Multistage Vector Quantizer Optimization for Packet Networks

Hosam Khalil, *Member, IEEE*, and Kenneth Rose, *Fellow, IEEE*

***Abstract*—A multistage vector quantizer (MSVQ) based coding system is source-channel optimized for packet networks. Resilience to packet loss is enhanced by a proposed interleaving approach that ensures that a single lost packet only eliminates a subset of the vector stages. The design is optimized while taking into account compression efficiency, packet loss rate, and the interleaving technique in use. The new source-channel-optimized MSVQ is tested on memoryless speech line spectral frequency (LSF) parameter quantization as well as block-based image compression. With LSF coding, a source-channel-optimized MSVQ is shown to yield gains of up to 2.0 dB in signal-to-ratio (SNR) over traditional MSVQ and to substantially enhance the robustness of packetized speech transmission. Substantial gains were also obtained in the case of block-based image compression. Although the formulation is given in the context of packet networks, the work is directly extendible to the broader category of erasure channels.**

***Index Terms*—Erasure channels, multistage vector quantizer, packet networks, source-channel coding.**

I. Introduction

PACKET networks have dramatically gained in importance and popularity in recent years, especially due to the widespread use of the Internet, which is a heterogeneous collection of networks. Currently, the Internet does not provide quality-of-service (QoS) guarantees, and transmission of audio and video places considerable strain on the resources of any connection due to the amount of data involved. In addition, transmission of audio and video poses a major challenge as their real-time nature and interactivity requirements typically preclude the use of reliable handshake-based protocols. The delays added by relying on retransmission mechanisms for packet loss recovery are generally not compatible with interactive or broadcast audio and video applications.

For this reason, connectionless protocols (as in the case of datagrams) are the transport protocol of choice, and packet losses are dealt with at the application level, but such protocols do not embed any congestion control mechanisms. Hence, unavoidable packet losses, delay, and delay jitter due to congestion make effective streaming of audio and video over the Internet a challenging task. Higher bandwidth access through cable and digital subscriber loop modems, network traffic, and server load reduction through the large-scale deployment of multicast units, as well as the increasing backbone bandwidth, will partially alleviate these network problems but will not completely eliminate them. Moreover, the continuous demand for improved quality as the network conditions improve, compounded with the potential increase in number of users, will cause the bandwidth needed for audio and video applications to increase. The heterogeneity of the Internet requires the audio and video applications to avoid congesting the network and to act in a network-friendly manner.

In an unreliable session, packets dropped by the network due to congestion are lost, and information about such losses is not made available to the encoder in a timely manner. In addition, for interactive or live applications, late packets expire and may be dropped by the receiver. Naturally, much research effort is currently focused on robustness to packet loss due to congestion and delays. Packet networks, hence, represent a special case of erasure channels. This paper is concerned with the design of source-channel coding systems for packet networks in particular and erasure channels in general. Another potential application is in wireless communications over deep fading channels, which may also be viewed as erasure channels.

Most existing robust video, image, and audio techniques heavily depend on decoder-driven concealment (see e.g., [1], [2], [3]). In the case of video, the goal of concealment is to mask the missing visual information based on the temporal or spatial redundancy that is present in video sequences or spatial redundancy in images. Of course, it may not be possible to completely reconstruct the lost image blocks. The basic idea, however, is that as long as the losses are infrequent and the quality requirements are not strict, the lost image blocks can be effectively approximated using the temporally or spatially adjacent blocks. The loss concealment approach can be combined with forward error correction (FEC) mechanisms. For example, a video transmission application can insert FEC-based loss recovery information at the encoder and then use loss concealment for any unrecoverable lost information at the decoder [4]. In [5], a block shuffling scheme for whole video blocks was suggested to help isolate erroneous blocks at the receiver, facilitating the process of error concealment. Examples from speech coding can be found in [6]–[8]. When information concerning whole video macroblocks or whole speech frames are interleaved for the purpose of distributing the effect of losses, then this scheme is called block interleaving (for video or image coding) or frame interleaving (for speech and audio coding).

Manuscript received January 8, 2002; revised December 31, 2002. This work was supported in part by the National Science Foundation under Grants MIP-9707764, EIA-9986057, and EIA-0080134, the University of California MICRO Program, Dolby Laboratories, Inc., Lucent Technologies, Inc., Mindspeed Technologies, Inc., and Qualcomm, Inc. Parts of this work appeared in the *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, June 2000. The associate editor coordinating the review of this paper and approving it for publication was Prof. Iaon Tabus

The authors are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, Santa Barbara, CA 93106–9560 USA.

Digital Object Identifier 10.1109/TSP.2003.812731

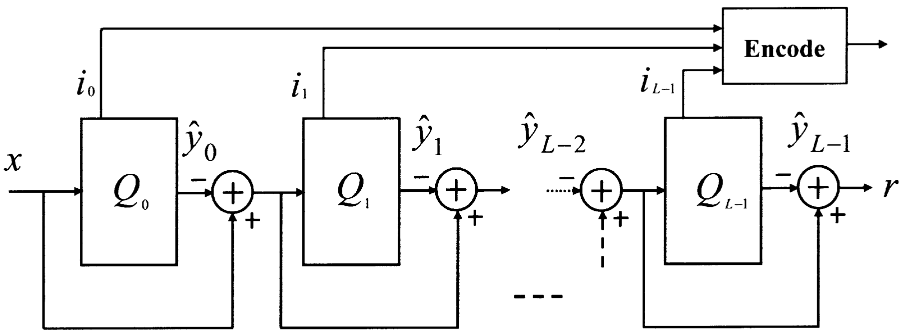


Fig. 1. Encoder block diagram for the MSVQ.

Currently, more research is shifting toward encoder-driven robustness techniques. A known approach to combat packet loss is the design of diversity systems that employ multiple description coding. This direction of research was stimulated by Vaishampayan's work on scalar quantizers [9] and was later pursued in the context of transform coding [10], and vector quantizers [11], [12]. The basic idea is to encode and transmit more than one description of the source over different channels or packets. In the event of channel failure or packet loss, the decoder reconstructs the data from the remaining received descriptions. The encoder is thus designed such that there is sufficient redundancy in the descriptions. Recent work on multiple descriptions for video and audio coding include [13]–[15]. One other interesting approach was presented in [16], where the encoder inserts redundant information of each packet in subsequent packets. A lost packet can then be recovered from the copies piggybacked in subsequent packets if they are successfully received. In [17], robust speech communication is achieved through the generation of multiple redundant descriptions of the speech stream and sending them over independent paths. Other methods in the speech coding literature offer certain resilience to information loss by separate encoding of partial information, e.g., TwinVQ [18] and conjugate structure CELP [19]. However, the resilience to information loss is not directly optimized.

