AN INTRA-FRAME RATE CONTROL ALGORITHM FOR ULTRA LOW DELAY H.264/AVC CODING

Yun-Gu Lee and Byung Cheol Song

Digital Media R&D Center
Digital Media Business
Samsung Electronics Co., Ltd
E-mail: yungu96.lee@samsung.com, bcsong@samsung.com

ABSTRACT

In this paper, we present an intra-fame rate control algorithm for ultra low delay H.264/AVC coding. In real time video coding, all the macro-blocks within a current frame may be unavailable before encoding them. Hence the proposed scheme predicts relative complexity of the current block from complexity of available macro-blocks within previous and current frames. Then, the algorithm allocates bits to each macro-block considering the relative complexity between the macro-block and the current frame. Quantization parameter of each macro-block is obtained by comparing the generated and allocated bits and is refined for improving coding performance. The required buffer size is only around one-third of average bits required for encoding a single frame. Simulation results show that the proposed algorithm can prevent the buffer from overflow and underflow while it provides better coding performance.

Index Terms— rate control, buffer overflow, buffer underflow, low delay

1. INTRODUCTION

The number of bits for encoding a video sequence varies with time even if the quality of a decoded video is similar. It is because complexity of each frame differs from other frames in the sequence. For transmitting a video into channel, a rate control algorithm is necessary in an encoder for meeting a channel rate by controlling the number of generated bits. Hence, the rate control plays a key role in a video encoder, and it has been a challenging research topic. Rate control schemes such as TM5 [1] for MPEG-2, TMN8 [2] for H.263, VM-8 [3] for MPEG-4, and JM [4] for H.264 have been developed for international video coding standards.

In an encoder and a decoder, there are buffers to temporally store encoded bits, which cause an end-to-end delay. A buffer size should be inevitably small in a real time video coding that requires small end-to-end delay. Hence, the rate control is more difficult in a low delay video streaming since it should avoid buffer overflow and underflow. Many rate control algorithms for the low delay environment have been proposed in [5]-[7]. Usually, because the complexity between successive frames significantly changes at scene boundaries, an encoder buffer may suffer from overflow. For a smaller buffer size i.e., lower end-to-end delay, this may be a more critical issue. Also, even though the number of bits for an intra frame is greater than that for an inter frame, most of the previous rate control algorithms focus on inter frames. On the other hand, intra frame-only coding scheme for professional applications has been standardized as an H.264 profile [8]. But, the previous schemes are not appropriate for only intra-frame coding requiring very low end-to-end delay.

Recently, Jing and Chau have developed a rate control algorithm for intra frame coding [9]. The scheme establishes an intra frame rate-quantization estimation model by using a gradient-based picture complexity measure. However, the algorithm requires pre-processing to measure the complexity of a current frame to be encoded, and it causes an additional end-to-end delay in real-time video coding. Also, the buffer size for the algorithm [5]-[8] generally is equal to or larger than bits generated for encoding a single frame. Considering both the encoder and the decoder, the total end-to-end delay amounts to more than two frames, i.e., over 30ms. If the wireless channel delay is added, the end-to-end delay will significantly increase. Since such end-to-end delay is large enough for humans to perceive, this algorithm is not also suitable for some applications requiring very low end-to-end delay for real-time interaction, e.g. the connection between TV and game consoles.

In this paper, we propose an intra-frame rate control algorithm for ultra low delay H.264/AVC streaming, where the total end-to-end delay between an encoder and a decoder is less than one frame. We also assume that an encoder get started as soon as any MB row is available. Firstly, the scheme predicts complexity of a current frame by examining complexity of the current MB row and previous frame in advance. Secondly, it allocates the proper bits to each MB based on the prior complexity knowledge. Since it predicts the proper bits, it prevents buffer overflow or underflow successfully. Therefore, the algorithm requires extremely small buffer size, while it is robust to scene changes.

The rest of the paper is organized as follows. Section II shows background information about gradient-based complexity. We will describe the proposed algorithm in Section III. Section IV provides the experimental results. Finally, we conclude in Section V.

