DCT-based scalable video coding with drift

Amy R. Reibman, Léon Bottou and Andrea Basso
AT&T Labs – Research
{amy,leonb,basso}@research.att.com

Abstract

Scalable video coders have traditionally avoided using enhancement-layer (EL) information to predict the base layer (BL), so as to avoid so-called "drift". As a result, they are less efficient than a one-layer coder. Fine Granularity Scalable (FGS) coders avoid using EL information to predict the EL as well, suffering even further inefficiencies. In this paper, we explore a scalable video coder that allows drift, by predicting the BL from EL information. However, we show that through careful management of the amount of drift introduced, the video quality at low rates is only marginally worse than the drift-free case, while the overall compression efficiency is not much worse than a one-layer encoder.

1 Introduction

Compressed video, which uses predictive coding algorithms and variable-length coding, is sensitive to network impairments since these can cause error propagation. A single bit error or erasure can cause substantial degradation if no action is taken to stop or limit the extent of error propagation. Motion compensation allows the error to propagate both temporally and spatially. Because of this, there has been extensive effort in the video community to design new techniques that limit the extent of error propagation [1]. However, almost all attempts to limit error propagation decrease the coding efficiency, some dramatically so. To ensure the best operation of the video coder in an error-prone channel, the balance between resilience and efficiency must be managed carefully.

Scalable coding algorithms create a partitioning of the compressed bitstream into more and less important parts. This allows a natural combination with different mechanisms to prioritize network transport, for example, marking less important parts for early discard [2], applying unequal error protection [3], or facilitating rate matching between encoder and network [4]. When used in conjunction with such techniques, scalable video can be very resilient to network-introduced errors.

Early scalable video coders (like MPEG-2 SNR scalability (SNRS) [2]) allowed drift (the propagation of enhancement-layer (EL) errors into the base-layer (BL) reconstruction) by using low-priority EL information to predict the high-priority BL. However, in recent years, the development of scalable video encoders (like H.263 SNRS and spatial scalability (SS) [1], and [5, 6, 7]) has focused on eliminating drift. In these algorithms, the BL is predicted only from BL. This strategy has been taken one step further in the development of MPEG-4 Fine Granularity Scalability (FGS) [4], in which the EL is also predicted only from BL information.

However, while recent scalable video coding algorithms are becoming more efficient at compressing the video, they lose compression efficiency because they ignore all EL information when predicting the BL. In particular, experiments show that with MPEG-2 SS, MPEG-4 and H.263 scalability modes all suffer from 0.5-1.5 dB losses for every layer [2, 8]. FGS has particularly poor compression inefficiency [9] because of its restricted prediction strategy.

Despite the predominance of arguments in the literature maintaining that systems should be designed not to allow drift, there is some evidence that drift need not be eliminated completely [2, 10]. Therefore, in [11] we introduced a DCT-based motion-compensated scalable video coder in which the BL can be predicted from past EL information, while the resultant drift is controlled. We showed that this encoder, with a simple heuristic decision algorithm, significantly outperforms both the FGS encoder and the one-layer encoder

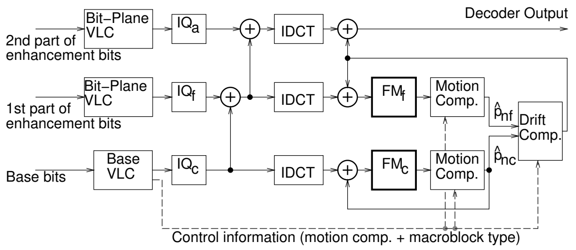

Figure 1: Two-loop decoder with drift control

across the range of channel rates. It has better compression efficiency than FGS for higher bit-rates, with only slightly degraded resilience for the lower bit-rates.

In this paper, we focus on improving the decision algorithm for the DCT scalable coder with drift. Section 2 presents the general scalable coder with drift control. In section 3 we present optimization strategies for the scalable video encoder. Section 4 demonstrates that our coder performs better than some alternative encoders across most of the range of channel bit-rates, even though our encoder suffers marginal performance degradation at the lowest bit-rates compared to the no-drift encoder alternative.

