Optimal Frame Skipping for H.263+ Video Encoding

Giovanni Motta, **James A. Storer**
Computer Science Department,
Brandeis University,
Waltham, MA-02454, USA
{gim, storer}@cs.brandeis.edu

Bruno Carpentieri
Dip. di Informatica ed App. "R.M. Capocelli",
Universita' di Salerno,
84081 Baronissi (SA), Italy
bc@dia.unisa.it

ABSTRACT

When encoding low bit-rate video at constant bit-rate, state of the art video encoders, like the H.263+, can be forced to skip frames. Frames are mostly skipped after "scene cuts", i.e. immediately after the beginning of a new scene. We address the problem of determining, given a video sequence, the minimal number and the exact positions of the frames that must be skipped in order to achieve, for a given coding quality, a predetermined target bit rate. Reducing the number of skipped frames improves the overall quality of the encoded video and reduces the jerkiness associated to the skips. As a side effect, SNR is also seldom improved because of the better bit allocation that our algorithm provides. The frame-layer rate control that we propose can be used in several existing video coders and it is also compatible with rate-distortion optimized macroblock-layer rate controls.

INTRODUCTION

Current video compression systems, such as H.263 and encoders of the MPEG family, use a hybrid coding scheme that combines motion-compensated prediction and block based Discrete Cosine Transform coding. The MC-prediction model breaks down when a scene changes. The first frame of the new scene cannot be predicted efficiently from the precedent and most macroblocks have to be "Intra" coded. This has the unpleasant effect that the encoded size of the current frame is substantially larger than the target bit rate. When the encoder tries to achieve a quasi-constant bit rate, for example to fully utilize a transmission channel, the frames that have a large encoded size present a problem. Transmission of large frames may require a time substantially higher than the time available for a single frame and in order to keep the decoder synchronized, the encoder is forced to skip the encoding of one or more subsequent frames. We present an optimal algorithm that minimizes the number of skipped frames while keeping the decoder synchronized.

Given a video sequence, our algorithm determines the minimal number and the exact positions of the frames that must be skipped in order to achieve, for a given coding quality, a predetermined target bit rate. Our algorithm uses a Dynamic Programming approach and it can be used in low-bandwidth applications where the encoder has some look-ahead capability. Its asymptotic time complexity is linear in the number of frames being encoded.

Although the optimization requires some additional encoding complexity, there is no change in decoding complexity (in fact, no change to the decoder at all). Possible areas that can benefit of this algorithm are video broadcasting, off-line video coding, wireless video transmission, video over IP, etc.

The minimum gain achievable was assessed experimentally in the framework of H.263+

video encoder on a set of standard and non-standard sequences by using a simplified skipping heuristic. Results confirm that our method provides an effective alternative to current frame skipping strategies and it can be used to improve the quality of low bit-rate video. Even with the simplified heuristic, the number of skipped frames is reduced and the quality of the scene cuts is improved with respect the TMN-8 rate control.

PAST WORK

Issues related to frame type selection and rate control have been considered by a number of authors. For example: Kozen et al. [4] consider a linear programming approach to optimal frame dropping where a fixed number of frames are dropped in order to minimize the interval of non-playable frames in MPEG encoded sequences. Wiegand et al. [10] present a dynamic programming algorithm that joint optimizes frame type selection and quantization steps in the framework of the H.263 video coding. Lee and Dickinson [5] address a similar problem in the framework of MPEG encoding, where each group of frames is isolated and both frame type selection and quantization steps are optimized by using a combination of dynamic programming and Lagrange optimization. Sullivan and Wiegand [9] present an approach based on rate-distortion optimization. Their approach constitutes the base for the H.263+ Test Model Near-Term Version 11 (TMN-11).

PROBLEM DESCRIPTION

Current video compression systems such as H.263 and MPEG use a combination of motion-compensated prediction and DCT coding. Each video frame is partitioned in *macroblocks* and each macroblock is encoded as a motion compensated difference with a previously sent macroblock (P-mode) or by direct quantization of its DCT coefficients (I-mode). Encoder performs mode selection by looking at the magnitude of the prediction error. We define "scene

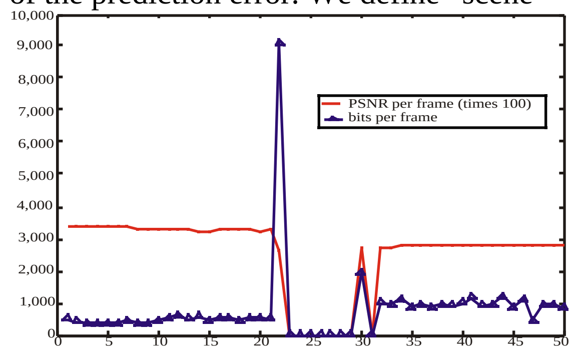

Figure 1: Sequence of 50 frames across a scene cut in one of the files used in our experiments (claire.qcif 80-100 followed by carphone.qcif 1-29) encoded at 32 Kbit/s with TMN-8 rate control.

