Robust Low-Delay Audio Coding Using Multiple Descriptions

G. Schuller[1], J. Kovačević[2], F. Masson[3] and Vivek K Goyal[4]

Abstract— **This paper proposes an encoding method for high-quality, low-delay audio communication that is robust to losses in packetized transmission. Robustness is provided by a multiple description vector quantization (MDVQ) technique that is designed to minimize the mean-squared error (MSE). The key to applying this technique effectively is the use of psycho-acoustically controlled pre- and post-filters that make the mean-squared quantization error perceptually relevant. Experiments show that the MDVQ-based encoder yields better results—in both MSE and subjective audio quality—than simple alternative coders with the same low delay.**

I. Introduction

Technological progress has made the public Internet infrastructure faster and has given more users high-bandwidth access to this infrastructure. Nevertheless, applications requiring both high data rate and low delay remain largely limited to private networks. Examples of such applications are video conferencing with high quality for both the video and the audio, musicians playing together remotely, wireless speakers and wireless microphones. The reason is simply that packet losses greatly impact the quality of streaming media, and eliminating packet losses introduces delay. We assert that now and for the foreseeable future, packet losses are significant; thus, media representations (encodings) for low delay applications must be made more tolerant to losses. Packet losses occur in wireless networks as a result of interference or noise, and in wired networks they occur from interactions from other traffic.

Those who would argue that network loss rates are decreasing must realize that most congestion control is achieved only as a response to packet losses. Therefore, even moderate aggregate link utilization by a set of network flows typically causes losses for all of the flows unless all of the flows operate at constant rate.

The problem of packet lossess could be alleviated with priority labled packets, where the network discards mostly the lower priority packets. But this requires a network with this feature. For wireless connections this would not be a solution, because interference and noise affects every packet with equal probability.

In this paper, we describe a technique for low-delay audio coding that is robust to packet losses. Robustness without added delay is obtained with multiple description coding [1]. Our requirements for the end terminals are threefold: First, the encoding/decoding process should add little delay to the signal path. A reasonable target for this delay is 10 ms or lower, which is on the lower end of the encoding/decoding delay of speech coders (see also [2] for a delay discussion). This is sufficient even for the most demanding applications. Then, both for delay and transmission reasons, the encoding/decoding scheme should provide graceful degradation in the presence of packet losses. Finally, the audio signal needs to be sufficiently compressed to be suitable for transmission over bit-rate restricted channels, as in wireless connections or over ISDN. We consider two aspects of the above problem: The first is a source (specifically audio) coding method with sufficient compression ratio and low delay, and the second is a source/channel coding scheme to treat transmission losses, again with low delay.

One of the simplest mechanisms to deal with packet losses is to retransmit the lost packets until they are correctly received. Such protocols require communication from the receiver to the sender—either acknowledgements of received packets or negative acknowledgements of lost packets. However, this technique is often not applicable in real-time systems because the acknowledgement and retransmission process adds too much delay.

Another possibility is to try to conceal the losses by predicting the lost samples from their neighbors. If one packet is lost, the receiver tries to guess the value of the lost samples by using the previous samples successfully transmitted. This technique works reasonably well for speech signals but can be problematic for non-speech signals like music.

Multiple description coding (MDC) is used to provide robustness to packet losses by introducing redundancy in the transmitted streams, without adding delay or prohibitive complexity. The price we pay is an increased bit rate. Instead of retransmitting packets, redundancy is added to the source before transmission by creating several descriptions of the source. MDC has the advantage that no delay is added, and that it does not rely on knowledge of the sound source.

II. Audio Compression

MDC techniques are generally developed to minimize the mean squared error (MSE) of the reconstructed signal. But for the playback of audio signals this error measure is not optimal because of masking effects of the ear. The audibility of distortions depends strongly on the underlying signal and the sensitivity of the ear across frequency and

This work was done while the authors were with Bell Labs and the third author was an intern at Bell Labs

[1]Now with (contact) the Fraunhofer Institute IDMT, Langewiesener Str. 22, 98693 Ilmenau, Germany, shl@idmt.fhg.de, phone: ++49-3677-467 110

[2]Now with Carnegie Mellon University, Pittsburg, PA

[3]Now with ELCA Informatique, Geneva

[4]Now with Massachusetts Institute of Technology, Cambridge, MA

time. These effects are described by the signal-dependent psycho-acoustic masking threshold. Distortions which are smaller than this masking threshold are not audible. To apply MSE-based MDC techniques to audio coding, we desire a mapping of the audio signal to a domain where MSE is approximately commensurate with the audibility of distortions. To obtain this mapping, we use a psycho-acoustically controlled adaptive pre-filter. It has the effect that it normalizes the signal to its masking threshold. On the decoding side we use a post-filter, inverse to the pre-filter.

Most present audio coders are based on subband coding. A good compression ratio requires a high number of subbands, typically 1024, at sampling rates of 32 to 48 kHz. However, this high number of subbands leads to a high encoding/decoding delay, on the order of 100 ms and more. The MPEG4 low-delay coder achieves a lower delay by using a smaller number of subbands, leading to a compromise in the compression performance. But the obtained delay (ca. 960 samples, which is 20 ms at 48 kHz sampling rate or 30 ms at 32 kHz sampling rate) is not as low as desired (10 ms). Speech coders achieve lower delays but do not perform well on non-speech signals such as music or room noise. Thus, to lower the delay without sacrificing performance, we take a different approach.

Predictive coding introduces little or no delay and has the same asymptotic coding gain as subband coding [3], [4]. However, predictive coding cannot easily be combined with a psycho-acoustic model. Our approach separates *irrelevance reduction* (quantization with a resolution that makes it imperceptible, at least with no transmission losses) from *redundancy reduction* (the exploitation of statistical relationships and non-uniform probability densities in the quantized data).

A. Irrelevance Reduction

The pre- and post-filter are linear adaptive filters, implemented in a structure like predictors, which provides invertibility. Their uses are illustrated in Fig. 1. The pre-filter frequency response $H(f)$ is a normalization of the signal to the masking threshold $M(f)$,

$$H(f) \;=\; \frac{1}{M(f)}.$$

This means that after pre-filtering, the masking threshold of the signal is at unity acorss frequency. The uniform (white) noise shape across frequency correponds to a constant variance noise in the time-domain. The perceptual model is tuned in a way that a simple rounding operation (unit step size) produces a suitable quantization noise at the masking threshold. Any distortion above above this level becomes audible, whereas distortions below it remain inaudible. The post-filter in the decoder is the inverse of the pre-filter. It has a frequency response like the masking threshold. Assuming the quantization distortions after the pre-filter are flat across frequency and time, the post-filter shapes the quantization distortions like the masking threshold, as desired. The coefficients of the pre- and post-filter are obtained by computing the linear predictive coefficients (LPC) from the output of the psycho-acoustic model [5], [6], such that a frequency response according to the model is obtained. The masking threshold is parameterized using the pre-filter coefficients, and transmitted as side information to the post-filter in the decoder.

In the original formulation and application of this pre-filter [5], [6], the output of the pre-filter was input to a uniform quantizer. The uniform scalar quantizer is replaced with a multiple description vector quantizer; this does not alter the spectral flatness of the quantization error.

