Adaptive video-blocking artifact removal in discrete Hadamard transform domain

Jun Xu, Shibao Zheng, and Xiaokang Yang
Shanghai Jiao Tong University,
Institute of Image Communication and Information
Processing, No.800 Dongchuan Road,
Shanghai 200240, China
E-mail: xujun05@sjtu.edu.cn

Abstract. We present a novel postprocessing algorithm for blocking artifact removal in the discrete Hadamard transform (DHT) domain, which does not require prior knowledge of quantization parameters and features low computational complexity. All block-based video coding methods suffer annoying blocking artifacts at low bit rates. We first acquire edge information for the frame, then calculate block activities from DHT coefficients so as to classify smooth and coarse regions adaptively. Blocking artifacts are adaptively filtered in the DHT domain according to block activities. Experimental results show that the proposed method is able to remove blocking artifacts effectively while preserving image details and object edges well, and that it achieves better visual quality compared with other methods.
[DOI: 10.1117/1.2280609]

Subject terms: discrete Hadamard transform; video deblocking; postprocessing.

Paper 060042LR received Jan. 18, 2006; revised manuscript received May 22, 2006; accepted for publication May 30, 2006; published online Aug. 16, 2006.

1 Introduction

Due to the coarse quantization of the block-based discrete cosine transform (DCT) coefficients in prevalent video compression techniques, neighboring blocks might have noncontinuous border effects that are particularly eye catching at low bit rates. The so-called postprocessing scheme is designed to reduce blocking artifacts, and thus improve the subjective quality of the video. Many deblocking methods have been proposed. Those methods can be roughly classified into three categories according to their operating domain, namely, in the spatial domain,[1–4] in the DCT domain,[5,6] and in the wavelet transform domain.[7] The algorithms operating in the spatial domain are usually simple but their results are not very satisfactory. The algorithms operating in the DCT or the wavelet domain yield better results, but the transform itself is complex and is not easy for hardware implementation. Many methods utilized prior knowledge of quantization parameters,[2,4–6] but the deblocking methods without knowledge of quantization parameters are more versatile in practical applications.

In this letter, we propose an adaptive postprocessing algorithm without requiring quantization parameters, which preserves object edges and image details while reducing blocking artifacts significantly. The proposed method is based on simple but effective discrete Hadamard transform (DHT), thus, the computational complexity of the algorithm is quite low. Furthermore, the algorithm exploits some cues of the human visual system (HVS) implicitly, and thus improves the visual quality well.

0091-3286/2006/$22.00 © 2006 SPIE

2 Deblocking Algorithm

Figure 1 shows a flowchart of the proposed deblocking algorithm. The algorithm takes the decoded $YCbCr$ sequences as input. To preserve the object edges, we use an edge detection module to acquire the edge information. We then calculate the local activity of each block employing a DHT. The local activity is used to adaptively control the size of a low-pass filter in the DHT domain. Finally, we perform inverse discrete Hadamard transform (IDHT) to acquire output.

2.1 *Adaptive Edge Detection*

Adaptive edge detection consists of two steps, direct current (DC) image generation and edge detection of the DC image. First, the input frame is divided into 4 × 4 nonoverlapping blocks. The mean value of every 4 × 4 block is calculated and the DC image is formed as the 2-D array of the mean values. Second, the Sobel operator is employed to differentiate the edges and the monotone areas in the DC image. The edge pixel in the DC image is then identified with an adaptive threshold[6] given by

$$T=\sqrt{3}\left[\sum_{i=1}^{N_r}\sum_{j=1}^{N_c}\nabla X(i,j)/(N_rN_c)\right], \tag{1}$$

where $X(i,j)$ is the pixel in the DC image; N_r and N_c are the numbers of rows and columns in the DC image, which equals 1/4 the number of rows and columns in the original frame; and ∇ is the Sobel edge detector. If $\nabla X(i,j)>T$, then pixel $X(i,j)$ is an edge pixel in the DC image that corresponds to a 4 × 4 block in the original frame which would not be filtered.

2.2 *DHT*

Here we adopt a sequence-ordered 4 × 4 Hadamard matrix:

$$\mathbf{H}=\frac{1}{2}\begin{pmatrix}1&1&1&1\\1&1&-1&-1\\1&-1&-1&1\\1&-1&1&-1\end{pmatrix}. \tag{2}$$

The DHT is computed exactly in integer arithmetic, thus avoiding the inverse transform mismatch problems of DCT and minimizing computational complexity significantly. The IDHT matrix is identical to Eq. (2), so the transform and inverse transform module are reusable when implemented in hardware.

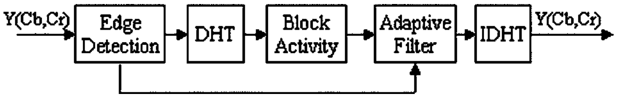

Fig. 1 System structure diagram.

