Generalized Multiple Description Vector Quantization *

Michael Fleming Michelle Effros

Abstract

Packet-based data communication systems suffer from packet loss under high network traffic conditions. As a result, the receiver is often left with an incomplete description of the requested data. Multiple description source coding addresses the problem of minimizing the expected distortion caused by packet loss. An equivalent problem is that of source coding for data transmission over multiple channels where each channel has some probability of breaking down. Recent work in practical multiple description coding explores the design of multiple description scalar and vector quantizers for the case of two channels or packets. This paper presents a new practical algorithm, based on a ternary tree structure, for the design of both fixed- and variable-rate multiple description vector quantizers for an arbitrary number of channels. Experimental results achieved by codes designed with this algorithm show that they perform well under a wide range of packet loss scenarios.

I Introduction

The multiple description source coding problem is the problem of data compression for applications where one or more increments of a binary source description may be lost during transmission. The goal of multiple description source code design is to achieve a code that yields good rate-distortion performance under a wide variety of information loss scenarios. As an example, consider the system shown in Figure 1, in which an image is described in three increments or "packets" of rates R_1, R_2, and R_3 bits per symbol respectively.[1] The decoder of this multiple description system must be able to reconstruct the data sequence for any combination of received data increments. If only the first increment of the binary description is received, then the decoder builds a source reconstruction with the first increment's rate-R_1 description; if the first and third increments are received, then the decoder reconstructs the image

*M. Fleming and M. Effros are with the Department of Electrical Engineering, MC 136-93, California Institute of Technology, Pasadena, CA 91125. This material is based upon work supported by the Pickering Fellowship, NSF Grant No. MIP-9501977, the Intel Technology for Education 2000 program, and a grant from the Charles Lee Powell Foundation.

[1] The number of increments shown here is helpful as a simple example but not necessarily representative. In most of the applications described in this work, the number of increments would be far greater than three, and the above explanation would apply to small subsets of an image rather than an entire image.

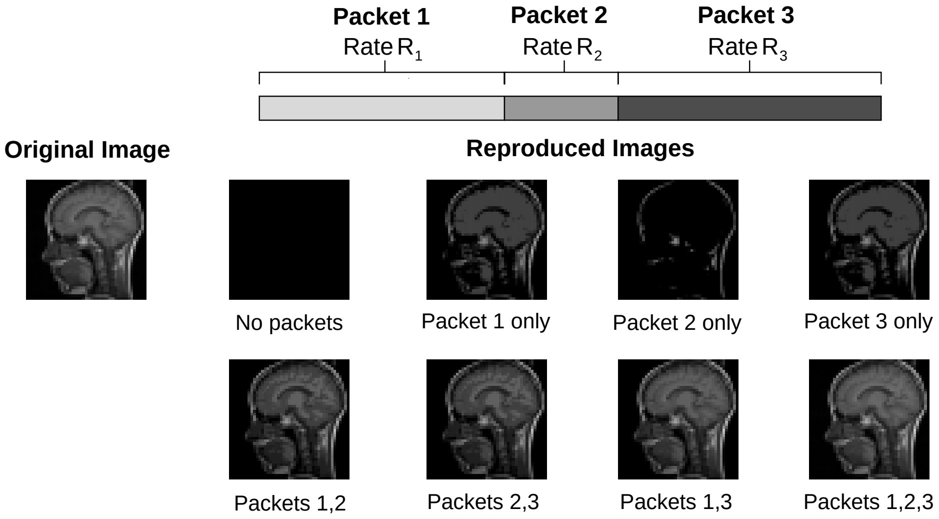

Figure 1: The concept behind a multiple description code. Each increment or packet contains a low-rate description of the source data. Combining the information from different increments results in a higher rate description of the data.

