Channel-Adaptive Rate Control for Low Delay Video Coding

Yun-Gu Lee

Department of Computer Science and Engineering, Kwangwoon University / 447-1, Wolgye-Dong, Nowon-Gu, 139-701 Seoul, Korea harmony96@gmail.com

Received September 2, 2016; Accepted October 4, 2016; Published October 30, 2016

** Regular Paper*

***Abstract*:** This paper presents a channel-adaptive rate control algorithm for low delay video coding. The main goal of the proposed method is to adaptively use the unknown available channel bandwidth while reducing the end-to-end delay between encoder and decoder. The key idea of the proposed algorithm is for the status of the encoder buffer to indirectly reflect the mismatch between the available channel bandwidth and the generated bitrate. Hence, the proposed method fully utilizes the unknown available channel bandwidth by monitoring the encoder buffer status. Simulation results show that although the target bitrate mismatches the available channel bandwidth, the encoder efficiently adapts the given available bandwidth to improve the peak signal-to-noise ratio.

***Keywords*:** Channel-adaptive rate control, Video coding, Low latency

1. Introduction

The Wi-Fi Alliance recently announced Miracast, which is a peer-to-peer wireless screencasting standard formed via Wi-Fi Direct connections [1]. This technology enables wirelessly streamed audiovisual data from one device to other devices over Wi-Fi. Nowadays, such real-time wireless video streaming scenarios have become popular. The channel bandwidth over Wi-Fi is not big enough to stream uncompressed high-quality videos in real time. Hence, video compression techniques like H.264 [2] are required to reduce the size of video data for real-time video streaming over Wi-Fi.

Due to video compression, streaming systems must inevitably consider the additional latency between encoder and decoder. That latency includes the processing time in both encoder and decoder, as well as buffering time and transmission time. Latency is a critical issue for applications requiring a fast response, such as a digital television wirelessly connected to a game console. Another interesting issue is the unique environment that the above streaming systems have. The above real-time video streaming scenarios usually assume that there is a 1:1 dedicated connection between the sender and receiver within a small area over Wi-Fi. Hence, the streaming service may allow full use of the available bandwidth in order to provide the user with the best video quality. Rate control for video coding can play a key role in satisfying the above requirements.

Although conventional rate control algorithms, such as Joint Model (JM) [3] in H.264, work well for conventional services, they are unfortunately not the best for the above applications requiring low latency. For example, the basic unit in a hypothetical reference decoder for H.264/AVC is a frame. Several frames are required for buffering in the decoder. There have been several studies on low delay video coding [4-7]. However, the latency in the existing work is over one frame or across several frames [8]. Previous work [9] significantly reduced latency by using temporal and spatial information. The rate control algorithm was also verified in a real-time system by applying it to the video CODEC chip [10, 11]. However, that previous work only dealt with intra-frame coding.

Since the available channel bandwidth over Wi-Fi varies, a fixed target bitrate is not desirable for low latency video streaming in real time with a 1:1 dedicated connection between sender and receiver. To fully use the available channel bandwidth, the target bitrate needs to adaptively change according to the available bandwidth. Here, for low latency video coding, it is necessary to precisely predict the available channel bandwidth with only a short delay. If the predicted bandwidth is not well matched to the real one at any moment due to inaccurate or delayed prediction, the encoder buffer may easily overflow or underflow. Hence, the above conventional algorithms may not be suitable for the above video streaming scenarios. Therefore, this paper proposes a channel-adaptive rate control algorithm for low delay video coding. The basic idea of the proposed method is that the status of the encoder buffer indirectly reflects the mismatch between the generated bitrate and the unknown available channel bandwidth. By monitoring the encoder buffer status, the proposed method can consider by how much they are mismatched.

The rest of the paper is organized as follows. Section 2 introduces the proposed algorithm in detail. The experimental results are given in Section 3. Finally, Section 4 contains the concluding remarks.

2. Proposed Method

The basic structure of the proposed method is the same as in previous work [9]. It consists of three parts: the frame, macroblock row, and macroblock-level controls. By assuming that the target bitrate is the same as the available channel bandwidth, the proposed method first allocates a bit budget to the frame, macroblock rows, and macroblocks. The difference between the proposed and previous methods for allocating the bit budget is a complexity measure to support rate control of inter frames. Then, the macroblock level control considers the mismatch between the generated bitrate and the available channel bandwidth by monitoring the status of the encoder buffer, and increases (or decreases) the bit budget according to the mismatch between them. Details are given in the following subsections.

