CHANNEL OPTIMIZED QUANTIZATION OF IMAGES OVER BINARY CHANNELS WITH MEMORY

by

Julian Cheng

A thesis submitted to the
Department of Mathematics and Statistics
in conformity with the requirements
for the degree of Master of Science (Engineering)

Queen's University
Kingston, Ontario, Canada
September 1997

Abstract

In response to the growing demand for mobile digital personal communication services, future wireless products will provide multimedia information transmission. In such applications voice, text, image and video data need to be reliably and efficiently transmitted over wireless communication channels. These channels are typically very noisy (high BER) and often exhibit memory. Based on Shannon's separation principle, source coding and channel coding are often treated as separate entities and interleaving techniques are used to render the channel memoryless. However, these operations limit the optimality that the system can achieve and often introduce additional complexity and delay.

This thesis addresses the technique of joint source-channel coding for image transmission over channels with memory as an alternative to the so-called tandem coding schemes. The channel model considered is a discrete binary additive noise channel, where the noise process is generated via stationary homogeneous Markov chains. The proposed Discrete Cosine Transform (DCT) based system consists of a channel optimized quantization scheme which exploits the channel memory by incorporating the characteristics of the correlated noise in the quantizer design. Experimental results show that this simple system – which employs a fixed bit allocation table – provides significant improvements over traditional tandem systems, especially during bad channel conditions (high bit error rate and noise correlation). The loss of optimality due to the use of the fixed bit allocation method is also assessed. It is shown that the loss is very small for various images; this suggests that a reduction in

the encoder/decoder complexity and the system bandwidth requirements can also be achieved.

Acknowledgements

First of all, I would like to thank my supervisor, Dr. Fady Alajaji for his excellent supervision, patience, careful proof-readings of my thesis, and support. He introduced me to the beautiful subject of information theory which was instrumental in conducting this work.

I am also very grateful to Dr. Bob Erdahl who provided me with a great opportunity to study communication theory at the Department of Mathematics and Statistics. When I first visited the Queen's campus in late November of 1995, he made many efforts for organizing a special one-day event to introduce me to the communication faculty in the department. Thank you Bob!

I would also like to thank my parents and friends for their encouragements and support. As a LaTeXnovice, I will not have been able to finish this thesis within the given amount of time without the numerous helps from Yi Song and Mark Earnshaw from the Department of Electrical and Computer Engineering.

I wish to thank Nortel for letting me use its computer facility for my thesis writing (after hours), while I was completing a summer internship there. I am also very thankful to my manager Dr. Jay Huang for granting me time off to present parts of my work at the Canadian Information Theory Workshop in Toronto.

This work was in part supported by R.S. McLaughlin Fellowship, the Natural Sciences and Engineering Research Council (NSERC) and the Telecommunications Research Institute of Ontario (TRIO).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Shannon's Separation Principle

One important problem addressed by the communication engineer is the achievement of reliable and efficient transmission of data signals from one point to another over an inherently noisy medium. Data signals may represent various types of information such as text, speech, image, video, or combination of these. Transmission usually takes place in space or time (storage) and the medium is often referred to as the communication channel which can be wired (twisted wire pair, cable) or wireless (satellite or radio channels).

Typically, a data source is compressed through a source encoder to remove its inherent redundancy; this results in reducing the bandwidth requirements. Such procedure is called source coding. Source redundancy is usually manifested by the source memory or the non-uniformity of its distribution, or a combination of both (e.g., an asymmetric Markov source). Unfortunately this removal of the redundancy can, in turn, introduce a great deal of sensitivity to transmission noise. Therefore, the role of the channel encoder is to add *controlled* redundancy into the output of the source encoder in order to combat the channel noise. This procedure is called channel

coding. The output of the channel encoder is then sent over a digital channel *. The digital channel output is subsequently decoded via a channel decoder and a source decoder. A replica of the originally sent message is produced but often with distortion. Given a source, a channel, and a distortion measure, the goal of a communication system engineer is to minimize the overall distortion under various constraints (on the system delay, spectrum, complexity, etc.). A communication system formed in this way is called a *tandem* source-channel coding system. In a tandem system, source and channel coders are separately designed and concatenated to form a complete system. Its optimality is justified by Shannon's source-channel separation principle [39], [40].

Shannon's source-channel separation principle is the direct consequence of his source and channel coding theorems. Assume we are transmitting a discrete, stationary and ergodic source over a discrete memoryless channel. According to Shannon's source coding theorem, there exists a mapping between the source samples and the reproduction codewords such that for a given distortion D, $R(D)$ bits per source sample are sufficient to enable the reconstruction of the source samples with an average distortion not exceeding D when we operate on source blocks with arbitrarily long length. Thus $R(D)$ provides a theoretical lower bound for the source coding rate R at given fidelity D. Conversely, if R is the given source coding rate, $D(R)$ is the minimum possible average distortion. Here $R(D)$ is called the rate-distortion function and its inverse $D(R)$ is called the distortion-rate function [6]. According to Shannon's channel coding theorem, information can be transmitted reliably (with asymptotically vanishing probability of decoding error) at rate R below a so-called capacity C of the channel. Conversely, if the information rate is greater than the channel capacity ($R > C$), then the probability of decoding error will approach one. Combining both source and channel coding theorems, we get the *information transmission theorem* [24], [31] which states that when information is transmitted over a

*A digital (or discrete) channel consists of the modulator, the waveform channel, and the demodulator. This thesis only considers digital channels.

noisy channel, it is possible to obtain a reconstruction with fidelity D, provided that its capacity C is greater than $R(D)$, i.e., $C > R(D)$. The significance of this theorem justifies the Shannon's separation principle since the channel only controls the rate of the source but not the accuracy.

1.2 Literature Review

A tandem system based on Shannon's separation principle has a noticeable drawback: both source and channel coding theorems suggest we need to operate on codes with arbitrarily long block length in order to achieve optimality. In practice, this translates into large delay and complexity. In other words, the optimality of the tandem system is constrained by the encoder/decoder delay and complexity. This drawback has motivated some researchers to investigate the design of source codes and channel codes *jointly*. By joint design, we mean that some characteristics of the source, channel and distortion measure are incorporated into the design procedure. A system in which the source and channel codes are jointly designed is called a *joint* source-channel coding system.

Joint source-channel coding can in general be classified into three categories [34]: (1) unequal error protection approach; (2) zero-redundancy channel coding approach; and (3) combined source-channel coding approach.

In the unequal error protection approach, the basic idea is to treat the information bits unequally and to use channel coding in such a way that the added bits provide the highest protection to the most important class of bits, provide the second highest protection to the second most important class of bits, and so on. In some cases, the least significant bits are left totally unprotected. As an example in [32], Modestino and Daut used 2-D PCM as the source encoder and provided selective error control protection on those bits which contribute most significantly to the image reconstruction. In a similar work in [33] the authors applied the same idea for images with

transform coding techniques. In that case, since not all transform coefficients carry the same amount of information, they are unequally channel coded.

In zero-redundancy channel coding, as the name implies, no channel coding technique is provided. Instead, it uses the residual redundancy from the source encoder output to combat the channel noise. In [38], the residual redundancy in the DPCM system was studied and used in a proposed joint source-channel coding system. As a result, significant improvements were gained over other reference tandem systems, especially at bad channel conditions. The only trade-off is an added complexity at the decoder.

In the above two approaches, no channel information is used in the design of the joint source-channel coding system. In the combined source-channel coding approach, both the characteristics of the source *and* the channel are incorporated into the design of a single code. This single code replaces both the source and channel codes.

Combined source-channel coding is sometimes called channel-optimized or channel-matched coding and it has received considerable attention recently. In the late 60's, Kurtenbach and Wintz [26] followed the work of Lloyd [28] and Max [30] by deriving the necessary conditions of an optimal scalar quantizer design for noisy channels. Later Farvardin and Vaishampayan [12] conducted similar studies. After the discovery of Vector Quantization (VQ) [21], [27] in the early 80's, Kumazawa et al. [25] extended Kurtenbach and Wintz's work on the design of scalar quantizers for noisy channels to the kth dimension. The resulting vector quantizer is often called COVQ (Channel-Optimized-Vector-Quantizer). Farvardin et al. extensively studied the complexity and design issues related to COVQ [14] [13]. Similar techniques were used in the design of the Trellis waveform coder [5] and channel-matched Multi-Stage VQ and Tree-Structured VQ [35]. A salient feature of a COVQ system is that by incorporating the channel characteristics into the quantizer design, the resulting quantizer trades higher quantization distortion for smaller channel distortion such

that the overall distortion is minimized.

The boundary between the unequal error protection approach and the zero-redundancy coding approach can sometimes be blurred. In [3], [48], the authors used both the source encoder output residual redundancy and unequal error protection techniques to combat the channel noise. This raises the question on how to distribute the total number of available bits between the source codes and the channel codes for a tandem coding system. However, in general, no such guideline exists. Recent work by Goldsmith [20] addressed the problem of optimal bit distribution between the source encoder and channel encoder. For a joint source-channel coding system, such question may not exist (e.g., combined source-channel coding). In most of the previous joint source-channel coding work addressed above, the authors reported that good performance was obtained with joint source-channel coding schemes. The performance degradation of the system designed jointly was not as severe as that of the tandem coding system under bad channel conditions. Thus the joint source-channel coding system becomes an attractive alternative in wireless communication application where the BER rates are usually high.

Almost all of the previous work on joint source-channel coding assumed a memoryless Binary Symmetric Channel (BSC) model with the exception of [47], [36], [23]. The assumption of this simple channel model is sometimes too optimistic for most practical wireless communication channels which often exhibit memory. One such example is the fading channel caused by multi-path propagation. Traditionally, interleaving techniques are often used to render the channel memoryless. However interleaving/de-interleaving are known to lower the channel capacity and introduce extra delay into the system. In contrast, this thesis addresses the utilization of the channel memory, an unfavorable condition, and its incorporation into the design of the quantizers.

In a related work, Kumazawa et al. [25] studied the COVQ performance on real

images over the BSC and showed that greater performance results can be obtained compared to a conventional vector quantizer which does not take into account the channel characteristics. In [41], the authors developed a combined source/channel coding scheme in which rate-compatible punctured convolutional codes are used for channel error protection to transmit still images over noisy channels using subband coding. More recently, Chen and Fisher [9] proposed an all-pass filtering technique to design a robust COSQ (Channel Optimized Scalar Quantizer) system for the transmission of transform and subband coded images. The purpose of the all-pass filter is to transform any of a wide class of source distributions into a near-Gaussian distribution which has good pdf-optimized COSQ performance. All the previous work in this area rely on the simple binary symmetric channel model in spite of the fact that real-life communication channels have memory. This thesis deals with image compression and transmission over noisy channels with memory.

1.3 Contributions

The contributions of this thesis are as follows:

(i) Channel optimized vector quantizers (COVQ) are designed for channels with additive Markov noise. It incorporates the memory of the channel into the quantizer design to obtain performance gains over COVQ systems that employ interleaving. Numerical results for generalized Gaussian sources and experimental results for image sources are obtained.

(ii) A simple fixed bit allocation Discrete Cosine Transform (DCT) based COSQ combined source-channel coding system is proposed for transmitting grey-level images over a binary Markov channel. Experimental results are obtained for numerous images demonstrating both objectively and subjectively that such a system can achieve a superior performance over traditional tandem coding systems.

(iii) An integer programming technique is derived to obtain the optimal bit allocation of the DCT coefficients such that the overall distortion of the COSQ system is minimized. The performance loss due to the incorporation of the fixed bit allocation scheme in the COSQ system, as opposed to the use of optimal bit allocation, is investigated.

1.4 Thesis Outline

The rest of the thesis is organized as follows. In Chapter 2, we introduce the basic channel models, in particular a binary additive channel with memory. This chapter also includes a detailed description of quantization techniques, with emphasis on channel optimized quantizer design. In Chapter 3, we present the basic theory of transform coding and address the problem of optimal bit allocation of DCT coefficients in the context of noisy channels. A DCT-based combined source-channel coding system for image transmission over Markov channels is proposed and implemented in Chapter 4. We summarize the results and discuss various issues for future investigation in Chapter 5.

Chapter 2

Channel Optimized Quantization

2.1 Channel Models

2.1.1 Discrete Memoryless Channels

A discrete (discrete-time, finite-alphabet) communication channel with input X and output Y is characterized by: a finite input alphabet $\mathcal{X}$, a finite output alphabet $\mathcal{Y}$, and a set of transition probabilities

$$p(y|x) \triangleq Pr\{Y = y|X = x\},\ x \in \mathcal{X},\ \text{and}\ y \in \mathcal{Y} \tag{2.1}$$

which indicate the probability of observing $\{Y = y\}$ at the output of the channel given that input $\{X = x\}$ is transmitted. More generally, for a sequence of n input variables $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ and a corresponding output sequence $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_n)$, the channel is governed by a sequence of n-dimensional distributions:

$$p(\mathbf{y}|\mathbf{x}) \triangleq Pr\{\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}\} \tag{2.2}$$

where $\mathbf{x} \in \mathcal{X}^n$ and $\mathbf{y} \in \mathcal{Y}^n$.

A discrete *memoryless* channel (DMC) is a discrete channel with the property that given the channel input at time k, the corresponding channel output is independent

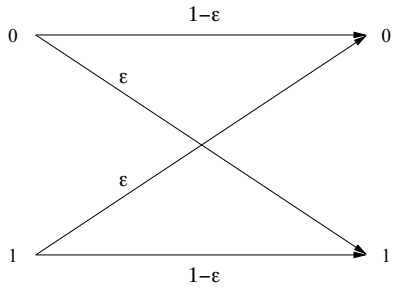


Figure 2.1: Binary symmetric channel model

of the channel inputs and outputs other than at time k. More specifically, if we are given a sequence of n successive inputs, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ with the corresponding output sequence $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, then for a DMC, we can write the conditional probability of $\mathbf{y}$ given $\mathbf{x}$ as

$$p(\mathbf{y}|\mathbf{x}) = \prod_{k=1}^{n} p(y_k|x_k). \tag{2.3}$$

One simple yet important discrete memoryless channel model is the binary symmetric channel (BSC) (Figure 2.1) with channel cross-over probability ϵ, where $\mathcal{X} = \mathcal{Y} = \{0, 1\}$ and $p(0|1) = p(1|0) = \epsilon$.

The most significant parameter associated with any channel is a number called the channel *capacity* (or *operational capacity*) and is denoted by C. The significance of the capacity C is manifested through Shannon's noisy channel coding theorem, which states that reliable transmission of information at a *fixed* data rate R is possible over a channel if and only if $R < C$ [39]. In order to characterize the channel capacity C of a DMC, we need first to define the term "mutual information". If $X = x$ is sent over the channel and $Y = y$ is received, then the *mutual information* between events $\{X = x\}$ and $\{Y = y\}$ is defined by

$$I(x; y) \triangleq \log_2 \frac{p(y|x)}{p(y)} = \log_2 \frac{p(x|y)}{p(x)}. \tag{2.4}$$

The mutual information can be considered as a measure of the amount of information that is transferred from the channel input to the channel output. More generally, the *mutual information* $I(X;Y)$ between random variables X and Y is given by:

$$\begin{aligned} I(X;Y) &\triangleq \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2\frac{p(y|x)}{p(y)} \\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2\frac{p(x|y)}{p(x)}. \end{aligned} \tag{2.5}$$

From the definition of entropy and conditional entropy [11], we can write the mutual information defined in Equation (2.5) as

$$I(X;Y) = H(Y) - H(Y|X) = H(X) - H(X|Y). \tag{2.6}$$

The two conditional entropies, $H(X|Y)$ and $H(Y|X)$, have important physical meanings. $H(X|Y)$, often called *equivocation*, can be interpreted as the amount of uncertainty about the input X given the received channel output Y; while $H(Y|X)$ can be considered as the amount of the noise entropy added to the channel (in case of additive noise channels). Formally, the capacity of a discrete memoryless channel is given by

$$C = \max_{p(x)} I(X;Y), \tag{2.7}$$

where the maximum is taken over all possible input distributions $p(x)$. Note that Equation (2.7) is the definition of the *information* capacity of the channel. Shannon's noisy coding theorem establishes that for a DMC the information capacity is equal to its *operational* capacity, which is defined as the maximum rate (in bits per channel use) at which information can be transmitted with arbitrarily low probability of error. In general, the calculation of the channel capacity given by Equation (2.7) doesn't always enjoy closed form expressions; numerical techniques are often employed for the capacity computation such as the ones described in [7].

For a BSC, its capacity C_{BSC} can be expressed via a simple closed form expression

$$C_{BSC} = 1 - h_b(\epsilon), \tag{2.8}$$

where ϵ is the channel cross-over error probability and $h_b(\cdot)$ is the binary entropy function, $h_b(\epsilon) \triangleq -\epsilon \log_2 \epsilon - (1-\epsilon)\log_2(1-\epsilon)$.

2.1.2 Channels with Memory

The assumption of "memorylessness" often contradicts the experimental data in real life communication channels: the channel error sequences are typically bursty. In practice the techniques of interleaving and de-interleaving are often employed to render the channel memoryless. However, it is well known that such operations may lower the channel capacity value (in the case of information stable channels [2]) and thus impose lower limits on the maximum data rate which can be reliably transmitted. In addition, the interleaving and de-interleaving operations can never be ideal; they often introduce delay into the system.

One large class of channels with memory is the class of finite-state channels [16], which includes the Gilbert-Elliott model [19], and is often used to model many realistic channels such as fading channels and channels with inter-symbol interference (ISI). In this thesis, we consider a more explicit family of channels with memory where the channel memory is exhibited via an additive noise process. More specifically, we consider a binary channel with additive noise described by

$$Y_i = X_i \oplus Z_i \tag{2.9}$$

for $i = 1, 2, 3, \ldots$, where $\oplus$ represents modulo 2 addition, and X_i, Z_i, and Y_i are the channel input, noise and output respectively. The input and noise sequences are assumed to be independent of each other. The noise process $\{Z_i\}_{i=1}^{\infty}$ is generated by the finite-memory contagion urn model described in [2]. According to this model,

the noise process $\{Z_i\}_{i=1}^{\infty}$ is a stationary ergodic Markov process of order M, i.e., for $i \geq M+1$,

$$Pr\{Z_i = e_i | Z_{i-1} = e_{i-1}, \ldots, Z_1 = e_1\} = Pr\{Z_i = e_i | Z_{i-1} = e_{i-1}, \ldots, Z_{i-M} = e_{i-M}\} \tag{2.10}$$

where $e_i \in \{0, 1\}$. Furthermore, $\{Z_i\}$ depends only on the sum of the M previous noise samples, and the noise transition probability is given by, for $i \geq M+1$,

$$Pr\{Z_i = 1 | Z_{i-M} = e_{i-M}, \ldots, Z_{i-1} = e_{i-1}\} = \frac{\epsilon + (\sum_{j=i-M}^{i-1} e_j)\delta}{1 + M\delta}, \tag{2.11}$$

where $e_j \in \{0, 1\}$, for $j = i-M, \ldots, i-1$, and ϵ is the channel bit error rate (BER), which determines the marginal distribution of the noise process,

$$Pr\{Z_i = 1\} = \epsilon = 1 - Pr\{Z_i = 0\}. \tag{2.12}$$

The non-negative parameter δ in Equation (2.11) determines the amount of correlation in $\{Z_i\}$ and it's a measure of the burstiness within the noise samples. The higher the noise correlation δ is , the more bursty the channel becomes. The correlation coefficient of the noise process is given by $\frac{\delta}{1+\delta}$ [2]. Note that when $\delta = 0$, the channel model reduces to the (memoryless) binary symmetric channel (BSC) with cross-over probability ϵ. We further observe that the above channel can be entirely described with only three parameters (ϵ, δ, and M). This channel model offers a possible alternative to finite-state channels such as the Gilbert-Elliott noise model [19].