Of interest to us in this work is the utilization of the strong compression capability of a popular coding structure and its generalization to the case of lossy channels. Multistage vector quantization (MSVQ) is widely used in the compression of audio signals [20], [21] and has also been applied to image [22] and video coding [23]. MSVQ decomposes the source vector into the sum of codevectors: one per stage. Historically, MSVQ was conceived as a sequential quantization operation, where each stage simply quantizes the reconstruction error of the previous stage. In [24] and [25], an unequally FEC-protected MSVQ design was proposed where the receiver estimates the channel conditions and decodes as many stages of the quantized signal as can be reliably decoded. In that work, the multistage coder is viewed as a tool for successive refinement, and thus, to decode stage n, all stages up to $n-1$ need to be correctly decoded. In [26], the greedy nature of simple sequential encoding was recognized, and efficient techniques were proposed to seek better approximation of the source vector as combination of stage vectors. Motivated by such improved search techniques, we propose an MSVQ coding system where any individual stage index is subject to loss and directly optimize the MSVQ for such general information loss patterns. The proposed MSVQ-based coding system is source-channel optimized for packet networks, where resilience to packet loss is achieved through an interleaving scheme. We interleave on the level of the MSVQ indices, and thus, we refer to this method as the MSVQ-index interleaving approach.

This paper is organized as follows. In Section II, we introduce basic means to exploit the robustness potential of MSVQ via appropriate interleaving techniques. In Section III, we propose a design technique to optimize the MSVQ for the given packet loss statistics. We then present simulation results and conclusions in Sections IV and V, respectively.

II. MSVQ FOR PACKET NETWORKS

A. Overview

The MSVQ structure gained popularity in the field of signal compression because of its acceptable search and memory complexity. The structure of the MSVQ encoder [27] consists of a cascade of VQ stages as shown in Fig. 1. For an L-stage MSVQ, an lth-stage quantizer $Q_l,\ l = 0, 1, 2, \ldots, L-1$ is associated with a stage codebook C_l. Each stage codebook C_l contains K_l stage codevectors. The set of stage quantizers $\{Q_0, Q_1, \ldots, Q_{L-1}\}$ are equivalent to a single quantizer $\boldsymbol{Q}$, which is referred to as the direct-sum vector quantizer. Since we place restrictions on the structure of the quantizer, we expect the SNR achievable using MSVQ to be lower than that achievable by an unconstrained VQ for the same bit rate. However, the complexity and storage benefits often outweigh this drawback, as it enables higher dimension implementations.

An input vector (of dimension n) $X^n \in \mathcal{X}^n$ is encoded into L indices $(i_0, i_1, \ldots, i_{L-1})$ that are concatenated into one overall codeword $\boldsymbol{i}$. The set $\mathcal{I}$ is the set of all possible codewords. The codevector selected at stage l, which corresponds to index i_l, is denoted by $\hat{\boldsymbol{y}}_l$. Note that stage quantizer Q_l quantizes the error $X^n - \sum_{i=0}^{l-1} \hat{\boldsymbol{y}}_i$. We denote the encoder by α: $\mathcal{X}^n \rightarrow \mathcal{I}$. The set of received codewords that have been subjected to possible losses during transmission is denoted by $\mathcal{I}'$. Thus, the decoder function is given by β: $\mathcal{I}' \rightarrow \mathcal{Y}^n$, where β maps received codeword $\boldsymbol{i}'$ to a reproduction vector $Y^n \in \mathcal{Y}^n$.

The best set of stage codevectors to represent an input vector can be obtained by an exhaustive search. However, a well-known and efficient approach for suboptimal search is the M-search [26], [28], which has been found to provide close to optimal performance at a small fraction of the complexity. In this case, indices are searched for sequentially (stage-by-stage). At each stage, the search is restricted to extending M candidates from the previous stage. By keeping only the best M solutions at each stage (the so-called survivors), manageable complexity is maintained, at the cost of hopefully minimal compromise in performance.

The design procedure of MSVQ involves the need to produce a direct-sum quantizer whose stages are jointly optimal. The procedure is initialized with a set of L stage codebooks that is typically obtained by traditional sequential design [27]. The quantizer codebooks are reoptimized iteratively using the following two steps: First, the training set is optimally partitioned using the current quantizer in what is often termed *encoder optimization*. This is achieved by encoding the vectors of the training set using the current quantizer and then recording the codevectors/training vectors associations. Second, the quantizer codebooks are optimized for the current training set partitions in what is termed *decoder optimization*. This is achieved by iterating over the stages and optimizing each stage codebook while keeping the remaining stages fixed [29]. The reoptimization of quantizer Q_l consists of updating its codebook. The update of codevector $\hat{\boldsymbol{y}}_l$ involves only the training subset $\boldsymbol{\chi}_l \subset \boldsymbol{X}$ that selects $\hat{\boldsymbol{y}}_l$ for its stage l codevector (see Fig. 2). It is convenient to remove the effect of all fixed stages from the training subset. Hence, for each training vector $\boldsymbol{x} \in \boldsymbol{\chi}_l$, we subtract the fixed codevectors' contribution

$$\varphi(\boldsymbol{x}) = \boldsymbol{x} - (\boldsymbol{Q}(\boldsymbol{x}) - \hat{\boldsymbol{y}}_l) \tag{1}$$

where $\boldsymbol{Q}(\boldsymbol{x})$ denotes an M-search quantization of $\boldsymbol{x}$ using the current MSVQ. The outcome of this operation is a new training set $\phi_l = \{\varphi(\boldsymbol{x}), \boldsymbol{x} \in \boldsymbol{\chi}_l\}$, where the fixed codevectors' influence has been eliminated.

It is easy to see that minimum distortion is achieved by adjusting $\hat{\boldsymbol{y}}_l$ to the centroid of ϕ_l:

$$\hat{\boldsymbol{y}}_l^{\text{new}} = \frac{1}{|\phi_l|} \sum_{\varphi \in \phi_l} \varphi. \tag{2}$$

Noting further that $|\phi_l| = |\boldsymbol{\chi}_l|$, we may equivalently write the update rule

$$\Delta \hat{\boldsymbol{y}}_l = \frac{1}{|\boldsymbol{\chi}_l|} \sum_{\boldsymbol{x} \in \boldsymbol{\chi}_l} (\boldsymbol{x} - \boldsymbol{Q}(\boldsymbol{x})). \tag{3}$$

This rule has a simple and intuitively satisfying interpretation: The stage codevector $\hat{\boldsymbol{y}}_l$ is chosen such that the expected reconstruction error of the corresponding training subset is zero-mean (unbiased). After optimizing all codevectors in stage l, all other stages are similarly optimized in order, and then, the whole procedure is repeated until the rate of change falls below a prescribed threshold.

B. MSVQ Index Loss

Traditional coding systems that utilize the MSVQ structure concatenate the codewords representing the best encoding indices and transmit the bitstream over the channel. In a system designed for better resilience to channel conditions, the bitstream may be protected using FEC methods. In [21], it was noted that the index for the first stage of MSVQ used to quantize line spectral frequency (LSF) parameters contained more perceptually important information than the indices from the other stages. Consequently, the index codeword for the first stage was protected with a Hamming code. In [25], the MSVQ coding system took into account channel conditions and protected each stage codevector with a different level of protection, each dependent on the significance of the stage information. In this way, depending on the channel condition, stage codewords can be decoded up to the stage, whose protection enables reliable decoding.

