Multiple Description Coding: Proposed Methods And Video Application

by

Saeed Moradi

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

Queen's University
Kingston, Ontario, Canada
August 2007

Abstract

Multiple description (MD) coding and quantization has received a lot of attention recently, and has been studied extensively and extended to many demanding applications such as speech and video. We propose two multiple description quantization schemes in order to design the codebooks and partitions of side and central quantizers. The applied framework originates in the multiple description quantization via Gram-Schmidt orthogonalization approach which provides systematic treatment of the achievable rate-distortion region by subtractive dithering and successive quantization along with quantization splitting. The basic idea of our proposed MD quantization schemes is to minimize a Lagrangian cost function (defined as the weighted sum of the central and side distortions) by an iterative technique which jointly designs side codebooks and consequently forms associated partitions. In the first scheme, multiple description vector quantization with weighted sum central decoder (MDVQ-WSC), the central decoder is formed by a linear combination (weighted sum) of the side codebooks. The parameters of this linear combination are also found to minimize the central distortion. Once the side codebooks are found by the iterative technique at the final iteration, we propose to replace weighted-sum central decoder with the optimal decoder to enhance the performance of central decoder and achieve lower central distortion. In the second scheme, multiple description vector quantization with optimum central decoder (MDVQ-OC), the central codebook is found by the optimal decoder. This increases the complexity of the scheme, but also achieves lower central distortion. We compare the performance of our proposed methods with the optimum MD quantization and also with the optimal rate-distortion bound for the case of a memoryless Gaussian source. We demonstrate by simulations that our proposed methods perform very closely to the optimum MD quantizer with considerably less complexity and with a few iterations.

We also propose a multiple description video coding technique motivated by human visual perception in order to generate two correlated streams. We first calculate two simple parameters to characterize the smoothness and edge features of each block of an MPEG video frame. We employ these two parameters as a measure of the perceptual tolerance of DCT blocks against visual distortion. We duplicate the key information such as motion vectors and some low-frequency DCT coefficients, and split the remaining DCT coefficients of prediction errors according to the calculated perceptual tolerance parameter. The implementation of the proposed MD video coding preserves the core of the standard video coder and does not significantly increase the coding complexity. Simulation results reveal that our simple MD video coding method achieves superior performance compared to other similar techniques, such as simple rate-splitting method, which lack to address perceptual distortion in the design problem.

Acknowledgments

First, I would like to express my appreciation and sincerest gratitude to my advisors Professor Saeed Gazor and Professor Tamás Linder for their excellent supervision, insightful guidance, support and careful proof-readings of my thesis. I am always grateful for their kindness, continuous encouragement and all I have learned from them.

I would also like to thank the committee members Professor Fady Alajaji, Professor Parvin Mousavi, Professor Il-Min Kim, and Professor J. Scott Parent for carefully reading this thesis. I also wish to thank my lab mates, friends, and the ECE staff specially Bernice Ison and Debie Fraser for their friendship and help.

I want to thank my family, specially my parents. I am thankful for their support, unconditional love, and faith in me. They have always encouraged me to strive for excellence.

Finally, I would like to give my special thanks to my wife, Sepideh, for her love, patience, support, understanding, and above all, for always being by my side through this journey of my life.

Contents

List of Figures

List of Abbreviations

Abbreviation	**Description**
CODEC	Coder-Decoder
CSF	Contrast Sensitivity Function
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
DWT	Discrete Wavelet Transform
ECDQ	Entropy Coded Dithered Quantizer
GOP	Group Of Pictures
IDCT	Inverse Discrete Cosine Transform
MC	Motion Compensation
MD	Multiple Description
MV	Motion Vector
MDC	Multiple Description Coding
MDLVQ	Multiple Description Lattice Vector Quantization
MDSQ	Multiple Description Scalar Quantization
MDVQ	Multiple Description Vector Quantization
PCM	Pulse Amplitude Modulation
PCT	Pairwise Correlating Transform
PSNR	Peak Signal to Noise Ratio
RLC	Run-Length Coding
RRD	Redundancy-Rate Distortion
VQ	Vector Quantization
VQM	Video Quality Metric
UEP	Unequal Error Protection

Chapter 1

Introduction

1.1 Introduction

The transmission of digital information over communication channels has become very challenging with the increasing demand of services such as multimedia applications. These applications naturally require significant amount of bandwidth and storage. This has been a strong motivation for development of efficient source coding techniques in order to reduce the required memory and bandwidth. Source coding methods can be classified as either lossless or lossy. The goal of both types of source coding is to encode the source into a compressed digital representation that can be used for transmission or storage.

In lossless source coding the aim is to generate a compact representation of the source that can be decoded to reconstruct the original signal without error. However, since the lossless coding of sources such as speech or video still requires significant bandwidth and memory, lossy source coding is widely used instead. Although lossy source coding does not generate the exact reconstruction of the original source, an acceptable approximation can be generated that achieves higher compression ratio than the lossless techniques. The performance of lossy source coding mainly depends on the data itself. In general, a data with more inherent redundancy results in better compression.

Now consider a source coding problem where we wish to send source information over a communication channel with non-zero probability of failure. In other words, the encoder sends packets of data through the channel, and at the receiver, we either receive the packet intact or it is lost. To provide more reliability for such a data communication scenario, we may provide a diversity system with more than one channel for packet delivery. In this case, the probability of all channels breaking down is much less than that in the one-channel case. We are hoping that in the case of multiple channels at least some packets are received by the decoders and, as a result, we can better avoid the situation where the receiver does not receive anything at all. Consequently, we need to generate more than one description of the source packet, and we refer to this as the multiple description coding (MDC) problem. Now consider a MDC system with two channels. If we send the same information over each channel and if both descriptions get through, then half of the received information has no value. This implies the importance of sending different information over each channel. Individual descriptions in this case must be generated such that if one channel breaks down and we receive only one description at the receiver side, the received description is sufficient to satisfy a minimum fidelity. However, if both descriptions are received at the receiver, the information received from one channel can be used to refine the information from the other channel in order to achieve a higher fidelity.

This MDC problem and its information theoretic issues were first posed by Gersho, Witsenhausen, Wolf, Wyner, Ziv, and Ozarow (see [1]) at the 1979 IEEE Information Theory Workshop. However, the practical motivations of multiple description coding goes back to 1970s. Like so much of communication technology, MDC was invented at Bell Laboratories for speech communication over the telephone network. The telephone system needs to be reliable. But outages of transmission links are inevitable. Thus, a mechanism is required to manage outage and achieve high reliability. In this context of speech coding, Jayant and Christensen [2], [3] developed a technique for combating speech quality degradation due to packet losses. Two different packets are created by even and odd sampling of a pulse amplitude modulation (PCM) bitstream. For instance, if only an even sample packet is lost, data contained in the odd packet is used to estimate the missing samples.

Digital video applications have become increasingly popular, and the demand of video data transmitted over networks is rapidly increasing. Several international standards, such as MPEG and H.263, have been developed to meet this increasing demand. These standards achieve a high compression ratio, and consequently, the encoded bitstream becomes more vulnerable to transmission errors and packet loss. Several techniques have been developed so far to protect the video signal against transmission errors. For instance, error concealment methods [4, 5] attempt to conceal the erroneous or lost blocks by making use of received information in the adjacent blocks. However, in the context of packet networks, the lost packets could result in a loss of synchronization, error propagation in the video signal, and severe data damage. In such cases, all the blocks are corrupted until the next synchronization codeword. This makes the error concealment technique inefficient, since no information is available from the adjacent blocks. Scalable video coding is another approach that encodes the video signal using two layers [6,7,8]. The base-layer bitstream consists of the essential information such as the motion vectors, and is transmitted with a high level of priority. The second-layer bitstream contains the less important information, which may be ignored if it is lost. The problem with this approach is that the base-layer can also be lost in a highly congested network and cause severe degradation in the decoded video quality.

Multiple Description (MD) video coding is an alternative approach to enhance the robustness of the transmitted video signal, and combat the effect of packet loss. Similar to a general MDC scheme, the video signal is encoded into two correlated descriptions, and then transmitted over separate channels to the decoder. If both descriptions are received, the decoder provides a high-quality reconstruction of the original video. On the other hand, if one of the descriptions is lost during the transmission, the decoder estimates it from the received description, and then provides a lower but acceptable quality reconstruction.

1.2 Contributions

The contributions of this thesis are as follows:

1. Motivated by the successive quantization scheme in [9] which achieves the simplified El Gamal-Cover rate-distortion region, we propose two multiple description quantization schemes with an iterative method to jointly design the codebooks to minimize a Lagrangian cost function. This Lagrangian function includes central distortion and target side distortions imposed by the design problem constraints. We find optimality conditions in order to obtain a joint codebook design algorithm for the two proposed MD quantization schemes. The

proposed schemes are much less complex than optimum MD vector quantizer and our simulation results reveal that the proposed methods asymptotically (in dimension) perform very closely to the optimum MD quantizer.

2. We also propose a multiple description video coding technique that uses visual distortion criteria to split a one-layer stream generated by a standard video coder into two correlated streams. This method employs two parameters, Smooth and Edge, to evaluate human visual perception characteristics and calculate the smoothness and the edge features of each DCT block in the DCT domain. It is well known that the motion-compensated prediction can effectively exploit the temporal correlation between video frames, and is a fundamental component in video coding standards. Thus, we herein take the straightforward strategy of duplicating the header information and motion vectors in both descriptions. We then apply a simple smoothness and edge detection technique in order to determine the amount of perceptually acceptable redundancy of each DCT block, and consequently split the DCT coefficients between descriptions.

1.3 Organization of Thesis

The rest of this thesis is organized as follows. As with other information theory problems, the first problem that needs to be addressed is the underlying theoretical limits and fundamental bounds. Thus, Chapter 2 presents the so-called El-Gamal Cover (EGC) achievable region, the exact rate-distortion region for memoryless Gaussian source and squared error distortion measure, and also the generalization of the multiple description problem to the case of more than two channels. We then review some practical MD scalar and lattice vector quantization techniques in this chapter. Multiple description transform coding, which is a different type of MD practical coding, is discussed in Chapter 2 as well.

Chapter 3 presents our proposed MD quantization schemes, the MDVQ-WSC scheme with central codebook as the weighted sum of the two side codebooks, and the MDVQ-OC scheme with optimum central codebook. These schemes use an iterative method to jointly design the codebooks to minimizes a Lagrangian cost function that includes central and side distortions and rate constraints. We then find an optimal solution for the parameters of the proposed quantization schemes. Simulation results and complexity comparisons are provided in this chapter as well.

Chapter 4 reviews some basic video coding concepts such as video signal format, spatial and temporal redundancy, video assessment metrics, and MPEG standard. We then briefly review multiple description coding concepts in video applications in this chapter.

Chapter 5 describes our proposed scheme for multiple description coding of video. We first introduce an algorithm in order to calculate the smoothness and edge characteristics of video DCT blocks. We then present our proposed method of MD video coding which uses visual distortion criteria to split a one-layer bitstream generated by a standard video coder into two correlated streams. Simulation results and performance of our proposed method are also provided in this chapter.

We finally conclude the thesis and discuss potential future work in Chapter 6.

Chapter 2

Multiple Description Coding: Information Theoretic Aspects and Coding Methods

2.1 Multiple Description Problem Definition and Information Theoretic Aspects

Figure 2.1 depicts a general multiple description coding scenario. An encoder is given the k-dimensional random source vector $\mathbf{X}$ to communicate to three decoders over two erasure channels. One decoder (the central decoder) receives information sent over both channels while the remaining two decoders (the side decoders) receive information only over their respective channels. In response to a source vector $\mathbf{X}$, the encoder generates two codewords (indices) $f_1(\mathbf{X})$ and $f_2(\mathbf{X})$ at rates

$$R_i = \frac{1}{k} \log |f_i(\cdot)|, \quad i = 1, 2$$

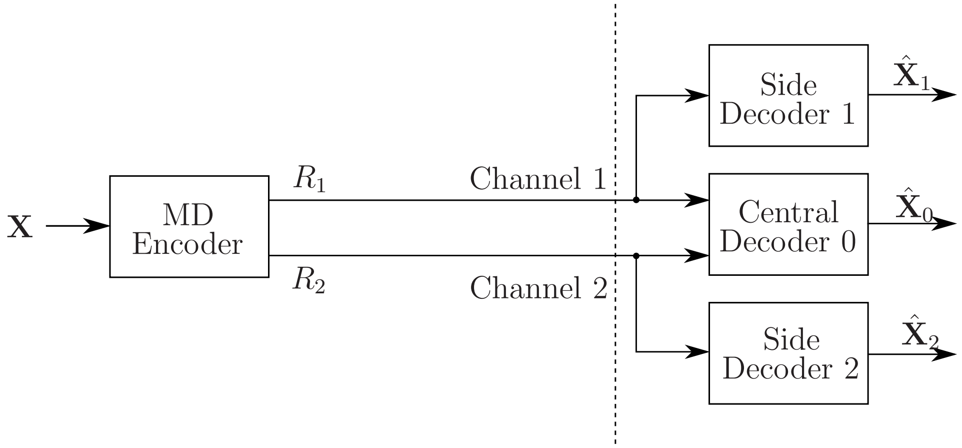


Figure 2.1: General MD Coder Scheme with two channels.

where $|f_i(\cdot)|$ denotes the size of the code $f_i(\cdot)$, and transmits codeword $f_i(\mathbf{X})$ through subchannel $i,\ i = 1, 2$. The two individual marginal (side) receivers and the combined central receiver then generate reconstructions $\hat{\mathbf{X}}_1$, $\hat{\mathbf{X}}_2$, and $\hat{\mathbf{X}}_0$, respectively, using the decoding functions g_1, g_2, and g_0:

$$\begin{aligned}\hat{\mathbf{X}}_1 &= g_1(f_1(\mathbf{X})),\\ \hat{\mathbf{X}}_2 &= g_2(f_2(\mathbf{X})),\\ \hat{\mathbf{X}}_0 &= g_0(f_1(\mathbf{X}), f_2(\mathbf{X})).\end{aligned}$$

Here receiving $\hat{\mathbf{X}}_1$ (resp. $\hat{\mathbf{X}}_2$) models the situation in which the packet corresponding to $f_2(\mathbf{X})$ (resp. $f_1(\mathbf{X})$) is lost, while if both packets are received, the decoder can reproduce $\hat{\mathbf{X}}_0$.

Distortions D_0, D_1, and D_2 are incurred at the respective decoders according to

$$D_i = \frac{1}{k}\sum_{j=1}^{k} E[d_i(X_j, \hat{X}_{i,j})], \qquad i = 0, 1, 2, \tag{2.1}$$

where $d_i(\cdot,\cdot)$ is a nonnegative real-valued distortion measure, $\mathbf{X} = (X_1, X_2, \cdots, X_k)$,

$\hat{\mathbf{X}}_i = (\hat{X}_{i,1}, \hat{X}_{i,2}, \cdots, \hat{X}_{i,k})$, $i = 0, 1, 2$, and expectation is performed over the distribution of $\mathbf{X}$.

The case of only one receiver is the classical rate-distortion problem [10]. For a stationary and memoryless source, corresponding to the source statistics and the distortion measure, the rate-distortion function is defined by

$$R(d) = \inf_{p(\hat{x}|x): E[d(X,\hat{X})] \leq d} I(X; \hat{X}), \tag{2.2}$$

where minimization is over all conditional distributions $p(\hat{x}|x)$ for which the joint distribution $p(x, \hat{x})$ satisfies the distortion constraint, and $I(X; \hat{X})$ is the mutual information between X and $\hat{X}$ defined as follows (in case the source and reproduction alphabet are discrete)

$$\begin{aligned} I(X; \hat{X}) &\triangleq \sum_{x \in \mathcal{X}} \sum_{\hat{x} \in \hat{\mathcal{X}}} p(x, \hat{x}) \log_2 \frac{p(\hat{x}|x)}{p(\hat{x})} \\ &= \sum_{x \in \mathcal{X}} \sum_{\hat{x} \in \hat{\mathcal{X}}} p(x, \hat{x}) \log_2 \frac{p(x|\hat{x})}{p(x)} \qquad (2.3) \end{aligned}$$

where $\mathcal{X}$ and $\hat{\mathcal{X}}$ are alphabets of random variables X and $\hat{X}$, respectively. The mutual information function is closely related to entropy. The entropy $H(X)$ of a discrete random variable X with alphabet $\mathcal{X}$ and distribution $p(x)$; $x \in \mathcal{X}$ is defined as [11]

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

and the conditional entropy of $\hat{X}$ given X is

$$H(\hat{X}|X) = -\sum_{x \in \mathcal{X}} \sum_{\hat{x} \in \hat{\mathcal{X}}} p(x, \hat{x}) \log_2 p(x|\hat{x}).$$

Then we can rewrite (2.3) as

$$I(X; \hat{X}) = H(\hat{X}) - H(\hat{X}|X) = H(X) - H(X|\hat{X}).$$

Closed form expression for rate-distortion function for arbitrary sources and distortion measure do not exist. However, for a memoryless Gaussian source with variance σ_X^2 and squared error distortion measure, the rate-distortion function in bits per source sample is [10]

$$R(d) = \begin{cases} \frac{1}{2}\log\frac{\sigma_X^2}{d} & \text{if } 0 < d < \sigma_X^2 \\ 0 & \text{if } d \geq \sigma_X^2. \end{cases} \tag{2.4}$$

Equivalently, the distortion-rate function is defined as

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

where minimization is over all $p(\hat{x}|x)$ for which the joint distribution $p(x,\hat{x})$ satisfies the rate constraint.

A natural problem for the network of Figure 2.1 is to characterize the set of achievable quintuples $(r_1, r_2, d_0, d_1, d_2)$. Specifically, $(r_1, r_2, d_0, d_1, d_2)$ is achievable if for sufficiently large k there exist encoding and decoding mappings such that

$$R_i \leq r_i, \quad i = 1, 2 \tag{2.5}$$

$$D_i \leq d_i, \quad i = 0, 1, 2. \tag{2.6}$$

Decoder 1 receives R_1 bits and hence cannot have distortion less than $D(R_1)$, where $D(\cdot)$ is the distortion-rate function of the source. By a similar argument for the other two decoders and using rate-distortion functions instead of distortion-rate functions gives the following bounds on the achievable region:

$$R_1 + R_2 \geq R(D_0) \tag{2.7}$$

$$R_1 \geq R(D_1) \tag{2.8}$$

$$R_2 \geq R(D_2) \tag{2.9}$$

As mentioned earlier, because of the conflict in making the individual and joint descriptions good, the above bounds are usually loose. In general, the rate-distortion region $R(\mathbf{D})$ for distortion $\mathbf{D} = (D_0, D_1, D_2)$ is the closure of the set of achievable rate pairs (R_1, R_2) inducing distortion $\leq \mathbf{D}$. Then, an achievable rate region is any subset of the rate-distortion region.

2.1.1 An Achievable Rate-Distortion Region for Multiple Description

The first general result on achievable rate-distortion region was derived by El Gamal and Cover [1]:

Theorem 2.1. Achievable rates for multiple description coding (EGC region) *Let $X_1, X_2, \cdots$ be a sequence of i.i.d. finite alphabet random variables drawn according to a probability mass function $p(x)$. Let $d_i(\cdot,\cdot)$ be bounded. An achievable rate region for distortions (D_0, D_1, D_2) is given by the convex hull of all (R_1, R_2) such that*

$$\begin{aligned} R_1 &> I(X;\hat{X}_1), \\ R_2 &> I(X;\hat{X}_2), \\ R_1 + R_2 &> I(X;\hat{X}_0,\hat{X}_1,\hat{X}_2) + I(\hat{X}_1;\hat{X}_2), \end{aligned}$$

for some probability mass function

$$p(x,\hat{x}_0,\hat{x}_1,\hat{x}_2) = p(x)p(\hat{x}_0,\hat{x}_1,\hat{x}_2|x), \tag{2.10}$$

such that

$$\begin{aligned} D_0 &\geq E[d_0(X, \hat{X}_0)], \\ D_1 &\geq E[d_1(X, \hat{X}_1)], \\ D_2 &\geq E[d_2(X, \hat{X}_2)]. \end{aligned}$$

To use this theorem, one chooses the distribution of the auxiliary random variables $\hat{X}_0, \hat{X}_1$, and $\hat{X}_2$ (jointly with X) to satisfy (2.10) and the distortion criteria. Each set of auxiliary random variables yields an achievable rate region. The convex hull of these regions is also achievable.

2.1.2 R-D Bound for Memoryless Gaussian Source with Squared Error Distortion

The achievable rate-distortion region is completely known only for a memoryless Gaussian source, and the distortion region was first found by Ozarow [12]. The equivalent Gaussian rate region [1, 13] is given in the following theorem [9]:

Theorem 2.2. *Let X be an i.i.d. zero-mean Gaussian process with variance σ_X^2, and $d(\cdot, \cdot)$ be the squared error distortion measure. For this Gaussian source, the quintuple $(R_1, R_2, D_0, D_2, D_2)$ is achievable if and only if*

$$R_i \geq \frac{1}{2} \log \frac{\sigma_X^2}{D_i}, \quad i = 1, 2, \tag{2.11}$$

$$R_1 + R_2 \geq \frac{1}{2} \log \frac{\sigma_X^2}{D_0} + \frac{1}{2} \log \psi(D_1, D_2, D_0), \tag{2.12}$$

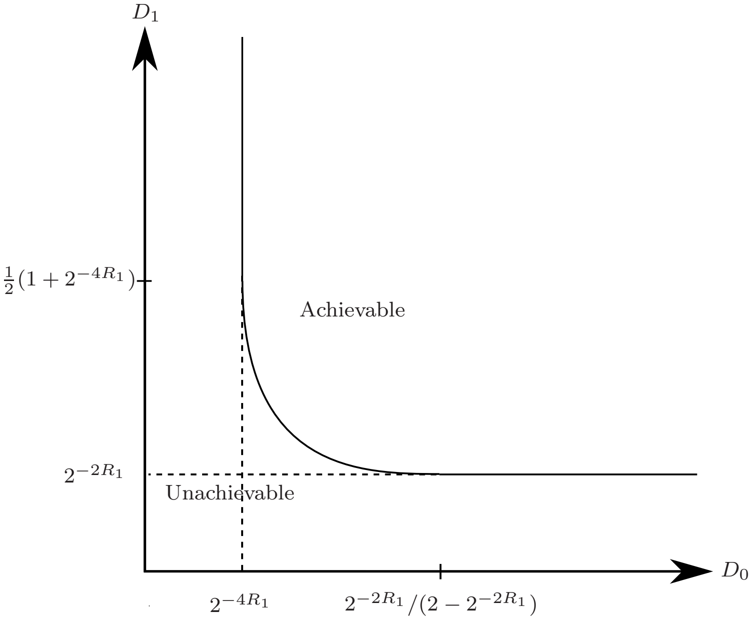


Figure 2.2: Achievable central and side distortions for multiple description coding of a memoryless Gaussian source with squared-error distortion. $D_1 = D_2$ and $R_1 = R_2$.

