Signal Processing: *Image Communication* 27 (2012) 260–268

Contents lists available at SciVerse ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Content-adaptive deblocking for high efficiency video coding

Zhiwei Xiong *, Xiaoyan Sun, Jizheng Xu, Feng Wu

Microsoft Research Asia, 13107, Building 2, No. 5 Dan Ling Street, Haidian District, Beijing 100080, PR China

ARTICLE INFO

Article history:
Received 13 July 2011
Accepted 31 January 2012
Available online 4 February 2012

Keywords:
Content-adaptive deblocking
Local directional feature
Orientation energy edge detection
High efficiency video coding

ABSTRACT

In this paper, a content-adaptive deblocking method is proposed to improve the visual quality of block-based DCT compressed videos. We find that the edge information obtained through the global orientation energy edge detection (OEED) on an initially deblocked image provides a robust partition of local directional features (LDFs). Based on this partition, for the directional featured region, we design corresponding filter orientation and thresholds to best preserve image details while reducing blocking artifacts; for the consecutive non-featured region, we impose extra smoothing to suppress the visually severe blocking artifacts. Experimental results demonstrate that our method effectively improves the visual quality while well maintaining the objective fidelity of block-based DCT compressed videos, compared with the H.264/AVC deblocking method and other existing directional deblocking methods.

1. Introduction

Current state-of-the-art video coding standard H.264/AVC uses motion compensated block-based DCT followed by quantization to remove spatial and temporal relevance in videos. At low bit-rate, this strategy gives rise to discontinuities between adjacent blocks, which is known as blocking artifacts. Consequently, adaptive deblocking filter [1] is required as a part of the coding standard.

The H.264/AVC deblocking filter is adaptive in two aspects. One is boundary strength decision based on the block type (intra or inter) and whether the block contains coded coefficients. The other is filter decision based on the intensity difference between boundary pixels in two adjacent blocks. The filter is switched off when there is a significant change across the block boundary, which is more likely to be original image features rather than blocking distortion. The thresholds defining "a significant change" depend on the quantization parameter (*QP*).

Although the adaptive deblocking in H.264/AVC helps reduce the blocking artifacts and improve the quality of reconstructed videos, it still has some limitations. First, deblocking is always performed along the direction perpendicular to the block boundary, without considering local features inside the block. Second, the same filter orientation and thresholds are applied to highly different image content, which potentially degrades the deblocking performance.

Tremendous methods have been proposed in literature for reducing blocking artifacts, such as lapped orthogonal transform [2], projection onto convex sets [3], maximum a posteriori [4], and filtering in transform domain [5]. In this paper, we will focus on the block boundary filtering approach which is compatible with that in H.264/AVC. In [12] Kirenko et al. propose a simple deblocking method for compressed video that allows the block grid position and its visibility to be determined without the need for access to the coding parameters. With a combination of this information and the results of local spatial analysis of luminance and chrominance components of a decoded image, the coding artifacts can be effectively suppressed while preserving the sharpness of object edges. Zhai et al. [13] propose another efficient deblocking method based on postfiltering on shifted windows of image blocks, where the threshold determining which blocks are used in smoothing is adaptive to both the image quality factor specified by the coder and the standard deviations of the

* Corresponding author. Tel.: +86 10 5917 4845.
E-mail addresses: zhxiong@microsoft.com (Z. Xiong), xysun@microsoft.com (X. Sun), jzxu@microsoft.com (J. Xu), fengwu@microsoft.com (F. Wu).

0923-5965/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2012.01.019

associated blocks, to take the global and local image characteristics into account, respectively. Chetouani et al. [14] propose to reduce the blocking artifacts in the compressed image by analyzing their visibility, and a perceptual map is obtained using some Human Visual System characteristics, which is then used as input to a recursive filter to reduce the blocking effect. However, the local directional features in the image are not taken into account explicitly in these methods.

The recent progress concentrates on edge-adaptive deblocking. For example, in [6] blocks are classified into smooth, edge, and texture based on an edge map and different filters are applied to different blocks. A more promising approach is directional deblocking, which performs filtering along local edge orientation [7,8]. For directional deblocking, the most challenging problem is how to distinguish local directional features (LDFs) given a low quality reconstructed frame. Huang et al. [7] propose to calculate edge orientation around block boundaries on the reconstructed frame, while Jeong et al. [8] argue that edge orientation deduced from the intra prediction mode is more accurate. On the other hand, high resolution videos have become more and more popular nowadays. For compressed high resolution videos at low bit-rate, the most annoying artifacts sometimes come from large smooth regions with small variation. Although the blocking degradation is reduced to a certain extent, it still visibly exists spatially and temporally, which heavily impairs the perceptual experience of audience. However, this issue is rarely addressed before.

In this paper, we would like to point out that the edge information obtained through the global orientation energy edge detection (OEED) [9] can provide a robust LDF partition at the precision of transformed block size, if performed on an initially deblocked image. Based on the LDF partition through OEED, we propose a content-adaptive deblocking method. For the LDF region (i.e., region containing LDF), we steer the filter orientation in accordance with the LDF and moreover, we design the filter thresholds appropriately to best preserve image details while reducing blocking artifacts. For the consecutive non-LDF region (i.e., large smooth region with small variation), we impose heavy and long distance smoothing to suppress the visually severe blocking and guarantee the perceptual quality of reconstructed videos. Since the deblocking filter is in-loop, once the blocking artifacts in an intra-frame are effectively removed, the blocking artifacts in subsequent inter-frames are also largely suppressed due to the improved prediction. So it is appropriate to apply the proposed method on intra-frames only from the efficiency point of view. Compared with H.264/AVC as well as previous deblocking methods, our approach improves not only filter orientation adaptivity, but also filter strength adaptivity to different video contents.