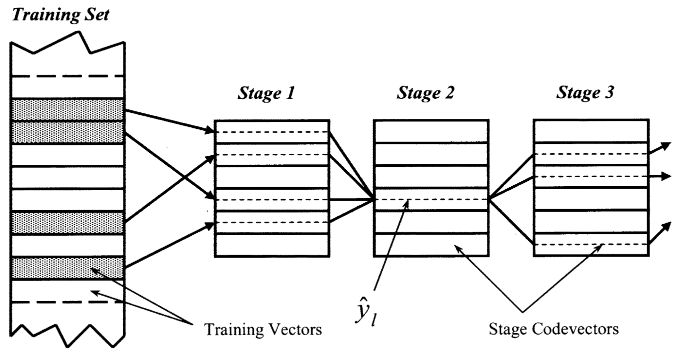


Fig. 2. Training subset for use in re-optimization of $\hat{\boldsymbol{y}}_l$. Only the shaded vectors in the training set use stage codevector $\hat{\boldsymbol{y}}_l$ ($l = 1$, and $\hat{\boldsymbol{y}}_1$ is the fourth vector in $\boldsymbol{Q}_1$).

In this section, we propose a different approach to how MSVQ indices are handled. We claim that each stage index in an MSVQ system carries important information about the original vector. An L-stage MSVQ quantizes an input vector by applying an M-search to find a set of indices $(i_0, i_1, i_2, \ldots, i_{L-1})$ pertaining to the (hopefully) best choice of stage-codevectors for its representation. As stated earlier, instead of greedily selecting the best vector at the current stage, the decision is postponed, and M "survivors" are temporarily stored. In other words, the search process puts more emphasis not on greedily finding the best stage-vector at each stage but on the best total combination of stage-vectors. This fact, coupled with the observation that, at high dimensionality, individual stage-vectors exhibit a high degree of mutual orthogonality, suggests that it is worthwhile to decode and use a stage-vector even if some preceding indices are missing. In order to illustrate this point, which is a central motivation for our approach, we will next make reference to a small relevant subset of the simulation results that are presented in a later section of this paper.

MSVQ is widely applied to coding of LSF parameter vectors. For low bit rate speech coding, LSFs are ten-dimensional vectors and, when coded, constitute a major portion of the bitstream. An experiment was conducted to test the decoding properties of MSVQ at different stage index loss patterns, and the results are shown in Table I. Results are averaged over a test set of 6000 LSF vectors. For a given L-stage MSVQ, the pattern of index losses can be described using an index transmission vector $T = (T_0, T_1, \ldots, T_{L-1})$, where

$$T_l = \begin{cases} 1, & \text{if index } l \text{ is correctly received} \\ 0, & \text{if index } l \text{ is lost.} \end{cases} \tag{4}$$

TABLE I
PERFORMANCE COMPARISON OF LSF RECONSTRUCTION USING SOURCE-CHANNEL-OPTIMIZED MSVQ INDICES AND TRADITIONAL SOURCE-OPTIMIZED MSVQ. SHOWN IS THE AVERAGE MSE AT CONDITIONS OF PARTIAL RECEIVED INFORMATION ACCORDING TO SHOWN TRANSMISSION ERROR PATTERNS. THE MSE WHEN THE PREVIOUS RECONSTRUCTED LSF IS USED AS CONCEALMENT IN PLACE OF THE MISSING LSF IS ALSO SHOWN

Transmission Vector	Source-Optimized MSVQ	Source-Channel-Optimized MSVQ
	MSE	MSE
11111	0.30	0.37
11110	0.69	0.82
11101	1.07	1.10
11100	1.39	1.53
11011	1.88	1.91
11010	2.22	1.98
11001	2.53	2.27
11000	2.80	2.78
10111	3.70	1.77
10110	4.06	2.35
10101	4.39	2.66
10100	4.67	3.22
10011	5.03	3.07
10010	5.33	3.73
10001	5.59	4.05
10000	5.82	4.69
01111	10.21	9.89
Concealment	8.76	8.76

TABLE II
PERFORMANCE COMPARISON FOR IMAGE RECONSTRUCTION USING SOURCE-CHANNEL-OPTIMIZED MSVQ-INDICES AND TRADITIONAL SOURCE-OPTIMIZED MSVQ. THE AVERAGE MSE AT CONDITIONS OF PARTIAL RECEIVED INFORMATION ACCORDING TO SHOWN TRANSMISSION ERROR PATTERNS IS SHOWN. THE MSE WHEN THE CONCEALMENT METHOD [30] IS USED TO RECOVER LOST IMAGE BLOCKS IS ALSO SHOWN

Transmission Vector	Source-Optimized MSVQ	Source-Channel-Optimized MSVQ
	MSE	MSE
11111	22.82	29.50
11110	32.47	36.45
11101	39.74	39.77
11100	47.75	51.55
11011	57.12	44.29
11010	65.85	58.85
11001	72.15	64.15
11000	79.25	83.54
10111	148.17	47.66
10110	156.92	64.34
10101	163.20	70.94
10100	170.32	92.45
10011	178.03	78.02
10010	185.86	102.32
10001	191.18	110.89
10000	197.38	140.02
01111	252.38	223.88
Concealment	196.29	196.29

In the first column of Table I, we show the index transmission vectors. The second column lists, for traditional source-optimized MSVQ design, the mean squared error (MSE) results for LSF reconstruction subject to the corresponding index transmission vector, averaged over the test set. For reference, we also included in the last row the average MSE when the last correctly reconstructed LSF is repeated for concealment—the most widely adopted recovery technique when speech information is lost [3]. We can see that on average, using all received indices to obtain the reconstruction is preferable to repeating the previous LSF (as if the whole LSF is lost). This observation remains true even when many of the indices are lost, as long as the transmission patterns indicate successful transmission of the first stage-index. The requirement that i_0 be received for preferable performance is due to the fact that i_0 contains the most significant information, and the loss of i_0 often renders the remaining indices virtually useless. In fact, in the above test, loss of index i_0 caused distortion higher than the concealment method (on average) in all cases. A representative example exhibiting loss of i_0 is shown in Table I. Usually, the first stage carries most of the *mean* information. For this reason, in some audio standards such as [21], i_0 is more heavily protected using FEC codes, as explained earlier.

A similar test for image coding is shown in Table II. In the first column of Table II, we list the index transmission vectors. The second column lists, for traditional MSVQ design, the MSE results for image block reconstruction subject to the corresponding index transmission vector, averaged over the test set. The image is broken into blocks of 8×8, which are rearranged into 64-dimensional vectors and fed into the MSVQ for compression. The last row in Table II shows the expected MSE that would result if the concealment method of [30] is used. The concealment method would be used when all information pertaining to a block is lost or discarded due to losses in transmission. It can be seen, as in the case of speech, that a lower MSE is expected when the reconstruction uses as many indices as received, rather than resorting to concealment. The conclusion we can draw is that in the situation where individual indices may become lost, using any remaining received indices (given that the first index has been received) contributes to lower average distortion than discarding those indices.

C. MSVQ Index Interleaving

We propose a framework for encoding the LSFs and image blocks into data packets such that the decoding is more robust to channel losses. Tables I and II strongly suggest that it is better to have a loss distributed over several LSFs than to have a complete loss of an LSF, and similarly, a loss distributed over several image blocks is better than a complete loss of an image block. An interleaving method is the natural solution to this problem. See Fig. 3 for an example of interleaving of two LSFs over two packets, where the indices for the first LSF and second LSF are denoted by a_l and b_l, $l = 0, \ldots L-1$, respectively. Notice that a_0 and b_0 are transmitted on both packets. Although this redundancy increases the bit rate, it was found to contribute significantly to robustness.