2. COMPLEXITY MEASURE

Here, we define complexity of a MB, a MB row, and a frame which is similar to [9]. The complexity of a MB at a position of (i, j) in the kth frame is defined as

$$G_{\mathrm{MB}}(i,j,k)=\sum_{n=0}^{7}\sum_{m=0}^{7}\left(|I_{x,y,k}-I_{x-1,y,k}|+|I_{x,y,k}-I_{x,y-1,k}|\right), \quad \left(x=16i+2m,\ y=16j+2n,\ 0\le i<\frac{M}{16},\ 0\le j<\frac{N}{16}\right). \quad (1)$$

Here, M and N are the horizontal and vertical dimension of a sequence, respectively. $I_{x,y,k}$ denotes the luminance of a pixel at a position of (x, y) in the kth frame. To reduce complexity burden in an encoder, gradient values of pixels are sub-sampled. The complexity of the jth MB row in the kth frame is obtained by adding MB complexity as follows.

$$G_{\mathrm{MB,R}}(j,k)=\sum_{i=0}^{M/16-1}G_{\mathrm{MB}}(i,j,k)\cdot \quad (2)$$

Also, the complexity of the kth frame is as

$$G_{\mathrm{F}}(k)=\sum_{j=0}^{N/16-1}G_{\mathrm{MB,R}}(j,k) \quad (3)$$

Accumulated gradient of MB row or $G_{\mathrm{ACC}}(j, k)$ is defined as

$$G_{\mathrm{ACC}}(j,k)=\sum_{m=0}^{j}G_{\mathrm{MB,R}}(m,k)\cdot \quad (4)$$

3. PROPOSED INTRA-FRAME RATE CONTROL

3.1. Frame level

The complexity of video varies along time. In order to preserve video quality, the amount of bits to be allocated for each frame should be different depending on its complexity. However, complexities of the present and future frames cannot be known before encoding them. Also, if the size of buffer is extremely small, it is hard to allocate different bits to each frame. Therefore, the proposed algorithm allocates bits for the target bit of a frame (or b_{F}) by considering a target bit-rate (or B_{T}) and buffer status.

$$b_{\mathrm{F}} = B_{\mathrm{T}} / Fr + (\mathrm{Buf}_{\mathrm{SIZE}} / 2 - \mathrm{Buf}_{\mathrm{USED}}), \quad (5)$$

where $\mathrm{Buf}_{\mathrm{SIZE}}$ and $\mathrm{Buf}_{\mathrm{USED}}$ represent the size of buffer and the used buffer. Here, Fr denotes the frame rate of a video.

3.2. MB row level

While the scheme allocates the target bits of a frame without considering complexity of a current frame, it reflects complexity of a MB row in determining the target bits of a MB row. If a MB row is more complex than ordinary MB rows, it allocates more bits than average bits, and vise versa. Let $b_{\mathrm{MB,R}}(j)$ be the allocated bits for the jth MB row. Then,

$$b_{\mathrm{MB,R}}(j) = G_{\mathrm{MB,R}}(j, k) \times b_{\mathrm{F}} / G_{\mathrm{F}}(k). \quad (6)$$

In real-time video streaming, it is hard to examine the whole frame to measure complexity of a current frame, and $G_{\mathrm{F}}(k)$ may not be known before encoding the frame. In order to predict $G_{\mathrm{F}}(k)$, the scheme examines similarity between MB rows of a previous frame and a current one. According to the result, the different bit allocation scheme is applied. If a MB row to be encoded is similar to that in a previous frame at the corresponding point, the scheme utilizes $G_{\mathrm{F}}(k-1)$ instead of $G_{\mathrm{F}}(k)$ in (6). Thereby it can allocate bits to the MB row according to its complexity. If it is not, the algorithm allocates the average bit as

$$b_{\mathrm{MB,R}}(j) = b_{\mathrm{F}} / (M/16). \quad (7)$$

The algorithm performs similarity check for every MB row independently. The similarity check for the jth MB row is performed by comparing $G_{\mathrm{ACC}}(j, k)$ with $G_{\mathrm{ACC}}(j, k-1)$. If $G_{\mathrm{ACC}}(j, k-1)/G_{\mathrm{ACC}}(j, k)$ is not within $(1-\mathrm{T}_1, 1+\mathrm{T}_1)$, the corresponding MB row is considered as '*new MB row*'. Otherwise the MB row is '*similar MB row*'. Here, T_1 is a threshold. Since the algorithm operates two modes according to the result of the similarity check, it makes the algorithm robust to scene changes. Also to check the similarity for every MB row prevents the buffer overflow at scene changes.

