ERASURE RECOVERY IN PREDICTIVE CODING ENVIRONMENTS USING MULTIPLE DESCRIPTION CODING

Raghavendra Singh, Antonio Ortega
Integrated Media Systems Center
Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA 900089 {rasingh,ortega}@sipi.usc.edu

Abstract - We propose an algorithm for erasure recovery in predictive coding schemes, where erasures can cause catastrophic error propagation. The recovery algorithm is based on sending multiple descriptions of the source and using a deterministic distance measure to find the most likely estimate for the lost data, given the received data and the side information. Results show that we can recover from short burst erasures and that for long bursts (more than 10% of the samples are lost) we can recover to within 0.4dB of the original DPCM performance.

INTRODUCTION

In recent years the volume of multimedia data transmitted over such "best-effort" networks as the Internet has continued to increase while, due to congestion, routing delay and network heterogeneity, packet losses and delays continue to be commonplace. In this paper we propose techniques for local recovery of erasures that are specifically designed for multimedia data. In particular, we tackle one of the key obstacles in erasure recovery for compressed video or audio, namely, the fact that predictive compression schemes are typically used (e.g., motion prediction in video coding, DPCM in audio coding). Predictive coding schemes take advantage of correlations in the source to achieve better performance than approaches, such as PCM, that treat a source as a set of independent samples [2]. However, the main drawback of these predictive schemes is that a single erasure causes decoding errors to propagate through all the samples following the erasure. In contrast, PCM schemes are more robust, since losses do not propagate, but have a much lower compression performance. A traditional approach to prevent error propagation in predictive coders is to restart the prediction loop by periodically inserting PCM-coded samples. The drawback of this approach is that it limits the length of the error propagation but it does not allow recovery of lost data.

In this paper, we propose a novel technique for erasure recovery in DPCM based on Multiple Description Coding (MDC) [1]. In MDC schemes, two or more descriptions of the source are sent to the receiver over different channels (see Fig. 1). If only channel S_1 (or S_2) is received the signal can be reconstructed with distortion D_{s_1} (or D_{s_2}.) If both channels are received, information from the two channels is combined to achieve a lower distortion reproduction D_c (i.e., $D_c \leq D_{s_1}$, $D_c < D_{s_2}$.)

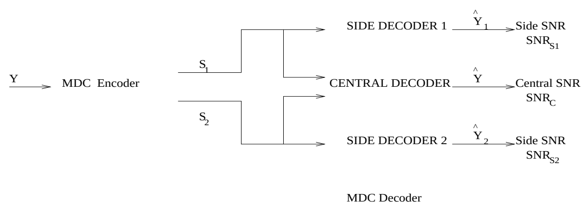

Figure 1: Multiple Description coding and decoding

MDC is particularly suited for scenarios such as those considered here because (i) the network does not provide transmission at different priorities and thus multiresolution techniques would not be useful, and (ii) local recovery is preferable to retransmission. These features help explain the recent revival of interest in MDC, which has led to the proposal of numerous practical MDC systems e.g. [4], [3]. Most of the recently proposed techniques are designed for memoryless coding environments and use Balanced Multiple Description Coding (BMDC), where both descriptions are coded at the same rate. One exception is the recent work of [5] where multiple description DPCM is proposed. Here, each of the channels is coded using a DPCM loop, and if both channels are received correctly, a better reproduction is possible, (i.e. for $D_c < D_{s_1}$ and $D_c < D_{s_1}$). However, due to the lack of robustness to error of DPCM it is assumed that if a channel suffers one erasure it will have to be completely discarded.

Instead, in our proposed approach, while also employing DPCM in each channel, we show how it is possible to approximate the lost data through processing at the decoder. Our algorithm is based on maximum likelihood estimation of the erased samples (from, say, S_1), where likelihood is defined in terms of a distance measure between S_1 and S_2 with the added constraint that the reconstructed S_1 samples be consistent with all the error-free data that has been received.