2.3 *Block Activity*

If we let $H_{m,n}(u,v)$ be the 4×4 DHT coefficients of the block with top-left point (m,n), the value of activity can be calculated as

$$\text{Act} = \frac{\sum_{u=0}^{3}\sum_{v=0}^{3} |H_{m,n}(u,v)\cdot \text{Mask}(u,v)|}{H_{m,n}(0,0)} - 1.0, \tag{3}$$

with

$$\text{Mask} = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_1 & a_2 & a_3 & a_4 \\ a_2 & a_3 & a_4 & a_5 \\ a_3 & a_4 & a_5 & a_6 \end{pmatrix}. \tag{4}$$

The elements a_i, $i=0,1,\ldots,6$, should increase rapidly with high-frequency components. In this implementation, we set $a_i=2^i$ by compromising the precision of activity estimation and the convenience of hardware implementation. A block with a large activity value corresponds to the coarse area or edges where the blocking artifacts might be masked and not visually detectable. A block with a small activity value stands for a smooth region. Since the DC coefficient is proportional to the local mean luminance of a block, the normalization by the DC coefficient $H_{m,n}(0,0)$ implicitly exploits the local luminance adaptation in line with Weber's law.[8]

2.4 *Adaptive Filter*

Motivated by the fact that the blocking artifacts in smooth regions are more eye-catching, while preserving image details and object edges, the adaptive filter with a $(2h+1)\times(2h+1)$ variable size window is mathematically formulated as

$$\hat{H}_{m,n}(u,v) = \frac{1}{W}\sum_{k=-h}^{h}\sum_{l=-h}^{h} \omega_{k,l}H_{m+k,n+l}(u,v), \tag{5}$$

where $\hat{H}_{m,n}(u,v)$ are filtered coefficients in the DHT block; $\omega_{k,l}$ are adaptive filter weights for neighboring DHT blocks; $H_{m+k,n+l}(u,v)$ are DHT coefficients of the block with its top-left point at row $(m+k)$ and column $(n+l)$; and W is the sum of weights of and given by

$$W = \sum_{k=-h}^{h}\sum_{l=-h}^{h} \omega_{k,l}, \tag{6}$$

with

$$\omega_{k,l} = \begin{cases} 3.0 & (k,l)=(0,0) \\ 1.0 & \text{otherwise}. \end{cases} \tag{7}$$

The filtering window size adapts to its activity in a piecewise way:

$$h = \begin{cases} 3 & \text{Act} < T/250 \\ 2 & T/250 \le \text{Act} \le T/50 \\ 1 & \text{Act} > T/50. \end{cases} \tag{8}$$

Equation (8) adapts well to blocks with different activities. For blocks of low activities where the blocking effects appear to be more visible, the filter window is enlarged to remove the blocking artifacts. On the contrary, the blocks with high activities are far less blurred with a small window size.

To avoid overfiltering a block centered at (m,n) in texture area, its neighboring block located at $(m+k,n+l)$ is excluded from the filtering operation if Eq. (9) is satisfied.

$$\frac{|H_{m,n}(0,0) - H_{m+k,n+l}(0,0)|}{H_{m,n}(0,0)} > \eta. \tag{9}$$

Note that η is set to be 0.1 empirically in this paper.

3 Experimental Results

The proposed algorithm was applied to video sequences compressed by the SVC codec from Microsoft Research Asia that could cover all testing points of core experiments 1 (Ref. 9). The Microsoft SVC coding scheme is based on block-based motion-compensated temporal filtering followed by 2-D spatial wavelet decomposition.

The "Foreman_352×288_15_96" is a "Foreman" sequence decoded with an image size of 352×288 at frame rate of 15 frames/s, and a bit rate of 96 kbits/s is taken as input. To evaluate the performance of the proposed algorithm, three existing methods (see Refs. 1, 3, and 7) without using prior knowledge of quantization parameters are compared. Since some reference methods are designed for postprocessing of images, for the fairness of comparison, only Y components are used for comparison. Their postprocessed images are given in Fig. 2(b)–2(d). From Fig. 2, it is

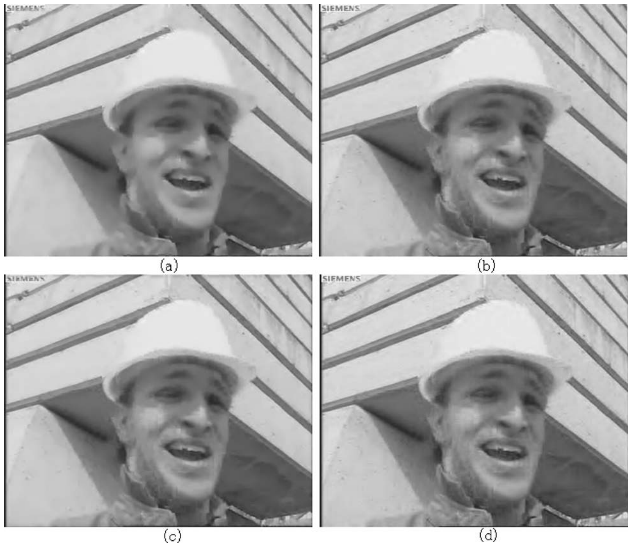

Fig. 2 Comparison of the eighth frame of the "Foreman" sequence postprocessed by four different methods: (a) postprocessed by the proposed method, (b) postprocessed in the spatial domain by the method of Ref. 1, (c) postprocessed in the wavelet domain by the method of Ref. 7, and (d) postprocessed by the H.263 method.[3]

Table 1 PSNR-Y comparison in decibels.

