SOURCE AND CHANNEL CODING OF IMAGES FOR NOISY CHANNELS

BY

PHILLIP GREGORY SHERWOOD

B.S., Massachusetts Institute of Technology, 1990
B.S., Massachusetts Institute of Technology, 1990
M.S., Massachusetts Institute of Technology, 1990

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2000

Urbana, Illinois

SOURCE AND CHANNEL CODING OF IMAGES FOR NOISY CHANNELS

Phillip Gregory Sherwood, Ph.D.
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, 2000
Kenneth Zeger, Advisor

Wireless multimedia devices and services are beginning to emerge and will continue to increase in popularity as capabilities improve. The demand to improve quality and increase the capacity of such systems within limited bandwidth resources motivates the interest in efficient and error-resilient image coding methods. An important problem is how to efficiently encode images for transmission over limited-bandwidth channels while maintaining good decoded image quality and resilience to channel impairments. This thesis investigates the problem of robust image transmission over memoryless, fading, and packet erasure channels.

First, a system consisting of a high-performance embedded wavelet coder followed by a concatenated channel code is proposed for transmission over memoryless channels. The system maintains the progressive nature of the source coder, has reasonable complexity, and exceeds the performance of previously known systems. Error resilience and progressivity are further explored through the problem of channel code parameter (block length and channel code rate) optimization for progressive transmission. A general performance measure for evaluating progressive image coding quality is proposed and a dynamic programming optimization algorithm is presented. Results are included for erasure channels and bit error channels. In addition, the performances of turbo codes, low-density parity check codes, and concatenated channel codes are compared in this context.

For fading channels, the concatenated code proposed for memoryless channels is extended to a product code structure. The new code shows improved performance on fading channels and slightly improved performance on memoryless channels. It also has variable amounts of delay, depending on the level of the channel noise, allowing the progressive performance under good channel conditions to remain high. A cumulative distribution of decoded PSNR values is proposed for evaluating performance on these variable channels instead of the traditional expected PSNR value. Modifications to the source coder to increase its inherent error resilience in combination with explicit channel coding are considered for varying channels, including those that introduce both bit errors and packet erasures. A general performance measure is proposed to facilitate the optimization of system parameters. Improvements are shown in terms of robustness to a range of channel conditions.

Finally, methods for introducing an adjustable amount of source redundancy are explored in the context of erasure channels. The erasure channel can be considered an abstraction of an arbitrary physical channel as long as reliable error detection is possible, so the methods are broadly applicable. A multistage coding structure is proposed which offers much flexibility, even in terms of the underlying compression algorithms, as well as low complexity. The method is demonstrated with two specific image coders. Extensions of multiple description scalar quantizers in combination with erasure correcting codes are proposed as another method.

To my wife, Lori

ACKNOWLEDGMENTS

I would like to thank my advisor Professor Ken Zeger for his support and guidance during my graduate study. I would also like to express my sincere appreciation to members of my committee, Professors Doug Jones, Upamanyu Madhow, Pierre Moulin, and Kannan Ramchandran.

I certainly want to thank the members of family for their support and encouragement, especially my wife Lori. Lori provided tireless support during the entire process. She was even willing to listen to practice talks and proofread papers!

During my years of graduate study I had an opportunity to collaborate on research with a number of people including Professor Pamela Cosman, Tamas Frajka, Jon Rogers, and Xiaodong Tian. I would like thank each of them for their contributions to my research and education.

TABLE OF CONTENTS

CHAPTER **PAGE**

1 INTRODUCTION 1
1.1 A Historical Perspective 1
1.2 Organization and Contributions 4

2 BACKGROUND 8
2.1 Introduction 8
2.2 Image Coding 8
2.2.1 Lossy image coding structures and terminology 8
2.2.2 Zerotree wavelet coding 12
2.3 Channel Models 15
2.3.1 Memoryless channels 15
2.3.2 Fading channels 17
2.3.2.1 Gilbert-Elliot channel 17
2.3.2.2 The Rayleigh channel 18
2.3.3 Multiple descriptions and the packet erasure channel 19
2.3.4 Combined packet erasure and fading channel 20

3 IMAGE CODING FOR MEMORYLESS CHANNELS 21
3.1 Introduction 21
3.2 A Concatenated Channel Code Using RCPC and CRC Component Codes . . . 23
3.2.1 Decoding complexity 24
3.2.2 Performance results 26
3.2.3 Parameter optimization and other extensions 30
3.3 Block Length and Rate Optimization for Progressive Coding 33
3.3.1 A performance measure for progressive transmission 35
3.3.2 Optimization using dynamic programming 36
3.3.3 Progressive transmission over erasure channels 39
3.3.4 Progressive transmission over bit error channels 43
3.3.5 Conclusion 47

4 IMAGE CODING FOR FADING CHANNELS 48
4.1 Introduction 48
4.2 A Product Channel Code 49
4.2.1 Performance on a binary symmetric channel 50
4.2.2 Performance on fading channels 51
4.2.3 Performance on the Gilbert-Elliot channel 52
4.2.4 Performance on the Rayleigh channel 54
4.2.5 Applying unequal error protection 58
4.3 Combining Robust Source Coding and FEC for Varying Channels 60
4.3.1 Parameter value optimization 62
4.3.2 Results 66
4.3.2.1 Visual quality 67
4.3.2.2 Mean decoded MSE 68
4.3.2.3 Cumulative distributions of MSE 69
4.3.2.4 With packet erasures 71
4.3.3 Conclusion 72
4.3.4 Code Parameters 72

5 IMAGE CODING FOR ERASURE CHANNELS 76
5.1 Introduction 76
5.2 Multistage Coding 77
5.2.1 A multistage zerotree wavelet coder 80
5.2.1.1 The stage schedule 80
5.2.1.2 Modifications to improve multistage performance 81
5.2.1.3 Overhead information 82
5.2.1.4 Experimental results 83
5.2.2 A packetized multistage image coder 84
5.2.2.1 Finding a good image partition 86
5.2.2.2 Reducing distortion at the decoder 88
5.2.2.3 Experimental results 88
5.3 Multistage Analysis 91
5.3.1 Analysis of the encoder and decoder design 92
5.3.2 Optimal decoder design 94
5.3.2.1 Training-set implementation 97
5.3.2.2 Multistage specialization 98
5.3.2.3 Extensions to long stages 100
5.3.3 Optimal encoder design 106
5.4 Multiple Description Coding for Packet Erasure Channels 107
5.4.1 FEC for erasure channels 109
5.4.2 Proposed image coding structure 110
5.4.3 Multidimensional multiple description scalar quantizer 112
5.4.4 Parameter optimization 116

6 CONCLUSIONS AND FUTURE WORK 118

REFERENCES 122

VITA 130

CHAPTER 1

INTRODUCTION

This thesis addresses the problem of source and channel coding of images for transmission over noisy channels. Image compression has been extensively studied for the past several decades for both noiseless and noisy channels. However, considerably more progress has been achieved for the noiseless channel case. Modern communication systems face significant challenges including such things as multipath fading channels and multiuser interference. Handheld wireless devices, for example, must operate in severe channel conditions with limited bandwidth and good power efficiency. Both hardware capabilities and bandwidth allocations are beginning to emerge to support multimedia applications on these platforms. It is thus an important and challenging problem for these channels to design image transmission systems that perform well in both good and bad channel conditions.

This introduction provides some historical perspective on the relevant aspects of information theory to the image transmission problem at hand. A brief history of important events and key developments in information theory is described. In Section 1.2 the key topics and organization of the thesis are outlined along with some discussion of related work.

1.1 A Historical Perspective

Transmission of information by electronic means dates back as far as the 1830s when messages were transmitted using Morse code via the telegraph. Morse code is an early example of data compression, where more common letters are represented with shorter symbols (e.g., a single dot for the letter “E”). While Morse code represents an early application, it was not until 1948 that key ideas of data compression were formally introduced with the birth of information theory. Because a Morse coded message can be perfectly recovered from the symbols (dits and dots) it can be referred to as a *lossless* compression scheme. For many types of messages only lossless compression is acceptable, but for analog information sources such as audio, image, video, and voice, some loss is usually acceptable and can often be made imperceptible (audibly or visually) to humans.

Voice communication by analog signals transmitted over wires became a reality with invention of the telephone in the 1870s. This invention was quickly followed in the 1890s and early 1900s by the inventions of the wireless telegraph and AM radio. At that point analog voice messages could be transmitted over wireless links. Transmission of visual information soon followed with the invention of the television in the mid 1920s.

Until the invention of pulse-code modulation in the late 1930s, no digital methods had been used to transmit analog continuous-time signals. The digital representation of analog signals opened up new possibilities for data compression and transmission. One of the first uses of lossy compression was the Vocoder [1] where the bandwidth requirement of a voice signal could be reduced at the cost of fidelity.

In 1948, Claude Shannon published “A mathematical theory of communication,” [2] which describes the fundamental laws of data transmission over noisy channels and data compression. This publication is considered to mark the birth of the field of information theory. The main idea in Shannon’s work was to model information transmission from a probabilistic viewpoint. In terms of source coding, Shannon showed in his 1948 paper that there is a lower bound (related to the statistics of the source) to the amount a source can be compressed and still be perfectly reconstructed. This result describes fundamental limits on the capabilities of lossless compression. Examples of lossless compression methods include Huffman, Lempel-Ziv, and arithmetic coding. Many texts on information theory such as [3, 4] can provide more details.

In the area of coding for noisy channels, Shannon dispelled the belief that the channel code rate had to go to zero to achieve arbitrarily small error probabilities. Instead he discovered a quantity known as the capacity which characterizes the channel’s transmission potential, and he showed that arbitrarily small error probabilities could be achieved at any rate below the capacity. The proof was based on random coding arguments showing that the average over all possible codes was good, so at least one good code exists. The question of finding practical codes with the promised performance remained an open problem in his paper. In more than 50 years since Shannon's seminal paper, much progress has been made in the area of channel coding and modulation, but the problem of achieving capacity is still not solved.

Shannon also provided the framework for lossy compression in his *source coding theorem subject to a fidelity criterion* [2, 5]. The goal of lossy compression is not to perfectly reconstruct the source, but to reconstruct within a given tolerance. Shannon defines a bound, called the rate-distortion function of the source, which gives the minimum rate at which the source information can be encoded within the specified distortion. He then shows that there exist source coders which can code at rates arbitrarily close to the bound by using long blocks of source samples.

Source and channel coding ideas are combined in Shannon's "separation principle," which states that in the limit of long block length there is no loss of optimality in designing the optimal source coder without concern for the channel (i.e., noise-free conditions) and optimal channel coder without concern for the source (i.e., assuming i.i.d. equally likely source bits) [3]. The result assumes no constraints on delay or complexity. For a practical system, these asymptotic conditions cannot typically be met, so some type of joint source-channel design can be beneficial in certain situations.

Applications such as telephone modems, compact disks, computer hard drives, and cellular telephones have driven many of the developments in the areas of source and channel coding. For example, the worldwide cellular phone market for handsets was at 160 million units in 1998, 250 million units in 1999, and there are some projections of as many as 500 million units in 2000. This rapid growth is causing a large demand for improvements. The third generation wireless standards, called "3G," will roll out soon and promise much higher bandwidths (e.g., 2 Mbits/s for stationary indoor, 144 kbits/s for mobile [6]). As capabilities have increased so have the demands for new and more complex products and services. Most recently, wireless multimedia products are on the horizon and present many challenges in the area of robust audio, image, and video coding.

1.2 Organization and Contributions

Almost all modern image compression algorithms are based on quantizing and coding of a transformed representation of the image (typically the output of a linear transform). For example, the current Joint Photographic Experts Group (JPEG) image coding standard [7] is based on quantization and coding of 8×8 discrete cosine transform (DCT) coefficients. More recently coders based on wavelet transforms [8, 9, 10, 11, 12] have been proposed which show significant improvements over the DCT-based methods, especially at low rates. In addition, many of these coders are *progressive*, meaning that the bit stream they produce can be decoded at more than one rate yielding improved quality with increasing rate. This property allows a single bit stream to satisfy a number of quality levels. However with the excellent compression performance typically comes high sensitivity to errors due to the large amount of state information required for correct decoding and the use of variable length codes. In many cases, errors or erasures cause the decoder to lose synchronization, causing even correctly received source information to be useless to the decoder since it cannot be interpreted correctly. If the coder is very likely to lose synchronization upon suffering an error and resynchronization typically requires a large number of bits (some coders may never resynchronize), the coder may be said to have a *serial decoding requirement* to emphasize the dependencies within the bit stream. Chapter 2 provides more background information on image coding as well as the channel models used throughout the thesis.

One approach to joint source-channel coding, often referred to as joint source-channel matching or concatenated source-channel coding, considers the cascade of known source and channel coders and determines how to best allocate rate, possibly with unequal error protection (UEP), between the source and channel codes. Numerous references exist for this approach, including [13, 14, 15, 16, 17, 18, 19]. Analytic results on the optimal rate allocation for vector quantizers are presented in [20].

In Chapter 3 a system composed of a high-performance progressive wavelet image coder and a concatenated channel code is presented for transmission over memoryless channels. The performance obtained exceeds that of previously known systems. In the same chapter, the problem of channel code parameter optimization, both blocklength and rate, is explored for progressive transmission over noisy channels. A general performance measure is defined and a dynamic programming solution is proposed. Results are provided for Reed-Solomon codes on erasure channels and a comparison between turbo, low density parity check, and RCPC/CRC codes is provided for bit error channels.

The coding method in Chapter 3 is extended to fading channels in Chapter 4 which are a more accurate model of many wireless environments.

The source-channel matching method may work well for certain known channels as the results in Chapter 3 will show, and in cases where a standard algorithm is required, it may be the only option. However, when the channel is unknown or varying, or when there are multiple simultaneous channels as in a broadcast environment, the designer may have to select a channel code for the worst-case channel conditions to get reasonable performance in that case. The reason is that the decoded error probability of most channel codes exhibits a fast transition to poor performance as the channel conditions become noisier than the design conditions (this was studied analytically in [21]). In that case, it is important for the source coding algorithm to degrade gracefully in the presence of errors even if some compression performance must be sacrificed under noiseless conditions. In a certain sense, this can be accomplished by adding source code redundancy instead of (or in addition to) channel code redundancy.

A number of methods have been previously developed to improve source reproduction performance on noisy channels by exploiting source redundancy. The source redundancy is generally either explicitly inserted or is a byproduct of source coder imperfections. The source redundancy is most commonly used to maintain synchronization, allow smoothing algorithms to reduce large distortions, or detect channel decoding errors [22, 23, 24, 25, 26, 27]. Unfortunately, the performance of most robust image coders lags significantly behind that of the best image compression algorithms designed for noiseless transmission [28]. Also, there is generally no convenient method for adjusting the level of source redundancy to better match channel conditions.

Source redundancy has also been used to modify the a priori bit probabilities in the channel decoding algorithm in [29, 30, 31, 32]. However, it is often not practical to accurately estimate the probabilities of source coder output sequences. Also, in sequence decoding there may be a significant delay before incorrect candidate sequences can be eliminated by the channel decoding algorithm (because they are improbable source sequences).

In Section 4.3, source redundancy is introduced through the use of a robust source coder in combination with forward error control (FEC) coding. A general optimization criterion is proposed for selecting the proper channel coding rate. The coder is shown to degrade gracefully and perform well on channels with both bit errors and packet erasures.

One problem with the previous coder is that the amount of source redundancy is not adjustable. It is desirable to develop coders where both the channel code and source code redundancy can be tuned for the requirements and channel conditions. One goal is to develop specific and practical methods for high-performance and robust image coding with this idea in mind. Chapter 5 presents methods and analysis for coders with a tunable amount of source redundancy.

In summary, contributions of the research in this thesis include:

- A highly flexible and practical image transmission system for memoryless channels consisting of a cascade of the SPIHT image coding algorithm and RCPC/CRC concatenated channel codes. This system has performance exceeding that of previously published results [19, 28].

- Extensions of the previous system to fading channels. A new flexible channel coding technique designed to fit well with the source coder and offer good protection over fading channels is presented. Although the system is designed primarily for fading channels, it actually exceeds the performance of the previous system on memoryless channels [33, 34].

- Definition of a general performance measure for progressive coding. A dynamic programming optimization of channel code block length and rate for maximizing the performance measure is presented [35].

- Definition of a general optimization criterion for coders operating on highly variable channels. An error-resilient coder is introduced combining robust source coding and explicit channel coding which degrades gracefully and performs well on channels that introduce both bit errors and packet erasures [36, 37].

- A macroscopic multistage encoding structure which provides a method for tunable source redundancy. The method is shown to provide improved robustness over recently published results [38].

Finally, a summary and some conclusions as well as suggestions for possible future research directions are discussed in Chapter 6.

CHAPTER 2

BACKGROUND

2.1 Introduction

This chapter presents terminology, definitions, and background information on image compression algorithms and channel models that will be used throughout the thesis. Section 2.2 discusses lossy image coding and Section 2.3 covers the channel models.

2.2 Image Coding

This section presents some background material on lossy image compression which is relevant to the experimental results and specific source coding concepts in this thesis. Terminology relating to image coding is explained, and a brief description, including a numerical example, of zerotree wavelet coding is provided.

2.2.1 Lossy image coding structures and terminology

A lossy compression algorithm is not constrained by the requirement that the decoded image must be identical to the original as in lossless compression. Instead the goal is to achieve good fidelity according to some measure with as few bits as possible. The distortion and rate can be traded off according to their relative importance for the application. In image coding, the rates are typically normalized by the number of pixels in the image and reported as bits per pixel (bpp) so that reported rates are independent of image dimensions.

A common distortion measure used in image coding is the mean squared error (MSE). The MSE for a reconstructed image $\hat{X}$, of dimension $N \times M$, with respect to the original image X is given by

$$\text{MSE} = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (X_{i,j} - \hat{X}_{i,j})^2. \tag{2.1}$$

The value is often converted to a logarithmic scale to make it easier to view large dynamic ranges. The most common measure is peak signal-to-noise ratio (PSNR) which is the ratio of the peak amplitude squared to the MSE or

$$\text{PSNR} = \frac{(\text{peak amplitude})^2}{\text{MSE}}. \tag{2.2}$$

Images used throughout this work were grayscale with resolution of 8 bpp, so in this case the PSNR equation reduces to

$$\text{PSNR} = \frac{(255)^2}{\text{MSE}}. \tag{2.3}$$

Almost all high-performance image compression algorithms consist of some type of transform (typically a linear transform) followed by quantization of the transform coefficients, and then lossless entropy coding of the quantized values. Figure 2.1 shows a block diagram of the basic encoder structure. The specific choices of transform, quantizers, and encoding of quantization symbols (including any entropy coding) lead to a wide variety of coders such as DCT-based coders like the JPEG standard [7] to more recent wavelet-based coders like Embedded Zerotree Wavelet (EZW) [11] and Set Partitioning in Hierarchical Trees (SPIHT) [10]. The transform stage acts to pack the energy into a smaller set which can be coded more conveniently than direct coding of the original image. All the encoding loss is confined to the quantization stage where the quantizer resolution provides the tradeoff between rate and distortion. Finally, the entropy coding removes some of the residual redundancy among the quantizer symbols. More information on transform coding can be found in [39].

These high-performance compression algorithms tend to use variable length coding as well as maintain a large amount of state information. Interpretation of input bits depends on the current state of the coder, which may be a function of many previously received bits. An image

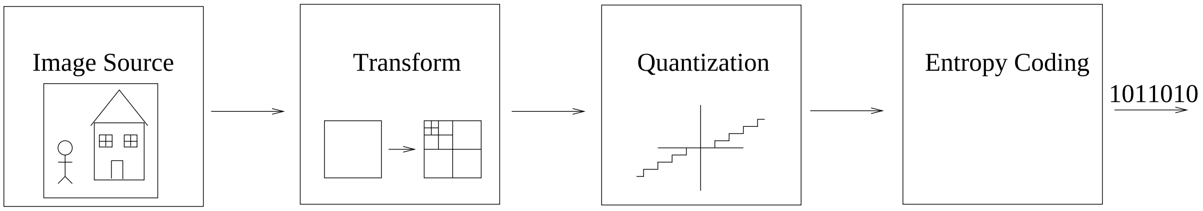
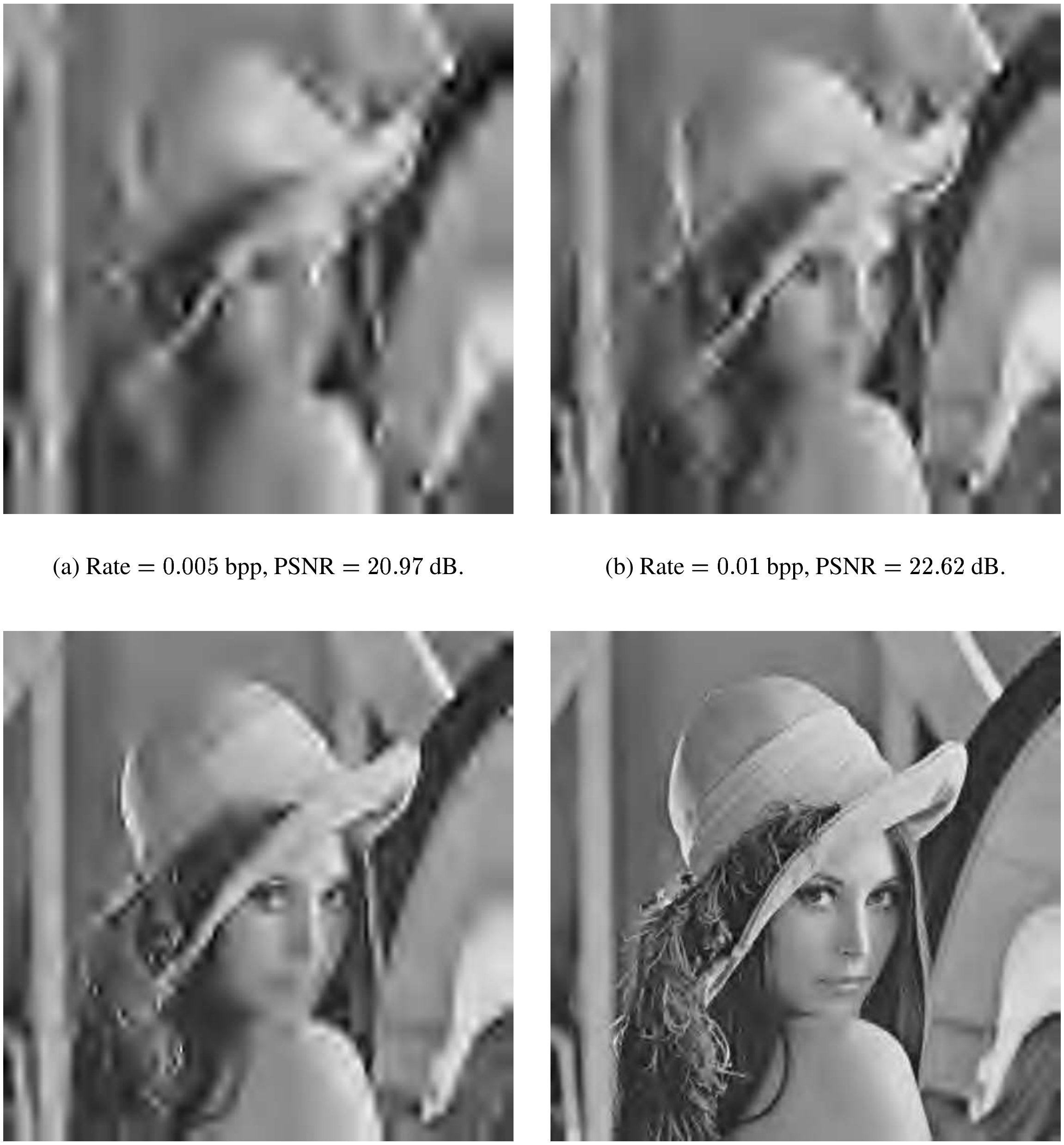


Figure 2.1 Block diagram of typical image coding system.

coder is said to have a *serial decoding requirement* if the decoder must decode the bit stream in a sequential and uninterrupted fashion for correct interpretation of the bits. This concept is basically concerned with the resynchronization time of the decoder after an error (i.e., how many correct bits after an error before the decoder is correctly decoding symbols). Algorithms with very long resynchronization times can effectively be considered to have a serial decoding requirement.

Within the class of lossy image coders, coders can be categorized according to certain properties of the bit streams they produce. A bit stream is said to be *progressive* if it can be decoded at more than one transmission rate to yield potentially coarser image qualities at lower transmission rates. One bit stream is said to be *embedded* in a second bit stream if the first bit stream is a prefix of the second bit stream. An image coder is *progressive* if the encoder output bit stream at every transmission rate is progressive. An image coder is said to be *embedded* if the encoder output bit stream for every transmission rate is embedded in the encoder output bit stream for any higher transmission rate. If an image coder is embedded, then it is clearly progressive. Figures 2.2(a) - 2.2(d) illustrate the progressive concept by showing a series of images decoded from a single bit stream at increasing rates.

The "amount of progressivity" of an image coder qualitatively refers to the number of different transmission rates that yield progressive bit streams. Similarly, the "amount of embeddedness" refers to the number of different transmission rates whose bit streams are embedded in those of higher transmission rates. Coders that are nonembedded typically produce different bit streams depending on the rate for which they are optimized.



(a) Rate $= 0.005$ bpp, PSNR $= 20.97$ dB.

(b) Rate $= 0.01$ bpp, PSNR $= 22.62$ dB.

(c) Rate $= 0.02$ bpp, PSNR $= 24.40$ dB.

(d) Rate $= 0.1$ bpp, PSNR $= 29.80$ dB.

Figure 2.2 Demonstration of progressive image coding for the 512×512 Lena image using the SPIHT coder.

Progressive coders are attractive because they allow a single transmission to provide different levels of service and can help reduce delay in browsing applications by allowing quick rejection of undesired images. Also the compression performance cost of using a highly progressive coder is usually not high. A tabulation of recent results on several natural images shows that current state-of-the-art embedded coders are usually within 1 dB in PSNR and often much closer to the performance of rate-optimized (nonembedded) coders [40].

2.2.2 Zerotree wavelet coding

The familiar tree-structured dependency of wavelet coefficients is shown in Figure 2.3. The figure shows a three-stage wavelet decomposition of the 512x512 Lena image with the tree structure for one particular spatial region highlighted. Coefficients with large magnitudes have dark gray levels in the image, and each subband has been normalized so the relatively large coefficients in each subband will be visible. The dependency between the subband coefficients is clearly demonstrated by the appearance of recognizable elements of the original image in each subband. Hereafter, a *wavelet tree* will refer to the set of wavelet coefficients with a *root* coefficient in the lowest frequency subband together with its descendants from the other subbands as depicted in Figure 2.3.

A number of high-performance image coding algorithms have been proposed in recent years based on exploiting the tree-structured dependency among wavelet subband coefficients [11, 10, 9, 12]. Shapiro's EZW [11] algorithm and the refined algorithm, SPIHT [10], by Said and Pearlman are two examples. Both coders are embedded and have excellent compression performance.

The term *zerotree* refers to an efficient method of encoding groups wavelet coefficients. The main idea is to efficiently encode the regions of insignificance (i.e., wavelet domain regions with small coefficient magnitudes relative to some threshold) with the zerotree symbol, which plays a role similar to the end-of-block symbol in JPEG coding. If a coefficient and all its descendants in a wavelet tree are insignificant with respect to the given threshold, then the coefficient belongs to a zerotree. Both EZW and SPIHT use this concept to encode the wavelet coefficients in a bitplane fashion.

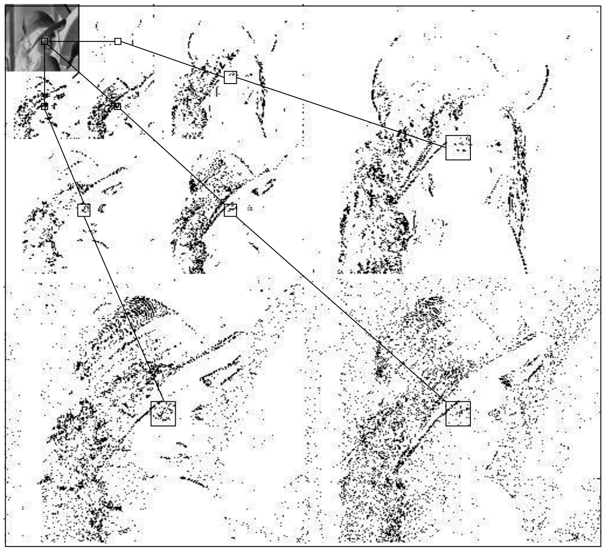


Figure 2.3 An example wavelet domain image demonstrating the tree-structured dependency of coefficients.

Figure 2.4 shows a numerical example demonstrating one iteration of the EZW coding algorithm. The wavelet coefficient values of a three-level wavelet decomposition are shown with the lowest frequency subband in the upper left corner. The table lists the coefficients in the order they are visited during the first iteration, the symbol value output by the encoder, and the decoder reconstruction estimate after decoding the symbol. In this example, the initial threshold was selected to be the largest power of two that is less than the maximum coefficient magnitude. The coefficients with magnitudes above the threshold (i.e., *significant* coefficients) are identified by either the POS or NEG symbols, which also indicate the sign value and result in a nonzero decoded value for that pass. In subsequent iterations, the previously found significant coefficients are further refined by one bit for each coefficient to reduce the size of the uncertainty interval (referred to as the *refinement pass*). As the threshold is reduced by a factor of two at each iteration, the result is a bit-plane encoding of the coefficient values. The encoder output symbols are then losslessly compressed with an entropy coder.

For the purpose of robust image coding, it is often important to partition an image into pieces prior to encoding. Each piece is treated like an independent image by the encoding algorithm. Partitioning an image can avoid error propagation by localizing all of the distortions

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

Subband	Value	Symbol	Decoded Value
LL3	63	POS	48
HL3	-34	NEG	-48
LH3	-31	IZ	0
HH3	23	ZTR	0
HL2	49	POS	48
HL2	10	ZTR	0
HL2	14	ZTR	0
HL2	-13	ZTR	0
LH2	15	ZTR	0
LH2	14	IZ	0
LH2	-9	ZTR	0
LH2	-7	ZTR	0
HL1	7	Z	0
HL1	13	Z	0
HL1	3	Z	0
HL1	4	Z	0
LH1	-1	Z	0
LH1	47	POS	48
LH1	-3	Z	0
LH1	-2	Z	0

POS = Positive, NEG = Negative, ZTR = Zerotree Root,
IZ = Isolated Zero, Z = Zero

Figure 2.4 Coding example of EZW algorithm on a three-level wavelet-domain image.

caused by losses in any one piece. A *subimage* refers to a piece (partition element) of an entire image which is coded independently. For the coders that follow, a subimage specifically consists of an ordered set of wavelet trees. Also, the partition of the original image into subimages will be called the *image partition*, and it refers to the partition of an ordered set throughout this document.

2.3 Channel Models

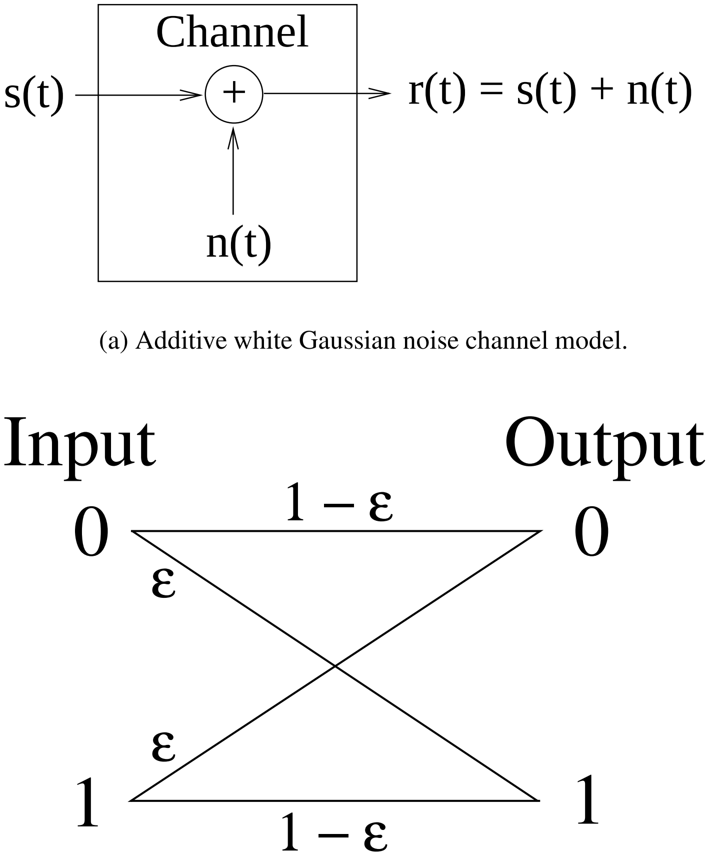
Different transmission environments (e.g., deep space, terrestrial wireless, packet wireline, etc.) introduce different impairments on transmitted data. The exact nature of the channel plays an important role in determining good error protection strategies. This chapter provides some background information on the types of channels used throughout this work.

2.3.1 Memoryless channels

Memoryless channels are the predominant type of channel model used in communication system design and analysis. In this class of channels, the noise affects each transmitted symbol independently. Figure 2.5(a) shows a diagram of the additive white Gaussian noise (AWGN) channel model where $s(t)$ is the transmitted signal, $n(t)$ is the white Gaussian noise process, and $r(t)$ is the received signal. Error probabilities for digital communication depend on the signal-to-noise ratio (SNR) as well as the modulation scheme. For binary phase-shift keying (BPSK) modulation the bit error probability for the AWGN channel is given by

$$P_b = Q(\sqrt{2 \cdot \mathrm{SNR}}) = \frac{1}{2}\mathrm{erfc}(\sqrt{\mathrm{SNR}}). \tag{2.4}$$

In some systems the demodulator can provide reliability information about the demodulated symbol values, which can be useful in decoding channel codes. For those systems that only provide hard bit decisions, the channel and demodulator together can be modelled by an equivalent discrete channel, the binary symmetric channel (BSC). A diagram of the BSC is shown in Figure 2.5(b) where the crossover probability ϵ is the bit error probability. This bit error probability is also referred to as the bit error rate (BER) of the channel. For each bit transmitted, the correct bit is received with probability $1 - \epsilon$; otherwise, the opposite bit is received.



(a) Additive white Gaussian noise channel model.

(b) Binary symmetric channel model.

Figure 2.5 Memoryless channel models.

2.3.2 Fading channels

In contrast, fading channels exhibit memory so errors tend to occur in clusters or bursts when the channel is in a bad state. When designing for a fading channel it is important to consider the average fade durations of various severities in addition to the average channel error rate.

2.3.2.1 Gilbert-Elliot channel

The Gilbert-Elliot channel model is a simple two-state Markov model for a discrete channel with memory. The state diagram of the model is shown in Figure 2.6. Within each state the channel acts like a BSC of the appropriate error rate (ϵ_{B} for the bad state and ϵ_{G} for the good state) and at each bit interval the channel changes state with probabilities governed by the model transition probabilities $P_{GB} = \Pr(good \to bad)$ and $P_{BG} = \Pr(bad \to good)$. The average fade duration or mean holding time in the bad state is given by

$$\bar{\tau} = \frac{1}{P_{BG}} \tag{2.5}$$

and the average bit error probability is

$$\bar{P_b} = \epsilon_{\mathrm{G}} \frac{P_{BG}}{P_{BG} + P_{GB}} + \epsilon_{\mathrm{B}} \frac{P_{GB}}{P_{BG} + P_{GB}}. \tag{2.6}$$

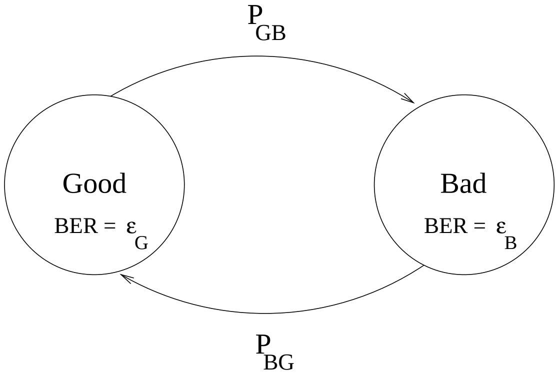


Figure 2.6 Gilbert-Elliot fading channel model.

2.3.2.2 The Rayleigh channel

The Rayleigh channel is a fading channel where the state varies over a continuous range as opposed to the two state Gilbert-Elliot model. In wireless transmission systems, the received signal power varies due to constructive and destructive interference between multiple copies of the signal arriving at the antenna along different paths. If the delay experienced by all of the paths is short relative to a bit interval, the channel is said to be a *flat-fading* channel. In this case, the signal waveform shape is not distorted, and the effect of the multipath is to introduce a random multiplicative gain α on the received signal. The random gain α has a Rayleigh distribution [41]. This document only discusses flat-fading Rayleigh channels.

For BPSK modulation, the error rate or average error probability is given by

$$\mathrm{BER} = \frac{1}{2}\left(1 - \sqrt{\frac{\overline{\mathrm{SNR}}}{1+\overline{\mathrm{SNR}}}}\right) \tag{2.7}$$

where $\overline{\mathrm{SNR}}$ is the average received SNR. Note that the SNR quantities in the equations are on a linear scale, whereas in the text they are reported in dB according to the usual convention. As an example, an average SNR of 10 dB implies a BER of 0.023, and an average SNR of 20 dB implies a BER of 0.0025. Unlike memoryless channels, such as the BSC and AWGN channels, the errors in the Rayleigh channel tend to occur in bursts. Therefore another channel statistic of interest is the average burst or fade duration.

Throughout this work, Jakes' method [42] is used to simulate BPSK transmission over a Rayleigh channel. With this model, the channel is characterized by two parameters—the average SNR and the normalized Doppler spread f_D. The maximum Doppler shift f_m of a signal with carrier frequency f_c and mobile speed v is given by

$$f_m = f_c \frac{v}{c} \tag{2.8}$$

where c is the speed of light. For example, if the carrier frequency is 900 MHz and the mobile speed is 4 miles/h, the resulting maximum Doppler shift is $f_m = 5.38$ Hz. Dividing f_m by the data rate gives the normalized Doppler spread (i.e., $f_D = f_m/R$ where R is the data rate). If the data rate is 500 kbits/sec and the other parameters are as previously mentioned, the normalized Doppler spread is $f_D = 1.07 \times 10^{-5}$.