In this thesis, we only consider the first-order ($M = 1$) Markov noise process case; the noise transition probability can be found from Equation (2.11)

$$Pr\{Z_i = 1 | Z_{i-1} = e\} = \frac{\epsilon + e\delta}{1 + \delta}, \tag{2.13}$$

where $e \in \{0,1\}$, or written in matrix form, the state transition probability becomes

$$\begin{bmatrix} Q(0|0) & Q(1|0) \\ Q(0|1) & Q(1|1) \end{bmatrix} = \begin{bmatrix} \frac{1-\epsilon+\delta}{1+\delta} & \frac{\epsilon}{1+\delta} \\ \frac{1-\epsilon}{1+\delta} & \frac{\epsilon+\delta}{1+\delta} \end{bmatrix}. \tag{2.14}$$

The capacity of this first-order Markov channel can be calculated as [2]

$$\begin{aligned} C &= \lim_{n\to\infty} \max_{p(x^n)} \frac{1}{n} I(X^n; Y^n) && (2.15)\\ &= \lim_{n\to\infty} \max_{p(x^n)} \frac{1}{n} \Big[H(Y^n) - H(Y^n|X^n) \Big] && (2.16)\\ &= 1 - \lim_{n\to\infty} \frac{1}{n} H(Z^n) && (2.17)\\ &= 1 - H(Z_2|Z_1), && (2.18) \end{aligned}$$

where $X^n \triangleq (X_1, X_2, \ldots, X_n)$, $Y^n \triangleq (Y_1, Y_2, \ldots, Y_n)$ and $H(Z_2|Z_1)$ is the entropy rate of the first-order Markov noise process. $H(Z_2|Z_1)$ can be calculated by using the method from [11], and we get

$$C = 1 - \Big[(1-\epsilon) h_b(\frac{\epsilon}{1+\delta}) + \epsilon h_b(\frac{\epsilon+\delta}{1+\delta}) \Big]. \tag{2.19}$$

Again, $h_b(\cdot)$ is the binary entropy function and the capacity is achieved with input distribution $p^*(x^n) = (\frac{1}{2})^n \ \forall n$. Note that the capacity of this channel model is monotonically increasing with δ (for fixed ϵ) and is monotonically decreasing with ϵ (for fixed δ), see Figure 2.2. When the noise correlation δ increases, the channel becomes more bursty; in the extreme case when $\delta \to \infty$, the noise process becomes deterministic ($H(Z_2|Z_1)$ goes to zero) and the channel becomes noiseless. When the noise correlation $\delta = 0$, we obtain (as expected) the capacity expression of the BSC

$$C = 1 - h_b(\epsilon). \tag{2.20}$$

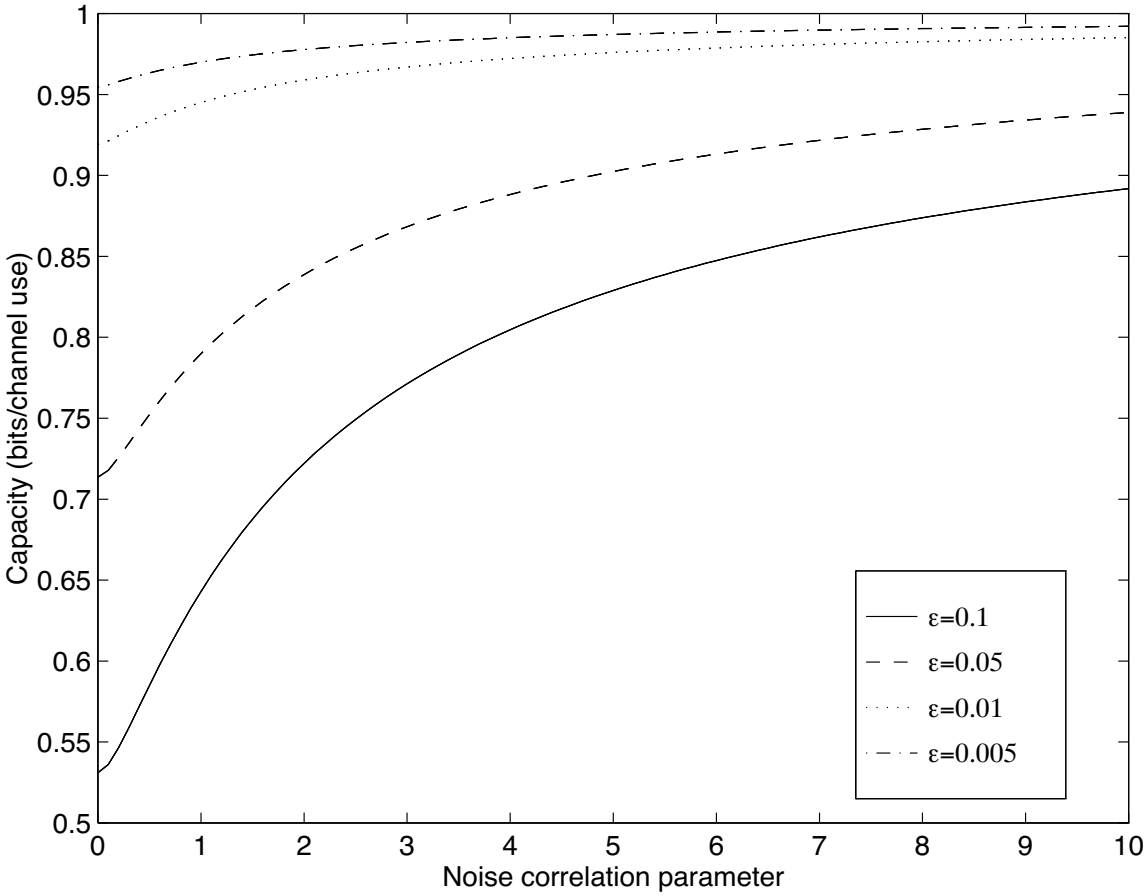


Figure 2.2: Markov channel capacity C versus noise correlation parameter δ for various BERs.

For an input block $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ and an output block $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_n)$, the block channel transition probability matrix $Pr\{\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}\}$ is given by [2], for $n \geq 2$,

$$\begin{aligned} Pr\{\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}\} &\stackrel{\triangle}{=} Q(\mathbf{y}|\mathbf{x}) && (2.21) \\ &= Pr\{\mathbf{Z} = \mathbf{e}\} && (2.22) \\ &= Pr(Z_1 = e_1) \prod_{i=2}^{n} \left[\frac{\epsilon + e_{i-1}\delta}{1+\delta}\right]^{e_i} \left[\frac{(1-\epsilon) + (1-e_{i-1})\delta}{1+\delta}\right]^{1-e_i} && (2.23) \end{aligned}$$

where $e_i = x_i \oplus y_i, i = 1, 2, \ldots, n$. This closed form channel distribution will be used later for designing channel optimized vector quantizers (COVQs).

2.2 Source Coding and Quantization

2.2.1 Source Coding

The purpose of source coding (also known as data compression) is to remove the redundant information in the source in order to reduce the bit rate requirements for its storage or transmission. The source redundancy can be classified into *statistical* redundancy and *non-statistical* redundancy. There are two types of data compression, lossless and lossy depending on what kind of redundancy is removed. The statistical redundancy is due to the source non-uniform distribution or the source memory or a combination of both. In lossless data compression, the goal is to remove all the statistical redundancy such that the recovered source has *zero* distortion with respect to the original source. In the context of image sources, the non-statistical redundancy can take the form of psycho-visual redundancy, thus certain properties of the human visual system can be exploited. This is due to the fact that the human vision system is insensitive to certain spatial frequencies. In lossy data compression, the goal is to

obtain the best possible fidelity for the source with a given bit rate, or equivalently to minimize the bit rate such that a specified fidelity is maintained. Lossy data compression is achieved via quantization; this will be treated in great details in the following subsections.

The statistical redundancy is related to the entropy or the entropy rate of the source. For a discrete memoryless source (DMS), $\{X_n\}$, the entropy of the source is the entropy of its first (or any) output $H(X_1)$. The entropy of any random variable X with alphabet $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ with known probabilities $p(x_i) \stackrel{\triangle}{=} p_i, i = 1, 2, \ldots, N$ is defined as the expectation of its self-information:

$$H(X) \stackrel{\triangle}{=} E[I(X)] = -\sum_{k=1}^{N} p_k \log_2 p_k. \tag{2.24}$$

It can be shown that

$$0 \leq H(X) \leq \log_2 N. \tag{2.25}$$

$H(X)$ measures the amount of randomness or uncertainty in the random variable X. For sources with memory (e.g., Markov source), the term entropy rate $H_\infty(X)$ is usually used and it is defined as

$$H_\infty(\mathbf{X}) \stackrel{\triangle}{=} \lim_{n\to\infty} \frac{1}{n}\mathbf{E}[-\log_2 p(\mathbf{X})] \tag{2.26}$$

provided the limit exists, where $\mathbf{X} = (X_1, X_2, \ldots, X_n)$. In general, for two sources with identical alphabets and letter probabilities,

$$\frac{1}{n}H(\mathbf{X})|_{memory} < \frac{1}{n}H(\mathbf{X})|_{memoryless} \leq \log_2 N. \tag{2.27}$$

To see this,

$$\frac{1}{n}H(\mathbf{X})|_{memoryless} \stackrel{\triangle}{=} \frac{1}{n}H(X^n)|_{memoryless} \tag{2.28}$$

$$
\begin{aligned}
&= \frac{1}{n}\sum_{i=1}^{n} H(X_i|X_{i-1},\ldots,X_1) && (2.29)\\
&= \frac{1}{n}\sum_{i=1}^{n} H(X_i) && (2.30)\\
&= H(X_1) && (2.31)\\
&\leq \log_2 N, && (2.32)
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{1}{n}H(X^n)|_{memory} &= \frac{1}{n}\sum_{i=1}^{n} H(X_i|X_{i-1},\ldots,X_1) && (2.33)\\
&\leq \frac{1}{n}\sum_{i=1}^{n} H(X_i) && (2.34)\\
&= H(X_1), && (2.35)
\end{aligned}
$$

where the inequality follows from the fact that conditioning reduces entropy.

The total statistical redundancy that a discrete stationary source $\{X_n\}$ contains can be expressed as

$$\rho_T = \log_2 N - H_\infty(\mathbf{X}). \tag{2.36}$$

This is the total amount redundancy one can remove through lossless data compression. ρ_T can further be decomposed into two types of redundancies: redundancy due to the source non-uniform distribution ρ_D; and redundancy due to the source memory ρ_M: $\rho_T = \rho_D + \rho_M$, where

$$\rho_D = \log_2 N - H(X_1), \tag{2.37}$$

and

$$\rho_M = H(X_1) - H_\infty(\mathbf{X}). \tag{2.38}$$

Shannon's (lossless) block source coding theorem states that the entropy (entropy rate) provides the theoretical lower bound for achieving minimum bit rate description

for a discrete stationary source without (with) memory. We next state Shannon's source coding theorem for sources with memory [8].

Theorem: Consider a discrete stationary ergodic source $\{X_n\}$ with entropy rate $H_\infty(\mathbf{X})$. It is possible to construct a (n,k) block encoder/decoder pair such that its coding rate $\frac{k}{n}$ is arbitrarily close to $H_\infty(\mathbf{X})$ and the probability of decoding error is arbitrarily close to zero for codes with sufficiently long block length. Conversely, if $\frac{k}{n} < H_\infty(\mathbf{X})$, then any (k,n) block encoder/decoder pair has probability of decoding error close to one for codes with sufficiently large n.

In lossy coding schemes, the achievable minimum bit rate is a function of the allowable distortion and this relationship is given by the rate distortion function $R(D)$. The *information* rate distortion function for a DMS $\mathcal{X}$ with distortion measure $d(X,\hat{X})$ is defined as

$$R^I(D) = \min_{p(\hat{x}|x):E[d(X,\hat{X})]\leq D} I(X;\hat{X}), \tag{2.39}$$

where minimization is over all $p(\hat{x}|x)$ for which the joint distribution $p(x,\hat{x})$ satisfies the expected distortion constraint. Equivalently, the *information* distortion rate function is defined as

$$D^I(R) = \min_{p(\hat{x}|x):I(X;\hat{X})\leq R} \mathbf{E}[d(X,\hat{X})]. \tag{2.40}$$

Shannon's rate-distortion theorem can be simply stated as follows [11].

Theorem: For a DMS $\mathcal{X}$ with distribution $p(x)$ and bounded distortion function $d(X,\hat{X})$, its operational rate-distortion $R(D)$ equals its information rate-distortion defined in Equation (2.39). That is $R^I(D)$ is the minimum achievable rate at distortion D.

2.2.2 Scalar Quantization

In many data-transmission systems, the conversion from continuous or discrete alphabet data to an approximating value from a pre-determined set is called quantization. In this subsection, we consider a simple memoryless and delayless scalar quantizer. The delayed-decision quantizer or vector quantizer will be treated in the next subsection.

A rate R, N-output level ($N = 2^R$), scalar quantizer q is a mapping from the real-line domain $\mathbf{R}$ to a set with finite number of elements,

$$q : \mathbf{R} \to \mathcal{C} \tag{2.41}$$

where

$$\mathcal{C} = \{y_1, y_2, \ldots, y_N\}. \tag{2.42}$$

The set $\mathcal{C}$ is called the codebook and the elements of the codebook y_k's are called the reproduction levels or the codewords. The real line $\mathbf{R}$ is partitioned into N non-overlapping regions, $S_i, i = 1, 2, \ldots, N$, described by

$$S_i = \{x : x \in \mathbf{R}, \quad q(x) = y_i\}, \quad i = 1, 2, \ldots, N \tag{2.43}$$

where S_i's satisfy

$$\bigcup_{i=1}^{N} S_i = \mathbf{R}, \tag{2.44}$$

and

$$S_i \bigcap S_j = \phi \quad (i \neq j). \tag{2.45}$$

The mapping is performed as follows: Let x be the input to the quantizer q, then the output of the quantizer $q(x)$ is obtained as

$$q(x) = y_i \text{ if } x \in S_i, \quad i = 1, 2, \ldots, N. \tag{2.46}$$

Therefore an instantaneous memoryless scalar quantizer can be completely defined by the codebook $\mathcal{C}$ and the partition set $\{S_i\}$. To evaluate the performance of the quantizer, we choose the mean square error (MSE) distortion measure: $d(x, y) = (x - y)^2$.

Let the random variable X be the input to the scalar quantizer with a known probability density function $p(x)$. The average (statistical) distortion D between the input X and the quantizer output $q(X)$ is defined as

$$D = \mathbf{E}[d(X, q(X))] = \sum_{i=1}^{N} \int_{S_i} (x - y_i)^2 p(x) dx. \tag{2.47}$$

When the quantizer input source is stationary and ergodic, the time averaged distortion converges to the statistical averaged distortion in probability, i.e.,

$$D = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} d(x_i, q(x_i)). \tag{2.48}$$

A classical quantization problem can often be described as follows. For a given independent and identically distributed (i.i.d.) source with known distribution $p(x)$, a given distortion measure, say $d(x, q(x)) = (x - q(x))^2$, and a given number of quantizer output levels N (fixed rate R), we wish to find the codebook $\mathcal{C}$ and the partition set $\{S_i\}$ such that the distortion defined in Equation (2.47) is minimized. The following design procedure which provides the solution to the above problem is the Lloyd-Max quantizer [28], [30].

Let us consider a quantizer with quantization regions of the form

$$S_i = \begin{cases} (x_{i-1}, x_i], & i = 1, 2, \ldots, N-1 \\ (x_{N-1}, x_N), & i = N \end{cases} \tag{2.49}$$

Here we adopt the convention $-\infty = x_0 < x_1 < x_2 < \ldots < x_N = \infty$. The average distortion can be written as

$$D = \sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} (x - y_i)^2 p(x) dx. \tag{2.50}$$

If we wish to minimize D for a fixed rate R, we can derive the *necessary* conditions by differentiating D with respect to x_i and y_i and get two sets of equations

$$x_i = \frac{y_i + y_{i+1}}{2}, \quad i = 1, 2, \ldots, N-1, \tag{2.51}$$

$$y_i = \mathbf{E}[X|X \in S_i] = \frac{\int_{x_{i-1}}^{x_i} x p(x) dx}{\int_{x_{i-1}}^{x_i} p(x) dx} \quad i = 1, 2, \ldots, N. \tag{2.52}$$

Equation (2.51) states that the threshold level x_i should be the midway between two adjacent reconstruction levels. Equation (2.52) suggests that the optimal reconstruction level y_i is the *centroid* of the area under $p(x)$ between x_{i-1} and x_i. Thus the problem of designing an "optimal" quantizer can be divided into two conceptually independent problems: (i) Given the code book $\mathcal{C}$, find the best partition of $\mathbf{R}$; (ii) Given the partition of the real line $\mathbf{R}$, find the optimal codebook $\mathcal{C}$ such that the average distortion is minimized. Equations (2.51) and (2.52) can be solved iteratively for x_k's and y_k's with the initial guess of y_1. After each iteration step, the averaged distortion D is calculated. The algorithm continues until the relative decrease in D from two consecutive iterations is less than a pre-defined threshold. The resulting quantizer is called the Lloyd-Max Quantizer. It should be emphasized that the obtained quantizer result is only a local optimal solution depending on the initial conditions. Fleischer [15] later showed that these conditions specified in Equations (2.51) and (2.52) are also *sufficient* conditions for source distributions which are *log concave*, i.e. if $\partial^2 \log p(x)/\partial x^2$ is negative. This includes sources with Gaussian, Laplacian and Rayleigh distributions. Hence the quantizer defined by Equation (2.51) and (2.52) is *globally* optimal for such distributions.

The above approach assumes the squared error distortion measure. Fortunately, we can extend the results in Equations (2.51) and (2.52) to more general distortion measures $d(\cdot,\cdot)$. These two general necessary conditions are known as the *Nearest Neighbor Condition* and the *Generalized Centroid Condition* [18].

- **Nearest Neighbor Condition**: For a given codebook $\mathcal{C}$ with its reproduction levels $y_1, y_2, \ldots, y_N$, the optimal partitions S_i's satisfy

$$S_i = \{x : d(x, y_i) \leq d(x, y_j), \quad \forall j \neq i\}, \tag{2.53}$$

- **Generalized Centroid Condition**: For a given set of partitions $\{S_i\}$, and an input random variable X, the optimal reproduction levels satisfy

$$y_i = \arg\min_y \mathbf{E}[d(X, y)|X \in S_i], \quad \forall i \in \{1, 2, \ldots, N\}. \tag{2.54}$$

These two generalized conditions can be used iteratively to obtain a locally optimal quantizer solution. The expectation operation within the generalized centroid condition implies that the probability density distribution of the source is known. In practice, training sequences from the source are used to obtain an empirical source distribution.

The necessary conditions for an optimal quantizer design can be extended to noisy channel quantizer design as well. Kurtenbach and Wintz [26] first expressed the necessary conditions of the optimal quantizer design for discrete memoryless channels. For a quantizer with rate R, the noisy channel quantizer can be treated as two separate mappings, an encoder mapping q_e and an decoder mapping q_d,

$$q_e : \mathrm{R} \to \mathcal{I} \tag{2.55}$$

$$q_d : \mathcal{I} \to \mathcal{C} \tag{2.56}$$

where $\mathcal{I} = \{b(1), b(2), \ldots, b(2^R)\}$ is a set of binary sequence of R digits representing the index of the reproduction level. After the index $b(i)$ is transmitted through the channel, index $b(j)$ is received which may not be the same as $b(i)$ due to channel errors. Let $p(b(j)|b(i))$ be the probability that the index $b(j)$ is received when the index $b(i)$ is sent. Then the mean square error distortion is written as

$$D = \sum_{i=1}^{N} \sum_{j=1}^{N} p(b(j)|b(i)) \int_{S_i} p(x)(x - y_j)^2 dx, \tag{2.57}$$

where the partition set S_i for the index $b(i)$ is described by

$$S_i = \{x : x \in \mathrm{R}, \quad q_e(x) = b(i), \quad b(i) \in \mathcal{I}\}. \tag{2.58}$$

The necessary conditions for optimality become [26]

$$S_i = \left\{ x : \sum_{j=1}^{N} p(b(j)|b(i))(x - y_j)^2 \leq \sum_{j=1}^{N} p(b(j)|b(l))(x - y_j)^2, \quad \forall l \neq i \right\}, \tag{2.59}$$

and

$$y_j = \frac{\sum_{i=1}^{N} p(b(j)|b(i)) \int_{S_i} x p(x) dx}{\sum_{i=1}^{N} p(b(j)|b(i)) \int_{S_i} p(x) dx}. \tag{2.60}$$

If we rearrange the terms, Equation (2.57) can be rewritten as

$$D = \sum_{i=1}^{N} \int_{S_i} \underbrace{\sum_{j=1}^{N} p(b(j)|b(i))(x - y_j)^2}_{d'} p(x) dx \tag{2.61}$$

Comparing (2.61) with (2.47), we note that these two equations are identical except that (2.61) uses a *modified* distortion measure d'. Furthermore, when the channel becomes noiseless, (2.59) and (2.60) can be simplified to the general necessary conditions for the noiseless case described by (2.53) and (2.54).

2.2.3 Vector Quantization

Vector Quantization (VQ) is a data compression scheme which maps a sequence of continuous or discrete *vectors* into one of finite number of pre-determined vectors called reproduction levels. Vector Quantization has been used very successfully in image and speech coding [21], [29], [18]. It can be shown that VQ can offer various advantages over scalar quantization as VQ exploits the statistical redundancy between the source samples. Indeed the performance of VQ approaches the rate-distortion bound when the dimension of the sample block goes to infinity. A brief but formal description of vector quantization is herein presented.