2.1 Modified Complexity Measure

Previous work allocated the proper bit budget to each macroblock row and each macroblock based on a gradient-based complexity. Since inter coding adopts a motion compensation technique, the gradient-based complexity is not appropriate to predict the generated bits in inter pictures. Hence, the proposed algorithm employs the energy of the residual block, which is the difference block between the original and predicted blocks or the input of quantization, as a complexity measure. Here, the energy is the sum of absolute values of the residual block. In intra and inter block modes, the predicted blocks represent intra-predicted and motion compensated blocks, respectively.

Let $C_{MB}(i,j,k)$ be the complexity of the macroblock at position (i,j) in the k-th frame. Then, the complexity of the j-th macroblock row in the k-th frame is defined as follows:

$$C_{MB,R}(j,k)=\sum_{i=0}^{M/16-1} C_{MB}(i,j,k) \tag{1}$$

Here, M is the horizontal size of the frame. The complexity of the k-th frame is obtained by adding the macroblock row complexities as follows:

$$C_F(k)=\sum_{j=0}^{N/16-1} C_{MB,R}(j,k) \tag{2}$$

where N is the vertical size of the frame.

2.2 Frame and Macroblock Row Level Controls

Since frame and macroblock row level controls for the proposed algorithm are similar to those in previous work [9], this subsection will give a summary. Let $b_F^T(k)$ be the bits allocated to the k-th frame:

$$b_F^T(k)=B^T/F_r+\left(B_{SIZE}/2-B_{USED}\right) \tag{3}$$

where F_r is the frame rate of the video, and B^T is the target bitrate. B_{SIZE} and B_{USED} are the encoder buffer size and the used buffer, respectively. Let $b_{MB,R}^T(j,k)$ be the bits allocated to the j-th macroblock row in the k-th frame:

$$b_{MB,R}^T(j,k)=b_F^T(k)\times\frac{C'_{MB,R}(j,k)}{C_F(k-1)} \tag{4}$$

$$C'_{MB,R}(j,k)=\begin{cases} C_{MB,R}(j,k-1) & if \quad j=0 \\ C_{MB,R}(j-1,k) & otherwise \end{cases} \tag{5}$$

In low delay video streaming, the complexity of the current frame cannot be measured until all of the macroblocks of the frame have been input and encoded. Thus, the proposed method predicts the complexity of the current frame from the information of the previous frame. Similarly, the complexity of the current macroblock row cannot be measured until the corresponding macroblocks are encoded, and the complexity of the upper macroblock row is used instead.

2.3 Rate Control for Macroblock Level

Fig. 1 depicts the structure of the proposed rate control at the macroblock level. The main difference between the previous work [9] and the proposed method is the new term (B_{COMP}) that is used to reflect the channel status, which is depicted as gray boxes in the figure. The previous work considered the buffer status only to prevent encoder buffer overflow. The proposed method considers the buffer status in order to adaptively use the available channel bandwidth, as well as to prevent buffer overflow.

The proposed scheme calculates the number of target bits for a macroblock, or $b_{MB}^T(i,j,k)$, and accumulates the target bits and generated bits as follows:

$$b_{MB}^T(i,j,k)=b_{MB,R}^T(j,k)\times\frac{C_{MB}(i,j,k)}{C'_{MB,R}(j,k)} \tag{6}$$