with the resulting rate-$(R_1 + R_3)$ binary description, and so on. Ideally, we might hope that if r bits per symbol are received then the image reconstruction would achieve distortion $D(r)$, where $D(\cdot)$ denotes the source's distortion-rate function. This distortion would in fact be achievable (at least in theory) if the pattern of lost and received increments were fixed and known at design time. In multiple description coding, however, the increment losses are assumed to be unknown. To combat these potential losses, we might try to transmit the same information in all increments to ensure some minimum quality even if only one increment is received. However, if the increments contain the same information and more than one increment is received, then much of the information is redundant. An alternative approach is to divide a traditional source code's description into non-overlapping increments. This alleviates the redundancy problem and achieves good performance if all increments are received, but fails entirely if even one increment is lost. The design problem for a multiple description code is a problem of combining the ideas from these two solutions to optimize the code with respect to either a distribution function or a priority schedule over the full range of packet loss scenarios.

Multiple description source codes can be applied in a wide variety of data communications and storage systems. Examples include packet-based communications systems, diversity channel coding applications, and both traditional and distributed data storage systems. A brief description of some of these applications follows.

Modern data communications, including communications over wired systems like the internet and wireless systems such as wireless paging, fax, and modem technology, are typically implemented using packet-based communication protocols. In packet-based systems, the information to be transmitted is broken into small packets, each of which is time-stamped, protected from channel noise, and independently transmitted over the channel. As the network traffic for a packet-based system increases, the probability of packet loss or delay also increases. Existing packet-based networks use a variety of methods to combat packet loss. Many networks employ a feedback system so that the sender is alerted by the receiver whenever packet loss occurs, and lost packets are simply retransmitted. Unfortunately, this solution requires a reverse channel and can involve a significant delay as the receiver requests the retransmission, receives an acknowledgment, and then receives the retransmission. In situations where a reverse channel is unavailable or delay is unacceptable (e.g., real-time video transmission), the retransmission of lost packets may not be feasible. Packet loss effects can also be reduced through error correcting techniques at the cost of increasing the required description rate. Using a maximum distance separable (MDS) code such as a Reed-Solomon code, transmitting an additional t error-correcting packets together with the data allows any combination of up to t lost packets to be completely reconstructed by the receiver. However, these codes are optimized for exactly t packet losses; if fewer packets are lost, then they do not perform better, and if more packets are lost, then the code may suffer a sharp degradation in performance. Thus coding schemes that achieve good performance under all packet loss scenarios without the need for a reverse communications link are of great practical use. Multiple description codes, which allow source reconstruction under the full range of packet loss conditions, are well-suited to this problem.

Multiple description source codes are also useful for diversity communications systems, where information is transmitted from a single sender to a single receiver over multiple communication channels. For example, consider a diversity system with two channels. The multiple description problem posed in such a framework is this: What information should the sender transmit over each channel so as to minimize information loss should one of the channels break down? Given a multiple description code designed for two data increments, we send one increment over each channel. If both channels work, then the combination of the information from the two channels allows the data to be reproduced with a very low distortion; if only one channel works then the data can still be reproduced, although with a higher distortion.

In both traditional and distributed data storage systems, the use of multiple description codes would provide a means of achieving greater robustness to memory failures. Using a multiple description code and an appropriate strategy for distributing data across partitions or subsystems, data may be recovered (to some accuracy) even when a disk partition fails or some of the hosts in a distributed data storage system are unavailable.

Finally, multi-resolution source coding [1, 2], a special case of multiple description coding, has a variety of uses for applications such as mobile communications and the world wide web. A multi-resolution code describes a source at a variety of rates. The coded information is organized so that the initial increment of the binary description yields a low-resolution data reproduction. Adding more and more increments improves the reproduction quality. It is assumed that no packet loss occurs, but that if the receiver is satisfied with the quality of the reproduction after ℓ increments have been decoded, then later increments are ignored. Code design uses some prior distribution or priority function (with respect to the values of ℓ) based on a profile of the intended receiver(s). This situation is analogous to a multiple description scenario in which packets $1, \ldots, \ell$ are transmitted successfully and subsequent packets are lost,