A vector $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ consisting of k source samples represents an element of the Euclidean space $\mathbf{R}^k$ with k-fold probability density function $p(\mathbf{x})$. An N-level, k-dimensional vector quantizer (VQ) is defined by the codebook, $\mathcal{C} = \{\mathbf{y}_i, i = 1, \ldots, N\}$ with N reproduction vectors; and the partition set, $\mathcal{S} = \{S_i; i = 1, \ldots, N\}$ consisting of subspaces of the kth-dimension Euclidean space $\mathbf{R}^k$. The operation of VQ is essentially a mapping $q(\cdot)$,

$$q(\mathbf{x}) = \mathbf{y}_i \quad \text{if} \quad \mathbf{x} \in S_i \tag{2.62}$$

where $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{ik})$ and the partition $\mathcal{S}$ satisfies

$$\bigcup_{i=1}^{N} S_i = \mathbf{R}^k, \qquad S_i \bigcap S_j = \phi. \tag{2.63}$$

After the source vector is mapped to one of the reconstruction levels, the *index* of the subspace to which the input vector $\mathbf{x}$ belongs is sent over the channel. The rate of the quantizer is defined as

$$R = \frac{1}{k} \log_2 N \quad \text{bits/sample.} \tag{2.64}$$

Element by element square-error distortion is commonly used to evaluate the error incurred by the quantizer when the input vector $\mathbf{x}$ is reconstructed as $\mathbf{y}$

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k} (x_i - y_i)^2. \tag{2.65}$$

To measure the performance of the VQ, we define the distortion per sample in the mean square error sense,

$$D = \frac{1}{k} \sum_{i=1}^{N} \int_{S_i} p(\mathbf{x}) d(\mathbf{x}, \mathbf{y}_i) d\mathbf{x}. \tag{2.66}$$

Thus, given the number of reproduction levels N, and the dimension of the source sample block k, the goal is to find the codebook $\mathcal{C}^*$ and the partition set $\mathcal{S}^*$ such that the mean square error distortion defined in Equation (2.66) is minimized.

The solution to the above problem was first proposed by Linde, Buzo and Gray [27]. In this now classical paper, the authors provide an iterative algorithm to design a locally optimum vector quantizer and the resulting VQ is often called LBG-VQ. For a given partition set $\mathcal{S}$, the necessary conditions for minimizing Equation (2.66) *without* considering channel errors are

$$\mathbf{y}_i = \frac{\int_{S_i} \mathbf{x} p(\mathbf{x}) d\mathbf{x}}{\int_{S_i} p(\mathbf{x}) d\mathbf{x}}. \tag{2.67}$$

If we assume the codebook $\mathcal{C}$ is given, when the channel errors are not considered, then the partition set $\mathcal{S}$ which minimizes Equation (2.66) is given by

$$S_i = \{\mathbf{x} | d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_l) \quad \forall i \neq l\}. \tag{2.68}$$

Equation (2.67) is the generalized centroid condition. When $k = 1$, it reduces to the scalar quantization condition given by Equation (2.52). The region defined in Equation (2.68) is called Voronoi region. Equations (2.67) and (2.68) are used iteratively to update the codebook and the partition set. Note that after each iteration, the distortion per sample decreases [27]; therefore, the LBG-VQ algorithm in general converges to a locally optimal solution. The choice of the initial codebook plays an important role in the LBG-VQ algorithm. There are several methods to choose the

initial codebook required by the algorithm. One method is to choose the first N vectors in the training sequence; another widely used initializing technique is called initial guessing by splitting [27]. In using the splitting algorithm on a training sequence, one starts with a one-level quantizer consisting of the centroid of the training sequence. Then this vector is split into two vectors which are used for designing a two-level quantizer. This process continues until we obtain the initial code vectors for the desired N-level quantizer.

One obvious shortcoming of the LBG-VQ algorithm is that the delay and computational complexity increase exponentially with kR since a full-search is needed to map every input block to the best matching reconstruct vector. More practical VQs such as Tree-Structured VQ or Multi-Stage VQ trade storage for reduced computational complexity, but they are sub-optimal [18].

2.3 Channel Optimized Vector Quantization

2.3.1 Introduction

The vector quantizer design is the generalization of the Lloyd-Max quantizer to the k-th dimension. Channel optimized vector quantization (COVQ) described in this section is the generalization of the noisy channel scalar quantizer studied by Kurtanbach and Wintz to the kth dimension. The basic idea of COVQ is to design a vector quantizer by incorporating the various channel parameters into the design procedure. The resulting quantizer trades quantization error for channel error such that the overall end-to-end distortion is minimized. In COVQ design, perfect *a priori* knowledge of the channel state information is assumed. It has been shown that the vector quantizer for noisy channel can achieve good performance *without* adding extra redundancy for error protection, especially for very noisy channel environments. In a recent work by Goldsmith et al. [20], the authors introduced an additional channel protection scheme

that exploits the channel soft-decision information to the COVQ output while maintaining a fixed overall encoder bit rate to achieve better performance. In this thesis, we do not consider such hybrid systems. COVQ has been extensively studied for ideal and image sources. Kumazawa et al. first formulated the necessary conditions for the optimal COVQ solutions [25]. Later Farvardin and Vaishampayan studied the performance and the complexities of the COVQ system [14]. It was found that the encoding complexity of the COVQ system is proportional to the number of encoding regions. It is interesting to mention that when the channel gets extremely noisy, the number of the encoding regions associated with the optimum system *becomes smaller*. That is for a very noisy channel, certain encoding regions becomes empty such that it is better not to send those indices associated with empty regions over the noisy channel at all. In [13], Farvardin successfully applied a combinatorial optimization technique called simulated annealing to the initial codebook design for the optimal COVQ system. Most of the previous work in this area has been restricted to digital memoryless channels. We will state the principle using this simple channel model and extend the same principle for our Markov channel model.

2.3.2 COVQ Design Algorithm

Assume that the source to be encoded is a real-valued, stationary, and ergodic process $\{X_t; t = 0, 1, \ldots\}$ with zero mean and unit variance. The source is encoded with a k-dimensional, N-output level vector quantizer and the output of the VQ is transmitted over a discrete memoryless channel with input and output alphabets $\mathcal{J} = \{1, 2, \ldots, N\}$. The overall encoder-decoder operation, as illustrated in Figure 2.3, can be decomposed into three separate mappings, namely the encoding mapping, γ; the channel index mapping, b; and the decoding mapping, g

$$\gamma : \mathbf{R}^k \to \mathcal{J} \tag{2.69}$$

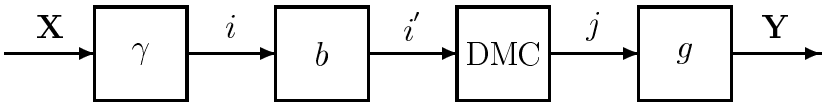


Figure 2.3: Block diagram of a COVQ system

$$b : \mathcal{J} \rightarrow \mathcal{J} \tag{2.70}$$

$$g : \mathcal{J} \rightarrow \mathbf{R}^k. \tag{2.71}$$

The encoding mapping, γ is described in terms of the partition $\mathcal{P} = \{S_1, S_2, \ldots, S_N\}$ of the k-dimensional Euclidean space $\mathbf{R}^k$ according to

$$\gamma(\mathbf{x}) = i, \quad \text{if} \ \ \mathbf{x} \in S_i, \ i \in \mathcal{J} \tag{2.72}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ is a source output vector consisting of k source samples. The channel index mapping, b is a one-to-one mapping, which assigns the encoder output i an index $i^{'} = b(i) \in \mathcal{J}$ and index $i^{'}$ is sent over the DMC. The DMC is characterized by the channel transition probability $p(j|i^{'})$ denoting the probability that the index j is received given that $i^{'}$ is transmitted. Finally, the decoding mapping is described in terms of the codebook $\mathcal{C} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ according to

$$g(j) = \mathbf{y}_j, \quad j \in \mathcal{J}. \tag{2.73}$$

Let us denote the distortion caused by representing the source vector $\mathbf{x}$ by a codeword $\mathbf{y}$ as $d(\mathbf{x}, \mathbf{y})$. The average overall distortion per source sample $D((\mathcal{P}, \mathcal{C}); b)$ is described by

$$D((\mathcal{P}, \mathcal{C}); b) \triangleq D(q; b) \tag{2.74}$$

$$= \frac{1}{k} \sum_{i=1}^{N} \sum_{j=1}^{N} p(j|b(i)) \int_{S_i} p(\mathbf{x}) d(\mathbf{x}, \mathbf{y}_j) d\mathbf{x}, \tag{2.75}$$

where $p(\mathbf{x})$ is the k-fold probability density function of the source. The encoding rate is given by

$$R = \frac{1}{k} \log_2 N \quad \text{bits/sample.} \tag{2.76}$$

For a given source, a DMC, a fixed dimension k and fixed rate R, we wish to find the optimal $\mathcal{C}^*$, $\mathcal{P}^*$ and b^* to minimize $D((\mathcal{P}, \mathcal{C}); b)$. According to [14] the importance of the index mapping b is not significant in COVQ design as any changes in $b(i)$ will only result in relabeling of the index of the $\mathcal{P}^*$. To simplify the notation, we assume that the index mapping is done according to $b(i) = i$. Upon rearranging the terms in Equation (2.75) we get

$$D(\mathcal{P}, \mathcal{C}) = \frac{1}{k} \sum_{i=1}^{N} \int_{S_i} p(\mathbf{x}) \left\{ \sum_{j=1}^{N} p(j|i) d(\mathbf{x}, \mathbf{y}_j) \right\} d\mathbf{x}. \tag{2.77}$$

It is easy to see that the problem of distortion minimization specified in Equation (2.77) is equivalent to the VQ design with *modified* distortion measure. For the universal square-error distortion criterion

$$d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||^2, \tag{2.78}$$

it can be shown [25] that given the codebook $\mathcal{C}$ the optimal partition set can be expressed as

$$S_i^* = \left\{ \mathbf{x} : \sum_{j=1}^{N} p(j|i)||\mathbf{x} - \mathbf{y}_j||^2 \leq \sum_{j=1}^{N} p(j|l)||\mathbf{x} - \mathbf{y}_j||^2, \quad \forall l \right\} \quad i \in \mathcal{J}. \tag{2.79}$$

Similarly, the optimal codebook given the partition set is

$$\mathbf{y}_j^* = \frac{\sum_{i=1}^{N} p(j|i) \int_{S_i} \mathbf{x} p(\mathbf{x}) d\mathbf{x}}{\sum_{i=1}^{N} p(j|i) \int_{S_i} p(\mathbf{x}) d\mathbf{x}} \quad j \in \mathcal{J}. \tag{2.80}$$

In essence, Equations (2.79) and (2.80) are the generalized nearest neighbor and centroid conditions with a modified distortion measure. In the cases where the source distribution is unknown, training sequences can be used for the quantizer design and Equations (2.77) and (2.80) can be modified as

$$D(\mathcal{P}, \mathcal{C}) = \frac{1}{n} \sum_{t=1}^{n} \sum_{j=1}^{N} \frac{1}{k} p(j|i(t)) d(\mathbf{x}_t, \mathbf{y}_j) \tag{2.81}$$

and

$$\mathbf{y}_j^* = \frac{\sum_{i=1}^{N} p(j|i) \sum_{l: x_l \in S_i} \mathbf{x}_l / n}{\sum_{i=1}^{N} p(j|i) |S_i| / n}, \tag{2.82}$$

where n is the size of the training sequences and $|S_i|$ denotes the number of training sequences belonging to subspace S_i.

The COVQ design procedure is a straightforward extension of the LBG-VQ design algorithm. The algorithm starts with an initial codebook, $\mathcal{C}^{(0)}$ to find the optimal partition set $\mathcal{P}^{(1)}$ using Equation (2.79). With this newly computed $\mathcal{P}^{(1)}$, it uses Equation (2.80) to update the optimal codebook $\mathcal{C}^{(1)}$. This process continues until the relative decrease in the average distortion is less than a specified threshold and the algorithm converges to a locally optimal solution. In the next subsection we address a technique to initialize the codebook $\mathcal{C}^{(0)}$ for the COVQ design.

2.3.3 Initial Codebook Design

As mentioned in the LBG-VQ design algorithm, the splitting technique is commonly used to obtain the initial codebook $\mathcal{C}^{(0)}$ for the design algorithm. Similarly, for COVQ design, the choice of the initial codebook should be chosen carefully since this initial condition can significantly impact the COVQ results. More elaborate methods should be used instead of the simple splitting algorithm. Farvardin [13] proposed a method based on simulated annealing to initialize the codebook for COVQ design. He showed that the resulting COVQ performance is better than the ones designed with the splitting method or other random methods.

In general, the total end-to-end distortion can be expressed as the sum of the distortion due to quantization ϵ_q^2, distortion due to channel impairment ϵ_c^2, and a cross-term distortion ϵ_*^2. To see this, let us denote $\mathbf{X}$ and $\hat{\mathbf{X}}$ as the input and output sequence of the quantizer respectively, and $\mathbf{Y}$ as the received sequence. Assuming a mean square error distortion, we can write the total distortion per source sample as

$$
\begin{aligned}
D_{tot} &= \frac{1}{k}\mathbf{E}\{||\mathbf{X}-\mathbf{Y}||^2]\} && (2.83)\\
&= \frac{1}{k}\mathbf{E}\{||(\mathbf{X}-\hat{\mathbf{X}}+\hat{\mathbf{X}}-\mathbf{Y}||^2\} && (2.84)\\
&= \underbrace{\frac{1}{k}\mathbf{E}\{||\mathbf{X}-\hat{\mathbf{X}}||^2\}} + \underbrace{\frac{1}{k}\mathbf{E}\{||\hat{\mathbf{X}}-\mathbf{Y}||^2\}} + \underbrace{\frac{2}{k}\mathbf{E}\{(\mathbf{X}-\hat{\mathbf{X}})^t(\hat{\mathbf{X}}-\mathbf{Y})\}} && (2.85)\\
&= \epsilon_q^2+\epsilon_c^2+\epsilon_*^2 && (2.86)
\end{aligned}
$$

In a related work [43], Totty and Clark studied the reconstruction error in waveform transmission for scalar quantizers. The authors showed that if the quantizer is the Lloyd-Max quantizer (optimal), the overall waveform distortion can be expressed as the sum of the quantization distortion and the channel distortion. This suggests that the cross-term in Equation (2.86) can be eliminated if the codewords we choose satisfy the centroid conditions.

Following the notation in Figure (2.3), we can write the average distortion per

sample due to quantization ϵ_q^2 as

$$\epsilon_q^2 = \frac{1}{k}\sum_{i=1}^{N}\int_{S_i} p(\mathbf{x})d(\mathbf{x},\mathbf{y}_i)d\mathbf{x} \tag{2.87}$$

and the average distortion per sample caused by the channel noise ϵ_c^2 as

$$\epsilon_c^2 = \frac{1}{k}\sum_{i=1}^{N}\sum_{j=1}^{N} p(\mathbf{y}_i)p(b(j)|b(i))d(\mathbf{y}_i,\mathbf{y}_j) \tag{2.88}$$

where $p(\mathbf{y}_i)$ is the *a priori* probability of the codeword. Recall the mapping $b : \mathcal{J} \to \mathcal{J}$ is one-to-one. The total overall distortion per sample caused by the vector quantization and the channel is given by

$$\epsilon^2 = \frac{1}{k}\sum_{i=1}^{N}\sum_{j=1}^{N} p(b(j)|b(i)) \int_{S_i} p(\mathbf{x})d(\mathbf{x},\mathbf{y}_j)d\mathbf{x}. \tag{2.89}$$

It can be shown (See Appendix A) that the overall distortion per sample can be simplified as

$$\epsilon^2 = \epsilon_q^2 + \epsilon_c^2 \tag{2.90}$$

if the codeword $\mathbf{y}_i$ is chosen as the centroid of the respective encoding region S_i. It is clear that this result is the direct extension of the work of Totty and Clark to the kth dimension. Equation (2.90) basically states that the overall distortion can be separated into the sum of the quantization distortion and the channel distortion. We should emphasize that this nice property is also based on the assumption of the squared error distortion measure. The above argument suggests that we should first design a LBG-VQ, which uses the splitting algorithm to obtain the initial codebook for VQ, and find the proper index assignment b such that the channel distortion ϵ_c^2 is minimized. To achieve this task, we use a technique called simulated annealing (SA).

Simulated Annealing belongs to a class of randomized stochastic relaxation algorithms. It was originally used by physicists in studying crystal growth in which the material is first heated to the melting point and then the temperature is allowed to

be cooled to form the crystal. This non-linear technique has been very useful to many optimization problems. In the channel coding context, it has been successfully used to find good channel codes [17].

In simulated annealing, the states of the systems are first defined. The next state configurations are generated in probabilistic way to allow local "hill climbing" to evade the local minima. It can be shown that if we allow a sufficiently slow cooling schedule, the simulated algorithm indeed can converge to the *global* minimum in probability. In practice, a more realistic faster cooling schedule is used, such as

$$T_k = \alpha T_{k-1}, \quad 0 < \alpha < 1 \tag{2.91}$$

where T is the effective temperature of the system. For our problem we define $\epsilon_c^2(b)$ as our objective function to be minimized or the energy to be reduced for a hypothetical system. We refer the index assignment $(b(\mathbf{y}_1), b(\mathbf{y}_2), \ldots, b(\mathbf{y}_N))$ as the state of the system. Formally, the simulated algorithm can be outlined as follows [13],

SA Algorithm

- **step 1** Raise the effective temperature to an initial high temperature T_0 and randomly choose an initial state b.

- **step 2** Choose the next state b' randomly and calculate the change in 'energy' $\delta\epsilon_c^2 = \epsilon_c^2(b') - \epsilon_c^2(b)$. If $\delta\epsilon_c^2 < 0$, replace b with b', go to **step 3**, else replace b by b' with probability $\exp\{-\delta\epsilon_c^2/T\}$ and go to **step 3**.

- **step 3** If after $\mathcal{N}$ number of perturbations, no energy drop occurs, go to **step 4**. Otherwise go to **step 2**.

- **step 4** Lower the effective temperature according to Equation (2.91). If the temperature T is below a prescribed freezing temperature T_f or system appears to be stable, stop with b as our final state. Otherwise go to **step 2**.

T_0	10.0
T_f	0.00025
α	0.97
$\mathcal{N}$	200

Table 2.1: Simulated annealing algorithm parameters

The SA algorithm parameters used in this thesis are borrowed directly from [17] [13] and are listed in Table (2.1).

2.3.4 Numerical Results

In this subsection we study the performance of COVQ for memoryless ideal sources. The channel model used here is the first-order ($M = 1$) finite-memory contagion channel described in Section 2.1.2. The source under consideration is an i.i.d. source whose p.d.f. is given by a class of generalized Gaussian distributions described by

$$p(x) = \left[\frac{\alpha\eta(\alpha,\beta)}{2?\,(1/\alpha)}\right]\exp\{-[\eta(\alpha,\beta)|x|]^{\alpha}\}, \quad -\infty < x < \infty \tag{2.92}$$

where

$$\eta(\alpha,\beta) = \beta^{-1}\left[\frac{?\,(3/\alpha)}{?\,(1/\alpha)}\right]^{-1/2} \tag{2.93}$$

with $\alpha > 0$ describing the exponential decay rate and is often called the shaping parameter, β is a positive quantity representing a scale parameter, and $?\,(\cdot)$ is the gamma function, $?\,(x) = \int_0^{\infty} y^{x-1}e^{-y}dy$. The variance of the associated random variable is given by $\sigma_x^2 = \beta^2$. For $\alpha = 2$, we get the Gaussian distribution, while for $\alpha = 1$ we get the Laplacian distribution. For values of α less than one, this generalized Gaussian distribution provides a useful model for broad-tailed processes. We

should also mention that when $\alpha \to \infty$, this distribution reduces to the uniform distribution.

We consider a k-dimensional, rate R bits/sample COVQ which is designed for a given source and first-order Markov channel model using the method described in Section 2.3.2. Again in COVQ design we assume perfect knowledge of the channel parameters (M, δ, and ϵ). For each block of k samples, the output kR bits of the COVQ are sent over the Markov channel directly. In addition, we assume that kR bits are large enough compared to M such that the memory in the channel can be exploited in kR channel uses. Thus by incorporating the characteristics of the channel into the COVQ design, we can exploit the *intra-block* memory of the channel but not the inter-block memory.

First a k-dimensional, rate R vector quantizer (VQ) is designed (for noiseless channel) using LBG-VQ algorithm in which 500,000 training vectors is used. The initial codewords for COVQ design for $\epsilon = 0.005$ is obtained by using simulated annealing technique (described in Section 2.3.3) to permutate the index of the VQ codebook. The final codebook for $\epsilon = 0.005$ is used as the initial codebook for $\epsilon = 0.01$. This process is continued until we reach the desired BER rate. Numerical results for COVQ over Markov channel with $\delta = 10$ and $M = 1$ for the generalized Gaussian sources with shaping parameter $\alpha = 2.0$, $\alpha = 1.0$ and $\alpha = 0.5$ are shown in Tables (2.2)-(2.4). The performance is measured in signal-to-noise ratio (SNR) in decibels (dB) for rates $R = 2, 3,$ and 4 bits/sample and channel BER $\epsilon = 0.0, 0.005, 0.01, 0.05,$ and 0.1. Here we keep the kR value to a manageable number since COVQ design is a full-search algorithm and the computational complexity grows exponentially with kR.