$$B_{ACC,MB}^T(i,j,k)=\sum_{m=0}^{i-1} b_{MB}^T(m,j,k) \tag{7}$$

$$B_{ACC,MB}^G(i,j,k)=\sum_{m=0}^{i-1} b_{MB}^G(m,j,k) \tag{8}$$

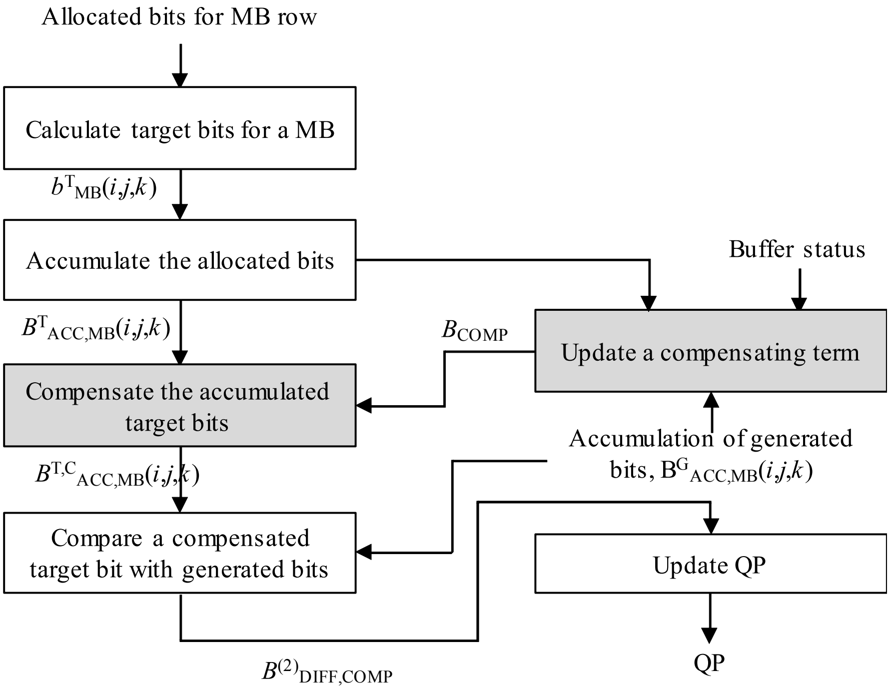

Fig. 1. Structure of bit allocation in a macroblock level.

Let B_{DIFF} be $(B^{G}_{ACC,MB}(i,j,k)-B^{T}_{ACC,MB}(i,j,k))$. When the current QP value is smaller than the proper QP value for the target bitrate, B_{DIFF} will gradually increase. Conversely, a larger current QP will decrease B_{DIFF}. If an encoder sets the proper QP for the given target bitrate, B_{DIFF} should move within a certain range. The algorithm can predict whether the current QP is the proper QP for the target bitrate or not by calculating (B_{DIFF} - $B_{DIFF,P}$). Here, $B_{DIFF,P}$ indicates the reference point used to check whether B_{DIFF} increases or decreases, and $B_{DIFF,P}$ is the value of B_{DIFF} at the last macroblock where the QP value changes.

Now consider the case where the available bandwidth does not match the target bitrate. Although the number of bits generated is the same as the number of target bits, when the available bandwidth is more (lower) than the target bitrate, the encoder buffer becomes empty (or full). Therefore, the method should consider the channel status (or available bandwidth) as well as the relationship between the generated bits and target bits. If the QP value is immediately updated according to the channel status, it will sensitively update the QP value even for small changes in the available bandwidth. Then, the visual quality is not consistent, and the subjective quality becomes worse. Thus, the proposed algorithm loosely controls the QP value. When the available bandwidth is more (or less) than the generated bitrate, the proposed method just prevents the QP value from increasing (or decreasing). This is achieved by considering a compensating term (B_{COMP}) with ($B_{DIFF} - B_{DIFF,P}$):

$$B^{(2)}_{DIFF,COMP} = B_{DIFF} - B_{DIFF,P} + B_{COMP} \tag{9}$$

B_{COMP} is a term used to reflect the relationship between the available bandwidth and the generated bits. (How B_{COMP} is calculated is described later.)

The proposed algorithm controls the QP value according to $B^{(2)}_{DIFF,COMP}$. When $B^{(2)}_{DIFF,COMP} < -T_{TOL}$, the encoder decreases the QP value by 1 to generate more bits. T_{TOL} is the tolerance used to ignore small fluctuations. In the same way, when $B^{(2)}_{DIFF,COMP} > T_{TOL}$, the encoder increases the QP value by 1. When $\left|B^{(2)}_{DIFF,COMP}\right| < T_{TOL}$, the proposed method assumes that $B^{(2)}_{DIFF,COMP}$ is within a certain bound.