Frame N.	n-2	n-1	n	n+1	n+2	n+3	n+4	n+5	n+6	…
Encode	n-2	n-1	n		*skip*		n+4	*skip*	n+6	…
Transmit	n-2	n-1		n			n+4		n+6	…
Display	…	n-2		n-1			n		n+4	n+6

Figure 2: Encoding, transmitting and decoding/displaying a sequence of frames with a H.263+ encoder (TMN-8 rate control). The sequence contains a scene cut between the frames ***n-1*** and ***n***.

cut" a video frame in which the MC-prediction is ineffective and a large number of I-macroblocks have to be sent to the decoder. This may happen immediately after a scene change or when a scene contains too much movement. Transmission of frames with a large number of I-macroblocks may require a time substantially higher than the time available for a single frame and in order to keep the decoder synchronized, the encoder is forced to skip the encoding of one or more subsequent frames. Fig.1 and Fig.2 provide a close look at the behavior of a H.263+ encoder that uses a TMN-8 rate control while encoding a scene cut. Except for the first frame of the sequence (always encoded in Intra mode) almost every frame is encoded by using MC-prediction (Inter mode). If we focus our attention around a scene cut (frame ***n*** in Fig.2) we note that, because of the I-macroblocks, this frame takes a considerable time to be transmitted (in this case, 3 additional frames of time). In the meanwhile, while waiting for a complete reception of the frame ***n***, the decoder keeps showing the frame ***n-1*** on the screen.

To maintain synchronization with the original sequence, the encoder is now forced to skip frames while waiting for the transmission of frame ***n*** and, because of this skipping the next frame to be encoded will be frame **n+4**. For the example depicted in Fig.2, the transmission of frame ***n*** requires the skipping of 3 frames. In general, there will be some number ***k≥0*** such that:

1. There are ***k*** extra units of time in which the frame ***n-1*** is frozen on the screen (**k=3** in Fig.2).

2. There is a "jerk" between the frame ***n*** and the frame ***n+k+1***.

3. Because the frames ***n*** and ***n+k+1*** are not contiguous in time and ***n+k+1*** is predicted from ***n***, the prediction error frequently generates a frame ***n+k+1*** that is too big to be sent in one unit of time (frame ***n+4*** in the Fig.2). This forces the encoder to skip also the frame ***n+k+2*** before encoding ***n+k+3***.

We consider reduction of the effects described in Items 2 and 3 through the optimal selection of the first frame that should be encoded immediately after the termination of a scene (frame ***n-1*** in Fig.2). This is of course only possible if the encoder has some look-ahead capabilities and it may be that encoding is not done in real-time. However, decoding is unaffected (the encoder is not aware that this optimization has taken place), which is all that matters for many practical applications such as video distribution, where a powerful encoder creates a compressed video sequence that is distributed to many real time decoders.

An Optimal Algorithm Based on Dynamic Programming

In this section we give a brief description of a dynamic programming algorithm that produces an optimal encoding under one very reasonable assumption. A more detailed description can be found in Motta et al. [6].

We consider an encoding of a sequence of frames at a given quality ***Q*** optimal if it skips as few frames as possible and among encodings that skip the same number of frames, uses as few bits as possible. In the following description we indicate with ***F[1],..,F[n]*** the sequence of frames being encoded and with ***D*** the maximum possible number of frames that can be skipped in sequence. For any fixed quality ***Q***, ***D*** is a constant independent of ***n***.

We make the following assumption:

- There is a cost ***P[i,d]*** of predicting ***F[i]*** from ***F[i-d]***, compressed at quality ***Q***, that is independent of how ***F[i-d]*** was compressed (so long as the quality of ***F[i-d]*** is ***Q***), ***1≤i≤n***, ***0≤d≤D***. ***P[i,0]*** is simply the cost of encoding ***F[i]*** as an I-frame.

Although this is not likely to be precisely true in practice, it is likely to be a very good approximation to what happens. That is, two encodings of a frame at the same quality are likely to be equivalent in their ability to predict a subsequent frame.

The proposed algorithm is a generalization of the optimal text-paragraphing algorithm and performs, in sequence, the following operations:

1. Compute ***P[i,0]*** for each frame ***1≤i≤n***.

2. Compute ***P[i,d]*** for ***1≤i≤n***, ***1≤d≤D***.
3. Build (right to left) two matrices ***R*** and ***S*** such that:

   a) ***R[i,d]*** is the maximum residual capacity when ***F[i],..,F[n]*** are encoded so that the first non-skipped frame is predicted by ***F[i-d]***, ***1≤i≤n***, ***0≤d≤D***.

   b) ***S[i,d]*** is the number of skipped frames that corresponds to the residual capacity ***R[i,d]***, ***1≤i≤n***, ***0≤d≤D***.