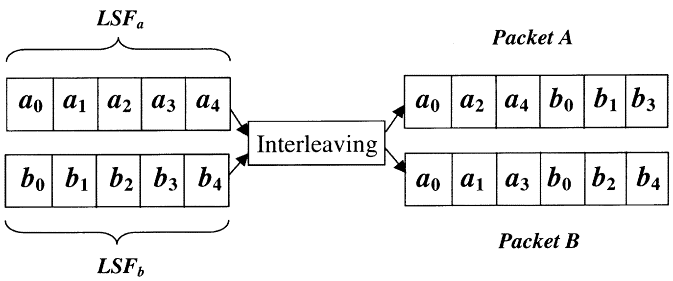


Fig. 3. Example of interleaving of LSF parameters between two data packets. Note that a_0 and b_0 are transmitted on both packets.

Using the interleaving scheme of Fig. 3 and given packet loss rate of p, we have four different transmission vectors—00000, 10101, 11010, 11111—with probabilities p^2, $p(1-p)$, $(1-p)p$, and $(1-p)(1-p)$, respectively. The transmission vector 00000 occurs when both packets are lost, and in this case, the last received LSF is used for concealment. For the case of image coding, the transmission vector 00000 also necessitates concealment. For a particular interleaving scheme, or a known loss pattern, the MSVQ can be directly optimized for such losses, as will be shown in the next section. For a frame interleaving scheme, only two transmission vectors exist—11111 or 00000—with probabilities $(1-p)$ and p, respectively. It can easily be seen that the all-zeros transmission vector happens much less frequently in the MSVQ-index interleaving approach, especially for low packet loss rate.

Sending the MSVQ parameters in interleaved form over packet networks would cause delay. Specifically, if two consecutive frames are interleaved, then the processing delay is equal to the time of two frames. In VoIP applications, each voice frame is usually encapsulated in one datagram. Each datagram carries an internet protocol (IP) header plus a user datagram protocol (UDP) header. Such overhead can be as high as 40 bytes.[1] For a speech coder operating at a rate of 2.4 kb/s, the actual payload may be as low as 7 bytes, but this inefficiency is tolerated in order to keep the processing delay as low as possible. For streaming applications, restrictions on processing delays are less strict and allow for even more interleaving and more efficient packet utilization.

III. Source-Channel Optimization of MSVQ

In this section, we propose an optimization algorithm to further strengthen MSVQ for use on packet network channels under interleaving schemes such as the one described in Section II. Specifically, the MSVQ structure is source-channel optimized for predefined loss patterns.

The aim of the new algorithm is to extend the traditional MSVQ design algorithm (as explained in Section II) to handle packet loss. Specifically, the method minimizes the end-to-end distortion averaged over all possible transmission vectors with respect to probabilities p_T, where $T \in \mathcal{T}$ is the set of possible transmission vectors.

[1]Header compression methods can be used to lower the header size to 7 bytes.

Encoder Optimization

An optimal encoder α^* maps each source vector X^n to the stage codevectors that would jointly minimize the expected distortion. If transmitted concatenated codeword $\boldsymbol{i}$ is subject to channel loss according to the pattern of transmission vector T, then the received concatenated codeword is determined by the masking function $f(\boldsymbol{i}, T)$. Moreover, the corresponding reconstructed vector produced by the decoder is given by $\beta(f(\boldsymbol{i}, T))$. Lost indices within codeword $\boldsymbol{i}$ are assumed to result in zero-valued stage-codevectors. Thus, the optimal $\alpha^*(X^n)$ is given by

$$\alpha^*(X^n) = \arg\min_{\boldsymbol{i}\in\mathcal{I}} \left\{ \sum_{T\in\mathcal{T}} p_T d(X^n, \beta(f(\boldsymbol{i}, T))) \right\} \qquad (5)$$

where $d(\cdot, \cdot)$ is the distortion measure. In other words, we search for the set of indices that minimizes the expected distortion over all possible transmission vectors (each weighted by its probability of occurrence). Note that for simplicity of description, we have made abstraction of the stage-wise encoding search. Nonetheless, an M-search can be employed to minimize the criterion given in (5). As with the lossless channel case, the selection of the best M candidates at stage l involves calculation of (5), where the MSVQ is assumed to have only l stages. However, since transmission losses are possible here, transmission vector patterns affecting the first l stages need to be accounted for.

For example, when searching for the best third-stage codevector of a four-stage MSVQ, only the first three stages are assumed to be subjected to possible loss. The number of transmission vector patterns are thus eight. Since we further require successful transmission of the first-stage vector, only four of these patterns are of interest in the optimization: 100, 101, 110, and 111. The probability of each of the three-stage patterns is obtained by summing the probabilities of the four-stage patterns consistent with it. For example, the probability of 101 is equal to the sum of the probabilities of 1010 and 1011.

Decoder Optimization

Given encoder α, we find the optimal decoder β^* that minimizes the expected distortion given the received channel codeword index $\boldsymbol{i}' \in \mathcal{I}'$:

$$\beta^*(\boldsymbol{i}') = \arg\min_{\boldsymbol{y}\in\mathcal{Y}^n} \{E_{X^n}[d(X^n, \boldsymbol{y}) | f(\alpha(X^n), T) = \boldsymbol{i}']\}. \quad (6)$$

For received codeword $\boldsymbol{i}'$ that has been subjected to loss T, there exists a set of possible transmitted indices $\mathcal{M} \subset \mathcal{I}$ such that $f(m, T) = \boldsymbol{i}', \forall m \in \mathcal{M}$. Thus, the solution in the case of squared error may be rewritten as

$$\begin{aligned}\beta^*(\boldsymbol{i}') &= E_{X^n}[X^n | \alpha(X^n) \in \mathcal{M}] \\ &= \sum_{\boldsymbol{i}\in\mathcal{M}} \frac{P(\alpha(X^n) = \boldsymbol{i})}{P(\alpha(X^n) \in \mathcal{M})} E_{X^n}[X^n | \alpha(X^n) = \boldsymbol{i}]. \end{aligned} \quad (7)$$

We interpret (7) as follows. Using the same example again, a four-stage MSVQ is subjected to four-stage transmission vectors. The best decoding stage codevector for the third stage is one that is only concerned with transmission vectors showing successful transmission of third stage index. Thus, only 1010, 1011, 1110, and 1111 are taken into account. In accordance, the best centroid to choose for a stage codevector is one that weighs the centroids from all four different transmission scenarios, according to their relative probability of occurrence. The centroid from each transmission scenario is obtained assuming a reduced MSVQ, as dictated by the corresponding transmission vector. For example, for the transmission vector 1010, the MSVQ is assumed to contain only two stages, and thus, the required centroid is optimized using the stage-removed method on a two-stage MSVQ.