The proposed algorithm categorizes the relationship between the generated bitrate and available bandwidth into three cases: Low, Equilibrium, and High. It indirectly predicts this relationship from the encoder buffer status. Low corresponds to when the generated bitrate is lower than the available bandwidth. In this case, the encoder buffer becomes empty. When $B_{USED} < T_L$, the algorithm considers the encoder to be in the Low category. Here, B_{USED} is the number of bits in the encoder buffer, and T_L is the threshold value. Since the available bandwidth is greater than the number of generated bits, the encoder can transmit more data than the current generated bitrate. Hence, if the current macroblock belongs to the Low category, the proposed method decreases the QP value by 1. Note here that there is a lower-bound QP value from the QP refinement process. (QP refinement will be given later, too.) On the other hand, even if the number of generated bits becomes more than the number of target bits, the encoder does not need to increase the QP value to satisfy the target bitrate constraint. Therefore, the algorithm prevents QP from increasing in the Low category. Since the algorithm increases the QP value for $B^{(2)}_{DIFF,COMP} >$ T_{TOL}, B_{COMP} should be set to a value that makes $B^{(2)}_{DIFF,COMP}$

much less than T_{TOL}. When $B^{(2)}_{DIFF,COMP} > 0$, the proposed algorithm sets the value of B_{COMP} to ($B_{DIFF} - B_{DIFF,P}$) in the Low category. Accordingly, $B_{DIFF,COMP}{}^{(2)}$ has an upper bound of 0, and QP cannot be increased. The Low category is where the available channel bandwidth is more than the generated bitrate. In the Low category, there is some room to generate more bits to further utilize channel bandwidth. Hence, if the current macroblock is in the Low category, the proposed method decreases the QP value by 1.

When the generated bitrate is similar to the available bandwidth, the encoder buffer will be within a certain range. When $T_L \leq B_{USED} < T_H$, the algorithm considers the encoder to be in Equilibrium. Here, T_H is the threshold value. In this case, the value of B_{COMP} should just be kept constant. When $T_H \leq B_{USED}$, the algorithm considers the encoder to be in the High category, where the generated bitrate is higher than the available bandwidth. In this case, the algorithm prevents QP from decreasing in the High category. When $B^{(2)}_{DIFF,COMP} < 0$, the proposed algorithm sets the value of B_{COMP} to ($B_{DIFF} - B_{DIFF,P}$) in the High category. When QP is updated, B_{COMP} should be set to 0 so the value of $B^{(2)}_{DIFF,COMP}$ becomes zero.

The next step is the QP refinement process. If the complexities of previous and current frames are similar to each other, the QP values for the macroblocks in the current frame will be similar to the QPs assigned to the macroblocks in the previous frame. The proper QP value for the current frame can be predicted from the QP values in the previous frame as follows:

$$QP_R = \frac{\left\{\sum_{i \in M B_s} QP_i^{'}\right\}}{\#\text{of macroblock}} \tag{10}$$

Here, QP_P is the reference QP for the current frame, and $QP_i^{'}$ is the QP value of the i-th macroblock in the previous frame. The value of QP_R for the proposed algorithm has a floating point value to reduce the rounding effect. When the current frame is similar to the previous frame in terms of similarity, the proposed method clips QP to integer values within the range of [QP_R - 1.5, QP_R + 1.5]. The QP value of the first macroblock in the current frame is set to the lower integer rounded by QP_R. As mentioned above, the proposed method decreases the QP value by 1 in the Low category. However, although the encoder is always in the Low category, the QP value cannot be less than [$QP_P - 1.5$]. Hence, visual quality cannot suddenly change within the frame.

3. Simulation Results

The proposed algorithm was implemented in the H.264 reference software JM 18.5 for the purpose of comparison. This paper includes the experimental results for five test video sequences, including *BasketballDrill* (BD), *Kimono1* (K), *ParkScene* (PS), *Keiba* (KE), and *Johnny* (J). Table 1 illustrates resolutions and frame rates for the test video sequences. All of the sequences consist of 100 frames.

Table 1. Test video sequences.