The average fade duration is also dependent on the fade margin, which characterizes the amount that the SNR can be reduced before communication becomes unreliable. For BPSK transmission, the probability of a bit error given a received signal-to-noise ratio SNR_r is given by (2.4) with $\mathrm{SNR} = \mathrm{SNR}_r$. A value $P_b = 0.1$ corresponds to $\mathrm{SNR}_r = -0.8556$ dB while $P_b = 0.01$ corresponds to $\mathrm{SNR}_r = 4.3232$ dB. Then from [43], the average fade duration in bit intervals is given by

$$\overline{\tau} = \frac{e^{\rho^2} - 1}{\rho f_D \sqrt{2\pi}} \tag{2.9}$$

where ρ is the received amplitude normalized by the RMS amplitude and is computed according to

$$\rho = \sqrt{\frac{\mathrm{SNR}_r}{\overline{\mathrm{SNR}}}}. \tag{2.10}$$

With the values $\overline{\mathrm{SNR}} = 10$ dB and $f_D = 10^{-5}$, the average fade duration for $P_e \geq 0.1$ is $\overline{\tau} = 11915$ bits and for $P_e \geq 0.01$ is $\overline{\tau} = 23832$ bits. Clearly the fade duration depends on the fade margin designed into the system (i.e., how much the channel error rate can increase before decoding failures become significant).

2.3.3 Multiple descriptions and the packet erasure channel

The packet erasure channel is different from the previous channels because lost data simply never reaches the receiver. In the previous channel models, data is lost due to excessive errors which cause channel decoding failures, but the data always reaches the receiver. The well known multiple description (MD) problem from information theory is typically focused on the special case of transmitting source information across two channels, each of which either carries the data without error or completely fails [44, 45]. When both channels are functioning the resulting reconstruction distortion is referred to as the *central channel* distortion, and the distortion resulting from the reconstruction with only one channel is referred to as the *side channel* distortion. The goal of the MD problem is typically posed as minimizing the central channel distortion subject to upper bounds on the side channel distortions. Some of the techniques used for traditional MD coding can be applied to the problem of coding for packet erasure channels.

The packet erasure channel can be considered to be a generalization of the MD channel model where there is an arbitrary number of channels, from which any subset of channels may fail. This model applies to transmission over a wireline network where packets are lost because of misrouting, queue overflow, or excessive delay. In the simplest model, packets are erased independently with a probability given by the channel erasure rate. More sophisticated models consider the bursty characteristic of erasures during heavy network congestion by introducing a state model analogous to the extension from the BSC to the Gilbert-Elliot channel.

2.3.4 Combined packet erasure and fading channel

In the channel models described so far, losses were either due to bit errors or dropped data never reaching the receiver. A natural extension is a channel with characteristics of both. Figure 2.7 shows a situation where this hybrid channel would occur. Data transmitted from servers on the wireline network to mobile receivers via a wireless connection would experience both erasure and bit error channels.

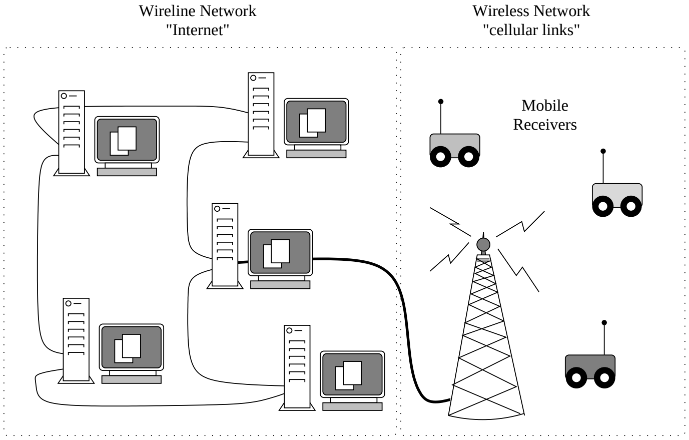


Figure 2.7 Combined packet erasure and fading channel.

CHAPTER 3

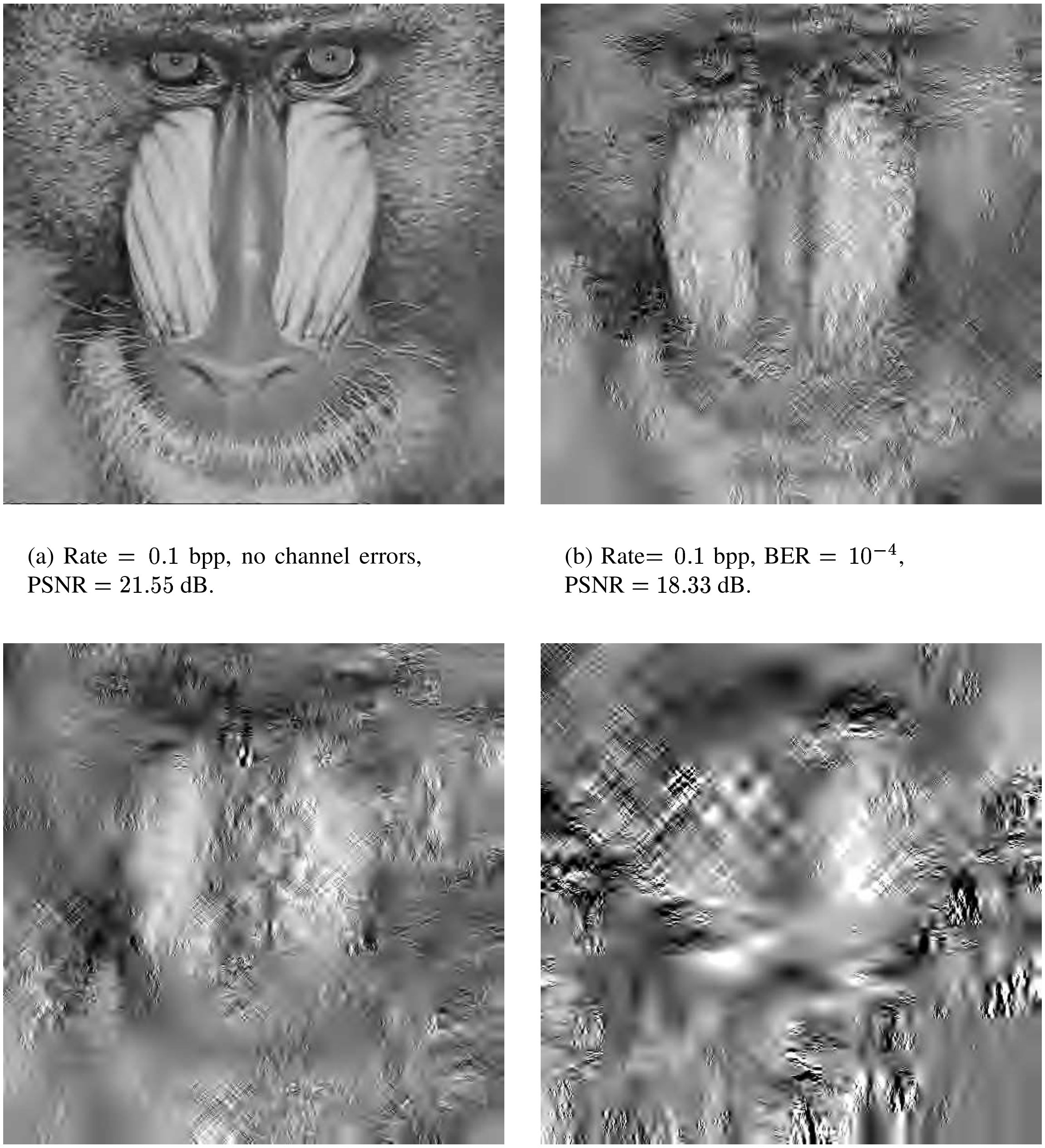
IMAGE CODING FOR MEMORYLESS CHANNELS

3.1 Introduction

The zerotree wavelet coders EZW and SPIHT provide excellent image compression performance with reasonable complexity. Also since they are both embedded coders, they produce bit streams with a high degree of progressivity. A significant problem when transmitting over noisy channels is that these algorithms are highly sensitive to bit errors. After even a single bit error, it is highly probable that all remaining bits will not be interpreted correctly. The effect of an error on decoded quality depends on its location in the bit stream since later bits have less impact on quality. To demonstrate the sensitivity of the SPIHT coder, Figures 3.1(a) - 3.1(d) show decoded images after transmission over a BSC of various rates. Clearly, the higher error rates result in a total collapse of image quality.

This chapter presents a channel coding method to protect the SPIHT image coder for transmission over a memoryless channel. The code has very flexible parameters and reasonable complexity. The combined system is shown to have better performance than previously published results. The results in this chapter related to the RCPC/CRC coding method are from [19, 28].

The problem of parameter selection (e.g., channel code rate and block length) is also discussed in this chapter. A natural approach is to consider UEP channel coding of the source information since it is not of equal importance to the reconstructed quality. In practice the gains associated with an optimal rate schedule are fairly small as discussed in Section 3.2.3.



(a) Rate $=$ 0.1 bpp, no channel errors, PSNR = 21.55 dB.

(b) Rate= 0.1 bpp, BER $=10^{-4}$, PSNR = 18.33 dB.

(c) Rate $=$ 0.1 bpp, BER $=10^{-3}$, PSNR = 16.96 dB.

(d) Rate $=$ 0.1 bpp, BER $=10^{-2}$, PSNR = 13.69 dB.

Figure 3.1 Demonstration of bit error sensitivity of the SPIHT (without arithmetic coding) coder using the 512×512 image Baboon.

For recent high performance channel codes like turbo codes and low-density parity check (LDPC) codes, there is a strong dependency between the code block length and the error protection capability. This property adds an interesting dimension to the problem of parameter selection when one of the design goals is to have the image refine as quickly as possible (i.e., a high degree of progressivity) in addition to being resilient to channel noise. The problem of block length and rate selection for progressive image coding is covered in Section 3.3, and the results in this section are from [35].

3.2 A Concatenated Channel Code Using RCPC and CRC Component Codes

One powerful method of error control coding is to use a concatenated code consisting of a Reed-Solomon outer code followed by a convolutional inner code (e.g., [46]). This approach, for example, was used in [47] for transmitting JPEG images across noisy channels. It was also used in [17, 18] for transmission across a Gaussian channel, although the implementation complexity appears extremely large in this case. The results in [17, 18] are also difficult to compare since it is not clear whether soft-decision decoding was used and some overhead bits seem to have been neglected in the overall bit rate computation. We adopt a related concatenated coding scheme with somewhat more flexibility and lower complexity to protect the output of the SPIHT coder.

The main idea is to partition the output bit stream from the SPIHT image coder into consecutive blocks of length N (typically we use $N = 200$ although its value is completely flexible). Then a collection of c checksum bits are derived based only on these N bits (we use $c = 16$). Finally m zero bits, where m is the memory size of the convolutional coder, are added to the end to flush the memory and terminate the decoding trellis at the zero state. The resulting block of $N + c + m$ bits is then passed through a rate r rate-compatible punctured convolutional (RCPC) coder [48].

The fact that the outer code is a cyclic redundancy code (CRC) used for error detection has the advantages of extremely low computational complexity and great flexibility in selecting block lengths (block lengths are unconstrained, in contrast to Reed-Solomon block lengths). The resulting bit stream is transmitted across a binary symmetric channel with bit error probability ϵ and then is decoded.

The decoder consists of a Viterbi decoder with the added feature that the “best path” chosen is the path with the lowest path metric that also satisfies the checksum equations. This additional constraint eliminates certain paths from consideration. In fact, whenever an undetected error would occur in the ordinary Viterbi decoder without the checksum bits, the correct path through the trellis is usually the one with the second lowest path metric. When the check bits indicate an error in the block, the decoder usually fixes it by finding the path with the next lowest metric. Systems of this type were analyzed in [49].

3.2.1 Decoding complexity

The complexity of the proposed channel decoder is quite reasonable, usually requiring little computation beyond that of the usual Viterbi decoding algorithm for the convolutional code. In addition, since punctured convolutional codes are used in this system, the trellis decoding is simplified because the trellis is that of a $1/N$ rate code for all the punctured rates. For a code of rate K/N this translates into a computation savings by a factor of $2^{K-1}/K$ per decoded bit.

In order to easily search for other likely trellis paths if necessary, some additional storage is required over the normal Viterbi algorithm. Usually only the survivor path and its metric must be stored for each state at the current trellis stage, as the decoding algorithm progresses. In this case, we store all paths and their metrics for each state and each trellis stage. The storage requirement amounts to two paths and metrics at each state and each stage of the trellis because the codes are punctured rate $1/N$ codes. For the packets of length 200 bits and a convolutional code with memory 6, the resulting memory requirement is about 40 Kbytes, which is reasonable for an image application.

Each candidate trellis path is checked by computing a 16-bit CRC. The CRC polynomial was selected from those listed in [50] and [51] based on the number of information bits in a packet. For example, with 200 information bits in a packet the selected CRC polynomial was $X^{16}+X^{14}+X^{12}+X^{11}+X^{8}+X^{5}+X^{4}+X^{2}+1$. Computing the check bits is very simple, only requiring bit shift and XOR operations, and table lookups can be used to trade off some of the computation with space requirements. For a 200-bit path, the byte-based algorithm requires about 50 XORs, 25 byte shifts, and 25 lookup operations. Typically the CRC only needs to be computed for the first candidate, but in the $0.3 - 3.0\%$ (which means at most about 3-4 packets of a rate 1.0 bpp image) of cases that require further paths to be checked, the correct path is among the top 10 candidates $98+\%$ of the time. Figure 3.2 shows a typical distribution of path search depths given that the output of the Viterbi algorithm did not satisfy the CRC check.

The search for alternate candidate paths is accomplished using a tree-trellis search algorithm. The same algorithm is used in speech recognition with hidden Markov models [52]. First, the maximum number of candidates M that will be decoded is selected since this determines storage requirements of a stack which will contain the M most likely paths as the search runs. The depth of the stack is selected to provide a good compromise among decoding time, probability of locating the correct path, and the probability of undetected errors due to an incorrect candidate path satisfying the CRC checks. In general, after determining the n^{th} best path, the next best path is determined by searching backwards in the trellis along the n^{th} best path starting from the trellis stage where it first branches from a higher ranking path. The search amounts to comparing the metric of a new path, whose initial segment consists of the path eliminated at that stage in the Viterbi algorithm and merges with the path in question thereafter, to the metrics stored on the stack. Since the metric and path values are stored as the Viterbi algorithm runs, the amount of computation is small. If the metric is low enough, the path is placed on the stack at the appropriate location according to the metric value. After searching to the beginning of the n^{th} path, the $(n+1)^{\mathrm{st}}$ will be stored at position $n+1$ on the stack. Also, we note that the search is terminated as soon as a path satisfies the CRC checks, so usually very few candidate paths need to be determined as indicated by the distribution in Figure 3.2.

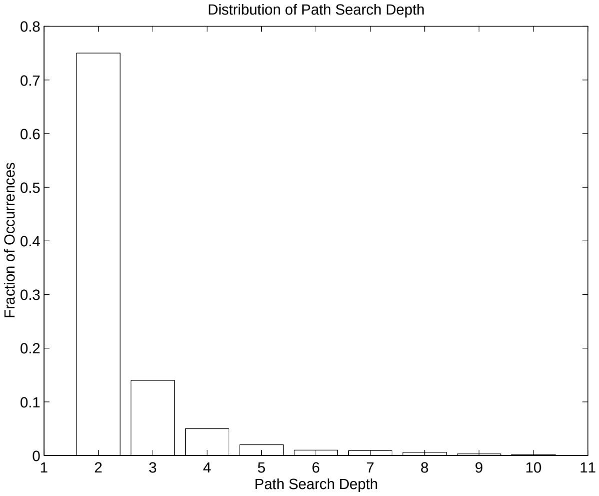


Figure 3.2 Distribution of path search depths given that the Viterbi decoder output fails the CRC check.

3.2.2 Performance results

The system was tested on two 512×512 images, the standard "Lena" and "Goldhill" images from the USC database. These particular test images were chosen to allow for some comparison with existing techniques. Each image was coded at bit error probabilities of $\epsilon = 10^{-1}$, $\epsilon = 10^{-2}$, and $\epsilon = 10^{-3}$, and at transmission rates ranging from 0 bpp up to 1 bpp, in increments of 0.05 bpp. All the RCPC codes used in testing were selected from tables in [48] or based on convolutional codes listed in [46]. In particular, a rate $2/7$ memory 6 (punctured rate $1/4$) code was used on the $\epsilon = 10^{-1}$ channel, a rate $2/3$ memory 6 (punctured rate $1/3$) code was used on the $\epsilon = 10^{-2}$ channel, and a rate $8/9$ memory 6 (punctured rate $1/3$) code was used on the $\epsilon = 10^{-3}$ channel. This scheme retains the progressive nature of the underlying source coder allowing almost a continuum of transmission rates. Each increment in the transmission rate can be as small as about 0.001 bpp which corresponds to the packet size we selected containing 200 source bits. Many of the previously reported image coding systems for noisy channels are not progressive.

For each image and each bit error probability, many thousands of independent trials were simulated on a computer for the various transmission rates, which translates into millions of packets. In these tests, the path search depth was limited to 100 candidate paths, and if none of these 100 paths satisfied the CRC check, then decoding for that image was stopped at that packet giving incomplete decoding. The effect of incomplete decoding is often acceptable since the output is simply the image decoded at a lower source rate. Figure 3.3 shows the trade-off between typical image PSNR and probability of incomplete image decoding for inner RCPC code rates of $\{1/4, 8/31, 4/15, 8/29, 2/7, 8/27, 4/13, 8/25, 1/3\}$. These results are for the Lena image coded at total (i.e., source plus channel coding) rates 1.0, 0.5, and 0.25 bpp using a channel with BER = 10^{-1}. For our purposes, the inner RCPC codes were selected so that the probability of incomplete image decoding was below 0.01 for the highest transmission rate of interest (1.0 bpp) for each channel BER. The highest incidence rate of incomplete image decoding in our tests was actually around 0.005. Higher acceptable incidence rates for incomplete decoding would allow higher typical PSNR values while more stringent incidence rates would lower typical PSNR values.

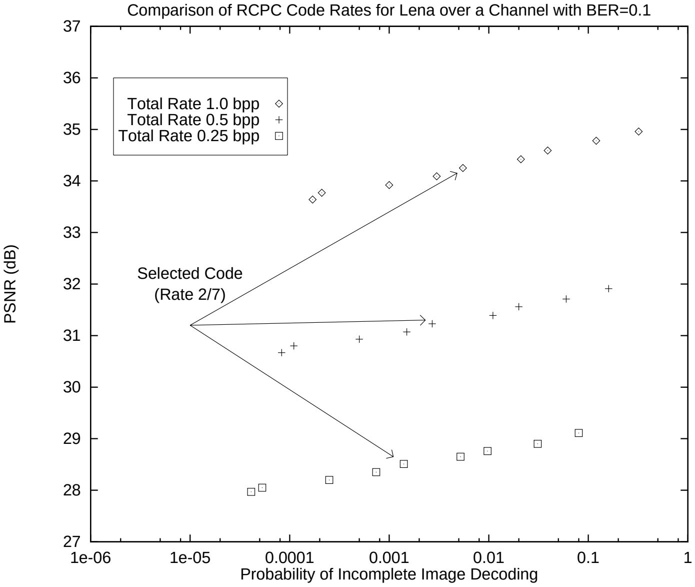
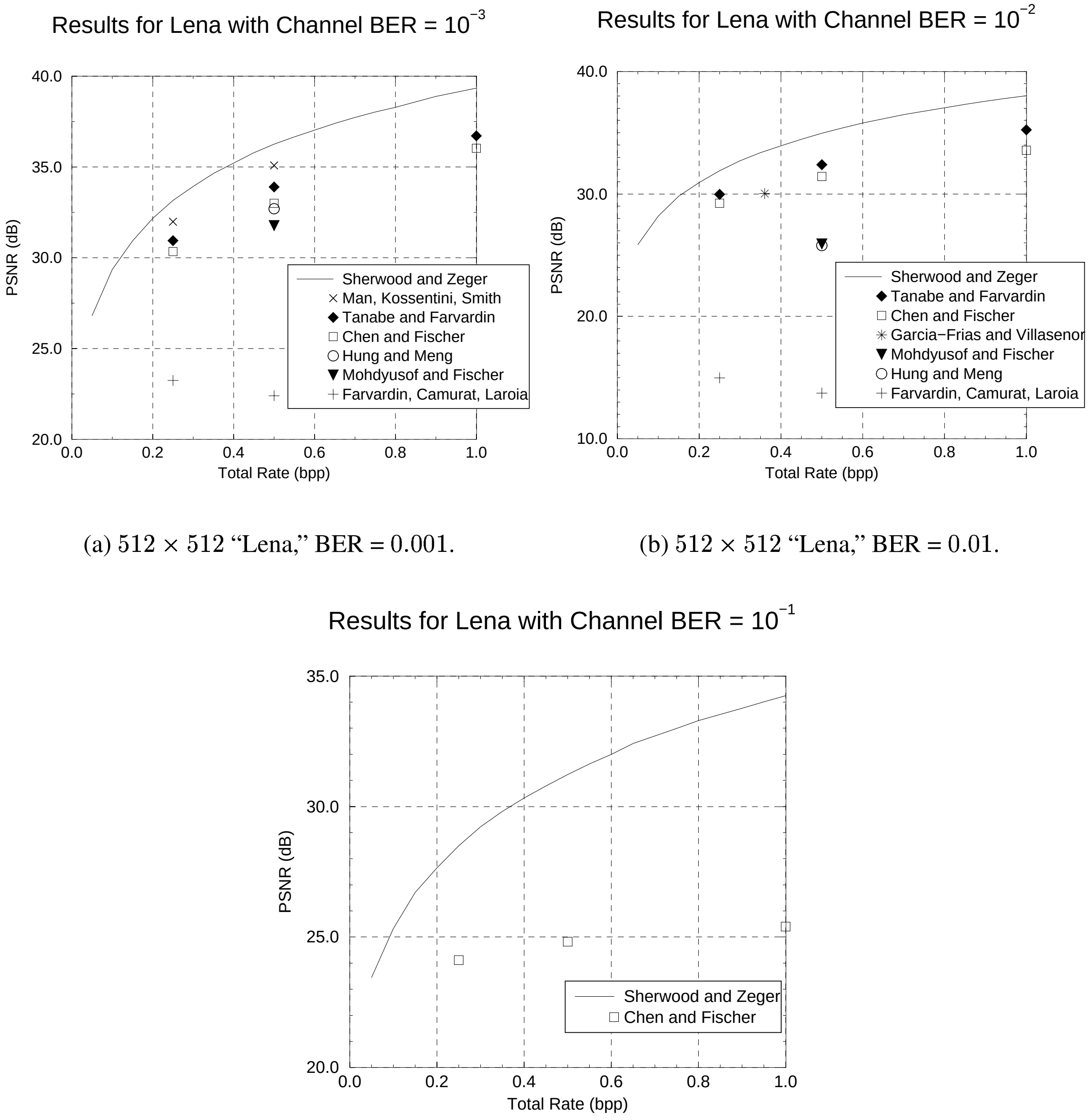


Figure 3.3 Trade-off between the likelihood of incomplete decoding and image quality given complete decoding, for convolutional code rates from $1/4$ to $1/3$.

The curves in Figures 3.4(c)-3.5(a) show the resulting PSNR of the cascaded source coding and channel coding system as a function of the overall transmission rate across the channel. Figures 3.4(c), 3.4(b), and 3.4(a) are for Lena, and Figures 3.5(c), 3.5(b), and 3.5(a) are for Goldhill. Each figure is for a fixed bit error probability ϵ on a binary symmetric channel. The

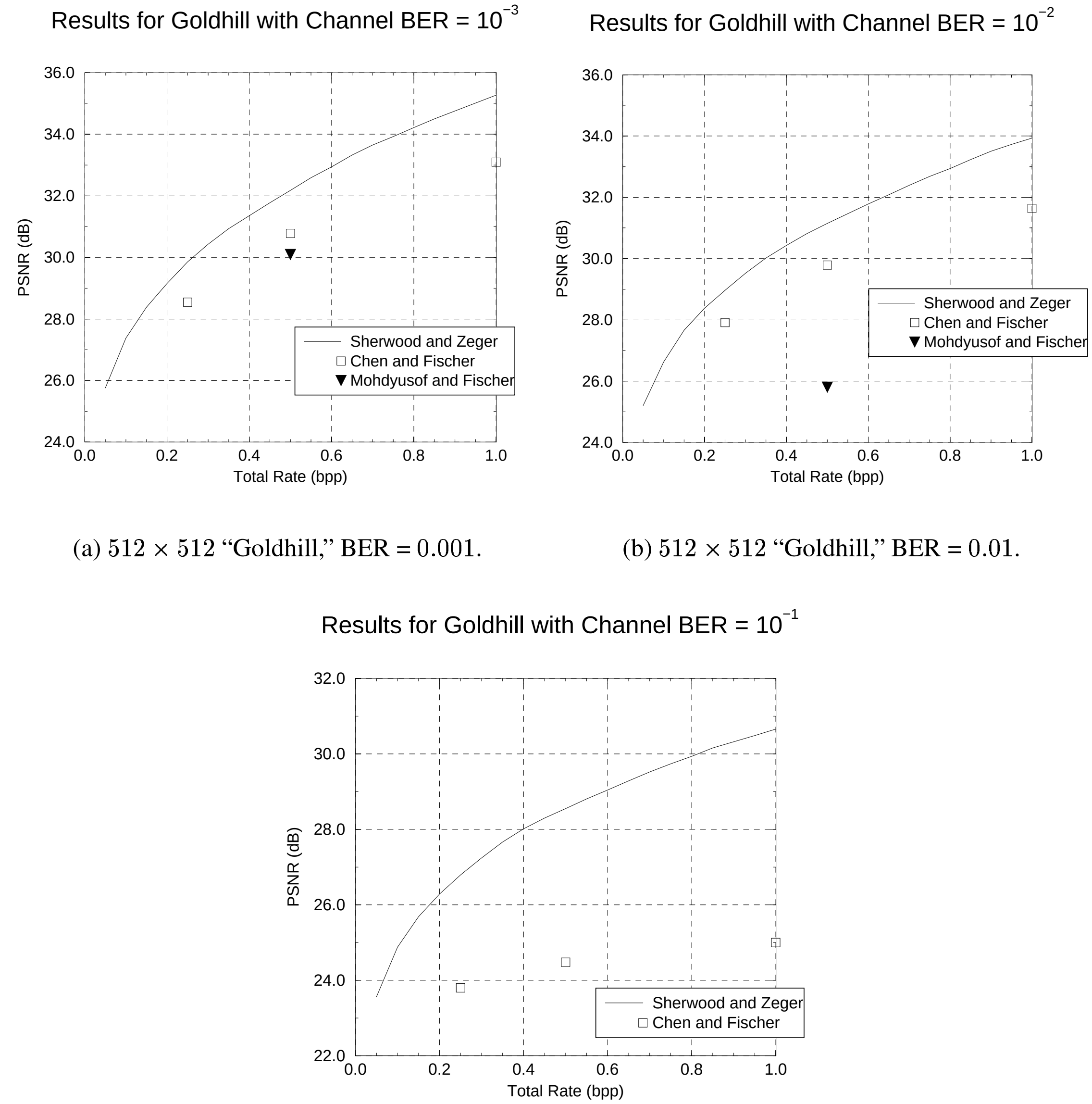


(a) 512×512 “Lena,” BER = 0.001.

(b) 512×512 “Lena,” BER = 0.01.

(c) 512×512 “Lena,” BER = 0.1.

Figure 3.4 Comparison of the performance of the RCPC/CRC + SPIHT coder with previously published results for image Lena transmission over a BSC.



(a) 512 × 512 “Goldhill,” BER = 0.001.

(b) 512 × 512 “Goldhill,” BER = 0.01.

(c) 512 × 512 “Goldhill,” BER = 0.1.

Figure 3.5 Comparison of the performance of the RCPC/CRC + SPIHT coder with previously published results for image Goldhill transmission over a BSC.

other points in the plots represent the performances of the best known image codes for noisy channels, prior to the proposed system. These include those of Tanabe and Farvardin [53], Chen and Fischer [54], as well as those in [55, 56, 57]. Numerous other results exist in the literature, but all of them appear to be inferior to the results reported in [53, 54], or else do not provide results for the test images we considered.

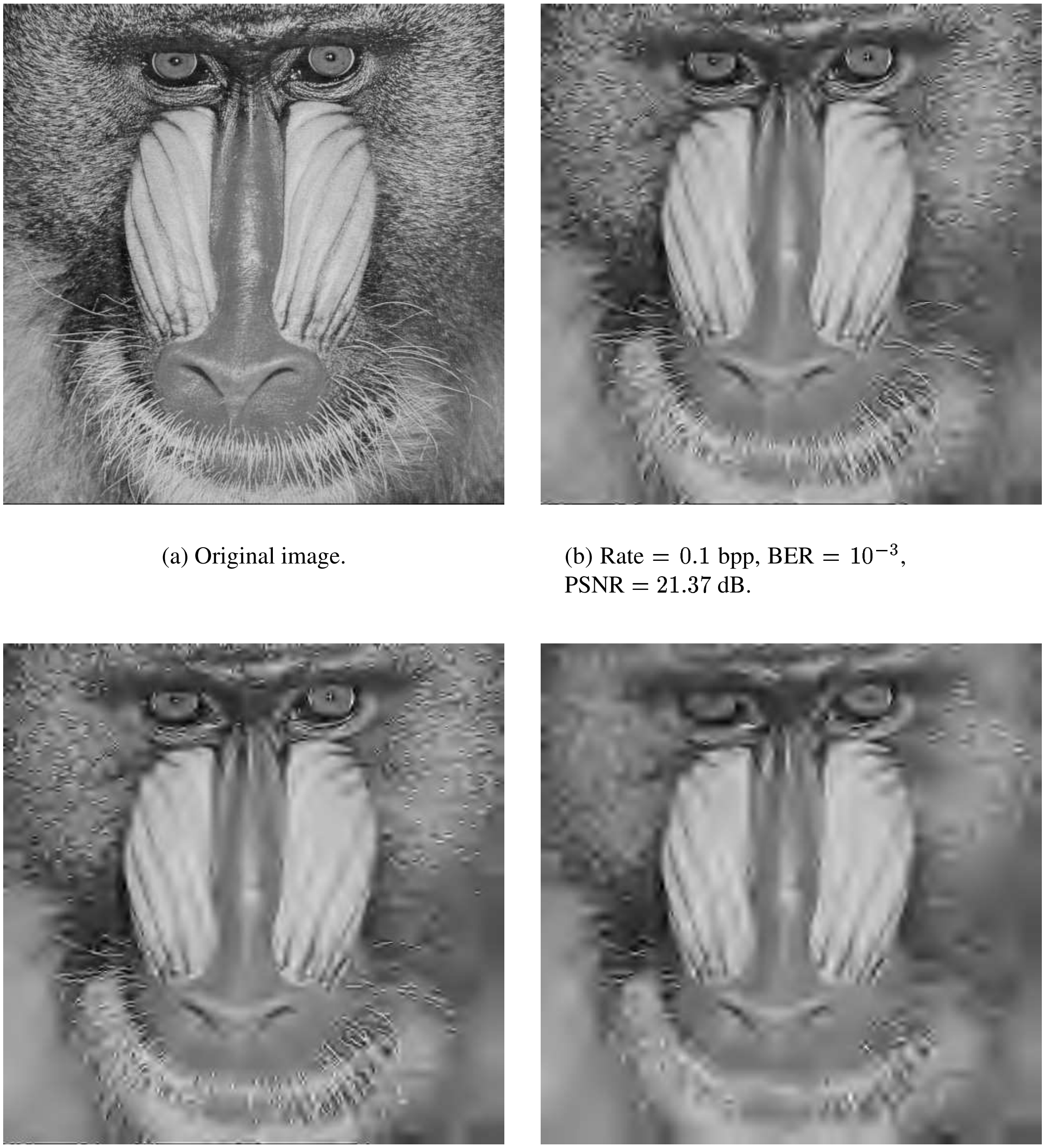
It can be seen from the Lena graphs at a transmission rate of 1 bpp that our proposed system improves the performance over the other reported SNRs by at least 2.6 dB, 2.8 dB, and 8.9 dB, for bit error probabilities of 10^{-3}, 10^{-2}, and 10^{-1}, respectively. At a transmission rate of 0.5 bpp, the gains for these three channels are 2.3 dB, 2.6 dB, and 6.4 dB, respectively. At a transmission rate of 0.25 bpp, the gains for these three channels are 2.2 dB, 1.9 dB, and 4.4 dB, respectively. For the Goldhill image at 1 bpp, the gains for these three channels are 2.2 dB, 2.3 dB, and 5.7 dB, respectively. At a transmission rate of 0.5 bpp, the gains for these three channels are 1.4 dB, 1.4 dB, and 4.1 dB, respectively. At a transmission rate of 0.25 bpp, the gains for these three channels are 1.3 dB, 1.1 dB, and 3.0 dB, respectively.

Example decoded images along with the original are provided in Figures 3.6(a) - 3.6(d) to demonstrate the visual quality of images transmitted over a BSC with error rates from 0.001 - 0.1.

3.2.3 Parameter optimization and other extensions

Since the output of the embedded image coder does not have uniform importance to the decoded image quality, applying UEP channel coding seems to be a likely way to improve performance. For the RCPC/CRC + SPIHT coder presented in this section, the error protection is modified by selecting different channel code rates for each packet. The output of the source coder generally has nonincreasing importance so the optimal channel code schedules tend to have nondecreasing channel code rate profiles.

The graph in Figure 3.3 provides some indication of the difficulty of adjusting the channel code rate. In this case, the probability of incomplete decoding covers more than three orders of magnitude while the range of PSNR values only spans 1 dB. Even though the probability of incomplete decoding is more stringent than the usual criterion of expected MSE, it provides



(a) Original image.

(b) Rate $= 0.1$ bpp, BER $= 10^{-3}$, PSNR = 21.37 dB.

(c) Rate $= 0.1$ bpp, BER $= 10^{-2}$, PSNR = 21.01 dB.

(d) Rate $= 0.1$ bpp, BER $= 10^{-1}$, PSNR = 20.10 dB.

Figure 3.6 Example images demonstrating the performance of the RCPC/CRC + SPIHT coder over the BSC.

an indication of how likely there are to be uncorrected errors leading to increased distortion. Similarly, the curves in Figure 3.7 show the rapid transition of the mean and median decoded PSNR as the error rate exceeds the design error rate. The combination of the serial decoding requirement (see section 2.2) of the source coder and the threshold effect of the channel codes [21] significantly constrains the amount by which the channel code rate can be increased for later packets.

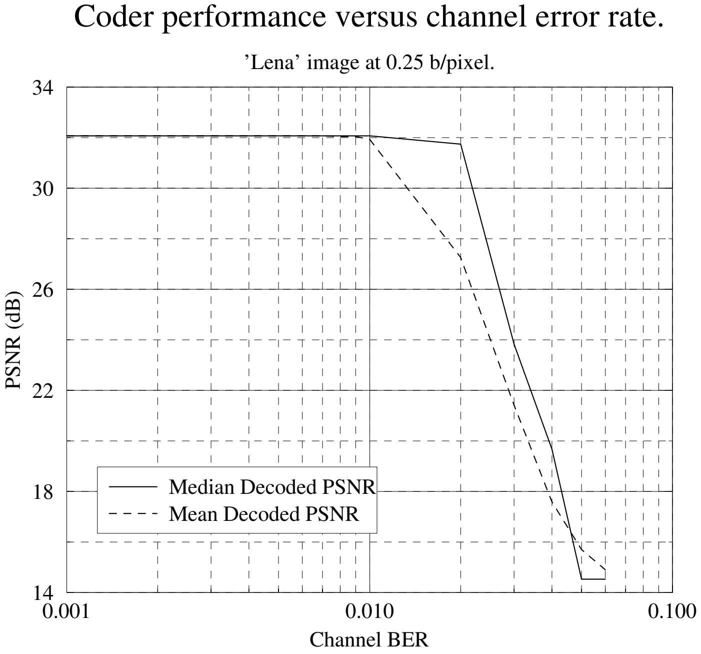


Figure 3.7 Variation of coder performance with error rate.

The problem of optimizing the channel code rate to minimize the expected distortion under a total transmission rate constraint was considered in [14] and [58]. In both cases, the conclusion was that very little can be gained over the best fixed rate channel coding scheme.

The optimal channel code rate for a particular packet depends on the total transmission rate. Since the bit stream is truncated at the first uncorrected error, an error at packet n is more important if there are more packets to follow. The embedded structure of the RCPC channel codes can be used to schedule the transmission of the redundancy bits for each packet so that the optimal channel code rate is achieved at several subrates up to the total transmission rate. This approach was used in [59] to optimize the number of available source bits at the decoder.

The next section discusses an extension of that work where both the block length and rate of the channel code are optimized.

The embedded structure of the RCPC channel codes can also be used in an ARQ system if a feedback channel to the transmitter is available. In [60, 61] the problem of designing the rate schedule was considered including the cost of feedback channel usage. In this system, additional redundancy bits are transmitted in steps until the packet is successfully decoded, as indicated by the receiver feedback, or until all bits for that packet have been transmitted. In this case, significant gains can be achieved over the systems without feedback.

3.3 Block Length and Rate Optimization for Progressive Coding

Systems composed of embedded wavelet-based image coders followed by channel coding result in some of the best systems currently known for transmitting images over certain noisy channels [28, 34, 59, 62]. Selecting the proper channel code rate is important so that bits are not wasted on unnecessary redundancy, while keeping the error probability sufficiently low. Adjusting the block length can further improve the performance of certain channel codes.

It is known from information theory that long block lengths can achieve good performance at rates close to capacity, and many known codes exhibit improved performance as the block length is increased. Examples include turbo codes, low-density parity check codes, and Reed-Solomon codes on erasure channels. The results presented in [62] for image transmission on a packet erasure channel demonstrate that excellent performance at a target transmission rate can be achieved by using the maximum possible channel codeword block length. For an erasure rate ϵ and channel code rate $r \leq 1 - \epsilon$, a useful upper bound on the block error probability is [4, p.531]

$$P_E(r, \epsilon, N) \leq 2^{-ND_2(1-r||\epsilon)}$$

where N is the block length and $D_2(a||b) = a \log_2(a/b) + (1-a) \log_2((1-a)/(1-b))$ is the information divergence. This upper bound shows the advantage of long block lengths (i.e., large N) for rates below capacity.

