (iv) a non-adaptive system designed for a noiseless channel (simulated annealing used for binary codeword assignment). These four systems are illustrated in Figure 7.2. For the first three schemes, we consider both the COVQ and CM-HTVQ as the basic coding unit. The COVQ and CM-HTVQ versions of the third scheme are referred to as NA-COVQ and NA-CM-HTVQ, respectively. Finally, for the forth scheme designed based on a noiseless channel, we consider both the VQ and HTVQ versions.

We present simulation results for the FSC's summarized in Table 6.1 and for each of the above-mentioned systems. Simulation results for a zero-mean, unit-variance memoryless Gaussian source (8-bit Lloyd-Max scalar quantized), a zero-mean, unit-variance Gauss-Markov source with correlation coefficient $\rho = 0.9$, (8-bit Lloyd-Max scalar quantized), and the 512×512 Lenna are reported in Tables 7.4, 7.5, and 7.6, respectively. A few important observations are in order.

As expected, the performance of the decoder-adaptive system is in between those of fully-adaptive and non-adaptive systems. In all cases, channel-matched quantizers outperform quantizers designed for a noiseless channel by a large margin. As an example, consider the PSNR results for coding Lenna at rate 1

bit/pixel over FSC 1 (Table 7.6). There is a 7 dB difference between the VQ designed based on a noiseless channel and the FA-COVQ system. Of this 7 dB, about 5.4 dB is picked up by the NA-COVQ system and an additional 0.6 dB by the DA-COVQ. Neither of these systems require a feedback channel. Also, the systems based on the CM-HTVQ result in a performance loss in the range 0.05-0.96 dB compared to their COVQ-based counterparts; the gap is smaller for higher bit rates and noisier channels. Finally, we observe that increasing the number of states in the FSC model results in improving the performance, albeit at the cost of an increase in memory requirement at the decoder in the decoder-adaptive systems and at the encoder and decoder in the fully-adaptive systems.

7.4 Channel Mismatch Issues

Since the design of CM-HTVQ and its adaptive extensions requires the knowledge of the exact statistics of the channel, an important question that arises is the extent of sensitivity of the CM-HTVQ performance to channel mismatch. Additionally, in many real-world applications, the exact value of the channel parameter(s), (e.g., BER) is not known. Rather, it is known to belong to a certain range. A related question in such situations is the appropriate value of the channel parameter for which the encoding system should be designed. In this section, we attempt to shed some light on these issues.

For a Gauss-Markov source, Table 7.7 summarizes $SNR(\epsilon_d, \epsilon_a)$, the SNR (in dB) performance of a CM-HTVQ designed for a BSC with BER ϵ_d, but applied to a BSC with BER ϵ_a. The mismatch loss is $SNR(\epsilon_a, \epsilon_a) - SNR(\epsilon_d, \epsilon_a)$. The

mismatch loss for different combinations of ϵ_d and ϵ_a can be inferred from the figures in Table 7.7. It is evident from the results in this table that for a given $|\epsilon_d - \epsilon_a|$, the mismatch loss is larger when $\epsilon_d < \epsilon_a$. Therefore, in situations where the channel BER is not known precisely but is known to be in a certain range, a more robust CM-HTVQ is obtained if the higher BER in the range is used for the design.

For finite-state channels, it is much more difficult to assess the sensitivity of the coding system to channel mismatch, simply because there are many more parameters which could lead to mismatch ($2K$ parameters for a K-state FSC: K state BER's and K state probabilities). One possibility would be to assess the loss when the average received SNR of the fading channel is different from the average SNR of the channel for which the encoding system was designed. Then, the mismatch can be described in terms of one parameter only: The average received SNR. This problem is tightly connected to the method used for deriving the FSC parameters from a given fading channel and is beyond the scope of this dissertation.

On the other hand, one important issue for coding systems operating over fading channels is the "transient" behavior of the system when the channel goes into a deep fade and stays there for awhile. In other words, the performance of the encoding system designed for a given FSC but applied to a BSC is also of interest. Furthermore, if the performance of such a system is computed for different BSC's, the performance over any FSC can be computed by calculating the average performance of the constituent BSC's.

Figure 7.3 provides the simulation results for FA-, DA-, and NA-CM-HTVQ's designed for Channel 4 in Table 6.1 but applied to a BSC with different BER's.

As can be seen from Figure 7.3, compared with the NA-CM-HTVQ, the DA-CM-HTVQ is more robust to channel mismatch when the error rate of the BSC is far from the average error rate of the FSC used for the design. This is due to the fact that the decoder-adaptive system has some flexibility at the decoder and when the channel goes into a deep fade, equivalent to high BER's, it can use the codebook used for the highest error rate in the FSC model. The NA-CM-HTVQ does not have such a flexibility.

For brevity, we limited our discussions and results to CM-HTVQ based systems. Similar behavior and trends are expected for COVQ based systems.

7.5 Channel-Matched Quantization vs. Joint Source-Channel Coding

In this section, for the FSC's of Table 6.1, we provide comparisons between the performance of the FA-COVQ and NA-COVQ and their counterparts designed based on a jointly-optimized tandem source-channel coding proposed in Chapter 6. The purpose of these comparisons is not to make conclusive statements or definitive recommendations for one approach versus the other. Rather, they are intended to provide a feel for the relative merits of adaptation in the two different approaches to source-channel coding.

The specific tandem system used for our comparisons in this section consists of a class of VQ's (with encoding rates $\mathcal{R}_c = \{0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0\}$) as the source encoder and an RCPC code [Hag88] as the channel coder. The RCPC code is based on the codes tabulated in Table I of [Hag88]. This system slightly differs from the one presented in Chapter 6 in that it uses VQ's, instead

of TSVQ's, for source coding. This makes the comparisons more meaningful as both the tandem system and the COVQ system are based on full-searched vector quantization. For completeness, we will also report results for the CM-HTVQ systems.

Simulation results for a zero-mean, unit-variance memoryless Gaussian source as well as a zero-mean, unit-variance Gauss-Markov source with correlation coefficient $\rho = 0.9$ are tabulated in Table 7.8. These results are for the different FSC's in Table 6.1 at $r_t = 2$ bits/sample. Three different systems are considered: (i) the jointly-optimized tandem source-channel code (referred to as Tandem), (ii) the COVQ, and (iii) the CM-HTVQ. In all three cases, we have considered both the fully-adaptive and non-adaptive versions. The results for the COVQ and CM-HTVQ are borrowed from Tables 7.4 and 7.5 and repeated here to facilitate comparison.