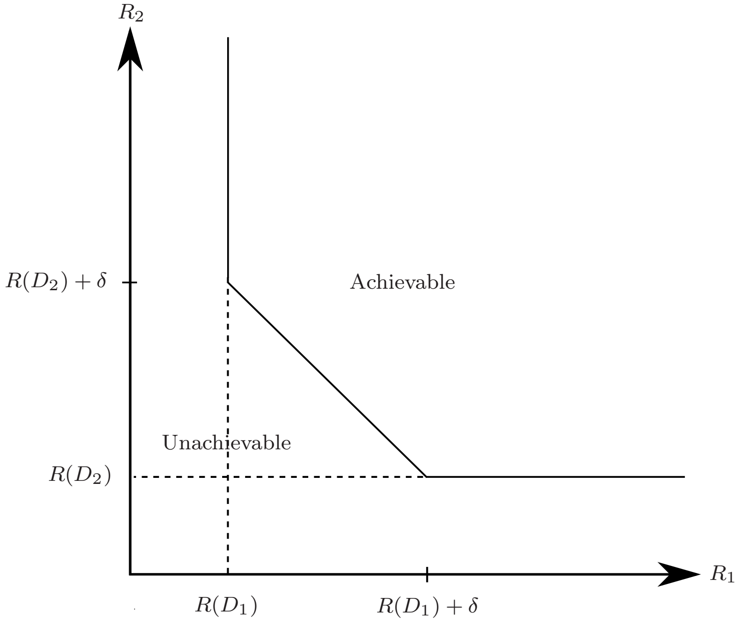


Figure 2.3: Achievable rates for multiple description coding of a unit variance memoryless Gaussian source with squared-error distortion. The minimum excess rate δ is a function of (D_0, D_1, D_2) and may be zero.

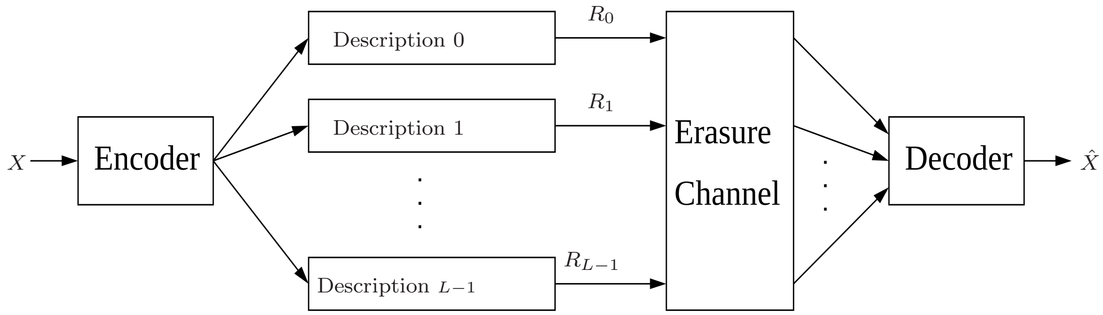


Figure 2.4: General L-channel system. Descriptions are encoded at rate R_i, $i = 0, \ldots, L-1$. The erasure channel either transmits the ith description without error or not at all.

where

$$\psi(D_1, D_2, D_0) = \begin{cases} 1, & D_0 < D_1 + D_2 - \sigma_X^2 \\ \frac{\sigma_X^2 D_0}{D_1 D_2}, & D_0 > \left(\frac{1}{D_1} + \frac{1}{D_2} - \frac{1}{\sigma_X^2}\right)^{-1} \\ \frac{(\sigma_X^2 - D_0)^2}{(\sigma_X^2 - D_0)^2 - [\sqrt{(\sigma_X^2 - D_1)(\sigma_X^2 - D_2)} - \sqrt{(D_1 - D_0)(D_2 - D_0)}]^2}, & \textit{otherwise.} \end{cases}$$

The region of achievable (D_0, D_1) pairs when $D_1 = D_2$ and $R_1 = R_2$ is shown in Figure 2.2. Achievable rate region for Gaussian source given D_0, D_1, and D_2 is shown in Figure 2.3, where δ is a function of (D_0, D_1, D_2) that represents the minimum excess (sum) rate necessary to achieve these distortions.

2.1.3 L-Channel Multiple Description Coding

The generalization of the multiple description problem to L-channels and $(2^L - 1)$-receivers is shown in Figure 2.4. In the L-channel and $(2^L - 1)$-receiver MD coding problem, there are L encoding functions $f_l(\cdot) : \mathcal{X} \rightarrow \{1, \cdots, M_l\}, l \in \mathcal{L} = \{1, \cdots, L\}$. The L descriptions $j_l = f_l(\mathbf{X}), l \in \mathcal{L}$ are sent over L channels. If we do not consider the case where all the packets are lost, and we also assume that each description is either received intact or is lost, then there are total $2^L - 1$ possible combinations of received packets which requires $2^L - 1$ receivers, denoted as $g_{\mathcal{K}}(\cdot)$, for each $\mathcal{K} \subset \mathcal{L}$ and $\mathcal{K} \neq \emptyset$. The reconstruction at each receiver is represented by

$$\hat{\mathbf{X}}_{\mathcal{K}} = g_{\mathcal{K}}(j_k : k \in \mathcal{K}).$$

Let $D_{\mathcal{K}}$ be the average per symbol distortion of the decoder $g_{\mathcal{K}}$. The rate-distortion region of the L-channel $(2^L - 1)$-receiver MD coding problem is composed of all achievable length-$(L + (2^L - 1))$ vectors $(R_l : l \in \mathcal{L}; \quad D_{\mathcal{K}} : \mathcal{K} \subset \mathcal{L}, \mathcal{K} \neq \emptyset)$. We introduce the notation $R_{\mathcal{K}} = \sum_{k \in \mathcal{K}} R_k$.

The following theorem in [14] presents an achievable rate-distortion region for the L-channel $(2^L - 1)$-receiver MD coding problem:

Theorem 2.3. **L-Channel MD Coding: Achievable Rate-Distortion Region)** *Let $X_{2^{\mathcal{L}}}$ be any set of 2^L random variables jointly distributed with X, where X_0 takes value in some finite alphabet $\mathcal{X}_0$ and each $\mathcal{X}_{\mathcal{K}}$ takes value in the reproduction alphabet $\hat{\mathcal{X}}_{\mathcal{K}}, \mathcal{K} \neq \emptyset$. Then the rate-distortion region contains the rates and distortions satisfying*

$$\begin{aligned} D_{\mathcal{K}} &\geq E[D_{\mathcal{K}}(X, X_{\mathcal{K}})], \\ R_{\mathcal{K}} &\geq (|\mathcal{K}| - 1)I(X; X_0) - H(X_{(2^{\mathcal{K}})}|X) + \sum_{\mathcal{M} \subset \mathcal{K}} H(X_{\mathcal{M}}|X_{(2^{\mathcal{M}} - \{\mathcal{M}\})}). \end{aligned}$$

2.2 Practical Multiple Description Schemes: Scalar and Lattice Quantization

All of the results discussed so far are non-constructive in the sense that they only provide bounds to performance when very long source sequences are coded.

When it comes to practical MD coding, the main problem is how to generate descriptions such that the resulting rates, incurred distortions and complexity satisfy some prescribed conditions. First consider the simplest ways to produce MDs. One is to partition the source data into several sets and then compress them independently to produce descriptions. Interpolation is then used to estimate the descriptions. The separation can be into odd- and even-numbered samples or a similar separation for more than two descriptions or multidimensional data. If a description is lost, the corresponding side decoder relies on the correlation between lost and received descriptions in order to estimate the lost one. A trivial way to send two descriptions is to send the same description twice. The description can be produced with the best available compression technique. When only one description is received, the performance is as good as possible; however, no advantage is gained from receiving both descriptions. A more flexible approach, which belongs to a family of joint source-channel coding techniques called unequal error protection (UEP), repeats only some fraction of data [15]. It would be advantageous if the repeated data were the most important, and thus this type of fractional repetition is naturally matched to progressive source coding.

We dedicate the rest of this section to introducing scalar and lattice vector quantization schemes which are well studied in the literature. We also discuss a very recently developed technique called multiple description quantization via Gram-Schmidt orthogonalization, and also MDC with dithered Delta-Sigma modulation.

2.2.1 MD Scalar Quantization

Multiple description scalar quantization (MDSQ) was first introduced by Vaishampayan in [16]. Conceptually, MDSQ can be regarded as the use of a pair of independent scalar quantizers to give two descriptions of a scalar source sample. As in all multiple description coding schemes, the design challenge is to simultaneously provide good individual descriptions and a good joint description. MD scalar quantization is flexible in that it allows a designer to choose the relative importance of the central distortion and each side distortion.

The scalar quantizer is an encoder map $f : \mathcal{R} \rightarrow \{1, 2, \cdots, M\}$ whose output is a codeword index and a decoder map $g : \{1, 2, \cdots, M\} \rightarrow \mathcal{R}$. The encoder partitions the real line $\mathcal{R}$ into M cells. The partition is represented by $A = \{A_1, A_2, \cdots, A_M\}$ where $A_i = \{x : f(x) = i\}, i = 1, 2, \cdots, M$. A scalar quantizer is completely described by its partition and its codebook. Let $d(x, \hat{x})$ be the distortion between x and $\hat{x}$. The objective of scalar quantizer design is to select A and g so as to minimize $E[d(X, \hat{X})]$.

Now assume that a diversity system has two channels, capable of transmitting information reliably at rates of R_1 and R_2 bits/source sample (bpss), respectively. Each channel may either be in a working or non-working state; this is not known in the encoder. The encoder sends a different description over each channel. Given the state of each channel, the source decoder forms the best estimate of the source output from the available data.

An (M_1, M_2)-level MDSQ maps the source sample x to the reconstruction levels $\hat{x}_0, \hat{x}_1$, and $\hat{x}_2$ that take values in the codebooks, $\hat{\mathcal{X}}_0 = \{\hat{x}_{0,ij} \ : \ (i, j) \in C\}, \hat{\mathcal{X}}_1 = \{\hat{x}_{1,i} \ : \ i \in I_1\}, \hat{\mathcal{X}}_2 = \{\hat{x}_{2,j} \ : \ j \in I_2\}$, respectively, where $I_1 = \{1, 2, \cdots, M_1\}, I_2 = \{1, 2, \cdots, M_2\}$ and C is a subset of $I_1 \times I_2$. Let $N = |C|$. An MDSQ can be broken

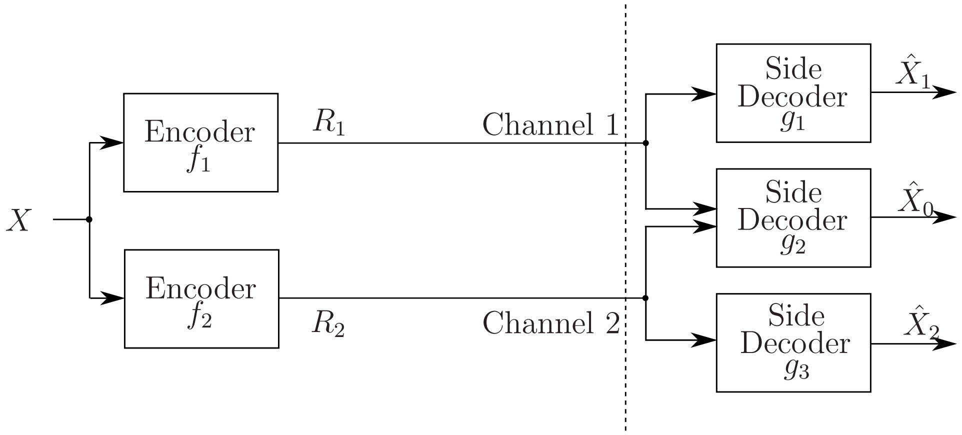
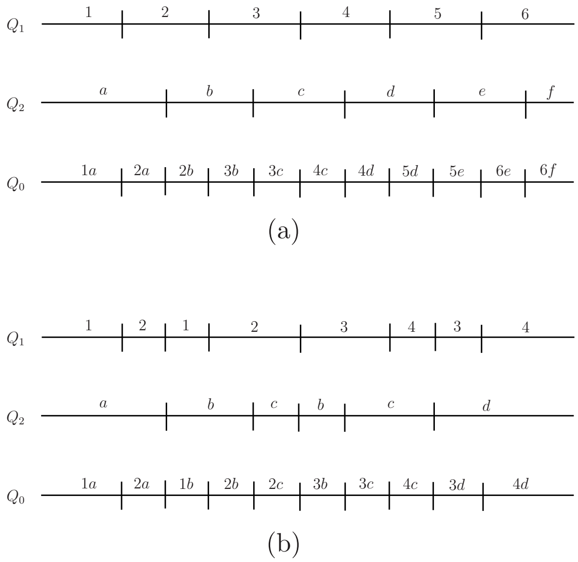


Figure 2.5: Multiple description scalar quantizer with two channels.

up into two side encoders, $f_1 : \mathcal{R} \rightarrow I_1$ and $f_2 : \mathcal{R} \rightarrow I_2$, which select the indexes i and j , respectively, and three decoders, $g_0 : C \rightarrow \mathcal{R}$ (central decoder), $g_1 : I_1 \rightarrow \mathcal{R}$ and $g_2 : I_2 \rightarrow \mathcal{R}$ (side decoders), whose outputs are the reconstruction levels with indexes ij, i, and j from the codebooks $\hat{\mathcal{X}}_0$, $\hat{\mathcal{X}}_1$, and $\hat{\mathcal{X}}_2$, respectively. The two encoders induce a partition $A = \{A_{ij}, (i, j) \in C\}$ on $\mathcal{R}$, where $A_{ij} = \{x : f_1(x) = i, f_2(x) = j\}$. The MDSQ is completely described by $A, \hat{\mathcal{X}}_0, \hat{\mathcal{X}}_1,$ and $\hat{\mathcal{X}}_2$. We refer to **f**$= (f_1, f_2)$ as the encoder, **g**$= (g_0, g_1, g_2)$ as the decoder, to A as the central partition, and to the elements of A as the central cells.

An MDSQ, as used over a diversity system, is shown in Figure 2.5. The outputs of the side encoders, i.e., the indexes i and j , are transmitted over channel 1 and channel 2, respectively. If both indexes are received, the central decoder is used to reconstruct the source sample. On the other hand, if only i (j) is received, then side decoder g_1 (g_2) is used to reconstruct the sample.

The simplest example is to have scalar quantizers with nested thresholds, as shown in Figure 2.6(a). Each quantizer, Q_1 and Q_2, outputs the quantization index of the same sample, for example, $Q_1(x) = k_1$ and $Q_2(x) = k_2$. Given $Q_i = k_i$, the


Q_1
1 2 3 4 5 6
Q_2
a b c d e f
Q_0
$1a$ $2a$ $2b$ $3b$ $3c$ $4c$ $4d$ $5d$ $5e$ $6e$ $6f$
Q_1
1 2 1 2 3 4 3 4
Q_2
a b c b c d
Q_0
$1a$ $2a$ $1b$ $2b$ $2c$ $3b$ $3c$ $4c$ $3d$ $4d$

(a)

(b)

Figure 2.6: Examples of MD scalar quantizers. (a) The simplest form of MDSQ. (b) An MDSQ based on Vaishampayan's quantizer index assignment.

reconstruction should be the centroid of the cell $Q_i^{-1}(k_i)$. The central decoder has both indices k_1 and k_2, and thus reconstructs to the centroid of the intersection cell $Q_1^{-1}(k_1) \cap Q_2^{-1}(k_2)$. In the example, the intersection cells are about half as big as the individual quantizer cells, so the central distortion is about a quarter of the side distortions. Asymptotically, if the side rates are $R_1 = R_2 = R$, then D_0, D_1, and D_2 are all $\mathcal{O}(2^{-2R})$, where $\mathcal{O}(\cdot)$ here indicates the asymptotic behavior of the distortions. This is optimal decay for D_1 and D_2, but far from optimal for D_0 [16].

Given a desired partition for the central encoder, the MDSQ design problem reduces to the assignment of indices to the individual quantizers. Vaishampayan's main results in [16] are this observation and an idealized index assignment scheme that gives the optimal combined exponential decay rates for the central and side distortions. An MDSQ designed with Vaishampayan's modified nested index assignment

Table 2.1: Another illustration for the example in Figure 2.6(b). Quantizer outputs are assigned to pairs of indices. At the decoder, when both descriptions are received, the indexed produced by π is found by looking up the intersection of the corresponding row and column.

	00	01	10	11
00	1	3		
01	2	4	5	
10		6	7	9
11			8	10

is shown in Figure 2.6(b). It systematically balances the central and side distortions while maintaining optimal joint asymptotic decay of distortion with rate. If the condition $R_1 = R_2 = R$ is kept, then D_0 has decay of $\mathcal{O}(2^{-4R})$. In fact, a better central distortion is achieved with the price of lower side distortions. In other words, for a given number of side levels, the central distortion is smaller, at the cost of higher side distortions, than for an MDSQ as in Figure 2.6(a).

As demonstrated in examples of Figure 2.6, MDSQ can be alternatively viewed as a partition of the real line along with a mapping between partition intervals and ordered pairs of indexes. Thus, the encoder consists of two steps: $\alpha_0 = \alpha \circ \pi$. π is a regular scalar quantizer which partitions real line into intervals. The mapping α takes the index produced by π and generates a pair of indexes i, j. The mapping α is invertible so that the central decoder can recover the output of π. Table 2.1 gives the alternative illustration for the example in Figure 2.6(b). Each quantization index is mapped to a pair of 2-bit binary codewords which are placed in the corresponding first row and first column, respectively. For example, if the output of π , i.e., quantization index, is 6 which corresponds to $3b$ in Figure 2.6(b), then, according to Table 2.1, the output of α will be $(10, 01)$. At the decoder, if both 10 and 01 are received, by looking up the Table 2.1, the central reconstruction level will be known as $3b$. When only a single description (say 01) is received, 01 will be used to decode and provides a coarse reconstruction. Through proper choice of the quantizer π and mapping α, MDSQ with more or less redundancy can be easily designed. The flexibility is one beneficial feature of this techniques.

Vaishampayan in [16] gives several heuristic techniques that likely get close the best possible performance. The basic ideas are to number from upper-left to lower-right and to fill from main diagonal outward.

Additional developments in this area include a design procedure for entropy-constrained codebooks [17], joint optimization of an orthogonal transform and MDSQ [18, 19], and a method for reducing granular distortion [20].

2.2.2 MD Lattice Vector Quantization

One disadvantage of scalar quantization is that it fails to exploit the correlation which exists among source samples. Samples of many sources of information such as speech, image and video are correlated in nature, and this motivates the need for a quantization technique which is able to exploit this inherent redundancy among the source samples and achieve a better performance. Vector quantization (VQ) has been used extensively to address this problem. Another advantage of VQ is that it offers extra freedom in choosing the partition cells whose shape more efficiently fills space [21]. This so-called space-filling property allows us to exploit dimensionality to minimize distortion in a way that is not possible with scalar quantization.

Vector quantization is a source coding technique in which vectors (blocks) of source samples are mapped into a finite set of reproduction points, also called code vectors or codewords. Consider a k-dimensional source vector $\mathbf{X} = (X_1, X_2, \cdots, X_k)$ in Euclidean space $\mathcal{R}^k$ with probability density function $p(\mathbf{x})$. A k-dimensional vector quantizer Q with N reproduction levels is identified by its codebook $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N\}$ containing reproduction levels $\mathbf{c}_i,\ i \in \mathcal{I} \equiv \{1, 2, \cdots, N\}$, and partition cells $S_i,\ i \in \mathcal{I} \equiv \{1, 2, \cdots, N\}$. In fact, each partition cell is associated with one code vector in the codebook such that

$$S_i = \{\mathbf{x} \in \mathcal{R}^k \ : \ Q(\mathbf{x}) = \mathbf{c}_i\}.$$

It follows from the definition of partition cells that

$$\bigcup_i S_i = \mathcal{R}^k \ \text{ and } \ S_i \bigcap S_j = \emptyset \text{ for } i \neq j.$$

The encoder of a vector quantizer maps the source vector to one of the code vector in its codebook, and sends the index of the corresponding code vector over the channel. We can then define the rate of such a fixed-rate vector quantizer in terms of bits per source sample (bpss) as

$$R_i = \frac{1}{k} \log_2 N.$$

The entropy of a vector quantizer is defined as

$$H(Q) = -\sum_{i=1}^{N} p(Q(\mathbf{X}) = \mathbf{c}_i) \log_2 p(Q(\mathbf{X}) = \mathbf{c}_i).$$

It is shown in [22] that per sample rate of a variable-rate quantizer $r(Q)$ is bounded as

$$\frac{1}{k} H(Q) \leq r(Q) < \frac{1}{k} H(Q) + \frac{1}{k}.$$

As a result, in high dimensions, we can simply redefine $r(Q)$ as

$$r(Q) = \frac{1}{k} H(Q).$$

The performance of a vector quantizer is usually measured with its associated distortion. The expected distortion per sample of vector quantization Q is given by

$$\begin{aligned} D(Q) &= \frac{1}{k}E[d(\mathbf{X}, Q(\mathbf{X}))] \\ &= \frac{1}{k}\int_{\mathcal{R}^k} d(\mathbf{x}, Q(\mathbf{x}))p(\mathbf{x})d\mathbf{x} \\ &= \frac{1}{k}\sum_{i=1}^{N}\int_{S_i} d(\mathbf{x}, \mathbf{c}_i)p(\mathbf{x})d\mathbf{x}, \end{aligned} \tag{2.13}$$

where $d(\cdot, \cdot)$ is a distortion measure function. The design problem then becomes how to find the codebook and partition cells such that expected distortion defined in (2.13) is minimized. This problem was first addressed in [23] which led to the following theorems for VQ necessary conditions for optimality.

Theorem 2.4. Nearest Neighbor Condition *For a given codebook $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_n\}$, among all N-level vector quantizers, a nearest neighbor quantizer achieves the minimum distortion where Q is a nearest neighbor VQ if for all $\mathbf{x} \in \mathcal{R}^k$*

$$Q(\mathbf{x}) = \arg\min_{\mathbf{c_i} \in \mathcal{C}} d(\mathbf{x}, \mathbf{c}_i)$$

or equivalently, for all $i \in \mathcal{I} = \{1, 2, \cdots, N\}$

$$S_i \subset \{\mathbf{x} \ : \ d(\mathbf{x}, \mathbf{c}_i) \leq d(\mathbf{x}, \mathbf{c}_j), \ j = 1, 2, \cdots, N\}.$$

Theorem 2.5. Centroid Condition *For a given partition $S_1, S_2, \cdots, S_N$, among all N-level vector quantizers, the quantizer Q with reproduction code vectors*

$$\mathbf{c}_i = \arg\min_{\mathbf{c} \in \mathcal{R}^k} E[d(\mathbf{X}, \mathbf{c})|\mathbf{X} \in S_i], \quad i = 1, 2, \cdots, N$$

has the minimum distortion.

Theorem 2.6. *For the squared error distortion measure, the centroid of S_i is uniquely given by $\mathbf{c}_i = E[\mathbf{X}|\mathbf{X} \in S_i]$.*

General vector quantization is usually very complex, computationally expensive, and requires considerable memory for codebook storage. However, structured vector quantization can significantly reduce the coding complexity, computation, and memory requirement. A simple way to achieve a structured code book is to use a set of output points which lie in a bounded region of a lattice. A lattice in k dimensions is defined by any nonsingular $k \times k$ matrix U so that if $\mathbf{m}$ is any k-dimensional vector of integers, the lattice Λ is the set of all vectors of the form $U\mathbf{m}$ [24]. The columns of U are points of the lattice and all other lattice points are formed by taking a linear combination of these basis vectors with integer-valued coefficients. A lattice quantizer may be defined as a quantizer whose output set $\mathcal{C}$ is a subset of a lattice Λ. For instance, in one-dimensional quantization, the only lattice quantizer is the uniform quantizer and the partition cells are intervals in $\mathcal{R}^1$.

