Eliminating Drift in SNR Scalable Video Coding

Amir Schricker
amirs@bme.jhu.edu

Erica Richstad
erichstad@netscape.net

EE 520.643: Digital Multimedia Coding and Processing
Final Project
The Johns Hopkins University
Spring 2003

1 Introduction

With the growing popularity of the Internet as a tool for sharing multimedia, the ability to deliver digital video efficiently is extremely valuable. Despite the large number of Internet users, many still access via low bandwidth connections or with computers that have limited resources. Many multimedia providers are thus attempting to address this problem by employing scalable video coding as a method of delivering videos of varying quality to customers based on their connection speed. Scalability allows for the generation of a single encoded representation of the input video from which decoded versions of varying quality can be extracted.

Aspects of the video quality that can be adjusted include its signal-to-noise ratio (SNR), temporal resolution, or spatial resolution. SNR scalability, which encodes the video signal into several layers, each with varying SNR quality, is a commonly used approach. However, SNR scalability has some inherent problems: one of these being an error commonly known as drifting. Drifting, a disparity due to error accumulation in motion compensation, is an unwanted effect that arises when there is a mismatch between the coder and decoder. For example, if the coder encodes two bitstreams but the decoder receives only one of them, drifting appears in the reconstructed video as the decoder tries to do motion compensation on different information than the encoder. Drifting is an especially serious problem because the visual effects last for longer that the duration of the frame, thus degenerating the video quality with time.

In this project, we investigate various SNR-scalable coders and decoders that attempt to minimize or eliminate the effects of drifting. The rest of this paper is organized as follows: Section 2 evaluates several codecs that exhibit and are limited by the drifting effect. We also present an encoder that eliminates drifting. Section 3 reports the results of this drift-free encoder, and Section 4 discusses its performance. Finally, conclusions follow in Section 5.

2 Encoder and Decoder Designs

All of the encoders in this paper build upon the standard closed-loop motion compensation encoder as seen in Figure 1. This encoder operates frame-by-frame on the luminance information of a QCIF video sequence and performs block motion estimation/compensation on 16x16 pixel macro blocks. For each block the set of motion vectors and the residue (prediction error) are written to the encoded bitstream.

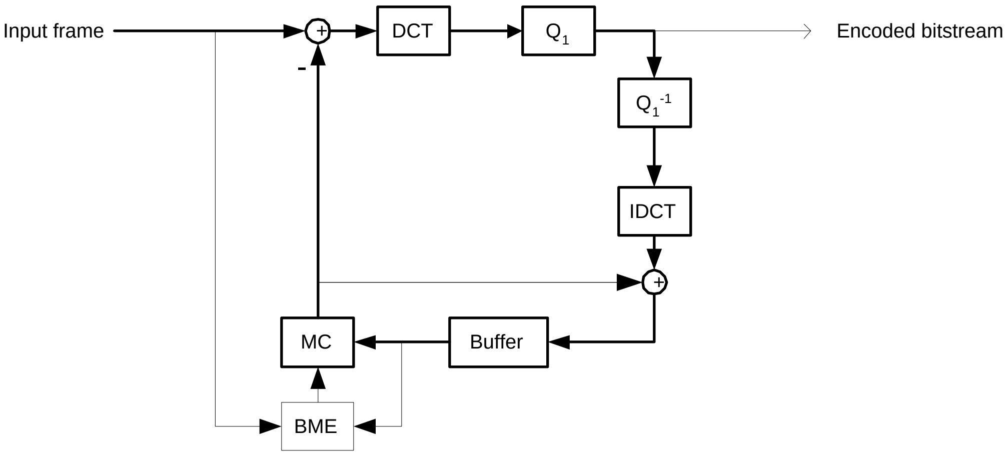

Figure 1: Standard motion compensation encoder block diagram. ("MC" = motion compensation)

2.1 Enhancement Layer Drifting

The first encoder we investigated is shown in Figure 2. The output of the discrete cosine transform (DCT), instead of passing through a single quantizer, now branches into two quantizers. Coarse quantization is performed in Q_1, fine quantization is performed in Q_2, and the base-layer and enhanced-layer bitstreams are created respectively. When the decoder, shown in Figure 3, reconstructs the base-layer bitstream, the resulting video sequence appears as expected. However, drifting occurs when the decoder receives the enhancement layer because the encoder performs block motion estimation (BME) on the base layer only while the decoder attempts to perform motion compensation using both the base and enhanced layer.