The results for the fully-adaptive tandem system are obtained by choosing the best (r_s, r_c) pair for each state of the FSC. These results should be compared against FA-COVQ and FA-CM-HTVQ. The results for the non-adaptive tandem system correspond to the case where a fixed (r_s, r_c) pair, resulting in the best performance, is employed. These results should be compared against those of the NA-COVQ and NA-CM-HTVQ systems.

First, in all cases considered, the COVQ outperforms the tandem system. Second, the gap between the performance of FA-COVQ and the adaptive tandem system is less than the gap between their non-adaptive counterparts. This observation is another indication of the relative robustness of COVQ. The CM-HTVQ performance, which is close to that of the corresponding COVQ, is, in all but two cases, superior to the performance of the tandem system. Notice

that the CM-HTVQ system is simpler than the tandem system as it has a simpler quantization operation (table-lookup search vs. full-search) and no explicit channel encoding/decoding. Furthermore, in the case the CSI is only available at the decoder, we may use the decoder-adaptive versions of the COVQ and CM-HTVQ, to obtain an additional modest improvement over the non-adaptive versions. For the tandem source-channel code, we do not know of a decoder-adaptive extension.

7.6 Use of Lookup Tables for Transcoding

Until now we focused attention on the use of lookup tables for simplifying the encoding operation in vector quantization. In this section, we discuss another application of lookup tables in transcoding from a high bit rate code to equal or lower bit rate codes or from a code designed for a given channel to one for another channel with the same or higher level of noise. This can be accomplished simply by adding relatively small lookup tables to the node responsible for performing transcoding.

One important application in which table-lookup transcoding might prove useful is multicasting in a "heterogeneous" network consisting of channels with different throughputs and different levels and types of transmission error. To describe this application, we consider the simple, but instructive, example provided in Figure 7.4, illustrating a network in which the source output is encoded at 2 bits/sample and transmitted over a noiseless channel to Node 1. At Node 1, the received bit stream must be sent to Nodes 2 and 3. The channel between Nodes 1 and 2 (Link 1-2) is a memoryless BSC with $\epsilon = 0.1$ operating at a through-

put of 2 bits/source sample; the channel between Nodes 1 and 3 (Link 1-3) is a log-normal fading channel, modeled by a 2-state FSC (Channel 1 in Table 6.1), with a throughput of 1 bit/source sample. Of course, at Node 1, one could decode the received bit stream and re-encode it using two different encoders, one suitably designed for Link 1-2 and one for Link 1-3. This is certainly undesirable because the decoding and re-encoding at each node amounts to added hardware (therefore cost) and delay. In what follows, we show that lookup tables similar to what was used for HTVQ-type encoders considered in the previous sections offer a powerful, elegant, and simple alternative for transcoding.

To describe table-lookup transcoding, assume that the source is quantized using a 4-dimensional, 2 bits/sample VQ. Therefore, for each 4-dimensional source vector, an 8-bit index is received at Node 1. At Node 1, the received bit stream is transcoded using two different lookup tables – one for Link 1-2 and another for Link 1-3. We refer to these tables as "transcoding tables."

Let us first focus on the design of the transcoding table for Link 1-2. A 4-dimensional, 2 bits/sample COVQ designed for a memoryless BSC with $\epsilon = 0.1$ is used to construct the transcoding table for Link 1-2. The table is constructed so that a received index (8-bit word), say i, describing an approximation $\hat{\mathbf{x}}$ to the source vector, is mapped to a new index, say j, if $\hat{\mathbf{x}}$ is in the COVQ's j^{th} encoding cell. Needless to say, the COVQ codebook must be used at Node 2 for decoding. Note that in this example, the role of the transcoding table for Link 1-2 is to tailor the received bit stream for robust transmission over Link 1-2 and **not** to achieve rate reduction. This table has 8 input bits, describing i, and 8 output bits, describing j, thus a total of 256 bytes!

Link 1-3 is modeled by a 2-state FSC described by Channel 1 in Table 6.1.

For transcoding the bit stream received at Node 1 over this link, we consider two possibilities: (i) The CSI is available at the decoder and encoder (Nodes 3 and 1), leading to a fully-adaptive encoding system, and (ii) the CSI is available only at the decoder (Node 3), leading to a decoder-adaptive system. Of course, one could consider a non-adaptive system as well, but this is uninteresting because its design would be similar to what we described for Link 1-2. For the fully-adaptive case, two 8-dimensional COVQ's, each with rate 1 bit/sample, designed for the two constituent BSC's of the FSC are used to construct two transcoding tables, one for each state of the FSC. Each of these tables, takes two 8-bit indices, corresponding to an approximation of the 8-dimensional source vector, and produces an 8-bit index, specifying the encoding cell in which this approximation falls. Therefore, in this case, the size of each transcoding table is 64 Kbytes. At the decoder there are two codebooks, one associated with each COVQ. In a similar fashion, for the decoder-adaptive case, an 8-dimensional, 1 bit/sample DA-COVQ designed for the given FSC is used to construct the transcoding table. In this case, the size of the transcoding table is 64 Kbytes. Again, in this case, there are two codebooks at the decoder, one associated with each state of the channel. Note that the transcoding operation for Link 1-3 achieves two purposes: First, it reduces the bit rate by a factor of two (two 8-bit words get mapped to one 8-bit word); second, it tailors the bit stream received at Node 1 for transmission over a noisy channel.

In what was described above, all VQ's and COVQ's can be replaced by HTVQ's and CM-HTVQ's, leading to a system which uses lookup tables both for encoding and for transcoding.

Table 7.9 summarizes our simulation results in terms of PSNR at each node

for multicasting the 512×512 Lenna image. Let us be more specific as to what the columns in this table correspond to. For the first two columns of results (labeled VQ and COVQ), it is assumed that a 2 bits/pixel VQ is used to encode and transmit the source to Node 1. The PSNR result at Node 2 in the VQ column corresponds to the case where the received sequence at Node 1 is directly delivered to Node 2 (therefore the 20 dB drop in PSNR compared to Node 1). Likewise, the PSNR result at Node 3 in the VQ column corresponds to decoding the received sequence and re-encoding it at 1 bit/pixel (assuming a noiseless channel). The PSNR result for Node 2 in the COVQ column corresponds to the case where a transcoding table as described above is used at Node 1 to tailor the received sequence to the characteristics of Link 1-2. The improvement in performance (compared with a simple re-directing of the bit stream to Node 2) achieved with a 256-byte transcoding table is remarkable. For Node 3, there are two results under the COVQ column, corresponding to the fully-adaptive and decoder-adaptive cases. In both cases, transcoding tables as described above are used at Node 1.