If decoding at multiple rates is not important, then using long block length channel codes provides the best performance in many cases. One penalty associated with long block length codes is increased decoding complexity. For progressive coding, there is the additional penalty of decoding delay. Since the decoder typically must wait until the entire block has been received and decoded before reliable source bits are available, the distortion vs. rate curve flattens out over the duration of a block length and decreases only at the end of the block. The result is that the curve is staircase shaped, where the size of the steps depends on the channel code block length as illustrated in Figure 3.8. The trade-off is that long block lengths improve error protection but they also decrease progressivity.

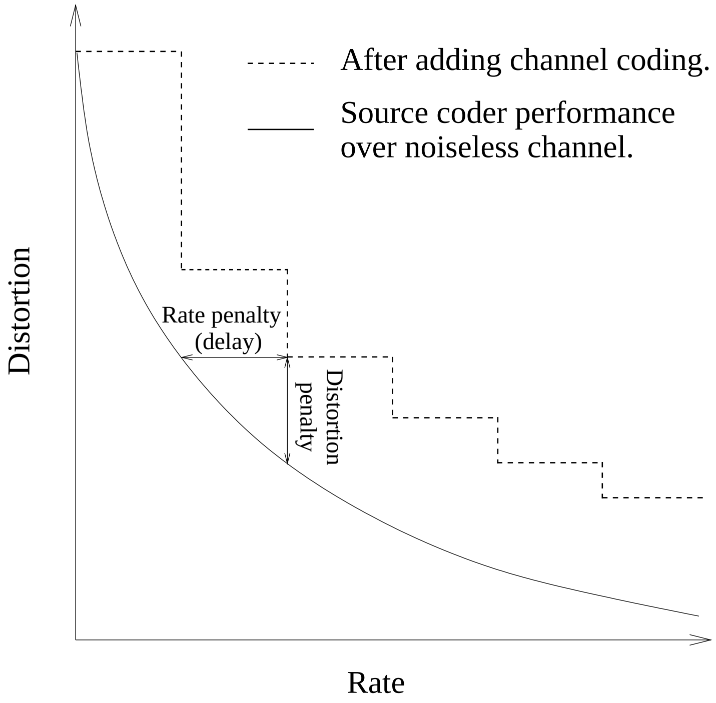

After adding channel coding.
Source coder performance over noiseless channel.
Distortion
Rate penalty (delay)
Distortion penalty
Rate

Figure 3.8 Distortion vs. rate with and without channel coding.

Previous work [14, 58, 59] has considered methods of selecting the best channel code rate for image transmission on noisy channels. A gradient-based technique was used in [58] to determine the best rate allocation between source and channel codes for a given transmission rate. In [14], Lagrangian methods were used to select the channel code rate schedule for fixed-length information blocks to minimize distortion at a final target rate. Although progressive source coders were used, progressive performance was not considered in [14, 58].

In [59], optimization was also performed for a specific transmission rate, but the optimization criterion and rate-compatible properties of the channel codes allowed optimal transmission at many lower rates. A dynamic programming approach was presented for determining channel code rate schedules for fixed-length information blocks using optimization criteria based on MSE, PSNR, and number of available source bits. The optimization based on number of available source bits was suggested as the best approach since it reduces complexity, eliminates the need to transmit the rate schedule, and allows a single rate schedule to incorporate the optimized rate schedules for many lower transmission rates as prefixes by using the rate-compatible properties of the codes considered.

We extend the work in [59] to consider the optimization of block length and channel code rate, and we use a more general performance function to characterize progressive performance. Specifically, this section includes the following contributions:

- A general performance measure is presented for evaluation of progressive performance.
- A dynamic programming solution is presented for optimizing block length and rate.
- Results for a number of error rates compare optimizations based on PSNR and optimizations based on available source bits with and without fixed information block sizes.
- Results for Reed-Solomon codes on erasure channels, and for turbo codes on bit error channels, demonstrate the importance of optimizing block lengths.

3.3.1 A performance measure for progressive transmission

In order to optimize the channel code parameters, it is necessary to define a performance measure for progressive transmission. In image coding, typically the expected MSE or PSNR at a target transmission rate is used. For progressive coding, the desire is generally to maximize the expected performance at all rates up to the target transmission rate. We propose a family of performance measures given by

$$\mu = \int_0^{R_T} W(x)F(x)dx \tag{3.1}$$

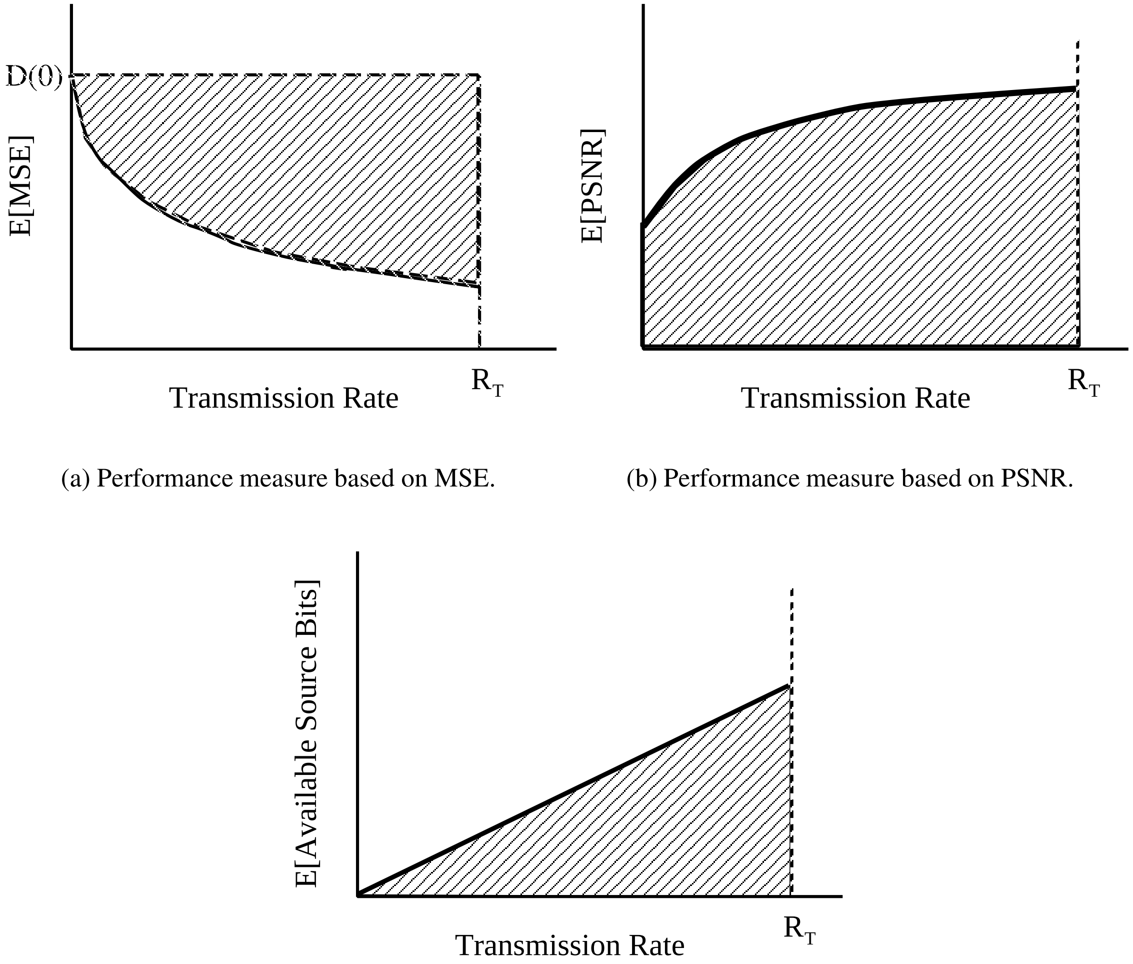
where R_T is the terminal transmission rate, $F(x)$ is the expected performance for a given channel at transmission rate x, and $W(x)$ is a nonnegative, real-valued "weighting function" which allows unequal emphasis for different rates. The performance function $F(x)$ is large for good performances and small for poor performances, in contrast to the more typical use of a "distortion" function. The performance measure μ includes, as a special case, the conventional approach of measuring performance at a single rate (or possibly a small set of rates), by using impulse functions as part of the weighting function (e.g., $W(x) = \delta(x - r_1) + \delta(x - r_2) + \dots$ where $\delta(x)$ is the Dirac delta function).

Possible performance measures include the expected MSE, expected PSNR, or number of available source bits as noted in [59]. The PSNR-based performance is closely tied to image quality but does not emphasize the large distortions at low rates as much as the MSE-based performance. Performance based on available source bits assumes all bits as equally important but it allows simplifications in the optimization and eliminates overhead information. In both of these cases, the goal (assuming $W(x) = 1$) is to maximize the area under $F(x)$. If we define $F(x) = D(0) - D(x)$, where $D(x)$ is the expected MSE at rate x, then the goal is also to maximize μ. In the context of dynamic programming algorithms discussed in the next section, μ is the reward. Figure 3.9 shows examples of these performance measures where the shaded region represents the area computed by the integral with $W(x) = 1$.

3.3.2 Optimization using dynamic programming

The optimal schedule of channel code parameters can be determined using dynamic programming where the goal is to maximize the reward based on the progressive performance measure (i.e., μ in (3.1)). The notation is based on that in [63].

Given a transmission rate constraint R_T, the goal of the optimization problem is to determine the *code schedule*, $\pi = \{(N_1, K_1), (N_2, K_2), \dots, (N_{M(\pi)}, K_{M(\pi)})\}$, that maximizes μ subject to $\sum_{i=1}^{M(\pi)} N_i \leq R_T$. Each component of the code schedule contains a block length parameter N_i and an information size parameter K_i. For a given system consisting of source coder, channel coder, and channel, each code schedule π determines an expected performance-rate function as in (3.1), denoted $F_\pi(x)$. Since the transmitted data consists of a sequence of



(a) Performance measure based on MSE.

(b) Performance measure based on PSNR.

(c) Performance measure based source bits.

Figure 3.9 Three possible progressive performance measures each equal to the area of the shaded region.

codewords, the overall reward will be based on the incremental reward associated with each codeword. The incremental reward of a codeword with parameters (N_i, K_i) under code schedule π is given by

$$\begin{aligned} r_\pi(N_i, K_i) = &\int_{\widetilde{R}_{i-1}}^{\widetilde{R}_i} W(x)(F_\pi(x) - F_\pi(\widetilde{R}_{i-1}))dx \\ &+ (F_\pi(\widetilde{R}_i) - F_\pi(\widetilde{R}_{i-1})) \int_{\widetilde{R}_i}^{R_T} W(x)dx \end{aligned} \tag{3.2}$$

where $\widetilde{R}_i = \sum_{j=1}^{i} N_j$ is the number of transmitted bits up to and including codeword i, and $\widetilde{R}_0 \equiv 0$. The first term of (3.2) is the incremental reward during transmission of the codeword while the second term is the accumulated incremental reward from the end of the codeword to the target rate. The integrals typically simplify to sums due to the discrete nature of the problem, and for certain performance measures such as the one based on available source bits, the integrand simplifies due to the constant reward for each bit.

The underlying assumption used in combining the incremental rewards from multiple codewords into the overall reward is the serial decoding requirement of the source decoder. A source decoder is said to have a *serial decoding requirement* if a substantial portion of the bit stream must be decoded in a sequential and uninterrupted fashion for correct interpretation. Many embedded coders, especially those employing adaptive entropy coding, have this property. The effect is that decoding terminates at the first uncorrectable (and detected) error so no reward is accumulated from subsequent correctly received codewords. A "reward-to-go" function associated with codeword i is defined as

$$J_i(R_T, \pi) = r_\pi(N_i, K_i) + (1 - P_E(N_i, K_i))J_{i+1}(R_T, \pi) \tag{3.3}$$

where $P_E(N_i, K_i)$ is the probability of block decoding error for codeword i. The $(1-P_E(N_i, K_i))$ term multiplying J_{i+1} is the result of the serial decoding requirement. The goal of the optimization problem is to find the optimal code schedule π^* that satisfies

$$\pi^* = \arg\max_{\pi} J_1(R_T, \pi) \tag{3.4}$$

with the condition that $J_i(R_T, \pi) = 0$ for some sufficiently large i under all admissible policies due to the transmission rate constraint R_T.

In the following sections, this general framework is applied to specific channel conditions and performance results are provided for the resulting rate schedules.

3.3.3 Progressive transmission over erasure channels

This section considers the problem of transmitting an image over a packet erasure channel with erasure rate ϵ. The packets are assumed to be fixed length, and this length is a parameter of the optimization. The information is protected using erasure correcting codes across the packets (similar to that found in [34, 62]), such as Reed-Solomon codes, which allow for error-free transmission of K information packets out of a block of length N as long as any K packets are received.

The codes are systematic with the information packets transmitted first so these packets are available immediately to the source decoder until the first erasure. After the first erasure, further source decoding is delayed until K packets of the current codeword have been received, at which point erasure correction is possible. Assuming the weighting function from (3.2) is unity (i.e., $W(x) = 1$) for simplicity, the incremental reward is derived below for this case.

For the first K_i packets of the codeword, the incremental reward is based on the expected number of packets before the first erasure. Let $f(x)$ be the performance of the source coder at transmission rate x on a noiseless channel with no channel coding (e.g., number of available source packets or bits (where $f(x) = x$), PSNR, etc). Note that the rate parameter is in units of packets rather than bits, for simplicity, and the conversion to bits involves a constant multiplicative factor due to the constant packet size. The expected performance over this range of transmitted packets under code schedule π is given by

$$F_\pi(M + \widetilde{R}_{i-1}) = \sum_{j=0}^{M-1} f(\widetilde{K}_i + j)(1-\epsilon)^j \epsilon + f(\widetilde{K}_i + M)(1-\epsilon)^M \qquad 1 \leq M \leq K_i \quad (3.5)$$

where $\widetilde{K}_i = \sum_{j=1}^{i} K_i$ is the number of cumulative information packets and ϵ is the packet erasure rate.

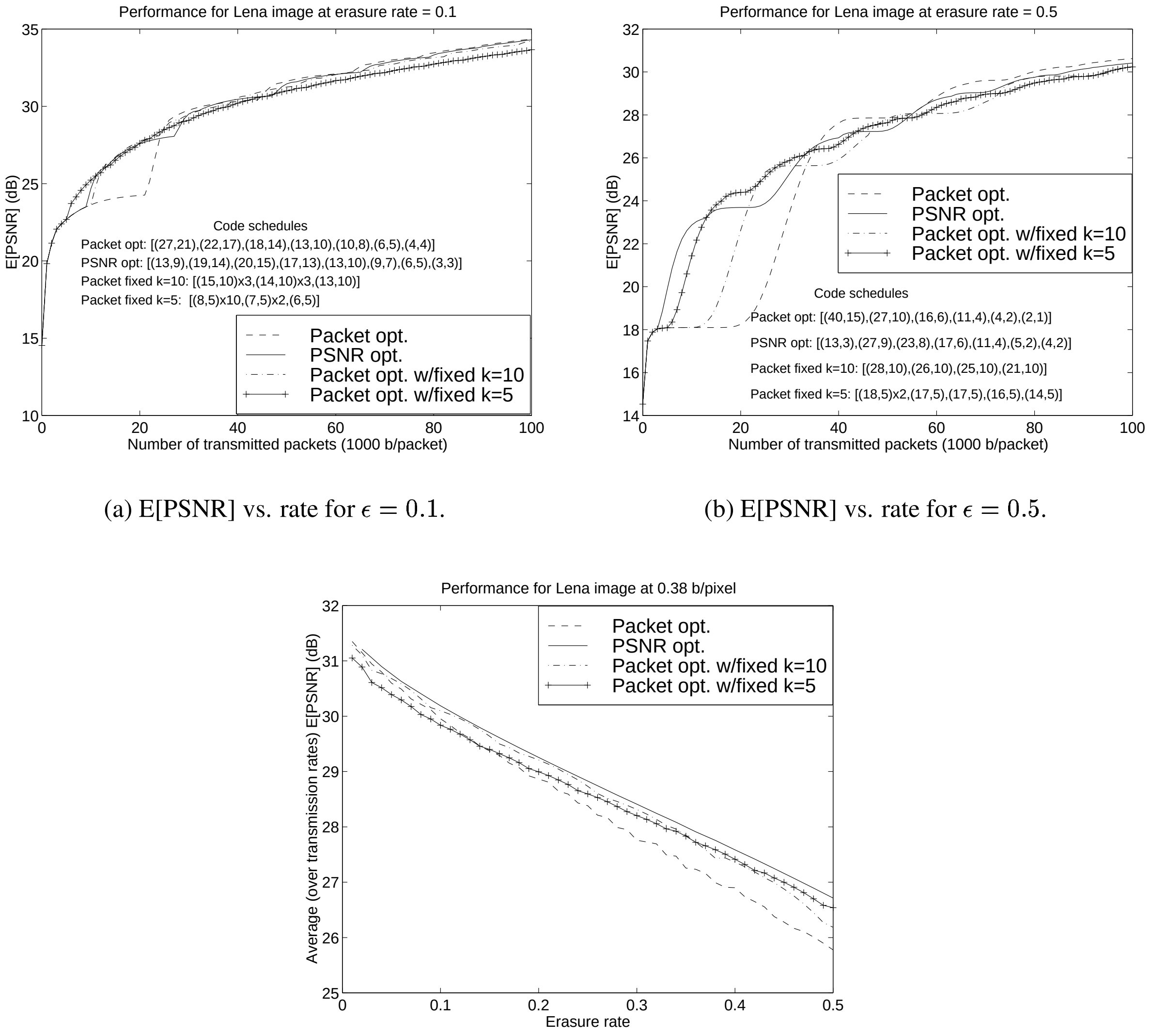
During the transmission of the $N_i - K_i$ parity packets, the performance only changes at the point where a total of K_i packets have been received for that codeword (at which point erasure correction is possible). The expected performance over this range is given by

$$F_\pi(M + \widetilde{R}_{i-1}) = \sum_{j=0}^{K_i-1} P_{me}(M-j-1, M-K_i) f(\widetilde{K}_i + j)(1-\epsilon)^j \epsilon + (1 - P_E(M, K_i)) f(\widetilde{K}_i + K_i) \qquad K_i + 1 \leq M \leq N_i \quad (3.6)$$

where $P_{me}(x,y) = \sum_{j=y}^{x} \binom{x}{j} (1-\epsilon)^{x-j} \epsilon^j$ is the multiple erasure probability (i.e., the probability of at least y erasures in x packets) and $P_E(M, K_i) = P_{me}(M, M-K_i+1)$ is the block decoding error probability of a code with parameters (M, K_i).

Optimization based on number of available source bits is attractive due to the reduced complexity and the elimination of the need to transmit the code schedule (i.e., the code schedule is image independent and can be computed by the receiver). However, performance measures based on MSE and PSNR are better indicators of image quality, so experiments were performed to compare the relative performance of the optimization methods. The overhead required for transmission of the code schedule in the case of the MSE or PSNR performance measures is assumed to be negligible so it is not included in the test results. This assumption is reasonable since the code schedule is typically fairly short (i.e., not many codewords) and can be compressed. Alternatively, the operational distortion-rate function is well modeled in practice by a function of the form $D(R) = \sum_{i=1}^{n} a_i e^{-\lambda_i R}$ with as few as two or three terms, as noted in [58]. This functional approximation could be used in determining the code schedule, and the function parameters could be transmitted instead of the actual schedule which would allow the receiver to compute the schedule.

Figure 3.10 shows the results of experiments using the SPIHT [10] image coder with arithmetic coding along with code schedules computed from performance measures based on PSNR and available source bits or equivalently packets. The resulting code schedules are evaluated in terms of expected PSNR for two different erasure rates in Figures 3.10(a) and 3.10(b). In addition, a method similar to that in [59], referred to as packet optimization with "fixed k," is applied to Reed-Solomon codes. Results are provided for code schedules with $k = 5$ and $k = 10$ information packets per codeword.



(a) E[PSNR] vs. rate for $\epsilon = 0.1$.

(b) E[PSNR] vs. rate for $\epsilon = 0.5$.

(c) Performance vs. erasure rate.

Figure 3.10 Performance results for the 512×512 Lena image transmitted over an erasure channel.

In the "fixed k" method, the information size of each codeword is fixed and the block length (equivalently rate) is selected, the performance measure is available source bits (packets), and the optimization is for the final transmission rate so $W(x) = \delta(x - R_T)$ in (3.2). The resulting code schedules from the optimization are monotonic nonincreasing in block length (or nondecreasing in channel code rate) due to the serial decoding requirement. Initial information packets are protected more heavily than later packets, even though every packet contains one unit of reward, since a packet can only be used if all earlier information packets are correctly received. The order of transmission of the codeword packets can be altered so that optimal (in terms of available source bits) code schedules are achieved at certain lower transmission rates as suggested in [59]. This progressive transmission is achieved by transmitting punctured versions of the codewords starting with the last codeword parameters in the schedule. Additional parity packets are sent as needed during transmission to achieve all the intermediate code rates specified in the schedule between the last codeword and the codeword in question. Optimal schedules in terms of available source packets are achieved for transmission rates which equal the cumulative sum of the block lengths starting from the last codeword and proceeding to the first codeword in the schedule. There are a few points to mention regarding the reordering for progressive transmission in [59]:

- The optimality is for the particular value of k and it corresponds to available source packets, not expected PSNR.
- The optimality is only guaranteed at certain transmission rates (equal to the cumulative sum of block lengths starting from the last codeword), so other transmission orders may be better at other intermediate transmission rates.
- The best order for performance at all intermediate transmission rates is dependent on the erasure rate, and it is generally better to transmit according to the natural schedule order (i.e., all packets of each codeword in sequence) under high erasure conditions.
- The transmission rates of optimality are not selectable, but are instead determined by the code schedule computed for the final transmission rate.

The results in Figures 3.10(a) and 3.10(b) show that the optimization based on available source packets and that based on expected PSNR achieve nearly the same expected PSNR at the final transmission rate. The code schedules are also listed in the graphs. Generally, the code schedules for the packet-based optimization consist of codewords with nonincreasing block lengths and nearly identical rates with the possible exception of the final codewords due to the transmission rate constraint. The PSNR-based optimization selects a shorter but lower-rate initial codeword to reduce the delay while still keeping the error probability low, and then the remainder of the schedule is similar to the packet-based schedule. As the first two graphs in Figure 3.10 demonstrate, the relative performance of the different methods depends on the erasure rate. Figure 3.10(c) shows the performance in terms of the expected PSNR averaged over all the intermediate transmission rates (essentially a normalized version of the area under the curve) over a range of erasure rates. The PSNR-based optimization has the best performance at all rates as expected, while the other curves cross at different erasure rates. Also the spread in relative performance increases with erasure rate.

3.3.4 Progressive transmission over bit error channels

By simply changing the incremental reward function, the same optimization techniques can be applied when coding for channels that introduce bit errors. For the erasure channel, information bits can be immediately used by the source decoder up to the first erasure since any data that arrives on this channel is known to be good. However, in this case, the information bits cannot be used before the entire codeword has been received since they are likely to contain errors which cannot be corrected until the remainder of the codeword arrives. The result is that the incremental reward of a codeword only has the second term from (3.2), and the performance curves have more of a staircase shape. The incremental reward for a codeword with parameters (N_i, K_i) under code schedule π is given by

$$r_\pi(N_i, K_i) = (1 - P_E(N_i, K_i))(f(\widetilde{K}_i + K_i) - f(\widetilde{K}_i))(R_T - \widetilde{R}_i). \tag{3.7}$$

For the block length optimization to be interesting, the channel code's error correction capability should have a significant dependence on block length. Therefore both turbo codes and low density parity check codes are considered in this section. Results for the RCPC/CRC codes from Section 3.2 are also provided for comparison.

Turbo codes, first introduced in 1993 [64], consist of either a parallel or serial concatenation of convolutional codes with a large interleaver before the input to one of the convolutional encoders. Decoding is done in an iterative fashion with soft information passed between the decoders for the constituent codes. Low density parity check (LDPC) codes, introduced in 1962 by Gallager [65], are linear block codes with very sparse parity check matrices. Typically, the codes are constructed so that each column has the same small weight (e.g., 3), all the rows have approximately equal weight, and the nonzero are otherwise chosen randomly within the first two constraints. Decoding is also done by an iterative algorithm. In fact, it has been shown [66] that the turbo decoding algorithm is an instance of Pearl's "Belief Propagation" algorithm used for the LDPC decoding. Recent results [67] have shown performance for LDPC codes to be quite close to that of turbo codes.

Regardless of which codes are used, it is important that decoding errors are detected, because the source decoder will lose synchronization and corrupt the image with high probability if its input contains errors. For the LDPC codes, the decoding algorithm is able to detect failures with high probability, so no further error detection is required. However, for turbo codes, an outer cyclic redundancy code (CRC) is used for error detection as well as providing a stopping criterion for the iterative turbo decoding. Also it is the block error probability that is important here and not the bit error probability so choice of turbo code interleavers becomes important for large block lengths. For this reason, s-random interleavers [68], which enforce a minimum spread between adjacent information bits after interleaving, were used because they result in better block error probabilities than random interleavers.

Experimental results showed that the performance difference between code schedules determined by PSNR-based measures and those based on available source bits were extremely close. Since the optimization based on available source bits is less complex and eliminates the overhead of transmitting the schedule, that method was used in results that follow. Also all decoding used hard decisions from the channel to allow easy comparison to an existing RCPC/CRC implementation.

Figure 3.11 shows the performance-rate curves comparing a schedule determined by a PSNR-based optimization, along with two fixed rate and block length turbo coding schemes. The turbo codes consisted of a family of punctured codes produced from a rate $1/3$ mother code. Note that using punctured codes allows the possibility of optimizing only the rate (i.e., the "fixed k" method mentioned in the previous section) and using the progressive transmission order, but recent results suggest that nonpunctured codes perform significantly better for the same code parameters [69]. The optimized code schedule has a similar structure to those computed for the erasure channel, where the code rate is essentially constant and the block length decreases monotonically. As can be seen in the graph, the optimized schedule strikes a balance between performance at low and high transmission rates relative to the fixed coding methods. There are also decoding complexity benefits to the shorter block lengths at the end of the optimized schedule.

Figure 3.12 shows a comparison of the performance of the turbo, LDPC, and RCPC/CRC codes over a range of error rates. The performance measure displayed is the value of the expected PSNR averaged over the entire transmission. This measure is simply the area under the performance-rate curve divided by the total transmission rate, or simply a normalized version of the PSNR-based performance measure. The LDPC and turbo codes had very similar performance for low error rates, and the turbo codes were slightly better for high error rates. The RCPC/CRC codes performed about 0.5 to 1.0 dB worse with the performance difference generally increasing with error rate.

Finally, a natural question to consider is how much the serial decoding requirement of the source coder costs in terms of additional required channel coding. Since decoding must terminate at the first uncorrected channel error, bits early in the bit stream will be given more protection to improve the chance that later correctly received bits can be used by the decoder. Answering the question would give some insight into the potential benefit of modifying the source coder to make the bit stream more independently decodable. Presumably such a modification would cost something in terms of source coding performance. To get an approximate bound of the benefit for channel coding, we can make the optimistic assumption that any error-free packet can be used and will lower the MSE by the same amount as corresponding bits

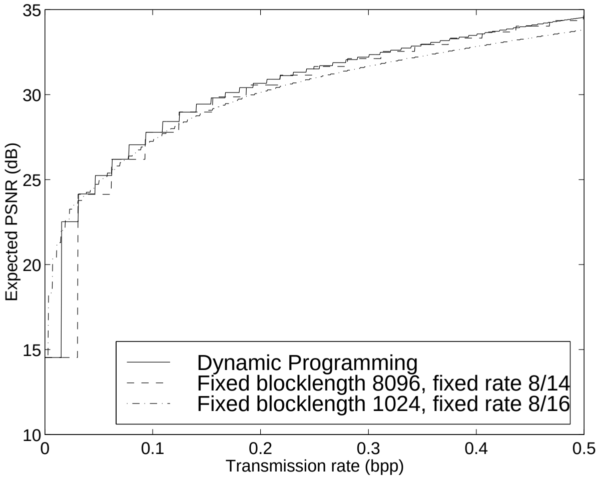


Figure 3.11 Performance results for the 512×512 Lena image using turbo codes over a BSC with BER $= 0.05$.

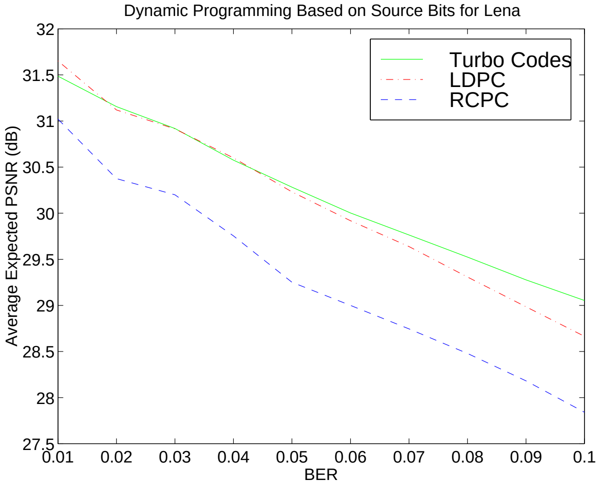


Figure 3.12 Performance comparison between RCPC/CRC, LDPC, and turbo codes.

under error-free conditions. Experimental results showed very small gains using these optimistic assumptions (e.g., typically less than 0.1 dB), so it appears that such modifications to the source coder would not be worthwhile, at least for the case of a memoryless channel with a known error rate. These type of changes to the source coder are more beneficial for channels with varying conditions.

3.3.5 Conclusion

We have proposed a general performance measure for evaluating progressive image transmission systems operating over noisy channels. Using this performance measure, a dynamic programming optimization algorithm was presented for determining the best code schedule where both block length and rate are optimized. While the performance gains of the image-dependent PSNR-based optimization may not warrant the additional complexity in every application, the method is useful for evaluating the relative performance of simpler coding schemes. Experimental results were provided comparing the performance of turbo codes, LDPC codes, and RCPC/CRC codes for progressive coding. Even though the progressive goals of the optimization tended to limit the maximum block lengths of the turbo and LDPC codes, they still performed significantly better across a range of channel error rates. The performance difference between the LDPC and turbo codes was only really noticeable at high error rates with turbo codes performing a bit better.

CHAPTER 4

IMAGE CODING FOR FADING CHANNELS

4.1 Introduction

In Chapter 3 methods and results were presented for image transmission over memoryless channels where the error rate is known. It was shown that excellent performance can be achieved by combining a state-of-the-art embedded image coder with strong channel coding. When the error rate is a known constant, then the channel coding can be closely matched to the error conditions leaving more rate available for source coding. However, when channel conditions vary, such as on fading channels or in a broadcast environment, the error conditions are more challenging.

This chapter presents two approaches to coding for varying channels. First, a product channel code is presented which can be viewed as an extension of the RCPC/CRC code from Chapter 3. The new code is designed to work well on fading channels, but it also improves on the performance of the RCPC/CRC code over memoryless channels. The product code results presented in this chapter are from [33, 34].

Second, a combination of the Packetized Zerotree Wavelet (PZW) coder [70, 71], a variant of the zerotree coder which uses data partitioning to introduce error resilience, and the RCPC/CRC channel codes is presented for varying channel conditions including those combining bit errors and packet erasures. A hybrid of the two methods is shown to degrade more gracefully than either method alone under severe channel conditions. The results on the hybrid coding method are from [36, 37].

4.2 A Product Channel Code

A product code is often described as a two-dimensional code constructed by encoding a rectangular array of information digits with one code along rows and with another code along columns [46]. In the product code used here, the row code is a concatenated code consisting of an outer cyclic redundancy code (CRC) and an inner rate-compatible punctured convolutional (RCPC) code while the column code is a systematic shortened and/or punctured Reed-Solomon (RS) code. The structure of the product code is depicted in Figure 4.1. Note that the RCPC codes used for the rows are not systematic, and the symbols for the RS codes are constructed from consecutive information bits of a row prior to encoding with the RCPC/CRC code.

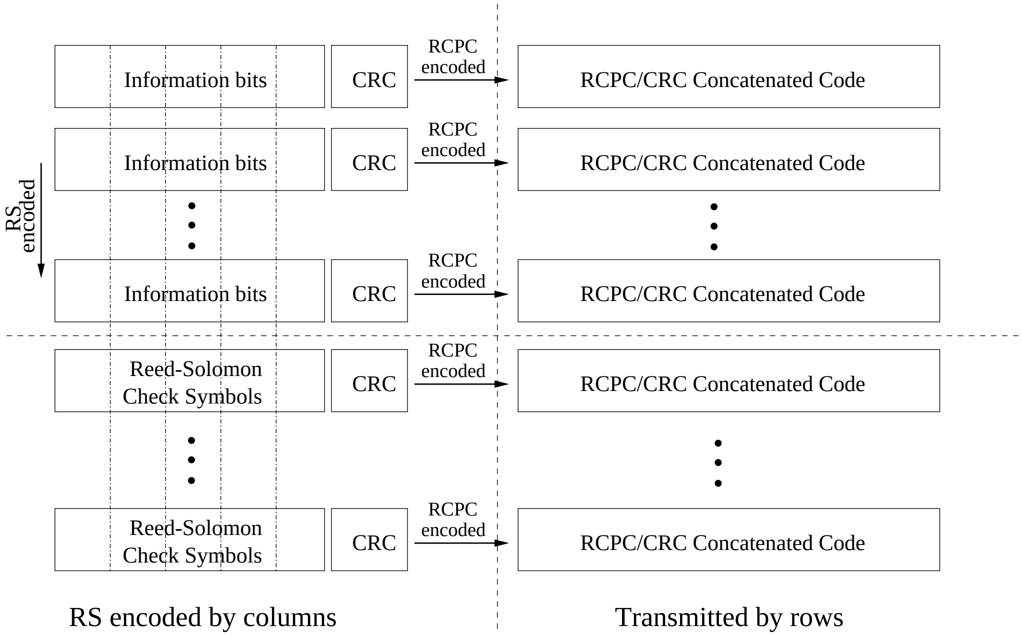


Figure 4.1 Schematic diagram of RCPC/CRC and RS product code.

The RCPC/CRC concatenated code allows substantial flexibility in choosing the code rate and block length of the rows. The row codes are decoded using the list-Viterbi algorithm which selects the trellis path with the best metric, that also satisfies the CRC check, from a ranked list of candidates. The correct path is typically among the first few top candidates so a long search is rarely necessary. Also a sequential version of the algorithm can reduce computational complexity by only searching for the next best path after higher ranking candidates have failed.

More thorough discussions of the list-Viterbi decoding algorithm and applications can be found in [28, 49], and recent applications of list-based decoding to turbo codes can be found in [72].

An important property utilized by the product code is that the CRC provides a high probability indication of the decoding success or failure. The decoding failures in the row codes are treated as erasures when decoding the column RS codes. Since the column codes typically only need to correct erasures, the computational complexity is reduced and twice as many incorrect rows can be recovered compared to a decoder without error detection on the rows. The Galois field of the RS code symbols should be chosen as large as possible to reduce the number of columns in a block and to give the most flexibility in choosing the block lengths via shortening and/or puncturing. The RS codes are in systematic format with information symbols transmitted first, so the final rows of the product codeword will be the RCPC/CRC encoded parity symbols of the RS column codes.

There is no requirement that the rows be consecutive in the bit stream, and, in fact, spreading out the rows is important for good performance over fading channels. Of course, another design goal is to minimize delay in order to achieve rapid improvement in image quality, and this will provide some constraints to limit the duration of the code. A useful feature of this particular product code is that decoding columns is unnecessary unless a decoding failure is detected in a row code. Therefore, the decoded bits from a row can be used immediately if no decoding failure is detected in the row, eliminating the delay cost when the channel is clear.

The product code is well suited for burst errors since entire rows can easily be recovered. This property is important since even a single bit error in a packet of data from the embedded zerotree algorithms usually renders the entire packet (and also the packets to follow) useless. The product channel code also performs excellently over memoryless channels like the additive white Gaussian noise (AWGN) or binary symmetric (BSC) channels. This observation is an additional side benefit.

4.2.1 Performance on a binary symmetric channel

We illustrate the effectiveness of this code by way of an example on a BSC. The source and channel were selected to allow comparison with the results from [28] and consist of the progressive zerotree wavelet coder with arithmetic coding by Said and Pearlman (SPIHT) [10] transmitted over a BSC with error rate 0.1. The total block length of the column RS codes with symbols from GF(256) was limited to 20 symbols to limit the decoding delay, and the source bit stream was protected uniformly by selecting the combination of RCPC/CRC and RS code rates which gave equivalent probability of error results as the RCPC/CRC code in [28]. Also the path search depth of the list-Viterbi algorithm was limited to 100 as in [28], but we note that a search depth of 10 will provide almost all the performance gain of list decoding. Under these conditions, the same probability of decoding error can be achieved with an overall channel coding rate of 0.295 for the product code versus a rate of 0.257 for the RCPC/CRC code alone. Therefore, nearly 15% more rate is available for source coding for a given overall transmission rate. This results in a gain in decoded image quality of about 0.5 dB in PSNR. The decoded PSNR values for the Lena image as a function of rate for these codes are shown in Figure 4.2 along with the noiseless channel results for the source coder without any channel coding for comparison.

Several modifications are possible to tune the properties of the code. In general, more efficient codes can be created using longer RS codes (i.e., more rows) at the expense of more delay required to correct row decoding failures. Also, some improvement in error performance can be obtained at the expense of complexity by using an iterative decoding algorithm (e.g., turbo decoding). Finally, while the above results for the BSC use hard decision decoding, soft decisions could easily be incorporated into the decoding process giving improved performance.