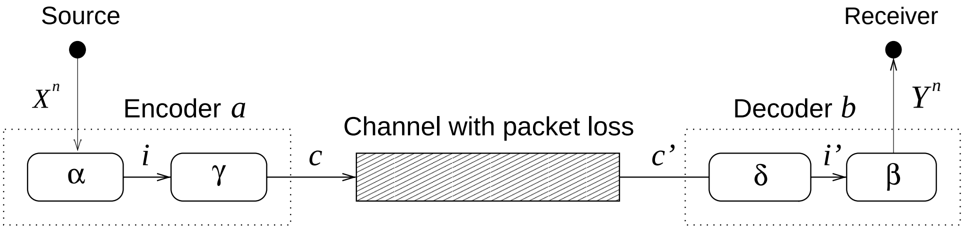

Figure 2: A variable-rate packet-based communications system.

and the same priority schedule on ℓ can be applied.

Previous research into multiple description coding has been performed in both theoretical and practical areas. Most of the literature focuses specifically on the two-channel problem. Derivations of the achievable rate-distortion bounds for data transmission over two channels are presented in [3, 4, 5, 6]. Practical research includes algorithms for the design of multiple description scalar quantizers for data transmission over two channels using both fixed-rate [7] and variable-rate [8] schemes. A design algorithm for two-channel multiple description vector quantizers appears in [9].

This paper presents a new algorithm for the design of multiple description vector quantizers (MDVQs). Our work differs from the work in [9] in that our MDVQ design is generalized so as to consider an arbitrary number of packets or channels rather than just the two channel case. This generalization employs a new ternary tree coding structure, and makes possible MDVQ coding for packet-based networks and large diversity systems, where the number of packets or channels is typically much greater than two. Furthermore, this work extends MDVQ design from fixed-rate codes with known constraints on the distortion achieved by each single-packet description to fixed- or variable-rate codes with arbitrary constraints on any subset of rates and distortions.

The paper is organized as follows. A mathematical description of the encoder, decoder, and channel is developed in Section II, and includes an explanation of the new ternary tree structure on which our MDVQ is based. The algorithm design itself is presented in Section III, and the experimental results appear in Section IV. The key contributions of the paper are summarized in Section V.

II Multiple Description Coding

A general variable-rate packet-based communication system, as shown in Figure 2, consists of an encoder, an information channel, and a decoder. The encoder blocks the source $\{X_i\}$ into vectors of length n. Each vector $X^n \in \mathcal{X}^n$ is represented by a channel codeword $c \in \mathcal{C}$, which is transmitted through the information channel. We assume that the channel is noiseless, but allow for the possibility that packet loss may occur, in which case only part of the transmitted codeword is received. We also assume that the decoder knows which packets it receives, and which are lost. The decoder takes each received channel codeword or partial codeword $c' \in \mathcal{C}'$, and maps it to a reproduction vector $Y^n \in \mathcal{Y}^n$.

Following the development in [10], we denote the encoder and decoder by $\mathbf{a} : \mathcal{X}^n \to \mathcal{C}$

and $\mathbf{b} : \mathcal{C}' \to \mathcal{Y}^n$ respectively. The encoder comprises two parts, $\mathbf{a} = \gamma \circ \alpha$, where $\alpha : \mathcal{X}^n \to \mathcal{I}$ and $\gamma : \mathcal{I} \to \mathcal{C}$. The set $\mathcal{I}$ is the index set of the codebook $\mathcal{C}$. The function α maps each source vector to a channel codeword index $i \in \mathcal{I}$, and the function γ maps this index to the corresponding channel codeword $c_i \in \mathcal{C}$.

