Adaptive Prediction Block Filter for Video Coding

Yeo-Jin Yoon, Seung-Won Jung, Hahyun Lee, Hui Yong Kim, Jin Soo Choi, and Sung-Jea Ko

In this letter, we propose a new prediction block filter that can reduce errors between the original and prediction blocks. The proposed filter adaptively adjusts filter coefficients by using the previously reconstructed adjacent blocks and their prediction blocks. Then, the filter is selectively applied to the current prediction block according to the rate-distortion optimization. Moreover, since the same filter coefficients can be derived in the decoder, they are not encoded into the bit-stream. The proposed method achieves a 4.65% bitrate saving on average compared with H.264/AVC.

Keywords: H.264/AVC, Wiener filter, adaptive filtering.

I. Introduction

H.264/AVC is the latest video coding standard established by the Joint Video Team [1]. Recently, many new coding tools have been developed to enhance the efficiency of H.264/AVC. Some of these tools are adopted in the key technical area (KTA) developed for preparing for the next generation video coding standard [2]. Specifically, the adaptive interpolation filter (AIF) [3], the adaptive loop filter (ALF) [4], and the high-precision interpolation filter [5] have been introduced to provide a more precise reference picture. Other efficient coding methods, such as the motion vector competition [6] and the extended macroblock (MB) [7], are also adopted in the KTA.

Both the AIF and ALF are designed based on the Wiener filter, which is a well-known optimal filter to cope with the degradation of image quality caused by additional noise and/or blurring [8]. The AIF is obtained by calculating the filter coefficients that make the reference picture closer to the original picture. Similarly, in the ALF, the loop filter coefficients are determined by alleviating the coding noise in the deblocked picture.

To improve the coding efficiency of H.264/AVC, this letter presents a new adaptive prediction block filter (APBF) based on the Wiener filter. The filter coefficients are calculated for each MB using the prediction and reconstruction results of the neighboring MBs. The proposed filter is applied to the prediction block of the current MB, and the filtered block is selectively used according to the rate-distortion (RD) cost [9]. For each MB, if the APBF is used, the coding efficiency is improved by reducing the number of bits required for encoding the residual signal between the prediction and original signal of the MB. Moreover, since the same filter coefficients can be derived in the decoder, they do not need to be encoded into the bit-stream. The proposed algorithm, adaptive filtering of the prediction signal, is a new approach of video coding, and experimental results support its effectiveness.

II. Proposed Method

The proposed method adopts the Wiener filter to transform the prediction signal more closely to the original signal. The Wiener filter coefficients can be calculated by exploiting the original and predicted pixel values of the current MB. However, in this case, encoding the filter coefficients is necessary for each MB because the original pixel values are not available in the decoder. To avoid encoding the filter coefficients, information available at both the encoder and the decoder should be utilized. Therefore, the previously reconstructed neighboring MBs of the current MB are used for the APBF coefficients calculation since the adjacent MBs have similar characteristics to the current MB in most cases. If the Wiener filter obtained by using the neighboring MBs is applied to the

Manuscript received Jan. 31, 2011; revised May 3, 2011; accepted May 19, 2011.

This research was supported by the Korea Communications Commission (KCC), Korea, under the ETRI R&D support program supervised by the Korea Communications Agency (KCA) (KCA-2011-11921-02001).

Yeo-Jin Yoon (phone: +82 3290 3672, yjyoon@dali.korea.ac.kr), Seung-Won Jung (jungsw@dali.korea.ac.kr), and Sung-Jea Ko (sjko@korea.ac.kr) are with the Department of Electronics Engineering, Korea University, Seoul, Rep. of Korea.