The essential issues of multiple description lattice vector quantization (MDLVQ) with index assignment are the choice of an appropriate lattice and the design of a good index assignment.

For the single-description problem, one of the benefits of vector quantization over scalar quantization is a reduction in granular distortion. This is because in higher dimensions it is possible to construct Voronoi cells that are more "spherical" than the hypercube. To be more specific, for a memoryless source with probability density function p, differential entropy $h(p) < \infty$ defined as $h(p) = -\int_{\mathcal{R}^k} p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x}$, and the squared-error distortion measure, uniform scalar quantization coupled with entropy coding is known to have mean squared error (MSE) $d(R)$ at entropy R bits/sample satisfying [25]

$$\lim_{R\to\infty} d(R)2^{2R} = \frac{2^{2h(p)}}{12} \tag{2.14}$$

whereas if an k-dimensional lattice Λ is used as a codebook, the distortion satisfies

$$\lim_{R\to\infty} d(R)2^{2R} = G(\Lambda)2^{2h(p)}, \tag{2.15}$$

where $G(\Lambda)$ is the normalized second moment of a Voronoi cell of the lattice, defined by

$$G(\Lambda) \overset{def}{=} \frac{\int_{V(0)} \|x\|^2 dx}{\nu^{1+2/k}},$$

where $\nu = |V(0)|$ denotes the k-dimensional volume of the fundamental Voronoi region $V(0)$ of the lattice. In dimensions greater than one, lattices exist for which $G(\Lambda)$ is strictly smaller than 1/12. For example, in 8 dimensions, it is possible to gain 0.66 dB by using the lattice E_8 as compared to uniform scalar quantization [26]. It is also known through a random quantizing argument [27] that quantizers exist for which the product $d(R)2^{2R}$ approaches $2^{2h(p)}/(2\pi e)$ as the rate increases. Furthermore, it follows from rate distortion theory [10] that no smaller value for $d(R)2^{2R}$ can be achieved as $R \to \infty$. The maximum possible gain over entropy-coded scalar quantization is 1.53 dB and lattices provide a useful method for closing this gap.

It has been shown [19] that for any $a \in (0, 1)$ there exist uniform entropy-coded multiple description quantizers such that as $R \to \infty$, the distortions satisfy

$$\begin{aligned} \lim_{R\to\infty} d_0^u(R)2^{2R(1+a)} &= \frac{1}{4}\left(\frac{2^{2h(p)}}{12}\right), \\ \lim_{R\to\infty} d_s^u(R)2^{2R(1-a)} &= \left(\frac{2^{2h(p)}}{12}\right). \end{aligned} \tag{2.16}$$

On the other hand, by using a random quantizer argument it was shown [20] that by encoding vectors of very large block length, it is possible to achieve distortions

$$\begin{aligned} \lim_{R\to\infty} d_0^r(R)2^{2R(1+a)} &= \frac{1}{4}\left(\frac{2^{2h(p)}}{2\pi e}\right), \\ \lim_{R\to\infty} d_s^r(R)2^{2R(1-a)} &= \left(\frac{2^{2h(p)}}{2\pi e}\right). \end{aligned} \tag{2.17}$$

Thus by using multiple description quantization it is possible to simultaneously reduce the two-channel and side-channel granular distortions by 1.53 dB. The main objective of MDLVQ is to give constructions for closing this “1.53 dB” gap and to analyze the resulting performance gains. Servetto, Vaishampayan, and Sloane proposed a lattice index assignment algorithm in [28] that produces equal side distortions from equal-rate channels, and achieves similar performance gain over unconstrained MD vector quantization. Diggavi et al. [29] extended this method to unbalanced descriptions by using two different sublattices and a similar index assignment technique.

2.2.3 MD Quantization Via Gram-Schmidt Orthogonalization

Frank-Dayan and Zamir [30] proposed a class of MD schemes which uses entropy-coded dithered lattice quantizers (ECDQs). Their system employs two independently dithered lattice quantizers as the two side quantizers, with a third dithered lattice quantizer to provide refinement information for the central decoder. This system is not optimal in general, and is only optimal in asymptotically high dimensions for the degenerate cases such as successive refinement and the “no excess marginal-rate” case [30]. The main difficulty lies in generating dependent quantization errors of two side quantizers to simulate the Gaussian multiple description test channel.

Chen et al. [9] have provided a systematic treatment of the El Gamal-Cover (EGC) achievable MD rate-distortion region and shown it can be decomposed into a simplified-EGC (SEGC) region and a superimposed refinement operation. They have shown that any point in the SEGC region can be achieved via a successive quantization scheme along with quantization splitting. For the quadratic Gaussian case, the MD rate-distortion region is the same as the SEGC region, and the proposed scheme employs the Gram-Schmidt orthogonalization method. Therefore, they use single-description ECDQs with independent subtractive dithers as building blocks for this MD coding scheme to avoid the difficulty of generating dependent quantization errors.

The proposed scheme in [9] is different from those in [30] in the sense that it can achieve the whole Gaussian MD rate-distortion region as the dimension of the optimal lattice quantizers becomes large. It is also considerably simpler than MDSQ and MDLVQ because it avoids the index assignment problem and relies on the lattice structure. The rest of this section briefly illustrates the proposed scheme in [9] and a geometric interpretation of the case of scalar MD quantization.

Entropy-Coded Dithered Quantization

Here we provide some basic definitions and properties of ECDQ from [30].

A k-dimensional lattice quantizer is formed from a lattice Λ. The quantizer $Q_k(\cdot)$ maps each vector $\mathbf{x} \in \mathcal{R}^k$ into the lattice point $\mathbf{l}_i \in \Lambda$ that is nearest to $\mathbf{x}$. The region of all k-vectors mapped into a lattice point $\mathbf{l}_i \in \Lambda$ form the Voronoi region $V(\mathbf{l}_i)$. The dither $\mathbf{Z}$ is a k-dimensional random vector, independent of the source, and uniformly distributed over the basic cell V_0 of the lattice which is the Voronoi region of the lattice point $\mathbf{0}$. The dither vector is assumed to be available to both the encoder and the decoder. The normalized second moment G_k of the lattice characterizes the second moment of the dither vector

$$\frac{1}{k}E\left[||\mathbf{Z}||^2\right] = G_k \nu^{2/k},$$

where ν denotes the volume of V_0. Both the entropy encoder and the decoder are conditioned on the dither sample $\mathbf{Z}$; furthermore, the entropy coder is assumed to be ideal. The lattice quantizer with dither represents the source vector $\mathbf{x}$ by the vector

$\mathbf{u} = Q_k(\mathbf{x}+\mathbf{z}) - \mathbf{z}$, where $\mathbf{z}$ is the sample of the random vector $\mathbf{Z}$. The resulting properties of the ECDQ are as follows.

1. The quantization error $\mathbf{U}-\mathbf{X}$ is independent of $\mathbf{X}$ and is distributed as $-\mathbf{Z}$. In particular, the mean-squared quantization error is given by the second moment of the dither, independently of the source distribution, i.e.,

$$\frac{1}{k}E||\mathbf{U}-\mathbf{X}||^2 = \frac{1}{k}E||\mathbf{Z}||^2 = G_k \nu^{2/k}.$$

2. Since the dither $\mathbf{Z}$ is known to the encoder and decoder, the coding rate of ECDQ is given by

$$H(Q_k|\mathbf{Z}) = H(Q_k(\mathbf{X}+\mathbf{Z}) - \mathbf{Z}|\mathbf{Z}) = H(Q_k(\mathbf{X}+\mathbf{Z})|\mathbf{Z}). \qquad (2.18)$$

It is proved in [31] that $H(Q_k(\mathbf{X}+\mathbf{Z})|\mathbf{Z}) = H(\mathbf{Y}) - H(\mathbf{N})$, where $\mathbf{N}$ is a random variable with the same probability density function as $-\mathbf{Z}$, and $\mathbf{Y} = \mathbf{X}+\mathbf{N}$. On the other hand, $H(\mathbf{N})$ can be alternatively expressed as $H(\mathbf{N}) = H(\mathbf{X}+\mathbf{N}|\mathbf{X}) = H(\mathbf{Y}|\mathbf{X})$. Thus (2.18) can be rewritten as

$$\begin{aligned} H(Q_k|\mathbf{Z}) &= H(\mathbf{Y}) - H(\mathbf{N}) \\ &= H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}) \\ &= I(\mathbf{X};\mathbf{Y}). \qquad (2.19) \end{aligned}$$

The equation (2.19) indicates that the coding rate of the ECDQ is equal to the mutual information between the input and output of an additive noise channel $\mathbf{Y} = \mathbf{X}+\mathbf{N}$, where $\mathbf{N}$, the channel's noise, has the same probability density function as $-\mathbf{Z}$.

3. For optimal lattice quantizers, i.e., lattice quantizers with the minimal normalized second moment G_k, the autocorrelation of the quantizer noise is "white" [32]

, i.e., $E\mathbf{Z}\mathbf{Z}^T = \sigma^2 I_k$ where I_k is the $k \times k$ identity matrix, $\sigma^2 = G_k^{opt}\nu^{2/k}$ is the second moment of the lattice, and

$$G_k^{opt} = \min_{Q_k(\cdot)} \frac{\int_{V_0} ||\mathbf{x}||^2 \mathrm{d}\mathbf{x}}{k\nu^{1+\frac{2}{k}}}$$

is the minimal normalized second moment of a k-dimensional lattice. Therefore, the samples of the quantization noise are uncorrelated and have the same power.

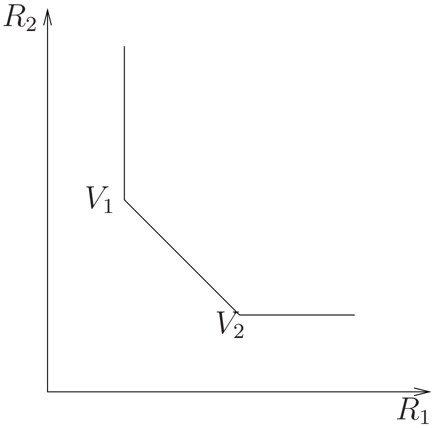


Figure 2.7: The shape of $\mathcal{R}(U_1, U_2)$.

Successive Quantization for a Memoryless Gaussian Source

The Gaussian achievable rate region, as derived in Theorem 2.2, is depicted in Figure 2.7. The coordinates of the vertices V_1 and V_2 can be computed as follows [9]

$$R_1(V_1) = \frac{1}{2}\log\frac{\sigma_X^2}{D_1}, \qquad R_2(V_1) = \frac{1}{2}\log\frac{D_1}{D_0} + \frac{1}{2}\log\psi(D_1, D_2, D_0)$$

and

$$R_1(V_2) = \frac{1}{2}\log\frac{D_2}{D_0} + \frac{1}{2}\log\psi(D_1, D_2, D_0), \qquad R_2(V_2) = \frac{1}{2}\log\frac{\sigma_X^2}{D_2}.$$

The objective is now to achieve the corner points V_1 and V_2. Then, an arbitrary rate pair on the dominant face (lower bound) of the rate region can be achieved by timesharing of coding schemes that achieve the two corner points.

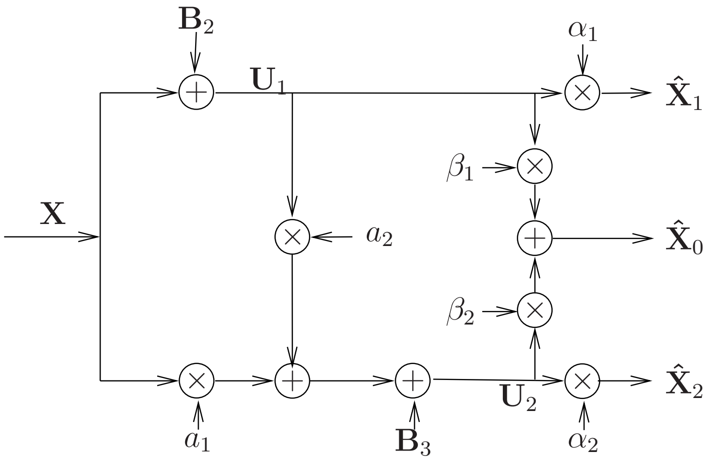


Figure 2.8: MD successive quantization scheme for V_1.

Figure 2.8 shows the successive quantization scheme proposed in [9] to achieve the corner point V_1 of the rate region of a zero-mean memoryless Gaussian source X with variance σ_X^2. It is shown in [9] that the successive quantization scheme to achieve corner point V_1 is as follows

1. Encoder 1 is a quantizer of rate $R_1(V_1)$ whose input is $\mathbf{X}$ and output is $\mathbf{U}_1$. The quantization error is $\mathbf{B}_2 = \mathbf{U}_1 - \mathbf{X}$, which is a zero-mean Gaussian vector with covariance matrix $E[B_2^2 I_k]$ [9].

2. Encoder 2 is a quantizer of rate $R_2(V_1)$ with input $a_1\mathbf{X} + a_2\mathbf{U}_1$ and output $\mathbf{U}_2$, where a_1 and a_2 are the scalars used for the linear estimation of $\mathbf{U}_2$ from $\mathbf{X}$ and $\mathbf{U}_1$. The quantization error $\mathbf{B}_3 = \mathbf{U}_2 - a_1\mathbf{X} - a_2\mathbf{U}_1$ is a zero-mean Gaussian vector with covariance matrix $E[B_3^2 I_k]$ [9].

It should be noted that the quantization noise of the optimal lattice quantizer is asymptotically Gaussian and the quantization noise is also independent of the input for entropy-coded dithered quantization [32]. As a result, the additive-noise components, $\mathbf{B}_2$ and $\mathbf{B}_3$, in Figure 2.8 can be arbitrarily well approximated by using ECDQ of large dimension.

At the decoder side, $\mathbf{U}_1$ is revealed to decoder 1 and central decoder 0, and $\mathbf{U}_2$ is revealed to decoder 2 and central decoder 0. Decoder i approximates $\mathbf{X}$ by $\hat{\mathbf{X}}_i = \alpha_i \mathbf{U}_i$, $i = 1, 2$. Central decoder 0 approximates $\mathbf{X}$ by $\hat{\mathbf{X}}_0 = \beta_1 \mathbf{U}_1 + \beta_2 \mathbf{U}_2$. The rates to reveal $\mathbf{U}_1$ and $\mathbf{U}_2$ are the rates of description 1 and description 2, respectively.

Successive Quantization for a General Memoryless Source Using ECDQ

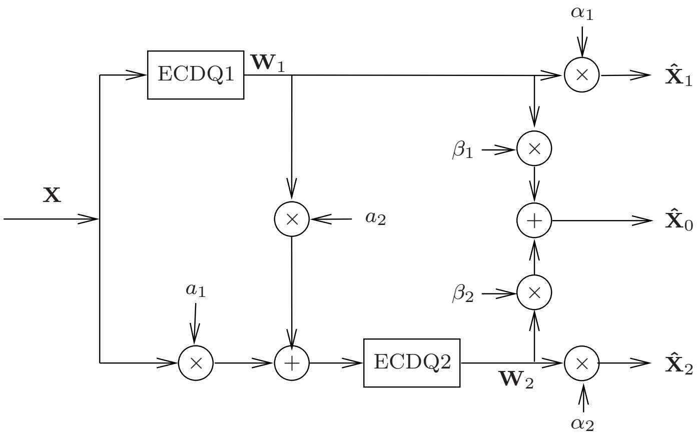


Figure 2.9: Successive quantization.

In this section we discuss the MD successive quantization scheme for a general non-Gaussian memoryless source which is proposed in [9]. Chen et al. prove in [9] that at high resolution, their proposed MD quantization scheme is asymptotically optimal for all i.i.d. sources that have finite differential entropy. Now consider the MD successive quantization scheme in Figure 2.9 that corresponds to the Gaussian MD coding scheme for corner point V_1. Let $Q_{1,k}(\cdot)$ and $Q_{2,k}(\cdot)$ denote optimal k-dimensional lattice quantizers. Let $\mathbf{Z}_1$ and $\mathbf{Z}_2$ be k-dimensional random vectors which are statistically independent and each is uniformly distributed over the basic cell of

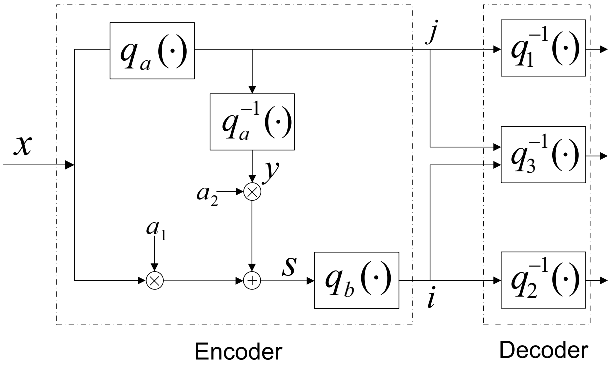


Figure 2.10: Coding scheme using successive quantization in terms of quantization encoder and decoder.

the associated lattice quantizer. The outputs of the dithered quantizers are given by

$$\mathbf{W}_1 = Q_{1,k}(\mathbf{X} + \mathbf{Z}_1) - \mathbf{Z}_1, \qquad \mathbf{W}_2 = Q_{2,k}(a_1\mathbf{X} + a_2\mathbf{W}_1 + \mathbf{Z}_2) - \mathbf{Z}_2,$$

where a_1 and a_2 are the same as discussed in section 2.2.3.

It is shown in [9] that the rates of $Q_{1,k}(\cdot)$ (conditioned on $\mathbf{Z}_1$), and $Q_{2,k}(\cdot)$ (conditioned on $\mathbf{Z}_2$) are upper-bounded as follows

$$\begin{aligned} R_1 &= \frac{1}{k}H(Q_{1,k}(\mathbf{X} + \mathbf{Z}_1)|\mathbf{Z}_1) \leq R_1(V_1) + \frac{1}{2}\log\left(2\pi e G_k^{opt}\right), \\ R_2 &= \frac{1}{k}H(Q_2(a_1\mathbf{X} + a_2\mathbf{W}_1 + \mathbf{Z}_2)|\mathbf{Z}_2) \leq R_2(V_1) + \frac{1}{2}\log\left(2\pi e G_k^{opt}\right). \end{aligned}$$

Since $G_k^{opt} \to \frac{1}{2\pi e}$ as $k \to \infty$, we have $R_1 \leq R_1(V_1)$ and $R_2 \leq R_2(V_1)$ as $k \to \infty$.

Scalar Successive Quantization Scheme and High Resolution Analysis

Figure 2.10 shows the successive quantization coding scheme when undithered scalar quantization is used. Lossy encoders $q_a(\cdot)$ and $q_b(\cdot)$ generate uniform partitions of $\mathcal{R}$, and output the indices j and i, respectively. At the receiver side, side decoders $q_1^{-1}(\cdot)$ and $q_2^{-1}(\cdot)$ decode the corresponding received index, while the central decoder $q_3^{-1}(\cdot)$ reconstruct the signal using both received indices i and j. Note that the input s to the second encoder $q_b(\cdot)$ is created by linear combination of the input signal x and the output level (codeword) of the first quantizer $y = q_a(q_a^{-1}(x))$ as the $s = a_1x + a_2y$.

Here we briefly discuss the high-resolution analysis of scalar successive quantization scheme in [9]. It is shown in [9] that we need to set $a_1 \approx 2$ and $a_2 \approx -1$ in order to have balanced side distortions. The performance of the scalar successive quantization scheme in Figure 2.10 can be tuned by proper selection of step-size of the side quantizers. Let Δ_a and Δ_b indicate the step-size of q_a and q_b, respectively. Then, the performance of MD quantization system in Figure 2.10 for several special cases is as follows [9].

1. $\Delta_a = \Delta_b$: In this case, side quantizers form two uniform scalar quantizers with their bins staggered by half the step size. Thus, the side distortion are $D_1 = D_2 \approx \frac{1}{12}\Delta_a^2 \approx \frac{2\pi e}{12}2^{-2R_1}\sigma_x^2$, where the second equality is true when entropy coding is used, and $D_0 \approx \frac{1}{4}D_1$ [33].

2. $\Delta_a \gg \Delta_b$: In this case, the partition by q_a is still uniform, and the performance of q_1 is given by $D_1 \approx \frac{1}{12}\Delta_a^2 \approx \frac{2\pi e}{12}2^{-2R_1}\sigma_x^2$. But since the most of the partition cells of second quantizer $C_s(i)$ are no longer intervals, a_1 and a_2 need to be tuned properly in order to get balanced side distortions. These values are found in [9] to be $a_1 = 2$ and $a_2 = -1.0445$. Using these values for a_1 and a_2, the side distortions become $D_1 = D_2 \approx \frac{2\pi e}{12}2^{-2R_1}\sigma_x^2$, and central distortion becomes $D_0 \approx 0.8974 \cdot \frac{2\pi e}{48}2^{-2R_2}\sigma_x^2$ [9].

2.2.4 MDC with Dithered Delta-Sigma Quantization

In this section we discuss a recent multiple description scheme using dithered delta-sigma quantization which was proposed by Østergaard and Zamir in [34]. The basic

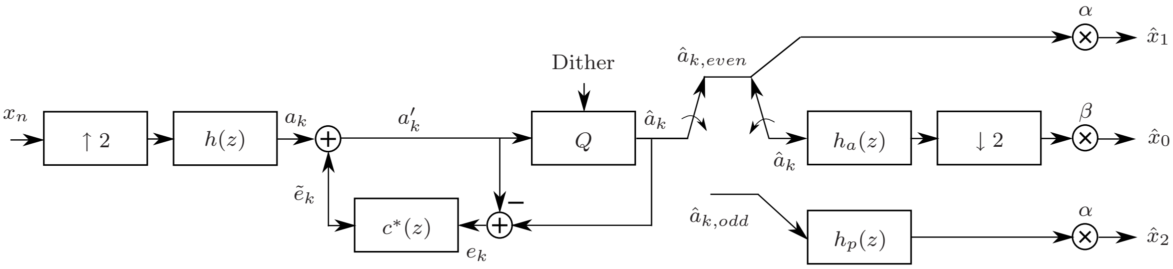


Figure 2.11: Delta-Sigma quantization scheme with two channels.

idea of their scheme is to use oversampling delta-sigma quantization in order to add controlled amount of redundancy to the original signal. Then, a noise shaping filter is employed to trade off central distortion for side distortion. Multiple descriptions are then obtained by downsampling the output of the quantizer.

Figure 2.11 shows the general scheme proposed in [34]. Let X be an i.i.d. zero-mean unit-variance Gaussian random process where x denotes a realization of X. First, the input signal x is upsampled by a factor of two to produce the oversampled signal a. The signal is then quantized by an entropy-coded dithered lattice quantizer which was introduced in previous section. This makes the quantization error E an i.i.d zero-mean random process of variance σ_E^2 that is independent of the input signal. The quantization error e_k of the k-th sample is then fed into the filter $c^*(z) = \sum_{i=1}^{p} c_i z^{-i}$, where p is the order of filter. The purpose of $c^*(z)$ is to shape the in-band and out-band noise and predict the in-band noise component $\tilde{e}_k$. The output of the quantizer can be represented by $\hat{a}_k = a_k + e_k + \tilde{e}_k$. The reconstruction error in oversampled domain is then given by $\epsilon_k = \hat{a}_k - a_k$. Since $\hat{a}(z) = a(z) + e(z) + c^*(z)e(z)$, the reconstruction error can be equivalently represented by $\epsilon(z) = c(z)e(z)$. The first description is obtained from even output samples and the second description from the odd output samples. The purpose of filter $h_p(z)$ is to correct the phase of the second description and post processing filters α and β are optimized later to achieve the dominant face (lower bound) of Ozarow's MD rate-distortion function.