In these tables, the performance results of the COVQ system are compared with another reference system called COVQ-IL using interleaving. In this thesis, we assume that the interleaving and de-interleaving operations are ideal such that the Markov channel is rendered memoryless. For the COVQ-IL system, a COVQ is first designed

for a memoryless system with the same bit error rate as the corresponding Markov channel. This COVQ is then used over the interleaved Markov channel (memoryless). From the tables, we observe that when the channel becomes noisier, the SNR performance for the COVQ does not degrade as sharply as that of the COVQ-IL system. More specifically, in Table (2.2), For $k = 2$, $R = 4$ bits/sample, the gain due to exploiting the intra-block channel memory can be as high as 5 dB.

Also included in the tables are the OPTA (Optimal Performance Theoretical Attainable) values. OPTA can be obtained by evaluating $D(RC)$, where $D(\cdot)$ is the distortion-rate function of the source with squared-error distortion measure and C is the channel capacity in bits per channel use. The rate-distortion function for a $\mathcal{N}(0, \sigma^2)$ source with square error distortion is given by [11],

$$R(D) = \begin{cases} \frac{1}{2}\log_2 \frac{\sigma^2}{D} & 0 \leq D \leq \sigma^2 \\ 0 & D > \sigma^2 \end{cases} \tag{2.94}$$

or expressed in terms of distortion-rate function

$$D(R) = \sigma^2 2^{-2R}, \tag{2.95}$$

where σ^2 is the variance of the source. In general, a closed form expression for $D(R)$ does not exist. However Blahut's algorithm [7] can be used for the numerical computation of $D(R)$.

To obtain the results for the channel optimized scalar quantizer (COSQ), we can simply set $k = 1$. Figure 2.4 illustrates the performance of the COSQ over Markov channel for Generalized Gaussian sources with shaping parameters $\alpha = 2.0$ and $\alpha = 0.5$.

R	k	System	$\epsilon = 0.0$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$
2	1	COVQ-IL	9.29	8.52	7.87	4.86	3.05
		COVQ	9.29	8.64	8.11	5.72	5.92
	2	COVQ-IL	9.51	8.70	8.06	5.44	3.86
		COVQ	9.51	8.97	8.41	7.08	6.62
	3	COVQ-IL	9.87	8.94	8.29	5.70	3.99
		COVQ	9.87	9.15	8.56	7.74	7.10
	∞	OPTA	12.04	11.95	11.86	11.31	10.74
3	1	COVQ-IL	14.60	12.04	10.50	6.47	4.67
		COVQ	14.60	12.39	11.17	9.29	7.47
	2	COVQ-IL	15.21	12.46	11.15	7.36	5.15
		COVQ	15.21	12.81	11.89	10.59	9.42
	3	COVQ-IL	15.66	12.01	11.40	7.67	5.37
		COVQ	15.66	13.55	12.90	11.35	10.05
	∞	OPTA	18.06	17.92	17.80	16.96	16.11
4	1	COVQ-IL	20.17	14.15	12.30	7.81	5.60
		COVQ	20.17	15.67	14.93	11.24	9.13
	2	COVQ-IL	21.06	15.28	13.70	9.06	6.40
		COVQ	21.06	16.70	16.11	13.28	11.52
	∞	OPTA	24.08	23.89	23.73	22.61	21.48

Table 2.2: SNR (dB) performance of COVQ and COVQ-IL over the Markov channel with $\delta = 10$ and $M = 1$; Generalized Gaussian source with shaping parameter $\alpha = 2.0$.

R	k	System	$\epsilon = 0.0$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$
2	1	COVQ-IL	7.55	6.95	6.45	4.01	2.51
		COVQ	7.55	6.99	6.54	4.53	4.52
	2	COVQ-IL	8.83	8.03	7.41	4.75	3.31
		COVQ	8.83	8.09	7.57	6.70	5.86
	3	COVQ-IL	9.48	8.50	7.80	5.13	3.59
		COVQ	9.48	8.71	8.16	7.27	6.32
	∞	OPTA	12.66	12.57	12.49	11.93	11.37
3	1	COVQ-IL	12.64	10.49	9.17	5.21	3.62
		COVQ	12.64	10.50	9.45	8.27	7.17
	2	COVQ-IL	14.25	11.67	10.28	6.60	4.47
		COVQ	14.25	11.88	10.97	10.00	8.64
	3	COVQ-IL	15.16	11.52	10.67	7.08	4.84
		COVQ	15.16	13.01	12.43	10.68	9.48
	∞	OPTA	18.69	18.54	18.42	17.59	16.74
4	1	COVQ-IL	18.08	12.76	11.03	6.82	4.79
		COVQ	18.08	13.57	13.19	10.54	8.61
	2	COVQ-IL	20.09	14.41	12.92	8.33	5.71
		COVQ	20.09	15.38	15.09	12.27	10.68
	∞	OPTA	24.74	24.51	24.35	23.24	22.10

Table 2.3: SNR (dB) performance of COVQ and COVQ-IL over the Markov channel with $\delta = 10$ and $M = 1$; Generalized Gaussian source with shaping parameter $\alpha = 1.0$.

R	k	System	$\epsilon = 0.0$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$
2	1	COVQ-IL	9.29	8.52	7.87	4.86	3.05
		COVQ	9.29	8.64	8.11	5.72	5.92
	2	COVQ-IL	9.51	8.70	8.06	5.44	3.86
		COVQ	9.51	8.97	8.41	7.08	6.62
	3	COVQ-IL	9.87	8.94	8.29	5.70	3.99
		COVQ	9.87	9.15	8.56	7.74	7.10
	∞	OPTA	12.04	11.95	11.86	11.31	10.74
3	1	COVQ-IL	14.60	12.04	10.50	6.47	4.67
		COVQ	14.60	12.39	11.17	9.29	7.47
	2	COVQ-IL	15.21	12.46	11.15	7.36	5.15
		COVQ	15.21	12.81	11.89	10.59	9.42
	3	COVQ-IL	15.66	12.01	11.40	7.67	5.37
		COVQ	15.66	13.55	12.90	11.35	10.05
	∞	OPTA	18.06	17.92	17.80	16.96	16.11
4	1	COVQ-IL	20.17	14.15	12.30	7.81	5.60
		COVQ	20.17	15.67	14.93	11.24	9.13
	2	COVQ-IL	21.06	15.28	13.70	9.06	6.40
		COVQ	21.06	16.70	16.11	13.28	11.52
	∞	OPTA	24.08	23.89	23.73	22.61	21.48

Table 2.4: SNR (dB) performance of COVQ and COVQ-IL over the Markov channel with $\delta = 10$ and $M = 1$; Generalized Gaussian source with shaping parameter $\alpha = 0.5$.

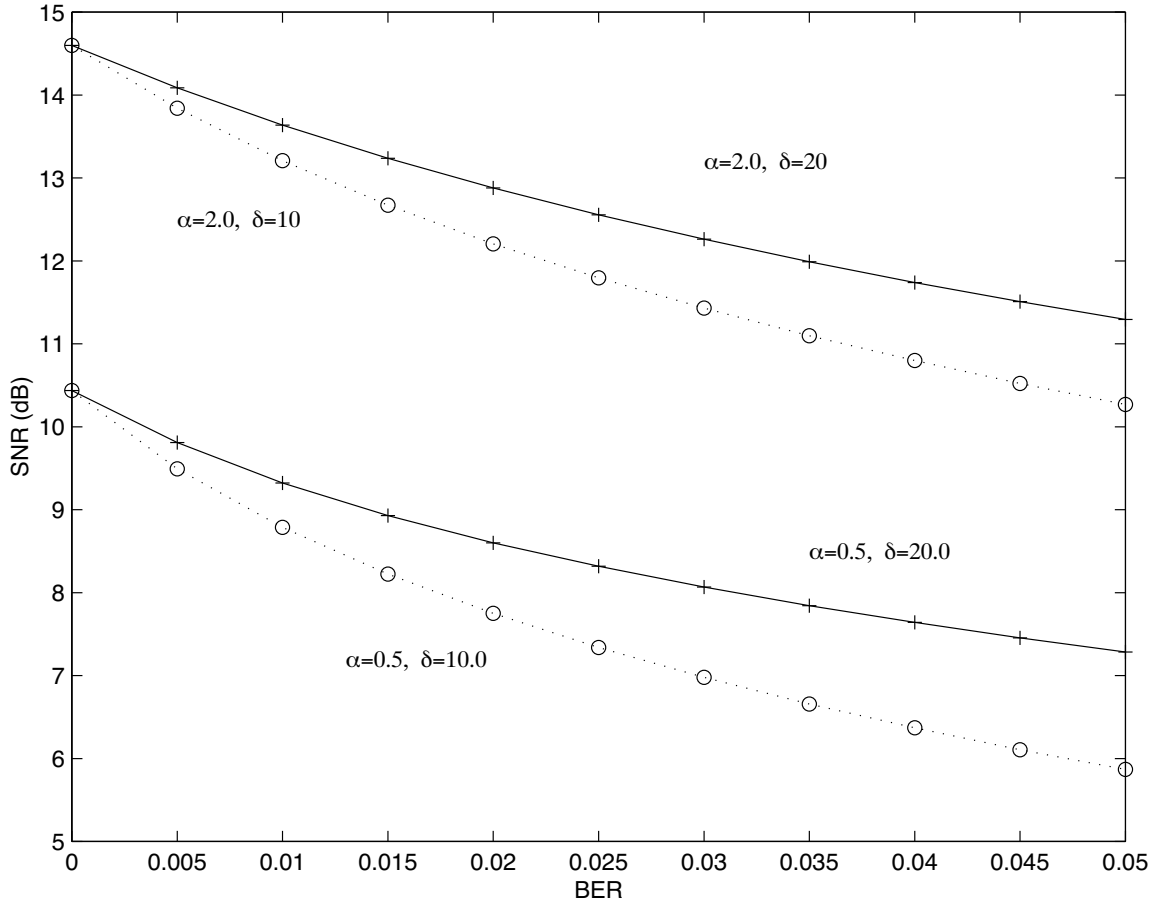


Figure 2.4: COSQ performances for generalized Gaussian sources over the Markov Channel ($R = 3$, $M = 1$).

Chapter 3

Transform Coding and Optimal Bit Allocation

3.1 Transform Coding

We consider blocks of k consecutive samples of a stationary random process and wish to efficiently encode each block with a specified number of bits. The sample vector is denoted by $\mathbf{X} = (X_1, X_2, \ldots, X_k)^t$. In general, these samples have a substantial amount of correlation (in space or time) and separate quantization of each sample would be inefficient. The basic idea of transform coding is to perform a linear transformation on $\mathbf{X}$ and obtain a new vector $\mathbf{Y}$ with the same number of components called transform coefficients. There are two important characteristics associated with the transform coefficients: (i) The transform coefficients are uncorrelated or nearly uncorrelated compared to the original samples; (ii) The information in $\mathbf{Y}$ is much more compactly contained; it is distributed among fewer coefficients than in $\mathbf{X}$. Typically, these coefficients are then separately quantized at different rates with a bank of scalar quantizers. Given a fixed number of bits for describing the transform coefficients, the problem of proper distribution of these bits to the coefficients constitutes the bit allocation problem. In the next subsection, we address this topic in the context of noisy channel transmission.

Alternatively, one can *directly* perform vector quantization on each vector of k samples and the performance of this coding scheme is guaranteed to be optimal by the Shannon's rate distortion theory when the k number of samples to be coded is made infinitely large [40]. In real life applications such as image or speech coding, the true distribution of the source is unknown. Training sequences are then used to obtain the approximate underlying probability distribution function of the data samples. However, due to the high computational complexity of the full-search VQ, the number of samples k to be coded is usually limited to a manageable number.

In practice, the transform coding method is often used, particularly for efficient image coding [10]. As in the case of the Fourier Transform (FT), the linear transform converts the data samples from the time (space) domain to the frequency domain. There is also a subjective reason why transform coding is important in many speech and image applications. By nature, the human auditory and visual systems seem to operate in the frequency domain; in particular they are insensitive to the low frequency transformed components. Typically, low frequency coefficients carry more energy (and hence information) than the high frequency ones; in general, they follow different probability distribution functions. Distinct scalar quantizers can be employed to quantize each coefficient with more bits given to low frequency coefficients and less bits to the high frequency coefficients. If some coefficients carry a significantly small amount of information, they are often not coded at all. In this way, the redundancies due to the human visual and auditory system are exploited. A typical transform coding system is illustrated in Figure (3.1), in which the reconstructed approximation to the original vector, $\hat{\mathbf{X}}$ is obtained by performing the inverse transform on the quantized transform vector $\hat{\mathbf{Y}}$,

$$\mathbf{Y} = T(\mathbf{X}), \quad \hat{\mathbf{Y}} = Q(\mathbf{Y}), \quad \hat{\mathbf{X}} = T^{-1}(\hat{\mathbf{Y}}). \tag{3.1}$$

Again, we assess the overall performance of the coding scheme using the universal

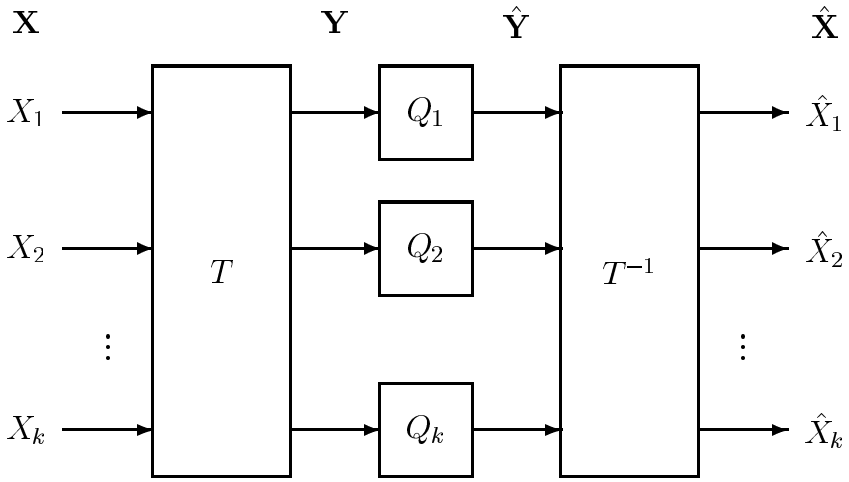


Figure 3.1: A transform coding system

squared error distortion

$$D = \sum_{i=1}^{k} \mathbf{E}[|X_i - \hat{X}_i|^2] = \mathbf{E}[||\mathbf{X} - \hat{\mathbf{X}}||^2]. \tag{3.2}$$

Obviously, the performance of our transform coding system depends on the choice of the transform matrix T. We will consider a class of orthogonal transforms which have the distance preserving property. A real-valued, $k \times k$ *orthogonal* transform matrix T satisfies the condition

$$T^t = T^{-1} \tag{3.3}$$

or $T^* = T^{-1}$ for complex-valued matrices, where T^* denotes the conjugate transpose of T. The above matrix is also called a *unitary matrix*. Unitary transformations enjoy a nice geometric interpretation and can be thought as simply the rotation of the vector in the original vector space. Alternatively, they can be thought as the rotation of basis coordinates and the components of the new vector are the projection of the

original vector on the new basis. This also implies that a unitary transformation preserves the length of the vector in the multi-dimensional vector space, or the signal energy. The distance preserving property can be easily shown with Equation (3.3) [18]. Another important characteristic of the unitary transforms is that they do not alter the determinant of the autocorrelation matrix of the input vector.

In theory, one important orthogonal transform is the Karhunen-Loeve transform (KLT). In essence, it can achieve perfect de-correlation of the input vector $\mathbf{X}$ such that the output of the transform $\mathbf{Y} = T\mathbf{X}$ have pairwise uncorrelated components. To see this, let $R_{XX} = E[\mathbf{X}\mathbf{X}^t]$ denote the autocorrelation matrix of the input vector $\mathbf{X}$ and $\mathbf{u}_i$ denote the eigenvectors of R_{XX} and λ_i be the corresponding eigenvalues. Since any autocorrelation matrix is symmetric and non-negative definite, there are k orthogonal eigenvectors and the corresponding eigenvalues are real and nonnegative. We assume without loss of generality that

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k \geq 0, \tag{3.4}$$

then the KLT is defined as $T = U^t$ where $U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$. In other words, the column vectors of U are the eigenvectors of the autocorrelation matrix R_{XX}. Then the autocorrelation matrix of the output vector $\mathbf{Y}$ is given by

$$R_{YY} = \mathbf{E}[\mathbf{Y}\mathbf{Y}^t] = \mathbf{E}[U^t\mathbf{X}\mathbf{X}^tU] = U^tR_{XX}U = diag(\lambda_i). \tag{3.5}$$

It follows that the variances of the transform coefficients are the eigenvalues of the autocorrelation matrix R_{XX}.

Despite its favorable theoretical properties, KLT is not used in practice. This is mainly due to the fact that the basis functions depend on the autocorrelation matrix of the input vector and there are no fast computational algorithms for their implementation. The class of orthogonal transformation includes the discrete Fourier transform

(DFT), the discrete cosine transform (DCT), the Wash-Hadamard transform and others. Among them DCT has been found to be the most effective, with a performance close to that of KLT; it is hence widely used in image and video compression.

For a 1-D time (space) signal $f(x)$, the 1-D discrete cosine transform $C(u)$ is defined by

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \tag{3.6}$$

for $u = 0, 1, 2, \ldots, N-1$. The inverse DCT is defined by

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \tag{3.7}$$

for $x = 0, 1, 2, \ldots, N-1$, where $\alpha(\cdot)$ is given by

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u = 1, 2, \ldots, N-1. \end{cases} \tag{3.8}$$

The corresponding 2-D discrete cosine transform is given by

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2x+1)v\pi}{2N}\right] \tag{3.9}$$

for $u, v = 0, 1, 2, \ldots, N-1$, and

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v) C(u,v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2x+1)v\pi}{2N}\right] \tag{3.10}$$

for $x, y = 0, 1, 2, \ldots, N-1$.

The 2-D DCT is an invertible and *separable* orthogonal transform. By separable, we mean that the 2-D complete orthonormal discrete basis function can be written as the product of two 1-D complete orthonormal basis vectors.

The DCT is the most popular transformation scheme in transform coding since it has a fixed (data-independent) basis function and can be efficiently computed

via FFT algorithms. Like the KLT, it has excellent energy compaction properties for correlated data and can result in highly uncorrelated transform coefficients. DCT has also been adopted as an international standard by ITU for JPEG image compression [46].

The 2-D DCT operates on sub-blocks of an image to trade smaller number of arithmetic operations for undesirable blocking artifacts. To see this, the number of arithmetic operations to compute the $n \times n$-point transform is $\mathcal{O}(n^2 \log_2 n^2)$. For an image of size $M \times M$, with DCT operating on image blocks of size $N \times N$, it can be shown that we can reduce the amount of arithmetic operation by a factor of $\log_2 M / \log_2 N$.

In addition to the above-mentioned advantages of DCT, 2-D DCT also has some nice statistical characteristics. We name the $(0,0)^{th}$ transform coefficient as the DC coefficient and the rest of the coefficients as the AC coefficients. Based on experimental evaluations for 2-D images [37], it has been shown that the DC coefficient roughly follows a Gaussian distribution and that AC coefficients are roughly governed by a Laplacian distribution; furthermore, these statistical properties are image independent. Consequently we can incorporate these statistical models into our quantizer design for each transform coefficient. It is also well known that inter-block DC coefficients are highly correlated [46]. In JPEG, predictive coding methods such as differential pulse code modulation (DPCM) are used to encode the DC coefficients more efficiently.

3.2 Optimal Bit Allocation

The optimal bit allocation problem is closely related to the optimal resource allocation question raised in our everyday life. That is, given a finite amount of resources (capital, time, etc.), we strive for an optimal solution to distribute these resources in order to maximize the benefit or gain.

For the transform coding system described in the previous subsection, the average number of bits allocated to each transform coefficient determines the total number of bits, or the resource we have since this parameter also defines the transmission bandwidth requirements. As a rule of thumb, the transform coefficients that carry the most information (energy) should be allocated with more bits (finer quantization); and those transform coefficients considered less important should be given fewer bits or none at all, while the total number of allocated bits remain fixed. Typically, the outputs of the quantizers are coded and the indices of the codewords are sent over a noisy channel. Our objective is to find an optimal bit assignment scheme under certain imposed constraints such that the overall end-to-end distortion due to this assignment is minimized.

We decompose this optimization problem for our image transmission system into two parts: (i) To design optimal zero-memory quantizers by taking into consideration the channel; (ii) To calculate the optimal bit allocation matrix by taking into consideration *both* the source and the channel characteristics. The first task can be achieved with the design of the channel optimized quantizers described in Chapter 2. We now concentrate on the second task and state our bit allocation problem as follows.