The quantizer produces the desired spectrally flat quantization distortions. The pre-filter together with the quantizer can be viewed as a stage for the irrelevance reduction, because it introduces distortions which are not audible (at least ideally), and after the quantization the signal has a lower entropy. This stage introduces some delay because the psycho-acoustic model is still subband based. However, the requirements for the time/frequency resolution for the psycho-acoustics are different than the requirements for subband coding in traditional audio coding. That is why the number of subbands can be chosen much smaller. In our implementation we chose 128 subbands, leading to a delay of about 128 samples.

B. Redundancy Reduction

The quantizer is followed by lossless coding, implemented with a predictor and an entropy coder, such as an adaptive Huffman coder. This stage can be viewed as redundancy reduction, because it uses only the statistical dependencies of the signal.

The stage for the redundancy reduction does not introduce much delay either. The prediction can be implemented with backward adaptation, which is based on past signal samples, and hence has no delay. Adaptive Huffman coding has a delay of about 20 samples in our implementation [7], [6]. The decoder does not introduce additional delay. This means that the overall encoding/decoding delay is on the order of 200 samples or 6 ms at 32 kHz sampling rate, which is below our targeted delay.

III. Multiple Description Background

Multiple description coding is a set of techniques that create several descriptions of a signal to transmit. The descriptions are self-contained but correlated. Each description can be viewed as a coarse approximation of the input signal. The different descriptions are transmitted separately to the receiver.

Descriptions can be lost on their way to the receiver if their corresponding channels are broken. At the receiver, the quality of the decoding is based on the number of descriptions correctly received. If M descriptions of the input signal are created, the receiver has $2^M - 1$ different decoding "behaviors", one for each nonempty set of descriptions received:

- If all descriptions are correctly received, the input signal can be reconstructed at full quality.

• If only a subset of the descriptions is received, the receiver can still reconstruct the signal and produce a coarse approximation of the source.

In MDC, the higher the number of descriptions received, the smaller the distortion between the input signal and its reconstructed value. In contrast to a layered coding scheme—where one channel is assumed to be received and there is an assumed priority order among the descriptions—in MDC every description is at the same priority level, and as soon as any of the descriptions is correctly received the decoder can compute an estimation of the original stream of data.

A basic two-description MD system is illustrated in Fig. 2. Two descriptions of the source are created and transmitted over two separate channels. The receiver uses one of three decoding procedures, depending on which descriptions are received. When both descriptions are received, the receiver uses the "central decoder" D_0; when only one description is received, the receiver uses one of the "side decoders" D_1 and D_2. The two side decoders have bigger distortions than the central decoder, but their outputs are still coarse approximations of the input signal. It is also possible that neither description is received, but in this case the receiver can do nothing more than approximate the signal by its mean. The overall goal of the design of an MD coder is to make the distortion of all of the 2^M-1 decoders as small as possible.

Two extreme cases of MDC are to: (a) repeat exactly the same description on all channels or (b) create completely independent descriptions. In the first case, the reception of one description already leads to full quality reconstruction. For a two-description system, this ensures complete robustness to the failure of one channel, but the transmission overhead introduced is 100%. In the second case, the descriptions are completely independent and no redundancy is introduced. However, no robustness is achieved either. If one description is lost, the information contained in the other description cannot be used to reconstruct lost information. Therefore, we see that there is a trade-off between the redundancy introduced during the creation of the descriptions and the robustness of the transmission. Good robustness to losses can be achieved, but a price is paid in the increase of the transmission rate. Next, we briefly review multiple description lattice vector quantization (MDLVQ), which will be used in our system for robust audio transmission. More details on MDC can be found in [1].

A. MD Lattice Vector Quantization

In a classical scalar quantization scheme, for each input sample the nearest quantizer codebook index is transmitted. In the MD case, the index of the scalar quantizer is not sent directly over the channel. Rather, an index assignment table is used to create two descriptions of every bin's index [8]. Then, each description is sent on a different channel, and there are three possible decodings at the receiver. Even if one description is lost, the other description can be used to produce a coarse approximation of the original sample.

Just as we can form descriptions by using separate quantizers on each scalar input sample, we can form descriptions on blocks of K input samples. This has the advantage of reducing the quantization error for a given bit rate (a property of vector quantization) as well as obtaining more flexibility in the design of our multiple description scheme, because we consider the quantization distortion cumulative over K samples, and not for each individually.

Here we apply two-dimensional quantizers, i.e., we encode with blocks of length $K = 2$. This allows us to use the example quantizers based on the hexagonal A_2 lattice presented explicitly in [9], which in turn are based on the optimizations for the A_2 lattice presented in [10]. The choice of $K = 2$ provides a concrete proof of concept and facilitates pictorial representations. It also has an audio inspiration: We do not want to make the dimensionality too high to avoid having the quantization error too unevenly distributed over the samples. Using psychoacoustic prefiltering with moderate- to high-dimensional vector quantization is an open research area that we cannot address significantly within the scope of this work.

Even without the multiple description flavor, vector quantizers suffer from great encoding complexity. A way to deal with this problem is to impose structure on the quantizer, such as forcing the points to belong to a lattice. In lattice vector quantization, every vector of data is quantized to one point of a given lattice. Finding the nearest point from a lattice has much lower complexity than finding the nearest point in an unstructured codebook [11].

In Multiple Description Lattice Vector Quantization (MDLVQ), instead of transmitting a label corresponding to the closest lattice point, one associates with the lattice point an ordered pair of points in a sublattice. The indices of these sublattice points are the descriptions and the sublattice points are the side decoder reconstructions. The association of lattice points to ordered pairs of sublattice points is one-to-one so that the central decoder reconstruction can be the original lattice point.

MDLVQ was introduced by Servetto, Vaishampayan and Sloane (SVS) [12], [10]. In addition to providing the basic framework, they gave an algorithm for determining optimal index assignments. Kelner, Goyal and Kovačević (KGK) [13], [9] recognized that the encoding procedure is inherently optimized for the central decoder, meaning it minimizes the average distortions for the case of no losses. They proposed an extension of MDLVQ in which the encoder is optimized for a weighted combination of the central and side distortions. We now provide details on the original SVS technique and the KGK modification that is used in our MD audio coder.

Let Λ be a lattice, and let $\Lambda' \subset \Lambda$ be a geometrically similar sublattice of Λ. This means $\Lambda' = cA\Lambda$ for some scalar c and some unitary matrix A with determinant 1, or that Λ' is obtained by scaling and rotating Λ. The index $N = |\Lambda/\Lambda'|$, which can be seen as relative density of the lattices, ultimately determines the redundancy of the system. Every point of the original fine lattice Λ is labeled

with a pair of points of the sublattice Λ' by using a one-to-one *index assignment* $\ell : \Lambda \longrightarrow \Lambda' \times \Lambda'$. Fig. 3 shows an example in which the original lattice is the two-dimensional hexagonal lattice A_2 and Λ' is an index-7 sublattice.

