JOINT TRANSCODING OF MULTIPLE MPEG VIDEO BITSTREAMS

Hani Z. Sorial, William E. Lynch, and André Vincent†

Centre for Signal Processing and Communications
Department of Electrical & Computer Engineering
Concordia University, Montreal, Quebec,
H3G 1M8, Canada

†Communications Research Centre
Ottawa, Ontario,
K2H 8S2, Canada

ABSTRACT

This paper addresses the problem of bit-rate conversion of a previously compressed video. We provide an MPEG joint transcoder for transcoding several video bitstreams simultaneously. We show that joint transcoding reduces the quality variation between multiple video sequences, as compared to independently transcoding each sequence at a fixed bit rate. Hence, joint transcoding results in a better utilization of the channel capacity. The joint transcoder can be used in a congested communication network as an alternative to data/packet dropping, and in applications which require multiplexing video signals onto a fixed communication channel such as video servers providing video on demand (VOD) service.

1. INTRODUCTION

The transmission of MPEG compressed video over channels with different capacities may require a reduction in bit rate if the transmission media has a lower capacity than the capacity required by the video bitstream. There are different approaches to the problem of bit rate conversion of MPEG compressed video. A straightforward approach is to fully decode the video, then to re-encode it at a lower bit rate. This approach has two disadvantages. First, the MPEG encoding algorithm usually requires high computational power. Thus, this approach is not an efficient solution in terms of implementation complexity, delay and cost. Second, errors are introduced due to the repeated compression/decompression of video [1, 2].

Another approach to bit rate conversion is to use scalable coding techniques [3, 4]. There are two disadvantages to this approach. First, scalability provides only a limited number of possible transmission bit rates. Second, in order to control transmission rates using layered coding, the encoder must take into account the congestion control policy of the network when a connection is set. This however may not be known when the video was first compressed for storage.

A more reliable approach is to transcode the compressed video to the desired bit rate. This is achieved by partial decoding of the bitstream, then performing the rate conversion by re-quantizing the DCT (Discrete Cosine Transform) coefficients. This approach has two advantages over full decompression followed by re-compression of the video. First, since motion estimation is the most computationally intensive operation in the MPEG encoding algorithm, the complexity is significantly reduced by using the same motion vectors in transcoding. Second, this approach avoids some of the errors introduced when the video is fully decompressed and then compressed in a second generation [1, 2].

This research was supported by a grant from the Canadian Institute for Telecommunications Research (CITR) under the NCE program of the Government of Canada.

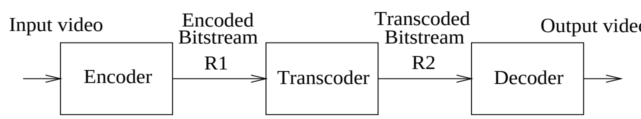

Figure 1: A basic video coding system including a transcoder.

Figure 1 shows a basic block diagram of a video coding system that includes a transcoder. The encoder compresses an input video at a bit rate $R1$, then subject to certain constraints, the transcoder converts this compressed video at a bit rate $R2 < R1$. Next, the decoder decompresses the transcoded video bitstream for display.

The problem of transcoding has been studied by many researchers [5, 6, 7, 8, 9, 10]. In [5], a two-stage coder for distribution of video at different bit rates is proposed. However, this method has a large overhead in storage capacity. In [6], a simple open loop transcoder architecture is presented. This technique however does not take into account the propagation of errors (drift errors) which results from re-quantizing a motion-compensated compressed video. Another approach for high-quality transcoding is to use a closed loop technique that compensates for drift errors, as proposed in [7, 8, 9, 10]. Compensation for drift errors is performed either in the DCT domain [7, 8] or in the spatial domain [9, 10]. Transcoding optimization based on the minimization of the transcoder distortion for a given bit rate using a Lagrangian algorithm is presented in [8]. A detailed discussion of transcoding complexity and performance, including an analysis of the extra distortion introduced by re-quantization, is given in [10].