If the size of used buffer is greater than the given threshold T_2, the algorithm refines $b_{\mathrm{MB,R}}(j)$ for preventing the buffer from overflowing.

$$b^{R}_{\mathrm{MB,R}}(j)=\begin{cases} b_{\mathrm{MB,R}}(j), & \mathrm{Buf}_{\mathrm{USED}}<T_2 \\ b_{\mathrm{MB,R}}(j)\times\dfrac{\mathrm{Buf}_{\mathrm{SIZE}}-\mathrm{Buf}_{\mathrm{USED}}}{\mathrm{Buf}_{\mathrm{SIZE}}-T_2}, & \text{otherwise}\end{cases} \quad (8)$$

where $b^{\mathrm{R}}_{\mathrm{MB,R}}(j)$ represents the refined bits of $b_{\mathrm{MB,R}}(j)$.

33. MB level

The scheme first allocates bits for each MB, calculates QP, and finally refines QP. Similar to the bit allocation for a MB row level, if a MB has large gradient, the scheme allocates more bits to the MB than ordinary MBs, and vise versa. Let $b_{\mathrm{MB}}(i, j)$ be the bits allocated for a MB. Then,

$$b_{MB}(i, j) = b^{R}_{MB,R}(j) \times G_{MB}(i, j, k) \,/\, G_{MB\_ROW}(j, k). \quad (9)$$

To calculate QP, the algorithm compares the allocated bits with the generated ones. If the generated bits are larger than the allocated bits, QP will be increased, and vise versa. Since complexity of a MB is not perfectly proportional to the generated bits, the comparison results are fluctuated. If QP is often changed, it may degrade coding performance. To solve this problem, we compare their accumulated bits, which are $B'_{ACC,MB}(i, j, k)$ and $B_{ACC,MB}(i, j, k)$.

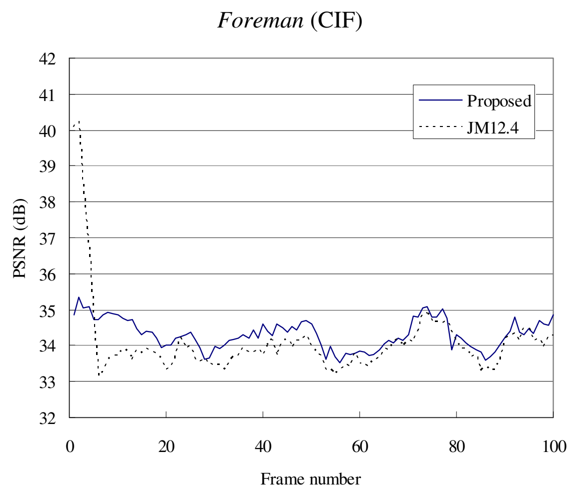
$$\begin{aligned} B'_{ACC,MB}(i,j,k) &= \sum_{m=0}^{i} b'_{MB}(m,j,k) \\ B_{ACC,MB}(i,j,k) &= \sum_{m=0}^{i} b_{MB}(m,j,k) \end{aligned} \quad (10)$$

Here, $b'_{MB}(i, j, k)$ represents the generated bits at (i, j) MB in the kth frame. Then, we define B_{DIFF} as $B'_{ACC,MB}(i, j, k) - B_{ACC,MB}(i, j, k)$. If $B_{DIFF} - B_{DIFF,P} < -T_3$, the scheme decreases QP by 1 (or if $B_{DIFF} - B_{DIFF,P} > T_3$, the scheme increase QP by 1.) and $B_{DIFF,P}$ is updated to B_{DIFF}. Here, T_3 is a tolerance to prevent QP from being updated. Note here that $B_{DIFF,P}$ is B_{DIFF} value at the last MB whose QP is increased or decreased.