In the SVS technique, a point is first encoded to the closest fine lattice point $\lambda \in \Lambda$ and then $(\lambda'_1, \lambda'_2) = \ell(\lambda)$ is computed. This lattice quantization uses the fast encoding algorithm described by Conway and Sloane in [11], [14] for the $\Lambda = A_2$ example, and creates hexagonal Voronoi regions. Recall that the Voronoi region of a lattice point is defined as the set of points closer to this lattice point than to any other. λ'_1 and λ'_2 are transmitted over channel 1 and 2, respectively. If only description i is received, the reconstruction is λ'_i. If both descriptions are received, the receiver can decode to the original lattice point λ. Therefore, the decoder provides coarse information if only one description is received, and finer information if both descriptions are transmitted successfully.

This approach suffers from the following drawback: Since the decoding is made at the resolution of the fine lattice only when both descriptions are received, it performs best for the central decoder (for which no description is lost), and does not consider the decoding performance of the side decoders based on the reception of only one description.

Therefore, KGK propose in [13], [9] a new criterion for the initial encoding step, applied before the index assignment. They encode $x \in \mathbb{R}^N$ to the lattice point $\lambda \in \Lambda$ that minimizes

$$\frac{1-p_l}{1+p_l} \cdot \|x-\lambda\|^2 + \frac{p_l}{1+p_l} \cdot \left(\|x-\lambda_1\|^2 + \|x-\lambda_2\|^2\right), \quad (1)$$

where $(\lambda_1, \lambda_2) = \ell(\lambda)$. This expression is a convex combination of the squared error at the central decoder $\|x-\lambda\|^2$ and the average squared error at the side decoders $\frac{1}{2}\left(\|x-\lambda_1\|^2 + \|x-\lambda_2\|^2\right)$, The parameter p_l controls the trade-off between central and side distortions. It can be considered the designed loss probability because the expression that is minimized is the expected squared error, conditioned on at least one description being received, when descriptions are lost independently with probability p_l. This encoding partitions $\mathbb{R}^N$ differently than nearest-neighbor encoding with respect to Λ.

KGK further propose to alter the locations of points in $\Lambda \setminus \Lambda'$ to minimize (1). An iterative algorithm for this perturbation is given in [9]. The shapes of the resulting partition cells are given in Fig. 4 for a few values of p_l. The evolution of the partition as p_l increases is interesting. When $p_l = 0$, the partition is the Voronoi partition used by SVS because the lattice is not perturbed and the side distortions are given no weight in (1). As p_l increases, the cells around the sublattice points become larger than the ones around the points of $\Lambda \setminus \Lambda'$. The sublattice points are preferred for encoding because when losses occur, these points are decoded without error even in the side decoders. In the extreme case where $p_l = 1$, the cells around points in $\Lambda \setminus \Lambda'$ disappear.[1]

[1]Animations of this evolution can be found at `http://lcavwww.epfl.ch/~goyal/MDVQ/`.

The encoding with the perturbed lattice is similar to the one with the regular (unperturbed) lattice, but requires more computation. Assume again that we want to encode the point $P = (p_1, p_2)$.

- First, P is vector quantized to the closest *sublattice* point $\lambda'_p \in \Lambda'$, using the same fast encoding algorithm described in the previous section. We cannot use this algorithm for vector quantizing to the fine lattice Λ as in the previous section, because (1) is not standard Euclidean distance and the fine lattice has been perturbed.
- Then, using the difference $P - \lambda'_p$, find the $\lambda \in \Lambda$ that minimizes (1). For the hexagonal lattice, $N = 7$ case, this is a search among 13 candidates.
- Once $P - \lambda'_p$ is determined, use the labeling ℓ to construct the two descriptions and transmit them over their respective channels. This labeling, determined with the SVS algorithm, is the same as the one used with the regular lattice.

The decoding algorithm is exactly the same as the one used for the regular lattice in the previous section.

IV. The MD Coder

We now present the MD coder we implemented for the encoding of the pre-filtered signal. Its block diagram is given in Fig. 5. The audio signal is first pre-filtered and then input into an MDLVQ encoder. This coder pairs the samples and outputs two descriptions for every vector created. Then, each description is passed through a lossless coder to remove the redundancy contained in the streams, before transmission over its channel. The lossless coder consists of a predictor and an entropy coder.

A. MDLVQ Encoder

As described in Section II-A, the psycho-acoustically controlled pre-filtering results in a signal for which uniform scalar quantizer step size $\Delta = 1$ is at the threshold of perceptibility. So that integer audio file formats allow sufficient resolution for our manipulations and comparisons, we scale the pre-filter output by 100. (Now $\Delta = 100$ is at the threshold of perceptibility.) Obviously, the factor of 100 is arbitrary and has little impact on our results.

We use an MDLVQ as a replacement for the uniform scalar quantizer. Specifically, we apply the modified version of MDLVQ from [9] with the A_2 lattice and an optimal index assignment function for sublattice index $N = 7$. As a design criterion, we want the central distortion with parameter $p_l = 0$ to be at the threshold of perceptibility, i.e., the same as the distortion obtained with a uniform quantizer with $\Delta = 100$. This simply requires an appropriate scaling for the A_2 lattice.

Let R be the radius of a circle inscribed in a hexagonal Voronoi region of the desired lattice. Under the usual high-rate analysis assumptions, the distortion for uniform scalar quantization is $\Delta^2/12$. This is the square of the scaling of the underlying $\mathbb{Z}$ lattice (Δ^2) times the normalized second moment of the $\mathbb{Z}$ lattice ($1/12$). Making the corresponding calculation for the two-dimensional quantizer gives distor-

tion $5R^2/24$. Thus we choose

$$R = \sqrt{\frac{2}{5}}\Delta \approx 63.24.$$

B. Lossless Predictive Coder

At the output of the MDLVQ encoder, on each MD channel, the sublattice points have statistical dependencies. This is why each description is passed into a lossless coder using a predictor to reduce the bit rate needed for the transmission.

We describe each (sub-)lattice point by two integers or coordinates (its projections onto a particular basis set) as shown in Fig. 3. The two descriptions of the pre-filtered signal are sequences of coordinates of sublattice points. An illustrative example for a first description might be: (-1,3), (0,-2),... and the second description might be: (0,3), (1,-2),....

We use prediction filters updated by the LMS algorithm [15], [16]. The LMS algorithm uses the prediction error to update the coefficients of the filter, as shown in Fig. 6. For example, assume that $H_{M,k}$ is the filter used to predict the coordinate x_k by using the M previous coordinates $X_k = (x_{k-1}, \ldots, x_{k-M})^T$. The prediction $\hat{x}_k$ of the current coordinate x_k is given by

$$\hat{x}_k \;=\; X_k^T \cdot H_{M,k} \;, \tag{2}$$

Observe that the predictor uses the coordinates of both bases. The error e_k of the prediction is

$$e_k \;=\; x_k - \hat{x}_k \;=\; x_k - X_k^T \cdot H_{M,k}. \tag{3}$$

This prediction error is used to update the filter $H_{M,k}$:

$$H_{M,k+1} \;=\; H_{M,k} + \mu e_k X_k,$$

where μ is the step size of the LMS algorithm. The larger the step size, the faster the convergence of the algorithm, but the larger the asymptotic average mismatch between the adaptive filter and the optimal filter. Therefore, there is a trade-off in the choice of the step size μ. A large μ will converge faster in the beginning, but after convergence the resulting filter will be worse than the filter obtained with a smaller step size. Under a standard but imprecise analysis, for stability and convergence, the step size μ must obey [16]

$$0 < \mu < \min_{i \in \{1,\ldots,M\}} \frac{1}{\lambda_i} \;,$$

where the λ_i are eigenvalues of $E(X_k X_k^T)$. Since $\max(\lambda_i) < M\sigma_X^2$, where σ_X^2 is the power of the input samples x_k, a sufficient and simpler upper bound is given by:

$$0 \;<\; \mu \;<\; \frac{1}{M\sigma_X^2}. \tag{4}$$

Since the samples contained in the streams are integers, the prediction must be an integer, too. Therefore, the prediction given in (2) is rounded to the nearest integer[2], and the prediction error defined in (3) is now given by

$$e_k \;=\; x_k - \left[X_k^T \cdot H_{M,k}\right].$$

These prediction errors e_k are the output of the predictive block. They are passed to the entropy coder and then transmitted to the receiver. The decoder uses them to exactly reconstruct the stream of coordinates of the sublattice points. The decoder has to use the same arithmetic (for instance the same precision) as the encoder to obtain an exact reconstruction.