For $N \times N$ image blocks transformed via DCT, we assume the DCT coefficients, $\{X_{mn}\}, m, n = 0, 1, \ldots, N-1$ are governed by a zero mean, stationary stochastic process. Let r_{av} be the required average number of bits per $N \times N$ image block ($B = r_{av}N^2$ is the total number of bits per $N \times N$ image block). Denote $D(r_{mn})$ as the distortion incurred in quantizing the random variable X_{mn} with r_{mn} bits. The optimal bit allocation problem becomes to find the bit allocation matrix $\mathbf{r}$, such that the total distortion $\sum\sum D(r_{mn})$ is minimized subject to the constraints: (i) $\sum\sum r_{mn} \leq B$, and (ii) $0 \leq r_{mn} \leq r_{max}$, where r_{max} is the maximum number of bits allowed to be allocated to each DCT coefficient.

Optimal bit allocation occurs frequently in image and video applications. In many practical cases, by invoking the high bit rate resolution approximation, *assuming noiseless channel conditions* and no constraints, we can extend the standard 1-D result from [18] to our 2-D problem as

$$r_{mn} = \bar{r} + \frac{1}{2}\log_2 \frac{\sigma_{mn}^2}{\rho^2}, \tag{3.11}$$

where

$$\bar{r} = \frac{B}{K}, \tag{3.12}$$

and

$$\rho^2 = (\prod_{m,n} \sigma_{mn}^2)^{\frac{1}{K}}, \tag{3.13}$$

where $m, n = 0, 1, \ldots, N-1$, and $K = N^2$ is the total number of transform coefficients in each $N \times N$ block, and σ_{mn}^2 denotes the variance of the mn^{th} coefficient in an $N \times N$ coefficient block. Note that the number of bits allocated to each coefficient is proportional to the variance of that coefficient, or the energy that coefficient carries. Despite the fact that this bit allocation scheme works well even under low bit resolution cases, it has two major drawbacks: (i) the assigned number of bits to any particular coefficient could be negative; (ii) the resulting bit could be a non-integer number. The first difficulty could be dealt with by suppressing the negative number of bits to zero. As for the second problem, if we perform round-off operation on the non-integer bits, we will not be able to meet the constraints on the total number of bits per image block (an integer). We now introduce an integer programming approach called the steepest descent algorithm to solve our bit allocation problem for the case of *noisy channels with memory*.

The steepest descent algorithm belongs to a class of greedy algorithms. The basic idea of this technique for our bit allocation problem is to start with the total number

of available bits B for each $N \times N$ image transform coefficient block. We give away one bit at a time to one of the coefficients such that the *overall distortion is minimized* in each step till we run out of bits. Thus for an $N \times N$ image block, only finite BN^2 steps are needed to calculate our 'optimal' bit allocation matrix. It is called steepest descent since we maximize our benefit or minimize the distortion as much as possible in each iteration. In a greedy sense this algorithm opts to minimize the distortion while neglecting the global effect of such choice. In general such algorithms result in sub-optimal solutions; however, as shown later, under certain conditions, steepest descent algorithms will indeed result in an *optimal* solution for our problem. Another obvious but less practical approach to the same problem is by exhaustive search. We will not resort to such computation intensive methods in this thesis.

Steepest descent algorithms have been successfully applied in [44], [45]. Our scheme is similar in spirit to the work in [45] but differ in the following aspects: (i) A Markov channel model is considered instead of a memoryless BSC; (ii) DC and AC transform coefficients are modeled separately as Gaussian and Laplacian distribution respectively; (iii) One motivation of our study is to examine the loss of optimality due to fixed bit allocation methods (see Chapter 4).

We recast our optimal bit allocation problem as follows. Assume the source is represented via a stationary stochastic process. Let X_{mn} denote the mn^{th} pixel of each $N \times N$ image block and $\hat{X}_{mn}$ be the reconstructed pixel. The objective is to minimize the end-to-end average distortion mean square error sense. That is to minimize

$$D = \frac{1}{N^2}\mathbf{E}\left[\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}(X_{mn} - \hat{X}_{mn})^2\right] \tag{3.14}$$

under the constraints,

$$\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} r_{mn} \leq B \tag{3.15}$$

and

$$0 \leq r_{mn} \leq r_{max} \tag{3.16}$$

where r_{mn} denotes the mn^{th} entry of an $N \times N$ bit allocation matrix $\mathbf{r}$. B is the total number of bits available for $\mathbf{r}$. We also impose the maximum number of bits r_{max} which is allowed to be allocated to each transform coefficient. In this thesis, we choose $r_{max} = 8$ bits.

Since 2-D DCT is a separable orthonormal transformation, it can be shown [24] that Equation (3.14) is equivalent to

$$D = \frac{1}{N^2}\mathbf{E}\Big[\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}(Y_{mn} - \hat{Y}_{mn})^2\Big] \tag{3.17}$$

where Y_{mn} and $\hat{Y}_{mn}$ denote the mn^{th} entry of a $N \times N$ transform coefficient block.

In Equation (3.17), since each summation term is a non-negative quantity, in order to minimize D, it suffices to minimize each component of D. For a given bit allocation matrix $\mathbf{r}$, let $d^*_{mn}(r_{mn})$ denote the minimum distortion when the mn^{th} transform coefficient is quantized with r_{mn} bits. Then the *minimum average distortion* for a given bit allocation matrix $\mathbf{r}$, denoted by $D^*(\mathbf{r})$, is

$$D^*(\mathbf{r}) = \frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} d^*_{mn}(r_{mn}). \tag{3.18}$$

To solve the optimal bit allocation problem, we wish to find the bit allocation matrix $\mathbf{r}^*$ such that $D^*(\mathbf{r}^*) \leq D^*(\mathbf{r}) \ \ \forall\, \mathbf{r}$, and $\mathbf{r}$ satisfies the constraints provided in Equations (3.15) and (3.16).

We assume that the DC coefficient follows the Gaussian distribution and all AC coefficients follow the Laplacian distribution, then Equation (3.18) can be written as

$$D^*(\mathbf{r}) = \frac{1}{N^2}\Big\{\sigma^2_{00} d^*_G(r_{00}, 1) + \sum_{m=1}^{N-1}\sum_{n=1}^{N-1}\sigma^2_{mn} d^*_L(r_{mn}, 1)\Big\} \tag{3.19}$$

where $d_G^*(r_{00}, 1)$ denotes the minimum distortion incurred when using r_{00} bits to represent a sample governed by a unit variance Gaussian distribution. Similarly, $d_L^*(r_{mn}, 1)$ denotes the minimum distortion incurred when using r_{mn} bits to represent a sample generated by a unit variance Laplacian distribution. Again, $0 \leq r_{mn} \leq r_{max}$ for $m, n = 0, \ldots, N-1$. The steepest descent algorithm is as follows [45].

Steepest Descent Algorithm:

- **step 1** Perform $N \times N$ forward DCT and estimate the variance $\sigma_{mn}^2; m, n = 0, 1, \ldots, N-1$.
- **step 2** Compute $d_G^*(r_{00}, 1)$ and $d_L^*(r_{mn}, 1)$ for $1 \leq r_{00}, r_{mn} \leq r_{max}; m, n = 1, 2, \ldots, N-1$.
- **step 3** Set $k = 0$, $r_{mn} = 0$, $m, n = 0, \ldots, N-1$.
- **step 4** Set $k = k + 1$, find the index $(m_k n_k)$ which satisfies

$$d_{m_k n_k}^*(r_{m_k n_k}) - d_{m_k n_k}^*(r_{m_k n_k} + 1) = \max_{0 \leq m,n \leq N-1} \{d_{mn}^*(r_{mn}) - d_{mn}^*(r_{mn} + 1)\}.$$

- **step 5** Set $r_{m_k n_k} = r_{m_k n_k} + 1$, if $k < B$, go to step 4, otherwise stop.

To show that the above algorithm indeed minimizes the objective function in Equation (3.19) under the constraints given in Equations (3.15) and (3.16), we state the following theorem (see [45] for proof).

Theorem: If the minimum distortion functions $d_{mn}^*(r_{mn})$ are *convex* and *strictly non-increasing*, then the bit allocation matrix $\mathbf{r}^*$ obtained with the steepest descent algorithm will minimize the distortion D in Equation (3.18) under the constraints given in Equations (3.15) and (3.16).

Here by convexity, we mean that $d(\cdot)$ satisfies

$$d(\lambda x + (1-\lambda)y) \leq \lambda d(x) + (1-\lambda)d(y) \quad \forall \lambda \in [0,1]. \tag{3.20}$$

Letting $x = r-1$, $y = r+1$, and $\lambda = 1/2$ yields

$$d_{mn}(r-1) - d_{mn}(r) \geq d_{mn}(r) - d_{mn}(r+1), \tag{3.21}$$

a property needed for the above theorem.

It should be emphasized that the minimum distortion $d^*_{mn}(r)$ is obtained via the channel optimized quantizer design with respect to the source distribution and the channel characteristics for the given quantizer bit rate r. Tables (3.1)-(3.6) list the minimum distortion for unit variance Gaussian and Laplacian sources with various bit rates and Markov channel conditions (ϵ, δ). The corresponding distortion versus rate curves for $\epsilon = 0.0005$ and $\epsilon = 0.1$ are plotted in Figure 3.2. We observe that $d^*_G(r_{00}, 1)$ and $d^*_L(r_{mn}, 1)$ are indeed convex and strictly non-increasing. Similar observations are made for other BER rates.

We apply this integer programming approach to the popular image Lena (512×512), obtained from the USC image data base, for our Markov channel model with $M = 1$. 8×8 image blocks are used in the DCT according to the JPEG standard. The resulting optimal bit allocation matrices are presented in Tables (3.7)-(3.9) for $\epsilon = 0.0, 0.005, 0.01,$ and 0.1, and channel correlations $\delta = 0.0, 5.0,$ and 10.0 for three different bit rates, $B = 76, 58,$ and 24 bits. Several interesting observations can be made by close examination of the bit allocation matrices. In all cases, most of the bits are concentrated on the low frequency coefficients. This reflects the fact that low frequency DCT coefficients carry more energy (variance) compared to the high frequency ones. For fixed δ, when the channel gets noisier, the first few coefficients receive the highest protection with the maximum number of allowable bits (8 bits). For fixed ϵ, when the channel gets more bursty (high δ), bits are slightly spread out to the next few DCT coefficients. Similar observations are made for other images but

we will not exhaustively list the tables.

Rate (bits)	$\epsilon = 0.0$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$
1	0.500234	0.510382	0.520032	0.595395	0.680555
2	0.175706	0.201662	0.226438	0.396853	0.561211
3	0.054390	0.089351	0.120960	0.320393	0.417480
4	0.015532	0.052444	0.083642	0.208550	0.331667
5	0.004147	0.033015	0.048400	0.147958	0.281028
6	0.001088	0.021719	0.034078	0.136509	0.221164
7	0.000281	0.013271	0.021244	0.096742	0.186213
8	0.000068	0.009176	0.015243	0.064382	0.156570

Table 3.1: Minimum distortion incurred for the Laplacian source over the Markov channel with $\delta = 0.0$ and $M = 1$.

Rate (bits)	$\epsilon = 0.0$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$
1	0.500234	0.510382	0.520032	0.595395	0.680555
2	0.175706	0.190567	0.204431	0.296277	0.383257
3	0.054390	0.069701	0.083717	0.172151	0.254452
4	0.015532	0.030663	0.043825	0.123954	0.180998
5	0.004147	0.021330	0.033032	0.082154	0.132266
6	0.001088	0.014987	0.023122	0.068755	0.103035
7	0.000281	0.010221	0.015689	0.044552	0.074372
8	0.000068	0.005906	0.009802	0.033140	0.056960

Table 3.2: Minimum distortion incurred for the Laplacian source over the Markov channel with $\delta = 5.0$ and $M = 1$.

Rate (bits)	$\epsilon = 0.0$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$
1	0.500234	0.510382	0.520032	0.595395	0.680555
2	0.175706	0.188207	0.200003	0.277932	0.352390
3	0.054390	0.064069	0.073249	0.133671	0.192102
4	0.015532	0.028096	0.039745	0.087388	0.134457
5	0.004147	0.015967	0.022136	0.051928	0.084372
6	0.001088	0.011092	0.017427	0.039196	0.063550
7	0.000281	0.007457	0.009938	0.031986	0.049095
8	0.000068	0.004114	0.006415	0.018539	0.031833

Table 3.3: Minimum distortion incurred for the Laplacian source over the Markov channel with $\delta = 10.0$ and $M = 1$.

Rate (bits)	$\epsilon = 0.0$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$
1	0.363666	0.376471	0.389076	0.484822	0.593322
2	0.117675	0.140689	0.163187	0.326655	0.495535
3	0.034686	0.062856	0.089716	0.211211	0.351710
4	0.009593	0.038767	0.055149	0.156847	0.259459
5	0.002563	0.025799	0.038930	0.110204	0.210865
6	0.000677	0.014622	0.021715	0.083357	0.171253
7	0.000174	0.007823	0.012859	0.059256	0.137999
8	0.000044	0.005948	0.009705	0.052581	0.117087

Table 3.4: Minimum distortion incurred for the Gaussian source over the Markov channel with $\delta = 0.0$ and $M = 1$.

Rate (bits)	$\epsilon = 0.0$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$
1	0.363666	0.376471	0.389076	0.484822	0.593322
2	0.117675	0.128323	0.138668	0.213089	0.290937
3	0.034686	0.045160	0.055329	0.126484	0.199011
4	0.009593	0.021902	0.032826	0.099871	0.139869
5	0.002563	0.012984	0.021026	0.059509	0.096052
6	0.000677	0.011936	0.018546	0.042729	0.073767
7	0.000174	0.006136	0.009822	0.033719	0.055171
8	0.000044	0.004471	0.007282	0.024245	0.040977

Table 3.5: Minimum distortion incurred for the Gaussian source over the Markov channel with $\delta = 5.0$ and $M = 1$.

Rate (bits)	$\epsilon = 0.0$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.1$
1	0.363666	0.376471	0.389076	0.484822	0.593322
2	0.117675	0.126027	0.134137	0.193063	0.255589
3	0.034686	0.041301	0.047772	0.094026	0.142325
4	0.009593	0.016589	0.023216	0.066068	0.110033
5	0.002563	0.011260	0.017477	0.046696	0.074954
6	0.000677	0.006197	0.009561	0.027250	0.042603
7	0.000174	0.003730	0.005866	0.017462	0.029549
8	0.000044	0.002582	0.004393	0.012069	0.020336

Table 3.6: Minimum distortion incurred for the Gaussian source over the Markov channel with $\delta = 10.0$ and $M = 1$.

;

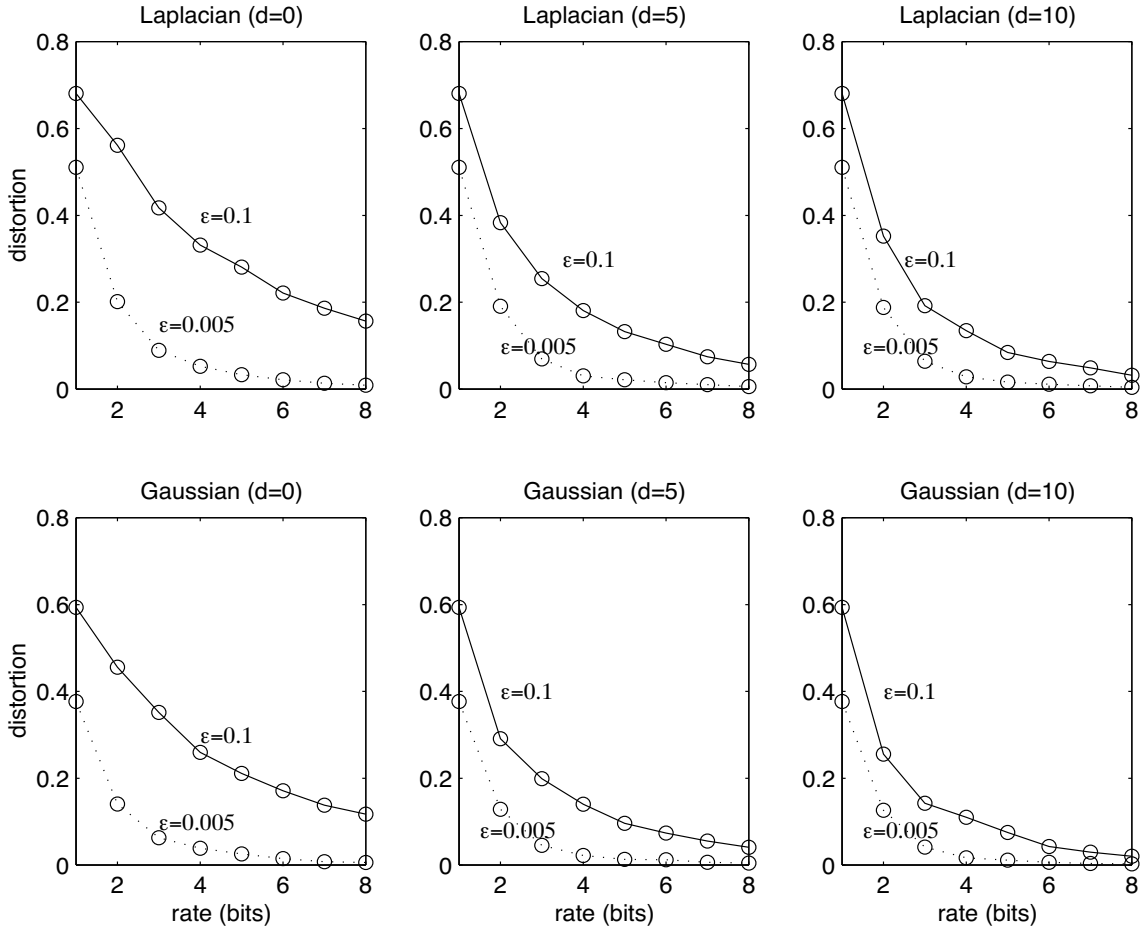


Figure 3.2: Distortion versus rate curves for generalized Gaussian sources.

	$\delta = 0.0$								$\delta = 5.0$								$\delta = 10.0$							
$\epsilon = 0.0$	7	6	4	4	3	2	1	0	7	6	4	4	3	2	1	0	7	6	4	4	3	2	1	0
	5	4	4	3	2	2	0	0	5	4	4	3	2	2	0	0	5	4	4	3	2	2	0	0
	3	3	3	3	2	1	0	0	3	3	3	3	2	1	0	0	3	3	3	3	2	1	0	0
	2	2	2	2	2	0	0	0	2	2	2	2	2	0	0	0	2	2	2	2	2	0	0	0
	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\epsilon = 0.005$	8	8	5	3	3	2	0	0	8	8	4	3	3	2	1	0	8	8	4	3	3	2	1	0
	6	5	3	3	2	1	0	0	5	4	4	3	2	1	0	0	5	4	4	3	2	1	0	0
	3	3	3	2	2	1	0	0	3	3	3	3	2	1	0	0	3	3	3	3	2	1	0	0
	2	2	2	2	1	0	0	0	2	2	2	2	1	0	0	0	2	2	2	2	1	0	0	0
	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\epsilon = 0.01$	8	8	5	3	2	2	0	0	8	8	4	3	3	2	0	0	8	8	5	3	3	2	0	0
	7	5	3	3	2	1	0	0	6	4	4	3	2	1	0	0	5	5	3	3	2	1	0	0
	3	3	3	2	2	1	0	0	3	3	3	3	2	1	0	0	3	3	3	3	2	1	0	0
	2	2	2	2	1	0	0	0	2	2	2	2	1	0	0	0	2	2	2	2	1	0	0	0
	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\epsilon = 0.1$	8	8	8	4	2	1	0	0	8	8	7	4	2	1	0	0	8	8	8	3	3	2	0	0
	8	8	6	4	1	0	0	0	8	7	5	3	2	0	0	0	8	6	5	3	2	0	0	0
	4	4	4	1	1	0	0	0	4	4	3	2	1	0	0	0	3	3	3	3	2	0	0	0
	1	1	1	1	0	0	0	0	2	2	2	1	0	0	0	0	2	2	2	2	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.7: Optimal bit allocation matrices for Lena at B=76 bits, or 1.19 bpp.