The running time is dominated by the time necessary to complete Step 3 that is $O(D^2 * n) = O(n)$ since ***D*** is a constant independent of ***n***. In practice since ***n*** is simply the number of frames, which is far less than the total number of bits in all frames, the time for Step 3 is likely to be insignificant when compared to Steps 1 and 2, which essentially encode each frame ***D+1*** times. For applications where encoding is done only once and decoding is done many times, increasing the encoding time by a factor of ***D*** may not be a significant problem (***D*** is equal to ***7*** for 30 fps QCIF video encoded at 32Kb/s).

EXPERIMENTAL RESULTS

The algorithm that we propose applies to many video compression systems. We will describe experimental results in terms of the H.263+ standard, because our method is intended specifically for low-bandwidth applications and, at the present, H.263+ is regarded as state of the art low bit rate video coding. Although we embedded our method in the Telnor/UBC H.263+ encoder [1] that uses TMN-8 rate control [2], the optimization can be easily combined with the newer and more powerful TMN-10 and TMN-11 rate controls. Details on the Test Models Near-Term rate controls can be found in Gardos in [4] and Wenger et al. [3].

To assess the performance of the proposed method, eight test sequences were generated so that they would contain a reasonably large number of scene cuts. The sequence *std* consists of the simple concatenation of 9 files widely used as a standard data set. The sequence *std100* was built by interleaving the files in the standard data sets, each of them taken in blocks of 100 frames. The six sequences *commercials* are continuous samplings of TV commercials. Fig.5 shows sample frames taken from both data sets. In order to determine the minimum achievable gain we have experimented with a simple (but not necessarily optimal) fixed strategy that always selects the frame that ends the

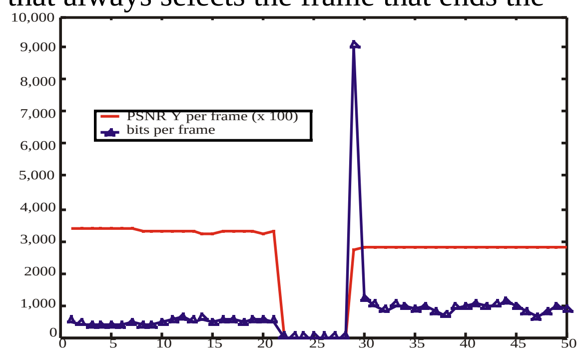

Figure 3: Sequence of 50 frames across a scene cut in one of the files we experimented with (claire.qcif 80-100 followed by carphone.qcif 1-29) encoded at 32 Kbit/s with improved rate control.

Frame N.	n-2	n-1	n	n+1	n+2	n+3	n+4	n+5	n+6	…
Encode	n-2	n-1	*skip*			n+3	n+4	n+5	n+6	…
Transmit	n-2	n-1	n+3				n+4	n+5	n+6	…
Display	…	n-2	n-1				n+3	n+4	n+5	n+6

Figure 4: Encoding, transmitting and decoding/displaying a sequence of frames with the proposed method. The sequence contains a scene cut between the frames n-1 and *n*.

Figure 5: The top strip shows a sample frame from each of the files forming *std* and *std100*; the bottom strips show sample frames from each commercial in the sequence *commercials*.

sequence of skipped frames in a TMN-8 encoder. This has the advantage of minimizing both the jerk and the distance to the next encoded frame.

This approach is easy to implement and reduces the computation to essentially at most two encodings of every frame. Fig.4 shows how this simple strategy works on the same sequence of frames described in Fig.2. As it can be seen from Fig.4, the frame ***n+3*** is encoded instead of the frame ***n*** and because ***n+4*** is predicted from the closer frame ***n+3***, no additional skipping is in general necessary. Fig.3 depicts the resulting improved encoding of the scene cut.