C. Coder Used in the Experiments

The coder uses two different prediction filters. One is used to predict the odd indexed coordinates, which is only updated on the odd indexed coordinates. Similarly, the other prediction filter is used to predict the even indexed coordinates, and is updated only on the even indexed coordinates. This structure is used for both descriptions, for a total of 4 predictors.

We used two different audio signals for conducting most of our experiments: "jazz" containing classical jazz music, and "mixed" , which is a commercial containing a mix of speech and music. Both signals have a duration of 10 seconds or 320 000 samples at 32 kHz sampling rate. These signals are simply called "jazz file" and "mixed file". We first used these files to test the behavior of the coder for the encoding of the streams generated by the MDLVQ encoder. For performance comparisons, we computed the first-order entropy of the transmitted symbols.

Since the coordinates to transmit will be grouped into packets and some of the packets will be lost, the prediction filter will be periodically reset. This reset is needed to avoid any mismatch between the adaptations in the decoder and in the encoder. We decided to perform this reset every 4096 coordinates. This does not imply that the size of a packet must be 4096 coordinates. Actually, the number of coordinates contained in a packet can be less or equal than 4096. A lower number is desirable to obtain a lower delay. However, if one packet containing some of the 4096 coordinates is lost, the corresponding description will be declared lost even if some of the coordinates contained in another packet are received.

This periodic reset favors the choice of a relatively large step size in the predictor to achieve fast convergence.

The experiments we conducted with the predictive filters had two goals: (a) to see the effect of the periodic reset, and (b) to find the optimal filter length and LMS step size to use for the filters.

Our simulations are conducted with two variables:

- M is the number of coordinates of the same type (that is, along the same basis vector E_i) used in the prediction. Since there are 2 coordinates for each vector or pair of pre-filtered signal samples, the length of the prediction filter is $2M$.
- μ_{factor}: This parameter is used in the computation of the step size μ. Using the simpler upper bound (4), the step size μ_j used for the prediction filter applied to the

[2]The notation $[x]$ is used for the nearest integer to x.

coordinates of description j is computed with the following formula:

$$\mu_j = \frac{1}{2M\mu_{factor}\sigma^2_{X_j}}. \quad (5)$$

For experiments to determine good values for M and μ_{factor}, we assumed a very noisy transmission channel, like in wireless communications, a 20% packet loss for the design of the perturbed lattice. We used four different values for the parameter M $(4, 8, 16, 32)$, and five different values for μ_{factor} $(5, 10, 15, 20, 25)$. For the prediction error we compute the first order entropy for pairs of coordinates. The entropies of both descriptions are added to get an approximation of the bit rate needed to transmit the stream. We also considered two different cases for the reset of the prediction filters. In the first series no reset is performed. In the second series, a reset is performed every 4096 coordinates. This helps us to estimate the influence of different reset periods for different predictor operating points.

As results of of our experiments we found that, when no reset is performed the larger the value of M, the smaller the bit rate (that is, the better the performance). This is due to two facts: First, since the step size is computed with (5), the larger the value of M, the smaller the step size, and the better the filter in the steady-state. Moreover, for many signal parts the longer prediction is better. However, this is not always exactly true in our experiments. We see that for the "mixed" file, the performance with $M = 16$ is better than the one obtained with a longer filter ($M = 32$) for the large values of μ_{factor}.

When a reset is performed (lower plots), the best performance is obtained with the smaller values of M, that is, with the short filters. Because of (5), the smaller M, the larger the step size, and the faster the convergence of the filter.

As an example, we set the value of M to 8 for the experiments that follow. Since the plots show that with $M = 8$ the smallest bit rate is achieved with $\mu_{factor} = 10$, we keep that value for the remaining MDLVQ experiments.

The total signal delay of this system consists of the 128 samples of the psycho-acoustic pre-filter plus the assembly delay for the packets. The decoder does not add delay. If the packets have a size of, say, 256 samples, this adds up to a total of 384 samples, or about 10 ms, which conforms to our goal.

V. Simple Comparison Coders

To give an impression of the performance of our system, a comparison to other MD schemes is useful. Since there are not many well-known MD coders, we also designed three simple comparison coders. Each one is introducing a different amount of redundancy. They are denoted by BC0, BC2 and BC4 coders ("BC" stands for "Basic Coders").

Coder BC0: The pre-filtered signal is first quantized with a scalar quantizer of bin width $\Delta = 100$. The output of the scalar quantizer is then passed into a lossless coder, consisting of a predictor and an entropy coder. The output of the lossless coder is split into two streams: packets of 2048 consecutive coded samples are created and then transmitted alternately over each channel.

This coder does not introduce any redundancy in the streams. If one packet is lost, zeros will be input in the decoder before post-filtering.

Coder BC2: As in the BC0 coder, the pre-filtered signal is first quantized, still with the same scalar quantizer of bin width $\Delta = 100$. Then, consecutive samples are paired and input in a integer-to-integer Hadamard transform (see, e.g., [17, App. I]). The outputs of the block correspond respectively to the low-pass (sum) and high-pass (difference) values.

To introduce some redundancy and robustness in the transmission, the low-pass components are repeated on both channels, and the high-pass components are split between the channels. The first packet contains all the low-pass samples output by the integer-to-integer Hadamard transform, as well as half of the bits of the high-pass stream. The second packet contains again all the low-pass samples, and the other half of the high-pass stream. On the receiving side, the decoder can have three different behaviors, depending on the number of packets received: (a) If both packets are successfully received, the original samples can be retrieved by inverting the transform. (b) If only one packet is received, only the low-pass samples are available. These samples are used as input in the inverse transform to get a coarse approximation of the original samples before post-filtering. (c) If both packets are lost, zeros are input in the post-filter.

We see that this scheme is actually a simple MD scheme, where one packet carries a coarse approximation of the original samples (the low-pass output of the transform in our case), and both packets allow a perfect reconstruction of the original samples. For more details on forming an integer-to-integer transform, see [17].

Coder BC4: This coder is similar to the BC2 coder, but now with a 4 $\times$4 integer-to-integer Hadamard transform. Consecutive samples are grouped into vectors of 4 samples and input in the transform block. After the transform, the low-pass component is, as in the BC2 coder, passed into a lossless coder and then repeated over both channels. The three other high-pass components are not repeated over both channels; their respective streams are split between the two channels.