By distributing the channel capacity among multiple video programs according to their degree of coding complexity, joint coding for multi-program video transmission has been shown to be more efficient than independent coding [11, 12]. In this paper, we

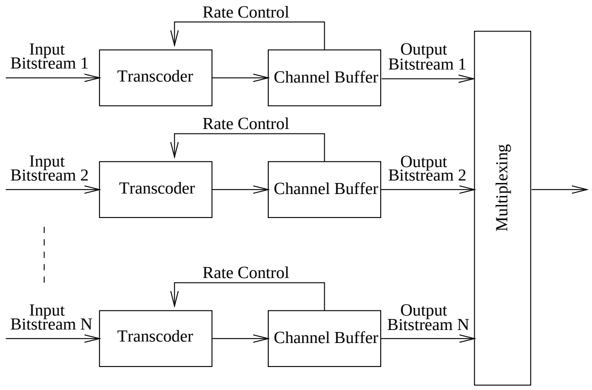

Figure 2: Independent transcoding of multiple video bitstreams.

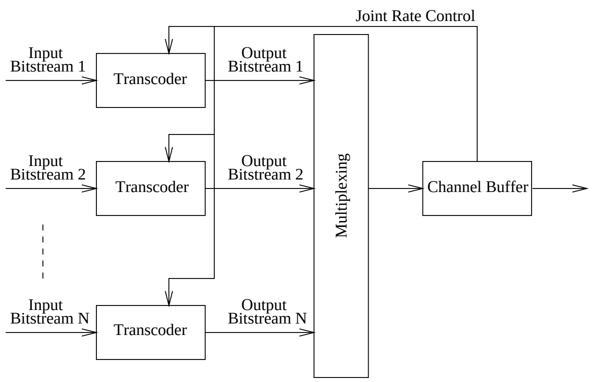

Figure 3: Joint transcoding of multiple video bitstreams.

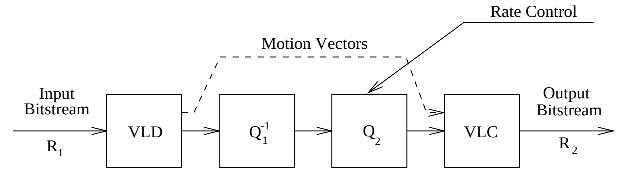

Figure 4: Transcoder structure with no drift correction.

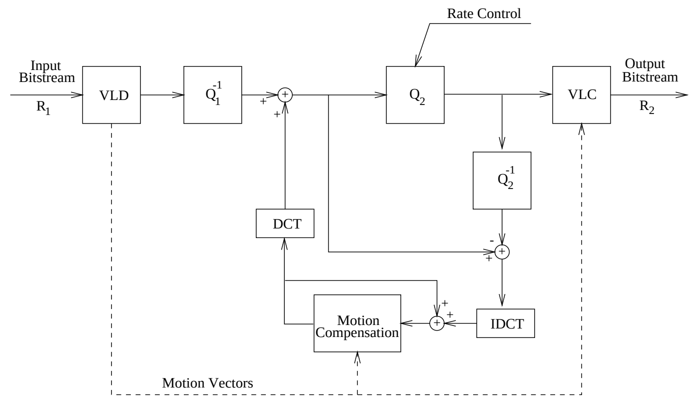

Figure 5: Transcoder structure with drift correction in the spatial domain.

present an MPEG joint transcoder for multiple video bitstreams. The organization of this paper is as follows. Section 2 presents the proposed joint transcoding method. Section 3 provides experimental results for independent and joint transcoding. Finally, Section 4 concludes the paper.

2. JOINT TRANSCODING

Figures 2 and 3 illustrate independent and joint transcoding of multiple video bitstreams, respectively. In independent transcoding, each rate control works independently and uses a channel buffer to distribute the bits between the frames of one sequence. In joint transcoding, the joint rate control distributes the bits between the frames of one sequence as well as among the sequences depending on their degree of complexity. For both transcoding cases, once the number of bits for a frame has been determined, a virtual buffer is used to distribute the bits between the macroblocks of the frame.

We provide two types of joint transcoders. The first uses a straightforward re-quantization method, without compensating for drift errors. The basic structure of a single transcoder with no drift correction is shown in Figure 4. The advantages of this structure are its lower complexity and smaller memory requirements. This type of transcoder does not require frame buffers. It is more suitable for low delay and low cost applications, and for video bitstreams having small group of pictures (GOP) structures, which reduce the effects of drift.