	$\delta = 0.0$	$\delta = 5.0$	$\delta = 10.0$
$\epsilon = 0.0$	7 5 4 3 2 1 0 0	7 5 4 3 2 1 0 0	7 5 4 3 2 1 0 0
	4 4 3 3 2 1 0 0	4 4 3 3 2 1 0 0	4 4 3 3 2 1 0 0
	3 3 3 2 1 0 0 0	3 3 3 2 1 0 0 0	3 3 3 2 1 0 0 0
	2 2 2 1 0 0 0 0	2 2 2 1 0 0 0 0	2 2 2 1 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$\epsilon = 0.005$	8 7 4 3 2 1 0 0	8 6 4 3 2 1 0 0	8 5 4 3 2 1 0 0
	5 4 3 2 1 0 0 0	4 4 3 2 1 0 0 0	4 4 3 3 2 0 0 0
	3 3 2 2 1 0 0 0	3 3 3 2 1 0 0 0	3 3 3 2 1 0 0 0
	2 2 2 1 0 0 0 0	2 2 2 1 0 0 0 0	2 2 2 1 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$\epsilon = 0.01$	8 8 5 3 2 1 0 0	8 7 4 3 2 1 0 0	8 5 4 3 2 1 0 0
	5 5 3 2 1 0 0 0	4 4 3 2 1 0 0 0	5 4 3 3 1 0 0 0
	3 3 2 2 1 0 0 0	3 3 3 2 1 0 0 0	3 3 3 2 1 0 0 0
	1 1 1 1 0 0 0 0	2 2 2 1 0 0 0 0	2 2 2 1 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$\epsilon = 0.1$	8 8 8 3 1 0 0 0	8 8 7 3 2 0 0 0	8 8 5 3 2 0 0 0
	8 7 4 1 0 0 0 0	8 5 4 2 0 0 0 0	6 5 3 2 0 0 0 0
	3 3 2 1 0 0 0 0	3 3 3 2 0 0 0 0	3 3 3 2 0 0 0 0
	0 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 2 2 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Table 3.8: Optimal bit allocation matrices for Lena at B=58 bits, or 0.9 bpp.

	$\delta = 0.0$	$\delta = 5.0$	$\delta = 10.0$
$\epsilon = 0.0$	6 4 3 1 0 0 0 0	6 4 3 1 0 0 0 0	6 4 3 1 0 0 0 0
	3 2 2 0 0 0 0 0	3 2 2 0 0 0 0 0	3 2 2 0 0 0 0 0
	1 1 1 0 0 0 0 0	1 1 1 0 0 0 0 0	1 1 1 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$\epsilon = 0.005$	8 4 2 1 0 0 0 0	5 4 3 1 0 0 0 0	7 4 3 1 0 0 0 0
	3 2 2 0 0 0 0 0	3 3 2 0 0 0 0 0	3 2 2 0 0 0 0 0
	1 1 0 0 0 0 0 0	1 1 1 0 0 0 0 0	1 1 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$\epsilon = 0.01$	8 5 2 1 0 0 0 0	8 4 2 1 0 0 0 0	7 4 3 1 0 0 0 0
	3 2 1 0 0 0 0 0	3 2 2 0 0 0 0 0	3 2 2 0 0 0 0 0
	1 1 0 0 0 0 0 0	1 1 0 0 0 0 0 0	1 1 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
$\epsilon = 0.1$	8 8 2 0 0 0 0 0	8 7 3 0 0 0 0 0	8 5 3 0 0 0 0 0
	4 1 1 0 0 0 0 0	4 2 0 0 0 0 0 0	3 3 2 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Table 3.9: Optimal bit allocation matrices for Lena at B=24 bits, or 0.375 bpp.

Chapter 4

Experimental Results

4.1 An Image Transmission System

In this section, we propose a DCT-based combined source-channel coding system to transmit grey-level images over the binary channel with additive Markov noise described in Section 2.1.2. It consists of a channel optimized quantization scheme that exploits the characteristics of the correlated noise in the quantizer design. A block diagram of our proposed system is illustrated in Figure 4.1.

A grey-level (8 bpp or 256 levels) image is first subdivided into 8×8 blocks and transformed via Forward Discrete Cosine Transform (FDCT) according to the JPEG standard. As mentioned in Section 3.1, the purpose of the DCT is to ensure that the transform coefficients are highly uncorrelated and to concentrate most of the energy over a few low frequency coefficients. After proper normalization, higher frequency DCT coefficients are zonally masked out since they are relatively insensitive to the human visual system [42]. *Fixed* bit allocation tables are used for each 8×8 image coefficient block. The inter-block DC coefficients are arranged in a zigzag sequence (See Figure 4.2) and are subsequently quantized via a bank of channel optimized scalar

quantizers (COSQ). Since the DC coefficient (the coefficient with zero frequency) contains most of the energy in each image block, it is quantized with an 8-bit rate quantizer; as for the AC coefficients, they are quantized at rates that correspond to their level of activities. After quantization, the indices of the codebook are coded via a Natural Binary Code (NBC) and sent over the binary channel. At the receiver end, they are decoded and the reconstructed image is obtained through the Inverse Discrete Cosine Transform (IDCT). A bank of COSQs are designed off line using the method described in Section 2.3.2 and we have assumed *a priori* knowledge of the channel conditions and the statistics of the quantizer input. The source distributions are assumed to be Gaussian for the DC coefficients and Laplacian for all the AC coefficients. In Section 4.3, we study the performance of the quantizer under mismatched channel conditions.

The rationale to arrange the inter-block DC coefficients in a zigzag sequence is as follows. It is well known that the inter-block DC coefficients are highly correlated ($\rho = 0.977$ for Lena and $\rho = 0.993$ for Baboon [22]). By arranging them in a zigzag fashion, it becomes desirable to exploit the memory within these transform coefficients by using COVQ. Unfortunately, such an approach is not feasible for a DCT-based coding system since the computational complexity of a full-search VQ grows exponentially with k and R. By taking $k = 2$ and $R = 8$, the size of the codebook can quickly become 2^{16}. In [34] [35], Phamdo et al. designed sub-optimal Multi-Stage COVQ and Tree-Structured COVQ (TS-COVQ) for the BSC to reduce the quantization complexity; however it is not trivial to apply these techniques directly to our discrete channel model with memory. Nevertheless, there is one additional advantage for arranging the inter-block DC (or even AC) coefficients in a zigzag

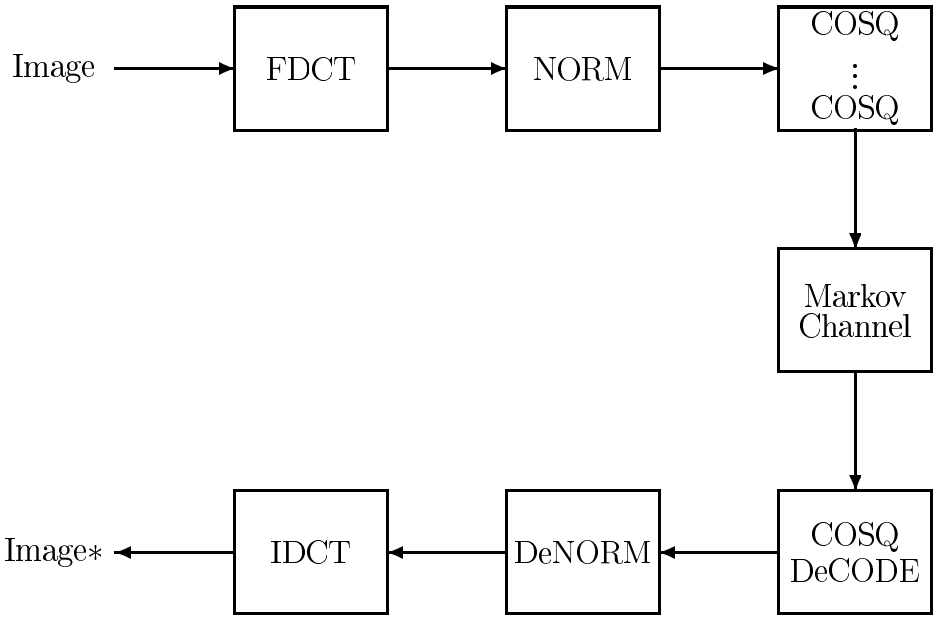


Figure 4.1: An image transmission system

order. In many image transmission applications, by sending the transform coefficients in this fashion, one can minimize the *information* transmission delay. The user can have an approximate version of the image without having received all the transform coefficients and this is the basic idea behind progressive image coding. We should also point out that we cannot resort to the variable length source coding method such as the run-length and Huffman coding techniques used in JPEG [46]; this is because the noisy channel can cause error propagation in the decoder.

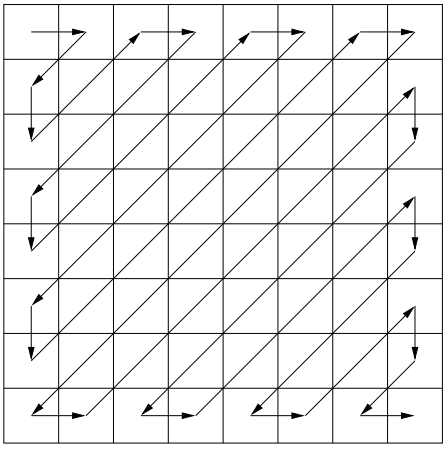


Figure 4.2: Zigzag sequence

4.2 Results with Fixed Bit Allocation Tables

Experimental results for this proposed system indicate that large improvement over usual tandem schemes, which employ interleaving and are designed for the noiseless channel, can be achieved. We performed the experiments on several images. To avoid exhaustive listings, we only present the numerical results for Lena (512×512). In Tables (4.2)-(4.4), the average PSNR values of the reconstructed Lena are displayed for various values of the channel correlation δ, BER ϵ, and overall operational rate in bits per pixel (bpp). The objective measure PSNR (in dB) is defined as

$$PSNR = 10 \log_{10} \frac{255^2}{\mathbf{E}\{(X_{ij} - \hat{X}_{ij})^2\}} \tag{4.1}$$

where X_{ij} and $\hat{X}_{ij}$ are, respectively, the transmitted and reconstructed ij^{th} image pixel.

All the numerical results were obtained by averaging over 25 experiments and it was observed that the PSNR values do not vary too much from experiment to experiment. Three overall operational rates were used and the fixed bit allocation

$B = 76$								$B = 58$								$B = 24$							
8	7	6	4	3	0	0	0	8	7	6	4	0	0	0	0	8	8	0	0	0	0	0	0
7	6	5	4	0	0	0	0	7	6	5	0	0	0	0	0	8	0	0	0	0	0	0	0
6	5	4	0	0	0	0	0	6	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.1: Global fixed bit allocation tables

tables are listed in Table (4.1). The total bits used for each 8×8 image block are 76, 58, and 24 bits (respectively yields rates 1.19, 0.9 and 0.375 bpp). The bit allocation table for the 1.19 bpp system is adopted directly from [42]; and in the 0.375 bpp system, only the first three transform coefficients are preserved and given the highest level of protection. By fixed bit allocation table, we mean that we apply the same table *globally* for *any* image under *any* channel condition. The advantage of using a fixed bit allocation method over using an adaptive optimal bit allocation technique is that the former one does not require overhead information; this results in a reduction of the encoder/decoder complexity and the bandwidth requirement of the overall system. The results obtained via this simple approach are not optimal. In Section 4.3, we study the loss of optimality by using such fixed bit allocation tables.

The performance results can be found in Tables (4.2)-(4.4) for decoded Lena. We denote COSQ as our proposed scheme. The reference system chosen for comparison purposes is SQ-IL which denotes the system with Lloyd-Max quantizer followed with NBC codeword assignment over an interleaved Markov channel. In this case, we assume that the Markov channel has been rendered memoryless (i.e. $\delta = 0$) via an

ideal interleaver. We shall call this reference system the tandem coding system.

As shown from the PSNR tables, the data in the $\epsilon = 0$ column represent the PSNR values for compressed Lena with the given bit allocation table, in which the distortions are exclusively due to the quantization errors. The data in the $\delta = 0$ row represent the PSNR values for the BSC. It is apparent that the COSQ system outperforms the reference system in all cases, especially in very noisy channel environments with high noise correlation. More specifically, the improvement can be as high as 12 dB for Lena with $\epsilon = 0.1$ and $\delta = 10.0$ at 1.19 bpp. As shown, the tandem coding scheme is extremely sensitive to the channel BER (ϵ); its performance degrades very quickly as the channel gets noisier. In contrast, when ϵ increases, the performance of the COSQ system degrades slowly and this is particularly true for channels with high correlation parameter. Similar observations can be made for the systems with overall operation bit rates at 0.9 bpp and 0.375 bpp.

In the context of image coding, no final judgment can be made without a subjective performance measure. In Figure 4.3, we show the decoded Lena with our proposed COSQ system and the tandem system SQ-IL. It can be seen that under very bad channel conditions $\epsilon = 0.1$, the quality of the image decoded with our proposed COSQ system is still acceptable. Additional image results for Baboon, for Goldhill at a medium bit rate, and for Peppers at a very low bit rate (0.375 bpp) are displayed in Figures 4.4, 4.5, and 4.6 respectively. We conclude that the proposed COSQ system outperforms the reference tandem coding scheme both objectively and subjectively.

We also observe that for a fixed ϵ, the performance of the COSQ scheme for the channel with memory outperforms the COSQ scheme with zero noise correlation

(interleaved system). Real life channels always exhibit some kind of memory. Traditionally the technique of interleaving/de-interleaving is used if one knows nothing about the channel memory. The question is that if one can characterize the channel memory (e.g. via Markov models), what kind of advantage does he have? One way of showing this advantage is to incorporate the memory into the quantizer design and the result can be seen from the PSNR values of decoded Lena. For $\epsilon = 0.1$, on average there is an extra 3.45 dB gain for a channel with $\delta = 5.0$ over an interleaved system ($\delta = 0$). In Figure 4.8, we observe, for Lena, that more gains due to the channel memory can be achieved in the noisy channel region. In Figure 4.7 we illustrate subjectively the COSQ gain due to the channel memory for Lena at BER rates $\epsilon = 0.05$ and $\epsilon = 0.1$. Thus we conclude that if we can characterize the channel statistics; it is better to exploit the noise correlation rather than to destroy it by using interleaving. The latter one introduces additional delay into the system.

4.3 Mismatch in the System Design

4.3.1 Bit Allocation Mismatch

As stated in Section 3.2, bit allocation addresses the proper distribution of the available bits to the transform coefficients. It determines which coefficients should be kept for coding and transmission and how coarsely the retained coefficients should be quantized [42]. Bit allocation is usually performed with either adaptive threshold coding or zonal coding. Threshold coding is an adaptive method which is specified in the JPEG image standard [46]. This coding scheme is based on the fact that different image blocks have different spectral and statistical behaviors so that adaptive

bit allocation methods should be used for each image sub-block. In threshold coding, a quantization matrix is usually applied to the sub-block coefficient matrix and only the coefficients with magnitudes above a certain threshold are retained. Thus the bit allocation table will differ from sub-block to sub-block. A common strategy in adaptive coding is to use a zig-zag sequence scan in conjunction with run-length coding methods. In zonal coding, the locations for the coefficients which carry the most information are retained via a zonal mask which is applied globally for each image sub-block. The zonal bit allocation table is adaptively calculated for each input image. In the presence of channel noise, zonal coding translates into adaptively computing the optimal bit allocation tables by considering the channel parameters. The bit allocation table is coded and transmitted along with the image coefficients.

In contrast, a simple fixed bit allocation scheme applies the same bit allocation table for any image and under any channel condition. It results in reducing the encoder/decoder complexity and does not require transmitting any overhead information. Obviously a system which employs a fixed bit allocation scheme will not be optimal. We herein evaluate the loss of optimality due to the incorporation of the fixed bit allocation method in our proposed system. This is achieved by comparing it to a similar system which employs an optimal bit allocation scheme which minimizes the overall distortion. Here we have assumed that the exact optimal bit allocation tables are available at the decoder. The method for calculating the optimal bit allocation table in the context of the Markov channel model is described in Section 3.2 and the results are presented in Tables (3.7)-(3.9) for Lena at 1.19, 0.90, and 0.375 bpp at various channel conditions. The fixed bit allocation tables are listed in Table (4.1).

In Tables (4.5)-(4.7), COSQ-OPT denotes the schemes using the optimal bit allocation table and COSQ-FIX denotes the fixed bit allocation schemes used in Section 4.2. Our results show that the loss of optimality by using the fixed bit allocation table is small. In most cases, the loss is only about 1.0 dB; this gap narrows as the channel conditions deteriorate. A similar behavior is observed for various other images.

4.3.2 Mismatch in channel parameters

In the design of the channel optimized quantizer, it is assumed that the knowledge of the source distribution and the channel conditions are known *a priori* for the quantizer design. In this thesis, we assumed that the DC and AC transform coefficients follow the Gaussian and Laplacian distributions respectively. To study the source mismatch, one can find out the shaping factor α of generalized Gaussian distribution by using the Kolmogorov-Smirnov test [37], [41] for each image. This exercise can not be done in real-time applications; however we believe the problem of the source mismatch is minor in most cases. In this subsection we examine the mismatch under various channel conditions for both ideal and image sources.

Channel mismatch results for the Gaussian and Laplacian sources are presented in Tables (4.8)-(4.11) for the channel optimized scalar quantizer (COSQ) with two different bit rates ($R = 4$ and $R = 8$). ϵ_a and ϵ_d denote the actual and designed BER; Similarly δ_a and δ_d denote the actual and designed noise correlation parameters. For the Gaussian source, under noiseless matched condition, the performance goes from 20.18 dB at $R = 4$ to 43.57 dB at $R = 8$ with 23.39 dB improvement. This reflects the well-known result that for the Gaussian source, every bit results in a 6 dB improvement. The tabulated data also illustrate the fact that the Gaussian source

outperforms the Laplacian source in pdf-optimized COSQ systems. For fixed δ, the quantizer performances indicate that the COSQ systems are relatively insensitive to the channel BER mismatch provided that $\epsilon_d > \epsilon_a$. For the data in the row of $\epsilon_a = 0.0$, the performance degradation is entirely due to the quantization distortion. As expected this degradation becomes large for larger bit rates and at larger ϵ_d values. For fixed ϵ, we observe that the system performance are not very sensitive to the mismatch of the noise correlation. The similar performance behavior due to the channel mismatch is shown in Tables (4.12) and (4.13) for Lena. We can conclude from the results that it is in general better to overestimate the true parameters than to underestimate them.

4.4 Results from Using COVQ

In this subsection, we present some experimental results by using COVQ directly on Lena with the Markov channel. The dimension was chosen to be 4×2 pixels (or $k = 8$) and a rate of 1 bpp was maintained. A set of training images (Goldhill, Airplane, Tiffany, Peppers, and Sailboat) other than Lena were used to obtain the empirical k-fold source distribution. Again the quantizers were assumed to be matched to both the source distribution and the channel conditions. All results were obtained over 25 experiments and the PSNR performance for decoded Lena can be found in Table (4.14). Here COVQ denotes the system which employs COVQ and the outputs of the COVQ are sent over the Markov channel directly. VQ-IL denotes the system with VQ (designed for noiseless channel) and indices of the codebook are sent over an ideal interleaved Markov channel. Similar observations can be made as the cases with DCT-based COSQ systems. Channel optimized quantization schemes outperform the

reference tandem systems in all cases, especially at bad channel conditions. A system which exploits the channel memory has more gains compared against a system which destroys the memory. By comparing the COVQ PSNR values with those from COSQ system with overall bit rate of 0.9 bpp in Table (4.3), we see that the performance of the COVQ system is quite comparable. However it should be emphasized that COVQ system will normally suffer from delays and depend on the training image sequences, while the DCT-based COSQ system is delayless (scalar quantizer) and the source distributions are image independent.

δ	System	ϵ=0	ϵ=0.005	ϵ=0.01	ϵ=0.05	ϵ=0.1
0	COSQ	32.33	30.42	29.50	25.18	22.42
0	SQ-IL	32.33	27.30	24.99	18.59	15.69
5	COSQ	32.33	31.06	30.32	27.43	25.87
5	SQ-IL	32.33	27.30	24.99	18.59	15.69
10	COSQ	32.33	31.41	30.83	28.98	27.74
10	SQ-IL	32.33	27.30	24.99	18.59	15.69

Table 4.2: Average PSNR (dB) of decoded Lena over the Markov channel ($M = 1$) with BER ϵ and correlation parameter δ using fixed bit allocation table at 1.19 bpp.

δ	System	ϵ=0	ϵ=0.005	ϵ=0.01	ϵ=0.05	ϵ=0.1
0	COSQ	30.49	29.26	28.63	24.92	22.29
0	SQ-IL	30.49	26.57	24.56	18.50	15.68
5	COSQ	30.49	29.58	29.01	26.93	25.58
5	SQ-IL	30.49	26.57	24.56	18.50	15.68
10	COSQ	30.49	29.85	29.49	28.23	27.20
10	SQ-IL	30.49	26.57	24.56	18.50	15.68

Table 4.3: Average PSNR (dB) of decoded Lena over the Markov channel ($M = 1$) with BER ϵ and correlation parameter δ using fixed bit allocation table at 0.90 bpp.