The iterative algorithm iterates between (5) and (7). First, (5) is executed, and the optimal training data assignment to stage codevectors is recorded. Then, (7) is executed to optimize the decoder. Stage-wise optimization is performed, given the fixed encoder assignments. The complexity of the design algorithm is significantly higher than that of traditional MSVQ design, and this is due to the fact that the optimization accounts for and averages over all transmission scenarios. Note, however, that transmission vectors showing one or more index losses allow us to limit the search through the MSVQ to only the remaining stages and yield a corresponding reduction in complexity. For an MSVQ with L stages, the complexity of source-channel optimized MSVQ design is approximately 2^{L-2} that of traditional source-optimized MSVQ. This is due to the number of transmission vectors showing successful delivery of first index 2^{L-1} divided by two to account for the fact that half of the indices are lost on average within the remaining transmission vectors. The exponential growth with L may appear alarming. Note, however, that L is typically a very small number. Note further that the complexity of the MSVQ is originally not very high, and that is one of the main advantages of incorporating MSVQ into many practical coding systems.

The iterative algorithm must be initialized with some MSVQ. A reasonable choice of initialization is with an MSVQ that was optimized assuming a lossless channel (i.e., a source-optimized MSVQ). One problem with this approach is that once a source-optimized MSVQ is subjected to loss patterns at considerable loss rates, many of the cells defined by the stage codevectors may become empty, and the MSVQ would hence be underutilized, leading to inefficiency. An algorithm to regularly fill such empty cells is needed. The actual algorithm we use here is implemented using a selective splitting procedure [31]. The MSVQ is initialized using selective splitting of stage codevectors in such a way that splits are locally optimized directly in the source-channel sense. The codebooks are thus grown gradually until the objective codebook sizes are achieved. Since the algorithm grows the codebooks directly to operate at the desired loss rate, the codebook will not suffer from empty cell problems, and any that do appear along the way are automatically filled with new codevectors. This same algorithm has been instrumental in optimizing other systems such as the one presented in [23].

IV. Simulation Results

In this section, we present several experiments including objective and subjective test results. The proposed source-channel optimized design of MSVQ was tested on speech and image coding for transmission under different scenarios. We first examine the performance of a source-channel optimized MSVQ under conditions of loss of individual indices. Then, we evaluate the performance within an interleaving scheme by simulating packet losses at the different packet loss rates. One additional transmission scheme is tested: a frame interleaving scheme with the first-stage index repeated in the subsequent packet. The delay in this case is the same as the index-interleaving schemes. For every two consecutive packets, if a single packet is lost, then one packet receives all five indices, and the other receives just the first index. If both packets are received, then the first index is redundant. If both packets are lost, then error concealment is employed. The M-search uses five candidates in all simulations.

A. Speech Transmission

In Section II, we demonstrated the performance of the MSVQ on LSF compression and transmission under conditions of index loss. In Table I, we demonstrate the performance of MSVQ source-channel optimized for an index loss probability $p = 0.30$ at the various transmission vectors indicated. The results demonstrate the improvements of source-channel-optimized MSVQ over traditional MSVQ. This is particularly evident from the lower average distortion shown in the second half of Table I, where more indices are lost. It can be seen that MSE for the more severe losses is significantly decreased. On the other hand, in the case of less severe losses, the MSE increases as expected. When the whole transmission problem is cast in probabilistic terms, we find that the overall expected MSE for a source-channel optimization is much lower than that of a traditional source optimization. For example, assuming an MSVQ-index interleaving scheme, the loss of a single packet may result in subjecting an LSF to the transmission vector 10101 and another LSF to the vector 11010. Thus, the average MSE for a source-optimized MSVQ would be $(4.39 + 2.22)/2 = 3.36$, whereas the average for a source-channel optimized MSVQ would be $(2.66 + 1.98)/2 = 2.32$ and, hence, considerably reduced.

A full speech coder breaks down the speech signal into several parameters that enable efficient compression. Such parameters include pitch values, gain values, excitation vectors, and MSVQ LSF parameters. Although a complete robust speech coder would require implementing error-resilient techniques for all the transmitted components, we provide results that concentrate on the LSF portion of the coder and assume that all other parameters are received successfully. This enables us to study the effect of the new design technique without interference of unrelated factors due to other components.

The MSVQ parameters are encoded into the bitstream and constitute a considerable portion of the allocated bandwidth. For example, in the mixed excitation linear prediction voice coder [21], LSF parameters use 25 bits of the allocated 54 bits per frame of speech. For the experiments, we assume a packet-based communication channel. The MSVQ-index interleaving scheme of Fig. 3 will be used. Training uses about 25 000 training vectors, and testing is performed on an independent set of 6000 vectors. The 2.4-kb/s MELP coder is used to compress all other parameters. One LSF is extracted for each 22.5 ms frame.

TABLE III
PREFERENCES IN AN INFORMAL LISTENING TEST INVOLVING 12 LISTENERS. A COMPARISON BETWEEN TRADITIONAL FRAME INTERLEAVING AND THE PROPOSED MSVQ-INDEX INTERLEAVING IS SHOWN. FRAME INTERLEAVING USES 25 BITS IN CASE A AND 30 BITS IN CASE B. MSVQ-INDEX-INTERLEAVING USES 30 BITS/LSF IN BOTH CASES. VALUES SHOWN IN PERCENT

	Traditional Frame-Interleaving	MSVQ-Index-Interleaving	No Preference
Case A	8.33	88.33	3.34
Case B	15.00	78.33	6.67

In the first experiment, we compare frame-interleaving with MSVQ-index interleaving. The MSVQ-index interleaving is used without channel optimization to demonstrate the power of the interleaving scheme alone. In MELP, 25 bits, in four MSVQ stages, are devoted to LSF quantization. We use a slightly different structure for the quantizer from the one used in the standard. A five-stage MSVQ is used, with each stage equally allocated 5 bits. Since the interleaving procedure used (Fig. 3) assumes sending the first stage index twice, the total bit rate used here is 30 bits/LSF. For the case of frame interleaving, we test two cases: In Case A, frame interleaving uses five stages, i.e., 25 bits/LSF, whereas in case B, frame interleaving uses six stages, i.e., 30 bits/LSF. We use case B to match the bit rate of the MSVQ-index interleaving coder. See Table III for the results of an informal blind listening test involving 12 listeners and a set of 16 MIRS sentences. Note that the proposed MSVQ-index interleaving method is overwhelmingly preferred over the traditional frame-interleaving method even without source-channel optimization. When the bit rate used is exactly the same (case B), the "nonoptimized" MSVQ-index interleaving method still outperforms the frame interleaving by achieving about 78% preference. For this test, we use $p = 0.30$.

In the second experiment, we further improve the performance of the MSVQ-index interleaving method by optimizing the system for packet losses. A source-channel-optimized MSVQ can thus be optimized using the design in Section III, and the given rate of transmission errors can be optimized as described in Section II. Here, we compare the source-optimized MSVQ-index interleaving with that of the more powerful source-channel-optimized MSVQ-index interleaving. Note that both methods use our proposed interleaving scheme. The objective is to evaluate the added benefit of source-channel design. In Table IV, we present informal speech listening tests at five different packet loss probabilities. It can be seen that a source-channel-optimized MSVQ significantly outperforms a source-optimized MSVQ and indeed further improves the speech quality of packetized speech. Each source-channel optimized MSVQ is trained at the same loss rate used for the corresponding test.