The decoder also comprises two parts, $\mathbf{b} = \beta \circ \delta$, where $\delta : \mathcal{C}' \to \mathcal{I}'$ and $\beta : \mathcal{I}' \to \mathcal{Y}^n$. The set $\mathcal{I}'$ is an index set enumerating all possible received channel codewords $c' \in \mathcal{C}'$. The function δ converts each received channel codeword c' to an index $i' \in \mathcal{I}'$, and the function β maps this index i' to a reproduction vector $Y^n \in \mathcal{Y}^n$. For simplicity, we assume that $\mathcal{C} \subset \mathcal{C}'$, $\mathcal{I} \subset \mathcal{I}'$, and that for any $i \in \mathcal{I}$, $\delta\left(\gamma\left(i\right)\right) = i$. Thus when no packet loss occurs, $c' \in \mathcal{C}$. If one or more packets is lost then $c' \in \mathcal{C}' - \mathcal{C}$ and $\delta\left(c'\right) \in \mathcal{I}' - \mathcal{I}$ since the decoder cannot uniquely determine the index i of the transmitted codeword.

Given an L-packet system, the pattern of packet losses can be described using a packet transmission vector $\mathbf{T} = (T_1, T_2, \ldots, T_L)$, where, for each $\ell \in \{1, \ldots, L\}$,

$$T_\ell = \begin{cases} 1 & \text{if packet } \ell \text{ is transmitted successfully} \\ 0 & \text{if packet } \ell \text{ is lost.} \end{cases}$$

The set of all transmission vectors is denoted by $\mathcal{T} = \{0, 1\}^L$. We associate with each packet transmission vector $\mathbf{T}$ a probability $p_{\mathbf{T}}$ for that transmission sequence, and we define the channel packet transmission function $\nu : \mathcal{C} \times \mathcal{T} \to \mathcal{C}'$ as the function that maps the transmitted channel codeword $c \in \mathcal{C}$ to the received channel codeword $c' \in \mathcal{C}'$ for a given $\mathbf{T} \in \mathcal{T}$.

The source codebook $\{\beta\left(i'\right) : i' \in \mathcal{I}'\}$ can be represented by a depth-L ternary tree as shown in Figure 3. In this tree, each leaf represents a single source codeword $\beta\left(i'\right)$ for some $i' \in \mathcal{I}'$. There are no codewords at the internal nodes. The sequence of L steps describing the unique path from the root to leaf i' is denoted $\phi^L\left(i'\right) = \left(\phi_1\left(i'\right), \phi_2\left(i'\right), \ldots, \phi_L\left(i'\right)\right)$. Considering the most general case, we treat each step in the path description of a source codeword leaf as an individual packet. The leaves associated with zero packet losses ($i' \in \mathcal{I}$) appear at the deepest level of the tree. Reproductions associated with all other packet loss scenarios are found at the remaining leaves, where each horizontal path denotes a packet loss, and each (left or right) downward path represents a successfully received packet. Thus the depth of a codeword in the tree specifies the number of packet losses associated with that codeword.

In general, the encoder will be a variable-rate system. We can obtain an expression for the received rate associated with the channel codeword of index i and a transmission vector $\mathbf{T}$ by summing the received incremental rates along the path $\phi^L\left(i\right)$. Let $R_\ell(i)$ be the incremental rate for encoding $\phi_\ell\left(i\right)$, the ℓth step of this path. Then the total received rate associated with channel codeword i and transmission vector $\mathbf{T}$ is

$$R_{\mathbf{T}}\left(i\right) = \sum_{\ell=1}^{L} R_\ell(i) T_\ell.$$

The performance of the code is judged in terms of the distortion of the reproductions as well as the required description rates. The per symbol distortion correspond-