Sequence name	Resolution	Frame rate (fps)
BasketballDrill	832 × 480	50
Kimono1	1920 × 1080	24
ParkScene	1920 × 1080	24
Keiba	832 × 480	30
Johnny	1280 × 720	60

Fig. 2 depicts the structure of the simulator used to emulate the situation where the target bitrate does not match the available channel bandwidth. In the simulator, the basic time unit is the time required to encode a single macroblock. After the encoder generates the new bits for a macroblock, the encoded bits are immediately stored in an encoder buffer. Then, the buffer status is fed to the rate control block. When the encoder encodes the next macroblock, the buffer status is considered to be in the rate control process. Simultaneously, the virtual channel moves the bitstream from the encoder buffer to the decoder buffer. The number of bits to be moved depends on the available channel bandwidth. The encoder does not know the channel bandwidth, and considers only the target bitrate and the buffer status. The simulator also detects an encoder buffer overflow or a decoder buffer underflow/overflow.

Latency from the encoder to the decoder is set to one-third of a single frame. Hence, the buffer size is as follows:

$$B_{SIZE} = \frac{1}{3 \times Framerate} \times B^T \tag{11}$$

T_L, T_H, and T_{TOL} are set to $B_{SIZE}/20$, $B_{SIZE}/4$, and 1000, respectively. While these values are empirically chosen, they do not sensitively affect performance.

Since the purpose of the proposed algorithm is different from that of the existing methods, the proposed algorithm may not outperform them in normal environments where the target bitrate is the same as the channel bandwidth. Nevertheless, its performance in normal environments should be comparable to that of the existing methods. If performance under normal situations is degraded, compared to that of the existing methods, this work will not be meaningful. As the reference rate control algorithm, this paper chooses the original rate control algorithm implemented in the JM 18.5 reference software (RCUpdateMode=0). Table 2 illustrates the experimental results where the target bitrate is the same as the channel bandwidth. On average, the performance of the proposed algorithm is comparable to that of the original rate control algorithm in the JM 18.5 reference software. It should be noted here that, while the JM algorithm does not guarantee low latency, the proposed algorithm guarantees low latency with a 1/3 frame. While the proposed algorithm provides slightly better performance than the JM algorithm at 40 Mbps and 10 Mbps, its performance is slightly worse than the JM algorithm for *BaseketballDrill*, *ParkScene*, *Keiba*, and *Johnny* sequences at 2 Mbps. The main reason for this performance degradation is the small buffer size of the proposed algorithm. As the target bitrate (or B^T)

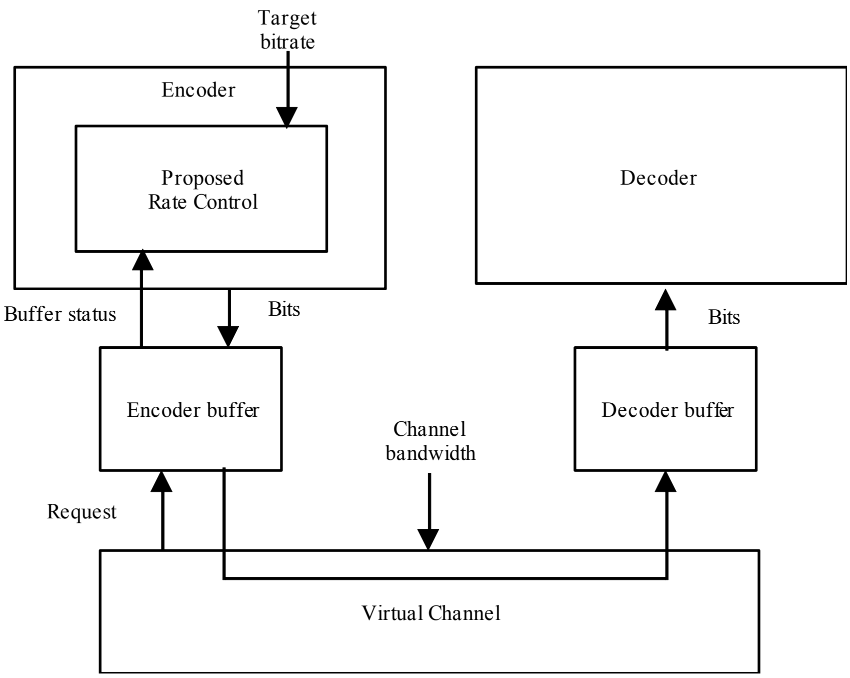

Fig. 2. Structure of the simulator for evaluation the proposed algorithm.