δ	System	ϵ=0	ϵ=0.005	ϵ=0.01	ϵ=0.05	ϵ=0.1
0	COSQ	26.16	25.80	25.60	23.74	21.71
0	SQ-IL	26.16	24.37	23.06	18.19	15.58
5	COSQ	26.16	25.91	25.07	24.72	23.98
5	SQ-IL	26.16	24.37	23.06	18.19	15.58
10	COSQ	26.16	26.00	25.89	25.36	24.92
10	SQ-IL	26.16	24.37	23.06	18.19	15.58

Table 4.4: Average PSNR (dB) of decoded Lena over the Markov channel ($M = 1$) with BER ϵ and correlation parameter δ using fixed bit allocation table at 0.375 bpp.

Original Lena (512×512)

Decoded Lena with SQ-IL,

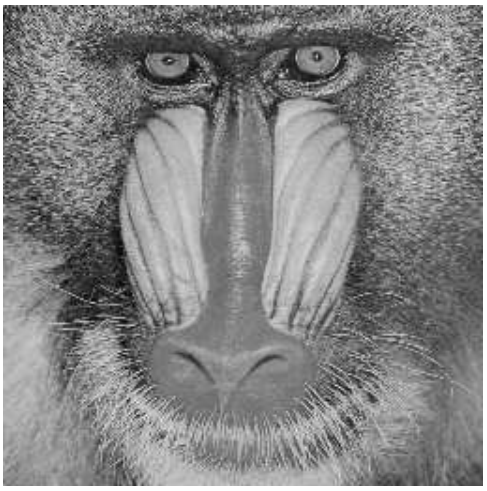
$PSNR = 15.84$ dB

Compressed Lena at 1.19 bpp

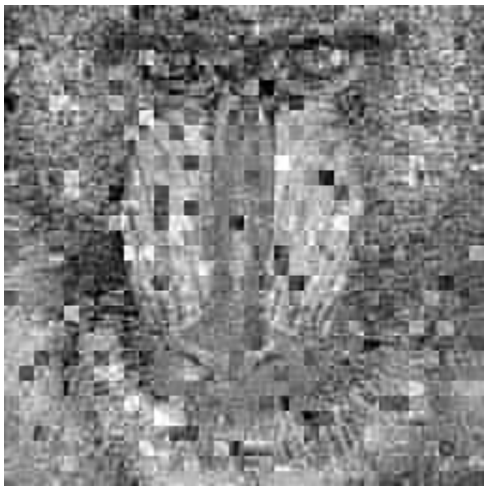
Decoded Lena with COSQ,

$\delta = 10.0$, $PSNR = 27.88$ dB

Figure 4.3: Lena: Overall rate is 1.19 bpp; Markov channel with $\epsilon = 0.1$.

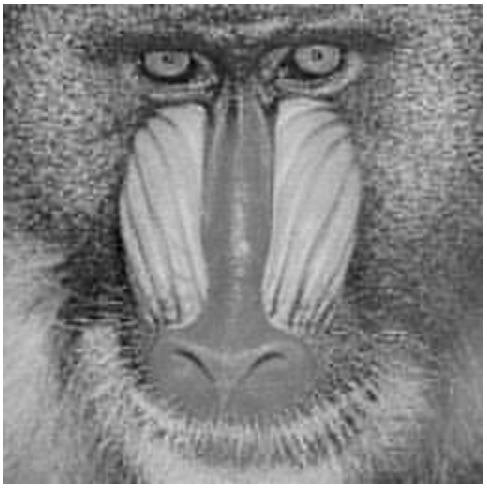


Original Baboon (256×256)

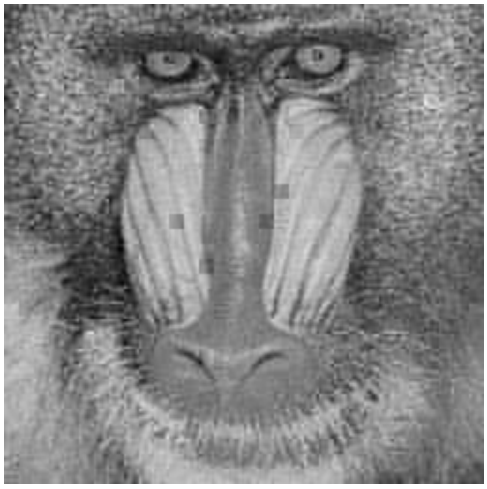


Decoded Baboon with SQ-IL,

$PSNR = 16.52$ dB



Compressed Baboon at 1.19 bpp

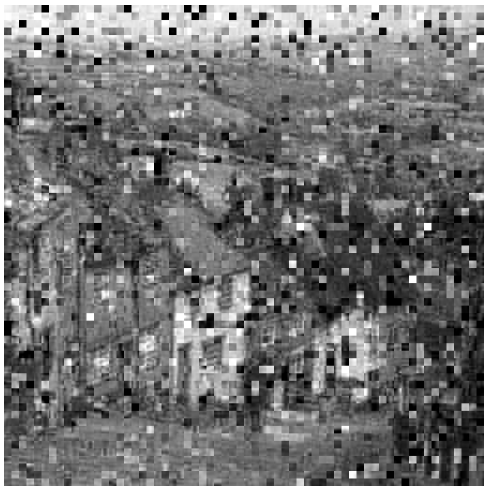


Decoded Baboon with COSQ,

$\delta = 10.0$, $PSNR = 21.32$ dB

Figure 4.4: Baboon: Overall rate is 1.19 bpp; Markov channel with $\epsilon = 0.1$.

Original Goldhill (512×512)



Decoded Goldhill with SQ-IL,

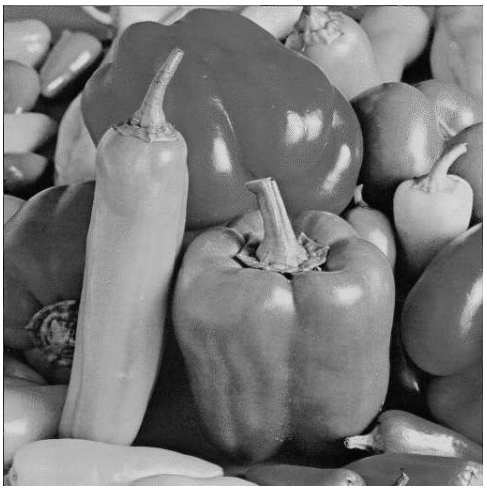
$PSNR = 16.32$ dB

Compressed Goldhill at 0.90 bpp

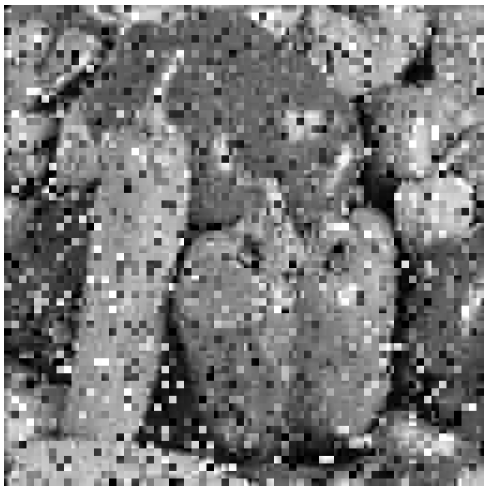
Decoded Goldhill with COSQ,

$\delta = 10.0$, $PSNR = 27.85$ dB

Figure 4.5: Goldhill: Overall rate is 0.90 bpp; Markov channel with $\epsilon = 0.1$.

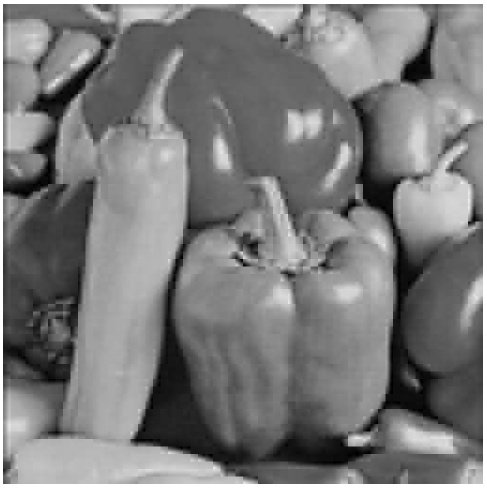


Original Peppers (512×512)

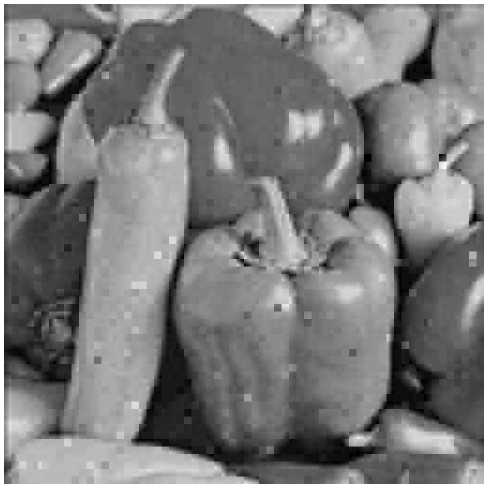


Decoded Peppers with SQ-IL,

$PSNR = 15.82$ dB



Compressed Peppers at 0.375 bpp



Decoded Pepper with COSQ,

$\delta = 10.0$, $PSNR = 25.22$ dB

Figure 4.6: Peppers: Overall rate is 0.375 bpp; Markov channel with $\epsilon = 0.1$.

Decoded Lena with $\delta = 0$,

$\epsilon = 0.05$, $PSNR = 25.49$ dB

Decoded Lena with $\delta = 10.0$,

$\epsilon = 0.05$, $PSNR = 29.16$ dB

Decoded Lena with $\delta = 0.0$,

$\epsilon = 0.1$, $PSNR = 22.67$ dB

Decoded Lena with $\delta = 10.0$,

$\epsilon = 0.1$, $PSNR = 27.88$ dB

Figure 4.7: Lena: Overall rate is 1.19 bpp; COSQ over the Markov channel.

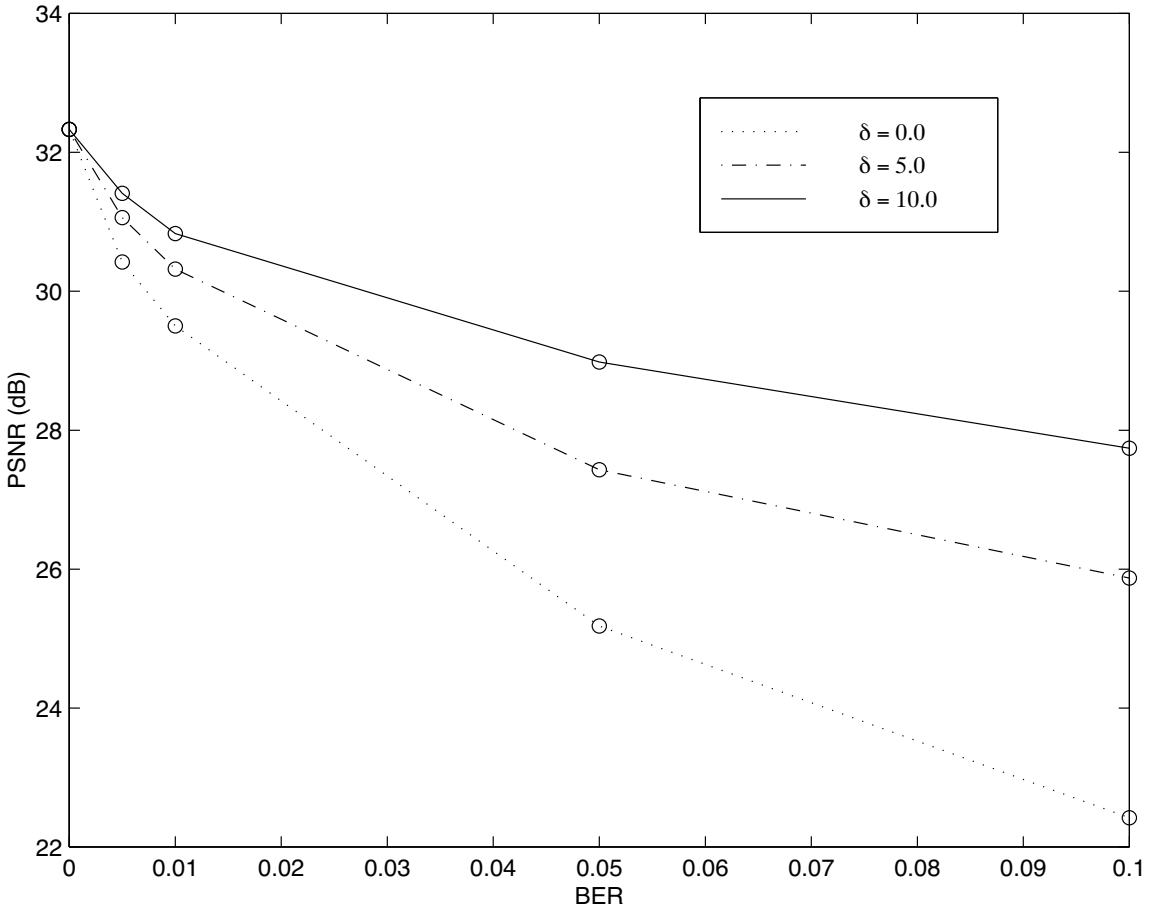


Figure 4.8: Gain due to memory for the Markov channel; Decoded Lena at 1.19 bpp.

δ	System	ϵ=0	ϵ=0.005	ϵ=0.01	ϵ=0.05	ϵ=0.1
0	COSQ-OPT	33.07	30.71	29.74	25.43	22.55
0	COSQ-FIXED	32.33	30.42	29.50	25.18	22.42
5	COSQ-OPT	33.07	31.48	30.61	27.55	26.00
5	COSQ-FIXED	32.33	31.06	30.32	27.43	25.87
10	COSQ-OPT	33.07	31.88	31.32	29.17	28.11
10	COSQ-FIXED	32.33	31.44	30.83	28.98	27.74

Table 4.5: Performance comparison between COSQ systems using fixed and optimal bit allocation tables; PSNR (dB) of decoded Lena over the Markov channel with BER ϵ and correlation parameter δ at 1.19 bpp.

δ	System	ϵ=0	ϵ=0.005	ϵ=0.01	ϵ=0.05	ϵ=0.1
0	COSQ-OPT	31.76	29.87	29.05	25.07	22.40
0	COSQ-FIXED	30.49	29.26	28.63	24.92	22.29
5	COSQ-OPT	31.76	30.40	29.62	27.12	25.75
5	COSQ-FIXED	30.49	29.58	29.01	26.93	25.58
10	COSQ-OPT	31.76	30.83	30.25	28.65	27.43
10	COSQ-FIXED	30.49	29.85	29.49	28.23	27.20

Table 4.6: Performance comparison between COSQ systems using fixed and optimal bit allocation tables; PSNR (dB) of decoded Lena over the Markov channel with BER ϵ and correlation parameter δ at 0.9 bpp.

δ	System	ϵ=0	ϵ=0.005	ϵ=0.01	ϵ=0.05	ϵ=0.1
0	COSQ-OPT	27.91	26.85	26.41	23.81	21.74
0	COSQ-FIXED	26.16	25.80	25.60	24.74	21.71
5	COSQ-OPT	27.91	27.00	26.76	25.24	24.29
5	COSQ-FIXED	26.16	25.91	25.70	24.72	23.98
10	COSQ-OPT	27.91	27.44	27.14	26.22	25.39
10	COSQ-FIXED	26.16	26.00	25.89	25.36	24.92

Table 4.7: Performance comparison between COSQ systems using fixed and optimal bit allocation tables; PSNR (dB) of decoded Lena over the Markov channel with BER ϵ and correlation parameter δ at 0.375 bpp.

		ϵ_d=0.0	ϵ_d=0.01	ϵ_d=0.1
$R=4$	ϵ_a=0.0	20.18	20.02	17.35
	ϵ_a=0.01	11.86	16.34	15.47
	ϵ_a=0.1	2.48	8.78	9.58
$R=8$	ϵ_a=0.0	43.57	31.01	25.98
	ϵ_a=0.01	12.09	23.57	23.55
	ϵ_a=0.1	2.09	14.65	16.92

Table 4.8: Channel mismatch results in SNR (dB) for the Gaussian source with $R=4$ and $R=8$, $k=1$: Mismatch in BER (δ_a=δ_d=10.0).

		δ_d=0.0	δ_d=5.0	δ_d=10.0
$R=4$	δ_a=0.0	12.58	9.52	8.91
	δ_a=5.0	11.60	14.84	14.92
	δ_a=10.0	11.51	15.97	16.34
$R=8$	δ_a=0.0	20.13	14.04	12.72
	δ_a=5.0	14.12	21.38	20.92
	δ_a=10.0	14.32	23.54	23.57

Table 4.9: Channel mismatch results in SNR (dB) for the Gaussian source with $R=4$ and $R=8$, $k=1$: Mismatch in δ (ϵ_a=ϵ_d=0.01).

		ϵ_d=0.0	ϵ_d=0.01	ϵ_d=0.1
$R=4$	ϵ_a=0.0	18.09	17.75	14.72
	ϵ_a=0.01	8.96	14.01	13.51
	ϵ_a=0.1	-0.54	6.32	8.71
$R=8$	ϵ_a=0.0	41.67	28.67	24.00
	ϵ_a=0.01	8.28	21.93	21.60
	ϵ_a=0.1	-1.74	13.11	14.97

Table 4.10: Channel mismatch results in SNR (dB) for the Laplacian source with $R=4$ and $R=8$, $k=1$: Mismatch in BER (δ_a=δ_d=10.0).

		δ_d=0.0	δ_d=5.0	δ_d=10.0
$R=4$	δ_a=0.0	10.77	7.34	6.94
	δ_a=5.0	9.74	13.58	12.80
	δ_a=10.0	9.61	14.87	14.01
$R=8$	δ_a=0.0	18.17	12.60	10.21
	δ_a=5.0	11.26	20.08	19.41
	δ_a=10.0	11.67	21.93	21.93

Table 4.11: Channel mismatch results in SNR (dB) for the Laplacian source with $R=4$ and $R=8$, $k=1$: Mismatch in δ (ϵ_a=ϵ_d=0.01).

	ϵ_d=0.0	ϵ_d=0.01	$\epsilon_d=0.1$
ϵ_a=0.0	32.33	31.88	31.02
ϵ_a=0.01	24.97	30.83	30.52
ϵ_a=0.1	15.63	26.28	27.74

Table 4.12: Performance (in dB) of decoded Lena under mismatch in ϵ ($\delta_a=\delta_d=$ 10.0).

	δ_d=0.0	δ_d=5.0	$\delta_d = 10.0$
δ_a=0.0	29.50	26.29	24.85
δ_a=5.0	26.33	30.32	30.01
δ_a=10.0	26.66	30.82	30.83

Table 4.13: Performance (in dB) of decoded Lena under mismatch in δ ($\epsilon_a = \epsilon_d = 0.01$).

δ	System	ϵ=0	ϵ=0.005	ϵ=0.01	ϵ=0.05	ϵ=0.1
0	COVQ	30.07	29.27	28.37	24.43	21.83
0	VQ-IL	30.07	22.24	19.67	13.56	11.23
5	COVQ	30.07	30.01	29.30	26.49	25.07
5	VQ-IL	30.07	22.24	19.67	13.56	11.23
10	COVQ	30.07	30.04	29.73	27.84	26.54
10	VQ-IL	30.07	22.24	19.67	13.56	11.23

Table 4.14: Average PSNR (dB) of decoded Lena over the Markov channel ($M = 1$) using COVQ with dimension = 4×2, $R = 1$ bit per sample.

Chapter 5

Conclusions and Future Work

5.1 Summary

In chapter 2 we described channel optimized quantization and designed a COVQ system which is applied for generalized Gaussian sources and an additive channel model with memory. In designing the COVQ systems, we exploit the intra-block memory of the channel by incorporating the knowledge of the noise correlation parameter into the quantizer design. With this scheme, significant improvements have been achieved over interleaved systems. The moral to be drawn from this work is that it is best to make good use of the channel memory, if such information exists, instead of trying to destroy it through interleaving.

Chapter 3 addresses the problem of optimal bit allocation in the context of transform coding and transmission over noisy channels with memory. An integer programming technique called steepest descent algorithm is successfully applied to find the optimal bit allocation tables such that the end-to-end distortion is minimized.

A DCT-based combined source-channel coding system is proposed in Chapter 4 for transmitting grey-level images over a binary channel with additive Markov noise. It

consists of a channel optimized quantization scheme described in Chapter 2. Extensive experimental results show that this simple system – which employs a fixed zonal coding bit allocation technique – provides significant improvements over traditional tandem systems, especially during bad channel conditions. The loss of optimality due to the use of fixed bit allocation tables is also examined. The loss is shown to be very small for various images; this suggests that a reduction in complexity and bandwidth requirements can further be achieved. Experiments are performed under various channel mismatched conditions, it is shown that the proposed system is fairly robust under certain conditions.