Another unwanted effect created by this encoder is the accumulation of quantization noise in the enhancement layer. Unlike the base-layer bitstream, the enhancement layer is not fed back into the loop. Quantization noise then builds up when the decoder uses the enhancement layer for video reconstruction. This noise accumulates quickly, as is evidenced by poor video quality within the first 10 frames of the sequence, and it only occurs when the enhancement layer is incorporated. Figure 4 shows equivalent frames of a base-layer reconstruction and a reconstruction that also utilizes the enhancement layer. This situation is similar to the relationship between open- and closed-loop differential pulse-code modulation (DPCM).

2.2 Base Layer Drifting

Our next step was to take advantage of the closed-loop system that we use for the base layer. By sending the enhancement layer coefficients through the closed loop we hoped to eliminate the quantization error accumulation seen in the first system. Now, although the quantization error should disappear, we expect to see drifting in the base layer as the encoder uses both layers in for motion estimation/compensation while the decoder may only receive the base. We also expect to see drifting artifacts increase as the enhancement-layer stepsize decreases, signifying that the enhancement layer contains more information. The block diagram for this system is seen in Figure 5 below. As before, the DCT coefficients are passed through a coarse quantizer to get the base layer and a fine quantizer to get the enhancement layer. This time, however, the finely quantized

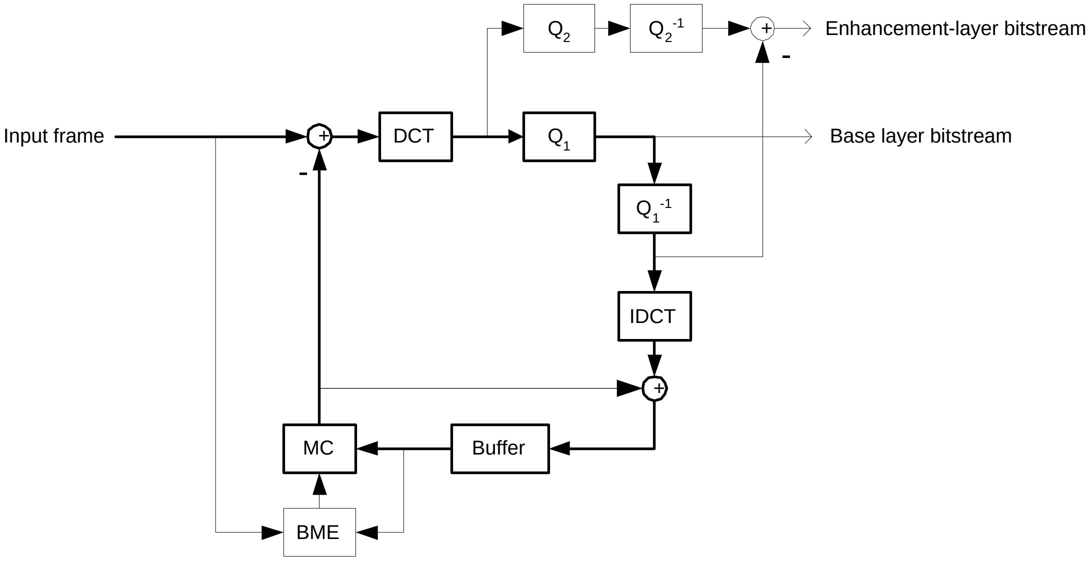

Figure 2: Encoder that exhibits drifting in the enhancement layer but not the base layer.

coefficients of the enhancement layer are inverse-quantized with the same stepsize and added back into the main closed-loop with the inverse-quantized base layer coefficients.

As expected, this method completely eliminated the quantization error accumulation seen in the first encoder. However, in Figure 6 drifting can be seen in the picture on the left. The video was encoded with a quantization stepsize of 64 for the base layer and 16 for the enhancement layer. The picture on the right is the decoded base and enhancement layers, and since the decoder operates on exactly the same information as the encoder, the reconstruction quality is good. The picture on the left, however, is the decoded base layer, which demonstrates drifting. Since the decoder performed the motion compensation on just the base layer, and not both layers as in the encoder, there is a mismatch, and drifting occurs.