On the receiving side, the decoder behaves similar as the BC2 coder does: If both packets are received, the 4 streams can be reconstructed and the inverse of the transform can be exactly computed. If only one packet is received, only the low-pass stream is entirely received and can be used to compute a coarse approximation of the original samples before post-filtering.

When compared to the BC2 coder, the BC4 coder introduces less redundancy since only one of four outputs of the transform is repeated on both channels. Therefore, its bit rate is smaller than the bit rate of the BC2 coder. However, the disadvantage is that the BC4 coder is less robust to losses, as we will see in the experiments.

A. Lossless Predictive Coder

As in the MDLVQ coder we use lossless predictive coders in the basic coders to reduce the redundancy of the streams and therefore reduce the bit rate. In these basic coders, since the streams to encode do not consist of pairs of coordinates, as in the MDLVQ case (cf. Section IV-B), a predictor H_M of length M is used to predict the next sample. For each of these coders, we have to choose the best filter length M and the best step size μ to lower the bit rate as much a possible. The step size μ is computed as with (5), but with 'M' instead of '$2M$'

For each coder, we ran the filter for $M \in \{4, 8, 16, 32\}$ and $\mu_{factor} \in \{5, 10, 15, 20, 25\}$. When the BC0 coder is used, the prediction filters are reset every 2048 samples, because we want to be able to recover just one description with the 2048 samples. The reset period for the BC2 and BC4 coders is 4096, like for the MDVQ case. At the output of the coders, the first-order entropy of the symbols is computed.

The results of our experiments show that $M = 16$ and $\mu_{factor} = 10$ is best for the BC0 coder. We use $M = 32$ for the BC2 and BC4 coders.

VI. Model for the Network

A. Number of Channels

In our experiments, we assumed that two descriptions are created and transmitted over their respective channels to the receiver. We optimistically made the assumption that one can establish two different independent connections, and that each description can be transmitted over one of these. This would allow the packet losses over each channel to be independent.

However, it is a common misconception that MD coding requires an *independent* path or transport mechanism for each description. While such a situation does make MD coding particularly attractive, as long as there is a chance that exactly one of two descriptions is received it may be beneficial to use MD coding. It is not a pre-requisite to have independent paths.

VII. Experimental Results

We now compare performances of the MDLVQ and the basic comparison coders. We first simulate the case that the descriptions are lost independently. Then, we study the performance when exactly one description is used for the decoding.

A. When Descriptions are Lost Independently

Every packet can be lost independently over each channel with probability p. Therefore, either both (with probability $(1-p)^2$), one (with probability $2p(1-p)$) or no (with probability p^2) descriptions are received by the decoder. We ran the coder for $p \in \{0, 0.1, 0.2\}$, and for 10 different parameters p_l of perturbations of the lattice in the MDLVQ coder. For an optimal encoding, p_l should be chosen as close as possible to the actual probability of loss p.

In each case we ran the coders five times, with different seeds for the random generator to obtain different "error patterns". After the first experiments, we immediately saw that the MSE of the decoded file is highly sensitive to the error pattern. This could seem to be a major problem for the comparison of the coders, but since the same errors patterns can be applied to the four different coders, this problem can be partially eluded. Recall that the MSE is a good indication of the subjective quality because of the psycho-acoustic pre-filter.

The distortion is the same for all the BC and MDLVQ coders when $p = 0$ and $p_l = 0$. The latter is since our design criterion for the design of the original lattice was to equate the BC coder distortions (see Section IV-A). As soon as $p_l > 0$, the perturbation of the lattice degrades the quality for $p = 0$, as one would expect.

For the MDLVQ coder, the experiments show that its overall rate/distortion behavior is better than the performance of any of the BC coders. The BC coders can be seen as lying on a curve in the rate distortion plane, and the MDLVQ coder is below that curve. The MDLVQ coder outperforms the BC4 coder in both bit rate and distortion. When compared to the BC0 coder, the MDLVQ coder has a larger bit rate, since the use of MDLVQ introduces redundancy in the streams, but the decrease in distortion is significant for $p > 0$. Compared to the BC2 coder, the distortion is slightly larger, but its bit rate is significantly lower. When we take a closer look at the distortions at different values of p_l and p in the MDLVQ case, we see that the smallest distortion is achieved when $p_l = p$, as expected. But we also observed that the differences in distortions resulting from different p_l are less than those resulting from different random seeds. This data suggests that the perturbation is helpful only with high loss probability. A rule of thumb can be: use no perturbation if the loss will vary from 0 to 20%, and use $p_l = 0.1$ perturbation if the loss will vary from 10 to 30%.

B. When Only One Description is Received

As final experiments with the coders, we assumed that exactly one description is always successfully received. The rate/distortion points obtained with the three coders are illustrated in Fig. 7. Again the BC coders can be seen as lying on a curve in the rate-dostortion plane, and the MDLVQ coder lies below that curve, which means it has the best rate/distortion performance.

VIII. Subjective Comparison

To obtain a more precise verification of our results, we conducted a subjective comparison test. Because our bit rates are estimates intended for comparisons with similar schemes, but not precise absolute numbers, we compared our MDLVQ scheme with our BC0 coder with its simple insertion of zeros for lost packets. We assume an operating point where we have bit rate available, which is high enough to obtain quantization noise at the masking threshold for the MDLVQ coder, if both descriptions are received. Further we assume similar bit rates for both, the MDLVQ

and the BC0 coder. To obtain similar bit rates for the MDLVQ and BC0 coders in our comparison, the bin-width of the BC0 coder was set to $\Delta = 60$. Table I shows the resulting estimated bit rates for the 2 coders for our five test signals. The different operating conditions are loss rates of $p = 0, 0.1$, and 0.01, and packet sizes of 1024 and 4096. Table II show the resulting mean squared error for the different loss rates and coders. Recall that an MSE of 833.3 is at the masking threshold of the psycho-acoustic model. Further observe that for the 10 % loss rate, and for some of the cases for the 1% loss rate, the MDLVQ achieves a lower mean squared error than the BC0 coder. We set $p_l = 0$ (no lattice perturbation), $M = 8$ for the MDLVQ, and $M = 16$ for the BC0 coder.

We chose 5 different signals such that they cover a wide variety of signal statistics. Each has a length of 10 to 20 seconds and 32kHz sampling rate. We used the already mentioned "mixed" and "jazz" signals (jazz is named "16cj" in the results figures), and in addition "mspeech", which is German male speech, "sc03", which is music with trumpets, and "es01", which is Suzanne Vega a capella. We used a MUSHRA test [18] for our comparison, where the test subjects were presented with a series of sets of signals. Each set contains 2 coded/decoded signals, from the MDLVQ and BC coder. In addition each set contains the original signal, the original low pass filtered at 7 kHz and the original filtered at 3.5 kHz, as anchors. The subjects evaluate each signal with sliders between 0 and 100, corresponding to "bad" to "excellent". We had 7 listeners with Stax headphones in an office environment.

