Macroblock-Level Deblocking Method to Improve Coding Efficiency for H.264/AVC

Thanh Ha Le, Seung-Won Jung, Chun-Su Park, and Sung-Jea Ko

A macroblock-level deblocking method is proposed for H.264/AVC, in which blocking artifacts are effectively eliminated in the discrete cosine transform domain at the macroblock encoding stage. Experimental results show that the proposed algorithm outperforms conventional H.264 in terms of coding efficiency, and the bitrate saving is up to 5.7% without reconstruction quality loss.

Keywords: H.26/AVC, blocking filters, coding efficiency.

I. Introduction

In many block-based video coding standards including H.264/AVC [1], the deblocking filter is adopted to remove the discontinuity between adjacent blocks, called blocking artifacts. It is well known that the deblocking filter can improve the appearance of decoded frames significantly. The H.264/AVC standard adopts the deblocking filter presented in [2]. The filtering is separately applied to the left vertical and top horizontal boundaries of 4×4 blocks for both luminance and chrominance components in the macroblock (MB). Filtering operation affects up to three pixels on either side of the boundary as shown in Fig. 1(a). Figure 1(b) shows a typical situation where blocking artifacts exist and filtering would take place. Based on the encoding modes and the differences among the pixels, a set of *n*-tap filters is implemented and applied to the pixels. More details of the filters can be found in [2].

Manuscript received Oct. 19, 2009; revised Nov. 24, 2009; accepted Dec. 10, 2009.

This research was supported by Seoul Future Contents Convergence (SFCC) Cluster established by Seoul R&BD Program and also supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. 2009-0080547).

Thanh Ha Le (phone: +82 2 3290 3683, email: ltha@dali.korea.ac.kr), Seung-Won Jung (email: jungsw@dali.korea.ac.kr), Chun-Su Park (email: cspark@korea.ac.kr), and Sung-Jea Ko (email: sjko@korea.ac.kr) are with the Department of Electronics Engineering, Korea University, Seoul, Rep. of Korea.

doi:10.4218/etrij.10.0209.0442

Recently, various methods have been proposed to further improve the efficiency of deblocking filters [3], [4]. However, in all these methods, since the deblocking filter is applied after a frame is encoded, the encoded bitstream still contains blocking artifacts which degrade the coding efficiency.

In this letter, we present a simple but effective method that can improve coding efficiency by eliminating blocking artifacts before encoding the current frame. In our proposed method, the filtering operation is implemented within the mode decision when the MB is encoded. Although all filtering approaches proposed in [2]-[4] are suitable to our work, the filter proposed in [2] is used. Blocking artifacts are removed by modification of the discrete cosine transform (DCT) coefficients. Then, the optimal encoding mode is selected by minimization of the rate-distortion (RD) costs between the original blocks and the filtered blocks.

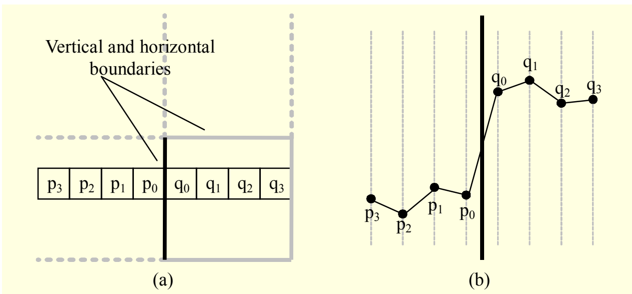

Fig. 1. (a) Pixels adjacent to the vertical boundary and (b) 1D visualization of the block edge with blocking artifacts.

II. Proposed Deblocking Method

1. MB-Level Deblocking Scheme

In H.264/AVC, the deblocking filter is applied to

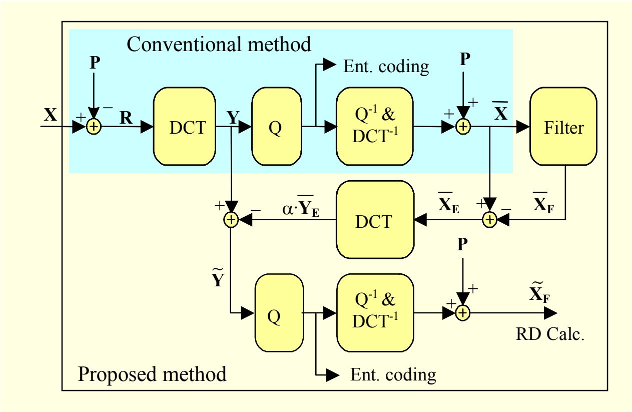

Fig. 2. Block diagram of block encoding.

reconstructed images, and the filtered images are used as references. Thus, the coded bitstream contains information which will be eliminated after deblocking filtering. Reducing the difference is advantageous to improve coding efficiency.