This error can be seen graphically in Figure 7 below. The upper curve represents the PNSR of the decoded base layer when encoded with both stepsizes equal to 64. In this case the enhancement layer contains no information, and there is no mismatch between the encoder and decoder. Therefore, the PNSR is constant, and in this case equal to 37dB. The lower curve plots the PSNR of the decoded base layer when the enhancement stepsize is changed to 16. As expected, the PSNR begins at 37dB as in the upper curve, but decreases with each frame as drifting accrues.

2.3 Two-Loop Design

Our final encoder design drew upon the advantages of the prior two encoders in an attempt to eliminate drifting effects from both layers. To eliminate drift effects in the base layer, the enhancement-layer bitstream should not be fed back into the main loop, but rather kept separate as seen in the first example. On the other hand, in order to eliminate drifting in the enhancement layer, and also to counteract quantization error accumulation, the enhancement layer should be generated in a closed-loop design as seen in the second example. Unlike these two encoders, though, these loops should ideally be kept independent so that reconstructing with or without the enhancement layer

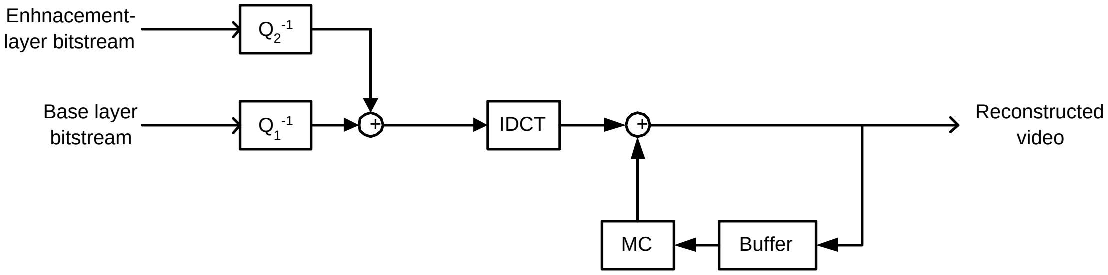

Figure 3: Decoder used in all the codecs.

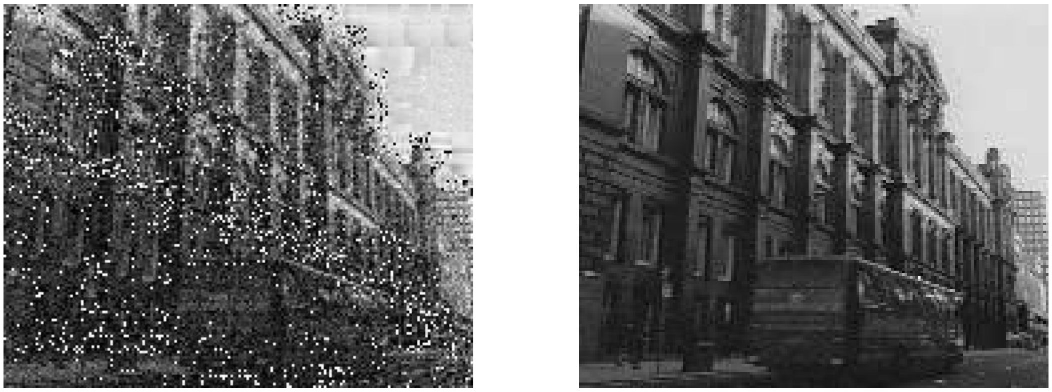

Figure 4: a) An image from a video reconstructed from both the base and enhancement layer. Note the excessive amount of quantization error accumulation, resembling noise. b) Image from a reconstruction using only the base layer, which exhibits poorer quality but no quantization error accumulation.

will not cause drifting error.

Combining both of these ideas we devised a two-loop encoder that has each layer in its own loop. As shown the block diagram in Figure 8, the reconstructed frame in the base layer is subtracted from the input frame before entering a closed loop that generates the enhancement layer. The base-layer is quantized coarsely while the enhancement layer uses a finer quantization stepsize. In addition, since motion compensation would also need to be performed in the enhancement loop, the motion vectors that are used in the enhancement layer are the same ones calculated for the base-layer. Because of these two independent closed loops, we expect that this encoder will eliminate drift altogether regardless of whether or not the enhancement layer is incorporated in the decoding process. This coding scheme has also been proposed by Arnold, *et al.* [1].