A. Results

The results are presented in Figs. 8 - 12. The vertical axis is the subjective grading, and the horizontal axis shows the different signals. The vertical bars show the 95% confidence intervals. They are an indication of the accuracy of our measurement, which depends on the number of listeners and how similar they graded. Within each signal are (in that order) the hidden reference, the 3.5 kHz filtered signal, the 7 kHz filtered signal, the BC0 coder (BC_1024 or BC_4096), and the MDLVQ coder (MD_1024 or MD_4096). Fig. 8 shows the results for zero packet loss. For the Multiple Description coders this means that both descriptions are received. It can be seen that in this case the BC0 (BC_1024) and MDLVQ (MD_1024) coder have indeed the same quality, as expected. There is no significant difference in their quality because their confidence intervals overlap. The further figures show that the MDLVQ coder is evaluated significantly better than the BC0 coder (their confidence intervals for "all items" don't overlap) for the case of packet losses, except for the case of packet length 4096 and 1% packet loss (Fig. 11), where the confidence intervals overlap. For 10% packet loss (Fig. 10 and Fig. 12) it can be seen that the MD coder is rated as good as the 7 kHz bandlimited original (hidden_reference7), and the BC0 coder as good as the 3.5 kHz bandlimited original (hidden_reference7). The difference for the 1% case with packet size 1024 (Fig. 9) has a similar magnitude.

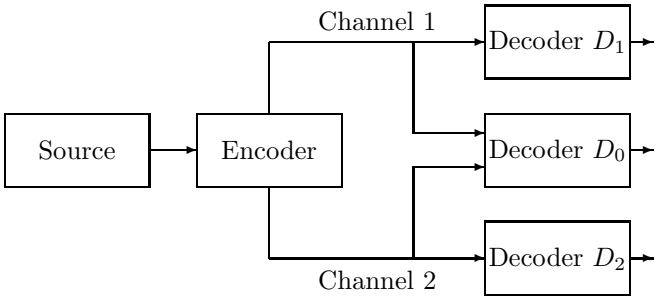


Fig. 2. Multiple description system with two channels.

Signal	MDLVQ	BC0
mspeech	2.79	2.80
sc03	2.83	2.73
es01	2.89	2.83
mixed	2.90	2.94
jazz	3.13	3.01

TABLE I

The bit rates (in bit/sample) for the test signals for the MDLVQ and BC0 coder.

This means there is a clearly audible advantage for the MD coder.

IX. Conclusions

Many of the newer multiple description techniques are designed for minimizing the mean squared error of the reconstructed signal. The psycho-acoustically controlled prefilter is used as a low delay conversion of audible difference into a mean squared distance. This makes it possible to apply these multiple description techniques to audio coding, with a distance measure suitable for audio. Comparisons show that a better rate-distortion operating point is achieved than with a coder with no added redundancy

		4096		1024	
Signal	$p = 0$	$p = .01$	$p = .1$	$p = .01$	$p = .1$
mspeech	824	1096	2853	1085	2428
sc03	794	1004	2916	1057	2639
es01	786	1012	3345	1074	2588
mixed	832	1081	3178	1136	3080
jazz	830	1087	3320	1142	3148
mspeech	302	853	4881	785	4079
sc03	327	910	5441	931	4814
es01	331	515	5020	1124	4546
mixed	303	997	5432	1251	5915
jazz	301	1207	6640	1390	6865

TABLE II

The resulting mean squared error for the different loss rates for packet sizes of 4096 and 1024 for the MDLVQ coder (above) and the BC0 coder (below).

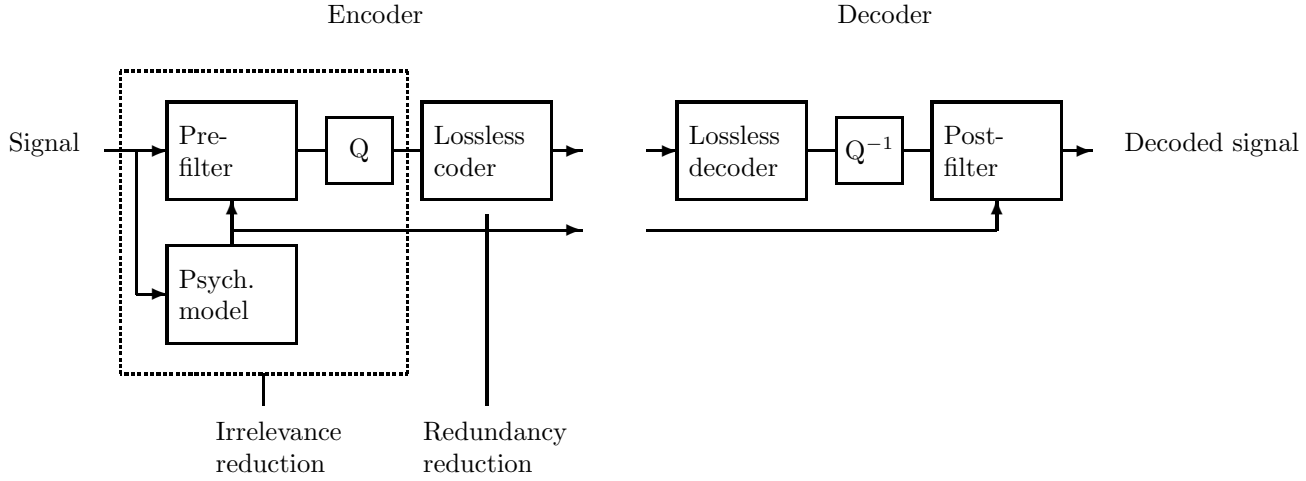


Fig. 1. An audio coding scheme with separated irrelevance and redundancy reduction, using a psycho-acoustic pre- and post-filter and lossless compression.

(coder BC0), or a coder with the lower half band repeated over two descriptions (coder BC2), or a coder with the lower quarter band repeated over two descriptions (coder BC4). Subjective comparisons of the MD and BC0 coder show, that the MD coder has a significantly better quality than the BC0 coder. Since both the psycho-acoustic pre-filter and the multiple description scheme add only very little delay, an overall delay of the multi-descriptive audio encoder/decoder of 10 ms can be obtained.

Together with the low complexity of the MD scheme, it makes this approach attractive for applications like wireless microphones, wireless speakers, or video conferencing.

References

[1] V. K Goyal. Multiple description coding: Compression meets the network. *IEEE Sig. Proc. Mag.*, 18(5):74–93, September 2001.

[2] G. Schuller and A. Harma. Low delay audio compression using predictive coding. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, Orlando, FL, 2002.

[3] K. Nitadori. Linear transform coding and predictive coding. *IECE Japan*, February 1970.

[4] N. S. Jayant and P. Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

[5] B. Edler and G. Schuller. Audio coding using a psychoacoustic pre- and post-filter. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume II, pages 881–884, Istanbul, Turkey, 2000.

[6] Gerald Schuller, Bin Yu, Dawei Huang, and Bernd Edler. Perceptual audio coding using adaptive pre- and post-filters and lossless compression. *IEEE Trans. on Speech and Audio Proc.*, 2002. To appear.

[7] Sean Dorward, Dawei Huang, Serap Savari, Gerald Schuller, and Bin Yu. Low delay perceptual lossless coding of audio signals. In *Proc. IEEE Data Compression Conf.*, pages 312–320, Snowbird, Utah, March 2001.