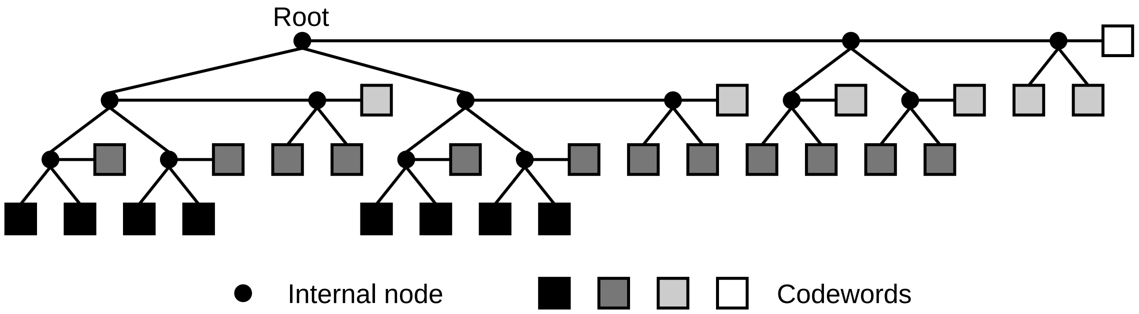

Figure 3: The ternary tree of codewords for a three packet or three channel system. Horizontal branches indicate packet loss.

ing to a particular transmission vector $\mathbf{T}$ is defined as

$$D_{\mathbf{T}} = \frac{1}{n}\mathrm{E}_{X^n}\left[d\left(X^n, Y^n\right)\right] = \frac{1}{n}\mathrm{E}_{X^n}\left[d\left(X^n, \mathbf{b}\left(\nu\left(\mathbf{a}\left(X^n\right), \mathbf{T}\right)\right)\right)\right].$$

In the multiple description problem as posed above, there are a large number of different distortions corresponding to different combinations of packets used in the reproduction. In general, there are 2^L different reproduction distortions for a single source vector coded into L packets. A data sequence is typically coded in small subsets, each of which may experience different packet losses. In such a scenario it becomes infeasible to evaluate the performance over all possible sequences of packet loss vectors $\mathbf{T}$. It makes more sense to view the distortion-rate relationship in the context of transmission over multiple channels. Then there are a total of 2^L possible distortion-rate pairs to consider. The description rate corresponding to each of these distortions is the sum of the description rates of the individual channels used in the reproduction. Let $\mathbf{D}$ be a vector of 2^L distortions and $\mathbf{R}$ be a vector of 2^L corresponding rates. The set of distortion-rate pairs $(\mathbf{D}, \mathbf{R})$ achievable by an MDVQ defines some region $\mathcal{S}$ in the 2^{L+1}-dimensional distortion-rate space. For the multiple channel case, $\mathcal{S}$ must be convex (this can be shown by a time-sharing argument[2]), and thus is entirely characterized by its support functional,

$$J = \mathrm{E}\left[\sum_{\mathbf{T}\in\mathcal{T}}\left(\mu_{\mathbf{T}} D_{\mathbf{T}} + \lambda_{\mathbf{T}} R_{\mathbf{T}}\right)\right],$$

where $D_{\mathbf{T}}$ and $R_{\mathbf{T}}$ denote the distortion and the rate associated with transmission vector $\mathbf{T}$, and $\mu_{\mathbf{T}}$ and $\lambda_{\mathbf{T}}$ are the associated Lagrangian constants. The MDVQ design algorithm is based on the minimization of this functional.

Just as the performance measure $D + \lambda R$ may be interpreted as a Lagrangian for minimizing the distortion subject to a constraint on the rate, the weighted performance measure J has a variety of interpretations dependent on the choice of the $\mu_{\mathbf{T}}$ and $\lambda_{\mathbf{T}}$. In particular, J may be viewed as a Lagrangian for minimizing: (1) the weighted sum of distortions subject to a collection of constraints on the rates, (2)

[2] Note that the time-sharing argument may break down for packet-based systems with fixed packet sizes since the finite packet size effectively precludes the use of infinite vector dimensions.

the weighted sum of rates subject to a collection of constraints on the distortions, or (3) the weighted sum of rate-distortion Lagrangians (at the same or differing slopes). In fact, minimizing J can be viewed as a minimization of any combination of rates, distortions, or Lagrangians subject to constraints on the remaining quantities. Two examples are described below:

1. *Minimizing the expected distortion over all packet loss scenarios.* For a fixed-rate system, setting $\mu_{\mathbf{T}} = p_{\mathbf{T}}$ and $\lambda_{\mathbf{T}} = 0$ minimizes the total expected distortion with respect to the distribution imposed by the packet loss probabilities.