The rest of this paper is organized as follows. Section 2 briefly introduces the employed OEED algorithm and the LDF partition process. Section 3 explains in detail the deblocking filter design for LDF regions as well as consecutive non-LDF regions. Experimental results are presented in Section 4, and Section 5 concludes the paper.

2. Local directional feature partition

The flowchart of our deblocking scheme is shown in Fig. 1. Given a decoded frame after inverse DCT and motion compensation, its luminance component is first processed with the H.264/AVC deblocking filter. Then a global OEED is conducted on the initially deblocked image to get an LDF partition. Based on the LDF partition, different deblocking modes are decided for all blocks. Finally, content-adaptive deblocking is performed on both the luminance and chroma components of the decoded frame, with some parameters (e.g., filter orientation and thresholds) updated by the deblocking mode decision, while others (e.g., boundary strength) are directly inherited from the initial H.264/AVC deblocking.

2.1. Orientation energy edge detection

We use a global OEED for LDF partition. Edges are extracted by orientation energy on an initially deblocked luminance image I

$$W(x,y,\theta) = (F_\theta^o * I)^2 + (F_\theta^e * I)^2 \quad (1)$$

where F_θ^o and F_θ^e are the first and second Gaussian derivative filters at orientation θ, respectively. These filters consist of a filter bank shown in Fig. 2(a) (a total of 8 orientations are used). For each pixel (x, y) in I, the filter output energy W will first have a maximum at the orientation θ parallel to the edge orientation, and then along a line perpendicular to the direction defined by θ, W will have a maximum right at the edge position. Therefore, edges can be found by marking all the points $p=(x, y, \theta)$ that satisfy

$$\frac{\partial}{\partial\theta} W(p) = 0, \ \frac{\partial}{\partial \mathbf{v}_\theta} W(p) = 0 \quad (2)$$

where $\mathbf{v}_\theta$ is the unit vector orthogonal to the direction defined by θ. (For more details of OEED, please refer to [9].)

After OEED, we will have an edge map recording the edge pixel locations, and an angle map recording the orientation of each pixel, as illustrated by Fig. 2(b) and (c). It has been proven in [9] that there is a systematic localization error for composite edges using any finite number of linear filters, while a non-linear, quadratic filtering approach is adequate. Compared with the generally used linear filter, e.g., Canny edge detector, the Gaussian derivative filters used in OEED are demonstrated to give more accurate edge location and orientation. Therefore, they are able to provide more reliable information to distinguish LDF regions. Note that an initial H.264/AVC deblocking on the luminance component is preferred

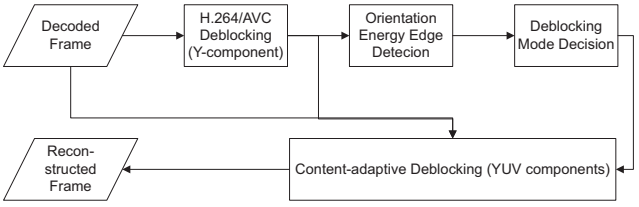

Fig. 1. Flowchart of our deblocking scheme.

Fig. 2. (a) Filter bank used in OEED. (b) and (c) An example of edge and orientation maps obtained by OEED. Edge pixels are marked white in (b) and different intensities indicate different orientations in (c).

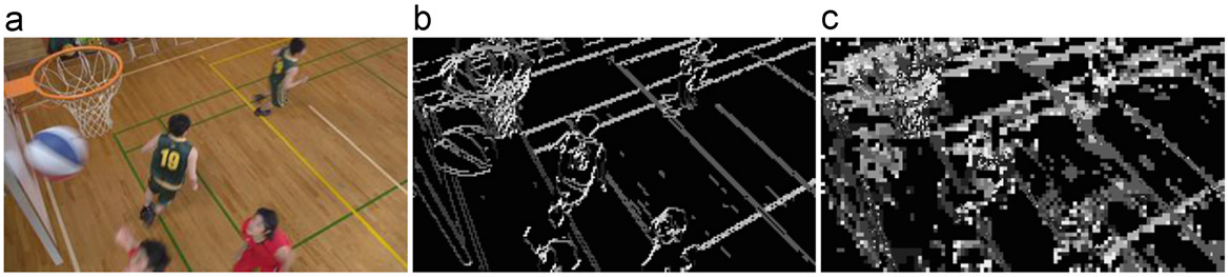

Fig. 3. LDF partition results on a reconstructed frame (832×480, $QP=37$). (a) Original frame, (b) LDF partition through OEED, and (c) LDF partition through intra-prediction. Different intensities indicate different deblocking modes in (b) and (c). Non-LDF regions are marked black.

prior to OEED to further improve its accuracy, from which certain parameters can be directly used in the content-adaptive deblocking later.

2.2. Deblocking mode decision

In our scheme, the embodiment of LDF partition is determining an appropriate deblocking mode for each block. In total we define 8 deblocking modes: the H.264/AVC mode (mode A), an extra smoothing mode (mode B), and 6 directional modes (mode C-m, $m=2$, 3, 4, 6, 7, and 8) in accordance with the filter bank of OEED except the horizontal and vertical directions (refer to Fig. 2(a)). The mode decision is based on the following rules.

Mode C: for each 4×4 or 8×8 transformed intra block B_I with its upper and left neighbors B_U and B_L available, if edge pixels appear in B_I, we then calculate the average orientation index m_I (an integer between 1 and 8 denoting the orientation occurs most frequently among all edge pixels in the block) and the orientation variance v_I (the average orientation deviation of all edge pixels in the block from m_I). Similarly we can get m_U, v_U and m_L, v_L, in case there are also edge pixels in the two neighbors, otherwise we set these parameters to 0. Denote $D(a, b)$ as the minimum difference between two orientation indices a and b in a circular manner corresponding to Fig. 2(a) (e.g., $D(1, 2)=1$ and $D(1, 8)=1$), once

$$D(m_I,m_U)+D(m_I,m_L)<T_1, \quad m_I \neq 1,5$$
$$v_I+v_U+v_L<T_2 \tag{3}$$

it suggests the current block has consistent directional features with its neighbors, and thus B_I is decided to be with mode C-m_I.