[8] V. A. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Trans. Inform. Th.*, 39(3):821–834, May 1993.

[9] V. K Goyal, J. A. Kelner, and J. Kovačević. Multiple description vector quantization with a coarse lattice. *IEEE Trans. Inform. Th.*, 48(3):781–788, March 2002.

[10] V. A. Vaishampayan, N. J. A. Sloane, and S. D. Servetto. Multiple-description vector quantization with lattice codebooks: Design and analysis. *IEEE Trans. Inform. Th.*, 47(5):1718–1734, July 2001.

[11] J. H. Conway and N. J. A. Sloane. Fast quantizing and decoding algorithms for lattice quantizers and codes. *IEEE Trans. Inform. Th.*, IT-28(2):227–232, March 1982.

[12] S. D. Servetto, V. A. Vaishampayan, and N. J. A. Sloane. Multiple description lattice vector quantization. In *Proc. IEEE Data Compression Conf.*, Snowbird, UT, March 1999.

[13] J. A. Kelner, V. K Goyal, and J. Kovačević. Multiple description lattice vector quantization: Variations and extensions. In *Proc. IEEE Data Compression Conf.*, pages 480–489, Snowbird, Utah, March 2000.

[14] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, 1992.

[15] B. Widrow and M. E. Hoff Jr. Adaptive switching circuits. In *IRE WESCON*, pages 96–104, 1960.

[16] Simon Haykin. *Adaptive Filter Theory*. Prentice Hall, Englewood Cliffs, NJ 07632, 1999.

[17] V. K Goyal and J. Kovačević. Generalized multiple description coding with correlating transforms. *IEEE Trans. Inform. Th.*, 47(6):2199–2224, September 2001.

[18] Method for the subjective assessment of intermediate quality levels of coding systems. (Recommendation ITU-R BS.1534-1 (01/2003)).

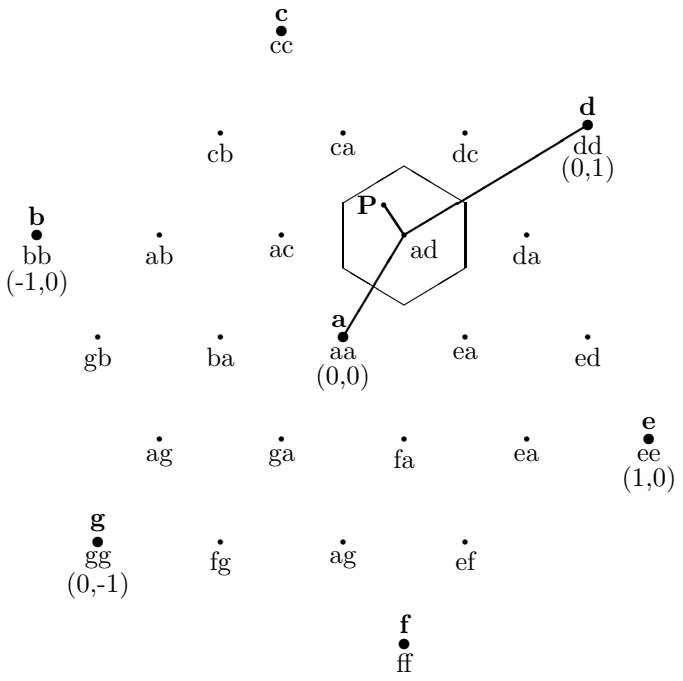


Fig. 3. Example of MDLVQ of a pair of samples. Each fine lattice point is labeled according to the SVS algorithm. The source vector **P** is quantized to the closest lattice point **ad**; therefore, its descriptions are **a** and **d**. Illustration of the coordinate indexing of the sublattice points.

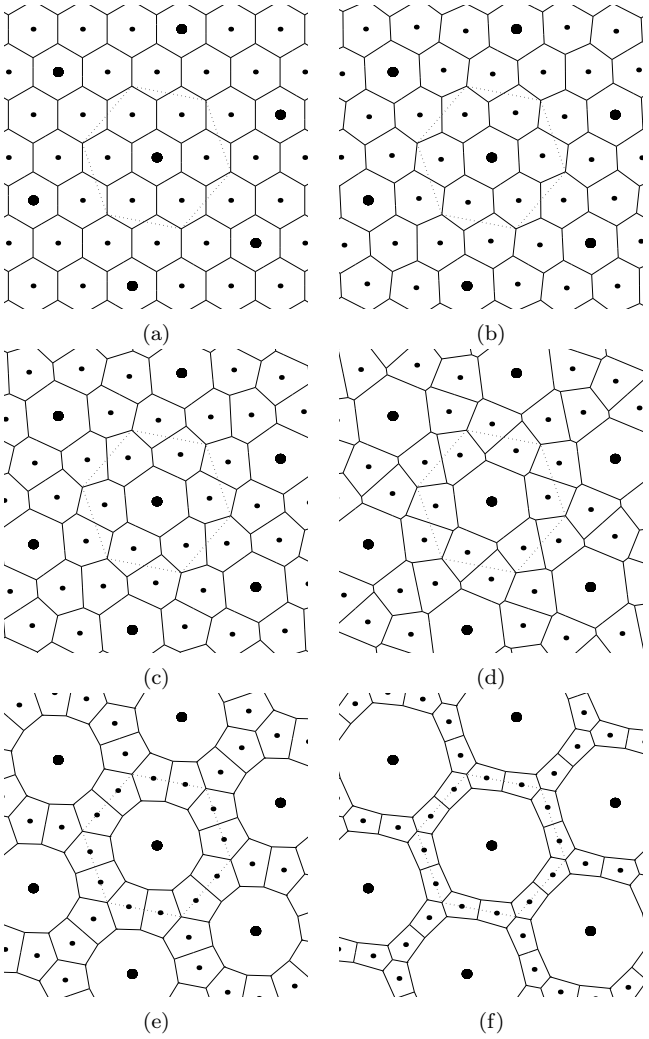


Fig. 4. Shapes of the Voronoi cells, with respect to the MD objective function (1), for different design parameters p: (a) 0, (b) 0.02, (c) 0.05, (d) 0.1, (e) 0.2, (f) 0.5.

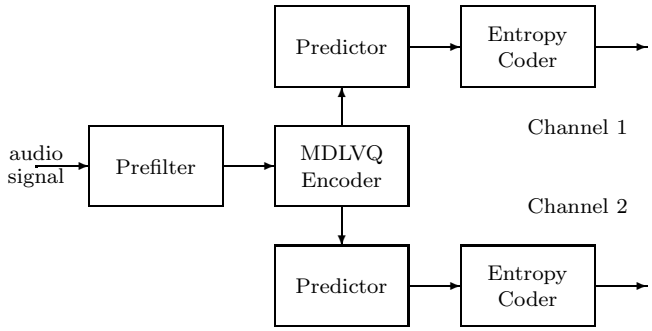


Fig. 5. Block diagram of the MD encoder.