2. *Multi-resolution coding.* In a a simple three-packet multi-resolution code, the set of possible transmission vectors $\mathcal{T}_{MR}$ is: $\{(0,0,0),(1,0,0),(1,1,0),(1,1,1)\}$. These vectors correspond to the receiver stopping after reading 0, 1, 2, and 3 packets respectively. By setting $\mu_{\mathbf{T}} = 0$ for $\mu_{\mathbf{T}} \notin \mathcal{T}_{MR}$ we reduce the multiple description problem to the multi-resolution problem as discussed in [1]. The $\lambda_{\mathbf{T}}$ and non-zero $\mu_{\mathbf{T}}$ parameters are chosen to implement some priority schedule.

III Algorithm

The aim of the algorithm is to minimize the Lagrangian functional

$$J(\alpha,\gamma,\delta,\beta) = \mathrm{E}_{X^n}\left[\sum_{\mathbf{T}\in\mathcal{T}}\left[\mu_{\mathbf{T}} d\left(X^n, \mathbf{b}\left(\nu\left(\mathbf{a}\left(X^n\right),\mathbf{T}\right)\right)\right) + \lambda_{\mathbf{T}} R_{\mathbf{T}}\left(\alpha\left(X^n\right)\right)\right]\right].$$

This functional combines a weighted sum of rates and distortions over all packet loss scenarios. By using this measure in both code design and implementation, we can optimize the entire multiple description code with respect to the distribution $\{p_{\mathbf{T}}\}$ or to a priority schedule of our choosing. The design of the MDVQ employs an iterative descent algorithm similar to the generalized Lloyd algorithm. It generates a sequence of codes for which the Lagrangian measure is non-increasing, and, since the Lagrangian is bounded below by zero, the algorithm is guaranteed to converge to a locally optimal solution. The function δ is uniquely determined by the codebook and index set of γ. Hence the optimization need only consider the functions α, γ, and β. The steps performed at each iteration individually optimize first the encoder, followed by the prefix code, and finally the decoder. Each of these steps is described in detail below. The algorithm iterates until convergence.

Optimizing the encoder

Let f be the function that maps the index of the transmitted codeword to the index of the received codeword

$$f(i,\mathbf{T}) = \delta\left(\nu\left(\gamma\left(i\right),\mathbf{T}\right)\right).$$

For a given γ and β, the optimal encoder α^* is the encoder that maps each source vector x^n to the "closest" channel codeword index, where the optimal $\alpha^*\left(x^n\right)$ is given by

$$\alpha^*\left(x^n\right) = \arg\min_{i\in\mathcal{I}}\left\{\sum_{\mathbf{T}\in\mathcal{T}}\left[\mu_{\mathbf{T}} d\left(x^n, \beta\left(f\left(i,\mathbf{T}\right)\right)\right) + \lambda_{\mathbf{T}} R_{\mathbf{T}}\left(i\right)\right]\right\}. \qquad (1)$$

Optimizing the prefix code

For a given α and β, an optimal prefix code γ is one that minimizes the expected description length. For any $i \in \mathcal{I}$, we describe index i with a sequence of entropy codes matched to the probabilities of each step in $\phi^L\left(i\right)$.