Mode B: for each 16×16 transformed intra-block B_I with its upper and left neighbors B_U and B_L available, if no edge pixels appear in B_I, B_U, and B_L, we then calculate the intensity variances v_I, v_U, and v_L in the three blocks. Once

$$v_I+v_U+v_L<T_3 \tag{4}$$

it suggests the current block is inside a consecutive non-LDF region, and B_I is decided to be with mode B.

Mode A: the remaining blocks.

Fig. 3(b) and (c) gives two LDF partition results for blocks with directional deblocking modes: one is based on OEED and the other is deduced from the intra-prediction mode as suggested in [8]. Obviously, OEED is more accurate than intra-prediction in distinguishing LDFs even at low bit rate. Fig. 4 gives another example for blocks with the extra smoothing mode. Videos with different contents may have quite different deblocking mode distributions, which in turn indicates the deblocking filter should be flexible.

3. Deblocking filter design

3.1. Directional deblocking filter

For deblocking mode C, the filtering involved pixels are no longer in a line perpendicular to the block boundary, but along 6 different orientations, as shown in Fig. 5. Suppose $p_3,\ldots,p_0,q_0,\ldots,q_3$ are the selected pixels for a 1D filter; during directional deblocking their values are modified as

$$\begin{aligned} p_n &= (\lambda_{n,1},\ldots,\lambda_{n,8})(p_3,\ldots,p_0,q_0,\ldots,q_3)+\lambda_{n,0} \\ q_n &= (\lambda_{n,1},\ldots,\lambda_{n,8})(q_3,\ldots,q_0,p_0,\ldots,p_3)+\lambda_{n,0} \end{aligned} \tag{5}$$

The above filter coefficients $\lambda_{n,m}(0 \le n \le 2, 0 \le m \le 8)$ are the same as that used in H.264/AVC.

Fig. 4. Deblocking mode decision in consecutive non-LDF regions on a reconstructed frame (1920 × 1080, $QP=37$). Blocks marked red are with the extra smoothing mode.

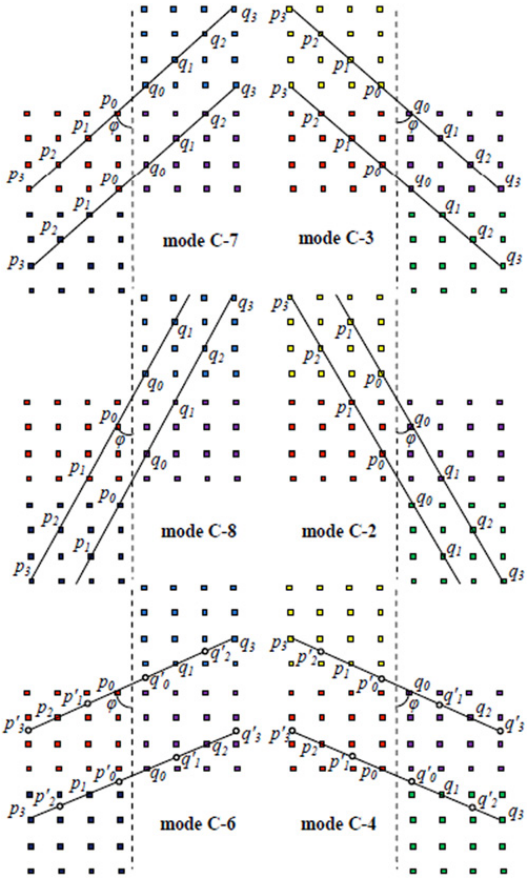

Fig. 5. Directional deblocking filters (mode C). Vertical block boundaries are taken for example. $p_3,\ldots,p_0,q_0,\ldots,q_3$ are the selected pixels for a 1D filter. For mode C-4 and C-6, interpolation is needed at certain half pixels marked $p'_3,\ldots,p'_0,q'_0,\ldots,q'_3$, but only integer pixels are filtered.

In H.264/AVC there are two thresholds α and β determined by the average QP employed over the block boundary, and deblocking only takes place if

$$\begin{aligned} |p_0-q_0| &< \alpha(QP) \\ |p_1-p_0| &< \beta(QP) \\ |q_1-q_0| &< \beta(QP) \end{aligned} \tag{6}$$

As the deblocking performance highly depends on these thresholds, they should be carefully designed for our directional deblocking, not only considering QP but also the directional mode. Since our directional deblocking is always performed along the edge orientation, the filter decision thresholds to distinguish edges from blocking degradation should be relaxed. Generally we have

$$\begin{aligned} \alpha' &= A(QP)B(m_I)\alpha \\ \beta' &= A(QP)B(m_I)\beta \end{aligned} \tag{7}$$

where A is a magnification parameter related to QP, and B is a modulation parameter related to the directional mode m_I.

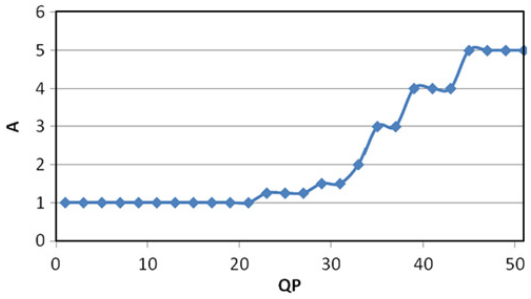

Fig. 6. Relationship between magnification parameter A and QP.