Another novelty in our work is that we choose an Unbalanced Multiple Description Coding (UMDC) framework, i.e. $D_{s_1} > D_{s_2}$. In BMDC the highest resolution reproduction was obtained when both channels were received, while in UMDC the highest resolution is obtained when S_1 is received. S_2 is coded at low resolution and used as explicit redundancy to correct S_1. In keeping with the MDC philosophy, S_2 is independently decodable and is of a quality acceptable to the receiver in case erasures in S_1 cannot be recovered. We compare BMDC and UMDC environments (for same total rate) and show that for long erasures in S_1, UMDC outperforms BMDC.

ERASURE RECOVERY ALGORITHM

We develop our algorithm using the UMDC case but, this can be easily extended to the BMDC case. In an UMDC environment one of the descriptions of the input, Y, is at high bit-rate and the other at low bit-rate. Let HR and LR be the reconstructed sequences at high and low resolutions, respectively, with X and x, denoting their respective prediction errors. Quantized variables are denoted with a hat and when subscripts are used, they denote specific samples, e.g., $\hat{X}$ is the quantized prediction error X and HR_i is the ith sample of HR. Also let C^{lr} and C^{hr} represent the codebook partitions of Q_{lr} and Q_{hr} respectively.

Assume that sample $\hat{X}_e$ of the high resolution description is lost at the decoder while the low resolution $\hat{x}$ is received error-free. Our goal is to estimate the lost sample by taking into account the information that was received. A key tool used in our algorithm is the verification of *consistency* of the estimate with LR where *Consistency* in simplified terms is defined as: a specific $\hat{X}_i$ is consistent with $\hat{x}_i$, if there exists an input y_i such that $\hat{X}_i = Q_1(y_i)$ implies that $Q_2(y_i) = \hat{x}_i$, where Q_1 and Q_2 are the low resolution and high resolution DPCM loops. Thus LR along with memory of the source helps in recovering erasures in our look ahead scheme, e.g. a "good" local estimate might invalidate consistency in the future.

The algorithm works through a 3 step process. (i) *Candidate Selection*: Of all the possible quantized values for $\hat{X}_e$ only those that are *consistent* with $\hat{x}_e$ are considered as candidates. (ii) *Path Consistency Check*: For each of the above candidates the high resolution description is decoded giving a different sequence of outputs for each candidate. Among these sequences, those that are consistent with the LR sequence are chosen. (iii) The consistent sequence closest to LR , in Euclidean distance, is chosen as the recovered HR. We now describe our algorithm more formally.

In the DPCM encoder, with the predictor coefficient α, for any sample i,

$$X_i = Y_i - \alpha HR_{i-1},\ x_i = Y_i - \alpha LR_{i-1} \Rightarrow X_i = x_i + \alpha(LR_{i-1} - HR_{i-1}) \quad (1)$$

At the decoder, let $\epsilon = LR_{i-1} - HR_{i-1}$, and given $\hat{x}_i = j$, i.e. $x_i \in {C_j}^{lr} \doteq [a_j, b_j]$, an interval R in which X_i has to lie can be found:

$$X_i \in R \doteq [a_j + \alpha\epsilon, b_j + \alpha\epsilon] \quad (2)$$

On the other hand if $\hat{X}_i = j$ is given, i.e. $X_i \in {C_j}^{hr} \doteq [A_j, B_j]$ then $x_i \in r$ where r is defined below:

$$x_i \in r \doteq [A_j - \alpha\epsilon, B_j - \alpha\epsilon] \quad (3)$$

For *Candidate Selection* in the first step of our algorithm, given HR_{e-1}, LR_{e-1}, $\hat{x}_e$ we use (2) to define the interval R. All the bins of Q_{hr} that intersect with R are candidates for $\hat{X}_e$. An example is shown in the Figure 2.