The next two columns in Table 7.9, labeled HTVQ and CM-HTVQ, pertain to cases where an HTVQ is used instead of a VQ to encode and transmit the source to Node 1. Everything else is the same as before. Figures 7.5-7.9 show the corresponding reconstructed images at Nodes 2 and 3.

For a moment consider a hypothetical situation where the source is directly connected to Node 2 through a BSC like the one in Link 1-2. The CM-HTVQ designed for such a hypothetical case, would be **different** from the system we have considered for encoding the source to Node 2 (an HTVQ followed by a transcoding table at Node 1). However, our simulation results (not included in

Table 7.1: Simulation Results (SNR in dB) for BSC's; Memoryless Gaussian Source.

Rate=0.5 bits/sample, Dimension=16				
ϵ	VQ	COVQ	HTVQ	CM-HTVQ
0.00	2.33	2.33	1.18	1.18
0.01	1.91	2.02	0.93	1.04
0.03	1.25	1.67	0.49	0.88
0.05	0.74	1.43	0.15	0.77
0.10	-0.12	1.00	-0.45	0.55
Rate=1 bit/sample, Dimension=8				
ϵ	VQ	COVQ	HTVQ	CM-HTVQ
0.00	4.85	4.85	4.05	4.05
0.01	3.82	4.21	3.22	3.57
0.03	2.42	3.59	2.02	3.08
0.05	1.45	3.14	1.19	2.72
0.10	0.00	2.26	-0.14	1.98

Table 7.9) indicate that the performance loss compared to such a hypothetical CM-HTVQ is only 0.14 dB. Unlike this case, the fully-adaptive and decoder-adaptive CM-HTVQ systems which use transcoding tables to send the source to Node 3 are exactly the same as the FA-CM-HTVQ and DA-CM-HTVQ designed for the FSC of Link 1-3.

Table 7.2: Simulation Results (SNR in dB) for BSC's; Gauss-Markov Source, $\rho = 0.9$.

Rate=0.5 bits/sample, Dimension=16				
ϵ	VQ	COVQ	HTVQ	CM-HTVQ
0.00	8.57	8.57	7.92	7.92
0.01	6.85	7.52	6.44	7.04
0.03	4.80	6.64	4.49	6.30
0.05	3.39	5.90	3.20	5.63
0.10	1.33	4.53	1.19	4.42
Rate=1 bit/sample, Dimension=8				
ϵ	VQ	COVQ	HTVQ	CM-HTVQ
0.00	11.36	11.36	10.73	10.73
0.01	8.57	9.68	8.17	9.31
0.03	5.62	8.41	5.50	8.13
0.05	3.93	7.50	3.83	7.29
0.10	1.53	5.83	1.49	5.76

Table 7.3: Simulation Results (PSNR in dB) for BSC's; 512×512 Lenna.

Rate=0.5 bits/pixel, Dimension=4×4				
ϵ	VQ	COVQ	HTVQ	CM-HTVQ
0.00	30.49	30.49	29.69	29.69
0.01	24.05	28.24	23.79	27.79
0.03	20.02	26.40	19.94	26.35
0.05	17.99	25.22	18.01	25.13
0.10	15.27	23.02	15.26	22.99
Rate=1 bit/pixel, Dimension=4×2				
ϵ	VQ	COVQ	HTVQ	CM-HTVQ
0.00	32.47	32.47	31.79	31.79
0.01	24.13	29.48	23.97	29.20
0.03	19.90	27.46	19.88	27.32
0.05	17.89	26.09	17.83	26.00
0.10	15.12	23.40	15.06	23.35

Table 7.4: Simulation Results (SNR in dB) for FSC's in Table 6.1; Memoryless Gaussian Source.

			COVQ			VQ	CM-HTVQ			HTVQ
FSC	Rate	Dim.	FA	DA	NA		FA	DA	NA	
1	0.5	16	1.97	1.87	1.82	1.61	1.02	0.96	0.96	0.73
1	1	8	4.08	3.88	3.82	3.11	3.46	3.31	3.27	2.62
1	2	4	7.71	7.33	7.23	5.42	7.56	7.20	7.10	5.33
2	0.5	16	1.99	1.91	1.84	1.64	1.03	0.98	0.97	0.75
2	1	8	4.09	3.92	3.84	3.21	3.47	3.34	3.28	2.71
2	2	4	7.76	7.38	7.21	5.56	7.61	7.25	7.09	5.47
3	0.5	16	2.16	2.12	2.10	2.04	1.11	1.08	1.08	1.01
3	1	8	4.51	4.40	4.38	4.13	3.80	3.72	3.70	3.47
3	2	4	9.07	8.77	8.75	7.87	8.85	8.58	8.56	7.70
4	0.5	16	2.21	2.18	2.15	2.10	1.13	1.11	1.10	1.04
4	1	8	4.58	4.50	4.44	4.25	3.85	3.79	3.75	3.57
4	2	4	9.17	8.89	8.78	8.05	8.95	8.69	8.58	7.88

Table 7.5: Simulation Results (SNR in dB) for FSC's in Table 6.1; Gauss-Markov Source, $\rho = 0.9$.

			COVQ			VQ	CM-HTVQ			HTVQ
FSC	Rate	Dim.	FA	DA	NA		FA	DA	NA	
1	0.5	16	7.22	6.78	6.71	5.53	6.78	6.41	6.36	5.25
1	1	8	9.30	8.54	8.41	6.57	8.94	8.26	8.16	6.36
1	2	4	11.73	10.37	10.15	7.52	11.60	10.28	10.07	7.47
2	0.5	16	7.22	6.74	6.68	5.54	6.79	6.37	6.33	5.26
2	1	8	9.22	8.49	8.28	6.57	8.87	8.21	8.05	6.37
2	2	4	11.61	10.25	9.92	7.58	11.48	10.16	9.84	7.53
3	0.5	16	8.02	7.73	7.70	7.28	7.47	7.24	7.19	6.81
3	1	8	10.42	9.99	9.98	9.23	9.94	9.56	9.56	8.86
3	2	4	13.78	12.82	12.82	11.33	13.56	12.63	12.63	11.19
4	0.5	16	8.09	7.81	7.80	7.34	7.54	7.29	7.29	6.87
4	1	8	10.60	10.06	9.98	9.33	10.08	9.60	9.55	8.93
4	2	4	13.98	12.81	12.72	11.35	13.75	12.63	12.55	11.22

Table 7.6: Simulation Results (PSNR in dB) for FSC's in Table 6.1; 512×512 Lenna.