The second type of joint transcoder uses a feedback loop to compensate for drift errors. This method provides a higher quality transcoding, but with more complexity and memory requirements. The extra complexity added is due to the operations required for drift correction, while the increase in memory requirements is due to the frame buffers needed for drift compensation. Figure 5 shows the block diagram of a single transcoder with drift correction in the spatial domain. If drift correction is used for P and B picture coding types, then two frame buffers are needed. Memory requirements can be reduced to one frame buffer by performing drift correction only on P pictures, since re-quantization errors in B pictures do not propagate.

For both transcoding types, the re-quantization process is done on a macroblock basis. Furthermore, the same motion vectors of the input bitstreams are re-used. This avoids motion estimation, the most computationally intensive operation of the MPEG encoding algorithm. As expected, this significantly reduces the complexity of the system, as compared to a cascaded decoder-encoder system. Macroblock types however can change after re-quantization. For example, a motion-compensated coded macroblock may be changed to a motion-compensated not coded macroblock if all the DCT coefficients are zero after re-quantization.

A brief discussion of quantization and rate control operations follows.

2.1. Quantization

In MPEG standards, the quantizer step size that is used for each DCT coefficient includes two components, a quantization coefficient which specifies a minimum step size for the particular DCT coefficient, and a quantization scaling parameter $Mquant$ used for bit-rate control. As $Mquant$ increases, the quantization of the DCT coefficients is coarser and consequently, the output bit rate decreases. The quantization process [7] can be expressed as

$$F_Q(u,v) = Round\left(\frac{F(u,v)}{Mquant \times Q(u,v)}\right) \qquad (1)$$

where $F(u,v)$ is the input DCT coefficient and $Q(u,v)$ is its corresponding entry from the quantization table, and $F_Q(u,v)$ is the quantized DCT coefficient.

At the decoder, inverse quantization [7] may be performed as

$$\tilde{F}(u,v) = F_Q(u,v) \times Mquant \times Q(u,v) \qquad (2)$$

where $\tilde{F}(u,v)$ is the inverse quantized DCT coefficient.

2.2. Re-quantization and Bit-rate Conversion

Bit rate conversion can be performed in the compressed domain by re-quantization of the DCT coefficients. In its simplest case, this can be represented by

$$F_{Q_{new}}(u,v) = Round\left(\frac{F_{Q_{old}}(u,v) \times Mquant_{old}}{Mquant_{new}}\right) \qquad (3)$$

where $F_{Q_{old}}(u,v)$ is the input quantized DCT coefficient and $F_{Q_{new}}(u,v)$ is its re-quantized version, $Mquant_{old}$ is the input quantization scaling parameter and $Mquant_{new}$ is the value required to meet the target bit rate.

Note however that the above equation does not take into account the propagation of re-quantization errors (drift) due to motion compensation.

2.3. Rate Control

In this paper, the joint rate control for transcoding is implemented as described in [12]. The bit rate control is based on the Test Model document, version 5 ($TM5$) [13]. The algorithm consists of three steps, a *target bit allocation* which estimates the number of bits available to encode the next picture, a *rate control* that uses a “virtual buffer” to set the reference value of the quantization parameter for each macroblock, and an *adaptive quantization* which modulates the reference value of the quantization parameter according to the spatial activity in the macroblock and the average activity in the picture in order to derive the value of the quantization scaling parameter $Mquant$ used to quantize the macroblock.

Usually, the activity is computed in the pixel domain and is an important parameter in quantization, as it reflects the relative complexity of different macroblocks within the picture. Since the Human Visual System (HVS) is more sensitive to coding errors in the low-frequency regions of the picture (i.e., low-activity regions), these regions are usually quantized with a finer quantization step size. However, since local activity (over a macroblock) and average activity (over the picture) are not included in the bitstream, they are estimated from local and average quantization parameters, respectively, as proposed in [6]. Specifically, the quantization scaling parameter for transcoding is expressed as:

$$Mquant_{new}(j,k) = Mquant_{ref}(j,k)\frac{Mquant_{old}(j,k)}{\overline{Mquant_{old}(k)}} \qquad (4)$$

where $Mquant_{old}(j,k)$ is the old quantization scaling parameter of the jth macroblock in the kth picture, $Mquant_{new}(j,k)$ is its corresponding parameter for re-quantization, $Mquant_{ref}(j,k)$ is the reference quantization parameter, and $\overline{Mquant_{old}(k)}$ is the average quantization parameter over the kth picture. The ratio $Mquant_{old}(j,k)/\overline{Mquant_{old}(k)}$ is the estimate of the normalized activity for the jth macroblock in the kth picture. In the experiments, we use the average quantization parameter of the last decoded picture of the same type for $\overline{Mquant_{old}(k)}$.