4.2.2 Performance on fading channels

While the performance of the product code is good over memoryless channels, one of its most important features is its suitability for fading channels which arise in wireless applications. A typical approach to error control for fading channels is to introduce a bit interleaver which spreads out adjacent bits by the interleaver depth before transmission over the channel. The goal is to produce an effective channel, after the de-interleaver, which is almost memoryless and then to use conventional error control coding (e.g., convolutional codes) to deal with the errors. One problem with this approach is that an interleaver of depth n introduces a

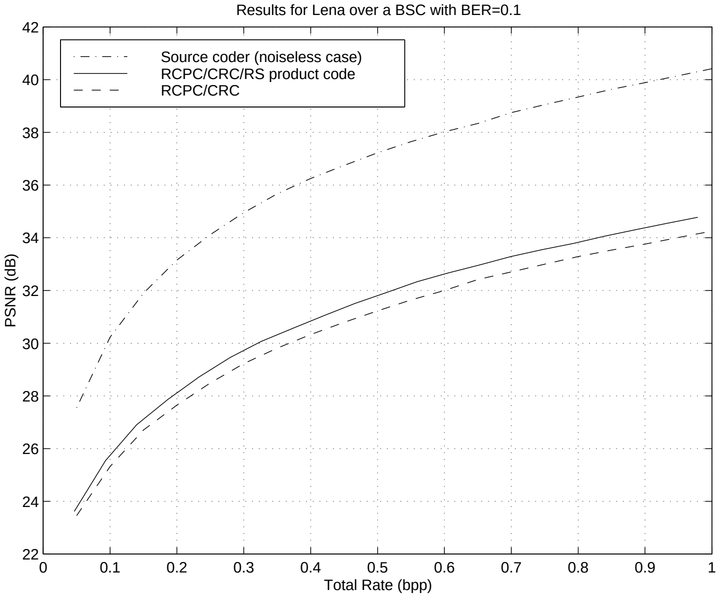


Figure 4.2 Comparison of code performance for the 512×512 Lena image over a BSC with BER $= 0.1$.

delay on the order of n^2, and this delay is constant regardless of the channel conditions. Any delay impacts the performance of the progressive coder because the goal is to improve quality (PSNR) as rapidly as possible, and delay shifts the PSNR vs. rate curve to the right, lowering the PSNR for a given rate. As mentioned earlier there is no significant delay with the product code unless there is a row decoding failure. In that case, the delay depends on the number of bits that must be received before the necessary number of RS check symbols are available (i.e., equal to the number of erased rows).

4.2.3 Performance on the Gilbert-Elliot channel

A basic channel model incorporating the memory associated with fading channels is the Gilbert-Elliot two-state model. A description of this channel model is given in Section 2.3.2.1. This same model was used in [73] to model a channel with memory.

The Gilbert-Elliot channel model was used to compare the effectiveness of the product code versus the RCPC/CRC code along with bit interleaving. The selected model parameters were $\epsilon_{\mathrm{B}} = 0.1$, $\epsilon_{\mathrm{G}} = 0.001$, $P_{BG} = \frac{1}{400}$, and $P_{GB} = \frac{1}{9}P_{BG}$. With these parameters, the steady state probability of being in the bad state is 0.1 and the mean burst duration is 400 bits. The performance requirement we chose for the codes was the same as that used for transmission over the BSC—at least 99% error-free transmission at a total transmission rate of 1.0 bpp. These parameters were chosen to illustrate the performance of this system. In practice, more realistic values would need to be determined.

An RCPC/CRC code of rate 0.257 used on the BSC with BER $= 0.1$ conservatively meets the performance requirement, as it codes for the worst case error rate seen in the bad state. However, using bit interleaving at the expense of a fixed delay, a higher code rate can achieve the same probability of error requirements. Fully interleaving the channel so that the errors appeared memoryless was considered impractical for this channel since the required interleaver depth would be excessive when considering the goals of low overall transmission rates and rapid image quality improvement. An interleaver depth of 60 was selected as a reasonable value, and although it does not completely remove the memory, an RCPC/CRC code of rate 0.36 is able to meet the probability of error requirements.

The product code was designed by starting with a row RCPC/CRC code with a rate of 0.81 which was selected because it met the performance requirement when the channel was in the good state. The column code was then chosen to handle the row decoding failures that occur when a portion of a packet is transmitted during the bad state. In order to reduce the necessary error correction capability (and thus the redundancy) of the column code, several product codes were interleaved so that a single error burst would not cause multiple row erasures within a single product code. Although interleaving in this manner does increase the delay when row erasures occur versus not interleaving, the delay is not fixed as in the case of bit interleaving and is typically much less than the full extent of the product code. The selected column code consisted of a (20, 11) RS code over GF(256) with a row spacing of 6 (i.e., 6 interleaved product codes). The rate of the resulting product code is about 0.44 which yields a 23% increase in the available source rate compared with the interleaved RCPC/CRC code. The higher rate of the channel code translates into an increase of about 1.0 dB in PSNR over a range of transmission rates for the Lena image as can be seen in Figure 4.3.

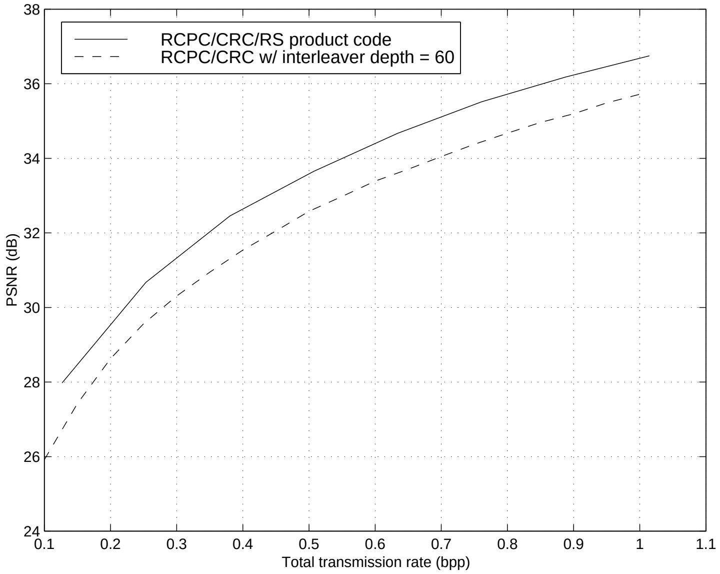


Figure 4.3 Comparison of code performance for the 512×512 Lena image over a Gilbert-Elliot channel.

4.2.4 Performance on the Rayleigh channel

Further simulations were performed for BPSK transmission over a flat-fading Rayleigh channel using Jakes' method [42] to model the channel. A description of this model is given in Section 2.3.2.2. The results presented are for a channel with average SNR of 10 dB and a normalized Doppler spread of 10^{-5} which is near the low end of many typical applications. An example leading to a normalized Doppler value of 10^{-5} would include a data rate of 500 Kbits/s transmitted at 900 MHz to/from a mobile traveling at about 4 miles/h (e.g., a person walking). With these parameters, the average duration of a fade with a channel bit error rate exceeding 0.1 is on the order of 12000 bits while the average duration of a fade with a channel bit error rate exceeding 0.01 is on the order of 24000 bits. The strength of the RCPC/CRC channel codes in this case will determine the channel error rate which can be handled and thus the fade margin. Normalized Doppler values lower than 10^{-5} result in fade durations that include such a large portion of the bits that the channel is either good or bad for almost the entire image transmission—a situation better combatted using spatial diversity, frequency diversity, etc.

The results presented in Figure 4.2 for the BSC essentially had a single decoded PSNR value at each rate (despite being an "average" value) for each of the two codes since the probability of decoding failure was so low. However, when examining the performance of a code after transmission over a noisy channel, it is often insufficient to consider only the mean decoded PSNR, especially for time-varying channels. Instead, a distribution of decoded PSNR values for each rate of interest is more appropriate since it shows the variability of the decoded image quality. A realistic performance measure should include some combination of the expected PSNR and a measure of the variability, although the best relative weighting is probably viewer and application dependent.

Therefore performance results for the fading channel will be presented using a cumulative distribution of decoded PSNR values at a particular total transmission rate. In this type of plot, curves with better performance will generally lie closer to the bottom and right edges of the plot indicating a higher frequency of large PSNR values. The trends are displayed in Figure 4.4 where the horizontal axis could be any fidelity measure and will be PSNR for the results presented here. Note that reported mean PSNR values are computed by averaging decoded MSE values and then converting the mean MSE to the corresponding PSNR value rather than averaging the PSNR values directly.

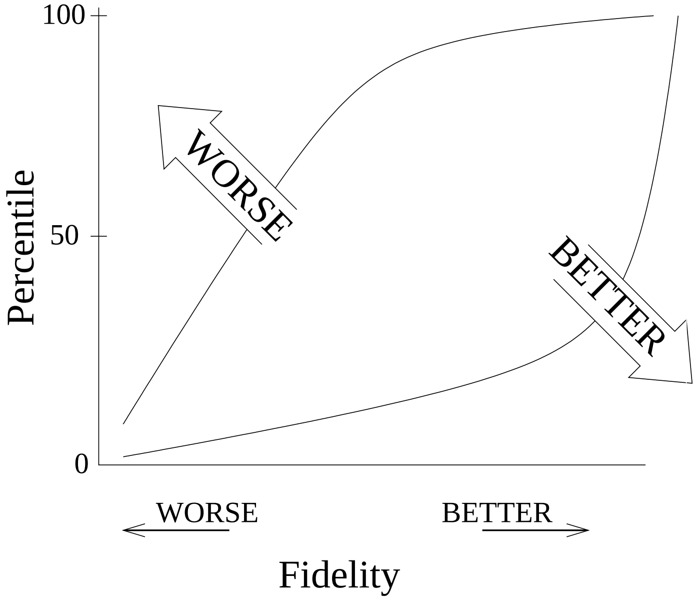


Figure 4.4 Cumulative distribution plot for performance evaluation.

For the results presented in this section, each of the codes was constructed from the output of the SPIHT [10] image coder with arithmetic coding. Each cumulative distribution curve represents at least 1000 independent trials. The 512×512 Lena image was used in each case and the total rate (including both channel and source coding) considered was 0.25 bpp. Blocks of 200 information bits were protected by a 16-bit CRC and encoded by RCPC codes of various rates. For the product codes, groups of eight consecutive information bits made up the symbol values for the RS codes over the finite field GF(256). The path search depth of the list-Viterbi algorithm was limited to 100 paths.

The first set of results shown in Figure 4.5 compares the performance of a RCPC/CRC concatenated code with RCPC rate $1/4$, the same code with a convolutional interleaver of depth 80, and a product code using RCPC rate $1/2$, a shortened (16, 10) RS code, and a row spacing of 4 (i.e., 4 interleaved product codes). The results show that the bit interleaver does not really improve the performance since the interleaver depth is small relative to the average fade duration given above. The use of the interleaver does reduce the tail of the distribution slightly, but the peak PSNR is also reduced due to the interleaver delay so the mean is only improved by 0.014 dB. The product code performs much better than either of the other two codes giving both a higher peak PSNR and a lower tail, which results in an improvement in peak PSNR of 1.2 dB and in mean PSNR of 1.6 dB.

As with the BSC, limiting the path search depth of the list-Viterbi algorithm to 10 paths makes very little difference in the distribution of decoded PSNR values for either the RCPC/CRC-only code or the product code on the fading channel. For the channel conditions described above, changing the path search depth from 100 to 10 results in almost no difference in the PSNR distributions. Figure 4.6 shows cumulative distributions for the RCPC/CRC code without interleaving and the product code shown in Figure 4.5 with trellis search depths 10 and 100 paths. Clearly, the distributions for each code are nearly identical using either 10 or 100 paths. In fact, limiting the search depth to 10 paths may slightly benefit the product code performance because searching deep in the list of candidate trellis paths increases the likelihood of mistakenly selecting an incorrect path, which is worse for column decoding than simply declaring an erasure.

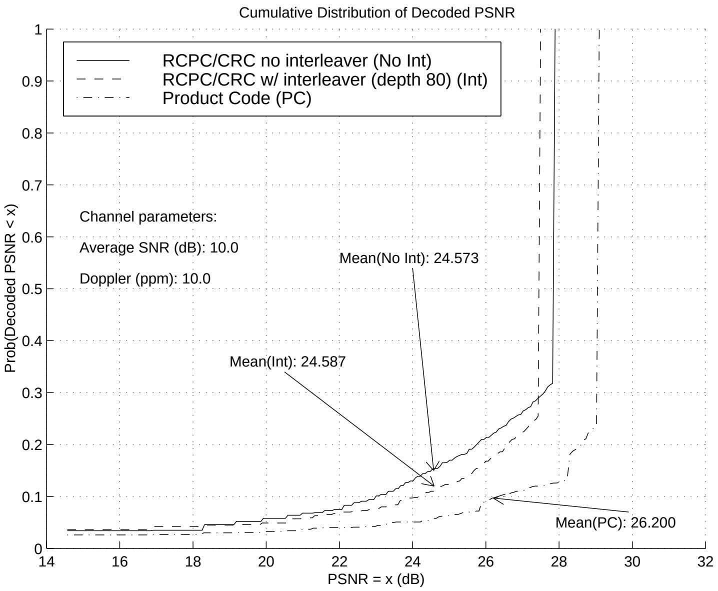


Figure 4.5 Comparison of code performance for the 512×512 Lena image transmitted over a Rayleigh fading channel with total rate 0.25 bpp.

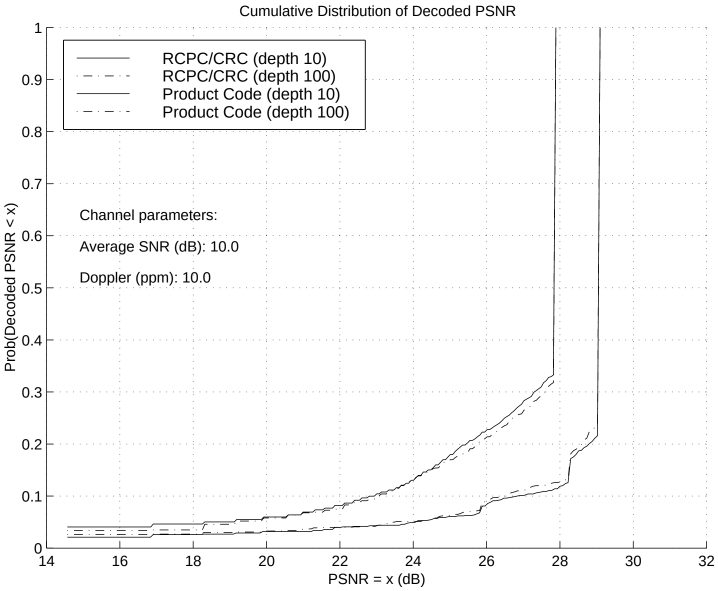


Figure 4.6 Comparison of code performance with different trellis search depths for the 512×512 Lena image transmitted over a Rayleigh fading channel with total rate 0.25 bpp.

4.2.5 Applying unequal error protection

The results presented in the previous sections were for channel protection applied uniformly to the source coder output. However, the bits do not have equal importance in determining the final decoded PSNR. As previously mentioned, the importance of the bits roughly decreases monotonically throughout the bit stream for the SPIHT algorithm. A better rate-distortion trade-off can be achieved by tailoring the channel protection based on the importance (i.e., UEP).

Because of the serial decoding requirement of the SPIHT algorithm, the decoder must stop decoding at the first detected but uncorrected error. This situation leads to the long flat tail in the distributions shown in Figure 4.5 where the minimum decoded PSNR value of 14.53 dB represents the case where the first packet was lost and only the image mean was used for reconstruction. Therefore, providing more protection for the first few packets can result in improved performance.

The code structure allows several ways to implement a UEP scheme including: varying the rate of the row RCPC/CRC code, varying the rate of the column RS code, or including important information rows in multiple product codes. For the channel conditions and codes used in Figure 4.5, the best approach for reducing the long tail of the distribution is probably to include the initial information packets in additional product codes. The reason is that additional RCPC coding may still have problems correcting the errors associated with a deep fade, and simply increasing the redundancy of the column codes may not allow tailoring the additional protection to the importance as accurately (i.e., the rows in a given product code can have a large variation in importance, especially at the beginning).

Results for two UEP schemes as well as the product code from Figure 4.5 are shown in Figure 4.7. Both UEP schemes were constructed by protecting the first two information packets with an additional shortened (4, 2) RS column code and transmitting the parity rows after half of all packets had been sent. In addition, the first 10 information packets were protected by a shortened (20, 10) RS column code with the parity rows sent as the final 10 packets of the image. The difference between the two was in the other coding parameters where UEP scheme

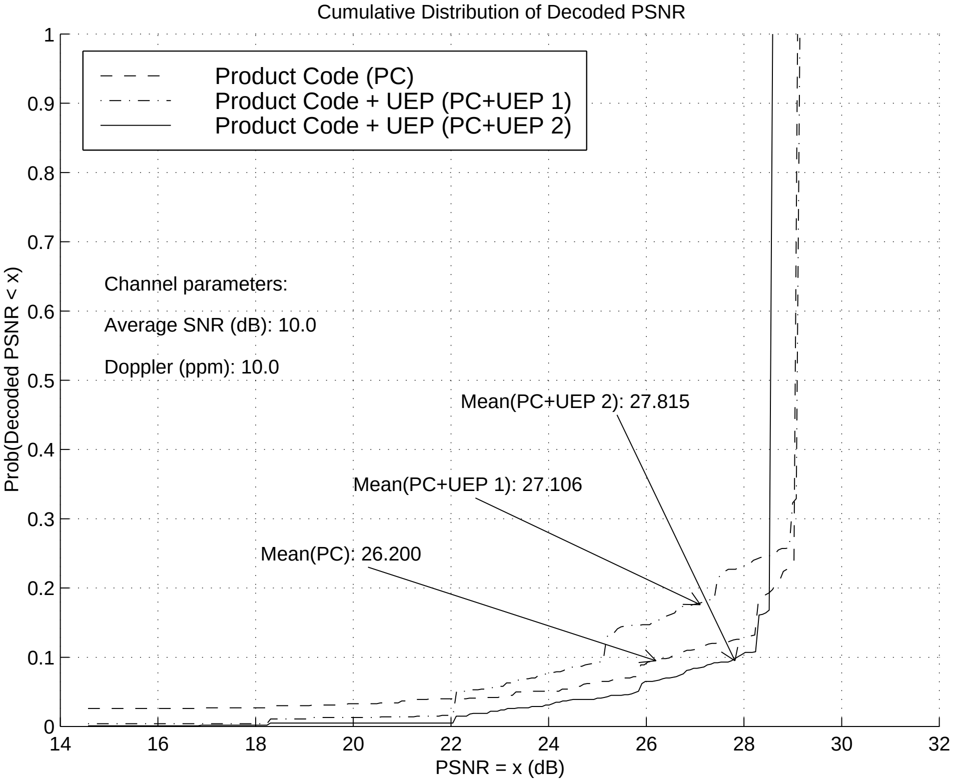


Figure 4.7 Comparison of code performance for the 512×512 Lena image transmitted over a Rayleigh fading channel with total rate 0.25 bpp.

1 used RCPC rate $2/3$, a shortened (15, 8) RS code, and row spacing of 4; and where UEP scheme 2 used RCPC rate $1/2$, a shortened (16, 10) RS code, and row spacing of 4.

As can be seen in Figure 4.7, both UEP schemes greatly reduce the tail of the distribution, which is reflected in the increased mean values with gains of 0.9 dB and 1.6 dB compared with the uniform product code. The fact that the mean for UEP 2 is greater than for UEP 1 comes at the expense of decreased peak (and typical) PSNR, which makes it difficult to judge which of the two is preferable and illustrates the point that mean PSNR may not always be the best metric.

4.3 Combining Robust Source Coding and FEC for Varying Channels

The hybrid coder consists of the PZW source coder [71] combined with the RCPC/CRC FEC coder described in Section 3.2. Figure 4.8 shows a block diagram of the proposed method. Image data is transformed into the wavelet domain. Groups of coefficient trees are placed into fixed-length packets (as in PZW). Each packet is then wrapped by a RCPC/CRC code (as in the SPIHT+RCPC/CRC algorithm). The protected packets are sent across the channel and decoded by the channel decoder followed by the source decoder. Missing wavelet coefficients in the low-low band (from packet erasures or channel decoding failures) are replaced by interpolating values of all available eight-neighbor pixels. Missing coefficients in higher bands are set to zero before inverse wavelet transforming.

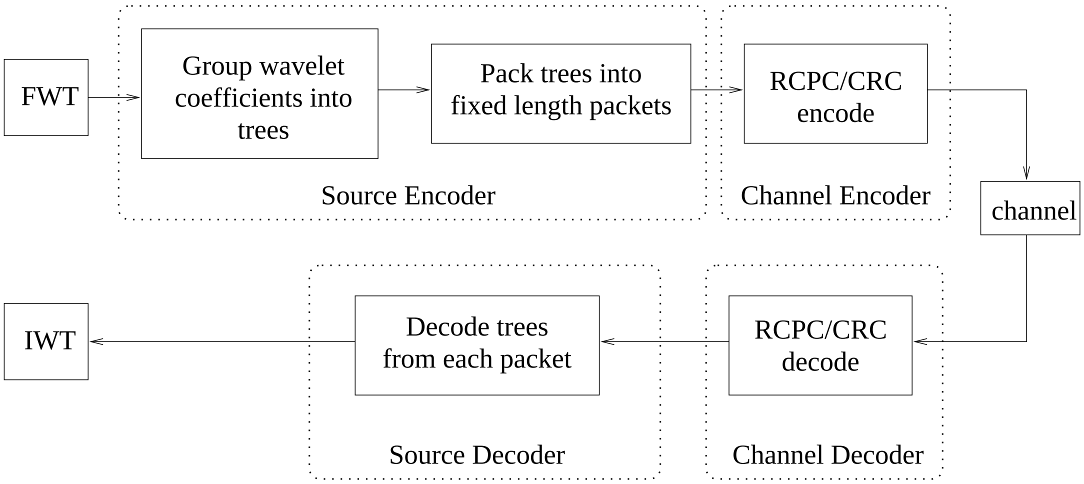

FWT
Group wavelet coefficients into trees
Pack trees into fixed length packets
RCPC/CRC encode
Source Encoder
Channel Encoder
channel
IWT
Decode trees from each packet
RCPC/CRC decode
Source Decoder
Channel Decoder

Figure 4.8 Block diagram of the proposed coder.

The two approaches used together are intended to help exploit the advantages of each other. Bit errors occurring throughout the stream (which would devastate the PZW coder) are corrected by the FEC; severe bursts of errors (which would overwhelm the FEC or would impose too severe a rate penalty if included in the FEC design) are absorbed by the underlying resilience of the PZW source coder. The PZW coder also handles packet erasures which would cause early truncation of the bit stream for the SPIHT coder.

The proposed coder is designed for a channel (discussed in detail in Section 2.3.4) consisting of a wireline portion and a wireless portion. Losses in the wireline portion consist of packet erasures where lost packets, due to buffer overflow or misrouting, do not arrive at the receiver. The wireless portion of the channel causes losses from excessive bit errors due to the fading channel characteristic.

In particular, the packet length at the output of the source encoder (used on the wireline portion of the channel) is fixed at 384 bits to match asynchronous transfer mode (ATM) packet payload sizes. The RCPC/CRC code is not designed for the worst channel, but rather for conditions in the middle of the expected range for the wireless portion of the channel. Thus, more transmission rate can be dedicated to source coding than with the SPIHT+RCPC/CRC coder which would have to be designed for the worst case. The method for selecting the appropriate channel code is described in Section 4.3.1.

Although the proposed method is not claimed to be optimal, the particular components are well suited for each other and for the goal of producing a robust coder on the combined packet erasure / bit error channel. First, the PZW coder works well for the small ATM packet lengths (i.e., the source coding performance is close to that without segmentation into packets). The small packet size helps provide robustness since a packet only contains a small portion of the image information which means the impact of a lost packet is small. Second, the RCPC/CRC FEC provides good error correction performance for the wireless portion of the channel which introduces bit errors. As part of the decoding process, channel decoding failures can be detected with high probability so the source decoder can drop uncorrected packets rather than corrupt the image by decoding those bits. Use of more recent FEC methods such as turbo codes provides the most benefits in situations where soft decision channel decoding is available, channel state information is fairly well known, and the added decoding complexity is acceptable. The improved performance of these codes typically requires longer block lengths, which may reduce robustness when channel conditions are worse than the design conditions. Also it is important to detect channel decoding failures, so additional coding would be necessary if this capability is not normally a feature of the code.

4.3.1 Parameter value optimization

In evaluating the performance of an image coder for noiseless channels, the PSNR versus bit rate curve is often used. In general, coder A is said to perform better than coder B if the curve of PSNR versus bit rate for A is pointwise greater than that for B. If the comparison between coders is performed at a specific bit rate, then the results can always be ordered, and the system designer has a clear optimization goal. But a single transmission rate does not capture the overall behavior of a coder used progressively, and may thus be misleading. Without focusing on a particular bit rate, there may not be a clear winner, since the curves may cross. Also, the optimization goal is difficult to specify when considering entire curves, since the performance at low bit rates or at high bit rates may be more important for specific applications.

In the noisy channel case, the situation is more complicated. Suppose the channel code rate and total bit rate are fixed. Then the performance of the system is characterized by the cumulative distribution function (CDF) of the mean squared error (MSE), denoted by $F(x) = \mathrm{Prob}($ Decoded Distortion $< x)$. The goal is to design a system with a high probability of producing images with low MSE. Thus, a CDF is good if it rises sharply at a low MSE. As before, we would say that system A outperforms system B if the CDF for A lies everywhere above the CDF for B. If the two CDF curves cross, the comparative evaluation is more difficult. As before, one can try to escape this complication by looking at a particular point, say, the median MSE. But this method is unsatisfactory, since it ignores overall behavior.

In this section, we describe a general class of performance measures which can be specialized for the specific design goals of an application. Given a performance measure and a range of channel conditions, the best choice of channel code for both the SPIHT+RCPC/CRC coder and the hybrid coder can be found. Let Φ be a specification of the system free parameters to be optimized. For our case, Φ is a channel code (or equivalently the corresponding channel code rate, when a family of codes parameterized by code rate is used). For any particular channel code, the performance of the system is characterized by the cumulative distribution function (CDF) of the MSE, denoted by $F_\Phi(x) =$ Prob$(\text{Decoded Distortion} < x)$. In order to quantify the performance of a system, it is convenient to reduce the function $F_\Phi(x)$ to a single real number. We thus define the fidelity criterion

$$J_\Phi = \int_0^\infty W(x)(1 - F_\Phi(x))\, dx \tag{4.1}$$

where $W(x)$ is a nonnegative "weighting" function which allows us to emphasize or de-emphasize the contribution of a particular MSE region to the total fidelity. In general, it results in a fidelity criterion which is a weighted conditional mean of the decoded MSE values. The goal is to minimize J_Φ over all Φ in some predefined family of codes.

Some specific examples are given below, where the unit step function is $u(x) = 1$ for $x \geq 0$ and $u(x) = 0$ for $x < 0$, and where $\delta(x)$ is the Dirac delta-function.

- *Example (i)*: $W(x) = 1$

  This fidelity criterion is the mean decoded MSE. Figure 4.9 demonstrates this example, where J_Φ is the shaded area in the graph.

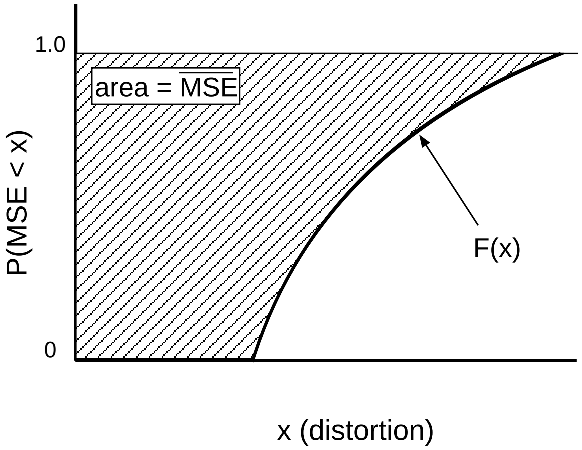


Figure 4.9 The shaded area equals the fidelity criterion J_Φ with $W(x) = 1$ which is the mean decoded MSE in this case. $F(x)$ in the graph represents a generic cumulative distribution function.

- *Example (ii)*: $W(x) = u(-x + x_h) + C\delta(x - x_h)$

  This fidelity criterion assumes a maximum level x_h of tolerable MSE for a decoded image. Images decoded with distortion greater than x_h are useless and therefore should not influence the optimization with respect to images having smaller MSE. The constant C allows the designer to trade off between the weighted conditional expected distortion of images decoded below x_h and the probability that images will be decoded above x_h. Figure 4.10 shows an example with such a weighting function where the fidelity criterion is equal to the area of the shaded region.

- *Example (iii)*: $W(x) = u(x - x_l)$
  This fidelity criterion assumes any MSE below a certain threshold x_l is essentially equal to zero. This can be due to limitations of the output device or limits of human perception, for example. Figure 4.11 shows an example where this fidelity criterion is equal to the area of the shaded region.

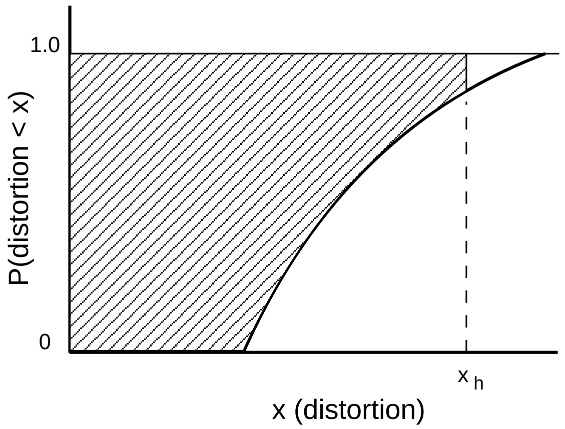


Figure 4.10 The shaded area equals the fidelity criterion J_Φ with $W(x) = u(-x+x_h)$.

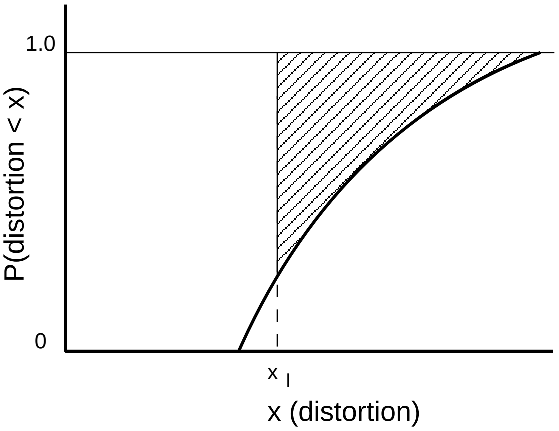


Figure 4.11 The shaded area equals the fidelity criterion J_Φ with $W(x) = u(x - x_l)$.

In addition, performance requirements can be incorporated into the optimization by restricting the set of system parameters. An *admissibility function* $A(x)$ is a nonnegative real function which provides a pointwise lower bound constraint for acceptable MSE CDFs. Specifically, we require $F_\Phi(x) \geq A(x)$ for all x. Figures 4.12 and 4.13 show two examples of admissibility functions in relation to distribution functions. An admissibility function can be used to capture typical constraints such as upper bounds on the number of images with high distortion or lower bounds on the number of images with low distortion. For example, Figure 4.12 shows an admissibility function where at least 90% of the images can be decoded with MSE less than a given value or equivalently where no more than 10% of the images can be decoded with

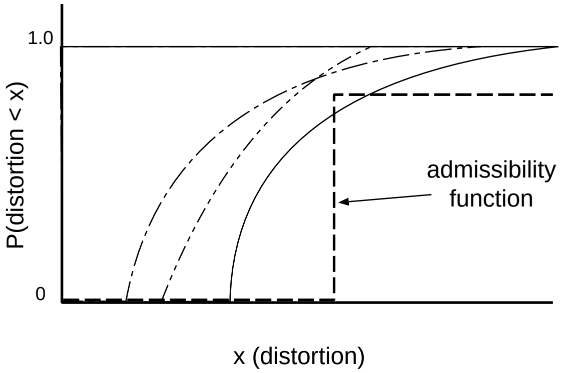


Figure 4.12 An admissibility function which places an upper bound on the acceptable probability of exceeding a certain MSE. Two of the three cumulative distributions satisfy the constraint.

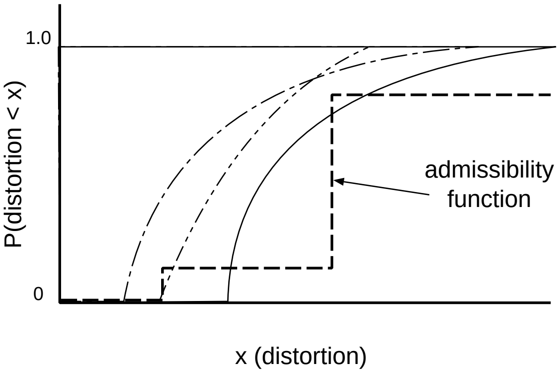


Figure 4.13 An admissibility function requiring minimum performance levels for two MSE ranges. Only one of the three distributions satisfies the constraints.

MSE greater than the same value. Figure 4.13 shows an admissibility function which requires a minimum probability for low MSE values in addition to the upper bound on the probability of large MSE values.

The discussion above considers the case of optimizing over a single channel. For systems that operate over a variety of channels, the optimization might need to include a combination of the fidelity criteria from the different channels each with possibly unique weighting and admissibility functions.

We introduce this framework in order to point out how channel code rates were chosen for the current work. For both the hybrid and SPIHT+RCPC/CRC coders, only a finite number of channel codes were available. For each channel code, 1000 trials were run over the channel for which the system was being optimized. The mean decoded MSE (or weighted mean MSE) was calculated for each code rate and the minimum was used as the optimal code. For both cases, the admissibility function was $A(x) = 0.95u(x - 818.6)$ which requires that no more than 5% of the images have an MSE greater than 818.6 (818.6 corresponds to PSNR $=$ 19dB). For the mean decoded MSE, the weighting function was unity for all x. For the weighted mean MSE, we used the same type of weighting function as in Example (ii) above with $x_h =$ 818.6 and $C = 0$. This function was chosen using data collected from perceptual recognition experiments by Serrano et al. [74] in which no successful recognition occurred for images with

$\mathrm{PSNR} < 19\mathrm{dB}$. For both algorithms, it was found that the optimal available code was the same for both the weighted and standard mean decoded MSE cases. In practice the admissibility and weighting functions can be chosen according to the specific constraints of a particular application.

4.3.2 Results

First we present results for the special case when there are no packet erasures, in order to demonstrate that the hybrid scheme survives well in this case, and in fact even outperforms its two component encoders under certain channel conditions. Then results for channels with packet erasures and bit errors are presented. In all tests, hard-decision decoding was used in the channel simulation.

To test the robustness of the coders, each was optimized for a fixed channel ($\overline{\mathrm{SNR}} = 13$ dB, $f_D = 10^{-4}$) and then tested over a range of channel conditions. The optimal code rate was chosen using the previously discussed parameter selection method (Section 4.3.1). For the hybrid and SPIHT+RCPC/CRC coders, the bits were interleaved using a convolutional interleaver prior to transmission over the fading channel to improve decoding performance. The detailed specifications for the selected codes can be found in Section 4.3.4. All tests were performed using the 512×512 Lena image, and the total transmission rate was fixed at 0.25 bpp. Each channel condition was tested with a minimum of 1000 independent trials and as many as 3000 trials on the slowest channels.

In evaluating the performance of the three algorithms, the standard measure of mean decoded MSE may not be sufficient. To improve the analysis, we looked at three characteristics: visual quality, mean decoded MSE, and cumulative distributions of MSE over all trials. The cumulative distributions provide information about the variability of the decoded image quality from one trial to the next. The visual quality, while more difficult to evaluate, is ultimately the performance measure of interest.

4.3.2.1 Visual quality

Figure 4.14 shows how distortion from a noisy channel is distributed over the image for the hybrid and SPIHT+RCPC/CRC algorithms. Because the hybrid coder groups wavelet coefficient trees together, when losses occur these trees are lost as a single unit. This localizes the region of the image which will be distorted. Trees which are not lost will be decoded with high relative quality (dependent only on the source coding rate). Furthermore, because the regions that were received have good quality, the hybrid algorithm can use the correlation of these neighbors to mitigate the effect of the lost regions. The SPIHT+RCPC/CRC coder distributes the distortion somewhat uniformly over the entire image. The maximum distortion for any pixel may be lower than in the hybrid scheme, but for large total distortion, a spatially distributed error can be perceptually undesirable. This difference in visual quality must be taken into account when interpreting the numerical results.

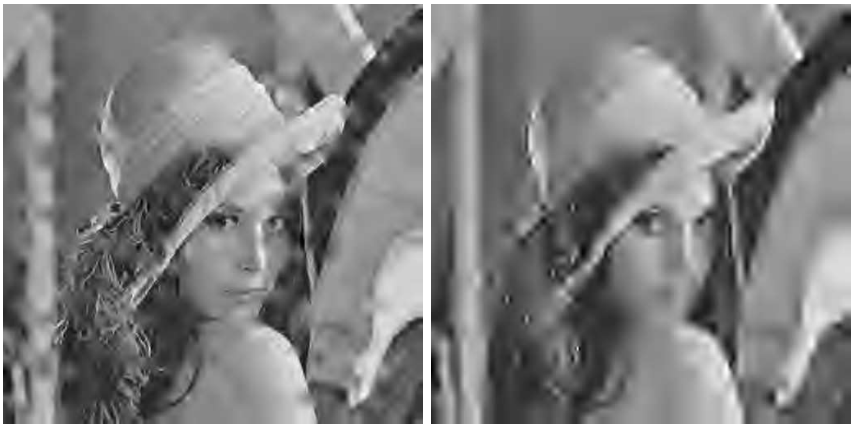


Figure 4.14 Images displayed here show the visual effects of loss for the hybrid coder (left) vs. the SPIHT+RCPC/CRC coder (right) at equal MSE (PSNR $= 23.5$dB). As shown in [74], the reconstructed images with more localized distortions tend to be preferred over the ones with global blurriness, both in terms of subjective quality and in terms of image content recognizability.

These observations are supported by results of a recent study [74]. In [74], human observers were asked to evaluate a series of images compressed with the SPIHT and PZW algorithms at equal PSNR values. The distortions in the images were similar to those that would result from operation over noisy channels. It was found that the PZW sequences allowed observers to make responses to objective recognition tasks at lower PSNR than the SPIHT sequences. In addition, subjective rankings on a five-point scale found that PZW images were preferred to SPIHT images at the same PSNR. From these human observer experiments we conclude that the cumulative distribution plots of PSNR are somewhat conservative on the side of underestimating the quality of the hybrid and PZW coders.

4.3.2.2 Mean decoded MSE

Table 4.1 shows the mean decoded MSE values as well as the weighted MSE values for a number of channel conditions which span the range of interest. The largest differences occur for the most severe channels in the upper left section of the table. In these cases, the channels were slow enough that interleaving was not effective and error rates were high due to the low received SNR. The initial packets were often lost for the SPIHT+RCPC/CRC coder which resulted in very high MSE values, and this greatly increased the mean MSE. The hybrid coder has the advantage of being able to continue decoding after error bursts have affected the initial packets (and the initial packets were not more vital to image quality than were later packets).