Hahyun Lee (hanilee@etri.re.kr), Hui Yong Kim (hykim5@etri.re.kr), and Jin Soo Choi (jschoi@etri.re.kr) are with the Broadcasting & Telecommunications Convergence Research Laboratory, ETRI, Daejeon, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.12.0211.0042

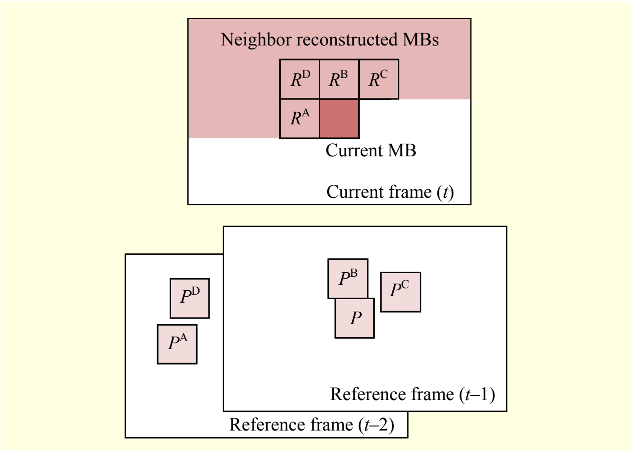

Fig. 1. Neighbor reconstructed MBs $\{R^A, R^B, R^C, R^D\}$ and their corresponding prediction MBs $\{P^A, P^B, P^C, P^D\}$.

current prediction signal, the filtered signal can be more similar to the original signal, resulting in the reduction of prediction errors.

The filter coefficients for the current MB are calculated by using four pairs composed of a prediction MB from $\{P^A, P^B, P^C, P^D\}$ and a reconstructed MB from $\{R^A, R^B, R^C, R^D\}$ in the neighbor as shown in Fig. 1 or by using each one of them. In other words, five types of APBFs, $\{C_{All}, C_A, C_B, C_C, C_D\}$, are utilized in the proposed scheme. The Wiener filter coefficients are calculated by minimizing the mean square error between the prediction and reconstruction signal of the neighboring MBs. For example, a set of filter coefficients using P^A and R^A is calculated as

$$C_A = \underset{\{c_{i,j}^A \mid -N \le i,j \le N\}}{\arg\min} E\left[\left(R_{x,y}^A - \sum_{i,j=-N}^{N} c_{i,j}^A \cdot P_{x+i,y+j}^A\right)^2\right], \quad (1)$$

where C_A is a $(2N+1)\times(2N+1)$ filter consisting of filter coefficients $c_{i,j}^A$ s, and $P_{x,y}^A$ and $R_{x,y}^A$ are the predicted and reconstructed pixel values at the (x, y) position in the neighboring MB, respectively. N pixels are padded at the boundary of the MB for calculation. In the proposed APBF scheme, the center symmetric filter [10] is employed. Thus, the positions located symmetrically with respect to the center point share the same coefficient. C_{All}, C_B, C_C, and C_D are also obtained by substituting P^A and R^A by the corresponding prediction and reconstructed MBs in Fig. 1. Likewise, the same filter coefficients can be derived at the decoder.

Figure 2 shows the flowchart of the APBF scheme at the encoder.

Step 1. Obtain the prediction MB, P, of the current MB.

Step 2. Compute five sets of the Wiener filter coefficients, $\{C_{All}, C_A, C_B, C_C, C_D\}$, for P using the predicted and reconstructed pixel values of the neighboring MBs.

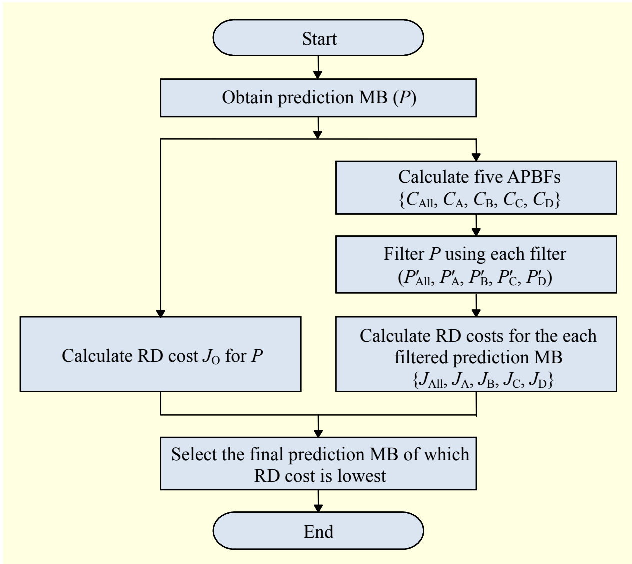

Fig. 2. Flowchart of proposed APBF scheme.