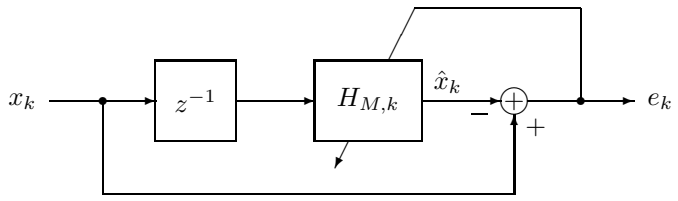
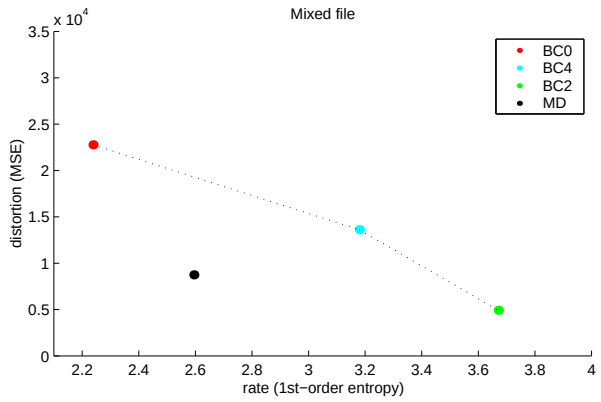


Fig. 6. Adaptive prediction filter.



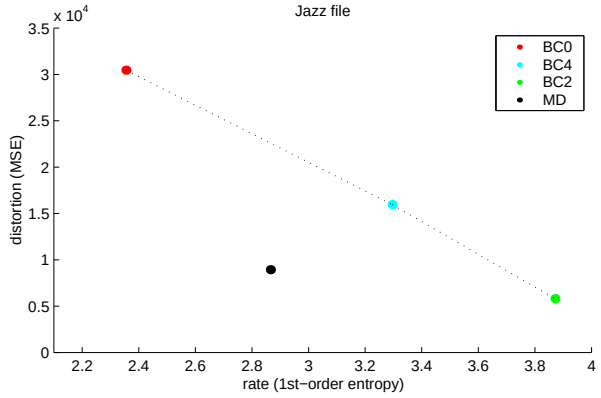


Fig. 7. Rate–distortion performances of the MDLVQ (MD), BC0, BC2 and BC4 coders when one and only one description is successfully transmitted.

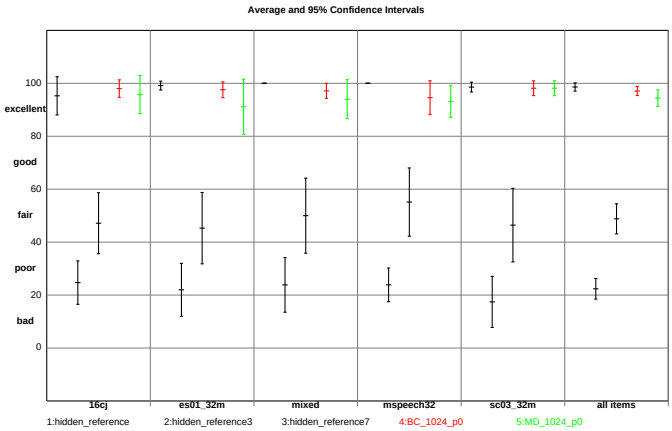


Fig. 8. Listening comparison, no packet loss.

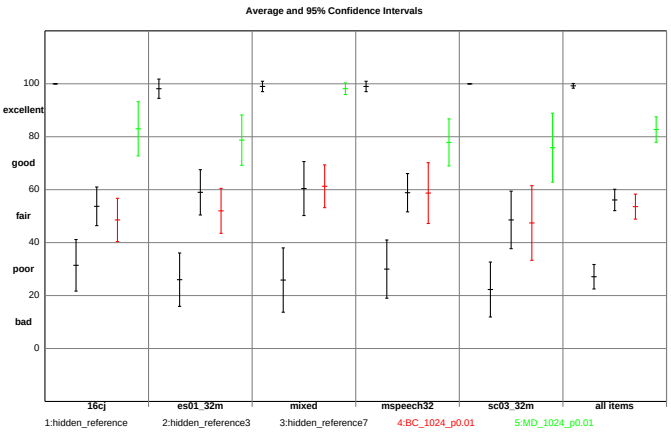


Fig. 9. Listening comparison, 1% packet loss, packet size 1024.

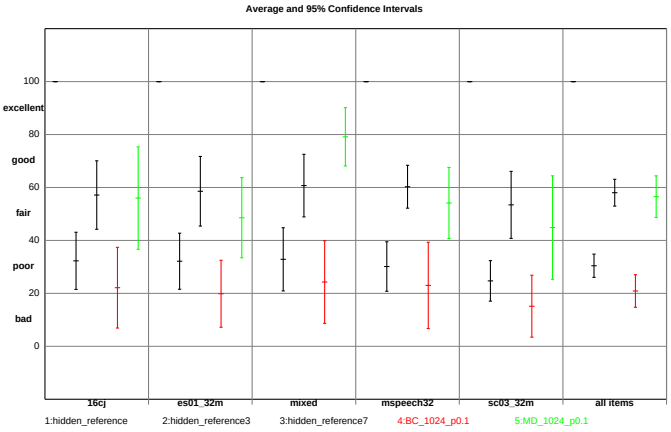


Fig. 10. Listening comparison, 10% packet loss, packet size 1024.

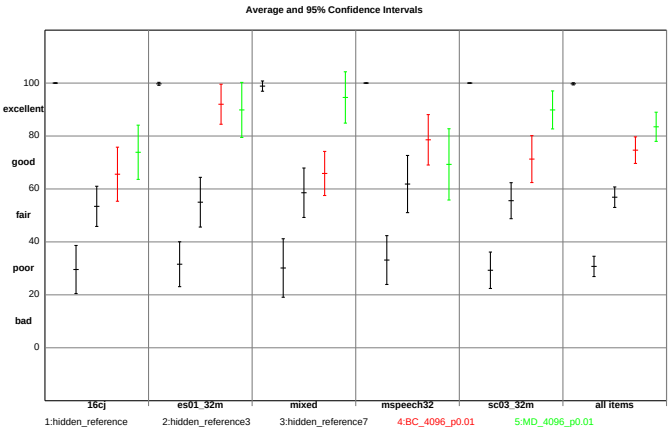


Fig. 11. Listening comparison, 1% packet loss, packet size 4096.

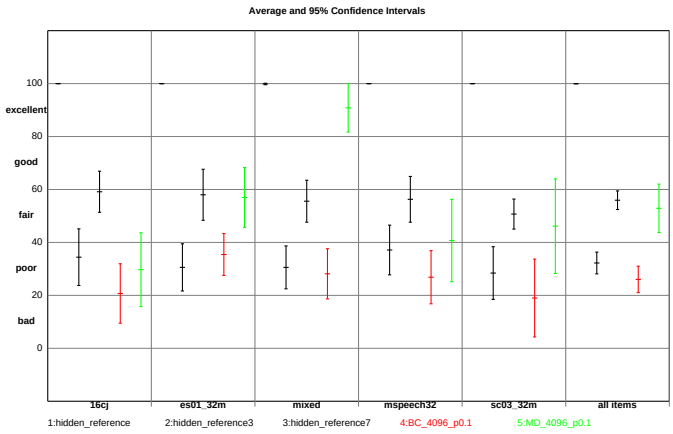


Fig. 12. Listening comparison, 10% packet loss, packet size 4096.