Optimizing the decoder

For a given α and γ, the optimal decoder β^* is the decoder that minimizes the expected reproduction distortion given the received channel codeword index $i' \in \mathcal{I}'$. Thus

$$\beta^*\left(i'\right) = \arg\min_{y^n\in\mathcal{Y}^n}\left\{\mathrm{E}_{X^n}\left[d\left(X^n, y^n\right)\left|f\left(\alpha\left(X^n\right), \mathbf{T}\right) = i'\right]\right\}. \qquad (2)$$

Note that the value of $\mathbf{T}$ is known since $\phi^L\left(i'\right)$ specifies a unique packet transmission sequence. Let $\mathcal{M} \subset \mathcal{I}$ be the set of indices of the full channel codewords that could have given rise to the received partial channel codeword of index i', i.e. $\mathcal{M} = \{i : f(i, \mathbf{T}) = i'\}$. Then the expected value that we are trying to minimize can be written as

$$\begin{aligned}&\mathrm{E}_{X^n}\left[d\left(X^n, y^n\right)\left|f\left(\alpha\left(X^n\right), \mathbf{T}\right) = i'\right]\right. \\ &= \sum_{i\in\mathcal{M}}\mathrm{E}_{X^n}\left[d\left(X^n, y^n\right)\left|\alpha\left(X^n\right) = i\right]\right.\frac{P\left(\alpha\left(X^n\right) = i\right)}{P\left(f\left(\alpha\left(X^n\right), \mathbf{T}\right) = i'\right)}\end{aligned}$$

In the case of squared error distortion, the solution for $\beta^*\left(i'\right)$ is given explicitly by

$$\beta^*\left(i'\right) = \sum_{i\in\mathcal{M}}\mathrm{E}_{X^n}\left[X^n|\alpha\left(X^n\right) = i\right]\frac{P\left(\alpha\left(X^n\right) = i\right)}{P\left(f\left(\alpha\left(X^n\right), \mathbf{T}\right) = i'\right)}$$

The algorithm

The steps of the algorithm are summarized below:

0. *Initialization.* Initialize the encoder, decoder, and prefix code.

1. *Optimize the encoder.* Optimize the encoder for the current decoder and prefix code. The optimal encoder is described by Equation (1).

2. *Optimize the prefix code.* Optimize all prefix codes. For fixed-rate coding no change occurs.

3. *Optimize the decoder.* Optimize the decoder for the given encoder and prefix code. The optimal decoder is described by Equation (2).

4. *Repeat steps 1 to 3 until convergence.*

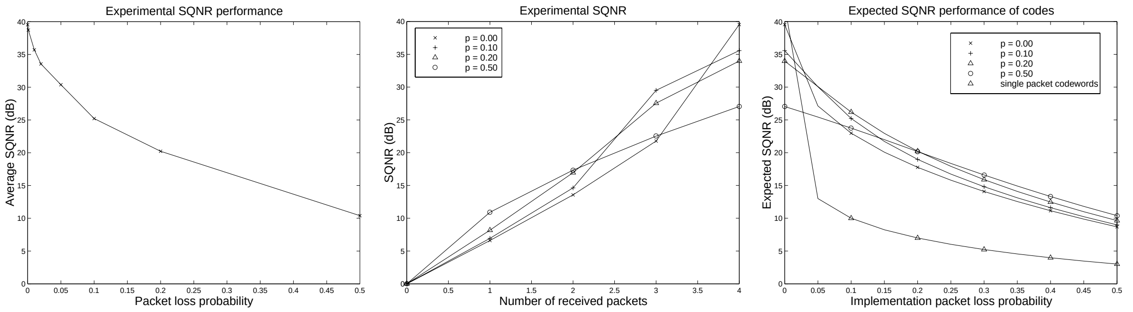

Figure 4: (a) The average SQNR as a function of packet loss probability. (b) SQNR as a function of the number of received packets. (c) The SQNR performance of different codes as a function of packet loss probability.