The magnification parameter A is trained on a set of images containing typical directional edges. Fig. 6 shows the relationship between A and QP. At high bit-rate (low QP), image details are still well preserved, so directional deblocking is less encouraged to avoid smoothing anti-directional texture, while at low bit-rate (high QP), directional deblocking can be much stronger as it will not blur the remaining edges.

For the modulation parameter B, we define it based on the following observation. Since the blocking degradation is caused by different quantizations in two adjacent blocks, it is most severe across the block boundary and least severe along the boundary. Thus deblocking should be stronger if its direction is closer to being perpendicular to the boundary. Suppose φ is the angle between the directional mode m_I and the block boundary (refer to Fig. 5), then

$$B(m_I) = \sin\varphi \tag{8}$$

Since the number of directional modes is limited, Eq. (8) can be efficiently replaced by a look-up table. To further clarify the threshold relaxing process, a brief algorithm description is given below.

Input: QP and directional mode m_I

Output: relaxed thresholds α' and β'

Begin

1. Obtain initial thresholds α and β according to QP from a look-up table, as in H.264/AVC deblocking.
2. Given QP, determine magnification parameter A from a look-up table in accordance with Fig. 6.
3. Given m_I, determine modulation parameter B from a look-up table in accordance with Eq. (8).
4. Calculate α' and β' from α, β, A and B through Eq. (7).

End

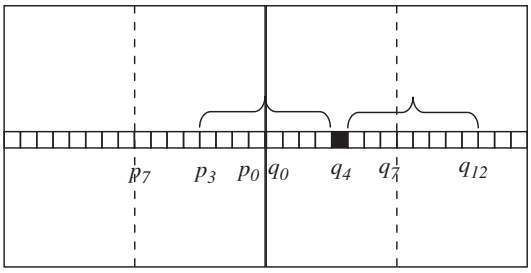

Fig. 7. Extra smoothing filter (mode B). A vertical boundary of a luminance block is taken for example. $p_7,\ldots,p_0,q_0,\ldots,q_7$ are the pixels to be filtered. In this example, the filtered value of q_4 is obtained by averaging $p_3,\ldots,p_0,q_0,\ldots,q_{12}$.

3.2. Extra smoothing filter

For deblocking mode B, the extra smoothing is performed along lines perpendicular to the upper and left block boundaries, as shown in Fig. 7. Suppose $p_{N-1},\ldots,p_0$, $q_0,\ldots q_{N-1}$ are the pixels in the same row or column of two adjacent blocks, during extra smoothing their values are modified as

$$p_n = \left(\sum_{i=0}^{i=n+N/2} p_i + \sum_{j=0}^{j=N/2-n-1} q_j \right) \Big/ (N+1)$$

$$q_n = \left(\sum_{i=0}^{i=N/2-n-1} p_i + \sum_{j=0}^{j=n+N/2} q_j \right) \Big/ (N+1)$$

$$0 \le n \le \frac{N}{2}-1, \quad N = \begin{cases} 16 & \text{for luminance block} \\ 8 & \text{for chroma block} \end{cases} \tag{9}$$

This is a heavy and long distance smoothing across the block boundary and all pixels in the consecutive non-LDF regions will be filtered. For the extra smoothing mode, no boundary strength and filter thresholds are needed.

Once the deblocking mode decision is done, the 8 different deblocking filters are simultaneously employed in a raster-scan order at the macroblock level for both luminance and chroma components.

4. Experimental results

We implement our deblocking algorithm in jm11.0/kta2.4 and evaluate its performance on the test sequences provided by MPEG (Motion Picture Experts Group)/VCEG (Video Coding Experts Group) for recent HEVC (High Efficiency Video Coding). The 1st second of each sequence is coded, in both all-intra and hierarchical-B (with an intra-period of 8) configuration. The frame rate is set exactly the same as that of the original sequence. As deblocking mainly works under low bit-rate, *QP* of intra-frame is set to 27, 32, 37, and 42. For the hierarchical-B case, there is 1 *QP* increment gradually for inter-frames at higher levels. In OEED, the standard variance and window size for 2D Gaussian derivative filters are set to (1.3, 3.9) and 17×17, respectively. The three thresholds T_1, T_2, and T_3 in deblocking mode decision are set to 2, 3, and 12, respectively, at both encoder and decoder sides. These thresholds we choose are actually lowerbounds below which the block appearance should be distinct (either directional or smooth), from the test on a variety of video contents. Comparing with data-dependent thresholds, this conservative yet simple way avoids inaccurate classification efficiently. In accordance, magnification parameters used in the directional deblocking are trained from a set of distinct directional image patterns shown in Fig. 8. The training images are first compressed under different *QP*. Since each image only contains edges with a single orientation, the corresponding deblocking mode is applied on the whole image. Under a certain *QP*, different magnification parameters are tested on the training set and the one giving the best deblocking performance is selected. Some experimental results compared with the H.264/AVC deblocking [1] and intra-predictive deblocking [8] are presented below. Note we only impose the proposed deblocking on intra-frame, and the quality of inter-fame is supposed to be enhanced along with the improved prediction.

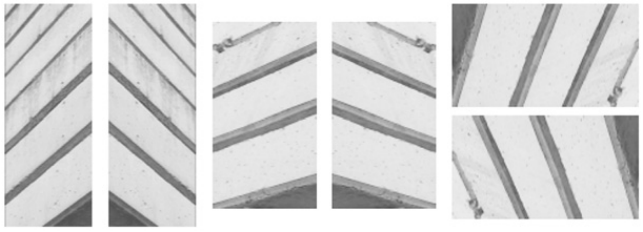

Fig. 8. Training images taken from *Foreman* and its rotated versions. Magnification parameters related to directional deblocking filter are trained from these image patterns.