For an objective comparison, we show in Fig. 4 the MSE of the different source-channel coding strategies. The first-index repetition scheme performs slightly better than the source-optimized index-interleaving method. Both are better than traditional frame interleaving even at 30 bits per vector for $p > 0.05$. However, it can be seen that the source-channel optimized

TABLE IV
PREFERENCES IN AN INFORMAL LISTENING TEST INVOLVING 12 LISTENERS. SOURCE-OPTIMIZED MSVQ VERSUS PROPOSED SOURCE-CHANNEL-OPTIMIZED MSVQ. VALUES SHOWN IN PERCENT

Loss Rate p	Source-Optimized MSVQ	Source-Channel-Optimized MSVQ	No Preference
0.10	16.67	54.17	29.16
0.15	8.33	75.00	16.67
0.20	12.50	79.17	8.33
0.30	12.50	62.50	25.00
0.50	16.67	66.67	16.66
Average	**13.33**	**67.33**	**19.17**

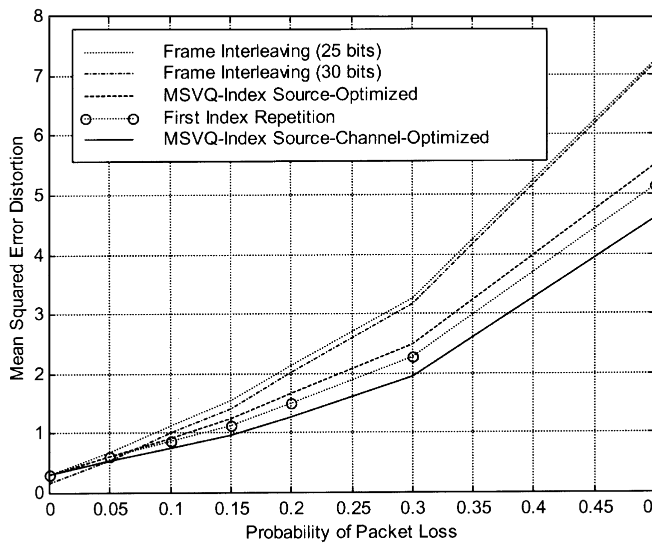


Fig. 4. MSE performance comparison of the following: i) Traditional frame-interleaving using five MSVQ stages (25 bits). ii) Traditional frame-interleaving using six MSVQ stages (30 bits). iii) Frame-interleaving with first index repetition. iv) Source optimized MSVQ-index interleaving using fiev stages packetized using six stages (30 bits). v) Proposed source-channel optimized MSVQ-index interleaving using five stages packeted into six stages (30 bits).

MSVQ outperforms all other strategies for $p > 0.05$. A gain in SNR of up to 2.0 dB can be obtained over the traditional frame-interleaving method. In addition, source-channel optimization outperforms the source optimization by up to 0.87 dB. A comparison in terms of spectral distortion (SD) was found to be slightly misleading. In this case, the proposed source-channel optimized method outperforms the source optimized method by up to 0.3 dB in SD, but the comparison between frame-interleaving and MSVQ-index interleaving was not clear cut. See Table V for the average SD of the various coding schemes. The average SD results seem to show that no gains are achievable for low loss rates. However, by studying the SD outliers shown in Table VI, we find that the source-channel optimized index-interleaving scheme exhibits fewer 2–4 dB outliers and substantially fewer outliers that are more than 4 dB. Since the listening tests in Table III show clear and substantial preference of the proposed method, we may conclude that the overall quality is heavily influenced by the outliers. Note that stabilizing methods

TABLE V
SPECTRAL DISTORTION FOR THE SCHEMES a) TRADITIONAL FRAME INTERLEAVING, b) SOURCE-OPTIMIZED MSVQ INTERLEAVING, AND c) SOURCE-CHANNEL OPTIMIZED MSVQ INTERLEAVING, ALL OPERATING AT 30 BITS PER VECTOR AT VARIOUS TEST LOSS RATES

Loss Rate p	Traditional Frame-Interleaving MSVQ	Source-Optimized-MSVQ	Source-Channel Optimized MSVQ
0.00	0.668	0.912	0.912
0.05	0.860	1.079	1.073
0.10	1.050	1.248	1.219
0.15	1.230	1.408	1.343
0.20	1.448	1.601	1.495
0.30	1.814	1.936	1.762
0.50	2.676	2.744	2.441

TABLE VI
SPECTRAL DISTORTION OUTLIERS (IN PERCENT) FOR THE SCHEMES a) TRADITIONAL FRAME INTERLEAVING, b) SOURCE-OPTIMIZED MSVQ INTERLEAVING, AND c) SOURCE-CHANNEL OPTIMIZED MSVQ INTERLEAVING, ALL OPERATING AT 30 BITS PER VECTOR

Loss Rate p	Traditional Frame-Interleaving MSVQ		Source-Optimized-MSVQ		Source-Channel Optimized MSVQ	
	2-4dB	>4dB	2-4dB	>4dB	2-4dB	>4dB
0.00	0.14	0.00	0.94	0.00	0.94	0.00
0.05	5.99	1.11	6.43	1.06	5.89	0.48
0.10	11.84	2.13	12.38	2.08	10.44	0.88
0.15	16.92	3.22	17.31	3.19	13.65	1.31
0.20	22.55	4.94	22.86	4.78	17.60	2.22
0.30	30.32	8.29	30.82	8.27	22.81	4.32
0.50	42.95	16.35	43.21	16.54	32.02	11.30

have been applied to all coding schemes to maintain stable inverse LSF transformations.

B. Image Transmission

Block-based image coding techniques involve dividing up an image into blocks of uniform size. Then, the blocks are subject to quantization (or transformation followed by quantization). A competitive image coder would most likely incorporate a variable rate quantizer. In an effort to remove complicating factors, the quantizer here is assumed to be fixed rate. For a variable-rate coder, synchronizing techniques can be used.

In Section II, we demonstrated the performance of a source-optimized MSVQ on image compression under conditions of index loss. The case where the MSVQ is source-channel optimized for an index loss probability $p = 0.30$ for the shown transmission vectors is also included in Table II. The results demonstrate the improvements of source-channel-optimized MSVQ over traditional MSVQ. Similar conclusions to those of the speech case can be drawn.

We next test the performance of the image coder in the context of an interleaving scheme for transmission over a packet network. In traditional interleaving techniques, interleaving is performed on the scale of whole image blocks (or MSVQ-parameters thereof), which we call block interleaving. Again, we propose an interleaving scheme that operates on the finer level of MSVQ-indices.

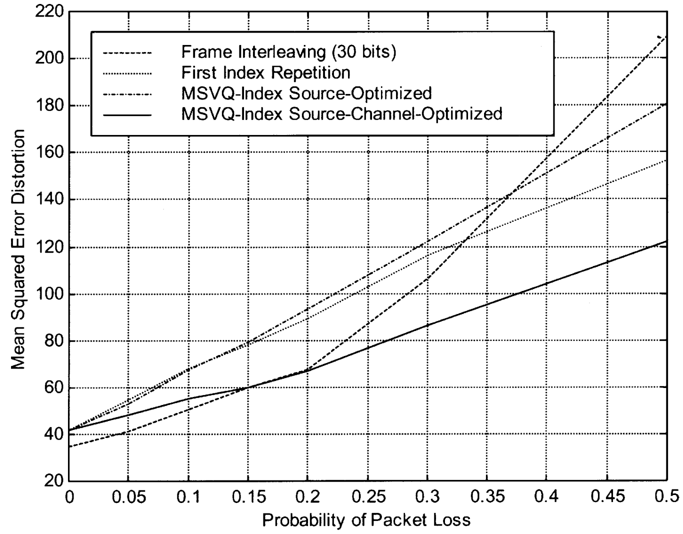


Fig. 5. MSE performance at different packet loss rates for the *Lena* image for i) traditional frame-interleaving (at 30 bits per vector), ii) frame-interleaving with first index repetition, iii) source-optimized MSVQ-index interleaving, and iv) proposed source-channel optimized MSVQ-index interleaving.