It is shown in [34] that this quantization scheme asymptotically (in lattice dimension and order of noise shaping filter) achieves the two-channel MD rate-distortion function for the memoryless Gaussian source and MSE distortion measure at any resolution.

2.3 Multiple Description Coding with Correlating Transform

The basic idea in transform coding is to produce uncorrelated transform coefficients because otherwise there is linear statistical dependency that could be exploited to improve performance. For MD coding, statistical dependencies between transform coefficients can be useful because the estimation of transform coefficients that are in a lost description is improved. This idea for MD transform coding was originated by Wang et al. in [35]. The transform in this technique explicitly adds redundancy whereas odd/even separation uses similar inherent redundancy.

Let X_1 and X_2 be independent zero-mean Gaussian random variables with variances $\sigma_1^2 > \sigma_2^2$. The simplest possible correlating transform scheme, as proposed in [35], is to transmit quantized versions of Y_1 and Y_2 given by

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}. \tag{2.20}$$

Since the variances of Y_1 and Y_2 are both $(\sigma_1^2+\sigma_2^2)/2$, the central decoder performance is [36]

$$D_0 = \frac{\pi e}{12}\sigma_1\sigma_2 2^{-2R}.$$

Now consider the situation at side decoder 1. The distortion is approximately equal to the quantization error plus the distortion in estimating Y_2 from Y_1. Since Y_1 and Y_2 are jointly Gaussian, $Y_2|Y_1 = y_1$ is Gaussian and $E[Y_2|Y_1 = y_1]$ is a linear function of y_1. Specifically, $Y_2|Y_1 = y_1$ has mean $(\sigma_1^2 + \sigma_2^2)^{-1}(\sigma_1^2\sigma_2^2)y_1$ and variance $(\sigma_1^2 + \sigma_2^2)^{-1}\sigma_1^2\sigma_2^2$. Thus [36]

$$D_s \approx \frac{2\sigma_1^2\sigma_2^2}{(\sigma_1^2 + \sigma_2^2)} + \frac{\pi e}{12}\sigma_1\sigma_2 2^{-2R}. \tag{2.21}$$

By varying the transform parameters, this method allows a trade-off between D_0 and D_s.

The idea of MD coding with orthogonal transform in (2.20) was later extended to nonorthogonal transform [37] which gives more flexibility over tuning side distortions and central distortion.

The following section describes the treatment of Wang et al. [37,38,39] to this problem. The redundancy-rate distortion (RRD) performance of a correlating transform with an arbitrary dimension N was analyzed for a generally asymmetrical channel environment. For the case of $N = 2$ and symmetric channels, a closed-form solution was obtained for the optimal transform. They also proposed to generate $N \geq 2$ descriptions by cascading 2×2 transforms. Using transforms to introduce correlation between multiple descriptions has also been considered by Goyal et al. [40, 41].

2.3.1 MDC Using Pairwise Correlating Transform

MDC with pairwise correlating transform (PCT) is well studied and analyzed by Wang et al [39]. They simply propose to group the N samples into $N/2$ pairs and apply a PCT to each pair. They also address two essential design problems. The first is how to optimize the RRD performance of a PCT operating on a pair. For two independent Gaussian variables, they analyze the RRD performance of an arbitrary 2×2 transform and derive the unique optimal transform that minimizes the single description distortion at a given redundancy. The second design problem is how to incorporate pairwise transforms in a system based on samples. Assuming the N-dimensional vector process is independent and identically distributed (i.i.d.) with Gaussian distributions, they consider how to optimally allocate redundancy among a prescribed set of pairs, and derive the optimal pairing strategy that achieves the best RRD performance. The rest of this section summarizes the proposed schemes and derived analysis in [39].

Redundancy Rate-Distortion Analysis of Pairwise Correlating Transform

A pairwise MDC transform $\mathbf{T}$ takes two independent input variables A and B, and outputs two transformed variables C and D

$$\begin{bmatrix} C \\ D \end{bmatrix} = \mathbf{T} \begin{bmatrix} A \\ B \end{bmatrix}. \tag{2.22}$$

The transform $\mathbf{T}$ controls the correlation between A and B, which in turn controls the redundancy of the MDC coder. As mentioned earlier, direct quantization of the transform coefficients results in skewed partition cells which always have greater distortion. Wang et al. solve this problem in [39] by first applying a scalar quantizer to the two input variables A and B to yield integer indices, and then applying a discrete version of the transform to yield integer indices in the transform domain.

A general transform $\mathbf{T}$ is parameterized by [39]

$$\mathbf{T} = \begin{bmatrix} r_2 \cos\theta_2 & -r_2 \sin\theta_2 \\ -r_1 \cos\theta_2 & r_1 \sin\theta_2 \end{bmatrix}, \tag{2.23}$$

and correlation is parameterized by $E\{CD\} = \sigma_c \sigma_d \cos\phi$. Allocating bit rate optimally within each pair, the redundancy ρ is denoted by the difference of the rates required for coding the pairs (C, D) (A, B).

The optimal transform that minimizes the average one-channel distortion per variable is derived as [39]

$$\mathbf{T} = \begin{bmatrix} \sqrt{\frac{\cot\theta_1}{2}} & \sqrt{\frac{\tan\theta_1}{2}} \\ -\sqrt{\frac{\cot\theta_1}{2}} & \sqrt{\frac{\tan\theta_1}{2}} \end{bmatrix}. \tag{2.24}$$

This shows that the optimal transform is formed by two equal length basis vectors that are rotated away from the original basis by the same angle in opposite directions.

In the high redundancy region ($\rho \gg 1/2$), the approximate average side distortion is found to be [39]

$$D_{s,optimal} \approx \frac{\sigma_A^2 - \sigma_B^2}{8} 2^{-4\rho} + \frac{\sigma_B^2}{2},$$

and the corresponding redundancy is

$$\rho_{max,optimal} = \frac{1}{2} \log_2 \frac{\sigma_C \sigma_D}{\sigma_A \sigma_B}.$$

Therefore, D_1 decays exponentially with ρ for high redundancies at a rate that depends on the difference of the variances of the two variables, but converges to a nonzero error $\sigma_B^2/2$.

Given the desired redundancy to add when coding N uncorrelated coefficients, another design issues that remains to be addressed is the choice of optimal pairing among $\prod_{m=1}^{M}(2m-1)$ possible pairing combinations, where $M = N/2$ where is the number of pairs. The optimal pairing scheme that minimizes the average one-channel distortion is reported in [39], and found by pairing the k-th largest variable with the $(N-k)$-th largest one.

Chapter 3

MDVQ by Joint Codebook Design

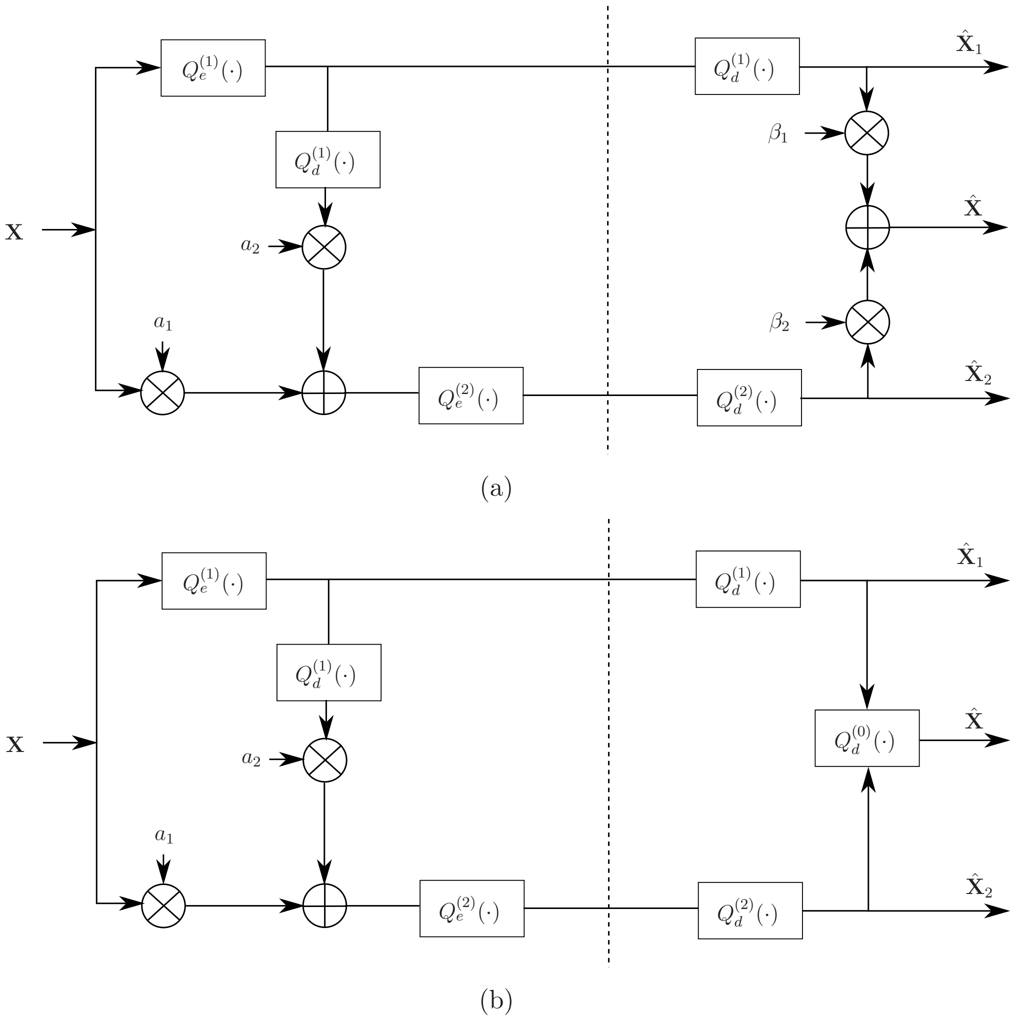
3.1 Introduction

The first constructive method towards multiple description scalar quantization (MDSQ) was proposed by Vaishampayan [16, 17]. The key component of this method is the index assignment, which maps an index to an index pair as the two descriptions. However, the design of the index assignment is a difficult problem, and Vaishampayan has provided several heuristic methods to construct balanced index assignments which are not optimal but likely to perform well. The analysis of this class of balanced quantizers reveals that asymptotically (in rate) it is 3.07 dB away from the rate-distortion bound [19] in terms of central and side distortion product, when a uniform central quantizer is used; this granular distortion gap can be reduced by 0.4 dB when the central quantizer cells are better optimized [42].

The framework of MDSQ was later extended to multiple description lattice vector quantization (MDLVQ) for balanced descriptions in [28] and for the asymmetric case in [43]. The design relies heavily on the choice of lattice/sublattice structure to facilitate the construction of index assignments. The analysis on these quantizers shows that the constructions are high-resolution optimal in asymptotically high dimensions; however, in lower dimension, optimization of the code-cells can also improve the high-resolution performance [44] [45]. The major difficulty in constructing both MDSQ and MDLVQ is to find good index assignments, and thus it would simplify the overall design significantly if the component of index assignment can be eliminated altogether.

As mentioned in Section 2.2.3, the MD quantization scheme proposed in [9] achieves the simplified El Gamal-Cover rate-distortion region. Inspired by this interesting successive quantization scheme, we propose two multiple description quantization schemes with an iterative method to jointly design the codebooks to minimize a Lagrangian cost function. This Lagrangian function includes central and side distortions. We find optimality conditions in order to obtain a joint codebook design algorithm for the two proposed MD quantization schemes. The proposed method enjoys simplicity and our simulation results reveal that the proposed method, for moderately large dimensions, performs rather closely to the optimum MD quantization.

The rest of this chapter is organized as follows. Section 3.2 explains the basic structure of the proposed MD quantizers. Section 3.3 presents the proposed methods. Optimal solutions for the parameters of the quantizer schemes are derived in Section 3.4. Complexity comparison is presented in Section 3.5. Section 5.5 provides simulation results. Finally, Section 3.7 concludes this chapter.



(a)

(b)

Figure 3.1: MDVQ with two channels. (a) MDVQ-WSC scheme where the central codebook is generated as the weighted sum of the two side coodbooks. (b) MDVQ-OC scheme that uses the optimum central decoder.

3.2 Successive MD Quantization Scheme

The structure of the proposed MDVQ scheme is depicted in Figure 3.1. This structure is similar to that of the asymptotically optimal MD-ECVQ scheme of [9] (see Figure 2.9), but the dithered lattice quantizers are replaced by ordinary nearest-neighbor vector quantizers. These multiple description systems produces two different lossy descriptions of the source with quantizers $Q^{(1)}(\cdot)$ and $Q^{(2)}(\cdot)$. The input is a k-dimensional vector $\mathbf{X}$. The quantizer $Q^{(1)}(\cdot)$ uses a codebook $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\}$, and the quantizer $Q^{(2)}(\cdot)$ uses a codebook $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M\}$. $Q^{(1)}(\cdot)$ and $Q^{(2)}(\cdot)$ are nearest-neighbor quantizers. Since we are interested in the balanced case where channels operating at equal rates, we take M to be equal to N, i.e. $M = N$. The Encoders $Q_e^{(1)}(\cdot)$ and $Q_e^{(2)}(\cdot)$ generate indices i and j which correspond to code vectors $\mathbf{y}_i$ and $\mathbf{z}_j$ respectively. In other words, if the input vector $\mathbf{x}$ lies in the partition region W_{ij}, then indices i and j are generated. This input will be mapped to code vectors $\mathbf{y}_i$ and $\mathbf{z}_j$ at the first and second side decoder respectively. As a result, we can introduce new partitions of the input, each associated with a particular side quantizer. We define partition regions

$$R_i = \bigcup_{m=1}^{M} W_{im}, \; S_j = \bigcup_{n=1}^{N} W_{nj}. \tag{3.1}$$

The set R_i is the set of all input vectors mapped to the first side quantizer index i, and S_j is similarly the set of all input vectors mapped to the second side quantizer index j. Input to the second quantizer, $Q^{(2)}(\cdot)$, is produced by linear transformation of the input vector $\mathbf{X}$ and $Q^{(1)}(\mathbf{X})$ (or equivalently $Q_d^{(1)}(Q_e^{(1)}(\mathbf{X}))$) with scalars a_1 and a_2.

The outputs of the encoders, i.e., the indices i and j are transmitted over the separate channels provided by the diversity system. If only one of the indices is received, the side decoders is used to reconstruct the source vector. However, if both indices are received, central decoder of MDVQ-WSC (MDVQ with Weighted Sum Central decoder) scheme reconstructs the source by linear transformation of received decoded descriptions with transformation matrices β_1 and β_2. After designing the encoder and decoder of the quantizers, the linear transformation can be replaced by the optimal MD decoder $Q^{(0)}(\cdot)$ instead. Optimized transformations a_1, a_2, β_1 and β_2 are discussed in Section 3.4. On the other hand, if both indices are received by the MDVQ-OC (MDVQ with Optimum Central decoder) system, the decoder uses the optimum central codebook to reconstruct the source.

3.3 Design Method

In this section, we present an iterative algorithm for designing the quantizers. The algorithm begins with a Lagrangian cost function which includes constraint on the side distortions. We also propose an iterative procedure in order to minimize this Lagrangian function. This iterative procedure leads to a possibly sub-optimal design of quantizers under given constraint. The Lagrangian cost function is given by

$$L = \lambda_0 D_0 + \lambda_1 D_1 + \lambda_2 D_2, \tag{3.2}$$

where $D_0 = E\left[\parallel \mathbf{X} - \hat{\mathbf{X}} \parallel^2\right]$, $D_1 = E\left[\parallel \mathbf{X} - \hat{\mathbf{X}}_1 \parallel^2\right]$, $D_2 = E\left[\parallel \mathbf{X} - \hat{\mathbf{X}}_2 \parallel^2\right]$, and expectation is performed by taking average over the training set. Since $\hat{\mathbf{X}}_1 = Q^{(1)}(\mathbf{X})$, $\hat{\mathbf{X}}_2 = Q^{(2)}(\mathbf{X})$, and $\hat{\mathbf{X}} = \beta_1\hat{\mathbf{X}}_1 + \beta_2\hat{\mathbf{X}}_2 = \beta_1 Q^{(1)}(\mathbf{X}) + \beta_2 Q^{(2)}(\mathbf{X})$, we can rewrite

Lagrangian function as

$$\begin{aligned} L &= \lambda_0 E\left[\| \mathbf{X} - \beta_1 Q^{(1)}(\mathbf{X}) - \beta_2 Q^{(2)}(\mathbf{X}) \|^2\right] \\ &\quad + \lambda_1 E\left[\| \mathbf{X} - Q^{(1)}(\mathbf{X}) \|^2\right] + \lambda_2 E\left[\| \mathbf{X} - Q^{(2)}(\mathbf{X}) \|^2\right]. \end{aligned} \tag{3.3}$$

We first drive optimality conditions in the following section and then introduce an iterative procedure to get close as much as possible to the optimum solution of the MD problem.

3.3.1 Optimality Conditions for the MDVQ-WSC system

For a fixed second side quantizer $Q^{(2)}(\cdot)$ and for a given first side quantizer $Q^{(1)}(\cdot)$ partition of the input space, the $Q^{(1)}(\cdot)$ codebook is optimal if for each i, $\mathbf{y}_i$ minimizes the conditional Lagrangian function in region R_i. As a result, the optimal value of $\mathbf{y}_i$ is the value of $\mathbf{y}$ that minimizes the following conditional Lagrangian function

$$L_{1,i} = \lambda_0 E\left[\| \mathbf{X} - \beta_1 \mathbf{y} - \beta_2 Q^{(2)}(a_1\mathbf{X} + a_2\mathbf{y}) \|^2 | \ \mathbf{X} \in R_i\right] + \lambda_1 E\left[\| \mathbf{X} - \mathbf{y} \|^2 | \ \mathbf{X} \in R_i\right]. \tag{3.4}$$

Since $\mathbf{y}$ is an argument of the quantization function $Q^{(2)}(\cdot)$, an explicit solution of minimization of Lagrangian function turns out to be intractable. However, if we ease the optimality similar to [46] and fix the $Q^{(2)}(\cdot)$, then the above Lagrangian function becomes quadratic in $\mathbf{y}$ and we can minimize it with an iterative technique which takes the encoder of the second quantizer to be fixed while optimizing the decoder of the first quantizer. Then we seek $\mathbf{y}$ to minimize the Lagrangian function

$$L_{1,i} = \lambda_0 E\left[\| \mathbf{X} - \beta_1 \mathbf{y} - \beta_2 \mathbf{U} \|^2 | \ \mathbf{X} \in R_i\right] + \lambda_1 E\left[\| \mathbf{X} - \mathbf{y} \|^2 | \ \mathbf{X} \in R_i\right], \tag{3.5}$$

where $\mathbf{U} = Q^{(2)}(a_1\mathbf{X} + a_2\mathbf{y}_i)$. Taking gradient of (3.5) with respect to $\mathbf{y}$ gives

$$\frac{\partial L_{1,i}}{\partial \mathbf{y}} = -2\lambda_0 \beta_1^T E\left[(\mathbf{X} - \beta_1 \mathbf{y} - \beta_2 \mathbf{U}) \mid \mathbf{X} \in R_i\right] - 2\lambda_1 E\left[(\mathbf{X} - \mathbf{y}) \mid \mathbf{X} \in R_i\right]. \tag{3.6}$$

The optimal $\mathbf{y}$ can then be found by solving the equation

$$(\lambda_0\beta_1^T\beta_1+\lambda_1 I)\mathbf{y}=\lambda_0\beta_1^T E\left[(\mathbf{X}-\beta_2\mathbf{U})\mid\mathbf{X}\in R_i\right]+\lambda_1 E\left[\mathbf{X}\mid\mathbf{X}\in R_i\right], \tag{3.7}$$

where I is the identity matrix. The solution of the above equation is given by

$$\mathbf{y}_i^*=\left(\lambda_0\beta_1^T\beta_1+\lambda_1 I\right)^{-1}\left(\lambda_0\beta_1^T E\left[(\mathbf{X}-\beta_2 Q^{(2)}(a_1\mathbf{X}+a_2\mathbf{y}_i))\mid\mathbf{X}\in R_i\right]+\lambda_1 E\left[\mathbf{X}\mid\mathbf{X}\in R_i\right]\right) \tag{3.8}$$

For a fixed $Q^{(1)}(\cdot)$ and for a given $Q^{(2)}(\cdot)$ partition of the input space, the $Q^{(2)}(\cdot)$ codebook is optimal if for each j, $\mathbf{z}_j$ minimizes the conditional Lagrangian function in region S_j. Then, the optimal value of $\mathbf{z}_j$ is the value of $\mathbf{z}$ that minimizes the following conditional Lagrangian function

$$L_{2,j}=\lambda_0 E\left[\parallel\mathbf{X}-\beta_1 Q^{(1)}(\mathbf{X})-\beta_2\mathbf{z}\parallel^2\mid\mathbf{X}\in S_j\right]+\lambda_2 E\left[\parallel\mathbf{X}-\mathbf{z}\parallel^2\mid\mathbf{X}\in S_j\right]. \tag{3.9}$$

Similarly, we seek $\mathbf{z}$ to minimize the Lagrangian function

$$L_{2,j}=\lambda_0 E\left[\parallel\mathbf{U}(\mathbf{X})-\beta_2\mathbf{z}\parallel^2\mid\mathbf{X}\in S_j\right]+\lambda_2 E\left[\parallel\mathbf{X}-\mathbf{z}\parallel^2\mid\mathbf{X}\in S_j\right], \tag{3.10}$$

where $\mathbf{U}(\mathbf{X})=\mathbf{X}-\beta_1 Q^{(1)}(\mathbf{X})$. Taking gradient of (3.10) with respect to $\mathbf{z}$ gives

$$\frac{\partial L_{2,j}}{\partial\mathbf{z}}=-2\lambda_0\beta_2^T E\left[(\mathbf{U}(\mathbf{X})-\beta_2\mathbf{z})\mid\mathbf{X}\in S_j\right]-2\lambda_2 E\left[(\mathbf{X}-\mathbf{z})\mid\mathbf{X}\in\mathbf{X}_j\right]. \tag{3.11}$$

The optimal $\mathbf{z}$ can then be found by solving the following equation

$$(\lambda_0\beta_2^T\beta_2+\lambda_2 I)\mathbf{z}=\lambda_0\beta_2^T E\left[\mathbf{U}(\mathbf{X})\mid\mathbf{X}\in S_j\right]+\lambda_2 E\left[\mathbf{X}\mid\mathbf{X}\in S_j\right], \tag{3.12}$$

where I is the identity matrix. The solution of the above equation is given by

$$\mathbf{z}_j^*=\left(\lambda_0\beta_2^T\beta_2+\lambda_2 I\right)^{-1}\left(\lambda_0\beta_2^T E\left[\mathbf{X}-\beta_1 Q^{(1)}(\mathbf{X})\mid\mathbf{X}\in S_j\right]+\lambda_2 E\left[\mathbf{X}\mid\mathbf{X}\in S_j\right]\right) \tag{3.13}$$

Equations (3.8) and (3.13) give the design conditions to enhance codebooks of the side quantizers in an iterative procedure in order to minimize the Lagrangian cost function.