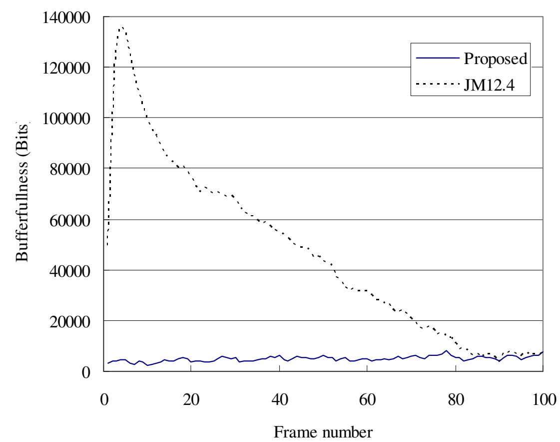
If complexity of a current frame is similar to that of a previous frame, the proper QP for the current frame also similar to QP for the previous frame. Since we already encode the previous frame, the proper QP for the current frame can be approximately estimated. If a MB is belongs to '*similar MB row*', the scheme refines above QP calculated for a MB. It is reasonable that although QP predicted from the previous frame is not always the optimal one, the value is around the optimal QP for the current frame. We limit the calculated QP to $[QP_P - QP_T, QP_P + QP_T]$. Here, QP_T is a threshold, and QP_P is the QP value estimated from the previous frame. We simply set QP_P to an average QP in the previous frame for reducing computational complexity.

4. SIMULATION RESULTS

For evaluating the proposed algorithm, we implement the algorithm on the H.264 reference software JM12.4 [10]. For experiment, we set CABAC mode, and rate-distortion optimization is enabled. All the frames are intra-coded. Seven sequences are simulated including *mobile and calendar*, *foreman*, *coastgard*, *hall monitor*, *news*, *Stefan*, and "*foreman* and *mobile and calendar*" sequences with a size of CIF. We intentionally make "*foreman* and *mobile and calendar*" sequence for testing abrupt scene changes. In the sequence, 10 frames of *foreman* and *mobile and calendar* is alternately repeated. *News* and *Stefan* sequences consist of 90 frames and the other sequences 100 frames. A frame rate of sequences is 30Hz. In the proposed algorithm, T_1, T_2, T_3, and QP_T are set to 1/8, $3\times Buf_{SIZE}/10$, $b^{R}_{MB,R}(j)/(M/16)$, and 1, respectively. In this paper, since we assume a real time video streaming, we examine a buffer status for every MB to check whether the buffer is full or not. In the simulation, although the buffer is full, a frame is not skipped.



(a)



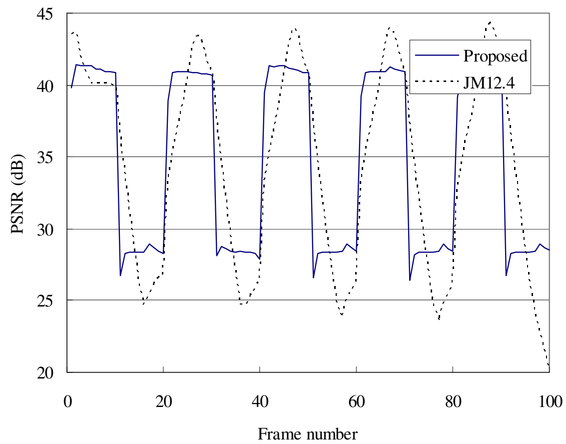
(b)

Figure 1. (a) PSNR and (b) buffer fullness comparisons for *foreman* sequence

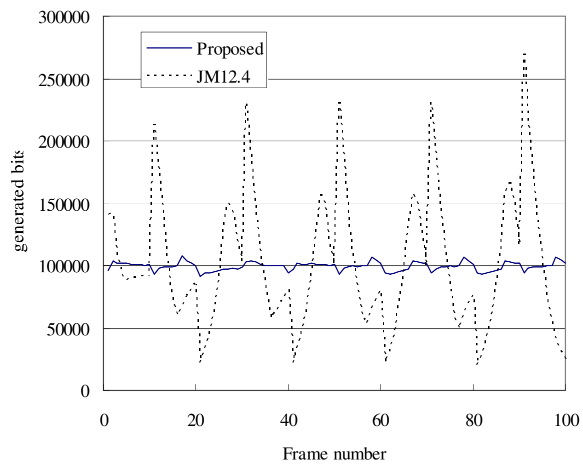
Figure 1 illustrates comparison results of PSNR and bufferfullness for sequence "*foreman*". Here, the bit-rate is set to 1Mbps and the size of a buffer is 10kbits, which is (a target bit-rate)/100. (All the frames are intra-coded.) It is noted that the buffer size is smaller than one-third of bits generated for encoding a single frame. As given in the figure, the proposed algorithm provides better performance than the JM12.4 rate control algorithm. At the beginning of a sequence, the JM12.4 scheme allocates many bits to a frame, and it causes buffer overflow. Hence, if a system can skip a frame, many frame skipping will be observed. Meanwhile, the proposed algorithm always meets buffer constraint.