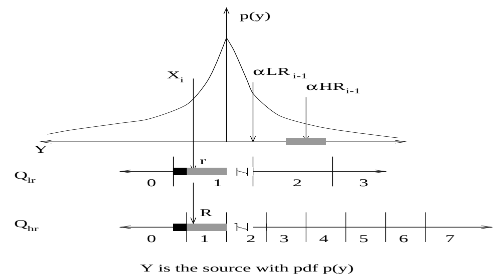

Figure 2: Quantization in DPCM is equivalent to using a scalar quantizer with its center shifted to the value of predictor. In this example Q_{hr} & Q_{hr} are shifted by αHR_{e-1} & LR_{e-1} respectively. If $\hat{x}_e = 1$ was received then the *candidates* for $\hat{X}_e$ are 0,1 or 2.

We further define *Path Consistency Check* as: Given a sequence of $\hat{X}_i$, reconstructed sequence HR_i is consistent with LR, if at each sample $i > e$, the interval r from (3) overlaps with the the quantization bin of $\hat{x}_i$.

To explain our algorithm we use Figure 3. Here, $X_0 = 1$ is lost but we assume that x_0, HR_{-1}, LR_{-1} and $X_i, x_i\ \forall i > 0$ have been received correctly. From *Candidate Selection*, we have 3 possible candidates for the erased sample, i.e X_0 could be 0,1 or 2. Decoding each of these 3 choices of X_0, N samples into the future, leads to three candidate paths for the HR description, shown by the dark lines in the figure.

Next, we apply our *Path Consistency Check* to each of the candidate paths. $HR[j]$ represents the decoded path given $X_0 = j$. In the figure we see that the $HR[2]$ is not consistent with LR. At sample 2, r (patterned box), defined by $HR[2]_1, LR_2, \hat{X}_2$, does not overlap with the quantization bin of $\hat{x}_2$(black box). Among the two consistent paths $HR[1]\&HR[0]$, the one closest to LR stream is the recovered HR.

Thus, the reconstructed output of our algorithm is consistent with all the data received and closest to the correctly received description. The algorithm is formally given below, where we are assuming that erasures occur in X from index e_b to e_e. HR_{e_b-1} is decoded and LR, x are known.

Step 1: Generate all candidates

For $j = e_b{:}e_e$

Find X_j, *given* HR_{j-1}, LR_{j-1}, x_j *using Candidate Selection.*

Decode $HR_j = \alpha\ HR_{j-1} + X_j$ *for each of above* X_j

end

Decode all candidates for N *future samples.*

Step 2: Eliminate all candidate paths that are not consistent with LR using *Path Consistency Check*.

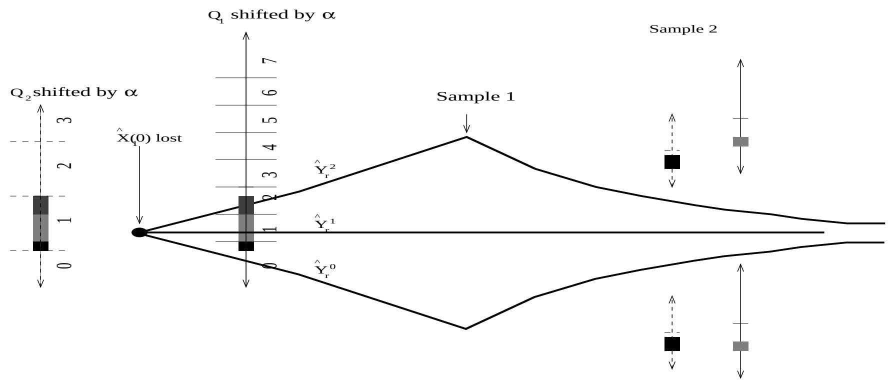

Figure 3: Erasure Recovery Algorithm. X_0 is erased. HR_{-1}, LR_{-1}, x_0 give 3 possible candidate for $X_0 \in [0, 1, 2]$. Next consistency check is done for the decoded paths, at sample 2 the top path is inconsistent, the black box $(\hat{x}_2)$ does not overlap with patterned box (r). Then of the two remaining paths we choose the one which is closest in MSE to LR

Step 3: Pick the candidate that is closest in MSE to LR

If both channels are received correctly then as shown in figure 2 the codebook partitions of Q_{hr} are further subdivided. These new partitions can be used to form a new codebook which would have larger number of bins over the same range thus necessarily improving the SNR performance of the quantizer. Even though we can improve performance by combining Q_{hr} and Q_{lr}, the original quantizers Q_{hr} or Q_{lr} should be used in the prediction loops, otherwise encoder and decoder would not be synchronized.