2 Drift–controlled coder

The scalable DCT decoder with drift control shown in Figure 1 takes three levels of input. The base bits, with bit rate R_{nc}, are assumed to be always available. The first part of the enhancement bits, with bit-rate $R_{nf} - R_{nc}$, may not be received by the decoder, but if received, are used to predict the next frame. The second part of the enhancement bits, with bit-rate $R_{na} - R_{nf}$, may not be received, and are never used to predict the next frame.

Both the decoder and the encoder maintain two frame memories. The *coarse frame memory* depends only on the base bits and never drifts. The *fine frame memory* is updated by first combining both motion-compensated frame memories, and then applying the base bits and the first part of the enhancement bits. The fine memory drifts when some of these enhancement bits are lost.

Let $\hat{p}_{nc}$ and $\hat{p}_{nf}$ be motion-compensated predictions from the coarse and fine memories for macroblock n. For each macroblock (MB), the drift compensation box in Figure 1 combines the coarse and fine predictions according to a MB type information. The first option eliminates drift by taking the coarse prediction $\hat{p}_{nc}$ only (as in FGS). The second option allows drift by taking the fine prediction $\hat{p}_{nf}$ only (as in MPEG-2 SNRS). The third option reduces – but does not eliminate – drift by averaging both predictions $(\hat{p}_{nc} + \hat{p}_{nf})/2$. For simplicity, we only consider here these three options.

The scalable DCT encoder (Figure 2) tracks both frame memories under the assumption that all bits are received by the decoder. The encoder makes several decisions that affect the amount of decoder drift in the fine memory. The first decision is the selection of a prediction mode for the drift compensation. The second decision involves the number of bit-planes that might be used in the prediction loop; this is accomplished by adjusting the quantization Q_f relative to the final quantization Q_a. Different images have different trade-offs between efficiency and resilience as a function of these drift control decisions. The encoder must make these decisions and send this information to the decoder. The encoder makes these choices on a MB basis with the goal of optimizing the total system performance as described in section 3.

To minimize the influence of drift in general, we use an embedded coder [11] to compress each individual frame. This allows more significant EL bit-planes to be received and decoded even if the network does not have sufficient bandwidth to send the entire EL. The BL VLC also relies on arithmetic bit-plane coding, but could instead be implemented using the usual Huffman method. Macroblock type information and motion vectors are included in the BL. We use the same motion vectors in both the BL and EL.

3 Encoder optimization

The traditional (often implicit) optimization when designing a scalable coder is to minimize the maximum possible distortion at the decoder, subject to the constraint that the channel rate R is in the range $\bar{R}_c \leq R \leq \bar{R}_a$. Typically, both $\bar{R}_c$ and $\bar{R}_a$ are known, although neither the instantaneous channel rate nor the average channel rate in some time interval is known. The maximum distortion is achieved for the minimum rate $\bar{R}_c$. Thus, op-

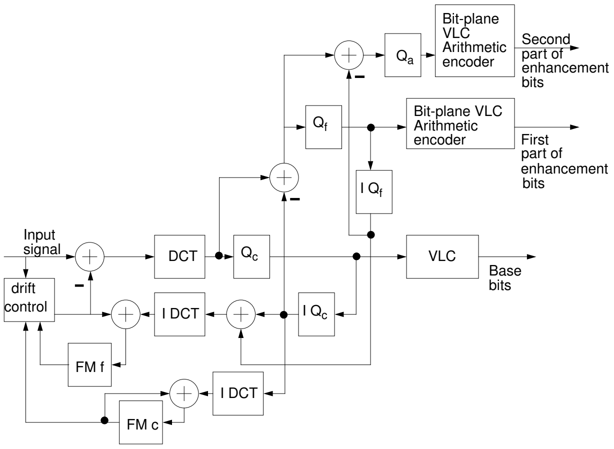

Figure 2: Two-loop encoder with drift control

timizing using this criterion leads to a prediction process that does not tolerate any drift in the BL. However, this also results in larger than necessary distortions at rates near $\bar{R}_a$.