			COVQ			VQ	CM-HTVQ			HTVQ
FSC	Rate	Dim.	FA	DA	NA		FA	DA	NA	
1	0.5	4×4	27.24	26.36	25.79	21.09	26.89	26.11	25.60	21.05
1	1	4×2	28.09	27.17	26.58	21.12	27.87	27.01	26.44	21.04
1	2	2×2	29.62	28.22	27.59	21.16	29.48	28.15	27.52	21.13
2	0.5	4×4	27.07	26.20	25.55	21.19	26.73	25.97	25.35	21.15
2	1	4×2	27.90	26.90	26.27	21.21	27.67	26.75	26.13	21.14
2	2	2×2	29.12	27.83	26.86	21.32	28.99	27.76	26.82	21.28
3	0.5	4×4	29.13	28.62	28.50	25.09	28.56	28.13	28.03	24.88
3	1	4×2	30.64	29.80	29.71	25.45	30.21	29.45	29.41	25.32
3	2	2×2	32.94	31.89	31.75	25.78	32.63	31.68	31.54	25.74
4	0.5	4×4	29.22	28.45	28.11	25.23	28.63	27.96	27.64	25.04
4	1	4×2	30.59	29.60	29.36	25.63	30.16	29.25	29.03	25.45
4	2	2×2	32.81	31.43	30.97	26.03	32.50	31.23	30.80	25.95

Table 7.7: Performance (SNR in dB) of CM-HTVQ in the Presence of Channel Mismatch; Gauss-Markov Source, $\rho = 0.9$.

Rate=0.5 bits/sample, Dimension=16					
Actual ϵ	Design ϵ (ϵ_d)				
(ϵ_a)	0.00	0.01	0.03	0.05	0.1
0.00	7.92	7.88	7.51	7.26	6.72
0.01	6.44	7.04	7.09	6.94	6.50
0.03	4.52	5.75	6.30	6.29	6.03
0.05	3.24	4.74	5.56	5.65	5.58
0.10	1.28	2.98	4.01	4.25	4.42

Rate=1 bit/sample, Dimension=8					
Actual ϵ	Design ϵ (ϵ_d)				
(ϵ_a)	0.00	0.01	0.03	0.05	0.1
0.00	10.73	10.45	9.86	9.44	8.40
0.01	8.18	9.30	9.27	9.01	8.16
0.03	5.45	7.61	8.16	8.15	7.65
0.05	3.82	6.32	7.12	7.30	7.14
0.10	1.48	4.08	5.04	5.38	5.75

Table 7.8: Performance (SNR in dB) of Tandem Source–Channel Coding, COVQ, and CM-HTVQ for FSC's of Table 6.1; Dimension=4; $r_t = 2$ bits/sample.

Memoryless Gaussian Source						
	Fully-Adaptive			Non-Adaptive		
FSC	Tandem	COVQ	CM-HTVQ	Tandem	COVQ	CM-HTVQ
1	7.05	7.71	7.56	5.48	7.23	7.10
2	6.97	7.76	7.61	5.64	7.21	7.09
3	8.27	9.07	8.85	7.93	8.75	8.56
4	8.80	9.17	8.95	8.10	8.78	8.58
Gauss-Markov Source ($\rho = 0.9$)						
	Fully-Adaptive			Non-Adaptive		
FSC	Tandem	COVQ	CM-HTVQ	Tandem	COVQ	CM-HTVQ
1	11.33	11.73	11.60	7.63	10.15	10.07
2	11.72	11.61	11.48	7.51	9.92	9.84
3	13.57	13.78	13.56	11.24	12.82	12.63
4	13.69	13.98	13.75	11.34	12.72	12.55

Table 7.9: Simulation Results (PSNR in dB) for the Multicasting Example of Figure 7.5; 512 × 512 Lenna.

Node	VQ	COVQ		HTVQ	CM-HTVQ	
1	35.72	35.72		35.13	35.13	
2	15.04	24.23		15.04	24.14	
		FA	DA		FA	DA
3	21.11	27.90	27.03	21.01	27.87	27.01

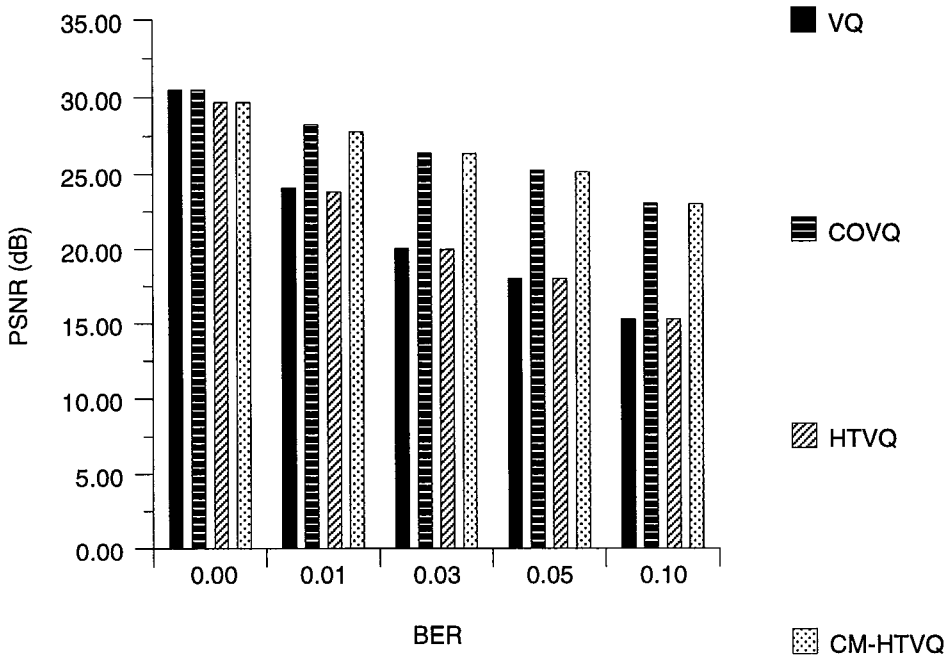


Figure 7.1: Simulation Results for a Binary Symmetric Channel with Different BER's; 512 × 512 Lenna; $r = 0.5$ bits/pixel; Dimension=4 × 4.