Figure 2 shows comparison results of PSNR and generated bits for sequence "*foreman* and *mobile and calendar*". Here, the bit-rate is set to 3Mbps and the size of

Foreman and *mobile and calendar* (CIF)



(a)



(b)

Figure 2. (a) PSNR and (b) generated bits comparisons for *foreman* and *mobile and calander* sequence

a buffer is 30kbits. As shown in the fig 3(b), while the bits generated in JM12.4 are much fluctuated, the proposed algorithm provides constantly generated bits. It verifies that the proposed algorithm is robust to abrupt scene changes.

Table 1 represents simulation result for other sequences. In the experiments, the buffer sizes for the sequences are set to (a target bit-rate)/100 and 1.5×(a target bit-rate)/100 depending on sequences. As shown in the table, the proposed algorithm provides better performance.

5. CONCLUSION

In this paper, we propose an intra frame rate control algorithm for ultra low delay H.264/AVC coding. The method has been developed for controlling the generated bits in real time video streaming. By using this method, the buffer size can be reduced by up to (a target bit-rate)/(3×frame rate), and the end-to-end delay between an encoder and a decoder can be significantly reduced. Since the algorithm directly encodes without pre-processing of a current frame, an encoder get started as soon as any MB row is available. So, the algorithm does not require an external memory to store a frame in an encoder, and it does not produce additional delay. Experimental results show that the buffer status of the proposed algorithm is very stable while providing the best video quality.

6. REFERENCES

[1] *MPEG-2 Test Model 5*, Doc. ISO/IEC JTC1/SC29 WG11/93-400, Apr. 1993.

[2] J. R. Corbera and S. Lei, "Rate control in DCT video coding for low delay communication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 172-185, Feb. 1999.

[3] A. Vetro, H. Sun, and Y. Wang, "MPEG-4 rate control for multiple video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 186-199, Feb. 1999.

[4] Z. G. Li, F. Pan, K. P. Lim, G. Feng, X. Lin, and S. Rahardja, "Adaptive basic unit layer rate control for JVT," JVT of ISO/IEC MPEG&ITU-T VCEG, JVT-G012-rl.doc, Mar. 2003.

[5] M. Jiang and N. Ling, "Bit allocation scheme for low-delay H.264/AVC rate control," *Proc. IEEE International Conference on Image Processing*, pp. 2501-2504, Oct. 2006.

[6] N. Eiamjumrus and S. Aramvith, "Rate control scheme based on cauchy R-D optimization model for H.264/AVC under low delay constraint," *iih-msp*, pp. 205-210, 2006 International Conference on Intelligent Information Hiding and Multimedia, 2006.

[7] Y. Liu, Z. G. Li, and Y. Ch. Soh, "A novel rate control scheme for low delay video communication of H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, pp. 68-78, Jan. 2007.

[8] S. Wittmann, S. Kondo, and H. Saito, "*Intra-only 4:4:4 Profile for H.264/AVC FRExt*," JVT of ISO/IEC MPEG&ITU-T VCEG, JVT-Q086.doc. Oct. 2005.

[9] X. Jin and L.P. Chau, "A novel intra-rate estimation method for H.264 rate control," *Proc. ISCAS 2006*, pp. 5019-5022, Greece, May 2006.

[10] JM12.4, http://iphome.hhi.de/suehring/tml/download/

Table 1. Performance comparison between proposed and JM12.4 method

Sequence	Bit-rate (kbps)	Buffer size (kbits)	Method	PSNR	Actual rate
mobile	**4000**	**10**	**Proposed**	**30.63**	**4007**
			JM12.4	**29.25**	**3990**
foreman	**1000**	**10**	**Proposed**	**34.21**	**982**
			JM12.4	**34.07**	**999.9**
coastguard	**3000**	**30**	**Proposed**	**35.62**	**2986**
			JM12.4	**35.4**	**2994**
hall monitor	**1000**	**15**	**Proposed**	**35.01**	**1003**
			JM12.4	**34.73**	**1000**
news	**1000**	**15**	**Proposed**	**33.62**	**1005**
			JM12.4	**33.39**	**1001**
stefan	**3000**	**45**	**Proposed**	**33.72**	**2960**
			JM12.4	**33.44**	**2994**