3.3.2 Optimality Conditions for MDVQ-OC system

The derivation of optimality conditions for MDVQ-OC system is almost identical to the argument in the previous section. Since this scheme uses the optimum central decoder, the first terms of $L_{1,i}$ and $L_{2,j}$ in equations (3.5),(3.9) vanish and the optimal $\mathbf{y}^*$ and $\mathbf{z}^*$ are found to be

$$\mathbf{y}_i^* = E\left[\mathbf{X} \mid \mathbf{X} \in R_i\right], \tag{3.14}$$

$$\mathbf{z}_j^* = E\left[\mathbf{X} \mid \mathbf{X} \in S_j\right]. \tag{3.15}$$

3.3.3 Design Algorithm

This section demonstrates an iterative technique to enhance the codebooks and, consequently, minimize Lagrangian cost function as the optimization criterion. The iterative algorithm is similar to GLA (Generalized Lloyd-Max Algorithm) [23]. However, unlike the GLA, it does not necessarily produce a non-increasing sequence of Lagrangian values.

Suppose we have a training set $\mathbf{T}$ including K training vectors $\mathbf{x}_k$, $k = 1, 2, \cdots, K$. The Lagrangian cost function can be approximated as

$$L = \frac{1}{T}\left[\lambda_0 \sum_{k=1}^{K} \parallel \mathbf{x}_k - \hat{\mathbf{x}}_k \parallel^2 + \lambda_1 \sum_{k=1}^{K} \parallel \mathbf{x}_k - \hat{\mathbf{x}}_{1k} \parallel^2 + \lambda_2 \sum_{k=1}^{K} \parallel \mathbf{x}_k - \hat{\mathbf{x}}_{2k} \parallel^2\right], \tag{3.16}$$

where $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_{ik}$ for $i = 1, 2$ are respectively central and side quantized training vectors. We also use superscript (n) to indicate variables in the nth iteration step. Suppose we have initial codebooks $\mathbf{Y}^{(0)}$ and $\mathbf{Z}^{(0)}$ for the first and second quantizer respectively which are obtained by traditional single description design. Let L_n denote the Lagrangian value in (3.3) computed in the nth iteration step. The optimized transformations β_1 and β_2 are derived in Section 3.4.

The iterative algorithm steps are as follows

1) *Encode and Partition Training Set:* Encode each vector in the training set with the current codebooks. Let $i(k)$ and $j(k)$ denote the indices generated in encoding vector $\mathbf{x}_k \in \mathbf{T}$. Compute L_{n+1} according to (3.14).

2) *Termination Test:* If $|L_n - L_{n+1}|/L_n < \delta$, where δ is a fixed small positive threshold, or if n has exceeded the number of maximum desired steps, the algorithm is terminated.

3) *Update $Q^{(1)}(.)$ Codebook:* Each code vector in the first side quantizer codebook is replaced by the conditional centroid according to (3.8) and (3.14) for MDVQ-WSC and MDVQ-OC respectively in order to obtain new codebook $\mathbf{Y}^{(n+1)}$.

4) *Encode and Repartition Training Set:* Produce a new set of indices $i(k)$ and $j(k)$ according to updated codebook $\mathbf{Y}^{(n+1)}$.

5) *Update $Q^{(2)}(\cdot)$ Codebook:* Each code vector in the second side quantizer codebook is replaced by the conditional centroid given in (3.13) and (3.15) for MDVQ-WSC and MDVQ-OC respectively in order to obtain new codebook $\mathbf{Z}^{(n+1)}$. Go back to step 1.

Since only by jointly searching the codebooks the encoder generates optimal partitions, in the above design algorithm the iteration procedure may lead to sub-optimal partitions. Consequently, it is possible for the Lagrangian value to be larger than that computed in the last iteration. The possibility of a non-monotonic Lagrangian value raises the issue of how to effectively terminate the iterative process. Similar to the remedy proposed in [46], the termination step may be modified so that the algorithm will terminate only when the relative change in L_n is less than δ for several consecutive steps, or when the total number of algorithm steps has reached a given limit. Another consequence of the non-monotonicity in L_n is that the final stage codebooks at termination may not be the best ones. This problem is easily solved by choosing the stage codebooks from an intermediate iteration with the lowest L_n.

Once the encoder and decoder of the MDVQ-WSC quantizers are found by the proposed iteration technique, the central decoder can be replaced by the optimal MD decoder instead. In other words, given received code vectors $\mathbf{y}_i$ and $\mathbf{z}_j$ from first and second channel respectively, the optimal central decoder $Q_d^{(0)}(\cdot)$ reconstructs the central description as $Q_d^{(0)}(\mathbf{y}_i, \mathbf{z}_j) = E[\mathbf{X}|\mathbf{X} \in W_{ij}]$. According to our simulation results, this adjustment results in better performance of central decoder.

3.3.4 Choosing Lagrangian Multipliers λ_i

We introduced an iteration technique in the previous section in order to minimize the Lagrangian cost function for a fixed set of Lagrangian multipliers λ_i, $i = 0, 1, 2$. For target side distortions, there is another problem left to address which is finding the optimal λ_i^*, $i = 0, 1, 2$ that leads to the side distortions converging to the desired target side distortion. In fact, each set of $(\lambda_0, \lambda_1, \lambda_2)$ corresponds to a single point on the convex hull of MD achievable distortion region. As a result, as the values of λ_i, $i = 0, 1, 2$ change, we have a trade off between central and side distortions. Therefore, appropriate selection of λ_i leads to the target distortions at side decoders.

The search for the $(\lambda_0, \lambda_1, \lambda_2)$ is somehow analogous to the search for the appropriate value for λ, or equivalently the slope of the rate-distortion function, in the design of entropy constrained vector quantizer (ECVQ) [47]. For ECVQ, [47] proposes a bisection approach to allow code design for a particular desired rate. The ECVQ algorithm designs a vector quantizer for a specific λ at the middle of a range $[\lambda_{min}, \lambda_{max}]$. The design process then shortens this range to the lower or higher half in the direction that decreases the gap between the obtained and desired rate.

Now consider the Lagrangian function introduced in (3.2). For balanced distortions where $\lambda_1 = \lambda_2 = \tilde{\lambda}$, and $D_s = \frac{1}{2}(D_1 + D_2)$, the Lagrangian function in (3.2) can be rewritten as

$$\begin{aligned} L &= \lambda_0 D_0 + \tilde{\lambda}(D_1 + D_2) \\ &= \lambda_0 D_0 + 2\tilde{\lambda} D_s. \end{aligned} \tag{3.17}$$

Since only the relative values of Lagrangian multipliers are meaningful [48], we can divide the equation (3.17) by λ_0. Then, the Lagrangian function reduces to

$$L = D_0 + \lambda D_s, \tag{3.18}$$

where $\lambda = 2\tilde{\lambda}/\lambda_0$. As proved in [49], a small value of λ leads to higher D_s, and a large value of λ leads to a smaller D_s. Thus, we can modify the iteration technique of previous section as follows. Similar to the approach proposed in [47], we limit the value of λ to the range $[0, 1]$, and set $\lambda = 0.5$ as the initial value. We then observe the obtained average side distortion D_s at the end of each iteration. If the obtained D_s is higher than the target side distortion, we simply shorten the range of λ to the higher half. Similarly, if the obtained D_s is lower than the target side distortion, we shorten the range of λ to the lower half. It should be noted that obtaining a D_s lower than the target side distortion is not necessarily desired, since it leads to a higher central distortion. For instance, at the end of the first iteration, if the observed D_s is higher than the target side distortion, we update $\lambda = 0.75$ which is the middle of the range $[0.5, 1]$.

Alternatively, consider the obtained Lagrangian function at the end of n-th iteration as

$$L_n = \lambda_{0,n} D_{0,n} + \lambda_{1,n} D_{1,n} + \lambda_{2,n} D_{2,n}. \tag{3.19}$$

For the target side distortions $D_{1,t}$ and $D_{2,t}$, we propose to modify the Lagrangian multipliers of (3.19) as follows

$$\begin{aligned} \tilde{\lambda}_{i,n+1} &= \lambda_{i,n} \frac{D_{i,n}}{D_{i,t}}, \qquad i = 1, 2 & (3.20) \\ \tilde{\lambda}_{0,n+1} &= \lambda_{0,n}. & (3.21) \end{aligned}$$

Assuming that sum of multipliers is one, we normalize the $\tilde{\lambda}_{i,n+1}$, $i = 0, 1, 2$ as

$$\lambda_{i,n+1} = \frac{\tilde{\lambda}_{i,n+1}}{\sum_{j=0}^{2} \tilde{\lambda}_{j,n+1}}, \qquad i = 0, 1, 2. \tag{3.22}$$

This way L_n remains a convex combination of individual distortions. In general, equation (3.20) simply scales $\lambda_{i,n}$, $i = 1, 2$ proportional to the ratio of the observed corresponding side distortion to the target side distortion, and equation (3.22) normalizes the resulting $\tilde{\lambda}_{i,n+1}$, $i = 0, 1, 2$. As a result, this may lead to the faster convergence of λ_i, $i = 0, 1, 2$ to the optimal ones. This simple procedure allows us to efficiently control the trade off between the central and side distortions. Figure 3.2 demonstrates the effect of tuning the Lagrangian multipliers according to (3.20)-(3.22) on the observed side distortions for a 4-dimensional memoryless Gaussian input source and target side distortions $D_{1,t} = D_{2,t} = 0.66$ and rate $R = 0.5$ bpss.

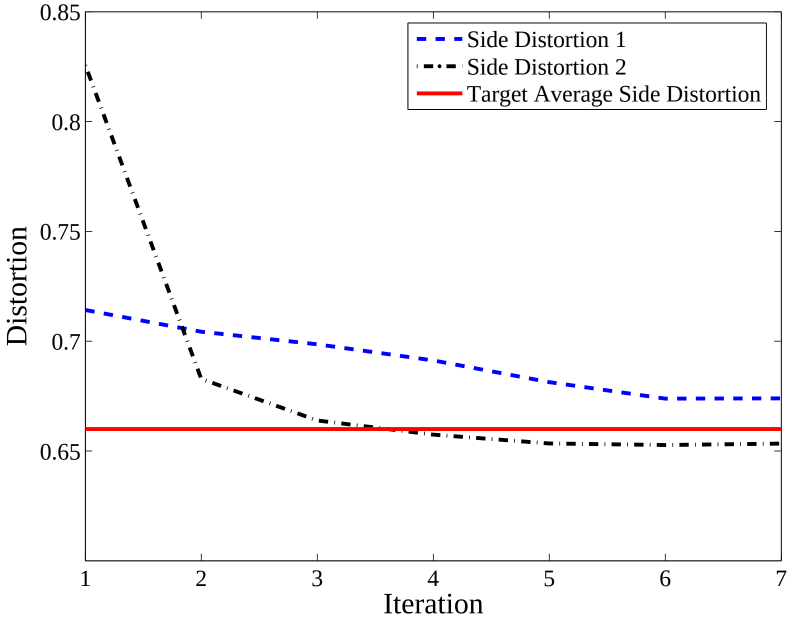


Figure 3.2: Effect of tuning the Lagrangian multipliers on side distortions for a memoryless Gaussian input source and target side distortions $D_{1,t} = D_{2,t} = 0.66$. $k = 4$, and $R = 0.5$ bpss.

3.4 Optimal Transformations a_i and β_i

3.4.1 Optimal a_i

Assuming a balanced case, a_1 and a_2 must be chosen carefully such that they lead to balanced side distortions, $D_1 \approx D_2$. We investigated the effect of a_1 and a_2 on the side distortions. Figure 3.3 shows the side distortions as the coefficient a_2 increases and a_1 is kept fixed at $a_1 = -1$ for $k = 4$ and $R = 0.5$ bpss. Side distortion of the first quantizer almost remains constant while the second side distortion slightly changes with various values of a_1 and a_2.

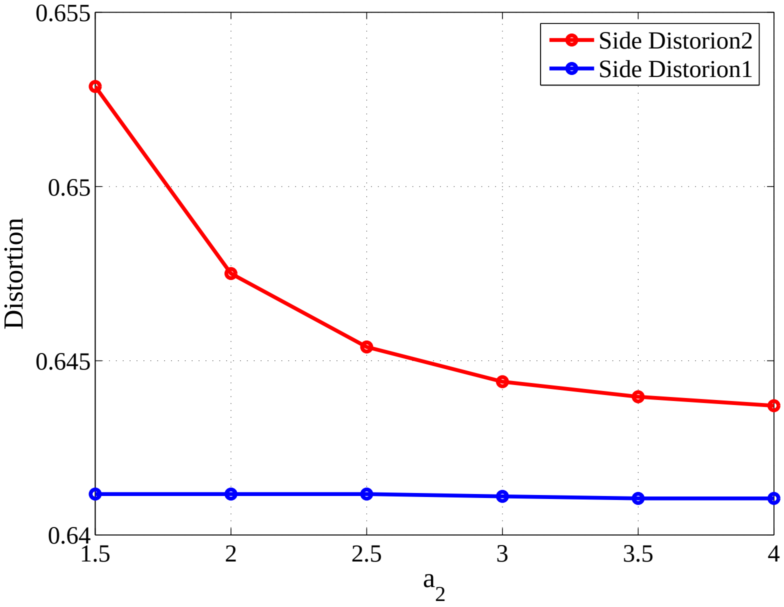


Figure 3.3: Distortion of the side decoders as the coefficient a_2 increases. $a_1 = -1$, $k = 4$, and $R = 0.5$ bpss.

3.4.2 Optimal β_i

We can easily drive optimal transformations β_1 and β_2 by minimizing the central distortion, $D_0 = E\left[\| \mathbf{X} - \beta_1\hat{\mathbf{X}}_1 - \beta_2\hat{\mathbf{X}}_2 \|^2\right]$, with respect to matrices β_1 and β_2. Using the orthogonality principle, the optimal β_1 and β_2 must satisfy

$$E\left[(\mathbf{X} - \beta_1\hat{\mathbf{X}}_1 - \beta_2\hat{\mathbf{X}}_2)\hat{\mathbf{X}}_1^T\right] = 0, \tag{3.23}$$

$$E\left[(\mathbf{X} - \beta_1\hat{\mathbf{X}}_1 - \beta_2\hat{\mathbf{X}}_2)\hat{\mathbf{X}}_2^T\right] = 0. \tag{3.24}$$

If we define $\beta = [\beta_1 \ \beta_2]$ and $\hat{\mathbf{X}}^T = [\hat{\mathbf{X}}_1^T \ \hat{\mathbf{X}}_2^T]$, then we can rewrite equations (3.23-3.24) as

$$E[(\mathbf{X} - \beta\hat{\mathbf{X}})\hat{\mathbf{X}}^T] = 0.$$

As a result, β can be found by solving $\beta E[\hat{\mathbf{X}}\hat{\mathbf{X}}^T] = E[\mathbf{X}\hat{\mathbf{X}}^T]$ which gives

$$\beta = E[\mathbf{X}\hat{\mathbf{X}}^T](E[\hat{\mathbf{X}}\hat{\mathbf{X}}^T])^{-1}.$$

Table 3.1: Computational Complexity and Memory Requirement Comparison. For k-dimensional input vector, $R_1 = R_2 = R$ bpss, and $N = 2^{kR}$.

	Computational Complexity	Memory Requirement
Optimum MDVQ	N^2k	$N^2k + 2N^2$
MDVQ-OC	$N^2 + 4Nk$	$4Nk + N^2$
MDVQ-WSC	$4Nk$	$4Nk + 2N$

3.5 Complexity and Memory Requirement

In this section the computational complexity and memory requirements of our proposed methods are compared with those of Vaishampayan's potentially optimum MD vector quantizer [16]. The scheme proposed in [16] is a very general form of MDVQ that, if its parameters are appropriately chosen, can achieve optimal performance for a given quantizer dimension. However, due to its general structure, the scheme is rather complex. Also, in practice its optimality is not guaranteed as the iterative algorithm for its design only ensures convergence to a locally optimum solution. The computational complexity, which is the number of multiplication and addition operations, and memory requirement are compared in Table 3.1. The number of needed operations for encoding each k-dimensional input vector in a conventional vector quantizer is mainly due to the calculation of distortion between two vectors, and is given by N^2k operations. In our proposed methods, the equations (3.8),(3.13) for MDVQ-WSC, and equations (3.14),(3.15) for MDVQ-OC give the number of required operations which is $4Nk$ and $N^2 + 4Nk$ respectively. Figure 3.4 also shows the complexity comparison as the bit rate increases. It is clear that our proposed methods, and in particular MDVQ-WSC method, can significantly reduce the computational complexity and memory requirements when squared error distortion measure is used for distortion.

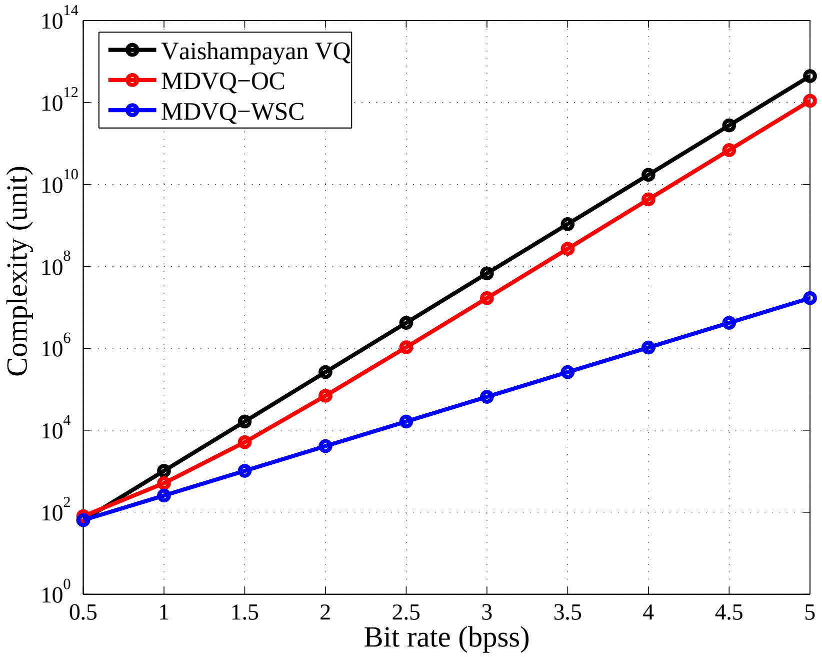


Figure 3.4: The complexity comparison as a function of bit rate for 4-dimensional source vector.

3.6 Simulation Results

Simulation results are provided in this section for MDVQ-WSC and MDVQ-OC schemes with two channels for a zero-mean, unit variance, stationary, first-order, Gauss-Markov source with correlation coefficient ρ. The encoding rates are set to $R_1 = R_2 = 0.5$ bpss (bits per source sample). Block sizes $k = 4$ and $k = 8$ are considered. We have also set $\lambda_1 = \lambda_2 = \lambda$ in all the results presented here. A training set of length 50000 source vectors was used along with a termination threshold of 0.001 in all cases.

Initialization of the design algorithm in order to obtain an initial set of codebooks is an important issue. The first applied initialization technique selects the codebook

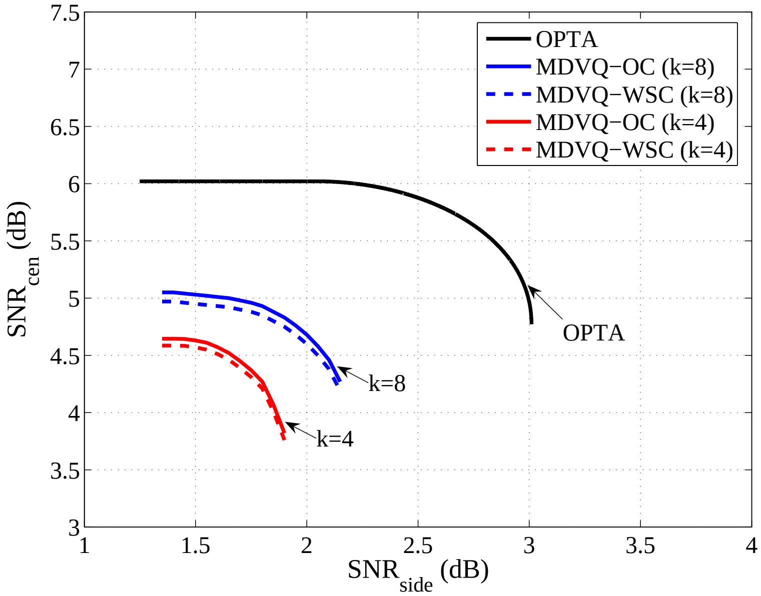


Figure 3.5: MDVQ with two channels for unit-variance memoryless Gaussian $k = 4$ and $k = 8$-dimensional source vector at $R = 0.5$ bpss (bits per source sample) for various values of λ.

obtained by uniformly partitioning the training set. We have also used two initialization techniques reported in [16]. The first technique randomly selects vectors from the training set for which the MDVQ is to be designed. The second technique simply applies the odd-even vector quantization technique and selects the obtained codebooks as the initial codebooks. Neither technique achieves results that are uniformly better than the other. The presented simulation results are the best that have been obtained using all three initialization techniques.

Figure 3.5 and Figure 3.6 show the simulation results for $\rho = 0$ (memoryless Gaussian source) and $\rho = 0.9$ (highly correlated Gauss-Markov source). We have plotted the SNR for the central decoder, $\mathrm{SNR}_{cen} = 10\log_{10}(1/D_0)$, as a function of $\mathrm{SNR}_{side} = 10\log_{10}(2/(D_1 + D_2))$, for various values of λ. The optimum Rate-Distortion bound is the same as given in [16].

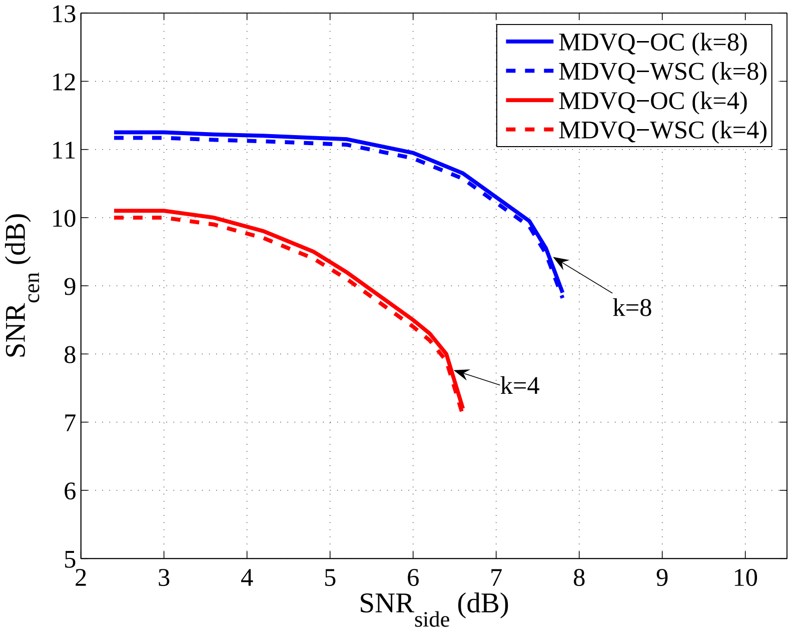


Figure 3.6: MDVQ with two channels for unit-variance Gauss-Markov $k = 4$ and $k = 8$-dimensional source vector with correlation coefficient $\rho = 0.9$ at $R = 0.5$ bpss (bits per source sample) for various values of λ.