The hybrid algorithm exhibits superior performance over the SPIHT+RCPC/CRC coder over many channels in terms of mean decoded MSE (Column A in Table 4.1). The weighted distortion (Column B) tends to benefit the SPIHT+RCPC/CRC coder more than the others. In this case, images with very large distortion (above x_h) are not included in the average and therefore do not skew the average. Only the percentage of such images affects this performance measure. Even with this performance measure, the hybrid shows superior performance on the most severe channels (upper left) compared with either of the two other algorithms.

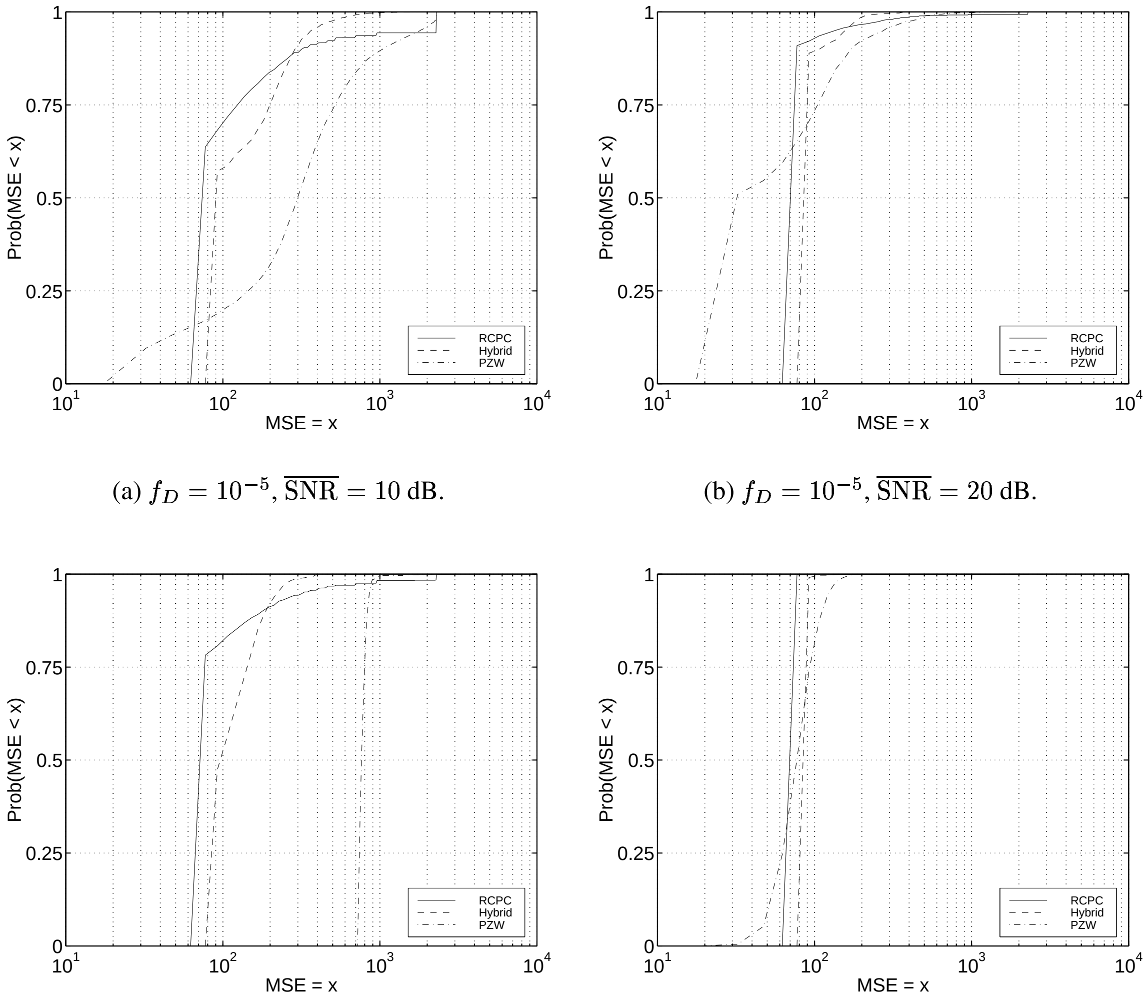
Table 4.1 Distortion values for the three algorithms over a range of channel conditions. HYBR and RCPC were optimized for the the channel with average SNR of 13 dB and normalized Doppler spread of 0.0001. PZW is the packetized zerotree wavelet scheme [70]. RCPC is the coder in [19]. HYBR is the proposed combined scheme. Column A shows the mean decoded MSE and column B shows a weighted mean using the weighting function from Example (ii): $W(x) = u(-x + 818.6)$.

			Average Received SNR(dB)							
			10		12		15		20	
			A	B	A	B	A	B	A	B
Normalized Doppler ($\times 10^{-6}$)	10	HYBR	160.5	158.6	136.6	135.4	115.7	114.5	96.9	96.6
		PZW	458.4	359.4	290.4	251.1	158.7	149.6	92.0	90.0
		RCPC	241.0	165.5	196.0	142.8	146.6	120.8	110.3	100.8
	50	HYBR	181.9	181.7	147.5	147.2	112.7	112.5	91.4	91.1
		PZW	391.6	378.9	303.2	300.4	150.9	150.7	94.6	94.4
		RCPC	244.4	188.1	166.0	139.7	123.1	109.4	93.1	91.2
	100	HYBR	163.3	163.0	122.0	121.7	99.6	99.3	89.0	88.7
		PZW	381.2	379.3	265.1	264.5	158.3	157.5	76.8	76.6
		RCPC	180.1	141.4	126.7	113.3	100.8	97.1	91.4	90.2
	200	HYBR	126.4	126.2	103.8	103.6	92.3	92.0	88.2	87.9
		PZW	784.7	774.6	278.9	278.7	169.6	169.4	87.2	87.0
		RCPC	131.8	119.1	106.6	99.6	96.0	92.7	89.7	89.3

4.3.2.3 Cumulative distributions of MSE

Further information on performance is provided in Figures 4.15(a) - 4.15(d). The plots show the cumulative distributions of the decoded MSE for the four channels whose parameter values are at the corners of Table 4.1. Curves closer to the left and top sides of the plot have better performance in this type of graph.

The initial packet losses for the SPIHT+RCPC/CRC coder previously mentioned are visible as a relatively high tail for the distribution in 4.15(a). Even though fewer than 10% of the decoded images have these large distortions (high MSE), the very large MSE values have a considerable effect on the mean MSE. Notice the performance generally improves for higher received SNRs (due to fewer channel errors) as well as for faster channels because the interleaver is more effective. By contrast, the PZW coder performance degrades for faster channels



(a) $f_D = 10^{-5}$, $\overline{\text{SNR}} = 10$ dB.

(b) $f_D = 10^{-5}$, $\overline{\text{SNR}} = 20$ dB.

(c) $f_D = 2 \times 10^{-4}$, $\overline{\text{SNR}} = 10$ dB.

(d) $f_D = 2 \times 10^{-4}$, $\overline{\text{SNR}} = 20$ dB.

Figure 4.15 Cumulative distributions of decoded MSE for the 512×512 Lena image transmitted over Rayleigh fading channels with a transmission rate of 0.25 bpp.

because the errors are less bursty, so more packets are lost for a given average error rate. In addition, because the effective visual performance in the high distortion regions is worse for the SPIHT+RCPC/CRC algorithm (Figures 4.14 and 4.17), we see that the overall performance of the hybrid is superior over the channels of interest.

4.3.2.4 With packet erasures

The next set of results includes the effects of the packet erasure channel. Table 4.2 compares the algorithms' performance over varying packet erasure channels (the wireless channel parameters are fixed). Notice that the SPIHT+RCPC/CRC performs well for a packet burst length of 10 (using either mean MSE or weighted mean MSE). The reason for the improved performance on bursty erasure channels is due to the higher likelihood of the first packet erasure occurring late in the transmission for a fixed erasure rate. Both PZW and the hybrid coders show robust performance over the varying channels, but the hybrid is able to produce higher quality images on average.

Table 4.2 Distortion values for the three algorithms over a range of channel conditions. For these channels, the fading parameters were held constant while the packet erasure channel parameters were varied. HYBR and RCPC were optimized for the channel with average SNR of 13 dB and normalized Doppler spread of 0.0001. PZW is the packetized zerotree wavelet scheme [70]. RCPC is the coder in [19]. HYBR is the proposed combined scheme. Column A shows the mean decoded MSE and column B shows a weighted mean using the weighting function from Example (ii): $W(x) = u(-x + 818.6)$.

			Burst Length			
			1		10	
			A	B	A	B
Packet Erasure Rate	0.01	HYBR	137.6	137.3	136.6	136.4
		PZW	273.0	272.6	274.0	273.6
		RCPC	195.2	167.9	137.5	121.5
	0.10	HYBR	206.9	206.7	200.5	200.1
		PZW	345.7	345.3	340.8	340.2
		RCPC	614.5	439.1	208.0	174.2

Figures 4.16(a) - 4.16(d) show the cumulative distributions of performance for these different packet erasure conditions. Numerically the hybrid performs competitively with the SPIHT+RCPC/CRC. After considering the fact that in the high distortion regime the hybrid visual performance is better than its numerical results might suggest (based on results in [74]), we see that the hybrid coder is superior over this range of channels. Visual results over two particular channels are shown in Figure 4.17. We see that the SPIHT+RCPC/CRC coder has strong performance over the channel dominated by bit errors, but performance is significantly degraded on the channel dominated by packet erasures. The PZW and hybrid schemes show consistent performance over both channels. But because of the additional error-correction capability, the hybrid decodes images at a higher quality on average.

4.3.3 Conclusion

In many applications, the system must operate in a highly variable environment. In these cases, the source and channel coders must be able to handle a large range of potential conditions. In addition, severe channels can lead to large variations in decoded image quality over different trials, making it difficult to decisively conclude which coding method is superior. Using the mean decoded MSE as well as cumulative distribution plots and the visual results obtained by this research, we conclude that the hybrid coder performs competitively across all channel conditions and degrades more gracefully under the most severe conditions.

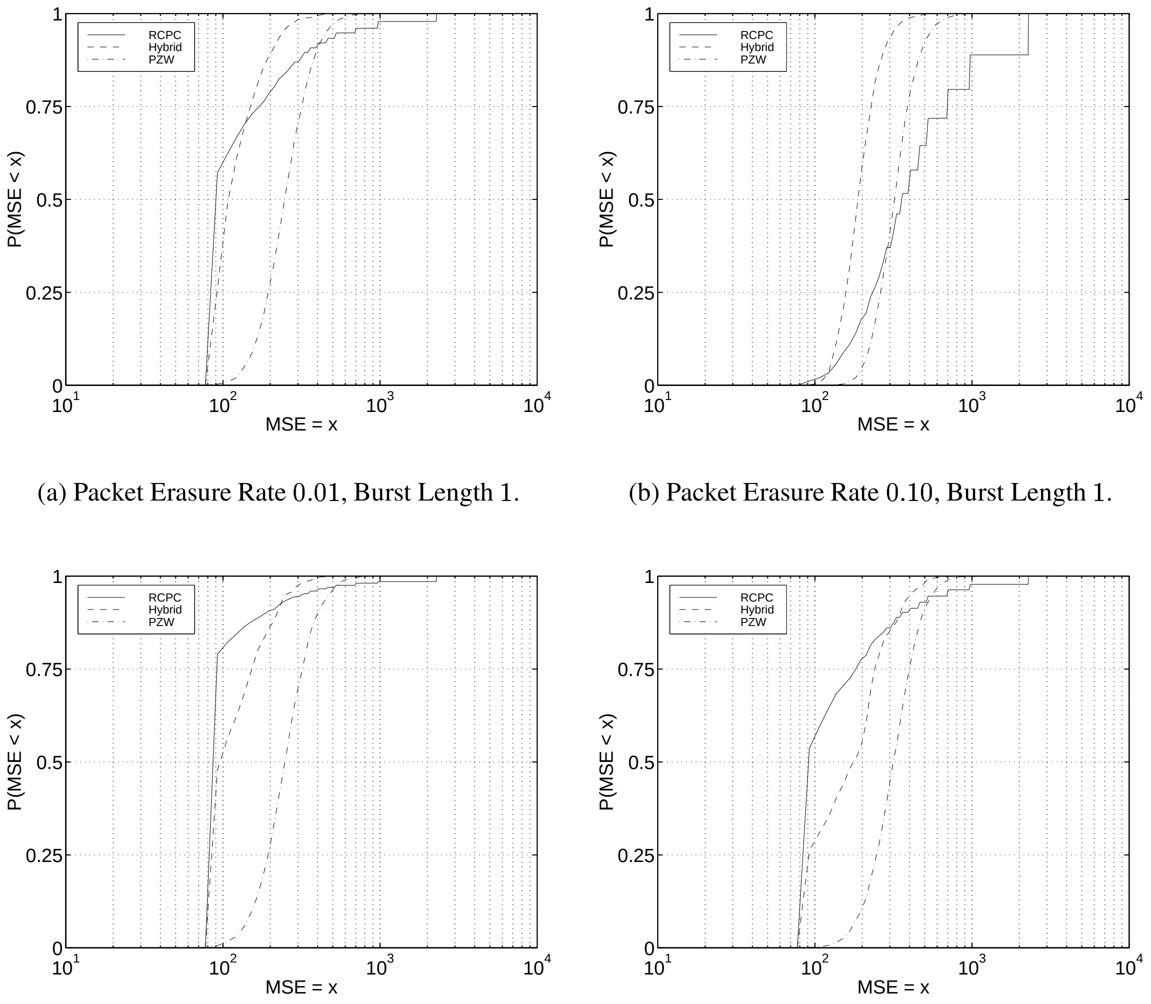
4.3.4 Code Parameters

This appendix lists the channel code parameters used in Section 4.3. All packets consisted of 384 source bits. The polynomials below will be expressed in octal notation (e.g., octal 13 is 001011 in binary which translates to the polynomial $X^3 + X + 1$).

All codes used a 16-bit CRC defined by the polynomial 254465. All RCPC codes in this paper were constructed from memory 6 mother codes, and each packet was terminated with enough zero bits to flush the state of the convolutional coder (i.e., 6 bits in this case). The search for the correct path in the list-Viterbi algorithm was terminated after 100 candidates as in [28]. However, note that limiting the search depth to 10 candidates gives almost the same performance. The hybrid coder used the rate $4/9$ RCPC code and the SPIHT+RCPC/CRC used the rate $4/13$ RCPC code. The puncturing matrices and mother codes are listed in Table 4.3. A convolutional bit interleaver was used for the hybrid and SPIHT+RCPC/CRC coders for transmission over the fading portion of the channel. This type of interleaver operates in a more continuous fashion than does a block interleaver, so it can be more easily matched to any total transmission rate. There is a penalty on the order of n^2 bits for an interleaver of depth n as initial zero bits in the memory are flushed. Increasing the interleaver depth tends to make the de-interleaved errors more uniform (helping the decoding performance of the RCPC codes), but there is a penalty in the total number of source bits received for any total transmission rate. The interleaving depth for the hybrid coder was 70 and for the SPIHT+RCPC/CRC code was 70.

Table 4.3 RCPC codes.

Rate	Mother Code	Puncturing Matrix	Rate	Mother Code	Puncturing Matrix
$4/9$	155	1 1 1 1 1 1 1 1	$4/13$	155	1 1 1 1 1 1 1 1
	123	1 1 0 1 1 1 1 0		123	1 1 1 1 1 1 1 1
	137	0 1 0 1 0 1 1 0		137	1 1 1 1 1 1 1 1
	147	0 0 0 0 0 0 0 0		147	0 1 0 0 0 1 0 0



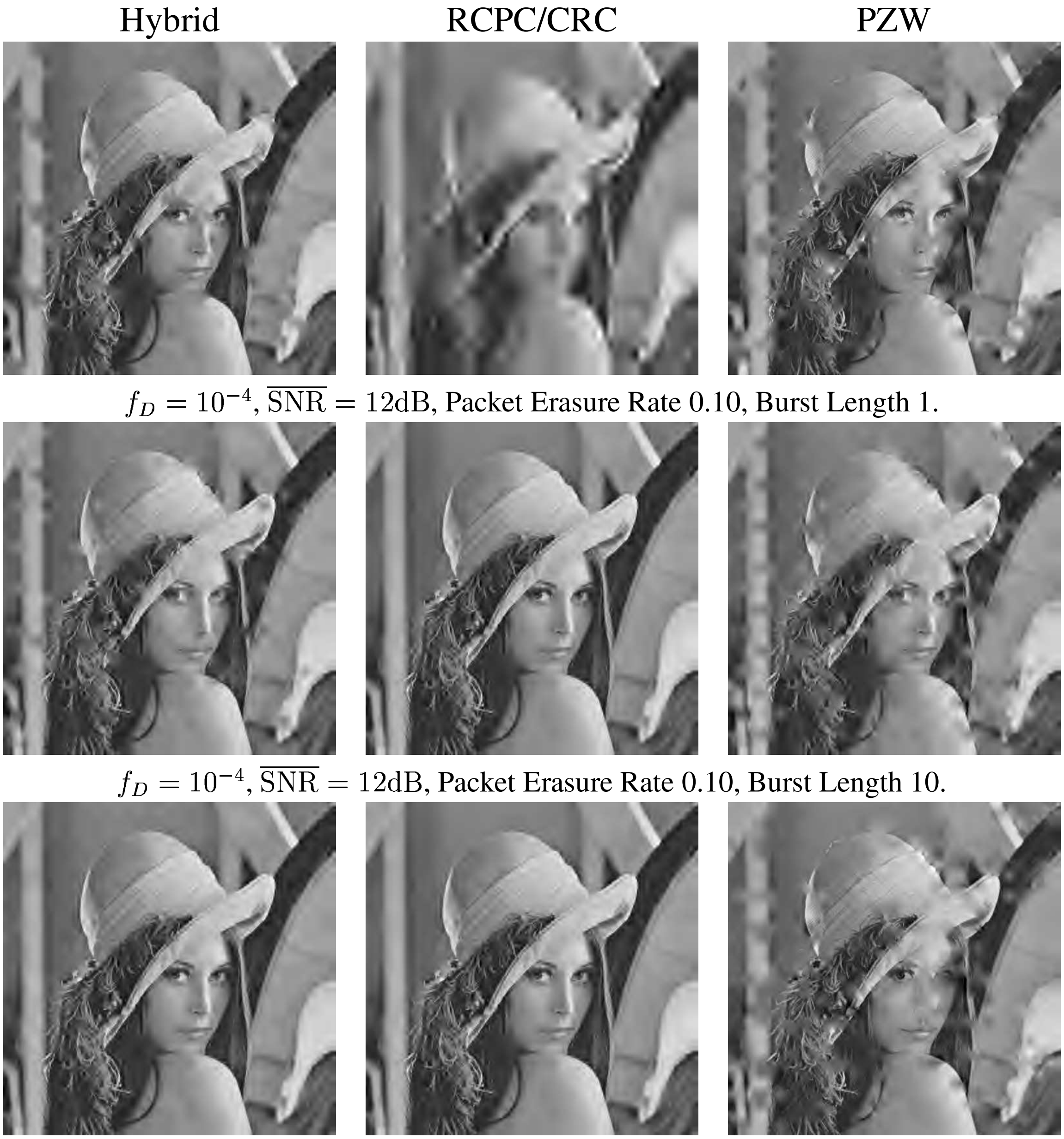
(a) Packet Erasure Rate 0.01, Burst Length 1.

(b) Packet Erasure Rate 0.10, Burst Length 1.

(c) Packet Erasure Rate 0.01, Burst Length 10.

(d) Packet Erasure Rate 0.10, Burst Length 10.

Figure 4.16 Cumulative distributions of decoded MSE for the 512×512 Lena image over channels with varying packet erasure parameters and fading parameters $f_D = 10^{-4}$, $\overline{\mathrm{SNR}} = 12$ dB. (Total transmission rate 0.25 bpp.)



Hybrid RCPC/CRC PZW

$f_D = 10^{-4}$, $\overline{\mathrm{SNR}} = 12$dB, Packet Erasure Rate 0.10, Burst Length 1.

$f_D = 10^{-4}$, $\overline{\mathrm{SNR}} = 12$dB, Packet Erasure Rate 0.10, Burst Length 10.

$f_D = 10^{-5}$, $\overline{\mathrm{SNR}} = 10$dB, No packet erasure.

Figure 4.17 Images displayed here show the median quality for the three algorithms under different channel conditions. The channel used for the top row of images was dominated by uniform packet erasures. The second row includes representative images over a channel with packet erasures occurring in bursts. The channel for the bottom row of images has no packet erasures but has a higher probability of bit errors, which occur in long bursts. The overall transmission rate was 0.25 bpp.

CHAPTER 5

IMAGE CODING FOR ERASURE CHANNELS

5.1 Introduction

In Section 4.3, the SPIHT image coder was modified so that each packet was independently decodable in order to improve the robustness over varying channel conditions. This change allowed the decoder to use each correctly received packet. The decodability of each packet is guaranteed since the the algorithm partitions the wavelet domain into wavelet trees and places the data into packets so that the bits associated with each wavelet tree are fully contained in a single packet. There is some loss of performance relative to the original SPIHT algorithm (i.e., it requires more bits to achieve the same PSNR) that occurs in this partitioning and packing operation. The additional rate required to achieve a given PSNR can be thought of as source redundancy to distinguish it from channel code redundancy, but both are intended to provide some error resiliency. The problem is that the algorithm does not offer a method of tuning the amount of source redundancy. Ideally, the redundancy would be selectable similar to the way the channel code rate can be selected to match the set of channel conditions.

When channel conditions can vary over a wide range, the designer is forced to provide channel coding for the worst case channel conditions which usually significantly sacrifices performance under good channel conditions. This motivates the desire for a more robust image coder which does not sacrifice too much compression performance. This chapter presents methods for introducing a tunable amount of source redundancy as a way of improving robustness to errors. The erasure channel mentioned in the title of the chapter can really be considered an abstraction of the actual channel as might be seen by an application at a higher layer in the network protocol stack. If the channel is not actually an erasure channel, it is simply necessary that channel decoding failures can be detected reliably such as with the RCPC/CRC codes presented in Chapter 3. However, the erasure channel applies directly to packet networks where packets are lost due to misrouting, queue overflow, or excessive delay.

Section 5.2 presents a multistage structure for image coding with results from [38]. Some analysis of the multistage encoder and decoder is provided in Section 5.3. Finally, a method employing a multidimensional extension of a multiple description scalar quantizer [45] is presented in Section 5.4.

5.2 Multistage Coding

This section presents a multistage coding structure for image compression that encodes entire images at each stage rather than the traditional multistage approach of coding relatively small vectors, so we call the method “macroscopic multistage coding” to emphasize the distinction. Each stage encodes the residual of the previous stage. The process of multistage encoding typically results in additional source redundancy in the end-to-end encoded bit stream, but the amount of additional redundancy is tunable. The additive structure of the multistage decoder effectively exploits the source redundancy to provide graceful degradation in cases of lost packets or uncorrected bit errors.

In addition, any nonprogressive image compression algorithm can be made progressive (by stages), by using the algorithm at each stage in the multistage structure (although some noiseless channel performance is generally sacrificed). Whereas the traditional use of multistage coding is for complexity reduction, our purpose of multistage coding is specifically to improve robustness over a range of channel conditions and maintain progressivity. In fact, the stages in our proposed scheme can be high-performance image compression schemes themselves, in contrast to plain vector quantization used in ordinary multistage coders. Also, the transmission rate allocated to each stage can be much greater than in more traditional multistage applications. We first provide some background on multistage coding, and then describe two specific applications and present numerical results on some severe channel conditions.

The basic structure of a multistage (or "residual") coder is depicted in Figure 5.1 and is characterized by multiple encoders arranged sequentially so that the quantization error signal from each stage is the input to the next stage. In source coding, this structure is used to reduce the encoder complexity and storage requirements for vector quantization (VQ) [75]. For example, a multistage VQ (MSVQ) consisting of m stages with codebook size N_i at stage i, has search complexity (assuming greedy stage-by-stage encoding) and storage equal to $\sum_{i=1}^{m} N_i$ instead of $\prod_{i=1}^{m} N_i$ (for an equivalent full search vector quantizer). The reproduction vector is a sum of codevectors, one from each stage (i.e., $\hat{X} = \sum_{i=1}^{m} \hat{X}_i$). An example of a predictive image coder based on this structure is [76]. In [77], design algorithms are presented for channel-matched tree-structured VQ (TSVQ) and MSVQ which improve performance when operating over noisy channels. The role of the multistage structure was complexity reduction and not robustness in that work.

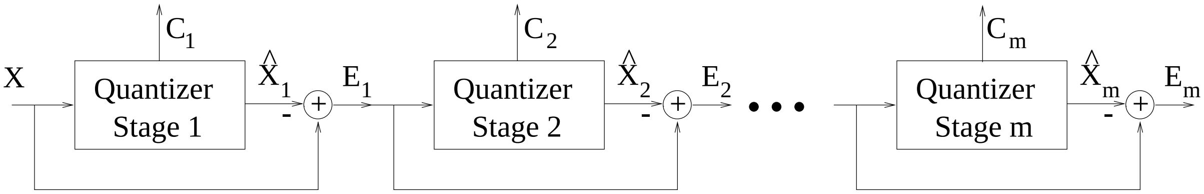


Figure 5.1 Structure of multistage (residual) source encoder.

We proposed to use at each stage in Figure 5.1 a full-image compression algorithm. This leads to an image coder with the following properties:

1. The coder is progressive at stages.

2. The coder can survive the loss or corruption of one or several stages.

3. The coder performs well if no stages are lost.

The multistage structure can be applied not only to images but to other sources as well. Moreover, there is freedom to allocate the transmission rate among the stages and even to alter the encoding algorithm for each stage. In a more general variation of multistage coding, each stage can depend on the encoded bit streams of previous stages. In this paper, we do not specifically consider this variation, but we note that it offers the possibility of better source coding performance at the expense of increased error sensitivity.

One reason that multistage encoders provide robustness to channel noise is the insertion of known resynchronization points. The transmitted bit stream has multiple independently decodable components which allows all correctly received stages to be used in the reconstruction. As with most other progressive coders, the stages typically do not have equal influences on the decoded quality. Instead, importance essentially decreases monotonically with stage number. This nonuniform importance allows more efficient channel coding through the use of UEP.

An interesting feature of our proposed multistage coder is that it simultaneously has progressive, embedded, and nonembedded coding characteristics. As noted in Section 5.2, the multistage structure naturally induces progressivity whether or not the component coders have this property. Each stage has a constrained transmission rate (by design) regardless of whether the stage encoder is embedded or not. At the same time, the macroscopic multistage structure results in some degree of embedded encoding of the source independent of the type of component source coders. Only the coding in the final stage within the transmission rate constraint changes if the transmission rate does not correspond to the end of a stage.

While the multistage structure allows arbitrary source coders to be used at each stage, there is no guarantee that cascading a particular coder will result in good performance. In fact, if a good image coding algorithm is used in the initial stages, the input to later stages typically deviates statistically from that of natural images. Therefore, using the same off-the-shelf image coding algorithm for each stage will not necessarily work well. In Sections 5.2.1 and 5.2.2 , two specific examples of multistage coders are discussed. We use wavelet-based coders because of their good single-stage performance and discuss modifications to improve the performance of a multistage implementation.

5.2.1 A multistage zerotree wavelet coder

This section describes a multistage coder based on zerotree wavelet coding. Under known and memoryless channel conditions, such as a BSC, it has been shown that channel coding can effectively protect the SPIHT bit stream and that the combined system has performance superior to most other known methods [19, 28]. However, when channel conditions are worse than the design conditions, this method does not degrade gracefully [37].

One approach to improving the robustness of the EZW or SPIHT coders is to exploit the decreasing importance of the bit stream, as the rate increases, by using UEP channel coding. However, in practice, this results in little improvement due to the combination of the serial decoding requirement of the algorithm and the rapid transition in performance of channel codes as a function of the channel code rate (e.g., see [14] and [19]). The multistage coding structure reduces the serial decoding requirement by only requiring an uninterrupted bit stream within a stage. The stages tend to have unequal importance (in terms of the decoded image quality) with the first stages being the most important. However, in this case UEP coding is more effective since it is possible to continue decoding beyond the first channel error. A multistage version of the SPIHT coder is discussed in the remainder of this section.

5.2.1.1 The stage schedule

We refer to a "stage schedule" as the ordered list of transmission rates assigned to the stages of a multistage coder. The rate of the last stage is usually left to absorb any remaining bits in the rate assignment procedure. We assume that the stage schedule is fixed and known at both the encoder and decoder. A possible direction of future research is to study the adaptation of the stage schedule based on the image and the channel conditions, in which case the schedule could be transmitted as overhead.

The maximum stage length (i.e., the number of packets or total rate in a stage) is constrained by the serial decoding requirement within a stage and the effective packet error rate, after the channel decoder, under the worst channel conditions of interest. If the probability of a packet error is p under these conditions, then there is little reason to create a stage much longer than $1/p$ packets since error-free packets beyond the first packet error are discarded within a stage. While each additional stage can offer improvement in robustness under poor channel conditions, the performance typically decreases under good channel conditions. As the probability of packet error under the worst channel conditions drives the maximum stage length down, loss in source coding performance under good conditions leads to the necessity of finding methods to improve performance. The trade-off between performance on good channels and on bad channels is determined by our choice of stage schedules, the source coders used at each stage, and the channel coding for each stage.

5.2.1.2 Modifications to improve multistage performance

While the SPIHT algorithm performs well on natural images it is not particularly suited for coding residual images. The residual image at the output of the first stage (coded by SPIHT) tends to have coefficients in lower frequency subbands with peak magnitudes about the same as those in higher frequency subbands. Also, the state information of SPIHT is lost, so extra bits must be spent at the start of each stage to describe the locations of the large coefficients. (In ordinary SPIHT this is implicit in the bit stream.) One way to reduce the performance loss in using SPIHT at every stage of a multistage encoder is to use fewer levels of wavelet decomposition in later stages. The motivation for this approach is that the coefficients in the coarser scales have generally been refined beyond their most significant bit and most of the tree-structured dependency has been exploited at these scales. Also the large coefficients at fine scales would not be as deeply nested in the tree-structure, so fewer bits are needed to code their locations. This method is useful regardless of the stage length (i.e., the transmission rate of a stage), but it is not sufficient to completely remove the problem associated with very short stages.

A problem associated with using very short stages is that extra bits are spent coding coefficients that have already been refined in earlier stages since the state information is reset at the beginning of the stage. This is similar to using a repetition channel code where the code rate is too small (i.e., it is over-coded). An effective solution to this problem is to partition the image within a stage into subimages (see Section 2.2.2). This method allows the stage length to increase because serial decoding is only required within a subimage. The visual quality of the reconstructed image can be improved under packet loss conditions by ensuring the subimages consist of spatially diverse sets of wavelet trees. This way, the distortion due to packet loss will be distributed throughout the image. This goal is accomplished by choosing a good ordering of the image wavelet trees prior to partitioning into subimages. The coder mentioned in Section 5.2.2 uses a similar method except the image is partitioned into many more subimages (i.e., the number of subimages equals the number of packets). Also with the multistage SPIHT coder, the partitioning only occurs in later stages with fewer levels of wavelet decomposition and only in those cases where packet loss rates significantly limit stage lengths.

5.2.1.3 Overhead information

Some overhead information is required for proper decoding of the the multistage version of SPIHT beyond that required for the original version. First the starting threshold value needs to be sent for each stage or partition. In the worst case this represents about 5 bits per stage, but can be reduced by coding differentially from a predicted value which would probably be a static value based on the stage schedule. In the experimental results that follow, the full 5-bit encoding was used.

The decoder must also be able to identify the stage and intrastage position of each correctly received packet. The amount of explicit coding required for this depends on the details of the system and the packet loss mechanism. For situations where all packets are received and the only loss is due to channel decoding failures, no extra information is necessary since the correct location of each packet is evident by counting received packets. If some packets never reach the receiver (e.g., misrouting or buffer overflow in a wireline network), then each packet must include enough overhead bits to resolve the uncertainty. For example, if the probability of N or more consecutive packet erasures is acceptably low over the channel conditions of interest, then a packet index of $\lceil \log_2(N) \rceil$ bits suffices to determine the intrastage position of each received packet. If packets can arrive out of order, the maximum delay would affect the number of bits required to resolve the correct position. In some systems packet index information is included in the packet header along with routing information regardless of the packet payload, so explicit coding by the source coder would be reduced or eliminated in this case.

The experimental results in Section 5.2.1.4 focus on systems where packet losses are due to channel decoding failures, as can happen in wireless links. Therefore no additional information is necessary to identify the correct position of a received packet. Section 5.2.2.3 considers the case where packets are lost due to misrouting or buffer overflow, as in a wireline network.

5.2.1.4 Experimental results

In this section we focus on transmission over a BSC with channel error rates between 0.01 and 0.05. The performance is measured by the mean decoded MSE, computed over at least 100 trials on a BSC of the given error rate. The channel conditions were selected to allow comparison with results in [22].

The output of each stage was protected with an RCPC/CRC concatenated channel code as described in [19]. A different RCPC rate was selected for each stage to exploit the unequal importance of their output bit streams. Each RCPC code was based on a memory 6 mother code, and a 16-bit CRC was used for error detection and as part of the list decoding process [19]. The packets consisted of 220 information bits.

The multistage coders described in this section use the version of SPIHT with arithmetic coding. Experiments indicated that arithmetic coding provides about 0.2 dB better PSNR under good channel conditions. The gain associated with arithmetic coding is less than usual because the arithmetic coder did not have time to adapt the model during the short stages (or subimages). These short stages were necessary for robustness under the severe channel conditions. It might be possible to obtain better source coding performance if the models are initialized with more accurate probability distributions. Results for two coders are provided to demonstrate two possible trade-offs between performance under the good channel conditions versus the performance under bad conditions.

The first coder consisted of three stages with the first stage length equal to 10 packets, the second stage length equal to 15 packets, and the third stage assigned the remaining rate. The number of levels of wavelet decomposition was also changed at each stage with 6 levels in the first stage, 5 levels in the second stage, and 4 levels in the third stage. Finally, for stages 2 and 3 the image was partitioned prior to coding. For stage 2, the wavelet trees were divided uniformly into three sets, and each set was coded independently to produce five packets each. Similarly, in stage 3, the set of all wavelet trees was divided uniformly to give $\lceil \text{stage packets}/4 \rceil$ partition elements with each coded to produce four packets (except perhaps the last partition element). The RCPC code rates for the three stages were $1/2$, $4/7$, and $2/3$ respectively. The second multistage coder had essentially the same structure except the channel code rates were lower for the last two stages. The RCPC rates for coder 2 were $1/2$, $8/15$, and $4/7$.

Figure 5.2 shows the performance of the two multistage coders as well as the results from [22] for comparison. We note that the results from [22] used different channel coding than the coders presented here (i.e., RCPC coding alone versus RCPC/CRC), so those results would probably improve if the same RCPC/CRC were used. However, the comparison shows that both multistage methods are performing well. To demonstrate the visual quality, two example decoded images from multistage coder 2 are shown in Figures 5.3(a)-5.3(b) for channel bit error rates (BERs) 0.01 and 0.05.

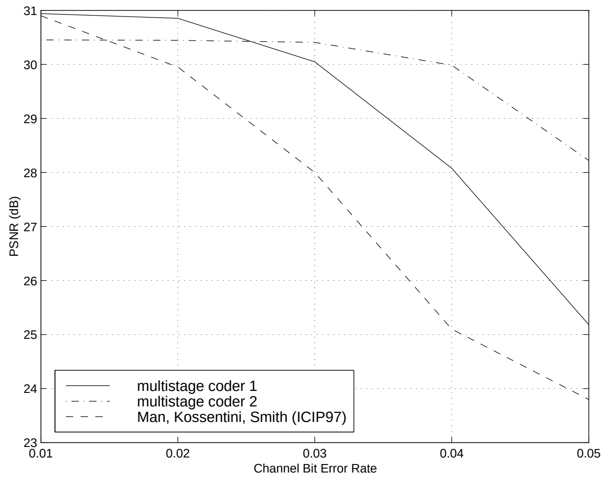
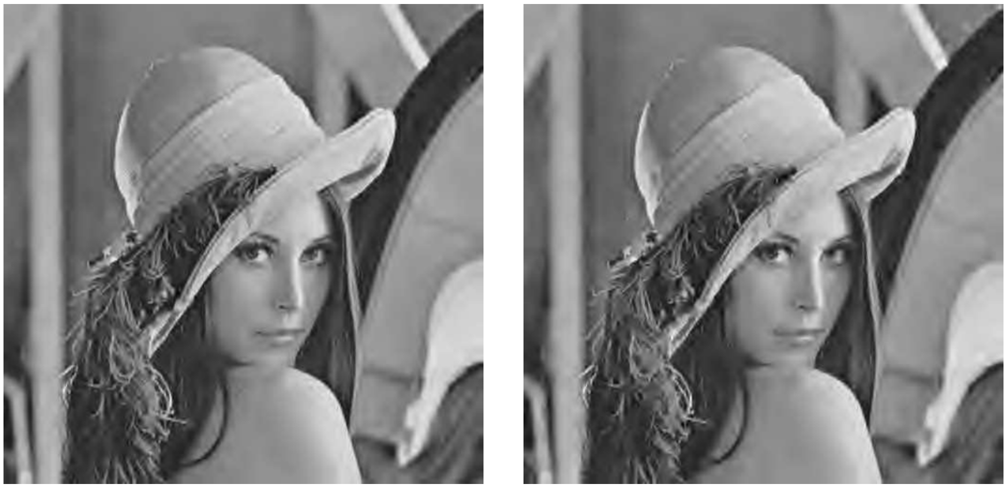


Figure 5.2 Comparison of two multistage coders with results from [22] for the 512×512 Lena image at 0.25 bpp total transmission rate sent over a BSC with different error rates.

5.2.2 A packetized multistage image coder

This section describes a multistage version of the PZW coder presented in [71]. The PZW coder modifies zerotree coding in order to be more robust to lost packets due to buffer over-



(a) BER=0.01, PSNR = 30.42 dB.

(b) BER=0.05. PSNR = 29.24 dB.

Figure 5.3 Example decoded images for multistage coder 2 for channel BER $= 0.01$ and 0.05.

flow or misrouting (i.e., lost packets never reach the receiver). Specifically, PZW partitions an image into subimages as explained in Section 2.2.2 and encodes each subimage with a zerotree algorithm so that the bits exactly fill one packet. The order of the wavelet trees is fixed according to a dither pattern matrix (selected to reduce the visual effect of lost packets) and the exact image partition is determined for each input image. Header information is included with each packet to allow independent decoding.