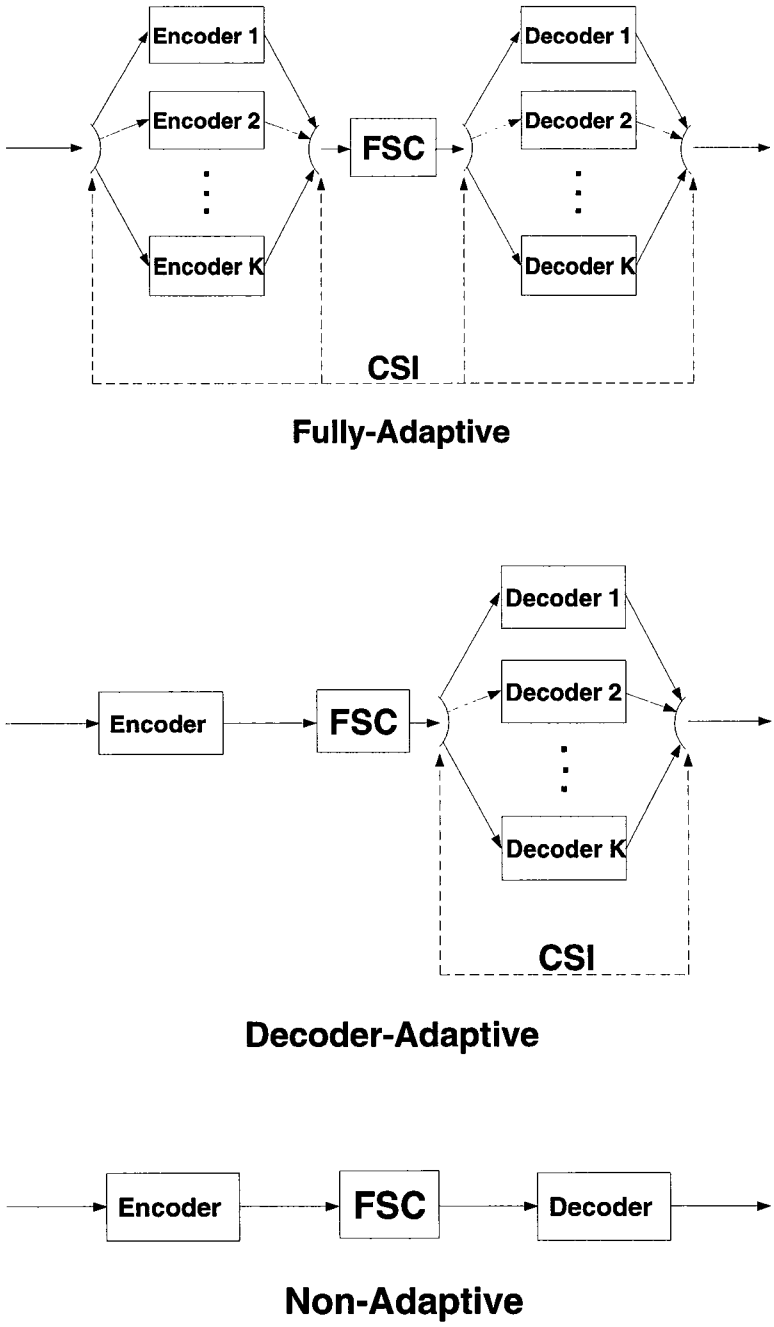


Figure 7.2: Channel–Matched Quantizers for Finite-State Channels.

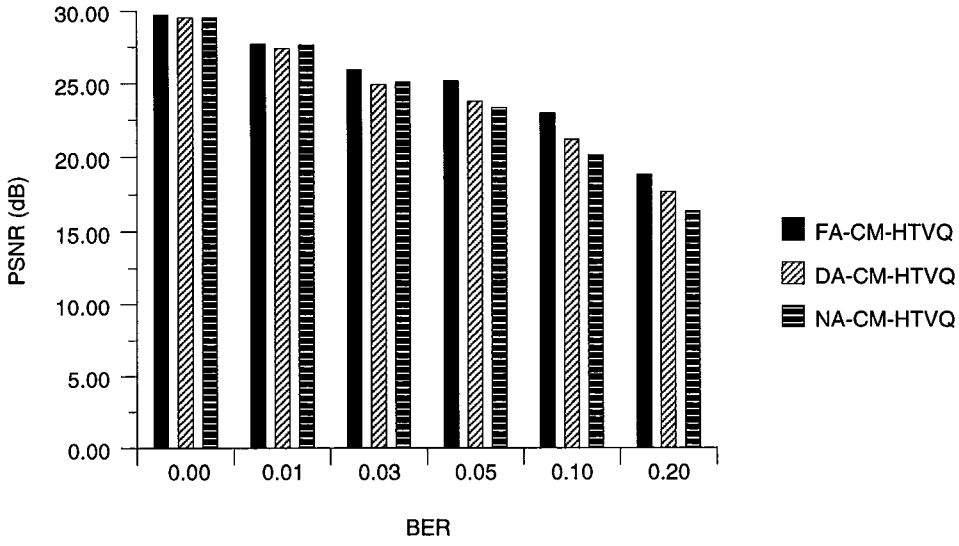


Figure 7.3: PSNR Performance of FA-CM-HTVQ, DA-CM-HTVQ, and NA-CM-HTVQ When Each System Is Designed for Channel 4 of Table 6.1 and Applied to a BSC; 512×512 Lenna; $r = 0.5$ bits/pixel.