3 Results of Two-Loop Encoder

In Figure 9 the result of the two-loop system can be seen. Both pictures are from a video encoded with a base-layer stepsize of 64 and an enhancement layer stepsize of 16, but the image on the left is

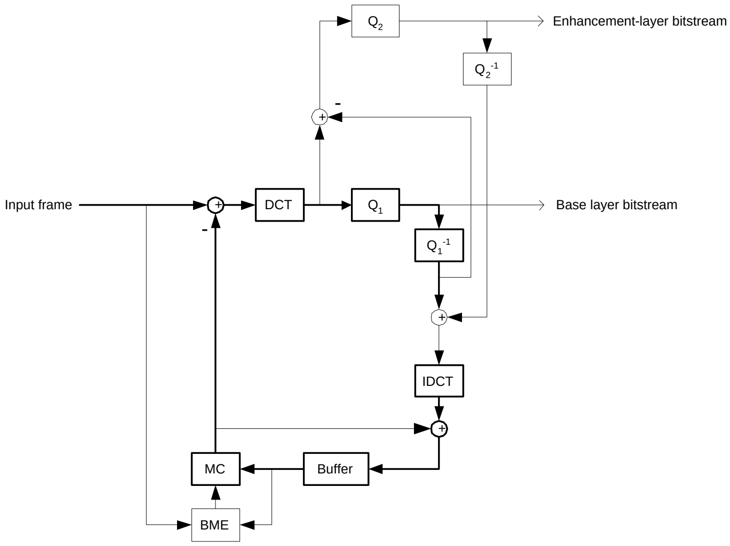

Figure 5: Encoder that contains drift in the base layer but not the enhancement layer.

the decoded base layer only while the image on the right is the result of both layers being decoded. Neither picture exhibits drifting, in contrast to method 2 where the base-layer picture drifted. The effects of drifting are evaluated quantitatively in Figure 10. The curves show the PSNR of the base-layer decoding and the base plus enhancement decoding. First notice that both curves have constant value for each frame, indicating that there is no drifting error. Also, the decoded base-layer has a constant PSNR of 37dB, while the decoded enhanced layer has a much higher constant PSNR of 48dB. This result clearly shows that adding the enhancement layer does increase the quality of the video as desired, with neither of the reconstructions exhibiting drift.

This two-loop design performs exactly as we had hoped and expected. The video quality increases with the addition of the enhancement layer, but the enhancement layer is truly optional: drifting will occur neither when it is present nor omitted. This completely satisfied the goal of having scalable video such that different users can have two different quality video streams from the same encoded bitstream, while at the same time eliminating quantization accumulation error and drifting.

Clearly, as described above, the greatest advantage of this design is scalable video with no drifting error. Unfortunately, this drift-free scalability comes at a price. With two loops being incorporated into this algorithm, the number of calculations per frame nearly doubles, not including those used by motion estimation/compensation. For long videos or videos with high resolution, the computation time will quickly increase and prevent this encoder from being used in real-time. If, however, real-time encoding is not desired and any arbitrary amount of time can be used for encoding, the computation intensity of this scheme should pose no serious problems.

4 Discussion and Future Work

As discussed in the introduction, scalable video coding is very applicable to the Internet and other environments involved with multimedia delivery, and it is a very active area of research. To allow for even greater flexibility and efficiency, several improvements can be made upon our encoding

Figure 6: Reconstructed videos from the encoder in Fig.5 a) Using only the base layer, drifting, the trailing left behind by the moving arms and hands, is very noticeable. b) Drift-free reconstruction results when the enhancement layer is also used.

algorithm.

One improvement is to add more enhancement layers so that more than two videos of differing quality can be reconstructed. However, this would obviously entail adding more loops, and as discussed above, a drawback of the multi-loop method is the increased time for computational. One approach to minimizing the number of computations is to focus on the block motion estimation algorithm. The most computationally expensive section of this video coder is the motion estimation, because an exhaustive search strategy is used. The computation time would be greatly reduced simply by replacing the exhaustive search with a more efficient search strategy, such as a divide-and-conquer, log, or spiral search.

Another possibility for future improvements is to encourage the motion vectors to be zero. The rationale behind this idea is the possibility for increased compression ratios after entropy coding. With more zero motion vectors, an entropy coder would be much more successful at reducing the size of the encoded information (bitstreams and motion vector data). In order to implement this idea, we can bias the motion estimation algorithm towards the zero motion vector by weighing the prediction error for the zero vector more heavily than those of other vectors. Specifically, if two motion vectors both minimized the prediction error for a macroblock by a similar amount, and one motion vector was the zero vector, it should be selected as the motion vector for that macroblock, even if its prediction error is slightly higher than the other.