In [11] we fixed the quantizers Q_a, Q_f, and Q_c and used a simple heuristic to choose which prediction should be used for each MB. Here, we consider changing both the quantizers and the prediction, using an optimal approach. It can be shown that, assuming the previous quantizers are fixed, the optimal choice for the prediction can be determined by choosing the prediction that minimizes $(1+\gamma)\log(\sigma_p^2) + \gamma \log(1 + 12M_p/Q_c^2)$, where p indicates the prediction being evaluated and M_p is the amount of mismatch between that prediction and $\hat{p}_{nc}$. Parameter γ indicates the relative importance of the rate-distortion performance when all bits are received and when only the coarse bits are received.

The choice above clearly depends on the current coarse quantizer. However, implicit in the choice is the dependence on the coarse and fine quantizers in the previous frame. For example, a finer fine quantizer for the four MBs in the previous frame used to create $\hat{p}_f$ will reduce σ_p^2 but increase M_p. Thus, joint optimization is required to choose the best quantizers in the previous frame and the prediction type in the current frame. Unfortunately, a fully joint optimization would require a prohibitive search space.

Hence, we propose to decouple the optimization by considering first the choice of the best quantizer for each MB in the previous frame assuming the other quantizers are identical, followed by the choice of the best prediction assuming those quantizers. For the first step, for each MB in frame $t-1$ we determine the affected MB's in frame t via reverse motion-compensation. For each of the possible quantizer values, we assume the surrounding MBs use the same quantizer, and we determine the best prediction of those affected MBs. Then we choose the coarse-fine quantizer pair that minimizes the weighted cost of the predictions in the affected MBs. Once all quantizers from the previous frame are decided, the choice of the best prediction follows as before.

4 Results

We compare the performance of our drift-controlled coder to a one-loop encoder with no drift control, and to an FGS encoder [4]. All three encoders are implemented using core components from H.263, with modifications to obtain the relevant prediction-loop structure. All three coders also have the following additional modifications: the H.263 quantizer is replaced by a scalable quantizer [6], and the bitstream encoder is replaced by an embedded DCT coder [11]. The embedded coder uses a binary adaptive Z-coder [12] associated with each of the BL, first part and second part of the EL information, to create efficient bit-plane encodings. The probability distributions associated with the Z-coder are learned through context variables, which are reset after every frame.

We use each encoder to create a single encoding, containing the BL and 3 bit-planes of EL information. Each coder uses a fixed identical quantizer in the BL. In the current implementation, our drift-controlled coder sets $Q_a = 4, Q_f = 2Q_a$, and $Q_c = 4Q_f$. The choice of prediction is as described in the previous section.

To obtain the performance comparisons in Figures 3 and 4 for the sequences *Hall monitor* and *Foreman*, respectively, we successively discard EL bit-planes and decode the remainder. The x-axis shows the decoded bit-rate, and the y-axis shows the PSNR of the resulting decoder reconstruction as bit-planes are discarded. Also shown is the performance of the one-loop encoder with no loss (solid line). This provides an upper bound on the performance of the scalable coders. To emphasize the impact of drift, we use one I frame, followed by continuous P frames for each coder. The three curves labeled "proposed" use $\gamma = 1, 2, 3$. Best performance for large and small rates are achieved

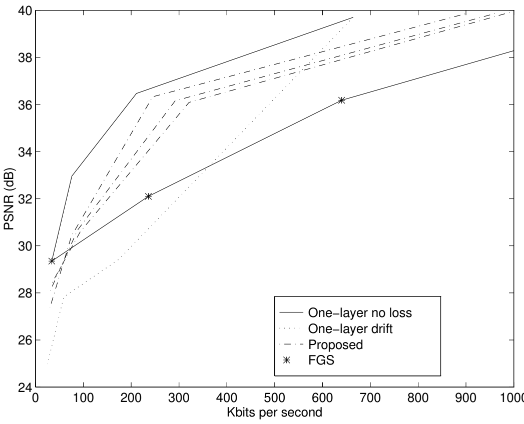

Figure 3: PSNR vs. rate for sequence *Hall*.