The purpose of this thesis is to *propose* a combined source-channel coding system for image transmission over very noisy channels with memory. It is worthwhile to point out that the tandem system we choose in Chapter 4 is *not* the best reference system. For comparison purposes, we kept the quantizer complexity fixed and did not allow room for additional error control codes. In a recent work [20], Goldsmith et al. address the problem of optimal bit allocation between source and channel encoders so that forward error correction (FEC) codes can be used in conjunction with the channel optimized quantizers. They consider an additive white Gaussian channel (AWGN) which is used with BPSK modulation. However it is not entirely clear that the performance gains over the COVQ system are due to the so-called optimal tandem system or due to the fact that their channel decoder employs soft-decision decoding.

5.2 Future Work

There are several directions for future work. For now, we have been working with the channel modeled by the Polya contagion urn [2]. We plan to use COVQ as a vehicle

to work with other channel models with memory, such as the Gilbert channel model. More specifically we wish to use COVQ to evaluate how well the Markov channel can model the Gilbert channel. We believe good results may be obtained by increasing the order of the Markov channel model.

In order to fully exploit the high correlation among inter-block DC coefficients, a COVQ system must be employed. However due to the complexity of the full-search VQ algorithm, we are not able to implement this feature. One approach to this problem is to design a sub-optimal COVQ scheme. Phamdo et al. [34], [35] studied this problem but only for the BSC model. We plan to extend this technique to channels with memory.

In this thesis, DCT is chosen in the source encoder process for the sake of its simplicity and popularity. Parallel results can also be obtained for subband coding methods. We expect the result for subband coded system to be slightly better than the DCT system. In addition, more powerful data compression schemes such as wavelet method can also be studied in the frame work of combined source-channel coding. This thesis only deals with image compression and transmission over channel with memory; similar work can also be extended to video applications. We also plan to incorporate a soft-decision decoding technique into the proposed combined source-channel coding system to obtain extra gains. This is inspired by the recent work in [4].

Appendix A

Quantification of Reconstruction Errors

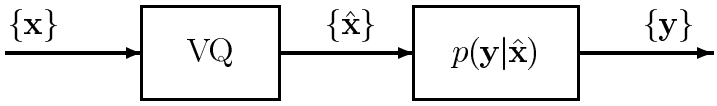


Figure A.1: Block diagram for a simple transmission system

Let $\mathbf{X}$ be the n-dimensional input vector to the vector quantizer (VQ), and $\hat{\mathbf{X}}$ be the reconstructed vector taking values from the alphabet $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N\}$. The output of the VQ is transmitted over a noisy channel described by the block channel transition probability $p(\mathbf{y}|\hat{\mathbf{x}})$. The received vectors $\mathbf{Y}$ have the same alphabet as $\hat{\mathbf{X}}$ (see Figure A.1). The total end-to-end mean square error distortion ϵ^2 can be expressed as

$$\begin{aligned}
\epsilon^2 &= \mathbf{E}\{|| \mathbf{X} - \mathbf{Y} ||^2\} \qquad \text{(A.1)} \\
&= \mathbf{E}\{|| \mathbf{X} - \hat{\mathbf{X}} + \hat{\mathbf{X}} - \mathbf{Y} ||^2\} \\
&= \mathbf{E}\{(\mathbf{X} - \hat{\mathbf{X}} + \hat{\mathbf{X}} - \mathbf{Y})^t(\mathbf{X} - \hat{\mathbf{X}} + \hat{\mathbf{X}} - \mathbf{Y})\} \\
&= \mathbf{E}\{(\mathbf{X} - \hat{\mathbf{X}})^t(\mathbf{X} - \hat{\mathbf{X}})\} + \mathbf{E}\{(\hat{\mathbf{X}} - \mathbf{Y})^t(\hat{\mathbf{X}} - \mathbf{Y})\} + 2\mathbf{E}\{(\mathbf{X} - \hat{\mathbf{X}})^t(\hat{\mathbf{X}} - \mathbf{Y})\}
\end{aligned}$$

$$
\begin{aligned}
&= \underbrace{\mathbf{E}\{|| \mathbf{X} - \hat{\mathbf{X}} ||^2\}}_{\epsilon_q^2} + \underbrace{\mathbf{E}\{|| \hat{\mathbf{X}} - \mathbf{Y} ||^2\}}_{\epsilon_c^2} + \underbrace{2\mathbf{E}\{\mathbf{X}^t\hat{\mathbf{X}} - \mathbf{X}^t\mathbf{Y} - \hat{\mathbf{X}}^t\hat{\mathbf{X}} + \hat{\mathbf{X}}^t\mathbf{Y}\}}_{\epsilon_*^2} \\
&= \epsilon_q^2 + \epsilon_c^2 + \epsilon_*^2,
\end{aligned}
$$

where ϵ_q^2 is the distortion due to the vector quantization; and ϵ_c^2 is the distortion due to the noisy channel. We wish to find out the condition under which the cross-term ϵ_*^2 becomes zero. To do this, we need to find out the joint probability density functions $p(\mathbf{x}, \hat{\mathbf{x}})$, $p(\mathbf{x}, \mathbf{y})$ and $p(\hat{\mathbf{x}}, \mathbf{y})$. According to the VQ operation, we have

$$
p\{\hat{\mathbf{x}} = \mathbf{c}_i\} = \int_{S_i} p(\mathbf{x}) d\mathbf{x} \tag{A.2}
$$

where $p(\mathbf{x})$ is the n-fold probability density function of $\mathbf{x}$. We define $\overline{p_i(\mathbf{x})}$ as

$$
\overline{p_i(\mathbf{x})} = \int_{S_i} p(\mathbf{x}) d(\mathbf{x}). \tag{A.3}
$$

Now let the block channel transition probability P_{ij} be

$$
P_{ij} = Pr\{\mathbf{y} = \mathbf{c}_j \mid \hat{\mathbf{x}} = \mathbf{c}_i\} \tag{A.4}
$$

which is the probability that the received code vector is $\mathbf{c}_j$ given that the vector $\mathbf{c}_i$ is sent. We also define $p_i(\mathbf{x})$ as

$$
p_i(\mathbf{x}) = \begin{cases} p(\mathbf{x}) & \mathbf{x} \in S_i \\ 0 & \text{otherwise.} \end{cases} \tag{A.5}
$$

With the above notations, it is straightforward to write the joint probability density functions of $\mathbf{x}$ and $\hat{\mathbf{x}}$ as

$$
p(\mathbf{x}, \hat{\mathbf{x}}) = \sum_i \delta(\hat{\mathbf{x}} - \mathbf{c}_i) p_i(\mathbf{x}) \tag{A.6}
$$

where $\delta(\cdot)$ denotes the Kronecker delta function which is defined as

$$\delta(\mathbf{x}-\mathbf{y}) \triangleq \begin{cases} 1 & \text{if } \mathbf{x}=\mathbf{y} \\ 0 & \text{otherwise,} \end{cases} \tag{A.7}$$

satisfying

$$\int_S f(\mathbf{x})\delta(\mathbf{x}-\mathbf{x}_0)d\mathbf{x} = f(\mathbf{x}_0). \tag{A.8}$$

To obtain the density function of the VQ output $\hat{\mathbf{x}}$, we can integrate $p(\mathbf{x},\hat{\mathbf{x}})$ with respect to $\mathbf{x}$ and get

$$\begin{aligned} p(\hat{\mathbf{x}}) &= \int_{\mathrm{R}^n} p(\mathbf{x},\hat{\mathbf{x}})d\mathbf{x} && \text{(A.9)} \\ &= \int_{\mathrm{R}^n} \sum_i \delta(\hat{\mathbf{x}}-\mathbf{c}_i)p_i(\mathbf{x})d\mathbf{x} && \text{(A.10)} \\ &= \sum_i \delta(\hat{\mathbf{x}}-\mathbf{c}_i)\overline{p_i(\mathbf{x})}. && \text{(A.11)} \end{aligned}$$

The other joint density functions can be simply expressed as

$$p(\mathbf{x},\mathbf{y}) = \sum_i \delta(\mathbf{y}-\mathbf{c}_i)\sum_j p_j(\mathbf{x})P_{ji}, \tag{A.12}$$

and

$$p(\hat{\mathbf{x}},\mathbf{y}) = \sum_i \sum_j \overline{p_i(\mathbf{x})}P_{ij}\delta(\hat{\mathbf{x}}-\mathbf{c}_i)\delta(\mathbf{y}-\mathbf{c}_j). \tag{A.13}$$

Now we can evaluate the individual cross term expectations as

$$\begin{aligned} \mathbf{E}\{\mathbf{X}^t\hat{\mathbf{X}}\} &= \iint_{\mathrm{R}^n} \mathbf{x}^t\hat{\mathbf{x}}\sum_i \delta(\hat{\mathbf{x}}-\mathbf{c}_i)p_i(\mathbf{x})d\mathbf{x}d\hat{\mathbf{x}} && \text{(A.14)} \\ &= \int_{\mathrm{R}^n} \mathbf{x}^t\left\{\sum_i \int_{\mathrm{R}^n} \hat{\mathbf{x}}\delta(\hat{\mathbf{x}}-\mathbf{c}_i)p_i(\mathbf{x})d\hat{\mathbf{x}}\right\}d\mathbf{x} && \text{(A.15)} \end{aligned}$$

$$= \int_{\mathrm{R}^n} \mathbf{x}^t \sum_i p_i(\mathbf{x}) \mathbf{c}_i d\mathbf{x} \tag{A.16}$$

$$= \sum_i \int_{S_i} \mathbf{x}^t p(\mathbf{x}) d\mathbf{x}\, \mathbf{c}_i, \tag{A.17}$$

and

$$\mathbf{E}\{|| \hat{\mathbf{X}} ||^2\} = \int_{\mathrm{R}^n} \hat{\mathbf{x}}^t \hat{\mathbf{x}} \sum_i \delta(\hat{\mathbf{x}} - \mathbf{c}_i) \overline{p_i(\mathbf{x})} d\hat{\mathbf{x}} \tag{A.18}$$

$$= \sum_i \mathbf{c}_i^t \mathbf{c}_i \overline{p_i(\mathbf{x})} \tag{A.19}$$

$$= \sum_i \mathbf{c}_i^t \overline{p_i(\mathbf{x})} \mathbf{c}_i, \tag{A.20}$$

and

$$\mathbf{E}\{\mathbf{X}^t \mathbf{Y}\} = \iint_{\mathrm{R}^n} \mathbf{x}^t \mathbf{y} \sum_i \delta(\mathbf{y} - \mathbf{c}_i) \sum_j p_j(\mathbf{x}) P_{ji} d\mathbf{x} d\mathbf{y} \tag{A.21}$$

$$= \sum_i \int_{\mathrm{R}^n} \mathbf{x}^t \sum_j p_j(\mathbf{x}) P_{ji} \mathbf{c}_i d\mathbf{x}, \tag{A.22}$$

and

$$\mathbf{E}\{\hat{\mathbf{X}}^t \mathbf{Y}\} = \iint_{\mathrm{R}^n} \hat{\mathbf{x}}^t \mathbf{y} \sum_i \sum_j \overline{p_i(\mathbf{x})} P_{ij} \delta(\hat{\mathbf{x}} - \mathbf{c}_i) \delta(\mathbf{y} - \mathbf{c}_j) d\hat{\mathbf{x}} d\mathbf{y} \tag{A.23}$$

$$= \sum_i \sum_j \overline{p_i(\mathbf{x})} P_{ij} \mathbf{c}_i^t \mathbf{c}_j. \tag{A.24}$$

Now let the reconstruction vector $\mathbf{c}_i$ be the centroid of the corresponding encoding region S_i

$$\mathbf{c}_i = \frac{\int_{S_i} \mathbf{x} p(\mathbf{x}) d\mathbf{x}}{\int_{S_i} p(\mathbf{x}) d\mathbf{x}}. \tag{A.25}$$

Substituting (A.25) into (A.17), (A.20), (A.22), and (A.24) respectively, yields

$$\mathbf{E}\{\mathbf{X}^t \hat{\mathbf{X}}\} = \mathbf{E}\{\hat{\mathbf{X}}^t \hat{\mathbf{X}}\} \tag{A.26}$$

$$\mathbf{E}\{\mathbf{X}^t \mathbf{Y}\} = \mathbf{E}\{\hat{\mathbf{X}}^t \mathbf{Y}\}. \tag{A.27}$$

Thus we conclude if the VQ is designed to be optimal, we can write the total end-to-end distortion ϵ^2 as the sum of distortion due to quantization ϵ_q^2, and distortion due to channel error ϵ_c^2.

Bibliography

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform," *IEEE Trans. Comp.*, vol. 23, pp. 90–93, 1974.

[2] F. Alajaji and T. Fuja, "A Communication Channel Model on Contagion," *IEEE Trans. Inform. Theory*, vol. 40, pp. 2035–2041, Nov. 1994.

[3] F. Alajaji, S. Al-Semari and P. Burlina, "An Unequal Error Protection Trellis Coding Scheme for Still Image Communication," *Proceedings of the 1997 IEEE International Symposium on Information Theory*, Ulm, Germany, June 1997.

[4] F. Alajaji and N. Phamdo, "A Vector Quantizer for Additive White and Colored Gaussian Noise Channels," *Proceedings of the 1997 IEEE International Symposium on Information Theory*, Ulm, Germany, June 1997.

[5] E. Ayanoglu and R. M. Gray, "The Design of Joint Source and Channel Trellis Waveform Coders," *IEEE Trans. Inform. Theory*, vol. 33, pp 855–865, Nov. 1987.

[6] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[7] R. E. Blahut, "Computation of Channel Capacity and Rate Distortion Function," *IEEE Trans. Inform. Theory*, vol. IT-18, pp 460–473, 1972.

[8] R. E. Blahut, *Principles and Practice of Information Theory*, Addison Weasley, Reading, MA, 1987.

[9] Q. Chen and T. R. Fischer, "Image Coding Using Robust Quantization for Noisy Digital Transmission," *Submitted to IEEE Trans. on Image Processing.*

[10] R. J. Clark, *Transform Coding of Images*. Academic Press, 1985.

[11] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 1991.

[12] N. Farvardin and V. Vaishampayan, "Optimal Quantizer Design for Noisy Channels: An Approach to Combined Source-Channel Coding," *IEEE Trans. Inform. Theory*, vol. 33, pp. 827-838, Nov. 1987.

[13] N. Farvardin, "A Study of Vector Quantization for Noisy Channels," *IEEE Trans. Inform. Theory*, vol. 36, pp. 799–809, July 1990.

[14] N. Farvardin and V. Vaishampayan, "On the Performance and Complexity of Channel-Optimized Vector Quantizers," *IEEE Trans. Inform. Theory*, vol. 37, pp. 155–160, Jan. 1991.

[15] P. E. Fleischer, "Sufficient Conditions for Achieving Minimum Distortion in a Quantizer," *IEEE Int. Conv. Rec, pt. 1*, pp. 104–111, 1964.

[16] R. G. Gallager, *Information Theory and Reliable Communication*. Wiley, 1968.

[17] A. A. El Gamal, L. A. Hamachandra, I. Shperling, and V. Wei, "Using simulated annealing to design good codes," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 116–123, Jan. 1987.

[18] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.

[19] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1265, 1960.

[20] A. Goldsmith and M. Effros, "Iterative Joint Design of Fixed-Rate Source Codes and Multiresolution Channel Codes," *Submitted to IEEE Trans. Commun.*, May 1997.

[21] R. M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, vol. 1, pp. 4–29, Apr. 1984.

[22] R. M. Gray, *Digital Image Processing Course Notes*, Stanford University, 1996.

[23] H. Jafarkhani and N. Farvardin, "Channel-Matched Hierarchical Table-Lookup Vector Quantization for Finite-State Channels," *Submitted to IEEE Trans. Inform. Theory*, Jan. 1997

[24] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[25] H. Kumazawa, M. Kasahara and T. Namekawa, "A Construction of Vector Quantizers for Noisy Channels," *Electronics and Engineering in Japan*, vol. 67-B, pp. 39–47, Jan. 1984.

[26] A. Kurtenbach and P. Wintz, "Quantizing for Noisy Channels," *IEEE Trans. Commun. Technology*, vol. 17, pp. 291–302, Apr. 1969.

[27] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95, Dec. 1980.

[28] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, pp. 129–137, Mar. 1982.

[29] J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," *Proc. IEEE*, vol. 23, pp. 1551–1588, Nov. 1985.

[30] J. Max, "Quantizing for Minimum Distortion," *IRE Trans. Inform. Theory*, vol. IT-6, pp. 7–12, Mar. 1960.

[31] R. J. McEliece, *The Theory of Information and Coding*. Addison Wesley, Readings, MA, 1977.

[32] J. W. Modestino and D. G. Daut, "Combined Source-Channel Coding of Images," *IEEE Trans. Commun.*, vol. COM-27, pp. 1644–1659, Nov. 1979.

[33] J. W. Modestino, D. G. Daut and A. L. Vickers, "Combined Source-Channel Coding of Images Using the Block Cosine Transform," *IEEE Trans. Commun.*, vol. COM-29, pp. 1261–1274, Sept. 1981.

[34] N. Phamdo, *Quantization Over Discrete Noisy Channels Under Complexity Constraints*. PhD thesis, Department of Electrical Engineering, University of Maryland, 1993.

[35] N. Phamdo, N. Farvardin and T. Moriya, "A Unified Approach to Tree-Structured and Multi-Stage Vector Quantization for Noisy Channels," *IEEE Trans. Inform. Theory*, vol. 39, pp. 835–850, May 1993.

[36] N. Phamdo, F. Alajaji and N. Farvardin, "Quantization of Memoryless and Gauss-Markov Sources Over Binary Markov Channels," *IEEE Trans. Commun.*, vol. 45, pp. 668–675, June 1997.

[37] R. Reininger and J. D. Gibson, "Distribution of the Two-Dimensional DCT Coefficients for Images," *IEEE Trans. Commun.*, vol. 31, pp. 835–839, June 1983.

[38] K. Sayood and J. C. Borkenhagen, "Use of Residual Redundancy in the Design of Joint Source/Channel Coders," *IEEE Trans. Commun.*, vol. 39, pp. 838–846, June 1991.

[39] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423 and 623-656, 1948.

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

[41] N. Tanabe and N. Farvardin "Subband Image Coding Using Entropy-Coded Quantization over Noisy Channels" *IEEE J. Sel. Areas Commun.*, vol. 10, pp. 926–943, June 1992.

[42] A. M. Tekalp, *Digital Video Processing*. Upper Saddle River, NJ: Prentice-Hall, 1995.

[43] R. E. Totty and G. C. Clark, "Reconstruction Error in Waveform Transmission," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 336–338, Apr. 1967.

[44] A. V. Trushkin, "Optimal Bit Allocation Algorithm for Quantizing a Random Vector," *Prob. Inform. Transmission*, pp. 156–161, Jan. 1982.

[45] V. A. Vaishampayan and N. Farvardin, "Optimal Block Cosine Transform Image Coding for Noisy Channels," *IEEE Trans. Commun.*, vol. 38, pp. 327–336, Mar. 1990.

[46] G. K. Wallace, "The JPEG Picture Compression Standard," *Commun. ACM*, pp. 31–43, Apr. 1991.

[47] H. S. Wang *Finite-State Modeling, Capacity, and Joint Source/Channel Coding for Time-Varying Channels*. PhD thesis, Department of Electrical Engineering, Rutgers University, 1992.

[48] W. Xu, J. Hagenauer and J. Hollmann, "Joint Source-Channel Decoding Using the Residual Redundancy in Compressed Images," *Proc. Int. Conf. Commun.*, Dallas, TX, June 1996.

Vita

Julian Cheng

EDUCATION

Queen's University	M.Sc.(Eng.)	Mathematics and Engineering	1996–97
University of Victoria	B.Eng. (first-class)	Electrical Engineering	1990–95

EXPERIENCE

Summer Student (1997) Nortel, Nepean, Ontario

Research Assistant (1997,1996) Mathematics and Statistics, Queen's University

Teaching Assistant (1997,1996) Mathematics and Statistics, Queen's University

Research Assistant (1995,1994,1993) Electrical and Computer Engineering, University of Victoria

Co-op Student (1994) Nortel, Nepean, Ontario

PC Application Specialist (1992) Ministry of Environment, British Columbia

Research Assistant (1991) Institute of Analysis, University of Victoria

PUBLICATIONS

J. Cheng and F. Alajaji, "Channel Optimized Quantization of Images over Bursty Channels," *The 1997 Canadian Workshop on Information Theory*, June 1997.

J. Cheng, M.A. Stuchly, Carlos de Wagter and L. Martens, "Magnetic field induced currents in a human head from use of portable appliances," *Phys. Med. Biol.* vol. 40, No. 4, pp. 495–510, Jan. 1995.

M.A. Stuchly, J. Cheng, L. Martens and C. DeWagter, "Magnetic field induced currents in a human head from use of portable appliances," *BEMS Annual Conference*, Boston, MA, June 1995.

M.A. Stuchly, J. Cheng and S. Zhao, "Modeling of human exposure to various sources of 60Hz magnetic fields," *ANTEM Symposium on Antenna Tech. & Applied Electromagnetics*, August 1994.