One final improvement would be to code some frames as “intra”, indicating a good match was not found in motion estimation. This enhancement would improve the general robustness and efficiency of this encoder. Furthermore, in the event that drifting has not been entirely eliminated (for any reason), forcing the encoder also to encode every few frames as intra would reset the amount of drift to zero at those frames.

5 Conclusion

In this project we investigated different methods of scalable video coding in an attempt to minimize drifting error from encoder/decoder mismatch. We first used two different quantizer stepsizes for the enhancement and base layers, but found that if the enhancement layer was not part of the

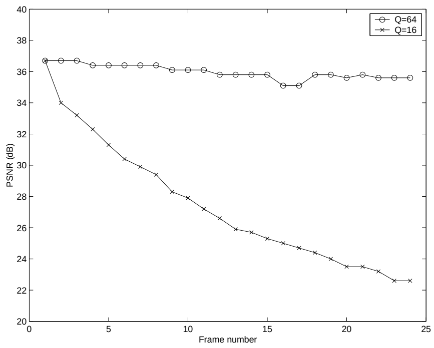

Figure 7: Quality of each frame in a base layer and base+enhancement layer reconstruction. The drifting in the base layer ($Q = 16$) is evidenced by a decreasing PSNR. For $Q = 64$, enhancement layer contains no information and thus reconstruction is normal.

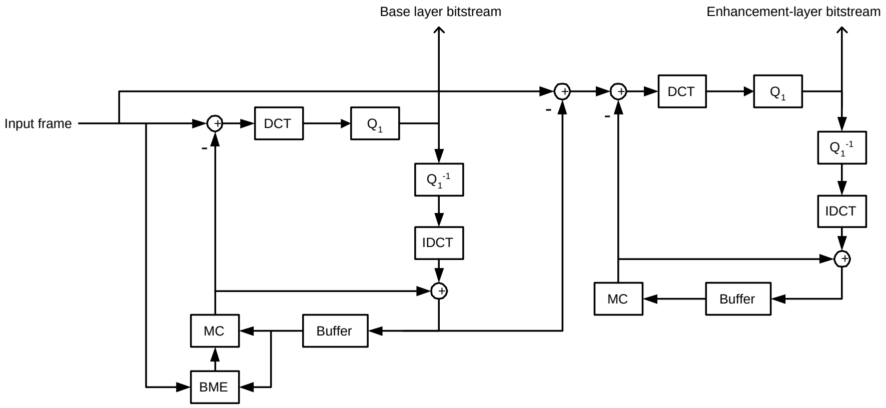

Figure 8: Two-loop encoder that eliminates drift in both layers.

closed loop, quantization error accumulation occurred. Next, in an attempt to eliminate this error we incorporated the quantized coefficients of the enhancement layer into the closed-loop base-layer encoder. This did indeed eliminate the error, but also caused drifting in the base layer when the decoder did not receive the enhancement layer. Finally, we incorporated motion compensation into the enhancement layer encoder by making two separate closed loops, but using the motion vectors calculated with the base-layer encoder. As hoped, this method eliminated drifting when just the base layer was decoded, and provided higher quality video when both base and enhancement layers were decoded.

Figure 9: Images resulting from the two-loop encoder: a) Base layer only ($Q_1 = 64$) b) Base layer plus enhancement layer ($Q_2 = 16$). Neither image exhibits drift.

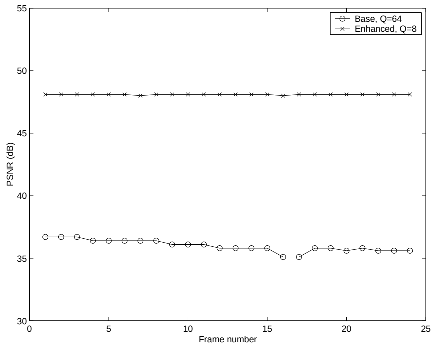

Figure 10: Quality of frames in base-layer reconstruction and base+enhancement layer reconstruction. Each reconstructed video is drift-free, and the enhancement layer is clearly of higher quality than the base layer.

References

[1] Arnold JF, Frater MR, Wang Y. “Efficient Drift-Free Signal-to-Noise Ratio Scalability” *IEEE Trans. Circuits and Systems for Video Technology.* 10:70-82 (2000)