Figure 2 shows the block diagram of our proposed method. After the conventional encoding loop, the reconstructed block $\bar{\mathbf{X}}$ containing the blocking artifact is obtained. To remove blocking artifacts, the deblocking filter is applied to every 4×4 block edge. Here, the filter coefficients are determined based on the boundary strength and pixel values across the block edges [2]. It is possible to extract the blocking artifact $\bar{\mathbf{X}}_E$ by subtracting the filtered block $\bar{\mathbf{X}}_F$ from $\bar{\mathbf{X}}$. To remove $\bar{\mathbf{X}}_E$ from the original DCT coefficient, the DCT is applied to $\bar{\mathbf{X}}_E$ to obtain $\bar{\mathbf{Y}}_E$. Then, we subtract $\bar{\mathbf{Y}}_E$ from the original coefficient block $\mathbf{Y}$ with a weight parameter α to give the refined coefficient block $\tilde{\mathbf{Y}}$:

$$\tilde{\mathbf{Y}} = \mathbf{Y} - \alpha \cdot \bar{\mathbf{Y}}_E, \qquad (1)$$

where the real-valued weight α is in the range [0, 1]. Finally, $\tilde{\mathbf{Y}}$ is quantized, scaled, and added to the prediction block. $\tilde{\mathbf{X}}_F$ is used as a reference for motion compensation in the next frames. The proposed scheme requires the decoder to apply the same deblocking filtering operation at the final stage of decoding the block.

Note that $\bar{\mathbf{X}}_E$ can be removed in the pixel domain by subtracting α-weighted $\bar{\mathbf{X}}_E$ from $\mathbf{R}$. Then, the resultant residual $\tilde{\mathbf{X}}$ is DCT transformed to construct $\tilde{\mathbf{Y}}$. In other words, the subtraction of a blocking artifact is the same for the pixel and DCT domains. In the pixel domain, the whole sequence of encoding and decoding is needed at every iteration to search for the optimal value of α. In our scheme, the forward DCT for block $\mathbf{Y}$ is excluded from the second iteration. Thus, the computational complexity of working in the DCT domain is lower than that of working in the pixel domain when the number of iterations is larger than two.

2. Selection of the Weight Parameter

The efficiency of removing the difference block $\bar{\mathbf{X}}_E$ depends on the selection of weight α. The optimal weight parameter minimizes the Lagrangian RD function:

$$J = \tilde{D} + \lambda \cdot \tilde{R}, \qquad (2)$$

where $\tilde{D}$ is the sum of the absolute difference between $\mathbf{X}$ and $\tilde{\mathbf{X}}_F$, $\tilde{R}$ is the bitrate required for coding the quantized coefficients of $\tilde{\mathbf{Y}}$, and λ defined in the standard [1] is the Lagrange multiplier associated with the rate.

Since an iteration of calculating $\tilde{\mathbf{X}}_F$ is needed for each value of α, searching for the optimum value of α in the range [0, 1] is a burden to real-time applications. Depending on the picture types and color planes, two candidate sets are empirically predefined as shown in Table 1. According to the target application, a user can select a candidate set of α to tradeoff the computational complexity overhead and the coding efficiency.

Table 1. Candidate sets for the weight parameter.

Image type	Color plane	Candidate set	
		Set 1	Set 2
I	Y	{0, 0.25, 0.5, 0.75, 1.0}	{0, 0.25, 0.5}
I	U, V	{0, 0.25, 0.5}	{0, 0.25}
P	Y	{0, 0.25, 0.5, 0.75}	{0, 0.25}
P	U, V	{0, 0.25}	{0}

III. Experimental Results

The simulation conditions are given in Table 2, and the experimental results are presented in Table 3 for various sequence resolutions (size), frame rates (FR), and numbers of encoded frames (FN). To evaluate the performance, the Bjøntegaard delta bitrate (ΔBR) [5] and Bjøntegaard delta PSNR (ΔPSNR) [5] between our proposed method and the

Table 2. Simulation conditions.