Our multistage modification of PZW obtains improved robustness by distributing information about any spatial region into multiple packets (instead of just one) and by introducing unequal packet importance which allows improved performance by exploiting efficient erasure correction coding. As a side benefit, the multistage PZW has some progressivity.

Each stage generally uses a different number of levels of wavelet decomposition similar to the multistage SPIHT coders in Section 5.2.1, with the number of levels typically decreasing monotonically with stage number. As previously mentioned, changing the number of levels allows improved coding of the residual images by the zerotree algorithm. A random permutation is selected for each stage and is fixed at the encoder and decoder to determine the order of wavelet trees for the stage. Random permutations at each stage sufficiently create spatially diverse subimages as well and reduce the chance that two spatial regions are in the same subimage at two different stages. The specific permutations selected slightly affect the compression performance of the algorithm, but experimental trials indicate that this affects the decoded PSNR by less than 0.1 dB for typical coding rates and 0.2-0.3 dB for short initial stages. The effect of the specific permutations also diminishes as more stages are used, which is due in part to selecting good partitions at each stage.

We partition the set of all wavelet trees in an image, ordered according to the permutation, so the number of subimages equals the number of packets for that stage. Since the encoded subimage length is small and each stage is partitioned prior to encoding, selecting a good partition is more important to performance than with the multistage SPIHT coder. Because the partition is not fixed, each packet contains information about the trees contained within it. To reduce the dependency between the packets, the partition information is encoded so that the set of trees in a packet can be determined without information from other packets. This amounts to encoding the starting tree index which requires $\log_2(M)$ bits (where M is the number of wavelet trees) and the number of trees in the packet (usually 4-5 bits is sufficient). The next section considers the problem of finding a good partition.

5.2.2.1 Finding a good image partition

As previously mentioned, the importance of finding a good partition of the image at each stage increases as the numbers of partitions and stages increase. The process of independently encoding subimages results in some variation in the rate allocated to individual wavelet trees compared to the case without image partitioning, which usually results in a loss of performance. It is thus of interest to find the best partition of wavelet trees into subimages which are each encoded with a rate constraint corresponding to the packet length. Typically one would minimize the reconstructed MSE, but other distortion measures can also be used.

This problem can be solved with dynamic programming [63] by noting that given n wavelet trees are assigned to the first m subimages, the best partition of the remaining trees is independent of the specific initial partition. Therefore, the cumulative number of trees assigned to subimages can serve as the state in a trellis diagram, and the subimage (or packet in this case) number can serve as the trellis stage index. Typically, there are further constraints imposed, such as the minimum and maximum number of trees per subimage, which limit the search space. Figure 5.4 shows an example trellis diagram with constraints on minimum and maximum trees per packet. The constraints limit the search region to the shaded area shown in the figure.

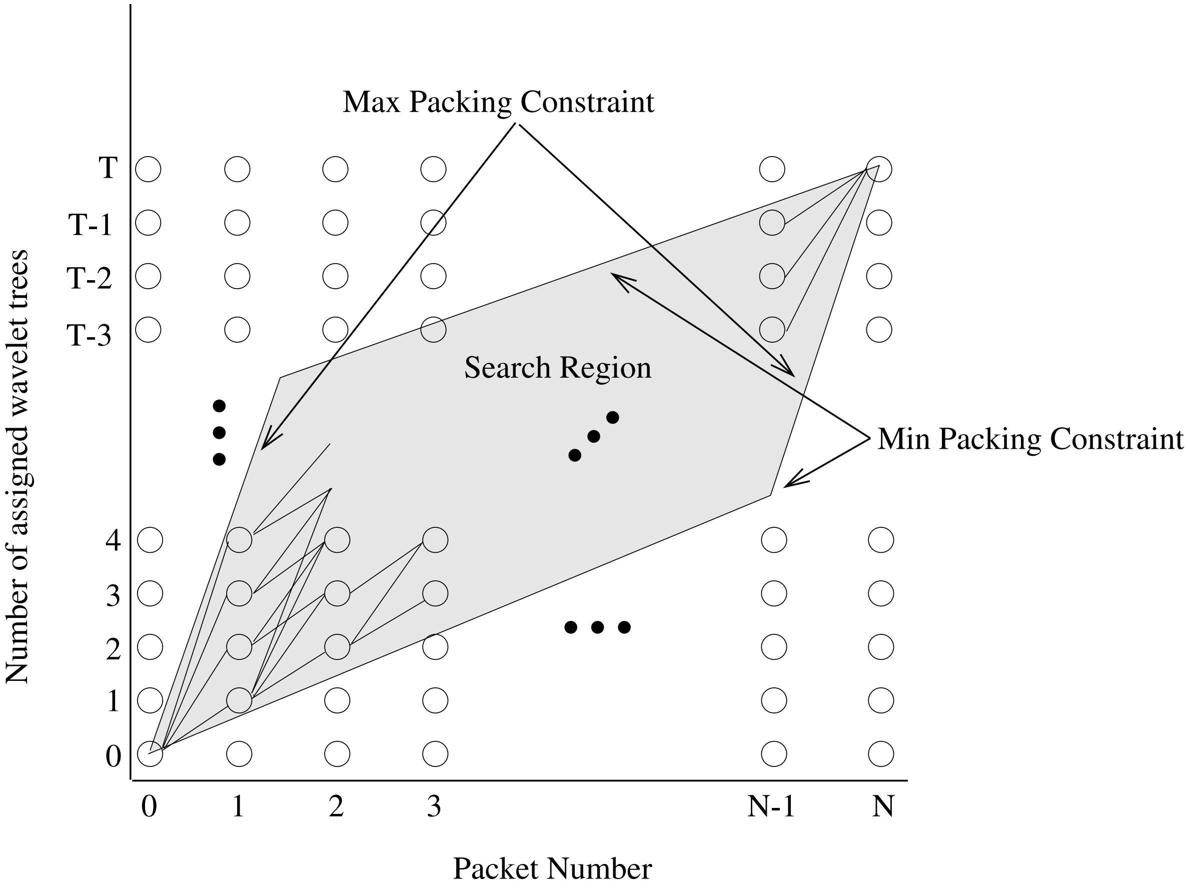


Figure 5.4 Trellis diagram for determining the optimal partitioning of wavelet trees into packets.

The distortion function for each subimage is a complicated function of the packet size and the set of wavelet trees in this case, but the values can be determined by running the encoding algorithm. Partial distortion values associated with a particular set of wavelet trees are used several times during the dynamic programming algorithm, so complexity can be reduced by storing these values in a table for later use. Also, simpler but coarser approximations to the desired metric can be used to trade off complexity and performance within the same dynamic programming framework.

Our multistage PZW coder uses this dynamic programming approach to optimize the partition selected at each stage. Experiments indicate that MSE optimized partitions in later stages can recover some of the performance loss in earlier stages caused by partitioning.

5.2.2.2 Reducing distortion at the decoder

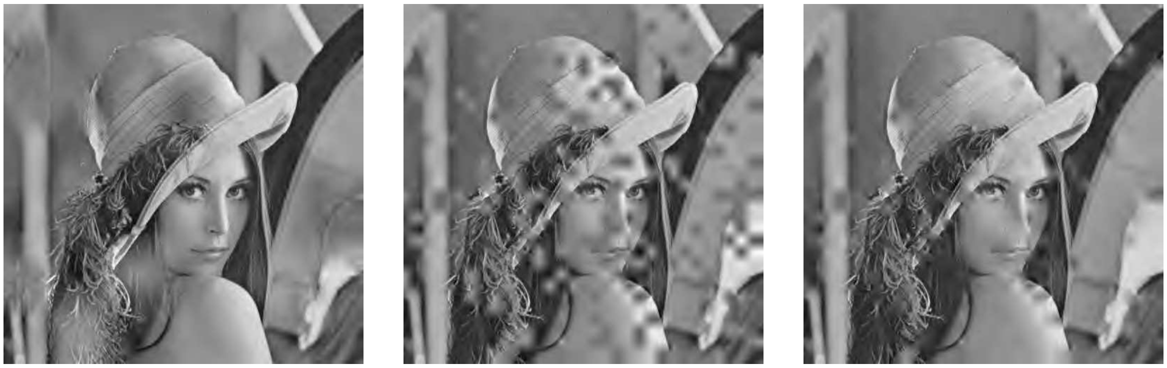
Packet erasures cause distortions in the spatial locations corresponding to the wavelet trees within the discarded packets. In PZW [71], the decoded distortion is reduced by interpolating missing wavelet tree roots from neighboring coefficients in the low-frequency subband.

Figures 5.5(a) and 5.5(b) show examples of the typical distortions that result due to missing tree information. Missing trees at coarser scales cause distortions over larger spatial regions and interpolation is not effective. At finer scales, the missing trees cause large distortions in small spatial areas and the distortion can typically be reduced by interpolation. The third image in Figure 5.5(c) shows a smoothed version of the image in Figure 5.5(b), where the improvement in quality can easily be seen, and numerically the PSNR is improved from 21.21 to 25.53 dB.

Interpolation is really only effective in the multistage PZW for reducing distortions caused by lost packets from the first stage. Also if the missing wavelet tree roots are from a coarse scale (i.e., many transform levels), then interpolating the value from neighbors does not work better than using the low-frequency subband mean (transmitted separately). However, using an initial stage with many levels of decomposition can be useful for robustness when UEP channel coding is feasible. Also if a short first stage is desirable for the purpose of increasing progressivity, then a coarse scale can be useful for achieving good source coding performance within that stage.

5.2.2.3 Experimental results

To test the performance of the multistage version of PZW, the channel was selected to match the conditions in [71]. Packets were erased independently with a probability corresponding to



(a) Effect of missing trees at coarse scale.

(b) Effect of missing trees at finer scale.

(c) Same image as (b) after smoothing.

Figure 5.5 Example images demonstrating the effect of erased trees on decoded image quality.

the given packet loss rate, the packet size was 384 bits, and the total transmission rate was 0.209 bpp.

In this channel model, lost packets do not arrive at the decoder (or they dropped due to errors or excessive delay), and it is assumed that the source coder must provide explicit coding to determine a packet's position in the transmitted sequence if necessary. Therefore, overhead bits are sent with each packet to identify the stage number, which consisted of one bit (i.e., two stages) for the coders in this section.

The PZW algorithm distributes the image information as uniformly as possible among the packets so the maximum loss associated with a packet erasure is minimized. For the multi-stage coder, the packets have unequal importance due to the progressive nature of the encoding structure. Therefore explicit erasure coding is used to protect the important first stage packets. Specifically Reed-Solomon (RS) codes are used across packets (similar to the method in [33]) so that erased packets can be recovered if the capabilities of the code are not exceeded. This method allows more levels of wavelet decomposition in the first stage for better source coding efficiency while alleviating the problem that smoothing algorithms would be ineffective (as mentioned in Section 5.2.2.2).

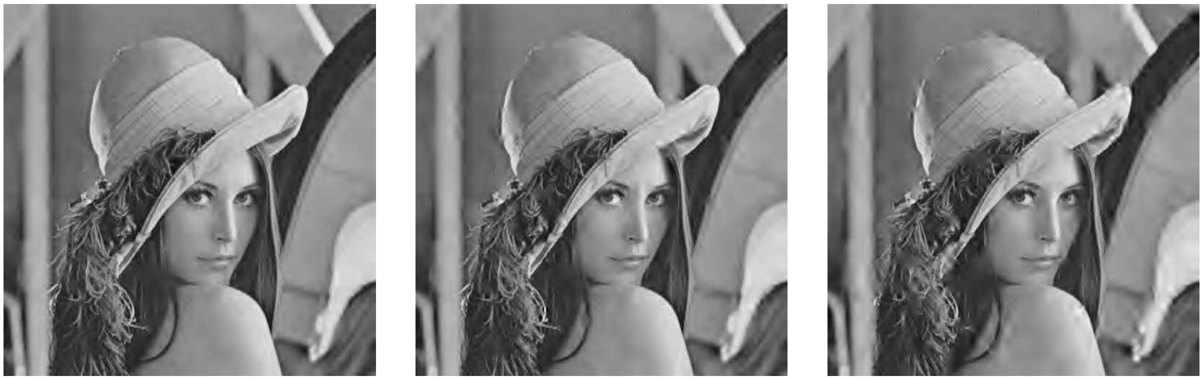
In order to correctly decode the RS code on the first stage packets, the positions of the missing packets must be identified. Therefore a packet index needs to be sent as overhead so the position within the stage can be identified for each received packet. The packet index bits and the starting tree number contained within each packet for partition information are somewhat redundant. The starting tree can be predicted from the packet index and stage schedule and only the error needs to be coded. Also, the starting tree index of the first packet is redundant since it is always zero, and only the starting tree index is necessary in the final stage packet since the remainder of the trees must be included.

The multistage encoder uses a straightforward encoding of header bits without the index prediction. It is composed of two stages with a first-stage length of 12 packets and 6 levels of wavelet decomposition followed by a second stage using 4 levels of decomposition. The 12 first-stage packets are protected by a systematic (20,12) RS code which was selected to provide good performance at the highest erasure rate of interest, 0.2, while not sacrificing error-free performance too much. Using this RS code at the highest erasure rate, the first-stage packets are available for decoding more than 99% of the time. The overhead in each packet for this coder includes 1 bit for the stage number, 5 bits for packet index (only in first stage), 5 bits for starting threshold value, 6 bits first-stage and 10 bits second-stage for the starting tree index, and 4 bits first-stage and 5 bits second-stage for the tree count. Table 5.1 shows performance results for the multistage coder and PZW over a range of packet erasure rates. The values in the table are mean decoded MSE converted to PSNR, and for the multistage coder, they are representative of the range of values that occurs as different permutations are selected for each stage's tree order (i.e., the values in the table are within about 0.2 dB of the performance of a randomly selected permutation set).

Table 5.1 Performance results for 512×512 Lena sent over a packet erasure channel. Values are PSNR (dB) converted from mean MSE.

Coder	No erasures	1% erasures	10% erasures	20% erasures
PZW[71]	32.2	31.3	26.3	24.6
Multistage PZW	31.8	31.6	30.2	28.4

As the numerical results show, the multistage coder is robust even under high-loss conditions while only sacrificing about 0.4 dB in PSNR under lossless channel conditions. These results give an indication of the performance possible with the multistage version of PZW. Implementing more efficient encoding of the packet overhead can yield an improvement of about 0.3 dB in PSNR. Also, the coder can be adjusted for a different trade-off between lossless and high-loss channel conditions. Example decoded images are provided in Figures 5.6(a)-5.6(c) for three erasure rates to demonstrate the visual quality.



(a) No erasures. PSNR = 31.96 dB.

(b) 10% erasures. PSNR = 30.50 dB.

(c) 20% erasures. PSNR = 28.45 dB.

Figure 5.6 Example decoded images for multistage PZW under three erasure conditions.

5.3 Multistage Analysis

The general multistage coding structure is shown in Figure 5.7. The functional blocks $f_i(x, y)$ represent general residual computation functions which are an abstraction of the common technique of subtracting the estimate from the stage input. The purpose of using alternate functions here is to propagate source information further down the encoding process to improve robustness in the event of stage loss during transmission. More details on these residual functions will be deferred until a later section.

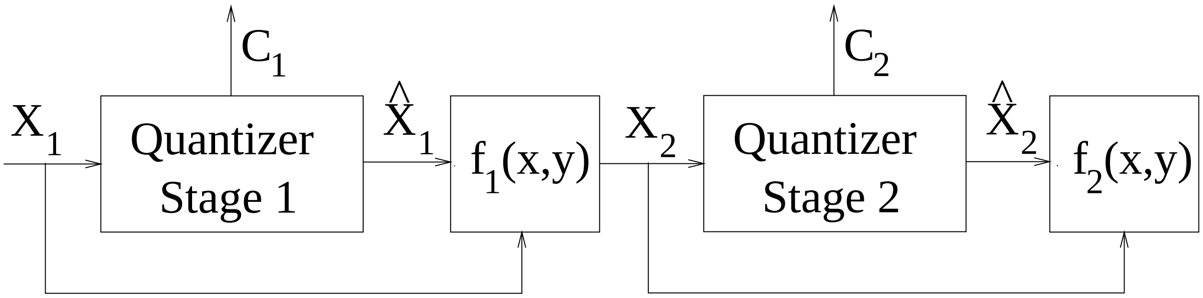


Figure 5.7 Block diagram of a general multistage encoder for the case of two stages.

The basic idea of the multistage structure is to code the image in separate pieces, each with a particular bit budget, and combine the individual stage estimates to create the final reconstruction. Such an approach has been suggested in [78] as a method to improve image compression for noiseless channels. The motivation comes from the fact that an image with a combination of smooth and textured regions will cause a “best-basis” wavelet packet algorithm to compromise between matching the two types of regions. Their approach is to instead vary the basis library within each stage so the first stage can capture smooth characteristics and the latter stages can handle local textured regions.

Another benefit of multistage coding is the improved robustness to noisy channels. The multiple stages provide natural resynchronization points which allow the decoder to recover when errors occur. Furthermore, information can be propagated to later stages by not computing the full residual (i.e., using the α_i gain terms in the figure above). In the sections that follow an erasure channel model is assumed when deriving the optimal decoding rules. Therefore there are no corrupted codewords - they either arrive intact or are known to be lost. This model is not overly restrictive since it applies directly to typical packet networks and to other channels where errors can be detected with high probability via efficient error detection codes.

5.3.1 Analysis of the encoder and decoder design

In this section, the encoder and decoder are analyzed at various levels of detail to gain insight into the proper designs. Initially the details of the encoder and decoder will be hidden in order to derive the most general results given the channel model. Therefore general results will be derived along the lines of traditional vector quantizer (VQ) analysis, and then these results will be refined for more detailed models of the multistage encoder. The most general system model is shown in Figure 5.8 and consists of an encoder which outputs M separate indices followed by an index erasure channel followed by the decoder. For each $j = 1, \ldots, M$ let $\mathcal{I}_j$ be a finite set of channel symbols that can be transmitted at the j^{th} stage. The input is a k-dimensional vector, $X \in \mathbb{R}^k$, which is mapped by the encoder, $Q : \mathbb{R}^k \to \mathcal{I}_1 \times \mathcal{I}_2 \times \ldots \mathcal{I}_M = \mathcal{J}$, into a vector of indices $\Lambda \in \mathcal{J}$. The model is essentially the same as that of the generalized multiple description problem where each index represents a description of the source. Much of the previous work on multiple description has considered the special case of two descriptions, but the intended application makes the consideration of an arbitrary number of descriptions desirable.

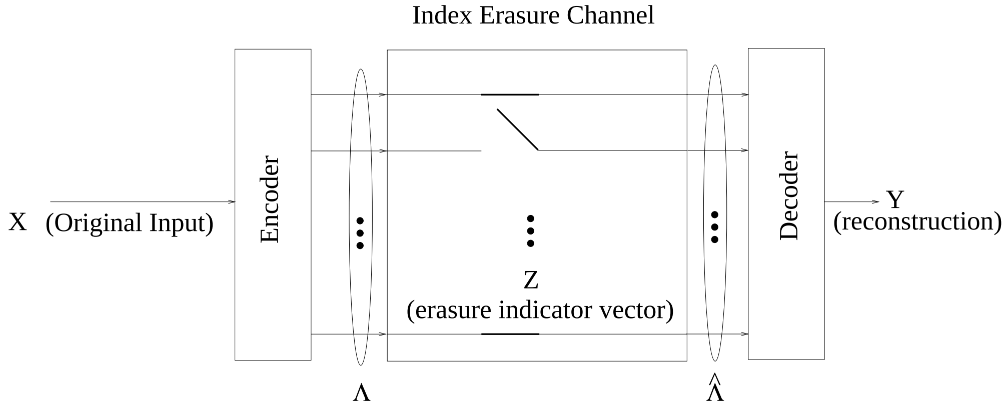


Figure 5.8 Abstract transmission model.

The channel operates by either passing the input index (i.e., the particular component of Λ) without error or by erasing the index. The channel state is fully specified by the index erasure indicator vector $Z \in \{0,1\}^M$ where the components equal to 1 correspond to the erased components of Λ. The components of the received vector $\widehat{\Lambda}$ are elements of the encoder index sets augmented by the erasure symbol e, $\widehat{\Lambda}_i \in \mathcal{I}_i \cup \{e\} \equiv \widehat{\mathcal{I}}_i$, and $\widehat{\Lambda} \in \widehat{\mathcal{I}}_1 \times \widehat{\mathcal{I}}_2 \times \ldots \widehat{\mathcal{I}}_M \equiv \widehat{\mathcal{J}}$. The received vector $\widehat{\Lambda}$ is determined by a function of the encoder output Λ and the channel indicator vector Z according to the mapping $h(\lambda, z) : \mathcal{J} \times \{0,1\}^M \to \widehat{\mathcal{J}}$ where the i^{th} component of the output is

$$h(\lambda, z)_i = \begin{cases} \lambda_i & \text{if} \quad z_i = 0 \\ e & \text{if} \quad z_i = 1 \end{cases} \qquad i = 1, \ldots, M. \tag{5.1}$$

5.3.2 Optimal decoder design

With these definitions in place, the goal is to determine the decoder mapping $D : \widehat{\mathcal{J}} \to \mathbb{R}^k$ which minimizes the expected distortion $E[d(X,Y)]$ between the input X and the decoder output Y. The expected distortion can be written in terms of the possible received sequences by first writing it in terms of the encoder output and channel state

$$E[d(X,Y)] = E[E[d(X,Y)|\Lambda = \lambda, Z = z]] \tag{5.2}$$

$$= \sum_{\lambda \in \mathcal{J}} \sum_{z \in \{0,1\}^M} E[d(X,Y)|\Lambda = \lambda, Z = z] P(\Lambda = \lambda, Z = z). \tag{5.3}$$

The distortion can now be written in terms of the decoder input $\widehat{\Lambda}$ by noting that its value is completely determined by Λ and Z through the mapping $h(\Lambda, Z)$ introduced above (i.e., $\widehat{\Lambda} = h(\Lambda, Z)$). Also the mapping $g(\widehat{\lambda}) : \widehat{\mathcal{J}} \to \{0,1\}^M$ given by

$$g(\widehat{\lambda}) \equiv \begin{cases} 1 & \text{if} \quad \widehat{\lambda}_i = e \\ 0 & \text{if} \quad \text{otherwise} \end{cases} \qquad i = 1, \dots, M \tag{5.4}$$

shows that $\widehat{\Lambda}$ contains all information about Z. Furthermore the events $\{\Lambda = \lambda, Z = z\}$ and $\{\Lambda = \lambda, \widehat{\Lambda} = \widehat{\lambda}\}$ are equivalent. For notational convenience, let $\widehat{\mathcal{J}}_\lambda \equiv \{h(\lambda', z) : \lambda' = \lambda, z \in \{0,1\}^M\}$ be the set of possible decoder input vectors for the encoder output Λ, and let $\mathcal{J}_{\widehat{\lambda}} \equiv \{\lambda \in \mathcal{J} : h(\lambda, g(\widehat{\lambda})) = \widehat{\lambda}\}$ be the set of possible encoder outputs associated with decoder input $\widehat{\lambda}$. Then, the distortion can be written

$$E[d(X,Y)] = \sum_{\lambda \in \mathcal{J}} \sum_{z \in \{0,1\}^M} E[d(X,Y)|\Lambda = \lambda, \widehat{\Lambda} = \widehat{\lambda}] P(\Lambda = \lambda, \widehat{\Lambda} = \widehat{\lambda}) \tag{5.5}$$

$$= \sum_{\lambda \in \mathcal{J}} \sum_{\widehat{\lambda} \in \widehat{\mathcal{J}}_\lambda} E[d(X,Y)|\Lambda = \lambda, \widehat{\Lambda} = \widehat{\lambda}] P(\Lambda = \lambda, \widehat{\Lambda} = \widehat{\lambda}) \tag{5.6}$$

$$= \sum_{\widehat{\lambda} \in \widehat{\mathcal{J}}} P(\widehat{\Lambda} = \widehat{\lambda}) \sum_{\lambda \in \mathcal{J}_{\widehat{\lambda}}} E[d(X,Y)|\Lambda = \lambda, \widehat{\Lambda} = \widehat{\lambda}] P(\Lambda = \lambda | \widehat{\Lambda} = \widehat{\lambda}). \tag{5.7}$$

The decoder input is fixed with the innermost expectation so the value of the decoder output, which will be denoted $y_{\widehat{\lambda}}$, is simply a constant. Furthermore, the source is independent of the channel state, so the channel can be removed from the conditioning event (since $\Lambda = \lambda$ contains all the information relevant to the source) giving

$$E[d(X, y_{\widehat{\lambda}})|\Lambda = \lambda] = E[d(X,Y)|\Lambda = \lambda, \widehat{\Lambda} = \widehat{\lambda}]. \tag{5.8}$$

Substituting (5.8) into (5.7) gives

$$E[d(X,Y)] = \sum_{\widehat{\lambda}\in\widehat{\mathcal{J}}} P(\widehat{\Lambda}=\widehat{\lambda}) \sum_{\lambda\in\mathcal{J}_{\widehat{\lambda}}} E[d(X,y_{\widehat{\lambda}})|\Lambda=\lambda]P(\Lambda=\lambda|\widehat{\Lambda}=\widehat{\lambda}). \tag{5.9}$$

For a given decoder input, only one term from the outer sum in (5.9) is relevant and the optimal decoder rule is given by

$$y_{\widehat{\lambda}} = \arg\min_{y} \sum_{\lambda\in\mathcal{J}_{\widehat{\lambda}}} E[d(X,y)|\Lambda=\lambda]P(\Lambda=\lambda|\widehat{\Lambda}=\widehat{\lambda}). \tag{5.10}$$

The value of $P(\Lambda=\lambda|\widehat{\Lambda}=\widehat{\lambda})$ can be computed as follows:

$$\begin{aligned}
P(\Lambda=\lambda|\widehat{\Lambda}=\widehat{\lambda}) &= \frac{P(\Lambda=\lambda,\widehat{\Lambda}=\widehat{\lambda})}{P(\widehat{\Lambda}=\widehat{\lambda})} && (5.11)\\
&= \frac{P(\Lambda=\lambda, Z=g(\widehat{\lambda}))}{P(Z=g(\widehat{\lambda}))\sum_{\lambda'\in\mathcal{J}_{\widehat{\lambda}}} P(\Lambda=\lambda')} && (5.12)\\
&= \frac{P(\Lambda=\lambda)P(Z=g(\widehat{\lambda}))}{P(Z=g(\widehat{\lambda}))\sum_{\lambda'\in\mathcal{J}_{\widehat{\lambda}}} P(\Lambda=\lambda')} && (5.13)\\
&= \frac{P(\Lambda=\lambda)}{\sum_{\lambda'\in\mathcal{J}_{\widehat{\lambda}}} P(\Lambda=\lambda')}. && (5.14)
\end{aligned}$$

From (5.10) and (5.14) it is clear that the optimal decoder rule is independent of the channel statistics. Since the channel is an erasure channel, there is no uncertainty about the channel state given the decoder input. The best decoder output is simply determined from a weighted sum of conditional expectations over all possible encoder regions for the given decoder input. This result is simpler than the equivalent result for channels where errors are not detected (index error channels as opposed to index erasure channels). In that case, the value of $\widehat{\Lambda}$ does not completely determine the channel state, so the channel probabilities do not cancel as in (5.13). Index error channels have been considered in work on channel-optimized VQ (COVQ).

For the special case of the common squared error distortion, $d(X,Y) = \|X-Y\|^2$, the expression for the optimal decoder value simplifies to

$$y_{\widehat{\lambda}} = \sum_{\lambda\in\mathcal{J}_{\widehat{\lambda}}} E[X|\Lambda=\lambda]P(\Lambda=\lambda|\widehat{\Lambda}=\widehat{\lambda}). \tag{5.15}$$

With the encoder $Q(x)$ fixed, the encoding region associated with encoder output $\Lambda = \lambda$ is defined as

$$R_\lambda \equiv \{x \in \mathbb{R}^k : Q(x) = \lambda\} \tag{5.16}$$

so

$$\Lambda = \lambda \iff X \in R_\lambda. \tag{5.17}$$

Therefore, the decoding rule can also be written as

$$y_{\widehat{\lambda}} = \sum_{\lambda \in \mathcal{J}_{\widehat{\lambda}}} E[X|X \in R_\lambda] P(X \in R_\lambda | X \in \bigcup_{\lambda' \in \mathcal{J}_{\widehat{\lambda}}} R_{\lambda'}) \tag{5.18}$$

which is a weighted sum of centroids of the encoder regions corresponding to the given decoder input.

Proof:

Let Y^* be the optimal decoder output as in (5.15) and let $\widetilde{Y}$ be the decoder output of any other rule. Then

$$\begin{aligned} E[d(X,\widetilde{Y})] &= E[\|X - \widetilde{Y}\|^2] && (5.19)\\ &= E[\|(X - Y^*) - (\widetilde{Y} - Y^*)\|^2] && (5.20)\\ &= E[\|(X - Y^*)\|^2] - 2E[(X - Y^*)^T(\widetilde{Y} - Y^*)] + E[\|(\widetilde{Y} - Y^*)\|^2] && (5.21)\\ &\geq E[\|(X - Y^*)\|^2] - 2E[(X - Y^*)^T(\widetilde{Y} - Y^*)] && (5.22)\\ &\geq E[d(X, Y^*)] - 2E[(X - Y^*)^T(\widetilde{Y} - Y^*)]. && (5.23) \end{aligned}$$

Now considering only the second term,

$$\begin{aligned} E[(X - Y^*)^T(\widetilde{Y} - Y^*)] &= E[E[(X - Y^*)^T(\widetilde{Y} - Y^*)|\widehat{\Lambda} = \widehat{\lambda}]] && (5.24)\\ &= E[E[(X - Y^*)|\widehat{\Lambda} = \widehat{\lambda}]^T(\widetilde{Y} - Y^*)] && (5.25) \end{aligned}$$

since $(\widetilde{Y} - Y^*)$ is a deterministic function of $\widehat{\Lambda}$. Then $E[Y^*|\widehat{\Lambda} = \widehat{\lambda}] = y_{\widehat{\lambda}}$ as in (5.15) and

$$\begin{aligned} E[X|\widehat{\Lambda} = \widehat{\lambda}] &= E[X| \cup_{\lambda \in \mathcal{J}_{\widehat{\lambda}}} \{\Lambda = \lambda, \widehat{\Lambda} = \widehat{\lambda}\}] && (5.26)\\ &= \sum_{\lambda \in \mathcal{J}_{\widehat{\lambda}}} E[X|\Lambda = \lambda, \widehat{\Lambda} = \widehat{\lambda}] P(\Lambda = \lambda | \widehat{\Lambda} = \widehat{\lambda}) && (5.27)\\ &= \sum_{\lambda \in \mathcal{J}_{\widehat{\lambda}}} E[X|\Lambda = \lambda] P(\Lambda = \lambda | \widehat{\Lambda} = \widehat{\lambda}) && (5.28)\\ &= y_{\widehat{\lambda}}. && (5.29) \end{aligned}$$

Substituting back into (5.25) gives

$$E[(X - Y^*)^T(\widetilde{Y} - Y^*)] = E[(y_{\widehat{\lambda}} - y_{\widehat{\lambda}})^T(\widetilde{Y} - Y^*)] = 0 \tag{5.30}$$

which means the last term in (5.23) is 0 and

$$E[d(X, \widetilde{Y})] \geq E[d(X, Y^*)]. \tag{5.31}$$

□

5.3.2.1 Training-set implementation

Vector quantizers are usually designed using a set of data representative of the source known as a *training set*. The encoder regions and codebook values are typically designed from this training set using an iterative algorithm like the generalized Lloyd algorithm (GLA). Since practical implementations are almost always based on training-set data it is important to consider how the optimal decoder output value is determined in this case.

The training set consists of representative samples from the source and is denoted $\mathcal{T} = \{x_1, x_2, \ldots, x_{|\mathcal{T}|}\}$ where $|\mathcal{T}|$ is the size of the training set. Finally, for convenience in writing the optimal decoder output in terms of the training-set data, define an indicator function of an event S as

$$I_S(x) \equiv \begin{cases} 1 & \text{if} \quad x \in S \\ 0 & \text{otherwise} \end{cases}. \tag{5.32}$$

Now the conditional probability in (5.14) can be written as

$$\begin{aligned} P(\Lambda = \lambda | \widehat{\Lambda} = \widehat{\lambda}) &= \frac{\sum_{x \in \mathcal{T}} I_{R_\lambda}(x)}{\sum_{\lambda' \in \mathcal{J}_{\widehat{\lambda}}} \sum_{x \in \mathcal{T}} I_{R_{\lambda'}}(x)} && (5.33) \\ &= \frac{|\mathcal{T} \cap R_\lambda|}{\sum_{\lambda' \in \mathcal{J}_{\widehat{\lambda}}} |\mathcal{T} \cap R_{\lambda'}|}. && (5.34) \end{aligned}$$

Therefore the conditional probability can be computed by counting the number of training-set elements in the encoding region R_λ divided by the number of training set elements in all the encoding set regions associated with the decoder input $\widehat{\lambda}$. Furthermore, the conditional expectation of the distortion in (5.10) can be written as

$$
\begin{aligned}
E[d(X,y|\Lambda=\lambda] &= \sum_{x\in\mathcal{T}} I_{R_\lambda}(x)d(x,y) &\quad (5.35)\\
&= \sum_{x\in\{\mathcal{T}\cap R_\lambda\}} d(x,y), &\quad (5.36)
\end{aligned}
$$

and combining the expressions in (5.34) and (5.36), the optimal decoder rule can written as

$$y_{\widehat{\lambda}} = \arg\min_y \sum_{\lambda\in\mathcal{J}_{\widehat{\lambda}}} \frac{|\mathcal{T}\cap R_\lambda|}{\sum_{\lambda'\in\mathcal{J}_{\widehat{\lambda}}}|\mathcal{T}\cap R_{\lambda'}|} \sum_{x\in\{\mathcal{T}\cap R_\lambda\}} d(x,y). \quad (5.37)$$

For the special case of squared error distortion, $d(X,Y) = \|X-Y\|^2$, the expression in (5.15), adapted for the training-set implementation, is

$$
\begin{aligned}
y_{\widehat{\lambda}} &= \sum_{\lambda\in\mathcal{J}_{\widehat{\lambda}}} \frac{|\mathcal{T}\cap R_\lambda|}{\sum_{\lambda'\in\mathcal{J}_{\widehat{\lambda}}}|\mathcal{T}\cap R_{\lambda'}|} \sum_{x\in\{\mathcal{T}\cap R_\lambda\}} \frac{x}{|\mathcal{T}\cap R_\lambda|} &\quad (5.38)\\
&= \frac{1}{\sum_{\lambda'\in\mathcal{J}_{\widehat{\lambda}}}|\mathcal{T}\cap R_{\lambda'}|} \sum_{\lambda\in\mathcal{J}_{\widehat{\lambda}}} \sum_{x\in\{\mathcal{T}\cap R_\lambda\}} x &\quad (5.39)
\end{aligned}
$$

which is simply the average value of the training set vectors in the encoding regions which could be mapped into the decoder input $\widehat{\lambda}$.

5.3.2.2 Multistage specialization

So far in Section 5.3 nothing about the multistage structure of the encoder has been used in determining the optimal decoder. The only assumptions used so far are that the encoder outputs a vector of indices and these indices are subject to erasures by the channel. The practicality of the most general encoder will depend on the number of indices and the size of each of the index sets.

In Figure 5.9 the details of the encoder have been included for the multistage structure with a residual function $f_i(x,y) = y - \alpha_i x$ with $\alpha_i \in [0,1]$. This residual function represents just one possible method of propagating source information to later stages. Values of α_i near 0 result in high redundancy, and values near 1 result in little information propagation ($\alpha_i = 1$ is the usual residual computation). Also the components of output vector Λ are determined by the individual stage encoders $Q_i : \mathbb{R}^k \to \mathcal{I}_i$ $i = 1, \ldots, M$.

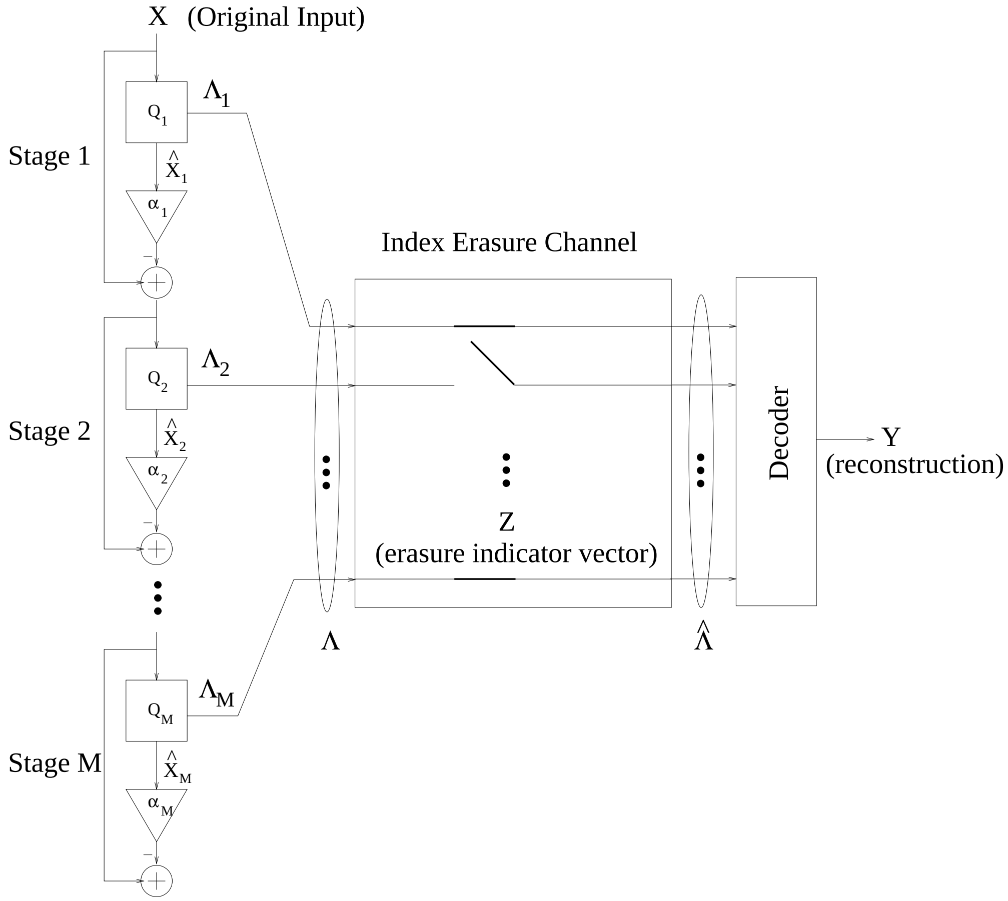


Figure 5.9 Multistage encoder implementation.