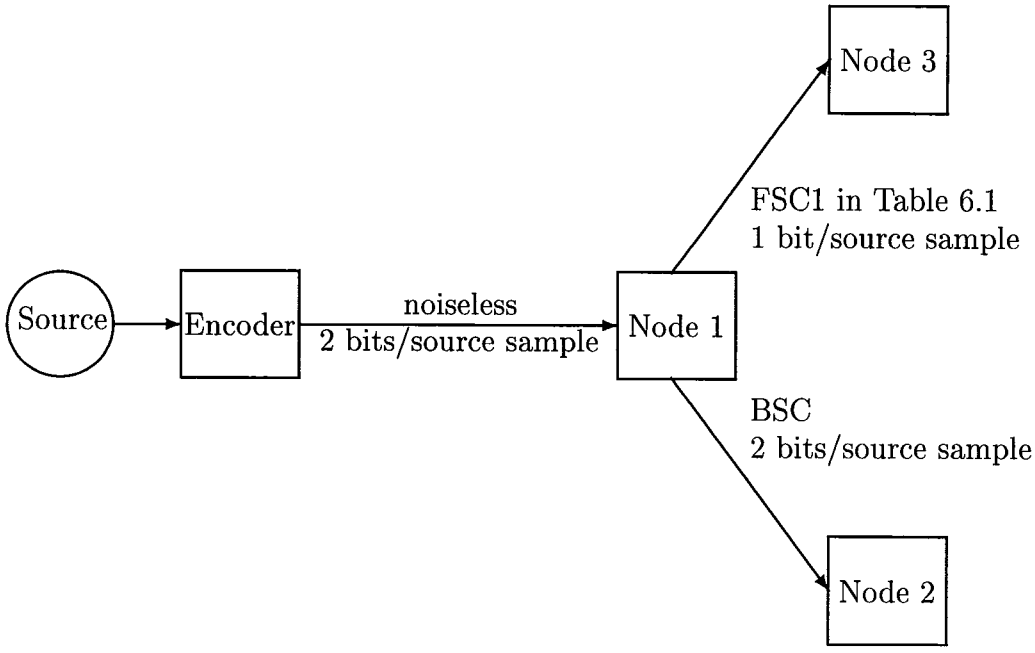


Figure 7.4: Multicasting Example.

Figure 7.5: Reconstructed Image for the Multicasting Example of Figure 7.5; Node 2 (HTVQ); 512×512 Lenna.

Figure 7.6: Reconstructed Image for the Multicasting Example of Figure 7.5; Node 2 (CM-HTVQ); 512×512 Lenna.

Figure 7.7: Reconstructed Image for the Multicasting Example of Figure 7.5; Node 3 (HTVQ); 512×512 Lenna.

Figure 7.8: Reconstructed Image for the Multicasting Example of Figure 7.5; Node 3 (FA-CM-HTVQ); 512×512 Lenna.

Figure 7.9: Reconstructed Image for the Multicasting Example of Figure 7.5; Node 3 (DA-CM-HTVQ); 512×512 Lenna.

Chapter 8

Conclusions

In this dissertation, we have studied the problem of image compression for different applications. In addition to proposing new image coding systems, we have considered different design constraints such as complexity and scalability. We have shown that different applications require different solutions. Some required properties of the system are so important that affect the whole design procedure. For example, designing an image coding system for noisy channels is a new challenge requiring novel strategies.

In Chapter 3, a new scheme for classifying image blocks based on their spectral content has been introduced. A VQ with an appropriate distortion measure has been designed to split the spectral space into a prespecified number of classes. It is shown that not only is spectral classification less complex than gain-based classification, but also it outperforms gain-based classification in an adaptive DWT coding system. The resulting DWT-based system, to the best of our knowledge, provides one of the best available rate-distortion performances in the literature. We have also introduced a table-lookup implementation and a 2-D

version of spectral classification. Different versions of the image coding system are proposed which provide a tradeoff between complexity and performance.

In Chapter 4, we have developed a scalable image coding system using DWT, classification, and MS-PTSVQ. We have shown that due to practical limitations in designing high rate PTSVQ's, using MS-PTSVQ results in a performance improvement. Also, we have investigated the advantages of choosing directional block sizes.

In Chapter 5, we have proposed a scheme for fast reconstruction of a subband-decomposed progressively-transmitted signal. Using the proposed approach, we can update the reconstructed signal after receiving the refinement of each new coefficient and create a continuously-refined perception of the output without any extra computational cost (compared to the conventional approach where the image is reconstructed after receiving a predefined number of bits). In existing scalable image coding systems [Sha93, Sai96, Tau94] (also the system of Chapter 4), insignificant coefficients remain zero at the decoder for a few steps of the refinement process. Also, at each step of the refinement, some of the coefficients remain unchanged. The unrefined coefficients do not add to the computational complexity of the proposed approach. On the other hand, the complexity of the conventional reconstruction scheme does not depend on the value of the coefficients (or refinements). Therefore, the larger is the number of unrefined coefficients the more significant will be the complexity reduction of the proposed approach.

In some applications, after receiving a preliminary draft of the signal, the receiver only needs to upgrade a particular region of interest of the signal. Using the approach proposed in Chapter 5, the complexity reduction in this case is

approximately equal to the ratio of the number of samples in the region of interest area to the number of samples in the entire signal.

In Chapter 6, we have proposed an adaptive rate allocation scheme to optimally choose the rates of the source coder and channel coder pair. We have expressed the end-to-end expected distortion as the sum of the source coder distortion and the channel distortion and proposed a modified Gilbert noise channel to model the combination of channel encoder, channel, and channel decoder. Based on our new model, we have estimated the channel distortion and have selected the best (r_s, r_c) pair to encode and transmit the source. This approach has provided a framework for adaptive transmission of sources over fading channels that requires only low-complexity analytical computations, instead of time-consuming simulations. Simulation results demonstrated the optimality of the rate allocation scheme and the advantage of the adaptation.

In Chapter 7, we have proposed a CM-HTVQ for a discrete memoryless channel and shown that it simultaneously provides low encoding complexity and robustness against transmission noise. The performance gap between the CM-HTVQ and COVQ is much less than the gap between the HTVQ and VQ. The gap shrinks as the channel becomes noisier. Additionally, we have developed two extensions of this work for quantization and transmission over a finite-state channel: (i) A fully-adaptive system for the case where the channel state information is available at the decoder and encoder and (ii) a decoder-adaptive version in which the channel state is known only at the decoder. We have shown that these adaptive systems utilize the knowledge of the channel state to improve the end-to-end performance of the system. Furthermore, we have studied the sensitivity of the CM-HTVQ to channel mismatch. In an interesting, but limited, set of

comparisons against a carefully-designed tandem source-channel code, we have shown that the CM-HTVQ systems (both adaptive and non-adaptive versions) perform close to or better than the tandem system. Finally, we have proposed the use of lookup tables for transcoding in heterogeneous networks. We have shown, through an example motivated by an application in multicasting, that the proposed table-lookup transcoding approach is an elegant alternative to the expensive and delay-introducing method of transcoding achieved by decoding and re-encoding.

Future work includes the extension of ideas in Chapters 3-5 to video coding. Chapters 6 and 7 provide powerful schemes for transmission of memoryless sources over noisy channels (especially finite-state channels). Preliminary results show the usefulness of these schemes for transmission of images over noisy channels. One open problem is the utilization of the source memory to design a robust and efficient image coding system for transmission over wireless channels.