Table 3.2: Selected performance results for memoryless Gaussian source and Gauss-Markov source with $\rho = 0.9$, and $R_1 = R_2 = 0.5$ bpss.

ρ	k	SNR_{side}(dB)	SNR_{cen}(dB)		Optimum MDVQ	OPTA (dB)
			MDVQ-WSC	MDVQ-OC		
0.0	4	1.65	4.47	4.52	4.55	6.02
0.0	8	1.75	4.89	4.96	5.01	6.02
0.9	4	6.15	8.13	8.21	8.29	-
0.9	8	7.45	9.82	9.91	10.01	-

Table 3.3: The gain achieved in replacing the linear transformation central decoder with the optimum central decoder in MDVQ-WSC. $R_1 = R_2 = 0.5$ bpss.

	Gauss-Markov		Uniform	Laplacian
	$\rho = 0.0$	$\rho = 0.9$		
$k = 4$	0.11 dB	0.14 dB	2.3 dB	0.34 dB
$k = 8$	0.15 dB	0.17 dB	2.8 dB	0.41 dB

Simulation results reveal the significant improvement in performance obtained by increasing the block size k in all cases, as expected by known property of vector quantization. An increase in block size results in a greater improvement for $\rho = 0.9$ than $\rho = 0$. Table 3.2 shows selected performance results for memoryless Gaussian source and also Gauss-Markov source with $\rho = 0.9$, and $R_1 = R_2 = 0.5$ bpss. These results are compared with the optimal bound, and also with the best experimental results achieved by Vaishampayan's optimum MDVQ in [16]. For instance, for memoryless Gaussian source where $\rho = 0$, 0.15 db is gained by increasing the block size from $k = 4$ to $k = 8$ at $\text{SNR}_{side} = 1.5$ dB for MDVQ-WSC scheme. However, for Gauss-Markov source with correlation coefficient $\rho = 0.9$, 1.15 dB is gained by increasing the block size from $k = 4$ to $k = 8$ at $\text{SNR}_{side} = 3.0$ dB. This implies the significance of increasing the block size for highly correlated sources such as speech and video. Since the tradeoff between central and side distortion can be carefully controlled by selecting the appropriate values for λ_1, λ_2 and λ_0, our approach provides the designer with greater design flexibility.

As we mentioned in Section 3.2, after designing the encoder and decoder of the quantizers of MDVQ-WSC scheme, the linear transformation central decoder can be replaced by the optimal MD decoder. This results in better performance of central decoder of MDVQ-WSC scheme. Table 3.3 summarizes the achieved gain of replacing the central decoder of MDVQ-WSC with the optimum central decoder for various source distributions. Since by heuristic considerations the linear transformation performs nearly as well as the optimum decoder for Gaussian sources, the achieved gain for Gaussian sources is expected to be negligible. However, since the output of side decoders are not Gaussian, a small gain is observed. On the other hand, for uniform source distribution, this gain is significant.

3.7 Conclusion

In this chapter, we proposed two successive multiple description quantization schemes with an iterative method to jointly design the codebooks to minimize a Lagrangian cost function. This Lagrangian function includes central and side distortions. We also find optimality conditions in order to obtain a joint codebook design algorithm for proposed MDVQ schemes. The proposed MD vector quantization scheme has relatively low complexity and for moderately large dimensions still performs close to the much more complex, potentially optimal, general MD vector quantization scheme in [16].

Chapter 4

Background on Video Compression and MD Video Coding

4.1 Introduction

A video signal comprises a sequence of frames that altogether make a three-dimensional array of pixels. Horizontal and vertical directions of a frame at any specific sampling time are considered as two spatial dimensions, and the third dimension represents the time domain. In general, the size of data representing such a three-dimensional array of pixels is very large and requires significant amount of bandwidth. However, network capacity is usually limited, and this inspires the use of video compression techniques to exploit the inherent redundancy of video data.

In this chapter we first introduce some basic video coding concepts such as video signal format, spatial and temporal redundancy, video assessment metrics, and MPEG standard. We then briefly review the extension of multiple description concept in video applications.

4.2 Digital Video Sequence and Sampling

As mentioned earlier, video signal is collection of frames which are the digital representation of video at a point in time. If a video signal is sampled at a rate of 15 frames per second (fps) or higher, the human visual system (HVS) is not able to distinguish each single frame, and this sequence of frames is seen as a continuous movement of the object in the scene. Spatial and temporal features are two key features that capture the color, brightness, and movement of objects of a video signal. Color and brightness are represented by spatial features of the video signal, while objects motion and change in illumination or color are captured by the temporal features of the video.

A natural video scene is continuous in time and space. However, digital video performs spatial and temporal sampling to represents the video scene in the digital domain. Spatial sampling stores brightness and color of pixels in each still frame captured at a point in time. The temporal sampling collects new still frames at each sampling time.

A digital colored video needs a method to store color information. The RGB is the simplest color space which uses combination of the three main colors (red, green and blue) to represent brightness and color of pixels [50]. However, video standards usually use YUV or YCbCr systems which use luminance (Y) and two chroma color differences (U and V, or Cb and Cr).

In fact, the human visual system is more sensitive to brightness than to color difference components [50]. This implies that we can represent the chrominance components with a lower spatial resolution than the required resolution for luminance component.

4.3 Digital Video Quality Assessment

The evaluation of quality of perceived video is usually very complicated. Several methods have been proposed for assessment of video quality, but none of them is believed to achieve sufficient video quality assessment by itself.

Video quality assessment is usually performed within two categories: objective and subjective. Objective video quality assessment tries to mathematically measure the quality of impaired video. On the other hand, subjective video quality assessment is performed by rating the sample video based on human perception. Here we describe some common methods in objective assessment of video quality.

4.3.1 Objective Video Quality Assessment

Many mathematical methods have been developed in order to evaluate the quality of video objectively. Here the term *objective* refers to the fact that human interaction is avoided. The early methods were applied to still images, and later extended to video sequences by simply applying the image quality metric to every frame of the video sequence

The peak signal to noise ratio (PSNR) is the simplest and widely used method in this category. If we denote the number of bits per sample by n, then $(2^n - 1)$ is the highest possible value in the frame, and PSNR is defined as follows

$$PSNR_{\mathrm{db}} = 20 \log_{10} \frac{(2^n - 1)}{RMSE}, \tag{4.1}$$

where RMSE is the root mean square error that calculates the distance between two frames as follows

$$RMSE = \sqrt{\frac{1}{K \times L}\sum_{k=0}^{K-1}\sum_{\ell=0}^{L-1}[f(k,\ell) - f'(k,\ell)]^2}, \tag{4.2}$$

where $K \times L$ is the size of the original frame $f(k,\ell)$ compared to the impaired frame $f'(k,\ell)$.

This method is quite simplistic, and the way the human eye perceives color and motion is ignored in calculation of PSNR. As a result, the distortion calculated by PSNR may not match the subjective perception of human being. For instance, if the impairments occur somewhere in spatial domain that human eye is not able to perceive, the PSNR method reports considerable distortion while subjective quality may not be affected. Furthermore, PSNR does not consider motions and sensitivity of human eye to contrast and spatial/temporal details.

The concept of contrast is simply defined as the ratio of the local intensity and the average image intensity, and is a very important concept in the context of human visual system. This is mainly due to the fact that the information is represented in human vision system as contrast, not the absolute luminance values. Generally, there should be minimum contrast, which is called the contrast threshold, to provoke a response from the neurons of human eye. Contrast sensitivity is simply defined as the inverse of contrast threshold. Contrast sensitivity varies with frequency. It has always been well known [51] that the human eye is more sensitive to lower spatial frequencies. This property of human eye is widely exploited to develop lossy video compression techniques.

Motivated by human vision system, video quality metric (VQM) [52] was developed by ITS (The Institute for Telecommunication Science) to provide an objective measurement for perceived video quality. It measures the perceptual effects of video impairments including blurring, jerky and unnatural motion, global noise, block and color distortions, and combines them into a single metric. The testing results [52] show that VQM has a high correlation with subjective video quality assessment and has been adopted by ANSI (American National Standards Institute) as an objective video quality standard.

4.4 Video Compression

Video compression generally refers to a process that converts video sequence signal to a format which requires less bandwidth than the original signal. Compression can be performed either lossless or lossy. Lossless compression results in an alternative for video signal that can be decoded to generate the original video sequence. However, since the losslessly compressed video still requires significant bandwidth, lossy compression is widely used instead. Although lossy compression does not generate the exact reconstruction of the original video, an acceptable approximation can be generated that achieves higher compression ratio than lossless techniques. The performance of lossy compression mainly depends on the data itself. In general, a data with more inherent redundancy, that can be exploited by compression techniques, results in better compression.

A standard video compression algorithm is usually based on three key techniques to exploit the redundancy and achieve better compression: spatial compression, temporal compression, and entropy coding. This section briefly describes these compression techniques.

4.4.1 Spatial Compression

Neighboring samples of each frame of a video sequence are usually highly correlated, and this property can be used to compress frames of the video sequence. Thus, spatial compression, or intra-frame compression, refers to a process which is applied to individual frames in order to exploit redundancy within each frame. Practical spatial compression techniques typically use the following techniques.

Prediction

Spatial prediction is the process of decorrelating data using previously encoded samples, and is sometimes referred to as differential pulse code modulation (DPCM) [53]. A DPCM system predicts the current sample based on previously coded samples, and encodes the residual which is subtraction of the predicted sample from the original sample. The energy of the residual is basically lower than original one, and the residual information can be represented with fewer bits.

Transform Coding

The transformation step converts a residual frame into transform domain where the correlation is lower and data are more compact. This stage is usually performed using the discrete cosine transform (DCT). The two-dimensional DCT is extensively used in many video coding standards such as MPEG that uses 8×8 DCT transform [50]. In practice, many of the DCT coefficients will be zero or have very small values and, therefore, will not be transmitted that eventually results in significant compression.

Quantization and Zig-Zag Scanning

Many DCT coefficients can be coarsely quantized without seriously affecting the quality of video [54]. For instance, since more noise can be tolerated at high spatial frequencies, MPEG standard uses a quantizer matrix which has smaller step sizes for low-frequency coefficients. Zig-zag scanning is then applied to send coefficients in descending order of magnitude probability.

4.4.2 Temporal Compression

In general, successive frames in a video sequence tend to be highly correlated. In other words, the frames change slightly over a small period of time, and if we encode only the difference between he current frame and previous reference frames, this results in significant temporal compression of inter-frames [55]. To address this issue, motion estimation is applied to determine the displacement of frames, and estimate the amount of motion between current block and previous reference blocks. This amount of motion is then represented by the associated motion vector (MV). The difference between the prediction and the original frame is computed to produce the motion compensated residual to be encoded. This technique is called Motion Compensation (MC). Since fewer bits are required to code both residual blocks and MVs than original blocks, a better compression is achieved.

4.4.3 Entropy Coding

Entropy coding is a lossless compression technique based on statistical properties of the video sequence to be compressed [50]. Basically, entropy coding encodes the most frequently occurring symbol with the least number of bits.

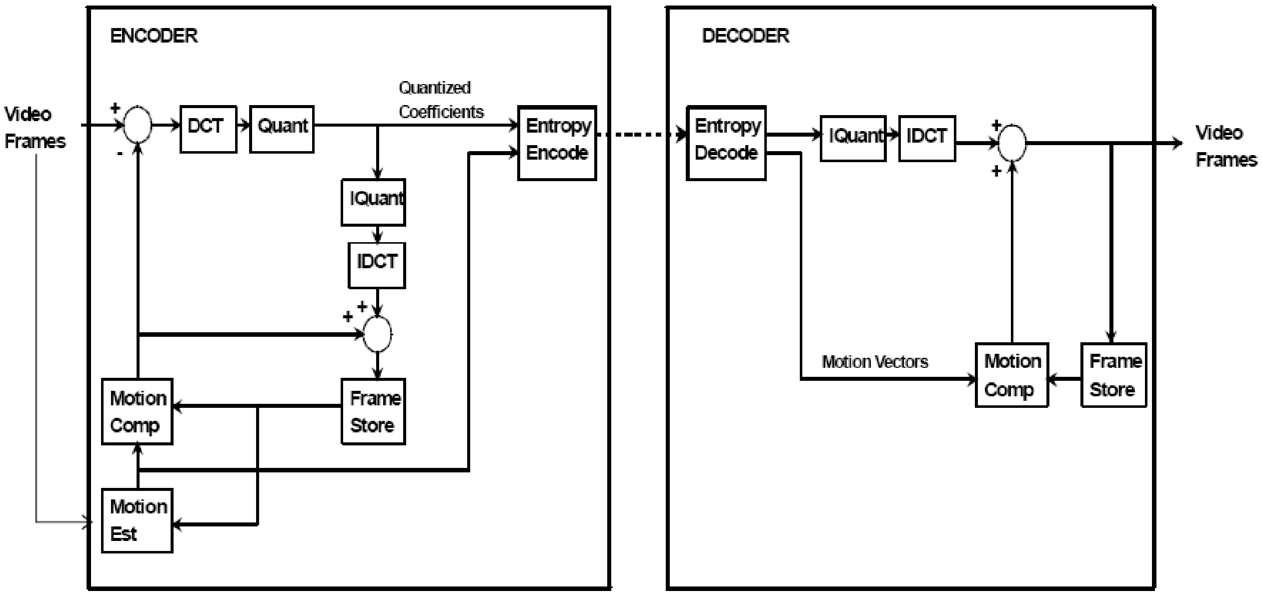


Figure 4.1: Generic DCT/DPCM CODEC.

Entropy coding of compressed video is a two step process: Zero Run-Length Coding (RLC) and Huffman coding. RLC is a representation of the quantized values with a pair of numbers. The first number represents the number of consecutive zeros while the second number represents the value between zero-run lengths. For instance, the sequence (0,0,0,0,15) is represented as (4,15) by RLC.

Huffman coding assigns a variable length code to the RLC data, producing a variable length bitstream [56]. As a result, frequently occurring RLC symbols are coded with the least number of bits.

4.5 MPEG Standard

In this section we focus on MPEG-1 as a core and building block of higher versions of the MPEG standard. The MPEG-1 standard supports coding of video and associated audio at a bit rate of about 1.5 Mbps [57]. The MPEG-1 coder-decoder (CODEC) can be described with a generic DCT/DPCM CODEC as shown in Figure 4.1. Different blocks of this CODEC are as follows [57].

1. **Encoder**

   The frame store which contains a reconstructed copy of the previous encoded frame is used as a reference for temporal prediction. The motion estimator calculates motion vectors for each macroblock of the current frame, where a macroblock consists of four 8×8 blocks of luminance samples, one 8×8 block of Cr samples and one 8×8 block of Cb samples. A motion-compensated version of the previous frame is then subtracted from the current frame to create an error frame. Each macroblock of this error frame is then transformed using the DCT and the coefficients are quantized and entropy encoded, together with the motion vectors. At the same time, the quantized coefficients are re-scaled (the IQuant block) and inverse DCT transformed (IDCT) to create a local copy of the encoded and decoded frame. This is used as the prediction reference for the next frame.

2. **Decoder**

   The coded data is entropy decoded and the coefficients are inverse quantized and inverse transformed to recreate the error frame. A motion compensated reference frame is created using the previous decoded frame and the motion vectors for the current frame. The current frame is reconstructed by adding the error frame to this reference frame. This frame is displayed and is also stored in the decoder frame store.

4.5.1 Bidirectional Prediction

MPEG standard employs bidirectional prediction which uses a combination of motion prediction from a previous frame (forward prediction) and motion prediction from a future frame (backward prediction). In order to have future reference frames for prediction, frames are encoded and sent in a order which is different from actual order of frames in natural video sequence. In general, reference frames are encoded first, and frames to be predicted are encoded with regard to the reference frames. This technique eventually results in significant improvement in compression efficiency over forward motion prediction.

There are three main types of coded frame in MPEG-1: I-frames which are intra-coded with no reference to any other frame, P-frames which are inter-coded using only forward prediction from the previous I or P-frames, and B-frames which are inter-coded using bidirectional prediction from the previous I- or P-frame and the next I- or P-frame in the sequence. The three frame classes are grouped together in GOPs (Groups Of Pictures) where a GOP consists of one I-frame followed by a number of P and B-frame.

4.6 Multiple Description Video Coding

In this section, we discuss the extension of multiple description coding to video applications. The design of MD video coder needs to address some challenging issues such as mismatch control [58]. As mentioned earlier, motion-compensated prediction can significantly exploit the temporal redundancy between video frames. Thus, motion estimation information such as motion vectors is a fundamental component in all video coding standards, and most MD video coders also apply motion-compensated prediction. However, whenever packet loss occurs, and an encoder uses a signal for prediction that is not available to the decoder due to packet loss, then mismatch occurs. Efficient predictors are more likely to introduce mismatch, and thus a technical challenge is to balance prediction efficiency with mismatch control. This gives rise to development of different class of predictors which will be explained in the following.

4.6.1 Predictive Multiple Description Video Coding

Consider a generic single-description video coder. The encoder typically forms a prediction of the current frame based on previously encoded frames, and encodes the prediction error to be sent through the packet network. Therefore, if the decoder receives all information, encoder and decoder are able to maintain identical prediction state.

Now consider a predictive MD coder with two channels. The encoder makes two predictions and two descriptions contain prediction error information. At the decoder side, there are three predictive decoders for two side decoders and the central decoder. Depending on which descriptions are received, the decoder ends up in three possible states. However, the encoder does not know which of these states is present at the decoder side. If the encoder uses a predictor that depends on state not available at the decoder, the encoder and decoder states will be mismatched. Several methods were developed to eliminate the potential mismatch by coding the mismatch signal and sending an explicit signal for mismatch reduction, or alternatively partially coding the mismatch. As a result, the possible MD predictors can be categorized into three classes based on the resulting trade off between the redundancy introduced by mismatch coding, and the side distortions caused by mismatch [58].

Class A Predictors with no Mismatch

Class A predictors include video coders that eliminate mismatch using a single predictor. Such a predictor must form its prediction using information that is common to both descriptions. Generally, this results in a predictor that is less efficient than a single-description predictor. For instance, the MD-SNR coder proposed in [59] is built using an H.263 SNR-scalable coder by duplicating the base layer into both descriptions and alternating the enhancement layer between the two descriptions. At the decoder, if only one description is received, only the base layer is used for reconstruction. Since the base layer predictor uses only the base layer information, MD-SNR introduces no mismatch. Redundancy in the MD-SNR codec is controlled by the size of the base layer.

Class B Predictors with no Redundancy

The Class B predictor is basically a single-description predictor. As a result, no extra redundancy is introduced for mismatch coding. However, coders of this class will eventually suffer from mismatch due to packet loss. For instance, Reibman et al. proposed the MD-split method in [60]. This method uses the simplest possible MD algorithm for the MD prediction error encoder which is based on duplication and alternation. Motion vectors are always duplicated, and a varying number of low-frequency coefficients are also duplicated. The number of coefficients to be duplicated can be adapted easily based on the source and channel statistics. This coder will always have mismatch unless all coefficients are duplicated.

Class C Predictors with Partial Mismatch Coding

The coders of class C have flexible redundancy allocation to adapt to varying channel conditions. Wang and Lin [61] proposed MD motion compensation (MDMC) coder. In MDMC, the central predictor forms a linear superposition of the past two reconstructed frames. The MD prediction error encoder uses temporal subsampling such that the side decoders will only receive every other frame. Therefore, without optional mismatch coding, the side decoders will have mismatch. The MDMC encoder [61] has also been designed to include mismatch coding. This makes the MDMC with mismatch coding highly flexible for controlling the introduced redundancy.

Chapter 5

A Multiple Description Video Coding Motivated by Human Visual Perception

5.1 Introduction

In this chapter, we propose a multiple description video coding technique that uses visual distortion criteria to split a one-layer stream generated by a standard video coder into two correlated streams. This method employs smooth block and edge detection in order to evaluate human visual system characteristics and calculate the smoothness and the edge character of the DCT-domain blocks. A complete MD video system should consider jointly optimal multiple descriptions for the side information, motion vectors, and the DCT coefficients. As is well known, motion-compensated prediction can effectively exploit the temporal correlation between video frames. As a result, it is a fundamental component in all video coding standards. Therefore, we

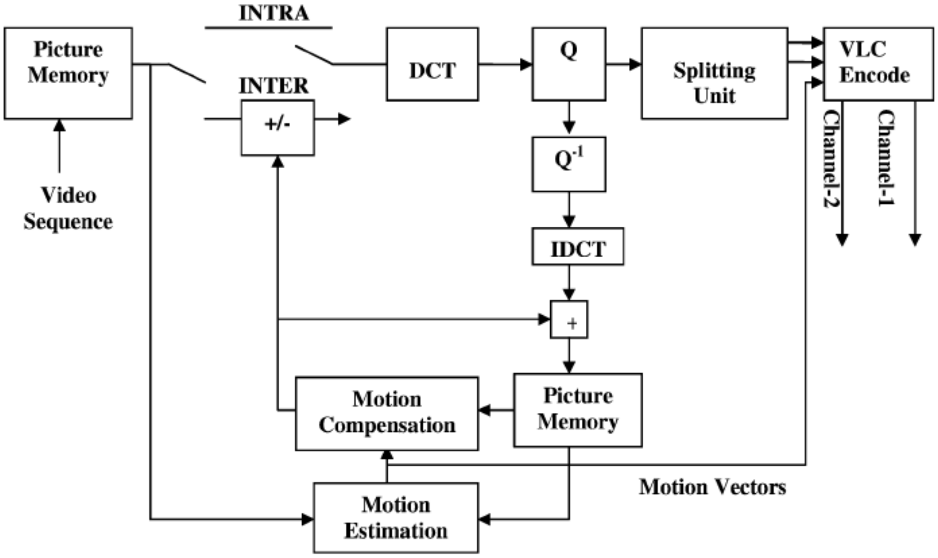


Figure 5.1: General MD video coder with rate splitting.

here take the straightforward strategy of duplicating side information and motion vectors in each side description. We then apply a simple smoothness and edge detection technique in order to determine the amount of associated redundancy of each block of DCT coefficients. Our objective is to maintain the low complexity of the original codec and avoid the extra overhead introduced by manipulating the prefixed quantization tables of the standard video coder. Thus, we keep the core of the standard video coder intact and do not take advantage of the proposed schemes in Chapter 3. The more sophisticated MD video coders are discussed in section 6.2.

The organization of this chapter is as follows. Section 5.2 explains a general MD video coder with rate splitting. Section 5.3 introduces an algorithm for detecting smoothness and edge characteristics of each DCT block and assess the visual distortion caused by rate splitting. Section 5.4 presents the proposed method. Simulation results are provided in Section 5.5. Finally, Section 5.6 concludes this chapter.

5.2 MD Video Coding with Rate Splitting

In general, the objective of a general MD video coding with rate splitting is to encode a video sequence into two video substreams with equal bit rates. Our proposed method employs a human visual perception criterion at the encoder to split the output stream of a MPEG video coder into two correlated substreams as shown in Figure 5.1.