Figs. 9–11 show some visual quality comparison results of intra-coded frames. As can be observed in Fig. 9, after H.264/AVC deblocking (b) or intra-predictive deblocking (c), there still remain artifacts around the zebra stripe (LDF region), which are alleviated by our method in (d). Meanwhile, the severe blocking on the road (non-LDF region) is also much more effectively removed by our method. Fig. 10 gives another example for non-LDF regions and Fig. 11 for LDF regions; both are susceptible to blocking artifacts. With our method, visual quality improvement can be perceived in large smooth regions and directional edge regions, compared with the other two deblocking methods. Fig. 12 shows a comparison result for different frame types. All inter-frames use the same H.264/AVC deblocking, whereas the quality of reference frames for them is different. It can be seen, with our method, the blocking artifacts are more effectively reduced not only on the intra-frame but also on the inter-frames, due to the improved prediction. (Please see the electronic version for better visualization.)

To evaluate the performance of our proposed deblocking method objectively, Figs. 13 and 14 give a comparison results using the non-reference blocking visibility measure proposed by Zhai et al. in [11]. This measure effectively estimates the boundary discontinuity caused by blocking artifact by jointly considering the influence of texture complexity and luminance magnitude in each block. In Fig. 13 the blocking visibility is measured frame by frame on a short sequence (all-intra-coded), from which we can see the proposed method steadily outperforms H.264/AVC

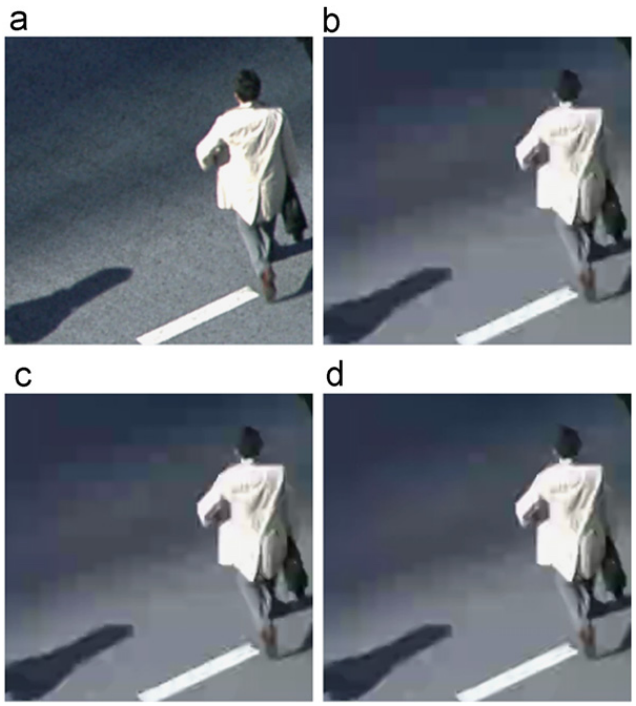

Fig. 9. Visual quality comparison between different deblocking algorithms from the test sequence *PeopleOnStreet_2560 × 1600_30_crop* (*QP*=42). (a) Original image, (b) H.264/AVC deblocking (30.586 dB), (c) intra-predictive deblocking (30.601 dB), and (d) our proposed deblocking (30.637 dB).

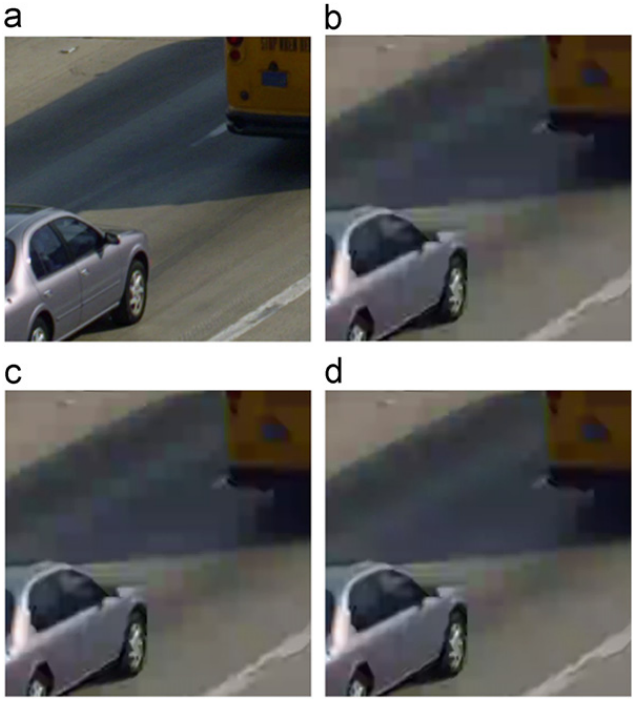

Fig. 10. Visual quality comparison between different deblocking algorithms from the test sequence *Traffic_2560 × 1600_30_crop* (*QP*=42). (a) Original image, (b) H.264/AVC deblocking (30.686 dB), (c) intra-predictive deblocking (30.792 dB), and (d) our proposed deblocking (30.816 dB).

and intra-predictive deblocking, under two different *QP*. Fig. 14 shows another example. (Low blocking visibility indicates better perceptual quality.)