Step 3. Process P by utilizing the filters in step 2 to obtain $\{P'_{All}, P'_A, P'_B, P'_C, P'_D\}$.

Step 4. Calculate the RD costs $\{J_{All}, J_A, J_B, J_C, J_D, J_O\}$ for P'_{All}, P'_A, P'_B, P'_C, P'_D, and P, respectively.

Step 5. Select the final prediction MB that yields the minimum RD cost. Then, the residue between the final prediction MB and the original MB is coded.

According to the selected prediction MB, an index I_{APBF} is transmitted to the decoder to notify the usage of APBF.

The above process is tested for all prediction modes, and the best one with the lowest RD cost is selected. At the decoder side, the same prediction signal used at the encoder side can be derived. If I_{APBF} indicates that the filtering was not used at the encoder side, the prediction signal is maintained. Otherwise, the Wiener filter is constructed, and then the prediction MB is filtered according to the decoded value of I_{APBF}.

III. Performance Analysis

To evaluate the coding efficiency, the proposed APBF scheme is implemented in JM11.0 reference software. For each sequence, 100 frames are encoded with IPPP prediction structure based on the conditions in [11] except for RDOQ. The test platform is Intel Core 2 Quad Q8400 2.66-GHz CPU and 8-GB RAM with Windows 7 64-bit operating system. Bjøntegaard Delta (BD) PSNR and bitrate [12] are calculated to compare the compression performance, and increasing percentages of the elapsed time at the encoder and decoder, ΔT_{Enc} and ΔT_{Dec}, are calculated for measuring the computational complexity.

Although the intra-modes are also exploited for inter-frames, the probability that the intra-mode is selected as a best mode is quite small as shown in Table 1. For this reason, applying APBF to the intra-block does not considerably affect the compression performance, but rather increases the computational complexity. Therefore, the APBF scheme is performed only for the inter-prediction modes.

Table 2 shows the compression performance and the computational complexity of the proposed algorithm. The 3×3 APBFs, 5×5 APBFs, and 7×7 APBFs are tested to evaluate the performance of the proposed filter according to the filter size. The results demonstrate that the smaller the filter tap used, the better the compression performance is compared with H.264/AVC. The best compression performance is obtained when the 3×3 APBF is employed, and the average BD-bitrate gain is about 4.65% for overall sequences. The complexities of the encoder and the decoder are necessarily increased due to the filter coefficient computation and filtering process. It should be also noted that the 3×3 APBF requires the least amount of increase in the computational complexity. The increasing rates of the encoder and decoder complexities are about 62.66% and 88.22% on average, respectively, when the 3×3 APBF is used. As a result, we verify that the APBF algorithm with 3×3 filter achieves the highest coding performance in terms of the bitrate reduction and time consumption.

Table 1. Selection ratio of intra-modes in P frame. Frames for 1 second are coded by the original H.264/AVC standard.

Sequence	Size	Frame rate (fps)	Selection ratio (%)
Kimono	1920×1080 (1080p)	24	9.00
ParkScene		24	3.36
Cactus		50	7.06
RaceHorses	832×480 (WVGA)	30	12.33
BasketballDrill		50	6.88
BQMall		60	2.91
RaceHorses	416×240 (WQVGA)	30	5.07
BasketballPass		50	0.44
BlowingBubbles		50	8.76

In addition, the performance of the proposed scheme is compared with the quadtree-based adaptive loop filter (QALF) [4], which is an in-loop filter applied to a deblocked picture. As shown in Table 3, adaptive filtering is more effective in a decoded signal than a prediction signal. However, when we apply the adaptive filter to both decoded and prediction signals, additional coding gain is achieved at the expense of increase in the computational complexity.