RESULTS

In our experiments we have found that N increases with the correlation in the source. For correlation of about 0.9, looking ahead 20 samples suffices. Also for long burst of erasures pruning is needed as the number of candidates grows exponentially. Right now we keep only the candidates which are closet to the second description at the pruning point. Using interleaving, the need to consider long burst of erasure can be avoided.

The results in Figure 4 are for UMDC with 3 bits and 1 bit channel. Our algorithm recovers nearly perfectly from single erasures, this is important because in a DPCM loop even single erasures are catastrophic. We show that for a burst erasure of 100 samples, (10 %of the samples) we would be doing better than the BMDC reported in [5]. The other interesting result is that we are gaining 0.6 db when both channels are received by using the simple algorithm given in previous section.

In the right plot of Figure 4 we have a BMDC system with bit rate 2 in each channel. We did not use the index assignment of [5], instead we used two quantizers shifted relative to each other and we get a gain of about 2.5 dB when both the channels are received. We show that if there are erasures

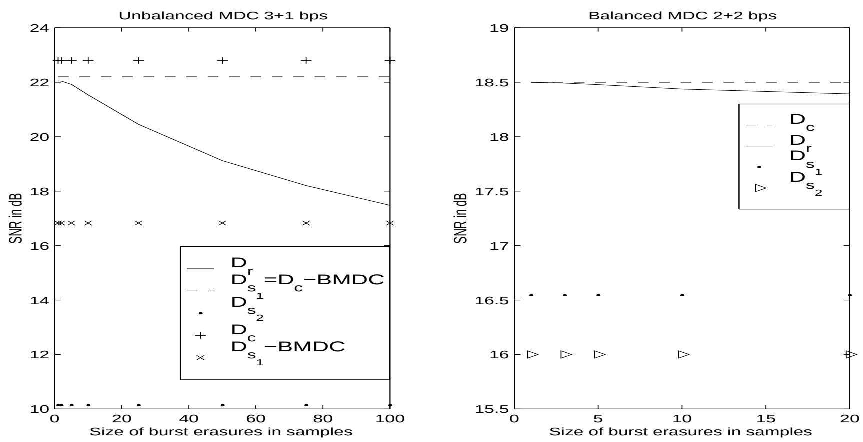

Figure 4: Results for our algorithm for both Unbalanced (3+1 bps) and Balanced (2+2 bps) case. D_r is the SNR after recovery, D_{s_1} is the side channel 1 SNR, D_c is Central Distortion, D-BMDC are results from [5]. All results for Gauss Markov Source, $\rho = 0.9$, results are averaged over 100 runs of 1000 samples each. Uniform Threshold Quantizers are used with entropy coding.

in S_1 we don't need to discard it, we can recover to within 0.1 dB of D_c for erasures of length 20 samples. In addition we did an experiment with interleaved packets where a packet is of the form: $\boxed{X_{i+1}, X_{i+2}, X_{i+3}, X_{i+4}, X_{i+5}, X_{i+21} X_{i+22}, X_{i+23}, X_{i+24}, X_{i+25}, X_{i+41} \ldots}$. A packet lost meant that 125 samples were lost for a 1000 sample stream. We recovered to within 0.4 dB of the original SNR.

The results show that for both BMDC and UMDC we can use our algorithm to recover erasures. In BMDC this allows the decoder to decode around D_c, in UMDC we can have a large burst of erasures before we will be doing worse than a same bit rate BMDC.

References

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

[2] N.S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice Hall 1984.

[3] W. Jiang and A. Ortega, "Multiple Description Coding via Polyphase Transform and Selective Quantization", in *Proc. of VCIP*, 1999.

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

[5] V.A. Vaishampayan and S. John, "Interframe Balanced Multiple Description Video Compression", in *Proc. of PVW*,1999.