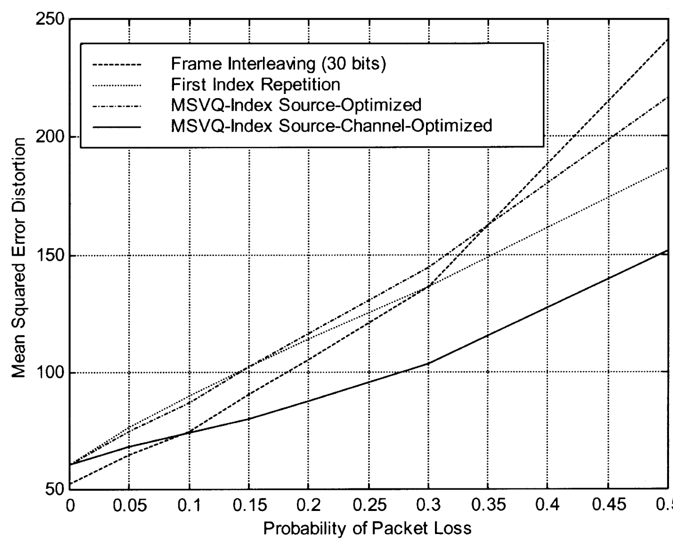


Fig. 6. MSE performance at different packet loss rates for the *Goldhill* image for i) traditional frame-interleaving (at 30 bits per vector), ii) frame-interleaving with first index repetition, iii) source-optimized MSVQ-index interleaving, and iv) proposed source-channel optimized MSVQ-index interleaving.

The source-channel optimized MSVQ interleaving scheme was tested as follows: A group of eight images comprised of Pepper, Zelda, Airplane, Boat, Bridge, Face, Man, and Speedboat were used for training a five-stage MSVQ according to (5) and (7). Blocks of size 8×8 are rearranged into vectors. The performance of the proposed technique is evaluated on an independent test set of two images (*Lena* and *Goldhill*) and compared to traditional source-optimized block-interleaved MSVQ. The nature of the images used for training is quite diverse and not chosen to mirror the statistics exhibited by the test images,

(a)

(b)

Fig. 7. *Lena* image at packet loss rate of 30%. (a) Noninterleaved source optimized MSVQ (PSNR = 27.88 dB). (b) Interleaved source-channel optimized MSVQ (PSNR = 28.78 dB).

thus ensuring reliability of the results. In the proposed method, the MSVQ indices are obtained according to (5) and are interleaved and sent in data packets according to Fig. 3. For the traditional method, the MSVQ indices are obtained using source-optimized MSVQ and are noninterleaved. The interleaved method requires a repetition of the first index, so at a rate of 6 bits per stage, a total rate of 36 bits per block (0.56 b/pixel) is used. For a fair comparison with the interleaved method, an extra MSVQ stage is provided for the noninterleaved method to equate the bit rate. In the case of a complete loss of a block, the conceal-

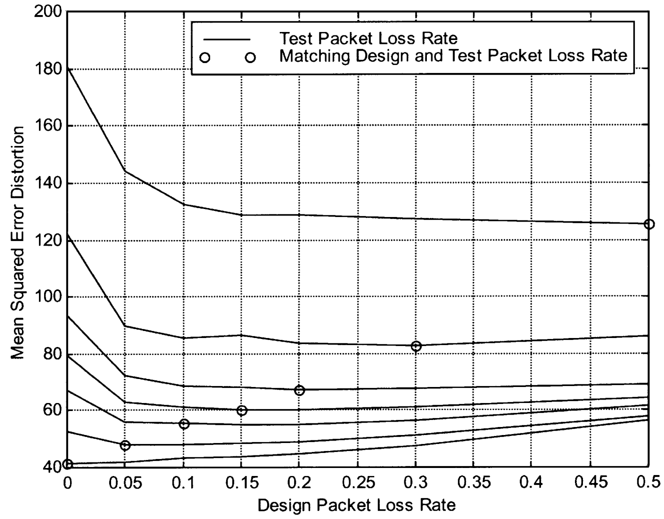


Fig. 8. Performance sensitivity due to *design* versus *actual* packet loss probability rate for the *Lena* image. The curves represent actual loss rates of $p = 0.00, 0.05, 0.10, 0.15, 0.20, 0.30, 0.5$ in order from bottom to top. For each actual loss rate (curve), the performance is shown for each possible training loss rate to demonstrate sensitivity to training. The circles show results when design and actual packet loss rates match and do coincide with each corresponding curve minimum.

ment method in [30] is used. MSE versus packet loss is shown in Figs. 5 and 6 for the images Lena and Goldhill, respectively. Again, considerable gains over traditional block-interleaving of more than 2 dB are achieved at high packet loss rates. The performance of the first-index repetition scheme that exhibits performance similar to the source-optimized MSVQ-index interleaving method if the loss rate is low to moderate, but better if the loss rate is high, is also shown.

In addition, the subjective quality is considerably improved. A subjective comparison at a loss rate of 30% is given in Fig. 7. Under such high loss rates, having the losses distributed among MSVQ indices rather than whole image blocks makes concealment more successful, rendering the image reconstruction much more appealing subjectively. The subjective gains of our proposed method are maintained at lower packet loss rates (the images have been omitted here to save space). It can thus be concluded that the interleaved MSVQ method, when optimized for general loss patterns, can be very robust to packet loss transmission.

The best results are achieved when the interleaved MSVQ is trained at the expected channel loss rate used for the test. In Fig. 8, we demonstrate the design versus test loss rate for the image Lena. Although the gains decrease with mismatched loss rate, they are nevertheless substantial. Such decrease in performance gains is expected and can be similarly observed in multiple descriptions systems.

V. Conclusions

We proposed a new interleaving scheme that is suitable for robust speech and image transmission over lossy packet networks. Further, we developed an optimization algorithm for MSVQ design for packet-switched losses. Both subjective and objective comparisons prove the merit of the proposed approach. Use of the new MSVQ coding approach within more complicated systems may require simple enhancements such as variable rate coding.

REFERENCES

[1] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, pp. 974–997, May 1998.

[2] S. Shirani, F. Kossentini, and R. Ward, "Error concealment methods, A comparative study," in *Proc. Canadian Conf. Elect. Comput. Eng., Eng. Solutions Next Millennium*, vol. 2, Edmonton, AB, Canada, May 1999, pp. 835–840.

[3] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, pp. 40–48, Sept./Oct. 1998.

[4] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Select. Areas Commun*, vol. 18, pp. 966–976, June 2000.