To reduce the encoder complexity of the APBF scheme, filtering can be applied only to the best inter-prediction mode. On the contrary, to further improve the coding efficiency, APBF can be applied to the subblocks in MB. Furthermore, we expect that both coding efficiency and computational complexity can be improved if other filter shapes are exploited instead of the (2*N*+1)×(2*N*+1) symmetric filter, for example, a

Table 2. Experimental results of APBF scheme compared to H.264/AVC.

Sequence	Size	H.264/AVC + 3×3 APBF				H.264/AVC + 5×5 APBF				H.264/AVC + 7×7 APBF			
		BD-PSNR (dB)	BD-bitrate (%)	ΔT_{Enc} (%)	ΔT_{Dec} (%)	BD-PSNR (dB)	BD-bitrate (%)	ΔT_{Enc} (%)	ΔT_{Dec} (%)	BD-PSNR (dB)	BD-bitrate (%)	ΔT_{Enc} (%)	ΔT_{Dec} (%)
Kimono	1920×1080 (1080p)	0.18	–4.70	63.56	82.03	0.16	–4.16	94.92	192.77	0.13	–3.54	157.06	411.28
ParkScene		0.13	–3.50	62.64	58.20	0.12	–3.23	95.76	156.42	0.10	–2.75	153.44	350.13
Cactus		0.16	–6.29	63.51	109.99	0.15	–5.97	94.58	272.07	0.14	–5.39	155.91	599.66
RaceHorses	832×480 (WVGA)	0.12	–2.68	60.21	95.89	0.11	–2.34	87.35	242.88	0.08	–1.81	143.21	494.84
BasketballDrill		0.36	–9.01	64.29	111.86	0.35	–8.65	93.72	297.50	0.31	–7.66	154.04	623.80
BQMall		0.27	–5.98	60.60	92.37	0.26	-5.84	91.27	237.60	0.24	–5.37	147.00	518.18
RaceHorses	416×240 (WQVGA)	0.10	–1.91	58.93	81.33	0.08	–1.53	83.53	202.07	0.06	–1.26	142.42	399.02
BasketballPass		0.20	–4.07	68.43	69.33	0.19	–3.85	92.67	187.80	0.15	–3.18	154.51	372.03
BlowingBubbles		0.15	–3.72	61.74	92.98	0.16	–3.80	83.26	246.28	0.14	–3.42	138.66	510.03
Average on 1080p seq.		0.16	–4.83	63.23	83.41	0.14	–4.45	95.08	207.09	0.12	–3.89	155.47	440.36
Average on WVGA seq.		0.25	–5.89	61.70	100.04	0.24	–5.61	90.78	259.33	0.21	–4.95	148.08	545.61
Average on WQVGA seq.		0.15	–3.23	63.03	81.21	0.14	–3.06	86.49	212.05	0.12	–2.62	145.20	427.03
Average on overall		0.19	–4.65	62.66	88.22	0.17	–4.37	90.78	226.15	0.15	–3.82	149.58	471.00

Table 3. Coding performance of QALF [4] and 3×3 APBF with QALF compared to H.264/AVC.