The splitting unit in Figure 5.1 operates on the quantized DCT coefficients. Two descriptions are created by duplicating the header information and motion vectors. Then, the splitting algorithm, donated as splitting unit in Figure 5.1, measures perceptual tolerance against visual distortion in order to determine the appropriate redundancy associated to each block. As a result, some of the lower DCT frequencies, including the DC coefficient, are duplicated in both descriptions. The rest of the coefficients, which represent the higher DCT frequencies, are split between descriptions. The splitting is done so that when a quantized coefficient value is sent to one of the descriptions, a corresponding zero is sent to the other description. This allows us to use standard MPEG CODEC to decode the received streams at side and central decoders.

At the receiver side, if both descriptions arrive at the central decoder, then the coefficients from the two streams are simply merged into a single stream that can be decoded by the MPEG standard decoder. However, if only one of the descriptions arrives at a side decoder, then the received stream is yet simply decoded by the MPEG decoder.

Suppose the bit rates used to send descriptions, in bits per source sample, are $R_1 = R_2$, and the total rate is $R = R_1 + R_2 = 2R_1$. Three situations are possible: both descriptions are received by the MD decoder or either one of the two descriptions is missing. The central decoder receives both descriptions and produces a high-quality reconstruction with central distortion D_0, while the two side decoders each receive only one of the two descriptions and produce lower, but still acceptable, quality reconstructions with side distortions $D_1 = D_2 = D_s$. An SD coder minimizes the distortion D for a fixed total rate R, and its performance is measured by its operational rate-distortion function $R(D)$. One way to measure the efficiency of an MD video coder is by using the redundancy-rate distortion (RRD) curve [39]. We define the distortion of the best single description (SD) coder to be D_0 when R^* bits are used. Then, redundancy is defined to be $\rho = \frac{R-R^*}{R^*}$, where R is the rate when the MD coder has central distortion D_0. Intuitively, ρ is the bit-rate sacrificed compared to the SD coder for the purpose of reducing D_s. The performance of an MD video coder with balanced distortion design is usually characterized by three variables R, D_0 and D_s.

5.3 Determination of Visual Distortion

In this section we introduce a simple method to determine the perceptual tolerance of DCT coefficients against visual distortion. In general, side descriptions of an MD video coding system provide coarse reconstruction of the original data at the side decoders, while central decoder uses each description to refine another one and achieve better performance over side decoders. As a result, video frames may be corrupted by degradations such as noise or blocking artifacts at the side decoders. These sources of degradation arise during generating two coarse descriptions of the original packet, and may have a perceptible effect on visual quality of the side reconstructed . Therefore, there is always a natural trade off between MD coding efficiency and video quality at the side decoders. In general higher redundancy results in better side video quality. Thus, we need to consider HVS characteristics and visual perception in efficiently designing the MD video coding algorithm such that the degradation in visual quality of reconstructed side video streams be as little as possible.

To address this issue, we employ a simple smoothness and edge parameters reported by Dittman et al. in [62], which was originally introduced for the purpose of video watermarking. However, we use these parameters to determine the amount of redundancy of each block of DCT coefficients of prediction errors. Although sophisticated techniques are developed to characterize the human visual perception, the calculation of smoothness and edge parameters of blocks is kept quite simple under this method. This is due to the fact that this calculation is to be done in the MPEG-stream domain and in real-time. We calculate the *Smooth* parameter simply as the number of DCT-coefficients which are not zero after quantization [62]. Thus, high value of Smooth indicates many frequency components in the block, and therefore a large perceptual tolerance against additional distortions introduced through splitting the DCT coefficients.

However, blocks with edge characteristics often have a lot of frequency components too. Thus, a second parameter *Edge* is also introduced to reduce the artifacts. The parameter Edge is calculated as the sum of the absolute values of the DCT coefficients 1, 2, 8, 9, 10, 16, and 17, as marked in the MPEG quantization matrix [62], which represent the lower DCT frequencies. High values in these components indicate that the block is likely to have edge characteristics which need to be preserved. The perceptual tolerance parameter *level* is then defined as [62]

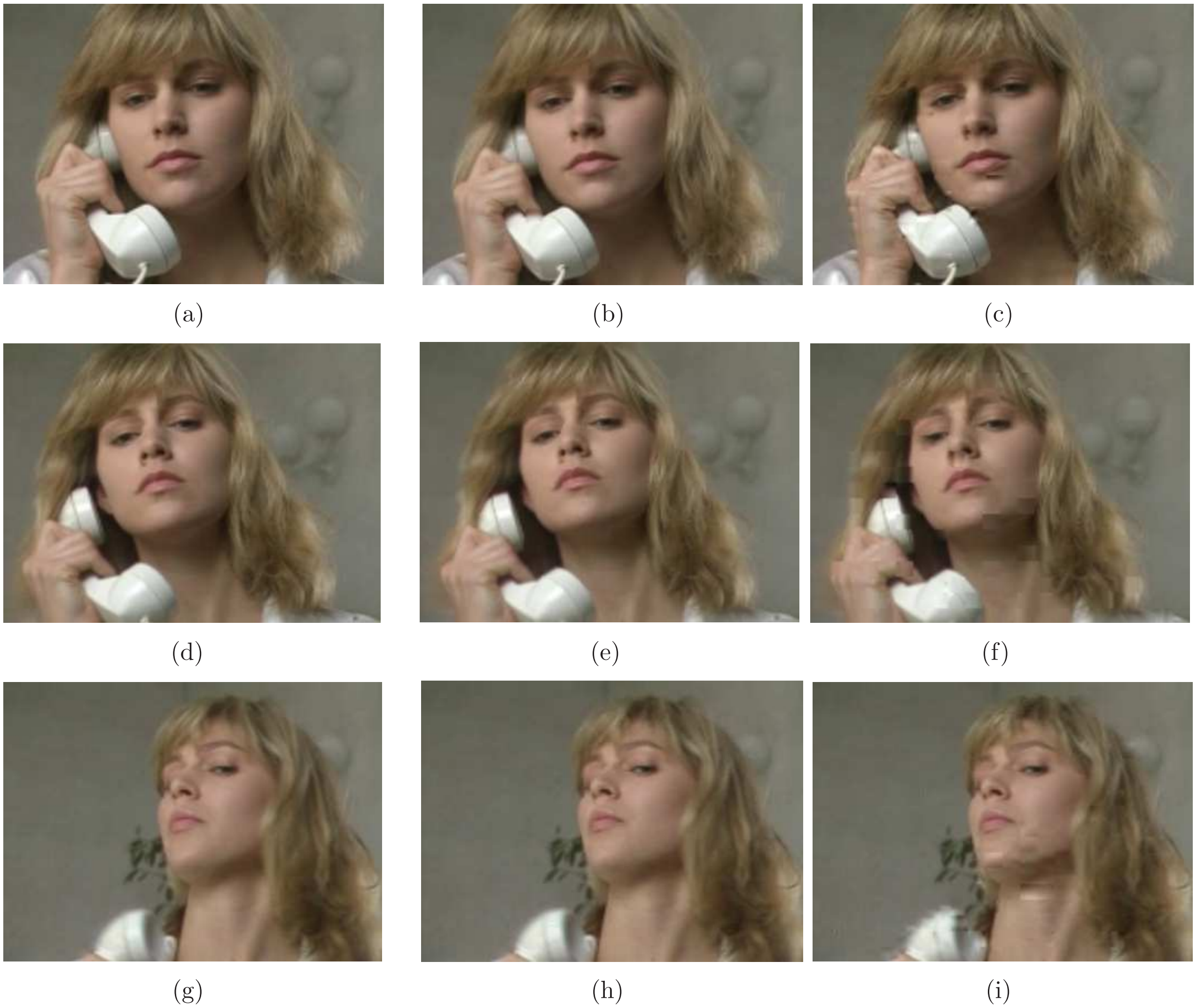
$$L = \max[\min(-10S + 0.27E + 50, 100), 0]. \tag{5.1}$$

where L is Level, S is Smooth, and E is Edge parameter. Min and max operators keep the value of L between 0 and 100.

The constant values in equation (5.1) are evaluated through experiments [62]. The splitting rate factor is then determined based on the computed Level. Generally, the larger values of Level indicate less perceptual tolerance against change of high frequency coefficients of the corresponding DCT block. We have investigated the effect of parameter Level on the rate-splitting strategy. Figure 5.2 shows the perceptual tolerance of DCT blocks of the MPEG video Suzie to the change of high frequency coefficients. In the first experiment, whose results are shown in the right column, 15% of the highest frequency coefficients of the smooth blocks that have higher Level parameter are set to zero. In the second experiment, whose results are shown in the middle column, 55% of the highest frequency coefficients of the blocks with lower Level parameter are set to zero. The left column contain original frames. It is very interesting to observe that even 55% of high-frequency DCT coefficients of the frames on the middle column are set to zero, these frames show considerably less distortion compared with the frames on the right column where only 15% of high-frequency DCT coefficients are set to zero. This implies that the blocks with more frequency components show higher perceptual tolerance against change of coefficients.

5.4 Proposed Rate Splitting Method

In this section we introduce an MD video coding technique based on the method introduced in Section 5.3. This technique applies the perceptual tolerance parameter Level introduced in Section 5.3 in order to determine the visually acceptable redundancy which can be introduced in a specific DCT block. Our proposed technique is



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.2: Perceptual tolerance of DCT blocks against the change of high frequency coefficients in edgy blocks (right column), and in smooth blocks (middle column). Figures on the left column are the original ones. (a),(b),(c) Frame No. 51. (d),(e),(f) Frame No. 87. (g),(h),(i) Frame No. 107.

Table 5.1: Percentage of number of prediction error bits versus motion vectors in MPEG video Foreman with GOP size 18 and in MPEG video Suzie with GOP size 12.

	Prediction Error (%)	Motion Vector (%)
Foreman	97.8%	2.2%
Suzie	97.1%	2.9%

very simple and allows only simple alternation and duplication of the non-zero DCT coefficients produced by a traditional one-layer encoder.

The basic scheme is as follows. As noted earlier, motion vectors constitute a crucial part of the video stream since they are used by the decoder to reconstruct the temporally predicted frames. Table 5.1 compares the percentage of prediction error bits versus motion vector bits in two video sequences. Figure 5.3 demonstrates the effect of loss of motion vectors and loss of the same number of prediction error bits separately. Even though the motion vectors constitute very small portion of video stream, based on motion activity of the temporally predicted frame, the loss of motion vectors results in significant distortion. Inspired by this, we duplicate motion vectors and the side information needed by the decoder in order to identify the received information and determine the mode of operation in each side description. Then, for each block in the frame, we compute the number of non-zero low-frequency DCT coefficients which needs to be duplicated in order to maintain a minimum visual quality. For each DCT block, the Level parameter is computed according to the algorithm discussed in Section 5.3. Then, the number of non-zero DCT coefficients that need to be duplicated is

$$M = N \cdot \left(1 - (1-\rho) \cdot (1 - \left[\frac{L}{100}\right])\right), \tag{5.2}$$

where N indicates the total number of non-zero DCT coefficients of the block, $0 \leq$

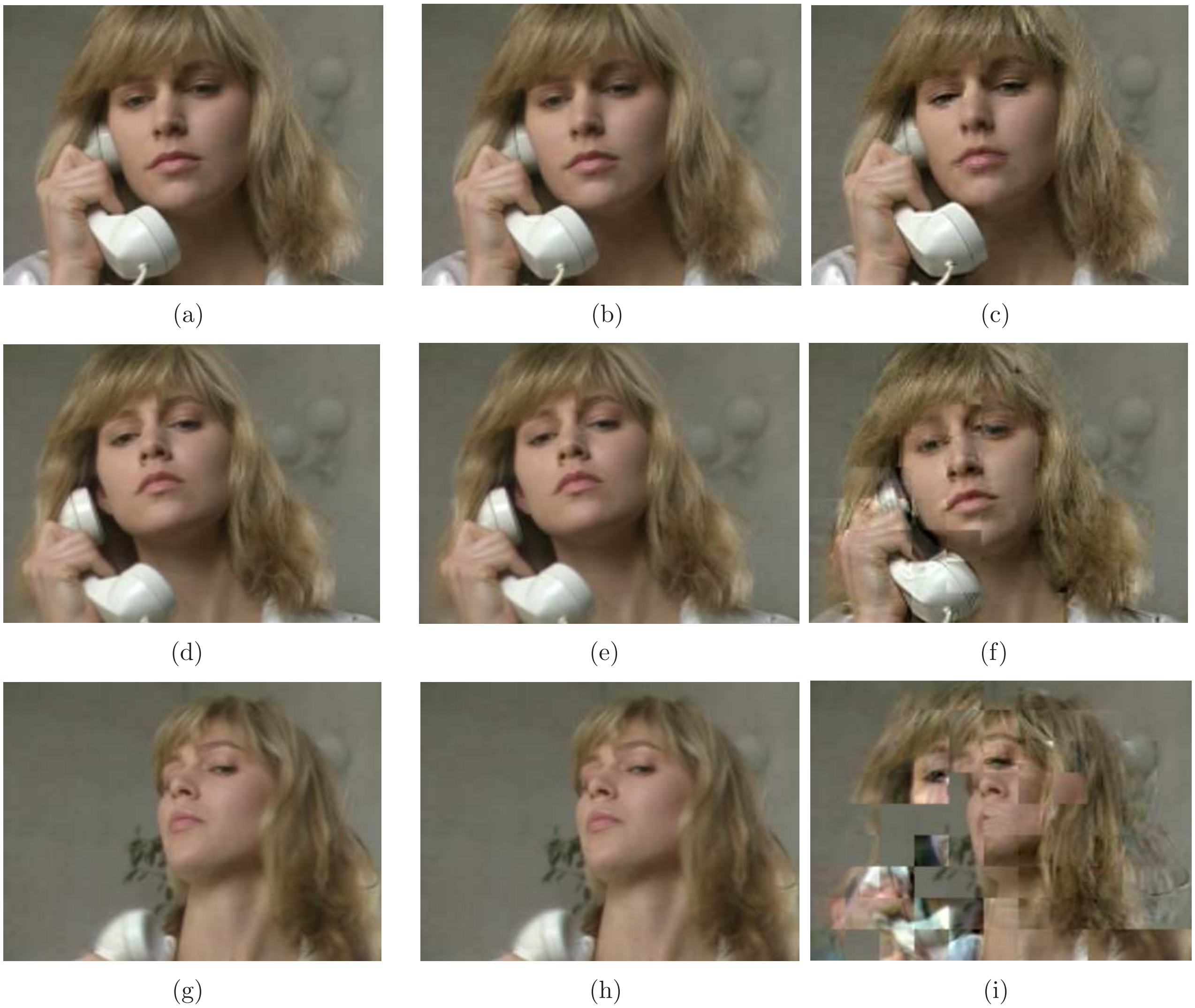


Figure 5.3: Effect of loss of motion vectors and loss of the same amount of prediction error bits on MPEG video Suzie. Frames on the left are the originals, motion vectors of frames on the right are lost, and the same number of prediction error bits in each frame in the middle column is lost too. (a),(b),(c) Frame No. 51. (d),(e),(f) Frame No. 87. (g),(h),(i) Frame No. 107.

$\rho \leq 1$ is the average target redundancy, $[x]$ returns the nearest integer to x, and L is the Level parameter. If we fix the Level value and increase the redundancy ρ, then M increases. On the other hand, if we fix ρ and decrease the Level value, then M decreases. Consequently, we duplicate the first M non-zero low-frequency coefficients into each description, and the remaining $N - M$ coefficients will be distributed in an alternation fashion between the descriptions according to their magnitude. It should be noted that we always duplicate the DC coefficient of each DCT block.

At the decoder, if both descriptions are received, it is a simple matter to merge the coefficients from the two bitstreams either into a single block of coefficients, or into a single bitstream that can be decoded by a standard MPEG decoder. In either case, the exact single-description video can be produced. If only one description is received, that bitstream can simply be decoded using the same method as a standard MPEG decoder.

Our proposed method is summarized in the following steps:

1) Extract each DCT block of prediction errors from MPEG video stream.

2) Compute the Smooth parameter S as the number of the non-zero quantized DCT coefficients. Compute the Edge parameter E as the sum of the absolute values of the DCT coefficients 1, 2, 8, 9, 10, 16, and 17, as marked in MPEG quantization matrix.

3) Calculate the Level parameter $L = \max[\min(-10S + 0.27E + 50, 100), 0]$.

4) Compute the number of non-zero DCT coefficients that need to be duplicated as $M = 64 \times \big(1 - (1 - \rho) \cdot (1 - \big[\frac{L}{100}\big])\big)$, where the number of coefficients of each DCT block is assumed to be 64, ρ is the target redundancy, and $0 \leq \rho \leq 1$.

5) Duplicate the first M non-zero DCT coefficients in two descriptions, and split the remaining $(64 - M)$ DCT coefficients between descriptions.

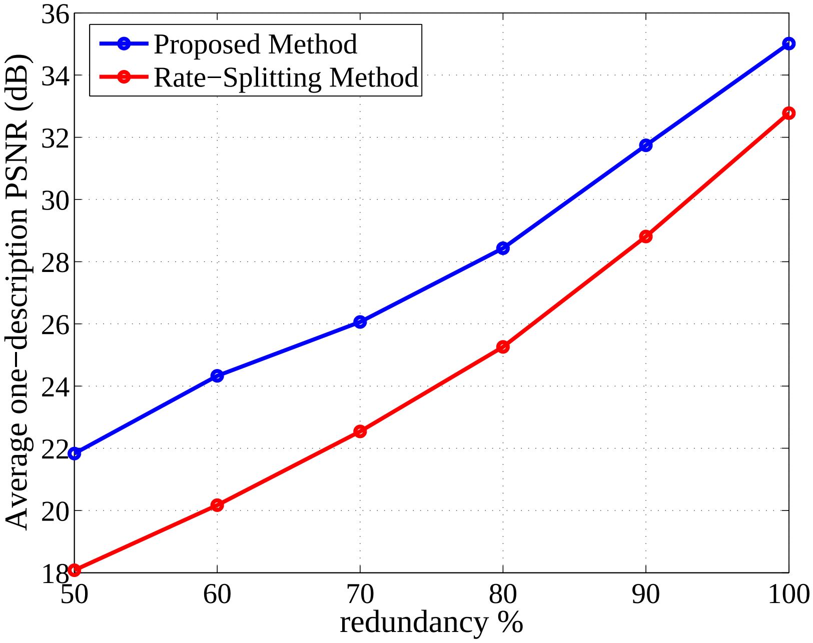
5.5 Simulation Results

We compare our proposed MD split video coder to the simple rate-splitting method introduced in [60] in terms of the single-channel reconstruction performance. In both cases, the coders are built using an MPEG-1 coder. This comparison is made by assuming an entire description is lost; each coder produces identical video when both descriptions are received.

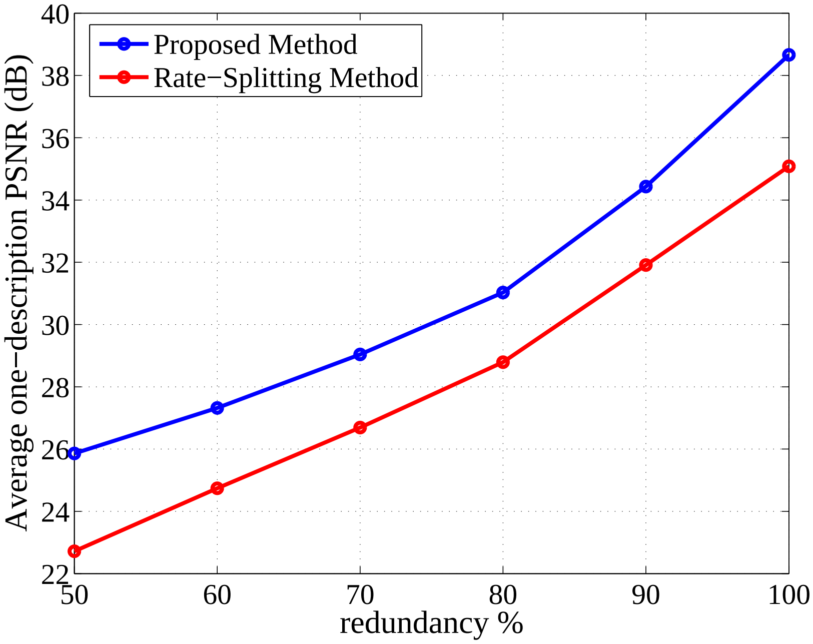
We consider two distortion measures to evaluate and compare the performance of our method. The first distortion measure is the average luminance PSNR across time. We have also evaluated the performance with VQM metric introduced in section 4.3.1. VQM [52] is developed by ITS to provide an objective measurement for perceived video quality. In general, the lower VQM value indicates higher perceptual video quality.

VQM takes the original and impaired video sequences as input and is computed as follows [52]:

- Calibration: This step calibrates the sampled video for feature extraction. It estimates the spatial and temporal shift as well as the contrast and brightness offset of the impaired video sequence with respect to the original video sequence.
- Quality Features Extraction: This step extracts a set of quality features that characterizes perceptual changes in the spatial, temporal, and chrominance properties of video sequence using a mathematical function.

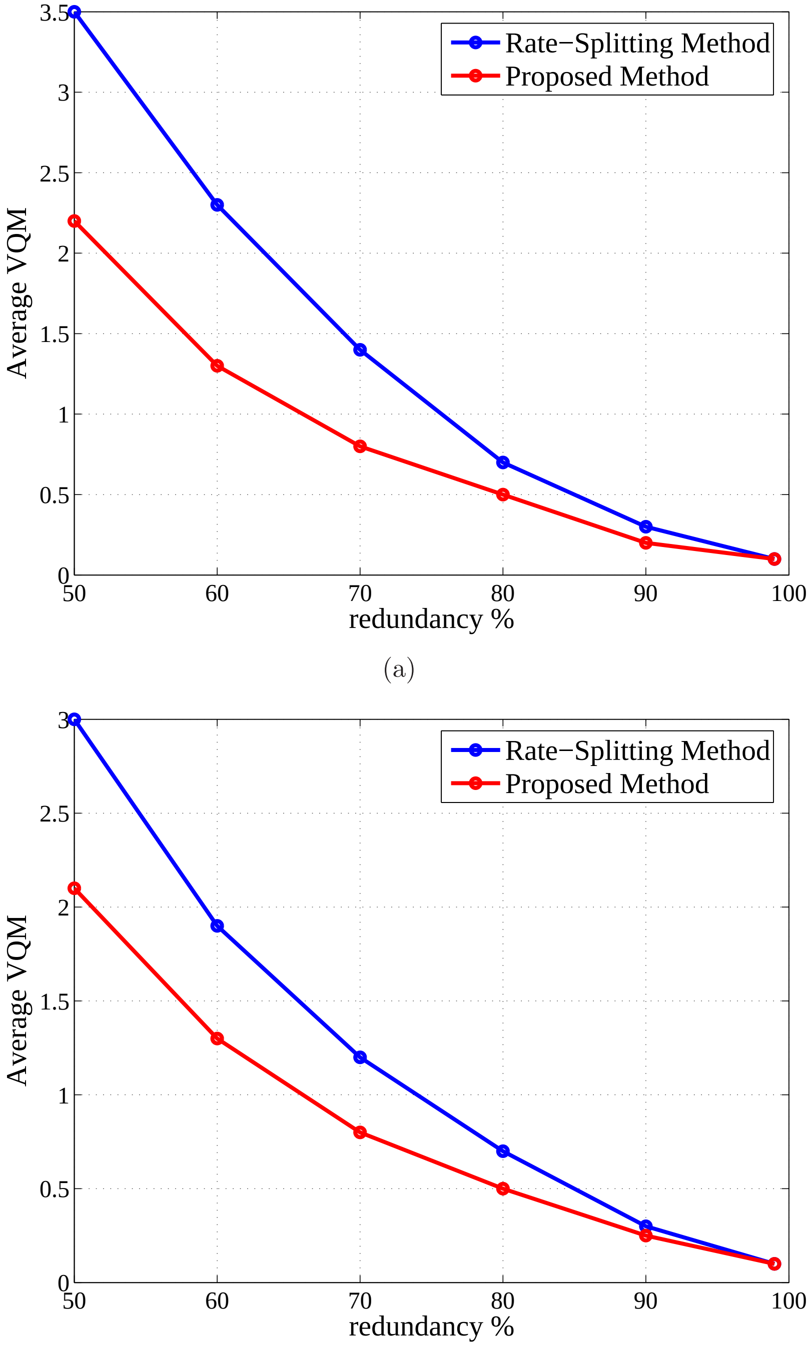


(a)



(b)

Figure 5.4: PSNR performance comparison of proposed method and simple rate-splitting method for (a) Foreman MPEG video, and (b) Suzie MPEG video.