[5] P. Yin, M. Wu, and B. Liu, "A robust error resilient approach for MPEG video transmission over Internet," in *Proc. Int. Conf. Visual Commun. Image Process.*, San Jose, CA, Jan. 2002.

[6] R. Hagen, "Robust LPC spectrum quantization-vector quantization by a linear mapping of a block code," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 266–280, July 1996.

[7] T. Eriksson, J. Linden, and J. Skoglund, "Interframe LSF quantization for noisy channels," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 495–509, Sept. 1999.

[8] J. Linden, "Channel optimized predictive vector quantization," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 370–384, July 2000.

[9] V. A. Vaishampayan, "Design of multiple description scalar quantizers," *IEEE Trans. Inform. Theory*, vol. 39, pp. 821–834, May 1993.

[10] Y. Wang, M. T. Orchard, and A. R. Reibman, "Optimal pairwise correlating transforms for multiple description coding," in *Proc. Int. Conf. Image Process.*, vol. 1, Chicago, IL, Oct. 1998, pp. 679–683.

[11] M. Fleming and M. Effros, "Generalized multiple description vector quantization," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1999, pp. 3–12.

[12] P. Koulgi, S. L. Regunathan, and K. Rose, "Multiple description quantization by deterministic annealing," in *Proc. Int. Symp. Inform. Theory*, Italy, June 2000, p. 177.

[13] W. S. Lee, M. R. Pickering, M. R. Frater, and J. F. Arnold, "A robust codec for transmission of very low bit-rate video over channels with bursty errors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1403–1412, Dec. 2000.

[14] A. R. Reibman, H. Jafarkhani, Y. Wang, M. T. Orchard, and R. Puri, "Multiple description coding for video using motion compensated prediction," in *Proc. Int. Conf. Image Process.*, vol. 3, Kobe, Japan, Oct. 1999, pp. 837–841.

[15] R. Arean, J. Kovacevic, and V. K. Goyal, "Multiple description perceptual audio coding with correlating transforms," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 140–145, Mar. 2000.

[16] V. Hardman, A. Sasse, M. Handley, and A. Watson, "Reliable audio for use over the Internet," in *Proc. INET*, June 1995.

[17] Y. J. Liang, E. G. Steinbach, and B. Girod, "Real-time voice communication over the Internet using packet path diversity," *ACM Multimedia*, pp. 431–440, 2001.

[18] N. Iwakami, T. Moriya, and S. Miki, "High-quality audio coding at less than 64 kbit/s by using TwinVQ," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 1995, pp. 937–940.

[19] A. Kataoka, T. Moriya, and S. Hayashi, "An 8-kb/s conjugate structure CELP (CS-CELP) speech coder," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 401–411, Nov. 1996.

[20] ITU-T G.723, "Dual rate speech coder for multimedia communication transmitting at 5.3 & 6.3 kbit/s," ITU-T, Oct. 1995.

[21] A. V. McCree and T. P. Barnwell, III, "A 2.4 kbits/s MELP coder candidate for the new U.S. federal standard," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, GA, May 1996, pp. 200–203.

[22] F. Kossentini, M. J. T. Smith, and C. F. Barnes, "Image coding using entropy-constrained residual vector quantization," *IEEE Trans. Image Processing*, vol. 4, pp. 1349–1357, Oct. 1995.

[23] H. Khalil and K. Rose, "Asymptotic closed-loop design of predictive multi-stage vector quantizers," in *Proc. Int. Conf. Image Processing*, Vancouver, BC, Canada, Sept. 2000.

[24] S. Gadkari and K. Rose, "Unequally protected multi-stage vector quantization for time-varying channels," in *Proc. Int. Conf. Commun.*, vol. 2, Atlanta, GA, June 1998, pp. 786–790.

[25] ——, "Unequally protected multistage vector quantization for time-varying CDMA channels," *IEEE Trans. Commun.*, vol. 49, pp. 1045–1054, June 2001.

[26] W. P. LeBlanc, B. Bhattacharya, S. A. Mahmoud, and V. Cuperman, "Efficient search and design procedures for robust multi-stage VQ of LPC parameters for 4 kb/s speech coding," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 373–385, Oct. 1993.

[27] B. H. Juang and A. H. Gray, "Multiple stage vector quantization for speech coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, Apr. 1982, pp. 597–600.

[28] F. Jelinek and J. B. Anderson, "Instrumentable tree encoding of information sources," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 118–119, Jan. 1971.

[29] C. F. Barnes and R. L. Frost, "Vector quantizers with direct sum codebooks," *IEEE Trans. Inform. Theory*, vol. 39, pp. 565–580, Mar. 1993.

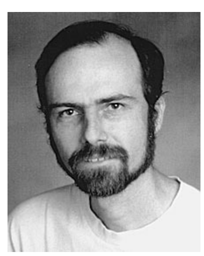
[30] Z. Wang, Y. Yu, and D. Zhang, "Best neighborhood matching: An information loss restoration technique for block-based image coding systems," *IEEE Trans. Image Processing*, vol. 7, pp. 1056–1061, July 1998.

[31] H. Khalil and K. Rose, "Selective splitting approach to entropy-constrained single/multi-stage vector quantization design," in *Proc. SPIE, Image Video Commun. Process.*, vol. 3974, San Jose, CA, Jan. 2000, pp. 81–88.

Hosam Khalil (S'98–M'01) was born in Washington, DC, in 1972. He received the B.S. and M.S. degrees in electrical engineering with honors from Cairo University, Cairo, Egypt, in 1993 and 1996, respectively, and the Ph.D. degree from the University of California, Santa Barbara, in 2001.

From October 1993 to July 1996, he was an Assistant Teacher at Cairo University. He was also employed part-time during that period with IBM, Cairo. In the summer of 1998, he was an intern with the Multimedia Communications Research Laboratory, Bell Laboratories, Murray Hill, NJ. From March 1999 to March 2000, he was a Software Engineer at SignalCom, Inc., Goleta, CA. In April 2000, he joined the Digital Media Division, Microsoft Corporation, Redmond, WA, where he is a Software Design Engineer. His research interests are in image, video, and speech compression and speech recognition.

Mr. Khalil was a recipient of the UC Regents Fellowship.



Kenneth Rose (S'85–M'91–SM'01–F'03) received the B.Sc. and M.Sc. degrees in electrical engineering from Tel-Aviv University, Tel-Aviv, Israel, in 1983 and 1987, respectively, and the Ph.D. degree from the California Institute of Technology, Pasadena, in 1991.

Since January 1991, he has been with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, where he is currently a Professor. His main research activities are in the areas of information theory and signal processing and include rate-distortion theory, source and source-channel coding, audio and video coding and networking, pattern recognition, and nonconvex optimization. He is particularly interested in the relations between information theory and statistical physics and in the application of information and estimation theoretic approaches to fundamental problems in signal processing.

Dr. Rose currently serves as Editor for source-channel coding for the IEEE TRANSACTIONS ON COMMUNICATIONS. He co-chaired the technical program committee of the 2001 IEEE Workshop on Multimedia Signal Processing. In 1990, he received (with A. Heiman) the William R. Bennett Prize Paper Award from the IEEE Communications Society.