Sequence	Size	H.264/AVC + QALF [4]				H.264/AVC + 3×3 APBF + QALF [4]			
		BD-PSNR (dB)	BD-bitrate (%)	ΔT_{Enc} (%)	ΔT_{Dec} (%)	BD-PSNR (dB)	BD-bitrate (%)	ΔT_{Enc} (%)	ΔT_{Dec} (%)
Kimono	1920×1080 (1080p)	0.52	–13.18	44.48	77.09	0.57	–14.10	108.96	132.00
ParkScene		0.26	–7.14	40.49	48.78	0.26	–7.09	102.23	77.70
Cactus		0.19	–7.34	36.92	31.68	0.26	–9.90	101.11	111.39
RaceHorses	832×480 (WVGA)	0.14	–3.21	41.17	46.46	0.20	–4.42	101.03	138.80
BasketballDrill		0.57	–13.92	43.33	31.54	0.71	–16.95	106.43	107.55
BQMall		0.34	–7.60	40.80	30.93	0.43	–9.42	100.99	93.30
RaceHorses	416×240 (WQVGA)	0.14	–2.72	53.49	35.96	0.17	–3.37	109.77	109.51
BasketballPass		0.34	–6.74	53.38	17.29	0.40	–8.07	117.30	66.53
BlowingBubbles		0.23	–5.49	51.94	20.79	0.26	–6.08	107.83	82.06
Average on 1080p sequences		0.32	–9.22	40.63	52.52	0.36	–10.36	104.10	107.03
Average on WVGA sequences		0.35	–8.24	41.77	36.31	0.45	–10.26	102.82	113.22
Average on WQVGA sequences		0.23	–4.99	52.94	24.68	0.28	–5.84	111.64	86.03
Average on overall		0.30	–7.48	45.11	37.84	0.36	–8.82	106.18	102.09

cross-shaped filter. Therefore, the APBF scheme needs to be adaptively adjusted depending on the target applications.

IV. Conclusion

In this letter, we proposed a new in-loop filtering algorithm for prediction signal to improve the coding efficiency. First, the Wiener filter is obtained by exploiting the predicted and reconstructed pixel values of the neighboring MBs. Then, the prediction signal of each MB is selectively filtered according to the RD optimization, which results in the reduction of prediction errors. Even though different filters are used for each MB, the coefficients do not need to be encoded since the same coefficients are calculated in the decoder. Consequently, the proposed algorithm achieves up to 9.01% and average 4.65% bit savings over the H.264/AVC standard.

References

[1] T. Wiegand et al., "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, July 2003, pp. 560-576.

[2] ITU-T VCEG KTA Reference Software. http://iphome.hhi.de/suehring/tml/download/KTA/

[3] Y. Vatis and J. Ostermann, "Adaptive Interpolation Filter for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, Feb. 2009, pp. 179-192.

[4] T. Chujoh, N. Wada, and G. Yasuda, "Quadtree-Based Adaptive Loop Filter," ITU-T SG16/Q.6 Doc. COM16-C181-E, Geneva, Switzerland, Jan. 2009.

[5] Y. Ye, P. Chen, and M. Karczewicz, "High Precision Interpolation and Prediction," ITU-T SG16/Q.6 Doc. VCEG-AI33, Berlin, Germany, July 2008.

[6] J. Jung and G. Laroche, "Competition-Based Scheme for Motion Vector Selection and Coding," ITU-T SG16/Q.6 Doc. VCEG-AC06, Klagenfurt, Austria, July 2006.

[7] P. Chenn, Y. Ye, and M. Karczewicz, "Video Coding Using Extended Block Sizes," ITU-T SG16/Q.6 Doc. VCEG-AJ23, San Diego, USA, Oct. 2008.

[8] Y. Liu, "Unified Loop Filter for Video Compression," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 20, no. 10, Oct. 2010, pp. 1378-1382.

[9] G.J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, Nov. 1998, pp. 74-90.

[10] H. Lee et al., "Enhanced Block-Based Adaptive Loop Filter with Multiple Symmetric Structures for Video Coding," *ETRI J.*, vol. 32, no. 4, Aug. 2010, pp. 626-629.

[11] ISO/IEC JTC1/SC29/WG11 and ITU-T SG16/Q.6, "Joint Call for Proposals on Video Compression Technology," WG11 Doc. N11113 and ITU-T SG16/Q.6 Doc. VCEG-AM91, Kyoto, Japan, Jan. 2010.

[12] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD-Curves," ITU-T SG16/Q.6 Doc. VCEG-M33, Austin, USA, Apr. 2001.