IV Experimental Results

We perform a series of experiments using the algorithm described in the previous section. Fixed-rate, 4-packet MDVQ codes of vector dimension 4 are trained on a sequence of 20 magnetic resonance brain scan images and tested on a non-overlapping sequence of 5 brain scan images. The rates reported in this section include only the source coding rate; rate spent in overhead (e.g., header information in each packet) is assumed to be fixed and is not included in the current analysis. For simplicity, it is assumed that the packet losses are described by independent random variables and that each packet has the same probability p of being lost. The Lagrangian weights are set according to $\mu_{\mathbf{T}} = p_{\mathbf{T}}$. Experiments are conducted using codes designed for four different values of $p : 0, 0.1, 0.2$ and 0.5.

The optimal signal to quantization noise ratio (SQNR) for a system with a particular packet loss probability will be achieved using an MDVQ code designed specifically for that loss probability. The SQNR achieved experimentally at each of the different values of p is shown in Figure 4(a). The SQNR decreases as p increases since fewer packets on average are received.

The SQNR as a function of the number of packets received is shown for the different MDVQs (corresponding to each of $p = 0, 0.1, 0.2, 0.5$) in Figure 4(b). For small values of p, the probability that none of the four packets are lost is very high. Thus, the design optimization will place the most emphasis on minimizing the expected distortion (maximizing the SQNR) for zero packet losses. As p increases, scenarios involving one or more packet losses become more probable, and the design optimization places more weight on these scenarios. As a result, some of the performance for the situation where zero packets are lost is traded for an increase in the performance observed under some packet loss. This behavior is also seen in Figure 4(c), which shows the expected performance of several MDVQs (each designed for a different value of p) over a wide range of packet loss probabilities. Each code performs better than the others close to its design value of p. The performance of the MDVQs is also compared to that of a non-MD scheme, where each codeword of a traditional tree-structured VQ is considered as a single packet, i.e. each codeword is either transmitted or lost in its entirety. The MDVQs show significantly better overall performance.

V Summary

We describe a new design algorithm for an optimal generalized MDVQ. The MDVQ optimizes the expected performance of a packet-based network for an arbitrary number of packets and for an arbitrary collection of priorities (or probabilities) over the different packet-loss scenarios. An alternative but equivalent application is that of optimizing the performance of a communication system with an arbitrary number of channels (e.g., a diversity system), some of which may break down. The MDVQ design uses a ternary tree structure that arises from the inclusion of a packet loss branch into a binary tree. The experimental results achieved show good performance under a wide range of packet loss scenarios.

References

[1] M. Effros. Practical multi-resolution source coding: TSVQ revisited. In *Proceedings of the Data Compression Conference*, pages 53–62, Snowbird, UT, March 1998. IEEE.

[2] H. Brunk, H. Jafarkhani, and N. Farvardin. Design of successively refinable scalar quantizers. April 1998. Preprint.

[3] A. El Gamal and T. M. Cover. Achievable rates for multiple descriptions. *IEEE Transactions on Information Theory*, IT-28(6):851–857, November 1982.

[4] L. Ozarow. On a source coding problem with two channels and three receivers. *Bell System Technical Journal*, 59:446–472, December 1980.

[5] J. K. Wolf, A. D. Wyner, and J. Ziv. Source coding for multiple descriptions. *Bell System Technical Journal*, 59:1417–1426, October 1980.

[6] Z. Zhang and T. Berger. Multiple description source coding with no excess marginal rate. *IEEE Transactions on Information Theory*, 41(2):349–357, March 1995.

[7] V. A. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Transactions on Information Theory*, 39(3):821–834, May 1993.

[8] V. A. Vaishampayan and J. Domaszewicz. Design of entropy-constrained multiple description scalar quantizers. *IEEE Transactions on Information Theory*, 40(1):245–250, January 1994.

[9] V. A. Vaishampayan. Vector quantizer design for diversity systems. In *Proceedings of the 25th Annual Conference on Information Sciences and Systems*, pages 564–569. IEEE, March 1991.

[10] P. A. Chou, T. Lookabaugh, and R. M. Gray. Entropy-constrained vector quantization. *IEEE Transactions on Acoustics Speech and Signal Processing*, 37(1):31–42, January 1989.