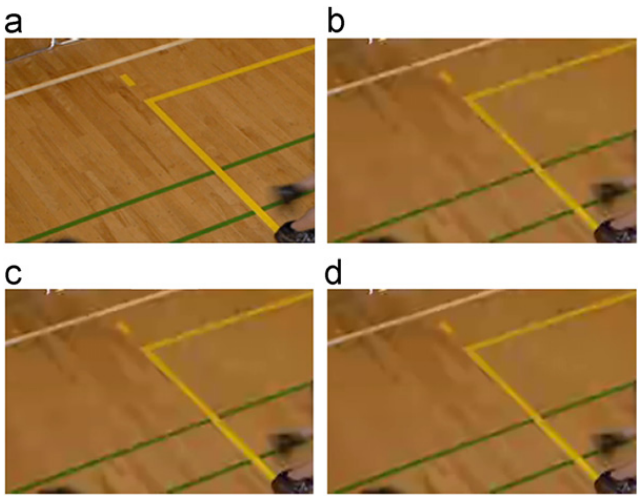

Fig. 11. Visual quality comparison between different deblocking algorithms from the test sequence *BasketballDrill_832 × 480_50* (*QP*=37). (a) Original image, (b) H.264/AVC deblocking (32.547 dB), (c) intra-predictive deblocking (32.586 dB), and (d) our proposed deblocking (32.628 dB).

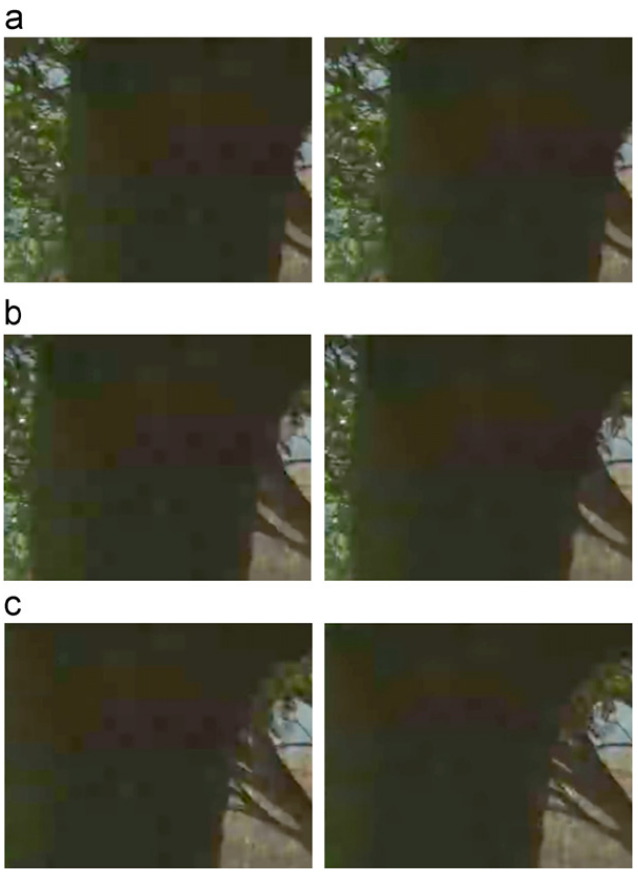

Fig. 12. Visual quality comparison on different frame types from the test sequence *ParkScene_1920 × 1080_24* (*QP*=37). (a) Intra-frame, (b) 2nd inter-frame, and (c) 4th inter-frame. Left: H.264/AVC deblocking, and right: our proposed deblocking (only on the intra-frame).

As mentioned all through this paper, our proposed content-adaptive deblocking method aims at improving the visual quality of block-based DCT compressed video at

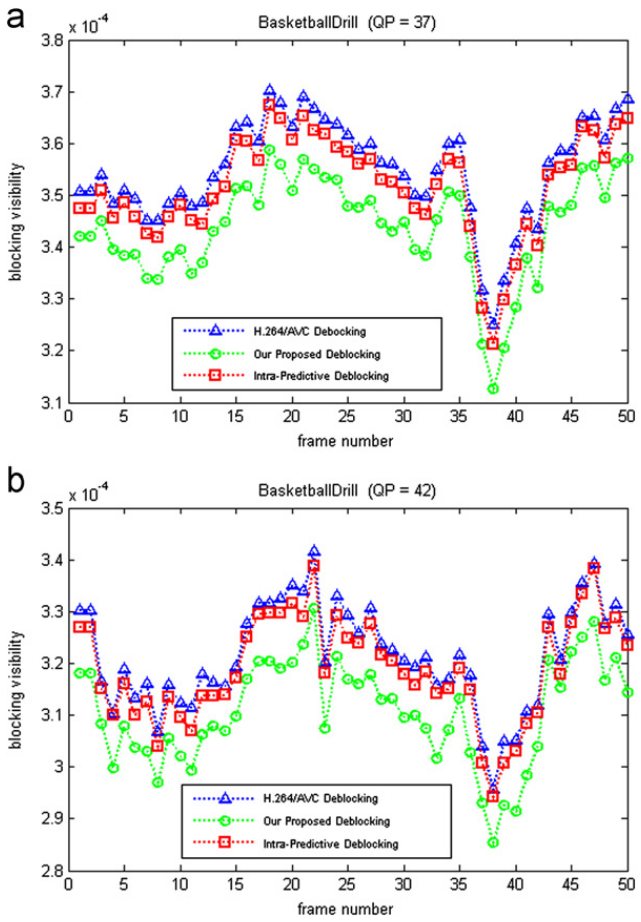

Fig. 13. Blocking visibility comparison between different deblocking algorithms on the test sequence *BasketballDrill_832 × 480_50*. Low blocking visibility indicates better perceptual quality.