Table 2. Performance comparison of the proposed algorithm with the existing algorithm in normal environments in terms of PSNR (dB) and Rate (Mbps).

Seq.	Method	Target: 40 Mbps		Target: 10 Mbps		Target: 2 Mbps	
		PSNR	Rate	PSNR	Rate	PSNR	Rate
BD	JM	48.1	39.87	41.29	10	34.7	2
	Prop.	48.56	39.82	41.51	9.89	34.51	1.92
K	JM	43.96	39.99	41.39	10	35.77	2
	Prop.	44.12	39.93	41.54	9.71	35.95	1.86
PS	JM	42.63	39.98	38.25	10	32.42	2.01
	Prop.	42.75	39.92	38.45	9.86	32.32	1.91
KE	JM	52.19	39.42	42.74	10	36.72	2
	Prop.	52.34	39.8	42.88	9.84	36.63	1.87
J	JM	46.44	40.01	43.25	10	40.73	2.01
	Prop.	46.84	39.91	43.69	9.9	40.63	1.9

Table 3. Additional performance comparison of the proposed algorithm with the original JM algorithm for 2 Mbps. The latency for the proposed method is set to one frame.

Sequences	Method	Target: 40 Mbps	
		PSNR (dB)	Rate (Mbps)
BasketballDrill	JM	34.7	2
	Proposed	34.66	1.96
ParkScene	JM	32.42	2.01
	Proposed	32.4	1.95
Keiba	JM	36.72	2
	Proposed	36.81	1.94
Johnny	JM	40.73	2.01
	Proposed	40.79	1.95

becomes small, the buffer size also becomes small, as in Eq. (11). Hence, when the encoder buffer gets close to full, the proposed algorithm increases the QP value without considering visual quality. Accordingly, while the rate control prevents buffer overflow, the peak signal-to-noise ratio (PSNR) performance is degraded. Table 3 shows

Table 4. Performance comparison of the proposed algorithm according to the target bitrate vs. the available channel bandwidth in terms of PSNR (dB) and Rate (Mbps). R: Ratio of the target bitrate to the channel BW. CBW: Channel bandwidth.

Seq.	R	CBW: 40 Mbps		CBW: 10 Mbps		CBW: 2 Mbps	
		PSNR	Rate	PSNR	Rate	PSNR	Rate
BD	0.5	48.41	39.13	41.12	9.33	34.29	1.81
	1	48.56	39.82	41.51	9.89	34.51	1.92
	2	48.48	39.97	41.61	10.01	34.06	1.71
K	0.5	44.08	39.68	41.35	8.91	35.45	1.71
	1	44.12	40.13	41.54	9.71	35.95	1.86
	2	44.11	40.22	41.58	10.01	36.16	1.97
PS	0.5	42.69	39.74	38.18	9.27	32.08	1.79
	1	42.75	40.12	38.45	9.86	32.32	1.91
	2	42.76	40.2	38.52	10.01	32.37	1.96
KE	0.5	52.05	38.68	42.44	8.84	35.95	1.64
	1	52.34	39.8	42.88	9.84	36.63	1.87
	2	52.4	39.97	42.93	9.98	36.52	1.93
J	0.5	46.71	38.77	43.47	9.05	40.27	1.69
	1	46.84	39.91	43.69	9.9	40.63	1.9
	2	46.79	39.76	43.73	9.9	40.3	1.55

additional experiments to confirm it. Here, the latency between encoder and decoder is set to one frame so that encoder buffer size increases. The performance of the proposed algorithm is improved with the large size of the encoder buffer. The performance between JM and the proposed method is almost similar. Note here that end-to-end latency is still one frame.