"Foreman" Frame	Decoded Video	Pixel[1]	Wavelet[7]	H.263 (Ref. 3)	Proposed
8	32.1991	32.3428	30.6064	32.0186	30.9718
15	32.6702	32.8170	30.8215	32.4060	31.1641
21	32.0794	32.3156	30.8771	31.9253	31.1429

Table 2 DMOS comparison.

Sequence	Decoded Video	H.263 (Ref. 3)	Proposed
Foreman_352 ×288_15_96	38.02	37.41	26.02
Crew_352 ×288_15_128	40.59	40.33	29.97
Carphone_352 ×288_30_72	35.47	34.29	28.17
Hall_monitor_352 ×288_30_48	39.13	38.02	30.34
Akiyo_352 ×288_30_36	34.51	32.84	29.86

evident that our proposed method is able to outperform the compared methods by removing blocking artifacts effectively while retaining edge sharpness. It validates the adaptive filtering process in the DHT domain.

In the preceding experiments for one frame, our method on average takes 0.7 s and methods in Refs. 1 and 3 all take no more than 1 s, while the wavelet-based method in Ref. 7 consumes 14 min. We chose the video with image size of 1920 × 1080 as input to validate the simplicity of DHT. The method adopting DCT in Ref. 5 takes about 12 min for a frame, while our method on average takes no more than 5 s. All experiments were run on a 1.8-GHz Pentium PC. Of course, the evaluated algorithms were not optimized for real-time applications. Thus, the data of computational complexity given here shows only that the proposed method may be closer to practical applications from the viewpoint of hardware simplicity.

Table 1 gives the peak SNR (PSNR)-Y results comparing the objective quality. Although we see that PSNR is not a good measure to evaluate such techniques, our proposed approach achieves higher PSNR gain than the method in Ref. 7.

Because human eyes are the final judges of video, we made a subjective test of some deblocking results according to double stimulus continuous quality scale method suggested by ITU-R BT.500-10 (Ref. 10). The mean opinion scores (MOS) were rescaled to a range of 0 to 100. The difference mean opinion scores (DMOS) were calculated as the difference between the original video and the test video. The DMOS of the method in Ref. 3 and the proposed method are compared in Table 2, which shows that the subjective rating of the proposed method is significantly better.

4 Conclusions

A postprocessing algorithm for blocking artifact removal was proposed in the DHT domain. The algorithm can remove blocking artifacts effectively while preserving image details and object edges well. It is a versatile method that does not require prior knowledge of quantization parameters and features low computational complexity. Since the basic operation unit in our method is a 4 × 4 block and DHT is inherently simple and computationally efficient, the algorithm is easy for hardware implementation and promising for real-time video postprocessing utilized in handheld devices.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant No. 60502034 and the Shanghai Rising-Star Program under Grant No. 05QMX1435.

References

1. R. Castagno, S. Marsi, and G. Ramponi, "A simple algorithm for the reduction of blocking artifacts in images and its implementation," *IEEE Trans. Consum. Electron.* **44**(3), 1062–1070 (1998).
2. ISO/IEC, "Mpeg4 video verification model version 18.0," ISO/IEC JTC1/SC29/WG11 Document N3908 (2001).
3. ITU-T, "Video coding for low bit rate communication," ITU-T Video Coding Experts Group, ITU-T Rec. H. 263 Std. (1998).
4. P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 614–619 (2003).
5. T. Chen, H. R. Wu, and B. Qiu, "Adaptive postifiltering of transform coefficients for the reduction of blocking artifacts," *IEEE Trans. Circuits Syst. Video Technol.* **11**(5), 594–602 (2001).
6. C. Wang, W. J. Zhang, and X. Z. Fang, "Adaptive reduction of blocking artifacts in DCT domain for highly compressed images," *IEEE Trans. Consum. Electron.* **50**(2), 647–654 (2004).
7. Z. Xiong, M. T. Orchard, and Y. Q. Zhang, "Deblocking algorithm for JPEG compressed images using overcomplete wavelet representations," *IEEE Trans. Circuits Syst. Video Technol.* **7**(2), 433–437 (1997).
8. W. Schreiber, *Fundamentals of Electronic Imaging Systems*, Springer Verlag, New York (1993).
9. M. van der Schaar and M. Domański, "Description of core experiments in SVC," N673, ISO/IEC JTC1/SC29/WG11, March 2004, Munich, Germany.
10. ITU-R, Recommendation BT.500-10, "Methodology for the subjective assessment of the quality of television pictures," March 2000.