RATE CONTROL FOR GOP-LEVEL RATE ADAPTATION IN H.264 VIDEO CODING

Fabio De Vito[1,2], Tanir Ozcelebi[2], Reha Civanlar[2], A. Murat Tekalp[2,3], Juan Carlos De Martin[4]
[1]Dip. Automatica e Informatica/[4]IEIIT-CNR, Politecnico di Torino, 10129, Torino, Italy
[2]Department of Electrical Engineering, Koc University, Sariyer, 34450, Istanbul, Turkey
[3]Dept. of Electrical and Computer Eng., Un. of Rochester, Rochester, NY 14627-0126, USA
Email: {fdevito,tozcelebi,rcivanlar,mtekalp}@ku.edu.tr, demartin@polito.it

***ABSTRACT*— It may be desirable to adapt the coding bitrate according to both bandwidth availability and user preferences in video transmission over packet networks. Classical approaches for rate adaptation has been bitstream switching, which requires storing several pre-coded versions of the same video at different bitrates, or layered (scalable) video coding, which has coding efficiency and/or complexity penalties. In this paper we propose a new GOP-level rate adaptation scheme for a single stream, variable target bitrate H.264 encoder, which allows each group of pictures (GOP) to be encoded at a specified bitrate, using a dynamically updated table to select the starting quantization parameter for each GOP. We first compare the performance of the standard H.264 rate control algorithm with the proposed one in the case of constant target bitrate. Then, we present results on how close the new technique can track a specified per-GOP target bitrate schedule. Results show that the proposed approach can obtain the desired target rates with less than 5% error.** [1]

I. INTRODUCTION

In recent years, technological improvements have created the basis for the development of several new network applications, among which the most challenging one seems to be the access to media content for mobile users. In the case of multimedia transmission, which requires a timely delivery of information, changes in bandwidth occurring in mobile communications can affect the quality of the received information. If insufficient bandwidth is available, the playout could be poor due to packet losses, and the stream should be recoded at lower bitrate according to the new condition. On the other hand, if more bandwidth is available, the media content can be safely coded at higher quality. Network conditions are not the only factors which can require a modification in the desired bitrate: if the user accesses the contents paying on a per-byte basis, he can desire to receive a low-quality stream when low-importance contents are played, and require better coding when the stream contains sequences he considers as high-importance.

In case of video transmission over packet data networks, modern video codecs like H.264 [1] can achieve very low bitrate coding of sequences. The use of such coders allows the distribution of video contents also on low-bandwidth links, like the ones involved in wireless communications. Unfortunately, the radio link suffers of wide bandwidth oscillations and, in particular at very low bitrates, concealment algorithms do not guarantee a satisfactory recovery of the eventually lost information, so degrading the perceived video quality. In those cases, it is preferable to recode the stream at lower quality, so avoiding losses, instead of keep constant coding bitrate and relying on concealment techniques.

For the above reasons, adaptivity of video streams has been extensively studied in recent years. Mainly, the adaptation is achieved by storing several versions of the same content, encoded at different bitrates, and then switching among the streams as required by the network condition. This approach is particularly suitable for video archives, where the access is *on demand* and there is enough time to perform multiple encodings. The server can then transmit at the appropriate bitrate according to the network available bandwidth.

Another well-known approach to the same problem is layered video coding. This consists mainly in coding a *base layer* at low quality, and then adding one or several *enhancement layers*. The receivers are able to decode the base layer independently, and the enhancement layers can refine the video quality if received: the bitrate adaptivity is obtained by changing the number of enhancement layers transmitted [2]. An extreme case of this technique is *fine grain scalability*, which allows a very small error in bitrate adaptation.

Both the above family approaches demonstrated to be useful in achieving good network utilization and high video quality [3], [4]. On the other hand, some of them can only achieve bitrates in a limited set, usually decided at coding time; for example, it is limited to the rates of the pre-coded versions in case of simultaneous storage, while it is given by the number of enhancement layers in case of layered video coding.

In literature it is possible to find several papers overviewing the above concepts (e.g.[5]), and extending them with techniques like frame skipping or coefficient dropping [6], [7], [8].

In this work we will propose some modifications to

[1]This work is supported both by 3DTV-NoE funded by the European Commission 6^{th} Framework Information Society Technology Programme, and by Centro Supercalcolo Piemonte (CSP) - Torino, Italy.

the standard H.264 bitrate control routine, in order to make the stream change its bitrate to an arbitrarily chosen value *on the fly*. With respect to the above mentioned approaches, this solution will output one single non-layered stream, containing GOPs potentially coded each one almost exactly at the required bitrate, so increasing the granularity and avoiding simultaneous storage of pre-coded information.