Table 4 shows the experimental results in cases where the available bandwidth does not match the target bitrate. The end-to-end latency in the experiments is set to 1/3 frames. In the table, R represents the ratio of the target bitrate to the available channel bandwidth. For example, when R is 0.5 with *CBW:40Mbps*, the target bitrate and the available channel bandwidth are set to 20Mbps and 40Mbps, respectively. Meanwhile, for R=2.0, the target bitrate is double the available channel bandwidth. Since the target bitrate is set to a value different from the channel bandwidth, how the proposed algorithm uses the available channel bandwidth can be examined. In the table, the PSNR performance significantly depends not on the target bitrate, but channel bandwidth. For example, when the target bitrates are set to 20Mbps, 40Mbps, and 80Mbps for *CBW: 40Mbps* for the *BasketballDrill* (BD) sequence, the PSNR performances are 48.41, 48.56, and 48.48 dB, respectively. Although the target bandwidth does not match the available bandwidth (R: 0.5 and 2.0), its performance is comparable to the normal case (R: 1.0). This means that the proposed algorithm efficiently uses the available channel bandwidth. It should be noted here that the rate control block does not know the value of the available channel bandwidth. The bold characters illustrate the cases where PSNR degradation from the normal case is more than 0.2dB. The maximum PSNR degradation is 0.78dB for the *Keiba* (K) sequence for *CBW:2Mbps*. However, it should be considered that even though its performance is degraded, compared to normal, it still uses the additional available channel bandwidth to improve video quality.

4. Conclusion

This paper introduces a new concept of rate control that adaptively uses the available channel bandwidth for low delay video coding. Previous work supported only control of the intra frame rate for low delay video coding. Therefore, the proposed algorithm extends the previous work to support control of the inter frame rate by changing the complexity measure. Then, the proposed algorithm considers the buffer status in order to fully use the available channel bandwidth to improve video quality. The proposed algorithm is suitable for applications requiring a 1:1 dedicated connection between sender and receiver within a small area over Wi-Fi.

Acknowledgement

This present research has conducted by the Research Grant of Kwangwoon University in 2015.

References

[1] "Wi-Fi Certified Miracast : Extending the Wi-Fi experience to seamless video display", Wi-fi.org, Oct. 2013.

[2] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video. Technol.*, vol. 13, no. 7, pp. 560-576, Jul. 2003. Article (CrossRef Link)

[3] 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.

[4] Y. Liu, Z. G. Li, and Y. C. Soh, "Novel rate control scheme for low delay video communication of H.264/AVC standard," *IEEE Trans. Circuits Syst. Video. Technol.*, vol. 17, no. 1, pp. 6878, Jan. 2007. Article (CrossRef Link)

[5] J. Tsai, "Rate control for low-delay video using a dynamic rate table," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 133137, Jan. 2005. Article (CrossRef Link)

[6] P. Nunes and F. Pereira, "Joint rate control algorithm for low-delay MPEG-4 object-based video encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 12741288, Sept. 2009. Article (CrossRef Link)

[7] M. Jiang and N. Ling, "Low-delay rate control for real-time H.264/AVC video coding," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 467477, June 2006. Article (CrossRef Link)

[8] C. Gong and X. Wang, "Adaptive transmission for delay-constrained wireless video," *IEEE Transactions on Wireless Communications*, vol. 13, no. 1, Jan. 2014. Article (CrossRef Link)

[9] Y. G. Lee and B. C. Song, "An intra-frame rate control algorithm for ultralow delay H.264/Advanced Video Coding (AVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 747752, May 2009. Article (CrossRef Link)

[10] B. C. Song, Y. S. Yi, Y. G. Lee, N. H. Kim, J. H. Ko, T. H. Kim, D. K. Lim, W. H. Ju, J. P. Moon, and K. H. Cho, "1080p 60 Hz Intra-Frame Video CODEC Chip Design and Its Implementation," *Journal of Signal Processing Systems*, vol. 67, issue 3, pp 291-303, June 2012. Article (CrossRef Link)

[11] P. S. Choi, Y. S. Yi, K. S. Ha, Y. G. Lee et al., "An 18ms-lantecy wireless high quality codec SoC for full HD streaming," *IEEE Solid State Circuits Conference (A-SSCC)*, pp. 9-12, Nov. 2011. Article (CrossRef Link)

Yun-Gu Lee received his BSc, MSc, and PhD in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, in 2000, 2002, and 2006, respectively. From 2006 to 2013, he was in the Media Processing Laboratory, DMC Research and Development Center, Samsung Electronics, Republic of Korea, where he was a principal engineer. In 2013, he joined the Department of Computer Science and Engineering, Kwangwoon University, Seoul, Republic of Korea, where he is currently an associate professor. His current research interests include the general areas of image and video coding, image and video processing, image stabilization, computer vision, and three-dimensional display systems.