The previous results in Section 5.3.2 for the optimal decoder rule can be written in terms of the parameters of the stage encoders. Let $R^i_{\lambda_i} \equiv \{x \in \mathbb{R}^k : Q_i(x) = \lambda_i\}$ be the encoding region of stage i associated with stage output λ_i and let $\widehat{X}^i_{\lambda_i}$ be the estimate used at stage i for the residual computation. Then the encoding region R_λ associated with the vector of indices λ can be written in terms of stage encoders as

$$R_\lambda = \bigcap_{i=1}^{M} (\sum_{j=1}^{i-1} \alpha_j \widehat{X}^j_{\lambda_j} + R^i_{\lambda_i}). \tag{5.40}$$

Therefore the overall encoding region is the intersection of shifted versions of the individual stage encoder regions.

5.3.2.3 Extensions to long stages

The model represents an abstraction of the actual system implementation details. Modern image coders often consist of simple scalar quantizers acting on individual transform coefficients. The output indices of these individual quantizers are usually coded in groups using adaptive entropy coding, so the relationship between the bit stream actually transmitted over the channel and the individual quantizer indices can be quite complex. However, because the previous analysis is based on an index erasure channel, the details of the actual index transmission are unimportant as long as errors can be detected. Therefore the model can be applied to individual quantizers or small groups of quantizers, which allows very long stages to be considered without quickly running into complexity problems.

It is important to make a clear distinction between the concepts of packets and stages. Packets, or transmission units, simply represent a method of segmenting the bit stream into convenient sizes for transmission. Thus they are simply a transport layer mechanism and do not change the model. Stages, on the other hand, represent independent sections of the encoding process which can be decoded without information from other stages.

Because a stage typically consists of several packets and decoding within a stage proceeds until the first packet loss (at which point synchronization is lost and decoding of that stage must terminate), each quantized transform coefficient is usually partially recovered (i.e., truncated). Therefore if the groups of coefficients are coded in a progressive fashion within a stage, the most important information is more likely to occur before the first error. Figure 5.10 shows a conceptual block diagram of this process of coding, transmission, and recovery of an image stage. Since the result of the channel losses on each coefficient is a truncation of the quantization index, the benefit of using multiple stages can be analyzed by considering the performance of a scalar quantizer with truncation.

First consider a uniform source on $[-\frac{1}{2}, \frac{1}{2}]$ quantized by a uniform quantizer with the quantization index transmitted progressively (i.e., MSB $\rightarrow$ LSB) and a squared error distortion. Assuming that decoding must stop at the first error which occurs on bit $n+1$, the expected

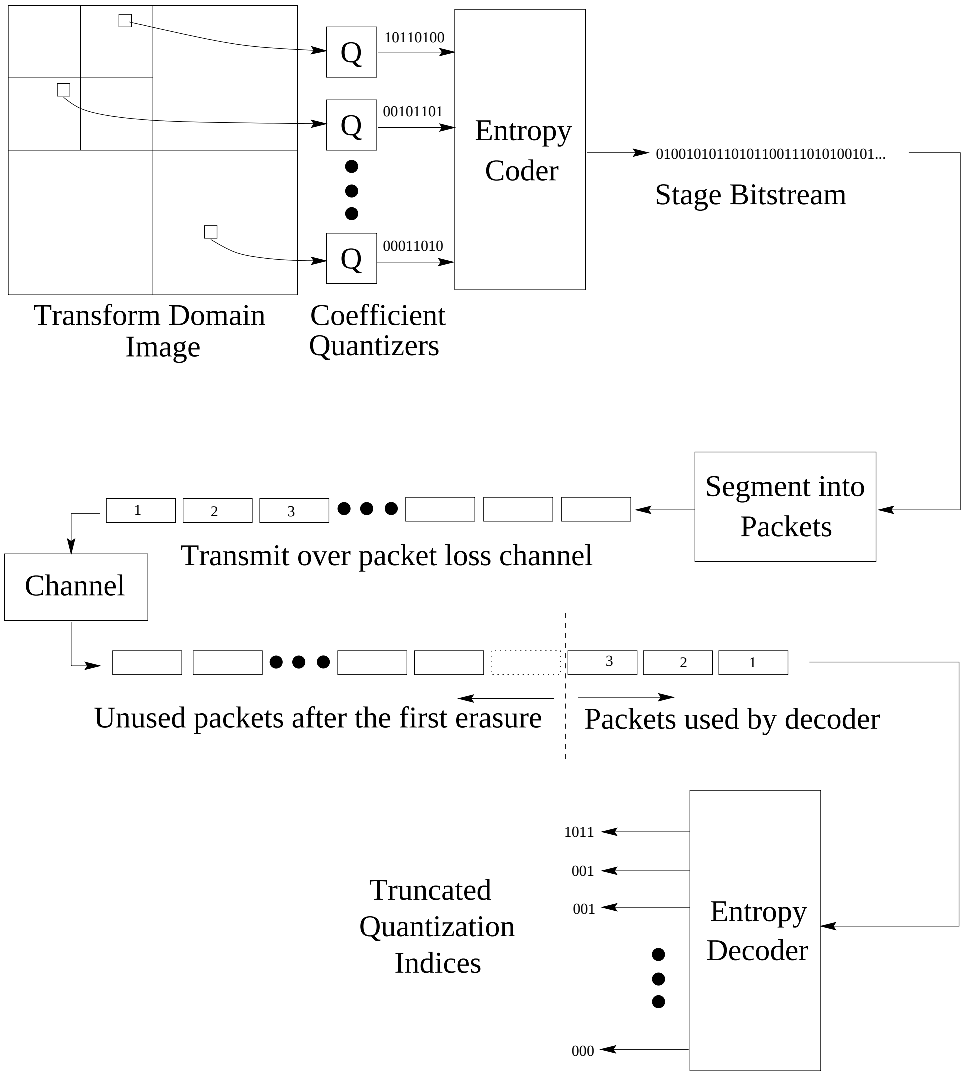


Figure 5.10 Single-stage encoding for an image.

distortion of the recovered value using n bits is [75]

$$D_n = \frac{1}{12} 2^{-2n}. \tag{5.41}$$

In the high-rate case, early truncation leads to significant distortion compared to the recovered value using the full index. This is not surprising since only a small portion of the bits can be used in the reconstruction. Coding multiple stages, or descriptions, allows more bits to be used since each stage can be decoded independently. In addition, allowing some of the information to be coded in multiple stages reduces the worst-case distortion at the expense of performance under noise-free conditions. Consider a situation where two stages are coded to equal rate with a structure as depicted in Figure 5.11. The gain term α in the residual computation is used to allow information to “leak” into the second stage so that the distortion is lower when an error occurs early in stage 1.

The input to stage 2 is given by

$$\begin{aligned} X_2 &= X - \alpha\widehat{X}_1 = (X - \widehat{X}_1) + (1-\alpha)\widehat{X}_1 && (5.42) \\ &= \delta + \gamma && (5.43) \end{aligned}$$

where δ is the first stage encoding error and γ is a scaled version of the stage 1 estimate. In order to obtain some tractable analytical results we assume that the quantities δ and γ are produced by high-resolution quantization and that they are independent. In practice these are often reasonable assumptions. Then γ will be a nearly continuous scaled version of the original source. Using these approximations, the pdf of X_2 is simply the convolution of δ and γ as shown in Figure 5.12 and has a pdf given by

$$f_{X_2}(y) = \begin{cases} \frac{1}{\mathrm{Min}\cdot\mathrm{Max}}\left(y + \frac{1}{2}(\mathrm{Max}+\mathrm{Min})\right) & -\frac{\mathrm{Max}+\mathrm{Min}}{2} \le y \le -\frac{\mathrm{Max}-\mathrm{Min}}{2} \\ \frac{1}{\mathrm{Max}} & -\frac{\mathrm{Max}-\mathrm{Min}}{2} \le y \le \frac{\mathrm{Max}-\mathrm{Min}}{2} \\ -\frac{1}{\mathrm{Max}\cdot\mathrm{Min}}\left(y - \frac{1}{2}(\mathrm{Max}+\mathrm{Min})\right) & \frac{\mathrm{Max}-\mathrm{Min}}{2} \le y \le \frac{\mathrm{Max}+\mathrm{Min}}{2} \\ 0 & \mathrm{otherwise} \end{cases} \quad (5.44)$$

where $\mathrm{Max} \equiv \mathrm{Max}(\Delta, (1-\alpha))$ and $\mathrm{Min} \equiv \mathrm{Min}(\Delta, (1-\alpha))$.

Using the resulting pdf for X_2 along with Bennett’s integral for the distortion of the optimal high-resolution quantizer [75, pp.186], the distortion at the output of stage 2 is given by

$$D_2(R) = \frac{\sigma_{X_2}^2}{12}\left(\int \tilde{f}_{X_2}(y)^{\frac{1}{3}} dy\right)^3 2^{-2R} \quad (5.45)$$

where $\tilde{f}_{X_2}(y) = \sigma_{X_2} f_{X_2}(\sigma_{X_2} y)$ is the normalized pdf which has unit variance.

The variance of X_2 is

$$\sigma_{X_2}^2 = \frac{\Delta^2 + (1-\alpha)^2}{12} \quad (5.46)$$

and the distortion is given by

$$D_2(R) = \frac{\sigma_{X_2}^2}{12}\left(\int_{-a}^{a} (\frac{\sigma_{X_2}}{\mathrm{Max}})^{1/3} dx - 2\int_{a}^{b} (\frac{\sigma_{X_2}^2}{\mathrm{Max}\cdot\mathrm{Min}}x - \frac{\sigma_{X_2}(\mathrm{Max}+\mathrm{Min})}{2\cdot\mathrm{Max}\cdot\mathrm{Min}})^{1/3} dx\right)^3 2^{-2R}$$

(5.47)

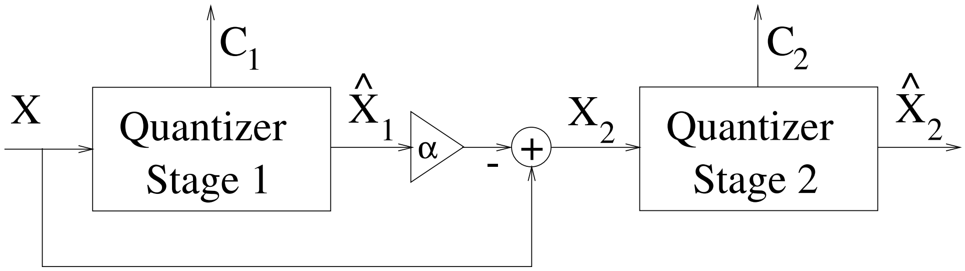


Figure 5.11 Two-stage encoding block.

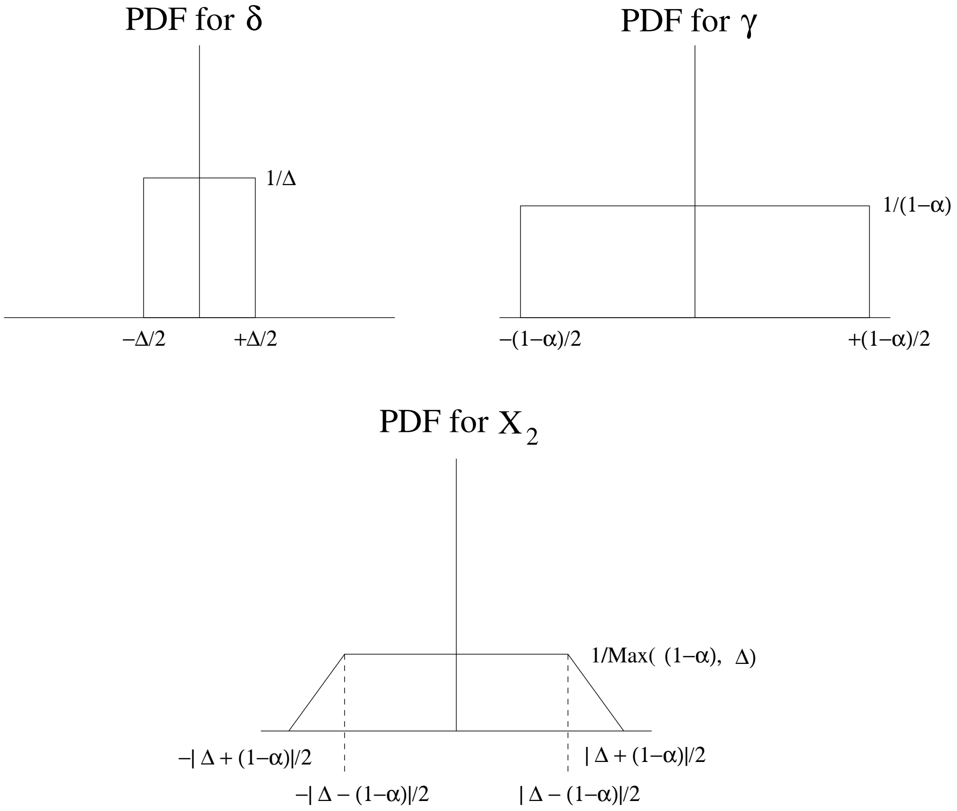


Figure 5.12 PDF of the input to stage 2.

where $a = (\text{Max} - \text{Min})/(2\sigma_{X_2})$ and $b = (\text{Max} + \text{Min})/(2\sigma_{X_2})$. After some algebra (5.47) simplifies to

$$D_2(R) = \frac{1}{12}\left(\text{Max}^{2/3} - \text{Max}^{-1/3}\text{Min} - 3\text{Max}^{-1/3}\text{Min}^{1/3}\right)^3 2^{-2R}. \tag{5.48}$$

Under the high rate assumption for the two stages as well as some redundancy for robustness, the relative magnitudes are $(1-\alpha) \gg \Delta$ and the distortion after combining stages is approximately

$$D_2(R) = \frac{1}{12}\left((1-\alpha)^{2/3} - (1-\alpha)^{-1/3}\Delta - 3(1-\alpha)^{-1/3}\Delta^{1/3}\right)^3 2^{-2R} \approx \frac{(1-\alpha)^2}{12}2^{-2R}. \tag{5.49}$$

The performance with both stages available must be traded off against the performance when only portions of each stage are available at the decoder. If $\widehat{X}_1$ and $\widehat{X}_2$ represent the stage reconstructions with all the information available and $\tilde{X}_1$ and $\tilde{X}_2$ represent the stage reconstructions with partial information, then the error can be written as

$$e = (X - \alpha\tilde{X}_1 - \tilde{X}_2) = (X - \alpha\widehat{X}_1 - \widehat{X}_2) + \alpha(\widehat{X}_1 - \tilde{X}_1) + (\widehat{X}_2 - \tilde{X}_2). \tag{5.50}$$

Since each of the error terms is approximately independent, the total distortion is given by

$$D(R) = D_2(R) + \alpha^2 E[(\widehat{X}_1 - \tilde{X}_1)^2] + E[(\widehat{X}_2 - \tilde{X}_2)^2]. \tag{5.51}$$

For the case where some of the bits are lost from stage 1 introducing a distortion $\alpha^2\epsilon_1^2 = \alpha^2 E[(\widehat{X}_1 - \tilde{X}_1)^2]$, the resulting reconstruction distortion is

$$D(R) = D_2(R) + \alpha^2\epsilon_1^2. \tag{5.52}$$

Since there is fundamental trade-off between performance when all information arrives and performance when only a portion of the information is available at the decoder, one common method is to optimize system parameters so that performance is optimized under no losses subject to a minimum performance when losses occur. Let D_w be the required worst-case distortion when some information is lost so that

$$D_2(R) + \alpha^2\epsilon_1^2 < D_w \tag{5.53}$$

is the distortion constraint. Using the approximation in (5.49), the problem can be written as an optimization problem for the parameter α as follows:

$$\min_{\alpha} \frac{(1-\alpha)^2}{12} 2^{-2R} \quad \text{subject to} \quad \frac{(1-\alpha)^2}{12} 2^{-2R} + \alpha^2 \epsilon_1^2 < D_w. \tag{5.54}$$

Since the error-free distortion decreases monotonically with α, the solution will be the largest value of α which meets the constraint. The constraint equation can be solved for equality to yield the following pair of values:

$$\alpha = \frac{1 \pm \sqrt{1 - (2^{-2R} + 12\epsilon_1^2)(1 - 12D_w)}}{2^{-2R} + 12\epsilon_1^2} \tag{5.55}$$

from which the largest value in the range $[0, 1]$ is selected. If there are no values in this range, then the constraint cannot be satisfied.

Although this approach is attractive in terms of complexity since each stage encodes and decodes in a fairly independent fashion, the difficulty is that the performance is poor between the two extremes of zero excess rate and complete redundancy. As seen from the derivation of the optimal decoder, the optimal decoder combines the stage information by using the intersection of encoding regions for each stage. If these regions are nearly identical (implying high excess rate), then the estimate using the information from both stages will not be significantly better than the single-stage estimate. Ideally the distortion of the combined estimate should decrease at an exponential rate slightly less than the sum of the encoding rates of the individual stages. Instead the combined distortion is reduced by a constant factor compared with the single stage 2 estimate. The stage quantizers need to be coordinated so that when all the information is available, each stage provides new information.

This problem motivates the consideration of multiple description (MD) quantizers which provide better performance when combining information from the descriptions. In order to achieve small quantizer cells (i.e., much smaller than with a single description) when all descriptions are combined, the MD quantizer region associated with a single description index consists of a union of disjoint cells. When regular quantizers are cascaded in the multistage structure, the encoding regions associated with each index consist of contiguous intervals which leads to large cell intersections in certain cases when the information is combined. It may be desirable to code more than two descriptions depending on the details of the channel, so extensions to the two-channel MD scalar quantizer are desirable.

In Section 5.4 a multidimensional extension of the two-description MD scalar quantizer is presented as another method of adding a tunable amount of source redundancy. This method provides a good way to improve error resiliency, especially when the number of packets (descriptions) to be transmitted is low.

5.3.3 Optimal encoder design

The optimal encoder given a fixed decoder is derived in this section. The encoder is a mapping $Q : \mathbb{R}^k \rightarrow \mathcal{J}$ and the goal is to determine a mapping to minimize the expected distortion between input and decoder output given a fixed decoder which can be represented by the output value $Y_{\widehat{\lambda}}$ for each possible decoder input $\widehat{\lambda} \in \widehat{\mathcal{J}}$. A lower bound on the expected distortion can be derived as follows:

$$\begin{aligned} E[d(X,Y)] &= \sum_{z \in \{0,1\}^M} P(Z=z) E[d(X,Y)|Z=z] && (5.56) \\ &= E[\sum_{z \in \{0,1\}^M} P(Z=z) d(X, Y_{h(Q(X),z)})] && (5.57) \\ &\geq E[\min_{\lambda' \in \mathcal{J}} \sum_{z \in \{0,1\}^M} P(Z=z) d(X, Y_{h(\lambda',z)})]. && (5.58) \end{aligned}$$

Furthermore this lower bound can be achieved by defining the regions within $\mathbb{R}^k$ that are mapped to encoder output λ as follows:

$$\begin{aligned} R_\lambda \equiv \{x \in \mathbb{R}^k : & \sum_{z \in \{0,1\}^M} P(Z=z) d(X, Y_{h(\lambda,z)}) \\ & \leq \sum_{z \in \{0,1\}^M} P(Z=z) d(X, Y_{h(\lambda',z)})\} \qquad \forall \lambda' \neq \lambda. \end{aligned} \quad (5.59)$$

For the common squared error distortion measure $d(X,Y) = \|X - Y\|^2$, the encoding regions are convex polytopes.

Proof:

For $X \in R_\lambda$ as defined in (5.59) we have

$$X \in R_\lambda \iff \sum_{z\in\{0,1\}^M} \left(\|X - Y_{h(\lambda,z)}\|^2 - \|X - Y_{h(\lambda',z)}\|^2\right) \cdot P(Z = z) \le 0 \qquad \forall \lambda' \neq \lambda \tag{5.60}$$

$$\iff \sum_{z\in\{0,1\}^M} \left(\|Y_{h(\lambda,z)}\|^2 - \|Y_{h(\lambda',z)}\|^2 - 2X^t(Y_{h(\lambda,z)} - Y_{h(\lambda',z)})\right) \cdot P(Z = z) \le 0 \qquad \forall \lambda' \neq \lambda \tag{5.61}$$

$$\iff X^t A_{\lambda,\lambda'} + B_{\lambda,\lambda'} \le 0 \qquad \forall \lambda' \neq \lambda \tag{5.62}$$

where $A_{\lambda,\lambda'}$ and $B_{\lambda,\lambda'}$ are constants equal to

$$A_{\lambda,\lambda'} = -2 \cdot \sum_{z\in\{0,1\}^M} (Y_{h(\lambda,z)} - Y_{h(\lambda',z)})P(Z = z) \tag{5.63}$$

$$B_{\lambda,\lambda'} = \sum_{z\in\{0,1\}^M} (\|Y_{h(\lambda,z)}\|^2 - \|Y_{h(\lambda',z)}\|^2)P(Z = z). \tag{5.64}$$

Therefore,

$$R_\lambda = \bigcap_{\lambda'} \{X \in \mathbb{R}^k : X^t A_{\lambda,\lambda'} + B_{\lambda,\lambda'} \le 0\} \tag{5.65}$$

$$= \bigcap_{\lambda'} H_{\lambda,\lambda'} \tag{5.66}$$

where $H_{\lambda,\lambda'}$ is a half-plane region ⇒ R_λ is the intersection of half-planes ⇒ R_λ is a convex polytope.

□

5.4 Multiple Description Coding for Packet Erasure Channels

The use of multiple description techniques to improve error resilience on packet erasure channels was motivated in Section 5.3.2.3 to allow a better trade-off between performance under packet loss and lossless performance. Multiple descriptions provide a way to achieve acceptable performance if a minimal subset of descriptions is available and to achieve increasingly better performance if more descriptions are available. Early work on multiple descriptions considered the case of two descriptions. Theoretical results for the achievable rate region include results in [44, 79]. Currently the achievable rate-distortion is completely known only for the two-channel memoryless Gaussian source. Recent work in the area of multiple descriptions includes [80, 81, 82, 83, 84, 85, 86, 87].

Previous work on MD coding includes methods which allow a wide range of redundancy levels to be introduced across a small set of descriptions (usually two descriptions). The MD scalar quantizer in [45] uses an index assignment to map a regular quantizer index to a pair of indices with a selectable amount of redundant coding between them. Application of MD scalar quantizers to image coding is considered in [85] including some discussion of extensions to more than two descriptions. The first part of [85] considers the application of the usual two-description MD scalar quantizers to generate two descriptions. The method is extended to many descriptions by applying FEC coding to the output of MD scalar quantizer to generate more than two descriptions. In particular an $(n, 1, n)$ repetition code is presented as a specific example. Correlating transforms are used to introduce redundancy between pairs of values in [82, 88]. This method is based on introducing a statistical redundancy between the pair of values. In [89] quantized overcomplete frame expansions are used to produce a small set of descriptions with some redundancy. In this case, the redundancy is deterministic and is caused by using a larger expansion set than is necessary to form a basis.

Data partitioning is another common technique used to add error resiliency to a source coder, and examples include the work in [38, 71, 90, 91]. High-performance compression algorithms, such as the SPIHT image coder [10], often require the source coder bits to be decoded in a sequential and uninterrupted fashion for correct interpretation. Carefully partitioning the data into separate streams prior to coding can improve the error resilience. The idea is that the source decoder can use more of the correctly received information because independent pieces of the transmission can be correctly interpreted despite losing other portions. Data partitioning often also facilitates the application of unequal error protection channel coding. This method is an attractive technique for situations where the decoder complexity is a primary concern and graceful degradation under varying channel conditions is important. The cost is a loss of compression performance because the dependencies among the partition elements cannot be fully exploited.

For situations where the number of packets is large, applying FEC channel codes to the output of the source coder works well. When the block lengths of these codes are long, a large number of rates are possible and thus the code rate can be matched to the channel very accurately. Also the FEC can be applied in an unequal fashion based on the importance of the information to provide improved efficiency and some graceful degradation. Examples of recent work using this method are [62, 92]. The major costs of this method include decoding complexity, decoding delay (i.e., loss of progressivity), and sensitivity to channel mismatch.

The best choice from the previous methods depends on the number of packets to be transmitted. Large packet sizes are desirable to reduce the overhead from the packet header information, but networks typically impose constraints on the maximum packet size (i.e., the path maximum transmission unit (MTU)). The packet size may also be limited to maintain responsiveness of the system. For image transmission over ATM networks, where packet payloads are 384 bits, the total number of packets to transmit a typical image may be on the order of hundreds. Alternatively, default maximum packet sizes for Internet routers and good choices for wireless connections like cellular digital protocol data (CDPD) are around 4600 bits which may imply from tens up to perhaps 100 packets at very high rates. Finally, Ethernet systems support even larger packet sizes (e.g., 12000 bits) which may imply a total of only tens of packets being transmitted for an image. These constraints should be considered when designing the image source and channel coding.

5.4.1 FEC for erasure channels

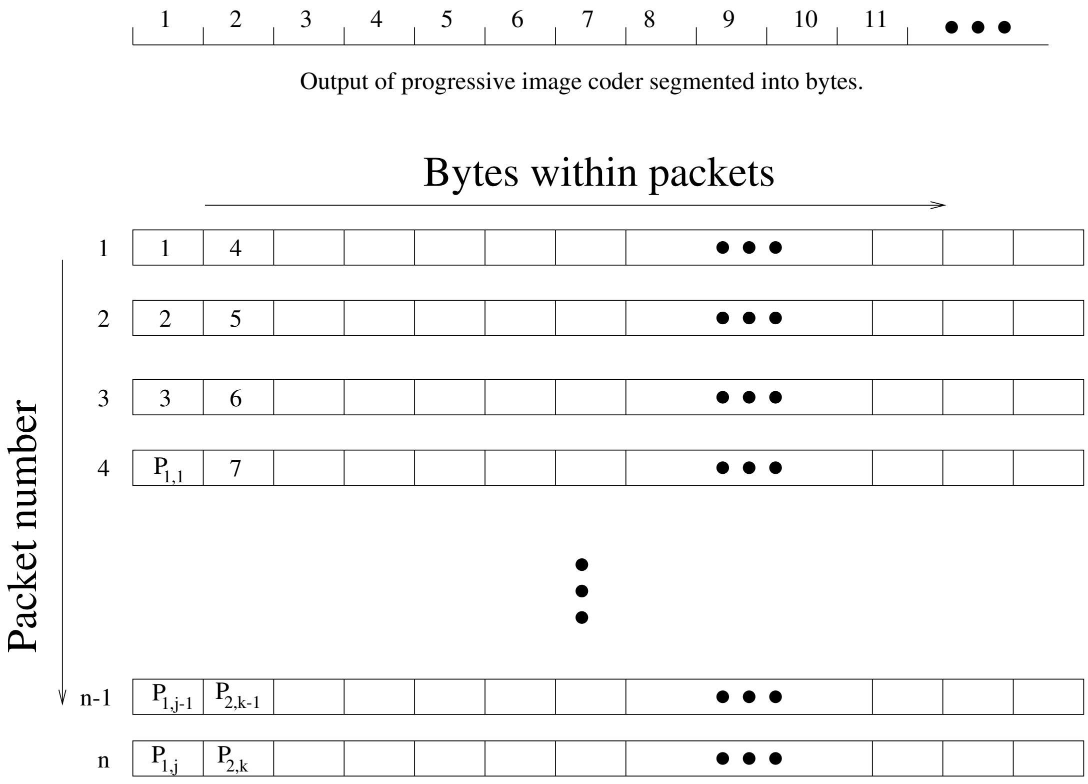
One of the first options to consider when designing an image transmission system for packet erasure channels is FEC methods. As the results from the previous chapters have shown, very efficient systems can be designed by combining excellent compression algorithms with strong error correcting codes. Delay and computational complexity are the two main costs of this approach.

When delay is not an issue, the output of the image coder can be segmented into pieces, which will be treated as symbols in the FEC code and interleaved over the length of the transmission. For clarity in the remaining discussion these pieces are assumed to be bytes, but other sizes are certainly possible. The information is grouped, according to importance, into codewords and is coded with the level of protection appropriate to the importance of the information, the channel erasure rate, and the constraint on the transmission rate. For a progressive coder like SPIHT, the bytes are approximately ordered from most to least important in terms of MSE distortion, so sorting the bytes according to importance is straightforward. The idea is depicted in Figure 5.13. Because the bit stream from the image coder has the serial decoding requirement, decoding terminates at the first nonrecovered byte. This requirement combined with the decreasing impact on MSE leads to monotonically nonincreasing levels of protection for the output bytes.

The coding technique is identical to the outer code of the product code presented in Section 4.2 for fading channels. Corresponding bytes from the packets are the symbols in the particular Reed-Solomon codeword. The difference here is that each codeword spans the entire length of the transmission, so that a particular set of information protected with an $(n.k)$ code can be recovered as long as at least *any* k of the n packets are received. When the number of packets to be transmitted is large (i.e., the packet size is much smaller than the total transmission size) and the channel erasure is a known constant, this method works well since the codes have very low error probabilities at rates near the capacity of the erasure channel. Difficulties arise when the channel error rate is varying or unknown.

5.4.2 Proposed image coding structure

Most recent high-performance image coding algorithms work by coding the quantized coefficients of a linear transform (often a wavelet transform). The coded bits can usually be divided into two broad classes: the map bits, which describe the locations of the large coefficients, and the quantization bits, which describe the quantized values of the coefficients identified in the map. The quantized values can be transmitted using techniques like multiple descriptions which allow graceful degradation even if there is partial loss of the information.



Numbers represent corresponding bytes from coder output.

$P_{i,j}$ = parity symbol (j) from codeword i.

Figure 5.13 FEC coding for an erasure channel.

A multidimensional extension of the usual two-description MD scalar quantizer is described below.

The map information, however, is not usually amenable to partial recovery like the quantized values. The decoder needs to recover the same sequence of map bits in order to stay synchronized with the encoder. For this reason, FEC protection seems to be the best choice for the map sequence. Simple repetition codes (i.e., just repeat the map sequence in each packet) would provide the maximum protection, since the entire map sequence can be recovered from a single packet, but result in an excessive redundancy when the number of packets to be transmitted is large. More generally, the map sequence can be split among the packets and coded with a (N, K) erasure correcting code which guarantees recovery of the map sequence given at least K packets are received. A block diagram of this coding structure is shown in Figure 5.14.

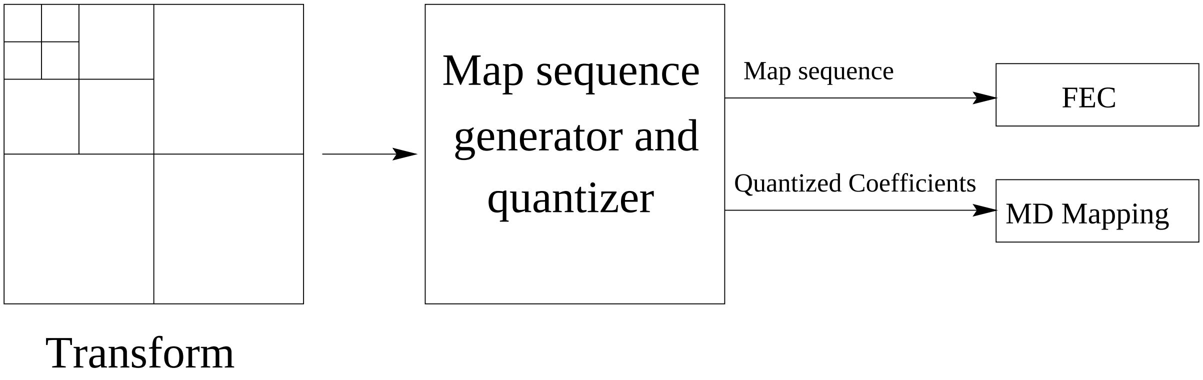


Figure 5.14 Block diagram of an image coder for packet loss channels.

5.4.3 Multidimensional multiple description scalar quantizer

The multiple description scalar quantizer described in [45] consists of a regular quantizer followed by an index assignment which maps to a pair of indices. The redundancy is tuned by selecting the number of central quantizer cells relative to the number of bits in the two description indices. An example of two index assignments is shown in Figure 5.15.

Each of the numbered cells in the mappings represents a cell of the central quantizer, and each quantized coefficient is mapped into a pair of coordinates which specify its position in the grid. As seen from these examples, the central quantizer indices are placed along diagonals of the grid, and the redundancy is controlled by selecting the number of diagonals. The same concept can be extended to more than two descriptions by using higher dimensional mappings. A straightforward extension is to create maps where cells from the central quantizer are placed along diagonals of an N-dimensional hypercube. Figure 5.16 illustrates the concept with a three-dimensional mapping which includes only the main diagonal resulting in a (3,1,3) repetition code of the central quantizer indices. Adding other diagonals around the main diagonal reduces the redundancy and allows the channel coding rate to be adjusted.

The redundancy of the mapping is directly related to the amount of empty space (i.e., the number hypercube cells *not* assigned a central quantizer index). With Q central quantizer

1	2						
3	4	5					
	6	7	9				
		8	10	11			
			12	13	15		
				14	16	17	
					18	19	21
						20	22

(a) Index assignment with high redundancy.

1	3	5	7				
2	8	10	12	14			
4	9	15	17	19	21		
6	11	16	22	23	25	27	
	13	18	24	29	30	32	34
		20	26	31	36	37	39
			28	33	38	41	43
				35	40	42	44

(b) Index assignment with less redundancy.

Figure 5.15 Example MD index assignment mappings for two descriptions.

cells, S indices for each description, and M descriptions, the channel code rate of the mapping is given by

$$r_c = \frac{\log_2(Q)}{M \log_2(S)}. \tag{5.67}$$

As can be seen from examining (5.67), difficulties arise in creating maps which provide high channel code rates with a small number of central quantizer cells and a large number of descriptions. These situations often occur in the context of image coding when coding the coefficients in the higher-frequency subbands. For these situations, data partitioning can be applied so that information about a specific high-frequency coefficient is only sent in a subset of the packets. A similar type of data partitioning was used in the context of image coding for packet erasures in [90, 91]. Also note that the description indices are typically fed into separate entropy coders (one for each description), so if the description is not a power of 2, it is not an inconvenience for coding.

Since these mappings are acting on the quantizer indices, they are similar to traditional channel codes (i.e., acting on bits). Any one-to-one mapping from the input alphabet of Q

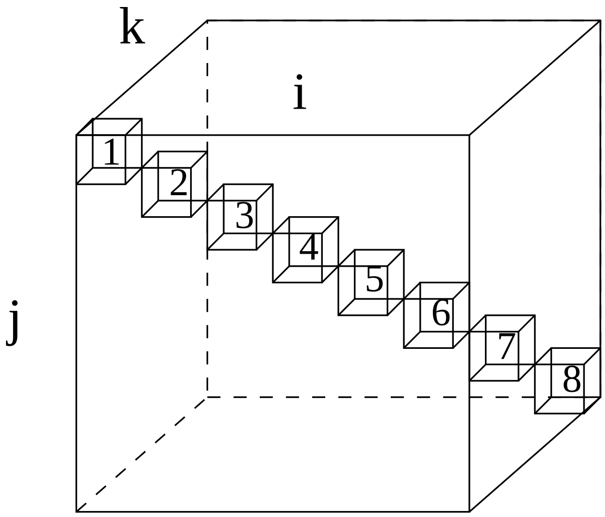


Figure 5.16 Example of a three-dimensional MD mapping.

symbols to the M output descriptions, each with an alphabet of size S, is possible. In particular, mappings can be generated which allow erasure correction by enforcing a minimum Hamming distance between the description indices. The main difference between these mappings and traditional channel codes is that the input symbol (bits) correspond directly to a quantized value instead of an arbitrary collection of bits, and therefore partial recovery to nearby cells is meaningful and useful.

Figure 5.17 shows an example three-dimensional mapping consisting of nine central quantizer cells (some cells are obscured in the figure) and description indices with an alphabet size of 3. The mapping allows any single erasure to be corrected (i.e., only two description indices are required to uniquely specify any of the central quantizer cells). The figure also helps illustrate the general trend that labeled cells tend to be distributed more diversely throughout the hypercube for mappings that provide more erasure correction capability. This wide distribution of labeled cells leads to the large distortions that result once the erasure correction capability is exceeded. In that case, the central quantizer indices cannot be assigned to cells to yield a small spread for all possible outcomes. Therefore guaranteed correction can be traded off against graceful degradation after the correction capability is exceeded.

As an example of the possible trade-offs, consider a case where the input consists of a 4-bit quantized value from a uniform quantizer acting on a uniform source on $[0, 1]$. The value is

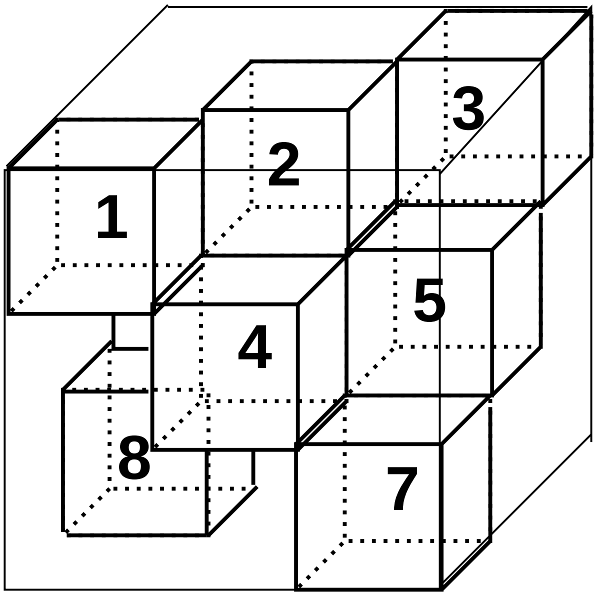


Figure 5.17 An example three-dimensional mapping which is capable of correcting one erasure.

mapped into four 2-bit descriptions. The performance of two possible mappings is shown in Figure 5.18. The first mapping is designed to correct two erasures, and the performance curve exhibits the sharp transition at three erasures. The second mapping is designed to correct one erasure, so the performance degrades for two erasures but it is better than the first system for three erasures.