(a)

(b)

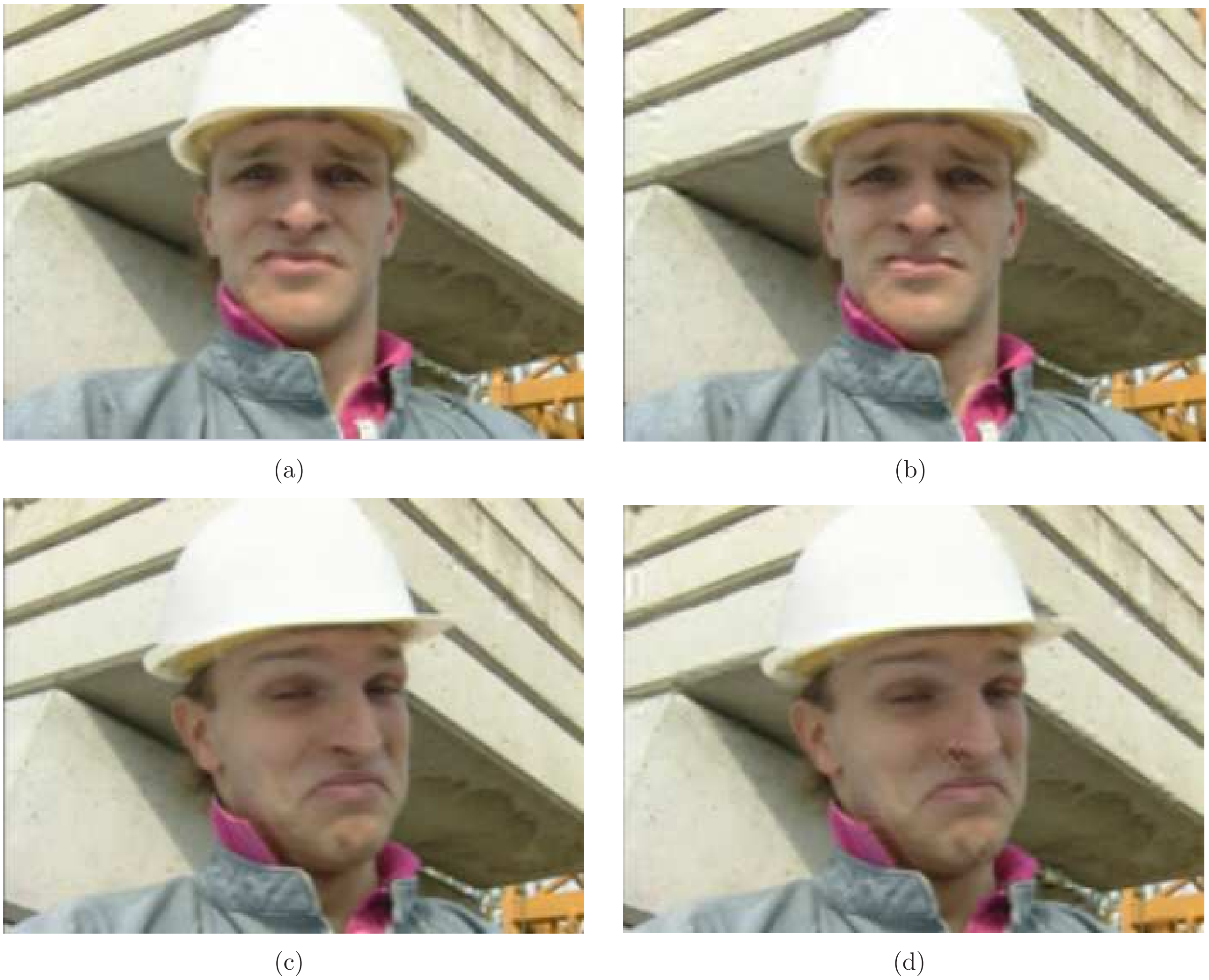
Figure 5.5: VQM performance comparison of proposed method and simple rate-splitting method for (a) Foreman MPEG video, and (b) Suzie MPEG video.

- Quality Parameters Calculation: This step computes a set of quality parameters that describe perceptual changes in video quality by comparing features extracted from the impaired video with those extracted from the original video.

- VQM Calculation: VQM is computed using a linear combination of parameters calculated from previous steps.

The general VQM model uses a linear combination of seven parameters. Four parameters are based on features extracted from spatial gradients of the Y luminance component, two parameters are based on features extracted from the vector formed by the two chrominance components (Cb, Cr), and one parameter is based on contrast and absolute temporal information features, both extracted from the Y luminance component.

The redundancy is expressed in terms of the overhead percentage over the reference total bit rate. The test sequences are two MPEG-1 videos, Foreman and Suzie, with a constant bit rate of 1.5 Mb/s and a frame rate of 25 frames per second and a GOP size of 18 and 12 frames, respectively. Figures 5.4 and 5.5 compare the performance of our proposed method to that of the simple rate-splitting method for the Foreman and Suzie MPEG video sequences in terms of the PSNR and VQM metrics. In both cases, our method outperforms the simple rate-splitting method. It is worth mentioning that since Suzie MPEG video is less edgy, the VQM of proposed method is very close to that of rate-splitting method. This suggests that our proposed method results in better performance when applied to video sequences with edgy frames. For higher redundancy, only very high-frequency DCT coefficients, which have almost imperceptible effect on visual distortion, are split between two descriptions. Consequently, it is expected that our proposed method performs better in lower redundancy,



(a)

(b)

(c)

(d)

Figure 5.6: Reconstruction results for Foreman MPEG video with $\rho = 0.6$. Frames on the left are the results of our proposed methods, and frames on the right column are the results of simple rate-splitting method (a),(b) Frame No. 107. (c),(d) Frame No. 345.

and this is confirmed in Figure 5.5 as well. Figure 5.6 presents subjective results for our proposed method against the simple rate spitting method observed for frames no. 107 and 345 of the Foreman sequence for $\rho = 0.6$ and a target bit rate of 1.5 Mb/s for the encoder.

5.6 Conclusion

In this chapter, we proposed a multiple description video coding technique that uses visual distortion criteria to generate two correlated streams. The main virtue of this technique is low complexity. Our method duplicates motion vectors and some low-frequency DCT coefficients of prediction errors in both descriptions, and then applies a simple smoothness and edge detection algorithm in order to determine the amount of redundancy in each block of DCT coefficients that can be tolerated perceptually. Our simulation results reveal that this MD video coding technique has superior performance in terms of visual quality and average PSNR compared to the traditional rate splitting methods which lack to address perceptual distortion in the design problem.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

We propose two constructive multiple quantization schemes in order to design the codebooks and partitions of side quantizers which are nearest neighbor quantizers. The general scheme is inspired by the multiple description quantization via Gram-Schmidt orthogonalization approach in [9]. This MD quantization technique promises to achieve the achievable rate-distortion region of a memoryless Gaussian source at all rates, and a memoryless non-Gaussian source (at high rates) by subtractive dithering and successive quantization along with quantization splitting. One of the side quantizers in this scheme is a traditional SD vector quantizer, while the input space of the second side quantizer is shifted by the quantization error of the first side quantizer. The basic idea of our proposed MD quantization schemes is to minimize a Lagrangian cost function by an iterative technique which jointly designs side codebooks and consequently forms associated partitions. This Lagrangian function comprises central distortion and target side distortions. The case of balanced distortions is considered in this thesis. In the first scheme, MDVQ-WSC, the central decoder is formed by a linear combination (weighted sum) of the side codebooks. The parameters of this linear combination are also found to minimize the central distortion. Once the side codebooks are found by the iterative technique at the final iteration stage, we propose to replace weighted-sum central decoder with the optimal decoder to enhance the performance of central decoder and achieve lower central distortion. In the second scheme, MDVQ-OC, the central codebook is found by the optimal decoder. This increases the complexity, however, achieves lower central distortion.

We compare the performance of our proposed methods with the best results achieved by the potentially optimum MDVQ proposed by Vaishampayan [16], and also optimal rate-distortion bound for the case of memoryless Gaussian source. We show by simulations that our proposed methods perform very closely to the optimum MD quantizer with considerably less complexity and with a few iterations. As expected, the simulation results also reveal the significant improvement in performance by increasing the block size of the input vectors. An increase in block size results in a greater improvement for sources with memory such as Gauss-Markov source. This implies the significance of increasing the block size for highly correlated sources such as speech and video. Since the tradeoff between central and side distortions can be carefully controlled by selecting the appropriate values for Lagrangian coefficients, our approach results in great design flexibility. We also compare the computational complexity and memory requirements of our proposed methods to those of Vaishampayan's proposed optimum MD vector quantizer. Our proposed methods, and in particular MDVQ-WSC method, significantly reduces the computational complexity and memory requirements.

We also propose a multiple description video coding technique motivated by human visual perception in order to generate two correlated streams. We first introduce two simple parameters to characterize the smoothness and edge features of DCT blocks of a MPEG video frame. We employ these two parameters to measure the perceptual tolerance of DCT blocks against visual distortion. We then duplicate the key information such as motion vectors, and split the remaining DCT coefficients of prediction errors according to the calculated perceptual tolerance parameter. We compare the performance of our proposed method with some standard objective video quality measure such as PSNR and VQM. Our proposed MD video coding method has very low complexity and achieves better performance compared to other techniques which lack to address perceptual distortion in the design problem.

6.2 Future Work

Multiple description quantization by Delta-Sigma modulation seems to be an interesting approach promising to achieve rate-distortion region by appropriate selection of noise-shaping filter. A possible application of this scheme is in MD video coding. The quantization component of this scheme may be replaced by the MPEG standard quantization matrix, and then the DCT coefficients of the prediction errors fed as input to generate correlated coefficients for each description. Since some DCT coefficients, such as DC coefficient, are very sensitive to noise, this might be a draw back to extension of this method to video applications. The solution is to duplicate some low-frequency DCT coefficients, including DC coefficient, and feed the rest of the DCT coefficients as input to generate correlated coefficients for each description.

As mentioned in section 3.3.3, our proposed iterative algorithm does not necessarily produce a non-increasing sequence of Lagrangian values. This implies that there might be an initial codebook for which the iterative algorithm never converge. To address this problem, we simply proposed to limit the number of iterations and pick the codebook at the end of the iteration which has the lowest Lagrangian value. However, this arises a possibility for more sophisticated design approach towards MDVQ problem which results in a convergent design algorithm.

We have simply assumed in our proposed MD video coding technique that each packet contains the entire of each DCT block of prediction errors. Therefore, our rate-splitting method can be extended such that it can handle more general cases with smaller packet size where each DCT block along with its associated motion vector are transmitted in several packets.

Bibliography

[1] A. A. El Gamal and T. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inform. Th.*, vol. 28, pp. 851–857, Nov. 1982.

[2] N. Jayant and S. Christensen, "Effects of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure," *Communications, IEEE Transactions on*, vol. 29, pp. 101–109, Feb. 1981.

[3] N. S. Jayant, "Subsampling of a DPCM speech channel to provide two self-contained half-rate channels," *Bell System Technical Journ.*, vol. 60, pp. 501–509, Apr. 1981.

[4] Q. F. Zhu, Y. Wang, and L. Shaw, "Coding and cell-loss recovery in DCT-based packet video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 248–258, June 1993.

[5] M.-J. Chen, L.-G. Chen, and R.-M. Weng, "Error concealment of lost motion vectors with overlapped motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 560–563, June 1997.

[6] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 771–781, June 1989.

[7] M. Ghanbari, "An adapted h.261 two-layer video codec for ATM networks," *IEEE Transactions on Communications*, vol. 40, pp. 1481–1490, Sept. 1992.

[8] R. Aravind, M. R. Civanlar, and A. R. Reibman, "Packet loss resilience of MPEG-2 scalable video coding algorithms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 426–435, Oct. 1996.

[9] J. Chen, C. Tian, T. Berger, and S. S. Hemami, "Multiple description quantization via GramSchmidt orthogonalization," *IEEE Transactions on Information Theory*, vol. 52, pp. 5197–5217, Dec. 2006.

[10] T. Berger, *Rate Distortion Theory.* Englewood Cliffs, N.J., USA: Prentice-Hall, Inc., 1971. ISBN 13-753103-6.

[11] T. Cover and J. Thomas, *Elements of Information Theory.* New York, NY: John Wiley & Sons, 1991.

[12] L. Ozarow, "On a source-coding problem with two channels and three receivers," *Bell System Technical Journ.*, vol. 59, pp. 1909–1921, Dec. 1980.

[13] H. Feng and M. Effros, "On the achievable region for multiple description source codes on Gaussian sources," in *Information Theory, 2003. Proceedings. IEEE International Symposium on*, June/July 2003.

[14] R. Venkataramani, G. Kramer, and V. K. Goyal, "Multiple description coding with many channels," *IEEE Transactions on Information Theory*, vol. 49, pp. 2106–2114, Sept. 2003.

[15] V. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Proc. Mag.*, vol. 18, pp. 74–93, Sep. 2001.

[16] V. Vaishampayan, “Design of multiple description scalar quantizers,” *IEEE Trans. Inform. Th.*, vol. 39, pp. 821–834, May 1993.

[17] V. A. Vaishampayan and J. Domaszewicz, “Design of entropy-constrained multiple-description scalar quantizers,” *IEEE Transactions on Information Theory*, vol. 40, pp. 245–250, Jan. 1994.

[18] J. C. Batllo and V. A. Vaishampayan, “Asymptotic performance of multiple description transform codes,” *IEEE Transactions on Information Theory*, vol. 43, pp. 703–707, Mar. 1997.

[19] V. A. Vaishampayan and J. C. Batllo, “Asymptotic analysis of multiple description quantizers,” *IEEE Transactions on Information Theory*, vol. 44, pp. 278–284, Jan. 1998.

[20] V. A. Vaishampayan, A. R. Calderbank, and J. C. Batllo, “On reducing granular distortion in multiple description quantization,” in *Information Theory, 1998. Proceedings. 1998 IEEE International Symposium on*, (Cambridge, MA), p. 98, Aug. 1998.

[21] J. Makhoul, S. Roucos, and H. Gish, “Vector quantization in speech coding,” *Proceedings of the IEEE*, vol. 73, pp. 1551–1588, Nov. 1985.

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

[23] Y. Linde, A. Buzo, and R. Gray, “An algorithm for vector quantizer design,” *Communications, IEEE Transactions on [legacy, pre - 1988]*, vol. 28, pp. 84–95, Jan. 1980.

[24] A. Gersho, "On the structure of vector quantizers," *IEEE Transactions on Information Theory*, vol. 28, pp. 157–166, Mar. 1982.

[25] H. Gish and J. Pierce, "Asymptotically efficient quantizing," *IEEE Transactions on Information Theory*, vol. 14, pp. 676–683, Sept. 1968.

[26] J. Conway and N. Sloane, *Sphere Packings, Lattices and Groups.* New York: Springer-Verlag, 3rd ed., 1989.

[27] P. Zador, "Asymptotic quantization error of continuous signals and the quantization dimension," *IEEE Transactions on Information Theory*, vol. 28, pp. 139–149, Mar. 1982.

[28] V. A. Vaishampayan, N. J. A. Sloane, and S. D. Servetto, "Multiple-description vector quantization with lattice codebooks: design and analysis," *IEEE Transactions on Information Theory*, vol. 47, pp. 1718–1734, July 2001.

[29] S. N. Diggavi, N. J. A. Sloane, and V. A. Vaishampayan, "Design of asymmetric multiple description lattice vector quantizers," in *Data Compression Conference, 2000. Proceedings. DCC 2000*, (Snowbird, UT), pp. 490–499, Mar. 2000.

[30] Y. Frank-Dayan and R. Zamir, "Dithered lattice-based quantizers for multiple descriptions," *IEEE Transactions on Information Theory*, vol. 48, pp. 192–204, Jan. 2002.

[31] R. Zamir and M. Feder, "On universal quantization by randomized uniform/lattice quantizers," *IEEE Transactions on Information Theory*, vol. 38, pp. 428–436, Mar. 1992.

[32] R. Zamir and M. Feder, “On lattice quantization noise,” in *Data Compression Conference, 1994. DCC ’94. Proceedings*, (Snowbird, UT), pp. 380–389, Mar. 1994.

[33] C. Tian and S. S. Hemami, “A new class of multiple description scalar quantizer and its application to image coding,” *IEEE Signal Processing Letters*, vol. 12, pp. 329–332, Apr. 2005.

[34] J. Ostergaard and R. Zamir, “Multiple-description coding by dithered delta-sigma quantization,” in *Data Compression Conference, 2007. DCC ’07*, (Snowbird, UT), pp. 63–72, Mar. 2007.

[35] Y. Wang, M. Orchard, and A. Reibman, “Multiple description image coding for noisy channels by pairing transform coefficients,” in *Proc. First IEEE SP Workshop Multim. Signal Processing*, (Princeton, NJ), pp. 419–424, Jun. 1997.

[36] R. M. Gray and D. L. Neuhoff, “Quantization,” *IEEE Transactions on Information Theory*, vol. 44, pp. 2325–2383, Oct. 1998.

[37] M. Orchard, Y. Wang, V. Vaishampayan, and A. Reibman, “Redundancy rate-distortion analysis of multiple description coding using pairwise correlating transforms,” in *Proc. IEEE Int. Conf. Image Proc.*, (Santa Barbara, CA), pp. 608–611, Oct. 1997.

[38] Y. Wang, M. T. Orchard, and A. R. Reibman, “Optimal pairwise correlating transforms for multiple description coding,” in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1, (Chicago, IL), pp. 679–683, Oct. 1998.

[39] Y. Wang, M. T. Orchard, V. Vaishampayan, and A. R. Reibman, "Multiple description coding using pairwise correlating transforms," *IEEE Transactions on Image Processing*, vol. 10, pp. 351–366, Mar. 2001.

[40] V. Goyal and J. Kovačević, "Optimal multiple description transform coding of Gaussian vectors," in *Proc. Data Compr. Conf.*, (Snowbird, UT), pp. 388–397, Mar. 1998.

[41] V. Goyal and J. Kovačević, "Generalized multiple description coding with correlated transforms," *IEEE Trans. Inform. Th.*, vol. 47, pp. 2199–2224, Sep. 2001.

[42] C. Tian and S. S. Hemami, "Universal multiple description scalar quantization: analysis and design," *IEEE Transactions on Information Theory*, vol. 50, pp. 2089–2102, Sept. 2004.

[43] S. N. Diggavi, N. J. A. Sloane, and V. A. Vaishampayan, "Asymmetric multiple description lattice vector quantizers," *IEEE Transactions on Information Theory*, vol. 48, pp. 174–191, Jan. 2002.

[44] V. K. Goyal, J. A. Kelner, and J. Kovacevic, "Multiple description vector quantization with a coarse lattice," *IEEE Transactions on Information Theory*, vol. 48, pp. 781–788, Mar. 2002.

[45] C. Tian and S. S. Hemami, "Optimality and suboptimality of multiple-description vector quantization with a lattice codebook," *IEEE Transactions on Information Theory*, vol. 50, pp. 2458–2470, Oct. 2004.

[46] W. Y. Chan, S. Gupta, and A. Gersho, "Enhanced multistage vector quantization by joint codebook design," *IEEE Transactions on Communications*, vol. 40, pp. 1693–1697, Nov. 1992.

[47] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, vol. 37, pp. 31–42, Jan. 1989.

[48] M. Effros, "Distortion-rate bounds for fixed- and variable-rate multiresolution source codes," *IEEE Transactions on Information Theory*, vol. 45, pp. 1887–1910, Sept. 1999.

[49] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers [speech coding]," *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, vol. 36, pp. 1445–1453, Sept. 1988.

[50] V. Bhaskaran and K. Konstantinides, *Image and Video Coding Standards - Algorithms and Architectures.* Kluwer Academic Publishers, 2nd ed., 1997.

[51] J. Mannos and D. Sakrison, "The effects of a visual fidelity criterion of the encoding of images," *IEEE Transactions on Information Theory*, vol. 20, pp. 525–536, July 1974.

[52] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting*, vol. 50, pp. 312–322, Sept. 2004.

[53] T. Jarske and Y. Neuvo, “Adaptive DPCM with median type predictors,” in *Consumer Electronics, IEEE Transactions on*, vol. 37, (Rosemont, IL), pp. 348–352, Aug. 1991.

[54] N. M. Nasrabadi and R. A. King, “Image coding using vector quantization: a review,” *IEEE Transactions on Communications*, vol. 36, pp. 957–971, Aug. 1988.

[55] Q. Wang and R. Clarke, “Motion estimation and compensation for image sequence coding,” *Signal Processing: Image Communication*, vol. 4, pp. 161–174, Apr. 1992.

[56] Y. Wang, A. R. Reibman, and S. Lin, “A method for the construction of minimum redundancy codes,” *Proceedings of the IRE*, vol. 40, pp. 1098–1101, Jan. 1952.

[57] ISO/IEC 11172-2 (MPEG-1 Video), “Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s,” Nov. 1991.

[58] Y. Wang, A. R. Reibman, and S. Lin, “Multiple description coding for video delivery,” *Proceedings of the IEEE*, vol. 93, pp. 57–70, Jan. 2005.

[59] A. R. Reibman, H. Jafarkhani, Y. Wang, M. T. Orchard, and R. Puri, “Multiple description coding for video using motion compensated prediction,” in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, vol. 3, (Kobe), pp. 837–841, Oct. 1999.

[60] A. Reibman, H. Jafarkhani, Y. Wang, and M. Orchard, "Multiple description video using rate-distortion splitting," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 1, (Thessaloniki), pp. 978–981, Oct. 2001.

[61] Y. Wang and S. Lin, "Error-resilient video coding using multiple description motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 438–452, June 2002.

[62] J. Dittmann, M. Stabenau, and R. Steinmetz, "Robust MPEG video watermarking technologies," in *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*, (New York, NY, USA), pp. 71–80, ACM Press, 1998.