The paper is organized as follows. In Section II we describe the background to the problem, and in Section III we describe the modifications implemented in the encoder. We present the coding results in Section IV, and draw the conclusions in Section V.

II. BACKGROUND

In this work, we will refer to functionalities of the JM 9.3 H.264 standard codec. This modern video codec allows achievement of very low bitrates. The reference software implements a rate control algorithm, which requires as input the target value and a starting quantization parameter for the first I-frame of the sequence. The output will be a constant bitrate (CBR) sequence, usually it will converge to after a period of one or two GOPs ant it will fluctuate around the selected value with reasonable approximation for the remaining part of the sequence. This standard rate control is useful to create a single sequence at a given bitrate. It is not possible to change the bitrate during the coding operation, and even if it was possible, the convergence time of a couple of GOPs would allow only a very low frequency in changes to get meaningful results.

Our goal will be modifying this rate control system to produce a single stream, encoded according to a per-GOP bitrate pattern, communicated either by some bandwidth estimation tool to adapt to changed link conditions, or by the user who desires lower or higher quality according to his own preferences and needs. In theory, this change in the desired bitrate can occur at *each* single GOP boundary, and so the convergence speed will become a key issue, in order to reach the target value before the following switch is required.

Changes in bitrate can obviously occur only when an I-frame is reached. As it is implemented in the reference codec, the GOP length is fixed by indicating the periodicity of I-frames and the number of B-frames in a run. If the number of frames within a GOP is small and the frame rate high enough, this fixed behavior does not represent a problem, since there can be several GOPs starting in a second of video and the granularity of the switch points for most of the applications can be considered satisfactory. Unfortunately, at very low bit rates, the frame rate is usually reduced to values around 15 fps or lower, and the GOP length is usually high in order to mitigate the presence of I-frames, which require more bits to be coded. If we desire to switch immediately, then also the selection of I-frame position should be changed. The constraint on fixed GOP structure should be relaxed to gain more flexibility in this case. It is possible to modify the length of each GOP dynamically without affecting the decodability of the sequence, since the decoder is able to operate with any I-/P-/B-frame pattern, regardless of the structure of previous GOPs.

If the two above described functionalities are active jointly, it will be possible to remotely drive the encoder via a simple socket program, by communicating at which frame the switching should occur and to what value of bitrate.

In this work, we will focus our attention only to the bitrate control algorithm, and we do not show the effect of variable GOP sizes. Thus, we will refer to a 1-second GOPs, containing 30 frames when working at 30 fps and 15 frames when working at 15 fps, if not differently specified.

III. GOP-LEVEL RATE ADAPTATION SCHEME

Being the desired effect to switch between bitrates within the same sequences, more flexibility in the definition of this parameter is required: in our implementation, the codec reads from a file the new value of the bitrate at each beginning of a GOP (I-frame). This approach is suitable for a remote-driving of the codec by simply employing a socket communication system whose task is to write the new desired bitrate value in this file. In the same way, the encoder can receive the desired length of the GOP being coded, so tuning also the position of I-frames. Once the new desired bitrate is read and stored in memory, the standard rate-control routine will converge to this new value.

It is still necessary to speed up the convergence to the specified bitrate. It is possible to have one bitrate switching request for each GOP and the standard encoder could not be able to converge to the new value in a so short time. The codec stores internally some statistics on previous GOPs, which become useless, and meaningless, when the target rate is modified; those statistics need to be tuned accordingly.

To ensure better convergence, we propose to recompute the initial quantization parameter for each GOP as described below. We implemented a static initial table, showing an approximate mapping between the quantization parameters and the bits per pixel obtained. Every time a new I-frame is being coded, the desired bitrate is read and the target bits-per-pixel (*bpp*) indicator is computed according to the frame size and frame rate. The initial quantization parameter is then chosen from the table as the one ensuring the closer bpp indicator. Every time a GOP terminates, and right before starting the following I-frame, the bpp obtained for the last GOP is stored in the table together with its average quantization parameter, so updating the starting *static* values at each step to better fit over the sequence characteristics.

This approach will produce better convergence also for the first GOP in the sequence, with respect to the standard

TABLE I

INITIAL TABLE, SOME QUANTIZATION PARAMETERS *Q* AND BITS PER PIXEL *bpp*; FOR REFERENCE, THE BITRATE OBTAINED WITH A QCIF FRAME SIZE AT 25 FPS IS ALSO SHOWN.

Q	bpp	bitrate for qcif at 25 fps
30	0.220909	139967
35	0.132545	83980
40	0.079527	50388
45	0.047716	30232
50	0.028630	18139

TABLE II

COMPARISON BETWEEN BITRATE ACHIEVEMENT OF STANDARD H.264 JM 9.3 CODEC AND THE MODIFIED VERSION; THE FIRST GOP IS EXCLUDED.