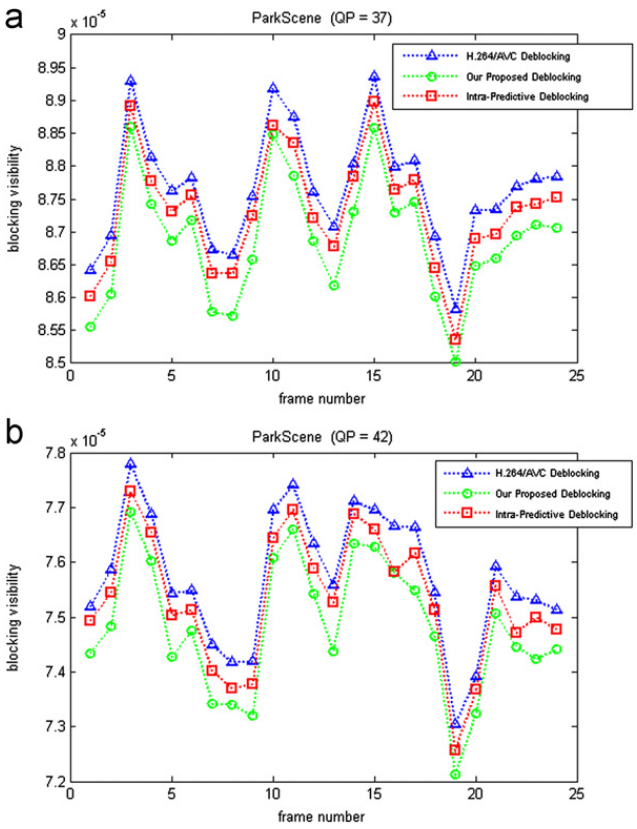

Fig. 14. Blocking visibility comparison between different deblocking algorithms on the test sequence *ParkScene_1920 × 1080_24*. Low blocking visibility indicates better perceptual quality.

low bit-rate, which is highly demanded in the next generation video coding standard. So its design does not emphasize on the objective fidelity. However, as the visual artifacts are effectively reduced, the objective quality is well maintained on the luminance component, and even slightly better on the chroma components. Numerical results evaluated in the Bjontegaard-Delta bit-rate (BD-rate) measure are enclosed in Table 1, for both all-intra and hierarchical-B cases. The following should also be noted:

1) Our method especially benefits sequences with directional edge regions and large smooth regions (which are quite common in high-definition video content), and deblocking artifacts are most observable in these regions. For texture regions, our method may cause slight quality loss. It can be observed (if carefully) from the two images in column (c) in Fig. 12 that there is slight distortion in the lower right corner of the bottom image, where there was a texture region (leaves) originally. However, high contrast and complicated pattern in texture regions usually cover up the difference caused by deblocking, so the influence on visual quality can be neglectable. Globally, the proposed method improves the perceptual quality at the expense of increased computational complexity.
2) Our method works better at low bit-rate (high *QP*), as shown by the PSNR improvement enclosed in Figs. 9–11. At high bit-rate (low *QP*), our method may cause slight PSNR loss; it can be observed from Table 1 that the BD-rate occasionally increases slightly. However, the blocking artifacts are already quite difficult to perceive at high bit-rate, so the influence on visual quality can be neglectable.
3) Our method provides larger improvement on the chroma components than the luminance component, which is consistent with the fact that chroma blocking is often more serious than luminance blocking at low bit-rate, as can be observed from the experimental results.
4) Although our deblocking is performed only on the intra-frame (inter-frames still use H.264/AVC deblocking), the visual and objective quality improvement for inter-frames is still maintained due to the enhancement of reference frame.

In a practical hierarchical-B case with periodical intra-placement for random access, the average increases of encoding and decoding times of our scheme are nearly 2% and 170% over the baseline jm11.0/kta2.4, respectively. On the other hand, directional or adaptive deblocking, including the one proposed in this paper, may require more hardware resources in practical implementation. To address these issues in our future work, we would like to further point out the following:

1) Despite its relatively high complexity, OEED is only one embodiment of our content-adaptive deblocking

Table 1
Objective quality improvement (BD-rate saving percentage) over H.264/AVC deblocking on the HEVC test set, $QP=27$, 32, 37, and 42.

HEVC test sequence	Proposed (all-intra)			Intra-predictive (all-intra)			Proposed (hierarchical-B)		
	Y	*U*	*V*	*Y*	*U*	*V*	*Y*	*U*	*V*
Class A 4 K									
S01 Traffic	−0.2	−0.6	−0.8	−0.2	−0.1	−0.4	0.0	−0.8	0.0
S02 PeopleOnStreet	−0.2	−1.4	−0.6	−0.2	−0.4	−0.2	0.2	−0.4	−0.4
Class B 1080p									
S03 Kimono	0.1	−0.5	−0.6	0.1	0.0	0.0	−0.1	−1.4	−0.7
S04 ParkScene	0.3	−0.7	−1.0	0.1	0.0	0.0	0.2	−1.0	−2.7
S05 Cactus	0.3	−0.8	−0.4	0.0	−0.2	0.0	0.1	−0.2	0.1
S06 BasketballDrive	0.6	−1.4	−0.5	−0.1	−0.3	−0.2	0.1	0.5	−1.1
S07 BQTerrace	0.3	−1.9	−1.4	0.0	−0.5	−0.3	0.6	−0.8	0.6
Class C WVGA									
S08 BasketballDrill	−1.7	−3.1	−3.8	−1.3	−1.4	−1.6	−0.7	−2.1	−3.0
S09 BQMall	0.1	−0.4	0.0	−0.1	−0.1	0.0	0.3	−0.4	−0.4
S10 PartyScene	0.4	−0.3	−0.4	0.1	0.0	−0.1	0.1	−0.7	0.8
S11 RaceHorses	−0.2	−0.8	−1.3	−0.2	−0.3	−0.6	−0.2	−0.7	−0.8
Class D WQVGA									
S12 BasketballPass	0.2	−0.5	−0.3	0.0	−0.1	−0.1	0.4	−0.4	0.1
S13 BQSquare	0.4	−0.4	−0.7	0.0	0.0	0.0	0.3	−0.3	−0.4
S14 BlowingBubbles	−0.1	−0.6	−0.7	−0.1	−0.1	−0.3	−0.1	0.1	0.1
S15 RaceHorses	−0.2	−0.9	−1.0	−0.1	−0.2	−0.4	−0.1	−0.1	−0.4
Class E 720p									
S16 Vidyo1	0.0	−1.7	−1.2	−0.3	−0.4	0.0	–	–	–
S17 Vidyo3	0.0	−1.4	−1.1	0.2	0.0	−0.1	–	–	–
S18 Vidyo4	0.0	−1.5	−1.3	−0.1	0.0	−0.1	–	–	–
Average									
Class A	−0.2	−1.0	−0.7	−0.2	−0.3	−0.3	0.1	−0.6	−0.2
Class B	0.3	−1.1	−0.8	0.0	−0.2	−0.1	0.2	−0.6	−0.8
Class C	−0.3	−1.1	−1.4	−0.4	−0.4	−0.6	−0.1	−1.0	−0.8
Class D	0.1	−0.6	−0.7	−0.1	−0.1	−0.2	0.1	−0.2	−0.2
Class E	0.0	−1.5	−1.2	−0.1	−0.1	−0.1	–	–	–
Average									
All	0.0	−1.0	−1.0	−0.1	−0.2	−0.2	0.1	−0.6	−0.5