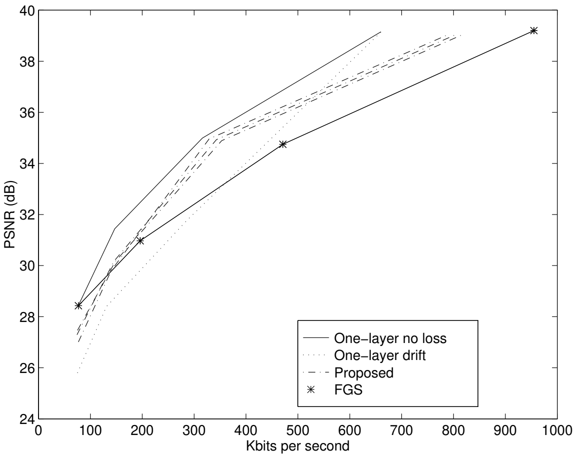

Figure 4: PSNR vs. rate for sequence *Foreman*.

by $\gamma = 1$ and $\gamma = 3$, respectively.

The FGS coder performs poorly at the higher rates, especially for the mostly-still Hall sequence. The one-layer decoder with drift suffers a 2.6-4.3 dB degradation at the lowest bit-rate, compared to the drift-free FGS decoder. Relative to the FGS coder, our drift-controlled coder with $\gamma = 2$ suffers about 1.3-1.4 dB performance degradation at the lowest bit-rate, but significantly outperforms it elsewhere. Our coder loses some efficiency at the highest rates compared to the one-layer coder, but has noticeably less drift as bit-planes are discarded.

Table 1 shows the PSNR averaged across channel rates, assuming a uniform distribution of rates between the smallest and the largest rate of the one-loop encoder. Although our implementation currently uses a simple optimization strategy, it still outperforms the other coders across the range of channel rates by 0.7-2.1 dB when there is only one I-frame.

seq.	One-loop	$\gamma = 2$	FGS	bound
Hall	33.14	35.19	33.11	36.77
Fore	33.22	34.09	33.08	35.03

Table 1: PSNR averaged across channel assuming uniform distribution.

References

[1] M.-T. Sun and A. R. Reibman, ed., *Compressed Video over Networks*, Marcel Dekker, Inc, New York, NY, 2001.

[2] R. Aravind et al. "Packet loss resilience of MPEG-2 scalable video coding algorithms", *IEEE Trans. on Ckts. and Syst. for Video Tech.*, vol. 6, no. 5, pp. 426-435, October 1996.

[3] P. A. Chou et al. "FEC and pseudo-ARQ for receiver-driven layered multicast of audio and video", *IEEE Data Compression Conf.*, pp. 440-449, Mar. 2000.

[4] H. Radha et al. "Fine-granular-scalable video for packet networks", *Packet Video Workshop*, New York NY, April 1999.

[5] J. F. Arnold et al. "Efficient drift-free Signal-to-Noise Ratio scalability", *IEEE Trans. Ckts. and Syst. for Video Tech.*, vol 10, no. 1, pp. 70-82, Feb. 2000.

[6] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video", *IEEE Trans. Image Proc.*, vol. 3, no. 5, pp. 572-588, Sept. 1994.

[7] M. Domanski et al. "Spatio-temporal scalability for MPEG video coding", *IEEE Trans. Ckts. and Syst. for Video Tech.*, vol. 10, no. 7, pp. 1088-1093, October 2000.

[8] K. Rose and S. L. Regunathan, "Towards optimality in scalable predictive coding", preprint.

[9] F. Wu et al. "DCT-prediction based progressive fine granularity scalable coding", *IEEE Intl. Conf. on Image Proc.*, Vancouver CA, Oct. 2000.

[10] A. R. Reibman et al. "Multiple description coding for video using motion compensated prediction", *Intl. Conf. on Image Proc.*, Oct. 1999.

[11] A. R. Reibman and L. Bottou, "Managing drift in a DCT-based scalable video encoder", *IEEE Data Compression Conf.*, March 2001.

[12] L. Bottou et al., "The Z-Coder Adaptive Binary Coder", *IEEE Data Compression Conf. 1998*, Snowbird, UT, 1998.