Sequence	Error (%) between target and actual bitrate					
	32 kbps		64 kbps		128 kbps	
	Std	Mod.	Std.	Mod.	Std.	Mod.
Foreman	1.97	0.94	0.54	0.66	0.54	0.55
Tempete	0.70	0.85	0.22	0.41	0.43	0.25
Paris	2.61	1.20	1.84	0.87	1.16	0.29
News	1.44	1.02	0.64	0.26	0.33	0.38
Mobile	2.85	1.20	1.28	0.62	0.54	0.32

implementation, because the initial quantization parameter is no longer required as input but internally determined; moreover, it will continue producing better results during encoding due to the dynamic update. The initial table does not need to provide the exact matching between bpp and quantization parameter Q because if the GOP contains a sufficient number of frames, the rate control algorithm will converge in any case to the desired target, after some inter-GOP oscillations.

The bpp values chosen for the initial table are shown in Table I, where we present only the low-bitrate portion of the table, $Q \in [30 - 50]$. We include also the case of 128 kbps to show that this approach works for a wide range of values.

The values shown in Table I have been obtained by setting $bpp = 4.27$ for $Q = 0$; this value has been chosen after a study of coding statistics for different sequences. All the other values in the initial table are obtained recursively by using Formula (1):

$$bpp_{i+1} = bpp_i \cdot 0.9 \qquad (1)$$

In Table I, bits per pixel are intentionally computed ignoring the presence of chrominance components. This will cause the algorithm to choose a smaller starting quantization parameter, so letting a better quality for the I-frame. The bitrate convergence routine will then take care of quantizing more the following frames to match the bitrate. Good results have been obtained even with small number of frames per GOP.

To show convergence accuracy, we report the average error obtained by the standard H.264 JM 9.3 encoder and the modified version in Table II, for a constant bitrate encoding of the sequences, at three different values.

This table shows the average percent error for different sequences encoded at different bitrates, using both the standard and the modified encoder. GOPs contain 30 frames each. The first GOP is excluded from the computation because the error obtained with the standard encoder is excessively high, even if, to speed up convergence, the quantization parameters for the first frame in the case of standard encoder have been set to 30, 37 and 45 respectively for the cases of 32, 64 and 128 kbps, which are values very close to the ones reported in the initial

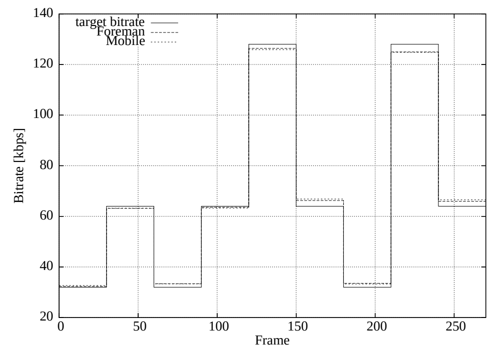

Fig. 1. Bitrate obtained for two sequences at 30 frames per second, compared with the target bitrate behavior.

table of the modified coder (see Table I for comparison). Results show that the new encoder can achieve at least the same precision of the standard one, outperforming it for the majority of sequences and bitrates considered. Furthermore, we obtain the desired bitrate avoiding the two-GOP convergence time.

IV. RESULTS

The described modified H.264 encoder has been employed to code different video sequences over different bitrate patterns, which have been chosen, as limit case, to force a switching at every GOP boundary. In this section we present coding results for two sequences at 30 frames per second and for four sequences at 15 fps.

Figure 1 shows the required bitrate pattern and the output that is obtained for the two sequences *foreman* and *mobile*. This pattern contains target values in the set $\{32, 64, 128\}$kbps. This is again a limit setting, since usually sequences at low bitrate are coded using sensibly less than 30 frames per second to gain better PSNR.

The plots result close to the reference, so we show the percent error in Table III for each one of the nine coded GOPs of the two sequences. This error results to be never higher than 5% and, as a consequence of imposing several changes within the sequence, these values are higher than the ones shown for the modified coder in Table II.

Better results can be obtained coding sequences at lower bitrates and at lower frames per second. We encoded four

TABLE III

ERROR IN ACHIEVING TARGET BITRATES AS SHOWN IN FIGURE 1, FOR TWO SEQUENCES AT 30 FRAMES PER SECOND.

GOP	Error (%)	
	Foreman	Mobile
1	0.97	2.05
2	1.34	1.21
3	4.28	4.10
4	0.66	1.14
5	1.28	1.69
6	3.50	4.45
7	4.78	3.90
8	2.38	2.51
9	3.00	3.94

TABLE IV

ERROR IN ACHIEVING TARGET BITRATES AS SHOWN IN FIGURE 2 FOR FOUR SEQUENCES AT 15 FRAMES PER SECOND.