scheme, which is employed to exploit the potential of our proposed method. We will continue to seek for simpler yet effective substitute of OEED.

2) Memory bandwidth is an important issue to be considered in practical implementation. Given a limited memory bandwidth, OEED can be readily performed in a more flexible way, e.g., slice-wise, macroblock-wise (for H.264/AVC), or coding-unit-wise (for HEVC), by treating each unit as a small image.
3) The increase of line buffer is an inevitable cost for directional or adaptive deblocking. However, with a slight change on the filter design, the line buffer increase can be largely diminished. Take the proposed extra smoothing filter for example; when it is performed on a horizontal boundary, the filter can be applied to the lowermost 4 pixels in the top block and the uppermost 12 pixels in the bottom block. In this way, the line buffer remains the same with traditional H.264/AVC deblocking.
4) In the current design, our algorithm deals with 4×4, 8×8, and 16×16 transforms. For larger transform (e.g., 32×32) enabled in HEVC, the proposed extra smoothing filter can be readily applied by extending the filter length and choosing a proper variance threshold. A subsequent related work can be found in [10].

5. Conclusion

In this paper, a content-adaptive deblocking method is proposed to improve the visual quality of block-based DCT compressed video, which is highly demanded in the forthcoming HEVC. Experimental results demonstrate the effectiveness of our method. Due to its full compatibility with the current H.264/AVC standard, the proposed deblocking algorithm can be readily integrated in the next generation video coding standard, with further optimization on the implementation complexity.

References

[1] P. List, A. Joch, J. Lainema, G. Bjontegaard, M. Karczewicz, Adaptive deblocking filter, IEEE Transactions on Circuits and Systems for Video Technology 13 (7) (2003) 614–619.
[2] W.C. Fong, S.C. Chan, A. Nallanathan, K.L. Ho, Integer lapped transforms and their applications to image coding, IEEE Transactions on Image Processing 11 (10) (2002) 1152–1159.

[3] A. Zakhor, Iterative procedures for reduction of blocking effects in transform image coding, IEEE Transactions on Circuits and Systems for Video Technology 2 (1) (1992) 91–95.
[4] T.P. O'Rourke, R.L. Stevenson, Improved image decompression for reduced transform coding artifacts, IEEE Transactions on Circuits and Systems for Video Technology 5 (6) (1995) 490–499.
[5] Z. Xiong, M.T. Orchard, Y.Q. Zhang, A deblocking algorithm for jpeg compressed images using overcomplete wavelet representations, IEEE Transactions on Circuits and Systems for Video Technology 7 (2) (1997) 433–437.
[6] H.S. Kong, A. Vetro, and H. Sun, EDGE map guided adaptive postfilter for blocking and ringing artifacts removal in: Proceedings of the IEEE International Symposium on Circuits and Systems, 2004.
[7] Y.M. Huang, J.J. Leou, and M.H. Cheng, A post deblocking filter for H.264 video, in: Proceedings of the International Conference on Computer Communications and Networks, 2007.
[8] J. Jeong, S. Kim, Y.G. Kim, Y. Choi, Y. Choe, A directional deblocking filter based on intra prediction for H.264/AVC, IEICE Electronics Express 6 (12) (2009) 864–869.
[9] P. Perona, and J. Malik, Detecting and localizing edges composed of steps, peaks and roofs, in: Proceedings of the IEEE International Conference on Computer Vision, 1990.
[10] Z. Shi, X. Sun, J. Xu, CE12, Subset 1: Report of Deblocking For Large Size Blocks, JCTVC-F198, July 2011, Torino, IT.
[11] G. Zhai, W. Zhang, X. Yang, W. Lin, Y. Xu, No-reference noticeable blockiness estimation in images, Signal Processing Image Communication 23 (6) (2008) 417–432.
[12] I. Kirenko, R. Muijs, L. Shao, Coding artifact reduction using non-reference block grid visibility measure, in: Proceedings of the IEEE International Conference on Multimedia & Expo, 2006.
[13] G. Zhai, W. Zhang, X. Yang, W. Lin, Y. Xu, Efficient image deblocking based on post filtering in shifted windows, IEEE Transactions on Circuits and Systems for Video Technology 18 (1) (2008) 122–126.
[14] A. Chetouani, G. Mostafaoui, A. Beghdadi, Deblocking filtering method using a perceptual map, Signal Processing Image Communication 25 (7) (2010) 527–534.