3. EXPERIMENTS

We use 150 frames (each 352×240) of Flowers (Flr.), Football (Ftb.), Table Tennis (Ten.), Miss America (Mis.), and Salesman (Slm.) sequences with a color sampling ratio of 4:2:0. The group of pictures (GOP) length is 15, with a structure IBBPBBPBBPBBPBB.

Figures 6 and 7 show the PSNR (Y-component) versus frame number for independent transcoding and joint transcoding of the above sequences, respectively. Both transcoding cases use drift correction. Each sequence was originally coded at 2Mb/s. The total input and output bit rates for joint transcoding are 10Mb/s and 7.5Mb/s, respectively. In independent transcoding, each sequence is transcoded separately at a constant bit rate of 1.5Mb/s. Notice that joint transcoding reduces the quality (PSNR) variation between the video sequences, as compared to independent transcoding. Variation is reduced more at higher bit rates and higher transcoding ratios.

Table 1 shows the average PSNR (Y-component) in both independent and joint transcoding of the above sequences. Results are shown for transcoding with drift correction and with no drift correction. Here, Δ denotes the difference in PSNR between joint and independent transcoding. The bit rates for both joint and independent transcoding are shown in Table 2.

Note that joint transcoding results in an increase in PSNR of higher complexity sequences, as compared to independent transcoding. For example, the PSNR of Flowers is increased by 3.0 dB and 3.8 dB for joint transcoding with drift correction and joint transcoding with no drift correction, respectively. Furthermore, the average gain achieved by using drift correction is 0.6 dB for both joint transcoding and independent transcoding. In general, this gain is higher for larger transcoding ratios.

Notice that the average gain achieved by using joint transcoding with no drift correction is higher than independent transcoding with drift correction. This may be important in real-time and low delay applications where joint transcoding with no drift correction can be used and still provides a better average PSNR than independent transcoding with drift correction.

4. SUMMARY

In this paper, we presented joint transcoding of multiple MPEG video bitstreams. Results show that joint transcoding distributes the channel capacity among the video sequences according to their degree of complexity. This reduces the quality variation between the video sequences, as compared to independent transcoding of each sequence at a fixed bit rate. Consequently, joint transcoding results in a better utilization of the channel capacity and leads, on average, to an increase in the PSNR of the transcoded sequences. We also presented in this work two types of joint transcoders with different complexity and memory requirements.

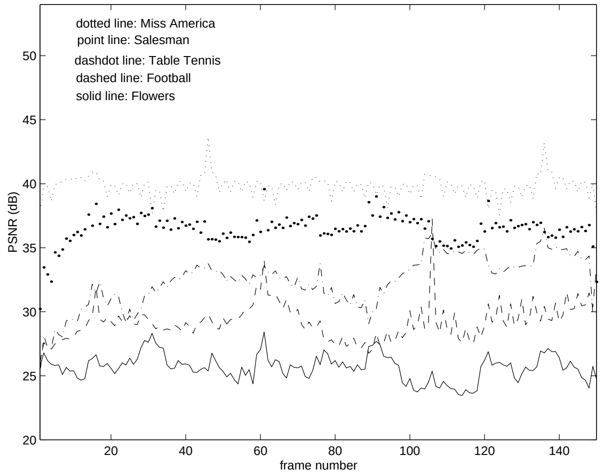

Figure 6: Independent transcoding (with drift correction) of five bitstreams. The input and output bit rates for each bitstream are 2Mb/s and 1.5Mb/s, respectively.