GOP	Error (%)			
	Foreman	Tempete	Paris	News
1	1.47	4.03	2.91	0.06
2	0.45	0.69	0.08	1.44
3	0.70	0.03	1.60	1.47
4	0.69	0.14	0.29	1.60
5	0.75	0.57	0.46	0.05
6	1.37	0.18	1.24	0.62
7	2.07	1.75	1.19	0.44
8	0.77	1.71	0.80	1.79
9	0.24		1.15	0.08

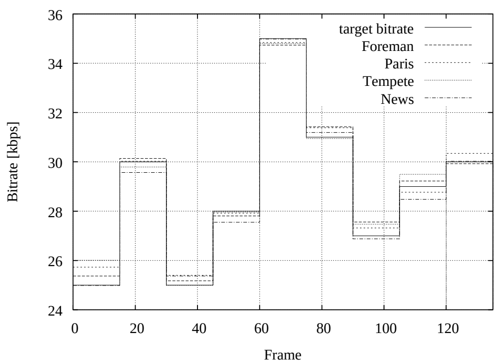

Fig. 2. Bitrate obtained for four sequences at 15 frames per second, compared with the target bitrate behavior.

sequences at 15 fps, requiring a very low bitrate value. In this scenario we set the required rate to switch between 25 and 35 kbps. The resulting behavior for each sequence is reported in Figure 2. Again, the performance of the rate control routine follows the target pattern. The percent errors placed in Table IV are smaller than the ones reported in Table III and comparable with the ones of Table II, due to the smaller difference in the values we switch among. Sequence *Tempete* is shorter and contains only 8 GOPs.

These results show that the proposed implementation can achieve the effect of encoding portions of a single sequence following a specified behavior from close up.

V. CONCLUSIONS

In this paper, we proposed some simple modifications to the standard H.264 JM codec, version 9.3, in order to allow on-the-fly switching of bitrate within the same sequence.

This implementation is mainly intended for a coding driven by the network condition or by the user preferences, avoiding the use of different pre-encoded and stored sequences. This approach is particularly suitable for real-time communications, provided that the feedback from the network or the user is immediate. We proposed the employment of an initially static table to select a suitable quantization parameter for each new bitrate request. This table has been obtained by performing different encodings of several sequences, and then observing the results. The proposed initial table can be computed by means of a recursive formula. During encoding, statistics on the already coded GOPs are used to update the table and so adjusting the values for the particular sequence content.

We demonstrated that this approach can achieve constant bitrate (CBR) coding with a higher precision than the standard encoder, being usually its error lower than 1%. Moreover, if information on per-GOP bitrate pattern is provided, the modified encoder can safely switch between the desired bitrates, fastly converging to the indicated value, so adapting nearly immediately to the new network available bandwidth or user preference with an error which can arrive up to 5% if the gaps between values we switch among are wide and the frames per second value is 30. Better results are obtained with lower fps and closer bitrate levels. This characteristic makes this implementation suitable for wireless communications, where low bitrates and fast network adaptivity are mandatory constraints for achieving satisfactory performance.

REFERENCES

[1] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, "Advanced video coding for generic audiovisual services," *ITU-T*, May 2003.

[2] T. Ahmed, A. Mehaoua, R. Boutaba, and Y. Iraqi, "Adaptive packet video streaming over ip networks: a cross-layer approach," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 385–401, February 2005.

[3] S.F. Chang and A. Vetro, "Video adaptation: Concepts, technologies and open issues," *IEEProceedings of the IEEE*, vol. 93, no. 1, pp. 148–158, January 2005.

[4] I. Radulovic, P. Frossard, and O. Verscheure, "Adaptive video streaming in lossy networks: versions or layers?," in *Proc. IEEE Int. Conf. on Multimedia & Expo*, Taipei, Taiwan, June 2004, pp. 1915–1918.

[5] B. Li and J. Liu, "Multirate video multicast over the internet: An overview," *IEEE Network, Special Issue on Multicasting: An Enabling Technology*, vol. 17, no. 1, pp. 24–29, January 2003.

[6] A. Ortega, Ed., *Variable Bit-Rate Video Coding, in Compressed Video over Networks*, pp. 343–382, M.-T. Sun and A. R. Reibman, Eds, New York, NY, USA, 2000.

[7] J.G. Kim, Y. Wang, and S.F. Chang, "Content-adaptive utility-based video adaptation," in *Proc. IEEE Int. Conf. on Multimedia & Expo*, Baltimore, Maryland, July 2003.

[8] C.C.J. Kuo H. Song, "Rate control for low-bit-rate video via variable-encoding frame rates," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 4, pp. 512–521, April 2001.