Reference software	KTA 2.0 [6]
Spatial resolution	QCIF, CIF, 720p
Number of references	4
MV search range	64
GOP structure	IPPP (high profile)
Entropy coding method	CABAC
QP	12/17/22/27 for I slices
	13/18/23/28 for P slices

Table 3. Experimental results.

Sequence	Size	FR/FN	Set 1			Set 2		
			ΔBR (%)	ΔPSNR (dB)	ΔT	ΔBR (%)	ΔPSNR (dB)	ΔT
Claire	QCIF	30/100	–4.730	0.221	5.1	–4.685	0.218	1.7
M&daughter	QCIF	30/100	–5.407	0.259	5.4	–5.259	0.255	1.8
News	QCIF	30/100	**–5.699**	**0.347**	5.4	**–5.493**	**0.331**	1.8
Deadline	QCIF	30/100	–4.622	0.272	5.0	–4.131	0.241	1.7
Students	QCIF	30/100	–5.102	0.285	4.8	–4.918	0.275	1.6
Average	QCIF		–5.112	0.277	5.1	–4.897	0.264	1.7
Highway	CIF	30/100	–3.331	0.068	4.9	–3.058	0.059	1.8
M&daughter	CIF	30/100	–3.846	0.133	4.8	–3.365	0.117	1.7
News	CIF	30/100	**–5.381**	**0.240**	5.0	**–4.986**	**0.223**	1.7
Paris	CIF	30/100	–4.500	0.271	6.0	–4.352	0.261	2.1
Silent	CIF	30/100	–4.945	0.247	5.3	–4.306	0.214	1.9
Average	CIF		–4.400	0.192	5.2	–4.013	0.175	1.8
Crew	720p	60/150	–2.198	0.077	5.3	–1.621	0.056	1.9
Cyclists	720p	60/150	–2.310	0.074	4.9	–1.613	0.052	1.7
Jets	720p	60/150	**–5.164**	**0.083**	5.1	**–4.482**	**0.065**	2.0
Raven	720p	60/150	–2.119	0.070	5.0	–1.392	0.047	1.8
Shuttle start	720p	60/150	–2.802	0.062	4.8	–2.166	0.049	1.7
Average	720p		–2.917	0.073	5.0	–2.255	0.054	1.8

conventional H.264/AVC are used. The computational complexity overhead (ΔT) is provided in the form of the proportion of the increased encoding time of the proposed method to a conventional one. It can be seen that our proposed method improves the RD performance of all tested sequences. For set 1, the maximum (average) ΔBR is achieved by 5.70% (5.11%), 5.38% (4.40%), and 5.16% (2.92%) for QCIF (176×144), CIF (352×288), and 720p (1280×720) sequences, respectively. Also, for the set 2 structure, the maximum (average) ΔBR is achieved by 5.49% (4.90%), 4.99% (4.01%), and 4.82% (2.26%) for QCIF, CIF, and 720p sequences, respectively. In comparison with set 1, the computational complexity overhead of set 2 is much lower while the coding gain is guaranteed.

IV. Conclusion

In this letter, an MB-level deblocking method is proposed. To improve the coding efficiency, the blocking artifacts are extracted and eliminated in the DCT domain at the encoding stage of the MB. Experimental results exhibit a significant coding gain. Further improvements in coding performance by adopting more effective deblocking filter operations and better searching methods for the parameter α will be the subject of future work.

References

[1] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264-ISO/IEF 14496-10 AVC)*, JVT-G050, Mar. 2003.

[2] P. List et al., “Adaptive Deblocking Filter,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, July 2003, pp. 614-619.

[3] Y. Chiu and L. Xu, *Adaptive (Wiener) Filter for Video Compression*, ITU-T SG16 Q.6 Document, VCEG-AI14, Berlin, July 2008.

[4] Y. Huang et al., *Adaptive Quadtree-Based Multi-reference Loop Filter*, ITU-T SG16 Q.6, VCEG-AK24, Japan, Apr. 2009.

[5] G. Bjøntegaard, *Calculation of Average PSNR Differences between RD-Curves*, ITU-T SG16 Q.6 Document, VCEG-M33, Austin, Apr. 2001.

[6] Joint Video Team (JVT), KTA Reference Software, http://iphome.hhi.de/suehring/tml/kta/.