Even with this simplified heuristic, our experiments show that it is possible to improve consistently the coding quality and

Sequence	N	Bit rate (Kb/s)	Skipped Frames		Gain in bit/PSNR (per frame)		
			TMN-8	***Mod***	***0.5 sec. Cuts (15 frames)***	***Scenes***	***Both***
commercials	4250	31.55	1802	1831	10.14 (15.4%)	-5.01	1.27
commercials1	5900	31.42	2221	2135	12.15 (19.0%)	-5.53	0.12
commercials2	4100	31.77	1106	1059	5.21 (10.0%)	-1.64	0.06
commercials3	5000	31.28	1127	1102	6.18 (12.9%)	-1.50	0.04
commercials4	4800	31.72	1955	1862	11.23 (17.1%)	-5.33	0.18
commercials5	3100	31.87	1423	1392	5.52 (7.41%)	-6.41	-0.02
std	4000	31.78	211	210	7.10 (30.5%)	-0.08	0.01
std100	4000	31.76	407	405	3.80 (14.3%)	-0.23	0.03

Table I: Experimental results – 32Kb/s. Gain in bit/PSNR in proximity of scene cuts (simplified method).

to reduce the number of skipped frames. Because the rate control strategy is unable to achieve the target bit rate with a sufficient precision, we have measured the performance as a *ratio* between the encoded size of a frame (in bits) and the PSNR achieved on the Y plane (that provides a normalization factor). Since the purpose of our approach is to improve the quality in proximity of scene cuts, in addition to the total number of skipped frames, we report the gain (using the ratio measure) for the data partitioned in:

- *cuts*: The first 15 frames after each scene cut (1/2 second).
- *scenes*: All other frames that are not part of the cuts.

Tables I and II report on experiments we made by encoding the test sequences both at 32Kb/s and 64Kb/s. The results show that an encoder that uses our algorithm, skips less frames than a TMN-8 encoder and consistently improves the quality of the scene cuts. Scenes are essentially left unchanged. During the experiments, we also observed small gains in the PSNR.

CURRENT RESEARCH

We are working on the implementation of both greedy strategies and variations of the dynamic programming optimization. The theoretical research includes the detailed analysis of the consequences of eliminating the assumption we made for the algorithm.

Sequence	N	Bit rate (Kb/s)	Skipped Frames		Gain in bit/PSNR (per frame)		
			TMN-8	***Mod***	***0.5 sec. Cuts (15 frames)***	***Scenes***	***Both***
commercials	4250	63.94	650	631	7.98 (9.47%)	-0.30	0.61
commercials1	5900	63.98	793	750	8.31 (9.31%)	-0.63	0.15
commercials2	4100	64.00	303	282	7.42 (9.46%)	-0.19	0.06
commercials3	5000	63.93	299	294	4.98 (6.54%)	-0.34	0.09
commercials4	4800	63.99	628	600	6.58 (7.61%)	-0.73	0.08
commercials5	3100	64.00	504	493	3.40 (4.28%)	-0.19	-0.01
std	4000	64.00	35	34	1.37 (3.70%)	-0.08	0.03
std100	4000	64.00	109	110	0.67 (1.64%)	0.18	0.24

Table II: Experimental results – 64Kb/s. Gain in bit/PSNR in proximity of scene cuts (simplified method).

ACKNOWLEDGMENT:

We thank M. Cohn and F. Rizzo for many helpful discussions.

BIBLIOGRAPHY

[1] G. Côté, B. Erol, M. Gallant and F. Kossentini, "H.263+: Video Coding at Low Bit Rates", IEEE Trans. on Circuits and Systems for Video Technology, Vol.8, No.7, Nov.1998.

[2] T. Gardos, "Video Codec Test Model, Near-Term, Version 8", Doc. ITU-T/SG16/Q15-D65d1.

[3] S. Wenger, G. Côté, M. Gallant and F. Kossentini, "Video Codec Test Model, Near-Term, Version 11 (TMN-11) Rev.2", Doc. ITU-T/SG16/Q15-G-16 R.2.

[4] D. Kozen, Y. Minsky and B. Smith, "Efficient Algorithm for Optimal Video Transmission", IEEE Data Compression Conference, DCC 1998.

[5] J. Lee and B.W. Dickinson, "Joint Optimization of Frame Type Selection and Bit Allocation for MPEG Video Encoders", ICIP 94, Vol.2.

[6] G. Motta, J.A. Storer and B. Carpentieri, "Improving Scene Cut Quality for Real-Time Video Decoding", Proceedings of the Data Compression Conference (DCC 2000), Snowbird (UH), Mar. 2000.

[7] J.I. Ronda, M. Eckert, S. Rieke, F. Jaureguizar and A. Pacheco, "Advanced Rate Control for MPEG-4 Coders", Proc. of VCIP'98, San Jose', Jan. 1998.

[8] H. Song, J. Kim and C.C. Jay Kuo, "Real-Time Motion-Based Frame Rate Control Algorithm for H.263+", Doc: ITU-T/SG16/Q15-F-14.

[9] G.J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression", Draft for submission to IEEE Signal Processing Magazine, Nov. 1998 issue.

[10] T. Wiegand, M. Lightstone, D. Mukherjee, T. George Campbell and S.K. Mitra, "Rate-Distortion Optimal Model Selection for Very Low Bit Rate Video Coding and the Emerging H.263 Standard", IEEE Trans. on Circuits and Systems for Video Technology.