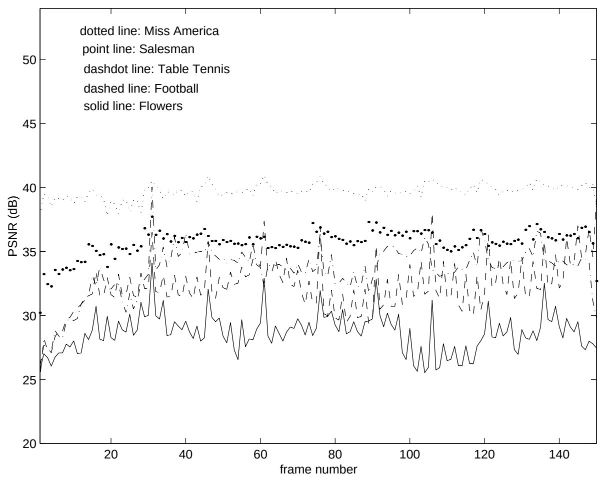

Figure 7: Joint transcoding (with drift correction) of five bitstreams. The total input and output bit rates are 10Mb/s and 7.5Mb/s, respectively.

Table 1: Transcoding average PSNR. The total input and output bit rates of the five sequences are 10Mb/s and 7.5Mb/s, respectively.

Transcoding with drift correction						
Case	PSNR (dB)					Average
	Flr.	Ftb.	Ten.	Slm.	Mis.	(dB)
Joint	28.7	32.3	33.5	35.7	39.7	34.0
Indep.	25.7	29.2	32.4	36.4	39.8	32.7
Δ	3.0	3.1	1.1	-0.7	-0.1	1.3

Transcoding with no drift correction						
Case	PSNR (dB)					Average
	Flr.	Ftb.	Ten.	Slm.	Mis.	(dB)
Joint	28.6	32.1	33.1	34.3	38.9	33.4
Indep.	24.8	28.7	32.0	36.1	39.1	32.1
Δ	3.8	3.4	1.1	-1.8	-0.2	1.3

Table 2: Transcoding output bit rates.

Case	Bit Rate in Mb/s				
	Flr.	Ftb.	Ten.	Slm.	Mis.
Joint (drift correct.)	2.0	2.0	1.8	1.1	0.6
Joint (no drift correct.)	2.0	2.0	1.8	1.0	0.7
Independent	1.5	1.5	1.5	1.5	1.5

5. REFERENCES

[1] A. T. Erdem and M. I. Sezan, “Multi-generation Characteristics of the MPEG Video Compression Standards,” *IEEE Proceedings International Conference on Image Processing*, vol. 2, pp. 933–937, November 1994.

[2] H. Z. Sorial and W. E. Lynch, “Multigeneration of Transform Coded Images,” *Proceedings of SPIE Visual Communications and Image Processing, San Jose, California*, vol. 3024, pp. 1394–1405, February 1997.

[3] T. Sikora, “MPEG Digital Video-Coding Standards,” *IEEE Signal Processing Magazine*, pp. 82–100, September 1997.

[4] ISO/IEC 13818-2, *Generic Coding of Moving Pictures and Associated Audio Information: Video*, May 1994.

[5] M. Pereira and A. Lippman, “Re-codable Video,” *IEEE Proceedings International Conference on Image Processing*, vol. 2, pp. 952–956, November 1994.

[6] Y. Nakajima, H. Hori, and T. Kanoh, “Rate Conversion of MPEG Coded Video by Re-quantization Process,” *IEEE Proceedings International Conference on Image Processing*, vol. 3, pp. 408–411, October 1995.

[7] R. J. Safranek, C. R. Kalmanek, and R. Garg, “Methods for Matching Compressed Video to ATM Networks,” *IEEE Proceedings International Conference on Image Processing*, vol. 1, pp. 13–16, October 1995.

[8] P. Assunção and M. Ghanbari, “A Frequency-Domain Video Transcoder for Dynamic Bit-Rate Reduction of MPEG-2 Bit Streams,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 953–967, December 1998.

[9] P. Assunção and M. Ghanbari, “Post-Processing of MPEG2 Coded Video for Transmission at Lower Bit Rates,” *IEEE International Conference on Accoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, pp. 1998–2001, May 1996.

[10] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, “Transcoding of MPEG Bitstreams,” *Signal Processing: Image Communication*, vol. 8, pp. 481–500, 1996.

[11] L. Wang and A. Vincent, “Joint Coding for Multi-Program Transmission,” *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 425–428, September 1996.

[12] L. Wang and A. Vincent, “Bit Allocation for Joint Coding of Multiple Video Programs,” *Proceedings of SPIE Visual Communications and Image Processing, San Jose, CA*, pp. 149–158, February 1997.

[13] ISO/IEC-JTC1/SC29/WG11, MPEG 93/457, *Test Model 5*, April 1993.