Bibliography

[Ant92] Antonini M.; Barlaud M.; Mathieu P. and Daubechies I. Image Coding Using Wavelet Transform. *IEEE Trans. Image Proc.*, pages 205–220, Apr. 1992.

[Buz80] Buzo A.; Gray,Jr. A.H.; Gray R.M. and Markel J.D. Speech Coding Based upon Vector Quantization. *IEEE Trans. Acoust., Speech, and Signal Proc.*, pages 562–574, Oct. 1980.

[Cha85] Chang P.C.; May J. and Gray R.M. Hierarchical Vector Quantizers with Table-Lookup Encoders. *Proceedings, Int. Conf. Acoust., Speech, and Signal Proc.*, pages 1452–1455, 1985.

[Cha96a] Chaddha N. and Gupta A. A Frame-work for Live Multicast of Video Streams over the Internet. *Proceedings, Int. Conf. Image Proc.*, pages 1–4, Sept. 1996.

[Cha96b] Chaddha N.; Chou P. and Gray R.M. Constrained and Recursive Hierarchical Table–Lookup Vector Quantization. *Proceedings, Data Compression Conf.*, pages 220–229, Apr. 1996.

[Che77] Chen W.H. and Smith C.H. Adaptive Coding of Monochrome and Color Images. *IEEE Trans. Commun.*, pages 1285–1292, Nov. 1977.

[Che96] Chen P.Y.; Acharya T. and Jafarkhani H. A VLSI Architecture for Image Compression Using Wavelet Transform. *Int. Conf. on Systems Engineering*, pages 309–314, July 1996.

[Cho89] Chou P.A.; Lookabaugh T.L. and Gray R.M. Optimal Pruning with Applications to Tree-Structured Source Coding and Modeling. *IEEE Trans. Inform. Theory*, pages 299–315, Mar. 1989.

[Con82] Conway J.H. and Sloane N.J.A. Voronoi Regions of Lattices, Second Moments of Polytopes and Quantization. *IEEE Trans. Inform. Theory*, pages 211–226, Mar. 1982.

[Cos95] Cosman P.C.; Perlmutter S.M. and Perlmutter K.O. Tree-Structured Vector Quantization with Significance Map for Wavelet Image Coding. *Proceedings, Data Compression Conf.*, pages 33–41, Mar. 1995.

[Cox91] Cox R.V.; Hagenauer J.; Seshadri N. and Sundberg C.-E.W. Subband Speech Coding and Matched Convolutional Channel for Mobile Radio Channels. *IEEE Trans. Signal Proc.*, pages 1717–1731, Aug. 1991.

[Cro76a] Crochiere R.E.; Webber S.A. and Flanagan J.L. Digital Coding for Speech in Subbands. *Bell Syst. Tech. J.*, pages 1069–1085, Oct. 1976.

[Cro76b] Croisier A.; Esteban D. and Galand C. Perfect Channel Splitting by Use of Interpolation / Decimation / Tree Decomposition Techniques. *Proceedings, Conf. on Information Sciences and Systems*, pages 443–446, Aug. 1976.

[Ell63] Elliott E.O. Estimates of Error Rates for Codes on Burst-Noise Channels. *Bell Syst. Tech. J.*, pages 1977–1997, Sept. 1963.

[Far84] Farvardin N. and Modestino J.W. Optimum Quantizer Performance for a Class of NonGaussian Memoryless Sources. *IEEE Trans. Inform. Theory*, pages 485–497, May 1984.

[Far90] Farvardin N. A Study of Vector Quantization for Noisy Channels. *IEEE Trans. Inform. Theory*, pages 799–809, July 1990.

[Far91] Farvardin N. and Vaishampayan V. On the Performance and Complexity of Channel-Optimized Vector Quantizers. *IEEE Trans. Inform. Theory*, pages 155–160, Jan. 1991.

[Fis86] Fischer T.R. A Pyramid Vector Quantizer. *IEEE Trans. Inform. Theory*, pages 568–583, July 1986.

[Fis92] Fischer T. and Wang M. Entropy-Constrained Trellis-Coded Quantization. *IEEE Trans. Inform. Theory*, pages 415–426, Mar. 1992.

[Ger92] Gersho A. and Gray R.M. *Vector Quantization and Signal Compression* . Kluwer Academic Publishers , 1992.

[Gil60] Gilbert E.N. Capacity of a Burst-Noise Channel. *Bell Syst. Tech. J.*, pages 1253–1265, Sept. 1960.

[Hag88] Hagenauer J. Rate-Compatible Punctured Convolutional Codes (RCPC Codes) and their Applications. *IEEE Trans. Commun.*, pages 389–400, Apr. 1988.

[Hag90] Hagenauer J.; Seshadri N. and Sundberg C.-E.W. The Performance of Rate-Compatible Punctured Convolutional Codes for Digital Mobile Radio. *IEEE Trans. Commun.*, pages 966–980, July 1990.

[Ita68] Itakura F. and Saito S. Analysis Synthesis Telephone Based upon the Maximum Likelihood Method. *in Conf. Rec., 6th Int. Congr. Acoust.*, 1968.

[Jai89] Jain A.K. *Fundamentals Digital Image Processing* . Prentice-Hall , 1989.

[Jos95] Joshi R.L.; Jafarkhani H.; Kasner J.H.; Fischer T.R.; Farvardin N.; Marcellin M.W. and Bamberger R.R. Comparison of Different Methods of Classification in Subband Coding of Images. *submitted to IEEE Trans. Image Proc.*, Oct. 1995.

[Jos96] Joshi R.L. and Vishwanath M. Improving the Performance of Hierarchical Vector Quantization Using Segmentation. *Proceedings, Int. Conf. Image Proc.*, pages 427–430, Sept. 1996.

[Jua82] Juang B.-H. and Gray, Jr., A.H. Multiple Stage Vector Quantization for Speech Coding. *Proceedings, Int. Conf. Acoust., Speech, and Signal Proc.*, pages 597–600, Apr. 1982.

[Kum84] Kumazawa H.; Kasahara M. and Namekawa T. A Construction of Vector Quantizers for Noisy Channels. *Electron. Eng. Japan*, pages 39–47, 1984.

[Lee93] Lee C.-C. and Farvardin N. Entropy-Constrained Trellis Coded Quantization: Implementation and Adaptation. *Proceedings, Conf. on Information Sciences and Systems*, pages 342–347, Mar. 1993.