Basic mappings are generated by assigning central quantizer indices to the diagonals of the hypercube. The input to the algorithm is the number of diagonals to include and the number of bits to use for each description index. The quantizer indices are assigned sequentially with one assignment made for each description axis for each iteration. The order of assignment within each iteration is based on selecting them in order of decreasing index spread. For those mappings designed for some erasure correction capability, the hypercube cells are selected according to the Hamming distance constraint, then the central quantizer labels are assigned so as to minimize the spread when the correction capability is exceeded.

As another example to illustrate the error resilience properties of the multidimensional MD quantizer, Figure 5.19 shows the MSE of a five-description quantizer with different levels of redundancy when between one and five packets are received. The source is a uniform source

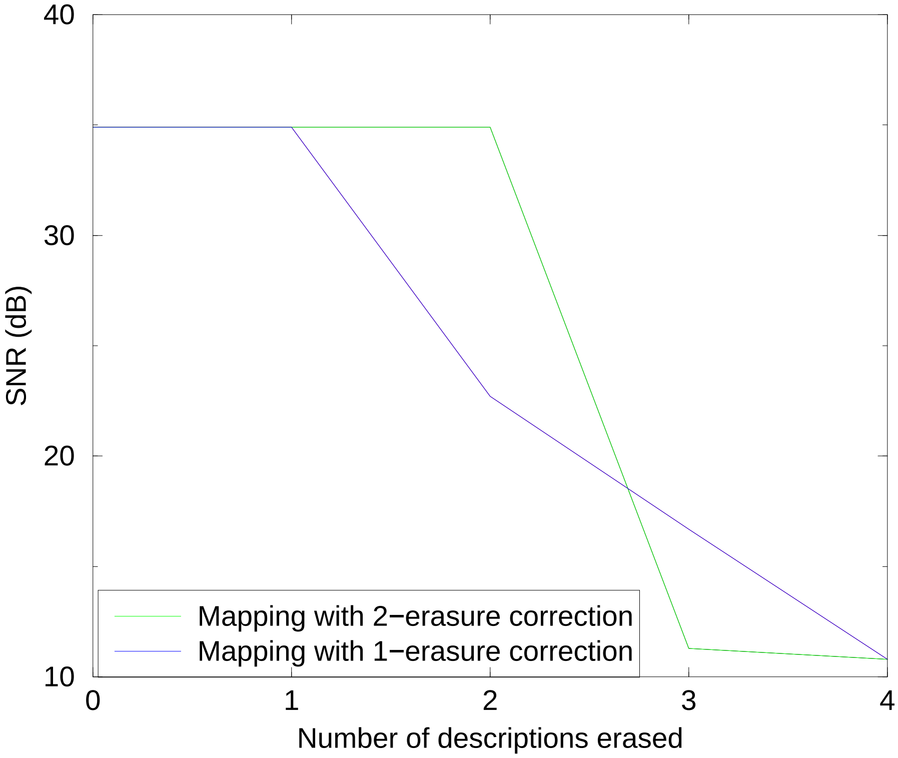


Figure 5.18 An example of performance trade-off for two mappings.

on $[0, 1]$ and the central quantizer is a uniform quantizer with 64 cells. The results for two mappings are shown, one with rate 0.4 and one with rate 0.6. In both cases the performance improves with each packet that is received, and the lower-rate mapping performs uniformly better at the cost of more bits to transmit.

5.4.4 Parameter optimization

Coefficients identified as significant in different passes in the coding of the map sequence have unequal importance in terms of their share of signal energy. The coefficients can be grouped according to the pass at which they became significant. Then the overall performance can be optimized by selecting different index assignment parameters (i.e., different amounts of source redundancy) over different the significance groups. In [85], it is shown that the optimization over subbands for the two-channel MD problem results in only marginal gains in performance. However, in this case the coefficients should be more accurately sorted by

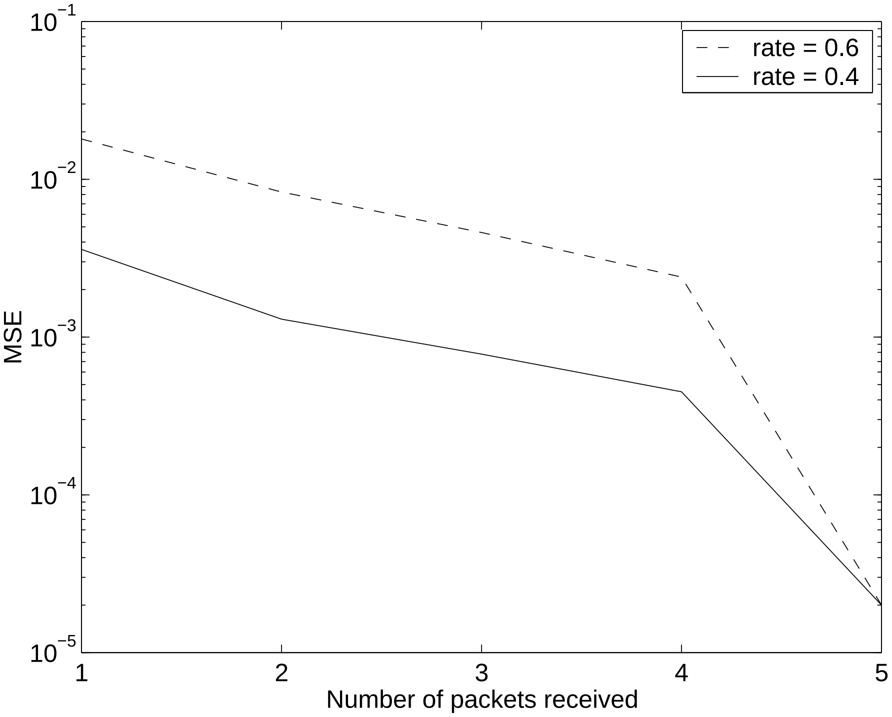


Figure 5.19 Performance for two five-description MDSQ mappings.

importance. Also, in the case of more than two descriptions, there are more possible channel outcomes, so the appropriate level of redundancy needs to be selected for the channel conditions.

We compute the proper selection via a dynamic programming (DP) algorithm where one dimension of the trellis is the significance group number and the other is the level of redundancy. In order to enforce the rate constraint during the DP procedure, some sort of pruning is used. Basically when selecting a survivor each candidate path is tested such that any path with cumulative bit-rate exceeding the rate constraint is pruned. Generally speaking this is a suboptimal approach, but for natural images using dyadic wavelet decomposition, in practice this suboptimal approach would lead to reasonable solutions.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

A number of coding methods for transmitting images over noisy channels have been presented in this thesis for memoryless, fading, and erasure channels. The results are summarized in this chapter, and possible future research directions are suggested.

In Chapter 3, a system combining a high-performance embedded wavelet coder and strong channel coding was proposed for transmission over a memoryless channel with a known error rate. This system showed excellent results compared to previously known robust image coders. Even when channel decoding failures terminate the bit stream early, the resulting image quality is often acceptable due to the embedded nature of the source coder. Only very marginal performance gains can be achieved using optimized channel code rate schedules, exploiting the unequal importance of the bit stream, compared with the best fixed-rate coding. The limited improvement can be attributed to the combination of the serial decoding requirement of the source coder along with the threshold effect of the channel codes.

Since the system combined separate source and channel coders, improvements in either would improve the overall system performance. In particular, the use of turbo codes for the channel coding would improve performance at the expense of complexity and possibly loss of some degree of progressivity due to the long blocks. The second part of Chapter 3 provided some results comparing performance of the system using RCPC/CRC, low-density parity check, and turbo codes, which did show improvements based on using the higher-performance channel codes. Even the product code, described in Chapter 4 and designed for fading channels, provided some improvement on memoryless channels. These results provided a recalibration of the benchmark for comparing the performance of new robust image coders. For memoryless channels with known error rates it appears difficult to beat a system based on Shannon's separation theorem, especially considering turbo coding methods are able to achieve performance very close to the Shannon limit by using long blocks. However, real-world systems and channels have other goals and constraints which provide many new challenges. The remainder of this thesis considered methods for dealing with fading channels, optimizations that included penalties for delay, and coding methods for erasure channels.

For progressive image coding, the goal is generally to improve the reconstructed image quality as much as possible for each transmitted bit. The progressive feature is useful for applications where many quality levels need to be reconstructed from a single bit stream or if fast recognition of the image content is important (e.g., browsing a database). The problem of optimizing channel code parameters for progressive image coding was examined in Chapter 3. Both the blocklength and rate of the channel code were optimized by dynamic programming using one of a general class of performance measures that was proposed to reflect the goals of progressive coding. Results from several images showed that the performance based on number of available source bits was nearly as good as that based on MSE or PSNR; the complexity is lower, and no overhead information is needed to describe the code parameters since the decoder can simply run the same algorithm to determine the schedule. The parameters tended to have nearly constant channel code rates and decreasing blocklengths over the transmission.

Extensions of this optimization method from still image to video is an interesting area of future research. The MPEG4 video encoding standard is based on coding separate video objects whose composition makes up the scene that is presented to the viewer. These video objects may include traditional video information, computer generated graphics, cartoon animations, and still image backgrounds. Each of these may require different levels of error protection based on the nature of the encoding, importance of content, temporal longevity, etc. Also, for two-way or interactive presentations, there are strict delay constraints. All of these factors lead to interesting questions in the context of error protection of visual information where delay is an issue.

Since modern systems often have to transmit over a variety of channel conditions, perhaps simultaneously as in a broadcast environment, it is desirable that the systems have some degree of robustness and degrade gracefully under more severe channel conditions. The work in Chapters 4 and 5 considered cases where the channel conditions vary and the goal was to improve the robustness of the system. In Chapter 4 a system combining good data partitioning with FEC was shown to have robustness under a variety of channel conditions. Proper data partitioning will continue to be an important tool for error protection by allowing UEP channel codes and the use of as much correctly received data as possible. New image coders such as the upcoming JPEG2000 standard will provide new challenges in this area.

The erasure channel considered in Chapter 5 can really be considered an abstraction from an arbitrary channel where reliable error detection provides the erasure indication. This model would likely apply to many real-world systems since the details of the physical channel are often hidden from the software (e.g., multimedia applications) operating at the application layer in the network protocol stack. This approach makes application development easier and allows a common software base to support platforms with different physical channels.

The proposed methods were centered around introducing robustness by combining source coder modifications and explicit FEC. An important aspect of the source coder modifications was the ability to introduce a tunable amount of source redundancy so the system could be matched to channel conditions. While simply using FEC for erasure correction works well when there are a large number of packets to be transmitted and the erasure rate is a known constant, the proposed multistage and multiple description methods are useful when complexity or delay is an issue or when the number of packets (descriptions) is small. For the multistage coder, improvements in residual coding are important. For the multidimensional multiple description coder, the challenge is improving the case where a large number of descriptions, moderate to low redundancy, and low quantizer resolution are simultaneously required for certain coefficients. Extensions from scalar to vector multiple descriptions as in [84] including complexity constraints would provide an additional challenge.

The protection of watermarking information is another interesting research direction. Watermarking is used to identify intellectual property ownership. Ideally the watermarking information should not noticeably distort the image and should be difficult to remove. The problem is how to preserve watermarking information under channel loss conditions while not degrading the image quality too much.

Many of the future research directions in this area will likely be driven by the problems and requirements of wireless multimedia due to the user demand for this functionality along with the highly challenging nature of the problems. Low-power wireless devices with constrained memory and processor resources that have the capability to support multimedia applications are just beginning to emerge. They must operate in severe multipath fading environments and will likely have significantly constrained transmission bandwidths (in terms of multimedia applications) for the quite some time. It is the combination of all these constraints which makes wireless multimedia extremely challenging.

REFERENCES

[1] H. Dudley, “The vocoder,” *Bell Labs Rec.*, vol. 18, Dec. 1939, pp. 122–126.

[2] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, 1948, pp. 379–423,623–656.

[3] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley-Interscience, 1991.

[4] R. G. Gallager, *Information Theory and Reliable Communication*. New York: John Wiley and Sons, Inc., 1968.

[5] C. E. Shannon, “Coding theorems for a discrete source with a fidelity criterion,” *IRE National Convention Record, Part 4*, 1959, pp. 142–163.

[6] CDMA Development Group Website, “`http://www.cdg.org/`.”

[7] J. Pennebaker and J. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.

[8] S. M. LoPresto, K. Ramchandran, and M. T. Orchard, “Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework,” *Proceedings DCC ’97. Data Compression Conference*, 1997, pp. 221–230.

[9] E. Ordentlich, M. Weinberger, and G. Seroussi, “A low-complexity modeling approach for embedded coding of wavelet coefficients,” *Proceedings DCC ’98. Data Compression Conference*, 1998, pp. 408–417.

[10] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, June 1996, pp. 243–250.

[11] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, Dec. 1993, pp. 3445–3462.

[12] Z. Xiong, K. Ramchandran, and M. T. Orchard, “Space-frequency quantization for wavelet image coding,” *IEEE Transactions on Image Processing*, vol. 6, no. 5, May 1997, pp. 677–693.

[13] S. Appadwedula, D. L. Jones, K. Ramchandran, and L. Qian, “Joint source channel matching for wireless image transmission,” *Proceedings International Conference on Image Processing*, vol. 2, 1998, pp. 137–141.

[14] J. Lu, A. Nosratinia, and B. Aazhang, “Progressive source-channel coding of images over bursty error channels,” *Proceedings International Conference on Image Processing*, 1998, pp. 127–131.

[15] J. W. Modestino and D. G. Daut, “Combined source-channel coding of images,” *IEEE Transactions on Communications*, vol. 27, no. 11, Nov. 1979, pp. 1644–1659.

[16] M. Ruf and J. Modestino, “Rate-distortion performance for joint source and channel coding of images,” *Proceedings International Conference on Image Processing*, vol. II, 1995, pp. 77–80.

[17] M. J. Ruf, “A high performance fixed rate compression scheme for still image transmission,” *Proceedings DCC ’94. Data Compression Conference*, 1994, pp. 294–303.

[18] M. J. Ruf, “Source and channel coding of images using the operational rate-distortion function,” *International Symposium on Information Theory and its Applications (ISITA)*, Victoria, Canada, Oct. 1996, pp. 751–754.

[19] P. G. Sherwood and K. Zeger, “Progressive image coding on noisy channels,” *Proceedings DCC ’97. Data Compression Conference*, 1997, pp. 72–81.

[20] B. Hochwald and K. Zeger, “Tradeoff between source and channel coding,” *IEEE Transactions on Information Theory*, vol. 43, no. 5, Sept. 1997, pp. 1412–1424.

[21] G. Zemor and G. D. Cohen, “The threshold probability of a code,” *IEEE Transactions on Information Theory*, vol. 41, no. 2, Mar. 1995, pp. 469–477.

[22] H. Man, F. Kossentini, and M. J. Smith, “A class of EZW image coders for noisy channels,” *Proceedings International Conference on Image Processing*, vol. 3, 1997, pp. 90–93.

[23] J. Cai, C. W. Chen, and Z. Sun, “Error resilient image coding with rate-compatible punctured convolutional codes,” *Proc. IEEE Int. Symp. Circuits and Systems, 1998*, vol. 4, 1998, pp. 110–113.

[24] I. Kozintsev, J. Chou, and K. Ramchandran, “Image transmission using arithmetic coding based continuous error detection,” *Proceedings DCC ’98. Data Compression Conference*, 1998, pp. 339–348.

[25] H. Li and C. W. Chen, “Joint source and channel optimized TCQ with layered transmission and RCPC,” *Proceedings International Conference on Image Processing*, 1998, pp. 644–648.

[26] A. E. Mohr, E. A. Riskin, and R. E. Ladner, “Recovering from bit errors in scalar-quantized discrete wavelet transformed images,” *Proceedings International Conference on Image Processing*, 1998, pp. 502–506.

[27] P. G. Sherwood and K. Zeger, “Error protection of wavelet coded images using residual source redundancy,” *Thirty-First Asilomar Conference on Signals, Systems and Computers*, vol. 2, 1997, pp. 980–984.

[28] P. G. Sherwood and K. Zeger, “Progressive image coding for noisy channels,” *IEEE Signal Processing Letters*, vol. 4, no. 7, July 1997, pp. 189–191.

[29] S. Emami and S. L. Miller, “DPCM picture transmission over noisy channels with the aid of Markov model,” *IEEE Transactions on Image Processing*, vol. 4, no. 11, Nov. 1995, pp. 1473–1481.

[30] J. Hagenauer, “Source-controlled channel decoding,” *IEEE Transactions on Communications*, vol. 43, no. 9, Sept. 1995, pp. 2449–2457.

[31] K. Sayood and J. C. Borkenhagen, “Use of residual redundancy in the design of joint source/channel coders,” *IEEE Transactions on Communications*, vol. 39, no. 6, June 1991, pp. 838–846.

[32] K. Sayood, F. Liu, and J. D. Gibson, “A constrained joint source/channel coder design,” *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 9, Dec. 1994, pp. 1584–1593.

[33] P. G. Sherwood and K. Zeger, “Error protection for progressive image transmission over memoryless and fading channels,” *Proceedings International Conference on Image Processing*, vol. 1, 1998, pp. 324–328.

[34] P. G. Sherwood and K. Zeger, “Error protection for progressive image transmission over memoryless and fading channels,” *IEEE Transactions on Communications*, vol. 46, no. 12, Dec. 1998, pp. 1555–1559.

[35] P. G. Sherwood, X. Tian, and K. Zeger, “Channel code blocklength and rate optimization for progressive image transmission,” *Wireless Communications and Networking Conference (WCNC)*, 1999, pp. 978–982.

[36] P. C. Cosman, J. K. Rogers, P. G. Sherwood, and K. Zeger, “Combined forward error control and packetized zerotree wavelet encoding for transmission of images over varying channels,” *IEEE Transactions on Image Processing* (to appear in 2000).

[37] P. C. Cosman, J. K. Rogers, P. G. Sherwood, and K. Zeger, “Image transmission over channels with bit errors and packet erasures,” *Thirty-Second Asilomar Conference on Signals, Systems and Computers*, 1998, pp. 1621–1625.

[38] P. G. Sherwood and K. Zeger, “Macroscopic multistage image compression for robust transmiss ion over noisy channels,” *Proceedings SPIE - The International Society for Optical Engineering*, vol. 3653, 1999, pp. 73–83.

[39] M. Verterli and J. Kovačević, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice Hall PTR, 1995.

[40] UCLA ICL, “`http://www.icsl.ucla.edu/˜ipl/psnr_results.html`.”

[41] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1995.

[42] W. C. Jakes, *Microwave Mobile Communications*. New York: Wiley-Interscience, 1974.

[43] T. S. Rappaport, *Wireless Communications: Principles & Practice*. Upper Saddle River, NJ: Prentice Hall PTR, 1996.

[44] A. A. El Gammal and T. M. Cover, “Achievable rates for multiple description,” *IEEE Transactions on Information Theory*, vol. 28, no. 6, Nov. 1982, pp. 851–857.

[45] V. A. Vaishampayan, “Design of multiple description scalar quantizers,” *IEEE Transactions on Information Theory*, vol. 39, no. 3, May 1993, pp. 821–834.

[46] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[47] T. P. O’Rourke, R. L. Stevenson, Y.-F. Huang, and D. J. Costello, Jr., “Improved decoding of compressed images received over noisy channels,” *Proceedings International Conference on Image Processing*, 1995, pp. 65–68.

[48] J. Hagenauer, “Rate-compatible punctured convolutional codes (RCPC codes) and their applications,” *IEEE Transactions on Communications*, vol. 36, Apr. 1988, pp. 389–400.

[49] N. Seshadri and C.-E. W. Sundberg, “List Viterbi decoding algorithms with applications,” *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, February/March/April 1994, pp. 313–323.

[50] G. Castagnoli, J. Ganz, and P. Graber, “Optimum cyclic redundancy-check codes with 16-bit redundancy,” *IEEE Transactions on Communications*, vol. 38, no. 1, Jan. 1990, pp. 111–114.

[51] T. V. Ramabadran and S. S. Gaitonde, “A tutorial on CRC computations,” *IEEE Micro*, vol. 8, no. 4, Aug. 1988, pp. 62–75.

[52] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[53] N. Tanabe and N. Farvardin, “Subband image coding using entropy-coded quantization over noisy channels,” *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 5, June 1992, pp. 926–943.

[54] Q. Chen and T. R. Fischer, “Robust quantization for image coding and noisy digital transmission,” *Proceedings DCC ’96. Data Compression Conference*, 1996, pp. 3–12.

[55] J. Garcia-Frias and J. D. Villasenor, “An analytical treatment of channel-induced distortion in entropy coded image subbands,” *Proceedings DCC ’97. Data Compression Conference*, 1997, pp. 52–61.

[56] A. C. Hung and T. H.-Y. Meng, “Error resilient pyramid vector quantization for image compression,” *Proceedings International Conference on Image Processing*, 1994, pp. 583–586.

[57] Z. Mohdyusof and T. R. Fischer, “Subband image coding using a fixed-rate lattice vector quantizer,” *Proceedings International Conference on Image Processing*, 1995, pp. 101–104.

[58] S. Appadwedula, D. L. Jones, K. Ramchandran, and I. Konzintsev, “Joint source channel matching for a wireless communication link,” *Proceedings ICC-98. International Conference on Communications*, 1998, pp. 482–486.

[59] V. Chande and N. Farvardin, “Joint source-channel coding for progressive transmission of embedded source coders,” *Proceedings DCC ’99. Data Compression Conference*, 1999, pp. 52–61.

[60] V. Chande, N. Farvardin, and H. Jafarkhani, “Image communication over noisy channels with feedback,” *Proceedings International Conference on Image Processing*, vol. 2, 1999, pp. 540–544.

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

[62] A. Mohr, E. Riskin, and R. Ladner, "Graceful degradation over packet erasure channels through forward error correction," *Proceedings DCC '99. Data Compression Conference*, 1999, pp. 462–475.

[63] D. P. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[64] G. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding: Turbo codes," *Proceedings ICC-93. International Conference on Communications*, 1993, pp. 1064–1070.

[65] R. G. Gallager, "Low density parity check codes," *IRE Transactions on Information Theory*, vol. 8, Jan. 1962, pp. 21–28.

[66] R. J. McEliece, D. J. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE Journal of Selected Areas of Communication*, Feb. 1998, pp. 140–152.

[67] D. J. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, Mar. 1999, pp. 399–431.

[68] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," *Proceedings ICC-95. International Conference on Communications*, 1995, pp. 54–59.

[69] S. Bennedetto, R. Garello, and G. Montorsi, "A search for good convolutional codes to be used in the construction of turbo codes," *IEEE Transactions on Communications*, vol. 46, no. 9, Sept. 1998, pp. 1101–1105.

[70] J. Rogers and P. Cosman, "Robust wavelet zerotree image compression with fixed-length packetization," *Proceedings DCC '98. Data Compression Conference*, 1998, pp. 412–428.

[71] J. Rogers and P. Cosman, "Wavelet zerotree image compression with packetization," *IEEE Signal Processing Letters*, vol. 5, no. 5, May 1998, pp. 105–107.

[72] K. R. Narayanan and G. L. Stüber, "List decoding of turbo codes," *IEEE Transactions on Communications*, vol. 46, no. 6, June 1998, pp. 754–762.

[73] B. S. Srinivas, E. A. Riskin, R. Ladner, and M. Azizoglu, "Progressive image transmission on a channel with memory," *Proceedings Thirty-Third Annual Allerton Conference*, 1995, pp. 265–274.

[74] N. Serrano, D. Schilling, and P. Cosman, “Quality evaluation for robust wavelet zerotree image coders,” *Proceedings of the Second Annual UCSD Conference on Wireless Communications*, Mar. 1999, pp. 128–134.

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

[76] S. A. Rizvi and N. M. Nasrabadi, “Predictive residual vector quantization,” *IEEE Transactions on Image Processing*, vol. 4, no. 11, Nov. 1995, pp. 1482–1495.

[77] N. Phamdo, N. Farvardin, and T. Moriya, “A unified approach to tree-structured and multistage vector quantization for noisy channels,” *IEEE Transactions on Information Theory*, vol. 39, no. 3, May 1993, pp. 835–850.

[78] F. G. Meyer, A. Z. Averbuch, J. O. Stromberg, and R. R. Coifman, “Multi-layered image compression,” *Proceedings SPIE - The International Society for Optical Engineering*, vol. 3458, 1998, pp. 128–139.

[79] L. Ozarow, “On a source coding problem with two channels and three receivers,” *Bell System Technical Journal*, vol. 59, Dec. 1980, pp. 1909–1921.

[80] M. Fleming and M. Effros, “Generalized multiple description vector quantization,” *Proceedings DCC ’99. Data Compression Conference*, 1999, pp. 3–12.

[81] V. K. Goyal and J. Kovačević, “Optimal multiple description transform coding of gaussian vectors,” *Proceedings DCC ’98. Data Compression Conference*, 1998, pp. 388–397.

[82] V. K. Goyal, J. Kovačević, R. Arean, and M. Vetterli, “Multiple description transform coding of images,” *Proceedings International Conference on Image Processing*, vol. 1, 1998, pp. 674–678.

[83] H. Jafakhani and V. Tarokh, “Multiple description trellis coded quantization,” *Proceedings International Conference on Image Processing*, vol. 1, 1998, pp. 669–673.

[84] S. Servetto, V. Vaishampayan, and N. Sloane, “Multiple description lattice vector quantization,” *Proceedings DCC ’99. Data Compression Conference*, 1999, pp. 13–22.

[85] S. Servetto, K. Ramchandran, V. Vaishampayan, and K. Nahrstedt, “Multiple description wavelet based image coding,” *IEEE Transactions on Image Processing* (to appear).

[86] S. D. Servetto, K. Ramchandran, V. Vaishampayan, and K. Nahrstedt, “Multiple-description wavelet based image coding,” *Proceedings International Conference on Image Processing*, vol. 1, 1998, pp. 659–663.

[87] X. Yang and K. Ramchandran, "Optimal multiple description subband coding," *Proceedings International Conference on Image Processing*, vol. 1, 1998, pp. 654–658.

[88] Y. Wang, M. T. Orchard, and A. Reibman, "Multiple description image coding for noisy channels by pairing transform coefficients," *Proceedings IEEE Workshop on Multimedia Signal Processing*, 1997, pp. 419–424.

[89] V. K. Goyal, J. Kovačević, and M. Vetterli, "Quantized frame expansions as source-channel codes for erasure channels," *Proceedings DCC '99. Data Compression Conference*, 1999, pp. 326–335.

[90] W. Jiang and A. Ortega, "Multiple description coding via polyphase transform and selective quantization," *Proceedings SPIE - The International Society for Optical Engineering*, no. 3653, 1999, pp. 998–1008.

[91] A. Miguel, A. Mohr, and E. Riskin, "SPIHT for generalized multiple description coding," *Proceedings International Conference on Image Processing*, vol. 3, 1999, pp. 842–846.

[92] R. Puri and K. Ramchandran, "Multiple description souce coding using forward error correction codes," *Thirty-Third Asilomar Conference on Signals, Systems and Computers*, 1999.

VITA

Phillip Gregory Sherwood was born on September 30, 1966, in Atlanta, Georgia. He received the B.S. degree in physics and the B.S. and M.S. degrees in electrical engineering in 1990 all from the Massachusetts Institute of Technology. From 1993 to 1999 he was a research assistant in the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign, and from 1996 to 1999 he was a visiting scholar at the University of California, San Diego.

From 1990 to 1993, he was a software/systems engineer with Engineering Research Associates, Vienna, VA. From 1997 to 1999, he was with TRW Avionics Systems Division, San Diego, CA, working on voice and data communications systems. In 1999, he joined PacketVideo, San Diego, CA, where he works on multimedia applications for embedded and wireless systems. He has been a member of the Institute of Electrical and Electronic Engineers since 1988. He has co-authored the following papers:

- P. G. Sherwood, X. Tian, and K. Zeger, "Channel code blocklength and rate optimization for progressive image transmission," *Wireless Communications and Networking Conference (WCNC)*, 1999, pp. 978–982.

- P. C. Cosman, J. K. Rogers, P. G. Sherwood, and K. Zeger, "Combined forward error control and packetized zerotree wavelet encoding for transmission of images over varying channels," *IEEE Transactions on Image Processing* (to appear in 2000).

- P. G. Sherwood and K. Zeger, "Macroscopic multistage image compression for robust transmission over noisy channels," *Proceedings SPIE - The International Society for Optical Engineering*, vol. 3653, 1999, pp. 73–83.

- P. G. Sherwood and K. Zeger, “Error protection for progressive image transmission over memoryless and fading channels,” *IEEE Transactions on Communications*, vol. 46, no. 12, Dec. 1998, pp. 1555–1559.

- P. C. Cosman, J. K. Rogers, P. G. Sherwood, and K. Zeger, “Image transmission over channels with bit errors and packet erasures,” *Thirty-Second Asilomar Conference on Signals, Systems and Computers*, 1998, pp. 1621–1625.

- P. G. Sherwood and K. Zeger, “Error protection for progressive image transmission over memoryless and fading channels,” *Proceedings International Conference on Image Processing*, vol. 1, 1998, pp. 324–328.

- P. G. Sherwood and K. Zeger, “Error protection of wavelet coded images using residual source redundancy,” *Thirty-First Asilomar Conference on Signals, Systems and Computers*, vol. 2, 1997, pp. 980–984.

- T. Frajka, P. G. Sherwood and K. Zeger, “Progressive image coding with spatially variable resolution,” *Proceedings International Conference on Image Processing*, 1997, vol. 1, pp. 53-56.

- P. G. Sherwood and K. Zeger, “Progressive image coding for noisy channels,” *IEEE Signal Processing Letters*, vol. 4, no. 7, July 1997, pp. 189–191.

- P. G. Sherwood and K. Zeger, “Progressive image coding on noisy channels,” *Proceedings DCC ’97. Data Compression Conference*, 1997, pp. 72–81.

LIST OF TABLES

Table **Page**

4.1 Distortion values for the three algorithms over a range of channel conditions. HYBR and RCPC were optimized for the the channel with average SNR of 13 dB and normalized Doppler spread of 0.0001. PZW is the packetized zerotree wavelet scheme [70]. RCPC is the coder in [19]. HYBR is the proposed combined scheme. Column A shows the mean decoded MSE and column B shows a weighted mean using the weighting function from Example (ii): $W(x) = u(-x + 818.6)$. 69

4.2 Distortion values for the three algorithms over a range of channel conditions. For these channels, the fading parameters were held constant while the packet erasure channel parameters were varied. HYBR and RCPC were optimized for the channel with average SNR of 13 dB and normalized Doppler spread of 0.0001. PZW is the packetized zerotree wavelet scheme [70]. RCPC is the coder in [19]. HYBR is the proposed combined scheme. Column A shows the mean decoded MSE and column B shows a weighted mean using the weighting function from Example (ii): $W(x) = u(-x + 818.6)$. 71

4.3 RCPC codes. 73

5.1 Performance results for 512×512 Lena sent over a packet erasure channel. Values are PSNR (dB) converted from mean MSE. 90

LIST OF FIGURES

Figure **Page**

2.1 Block diagram of typical image coding system. 10
2.2 Demonstration of progressive image coding for the 512×512 Lena image using the SPIHT coder. 11
2.3 An example wavelet domain image demonstrating the tree-structured dependency of coefficients. 13
2.4 Coding example of EZW algorithm on a three-level wavelet-domain image. . . 14
2.5 Memoryless channel models. 16
2.6 Gilbert-Elliot fading channel model. 17
2.7 Combined packet erasure and fading channel. 20

3.1 Demonstration of bit error sensitivity of the SPIHT (without arithmetic coding) coder using the 512×512 image Baboon. 22
3.2 Distribution of path search depths given that the Viterbi decoder output fails the CRC check. 26
3.3 Trade-off between the likelihood of incomplete decoding and image quality given complete decoding, for convolutional code rates from $1/4$ to $1/3$. 27
3.4 Comparison of the performance of the RCPC/CRC + SPIHT coder with previously published results for image Lena transmission over a BSC. 28
3.5 Comparison of the performance of the RCPC/CRC + SPIHT coder with previously published results for image Goldhill transmission over a BSC. 29
3.6 Example images demonstrating the performance of the RCPC/CRC + SPIHT coder over the BSC. 31
3.7 Variation of coder performance with error rate. 32
3.8 Distortion vs. rate with and without channel coding. 34
3.9 Three possible progressive performance measures each equal to the area of the shaded region. 37
3.10 Performance results for the 512×512 Lena image transmitted over an erasure channel. 41
3.11 Performance results for the 512×512 Lena image using turbo codes over a BSC with BER $= 0.05$. 46
3.12 Performance comparison between RCPC/CRC, LDPC, and turbo codes. 46

4.1 Schematic diagram of RCPC/CRC and RS product code. 49

4.2 Comparison of code performance for the 512×512 Lena image over a BSC with BER $= 0.1$. 52
4.3 Comparison of code performance for the 512×512 Lena image over a Gilbert-Elliot channel. 54
4.4 Cumulative distribution plot for performance evaluation. 55
4.5 Comparison of code performance for the 512×512 Lena image transmitted over a Rayleigh fading channel with total rate 0.25 bpp. 57
4.6 Comparison of code performance with different trellis search depths for the 512×512 Lena image transmitted over a Rayleigh fading channel with total rate 0.25 bpp. 57
4.7 Comparison of code performance for the 512×512 Lena image transmitted over a Rayleigh fading channel with total rate 0.25 bpp. 59
4.8 Block diagram of the proposed coder. 60
4.9 The shaded area equals the fidelity criterion J_Φ with $W(x) = 1$ which is the mean decoded MSE in this case. $F(x)$ in the graph represents a generic cumulative distribution function. 63
4.10 The shaded area equals the fidelity criterion J_Φ with $W(x) = u(-x + x_h)$. . . . 64
4.11 The shaded area equals the fidelity criterion J_Φ with $W(x) = u(x - x_l)$. 64
4.12 An admissibility function which places an upper bound on the acceptable probability of exceeding a certain MSE. Two of the three cumulative distributions satisfy the constraint. 65
4.13 An admissibility function requiring minimum performance levels for two MSE ranges. Only one of the three distributions satisfies the constraints. 65
4.14 Images displayed here show the visual effects of loss for the hybrid coder (left) vs. the SPIHT+RCPC/CRC coder (right) at equal MSE (PSNR $= 23.5$dB). As shown in [74], the reconstructed images with more localized distortions tend to be preferred over the ones with global blurriness, both in terms of subjective quality and in terms of image content recognizability. 67
4.15 Cumulative distributions of decoded MSE for the 512×512 Lena image transmitted over Rayleigh fading channels with a transmission rate of 0.25 bpp. . . . 70
4.16 Cumulative distributions of decoded MSE for the 512×512 Lena image over channels with varying packet erasure parameters and fading parameters $f_D = 10^{-4}$, $\overline{\text{SNR}} = 12$ dB. (Total transmission rate 0.25 bpp.) 74
4.17 Images displayed here show the median quality for the three algorithms under different channel conditions. The channel used for the top row of images was dominated by uniform packet erasures. The second row includes representative images over a channel with packet erasures occurring in bursts. The channel for the bottom row of images has no packet erasures but has a higher probability of bit errors, which occur in long bursts. The overall transmission rate was 0.25 bpp. 75

5.1 Structure of multistage (residual) source encoder. 78

5.2 Comparison of two multistage coders with results from [22] for the 512×512 Lena image at 0.25 bpp total transmission rate sent over a BSC with different error rates. 84
5.3 Example decoded images for multistage coder 2 for channel BER $= 0.01$ and 0.05. 85
5.4 Trellis diagram for determining the optimal partitioning of wavelet trees into packets. 87
5.5 Example images demonstrating the effect of erased trees on decoded image quality. 89
5.6 Example decoded images for multistage PZW under three erasure conditions. . 91
5.7 Block diagram of a general multistage encoder for the case of two stages. . . . 92
5.8 Abstract transmission model. 93
5.9 Multistage encoder implementation. 99
5.10 Single-stage encoding for an image. 101
5.11 Two-stage encoding block. 103
5.12 PDF of the input to stage 2. 103
5.13 FEC coding for an erasure channel. 111
5.14 Block diagram of an image coder for packet loss channels. 112
5.15 Example MD index assignment mappings for two descriptions. 113
5.16 Example of a three-dimensional MD mapping. 114
5.17 An example three-dimensional mapping which is capable of correcting one erasure. 115
5.18 An example of performance trade-off for two mappings. 116
5.19 Performance for two five-description MDSQ mappings. 117

LIST OF ACRONYMS

ATM asynchronous transfer mode. Network technology using 53 octet (424 bits) packets with 48 octet (384 bits) payload.

AWGN additive white Gaussian noise. An analog memoryless channel.

BER bit error rate. Average bit error probability of the channel.

bpp bits per pixel. Unit to express the rate of an encoded image after normalizing by the dimensions.

BPSK binary phase-shift keying. A two-symbol modulation scheme.

BSC binary symmetric channel. A discrete memoryless channel with a single parameter determining the bit error probability.

CDPD cellular digital protocol data. A method of transmitting data over existing cellular networks.

CRC cyclic redundancy code. A shortened cyclic code typically used for error detection.

DCT discrete cosine transform. A linear transform often used in image coding (e.g., the JPEG standard coder).

EZW Embedded Zerotree Wavelet. Shapiro's embedded wavelet-based image coder [11].

FEC forward error control.

JPEG Joint Photographic Experts Group. International group which sets standards on still image compression algorithms. The name is often used to refer to the compression algorithm.

LDPC low-density parity check. Linear block codes with a very sparse parity check matrix introduced by Gallager [65].

MD multiple description. A robust source coding structure in which source information is sent over separate channels each subject to failure.

MSE mean squared error. A distortion measure.

MSVQ multistage VQ. A reduced complexity VQ method using a cascade of quantizers.

MTU maximum transmission unit. The largest packet size supported by the network (e.g., the MTU for Ethernet is 1500 bytes).

PSNR peak signal-to-noise ratio. A common fidelity measure with a logarithmic scale used in image coding.

PZW Packetized Zerotree Wavelet. Robust zerotree coder designed for packet erasure channels by Rogers and Cosman [71].

RCPC rate-compatible punctured convolutional. A class of convolutional error correcting codes which allow a common encoding/decoding structure for a family of codes of many rates [48].

RMS root mean square.

RS Reed-Solomon. An error correcting block code.

SNR signal-to-noise ratio. A measure of the relative strength of the signal versus the noise.

SPIHT Set Partitioning in Hierarchical Trees. An embedded wavelet-based image coder by Said and Pearlman [10].

TSVQ tree-structured VQ. A reduced complexity VQ with quantizers arranged in a tree structure.

UEP unequal error protection. Refers to applying different levels of channel coding to the source bits based on their importance to the decoding.

VQ vector quantization. A quantization / lossy compression technique.