[Lin80] Linde Y.; Buzo A. and Gray R.M. An Algorithm for Vector Quantizer Design . *IEEE Trans. Commun.*, pages 84–95, Jan. 1980.

[Loo89] Lookabaugh T. and Gray R.M. High-Resolution Quantization Theory and the Vector Quantizer Advantage. *IEEE Trans. Inform. Theory*, pages 1020–1033, Sept. 1989.

[Mal89] Mallat S.G. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Trans. Pattern Anal. and Mach. Intel.*, pages 674–693, July 1989.

[Mar94] Marcellin M.W. On Entropy-Constrained Trellis Coded Quantization. *IEEE Trans. Commun.*, pages 14–16, Jan. 1994.

[Meh96] Mehrotra S.; Chaddha N. and Gray R.M. Predictive Hierarchical Table-Lookup Vector Quantization with Quadtree Encoding. *Proceedings, Int. Conf. Image Proc.*, pages 407–410, Sept. 1996.

[Mey85] Meyer Y. Principe dínncertitude, bases hilbertiennes et algebres dóperateurs. *Seminaire Bourbaki*, 1985.

[Moa91] Moayeri N.; Neuhoff D.L. and Stark W.E. Fine-Coarse Vector Quantization. *IEEE Trans. Signal Proc.*, pages 1503–1515, July 1991.

[Mus89] Mushkin M. and Bar-David I. Capacity and Coding for the Gilbert-Elliott Channels. *IEEE Trans. Inform. Theory*, pages 1277–1290, Nov. 1989.

[Pea90] Pearlman W. A. Adaptive Cosine Transform Image Coding with Constant Block Distortion. *IEEE Trans. Commun.*, pages 698–703, May 1990.

[Pen93] Pennebaker W.B. and Mitchell J.L. *JPEG Still Image Data Compression Standard.* Van Nostrand Reinhold, 1993.

[Per94] Perlmutter S.M. and Gray R.M. A Low Complexity Multiresolution Approach to Image Compression Using Pruned Nested Tree-Structured Vector Quantization. *Proceedings, Int. Conf. Image Proc.*, pages 588–592, Nov. 1994.

[Pre92] Press W.H.; Teukolsky S.A.; Vetterling W.T. and Flannery B.P. *Numerical Recipes in C: The Art of Scientific Computing.* second edition, University Press, 1992.

[Ram86] Ramamurthi B. and Gersho A. Classified Vector Quantization of Images. *IEEE Trans. Commun.*, pages 1105–1115, Nov. 1986.

[Rap96] Rappaport T.S. *Wireless Communications Principles and Practice.* IEEE Press, New York, 1996.

[Ris91] Riskin E. A. Optimal Bit Allocation via the Generalized BFOS Algorithm. *IEEE Trans. Inform. Theory*, pages 400–402, Mar. 1991.

[Sai96] Said A. and Pearlman W.A. A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees. *IEEE Trans. Circ. & Syst. Video Tech.*, pages 243–250, June 1996.

[Sha48] Shannon C.E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.*, pages 379–423, 1948.

[Sha59] Shannon C.E. Coding Theorems for a Discrete Source with a Fidelity Criterion. *IRE Nat. Conv. Rec.*, pages 142–163, Mar. 1959.

[Sha93] Shapiro J.M. Embedded Image Coding Using Zerotrees of Wavelet Coefficients. *IEEE Trans. Signal Proc.*, pages 3445–3462, Dec. 1993.

[Sho88] Shoham Y. and Gersho A. Efficient Bit Allocation for an Arbitrary Set of Quantizers. *IEEE Trans. Acoust. Speech, and Signal Proc.*, pages 1445–1453, Sept. 1988.

[Stu87] Stuart A. and Ord J.K. *Kendall's Advanced Theory of Statistics; Vol. 1: Distribution Theory.* fifth edition, Charles Griffin, 1987.

[Tan92] Tanabe N. and Farvardin N. Subband Image Coding Using Entropy-Coded Quantization over Noisy Channels. *IEEE J. Select. Areas in Commun.*, pages 926–943, June 1992.

[Tau94] Taubman D. and Zakhor A. Multirate 3-D Subband Coding of Video. *IEEE Trans. Image Proc.*, pages 572–588, Sept. 1994.

[Vem95] Vembu S.; Verdu S. and Steinberg Y. The Source-Channel Separation Theorem Revisited. *IEEE Trans. Inform. Theory*, pages 44–54, Jan. 1995.

[Vet91] Vetterli M. Multirate Filter Banks for Subband Coding. *in Subband Image Coding*, pages 43–100, 1991.

[Vis94] Vishwanath M. and Chou P. An Efficient Algorithm for Hierarchical Compression of Video. *Proceedings, Int. Conf. Image Proc.*, pages 275–279, Nov. 1994.

[Wan90] Wang H.S. and Moayeri N. Optimal Quantization for a Finite-State Noisy Channel. *Abstracts, Int. Symp. Inform. Theory*, page 79, Jan. 1990.

[Wan95a] Wang H.S. and Moayeri N. Finite-State Markov Channel – A Useful Model for Radio Communication Channels. *IEEE Trans. Vehicular Tech.*, pages 163–171, Feb. 1995.

[Wan95b] Wang H.S. and Moayeri N. Finite-State Markov Channel–A Useful Model for Radio Communication Channels. *IEEE Trans. Vehicular Tech.*, pages 163–171, Feb. 1995.

[Wit87] Witten I.H.; Neal R.M. and Cleary J.G. Arithmetic Coding for Data Compression. *Communications of the ACM*, pages 520–540, June 1987.

[Woo86] Woods J.W. and O'Neil S.D. Subband Coding of Images. *IEEE Trans. Acoust., Speech, and Signal Proc.*, pages 1278–1288, Oct. 1986.

[Woo92] Woods J.W. and Naveen T. A Filter Based Bit Allocation Scheme for Subband Compression of HDTV. *IEEE Trans. Image Proc.*, pages 436–440, July 1992.

[Xio96] Xiong Z.; Ramchandran K. and Orchard M.T. Space-Frequency Quantization for Wavelet Image Coding. *submitted to IEEE Trans. Image Proc.*, Oct. 1996.

[Zeg90] Zeger K. and Gersho A. Pseudo-Gray Coding. *IEEE Trans. Commun.*, pages 2147–2158, Dec. 1990.