POSTPROCESSING OF BLOCK-CODED VIDEOS FOR DEFLICKER AND DEBLOCKING

Jie Ren, Jiaying Liu, Mading Li and Zongming Guo*

Institute of Computer Science and Technology, Peking University, Beijing, China 100871

ABSTRACT

In this paper, we propose a novel postprocessing method to suppress both the flickering and blocking artifacts in block-coded videos. For reducing the flickering effect between adjacent frames, we propose an adaptive multi-scale motion filtering method to maintain the motion coherence of processed video. For blocking artifacts suppression, we adopt a patch-based scheme in which similar patches are grouped in a spatio-temporal domain and each patch group is recovered by solving a low rank matrix completion problem. Experimental results show that the proposed method can significantly reduce the flickering and blocking artifacts in the decoded videos.

Index Terms— Video deblocking, deflicker, low rank matrix completion, adaptive multi-scale motion filtering

1. INTRODUCTION

Block-based Discrete Cosine Transform (BDCT) has been widely used in most image/video compression standards including MPEG and H.264 [1] because of its energy compacting property and relative ease of implementation. However, as each block is transformed and quantized independently, blocking artifacts appear along the block boundaries. Therefore, postprocessing of the block-coded videos is necessary to improve the visual quality of decoded videos.

One intuitive way to deal with the video deblocking problem is to process each frame in the decoded video separately therefore image deblocking algorithms and techniques can be utilized to fulfill the video deblocking task. Generally, there are two categories of image deblocking artifacts suppression techniques existing in the literature: *image enhancement-based* and *image restoration-based*. Image enhancement-based methods focused on reducing the visual perception of blocking artifacts and applied spatial or transform-domain filtering to smooth the block boundary [2]. Image restoration-based methods treated the blocking artifacts reduction as an inverse restoration problem of the compressed image, *e.g.*, POCS-based [3] and TV-based method [4]. In recent years, patch-based methods have been proved to be effective in image restoration problems [5, 6]. The success of patch-based methods mainly rely on the fact that natural visual signals such as images and videos have a large amount of redundant information about local image structures. These redundant information can be well utilized as prior model to restrain the ill-posed inverse restoration problems.

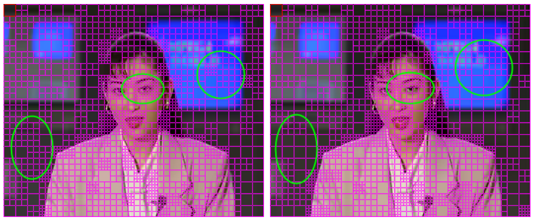
Comparing to image deblocking, video deblocking aims at removing the blocking artifacts from all frames of a video efficiently. Although the image deblocking techniques can greatly reduce the blocking artifacts in each frame, the whole processed video sequences commonly have low visual quality. The flickering effect between adjacent frames seriously degrades the visual experience of the processed video, which is mainly caused by the independent quantization of the similar blocks on a motion trajectory. Therefore, it is reasonable to reduce the flickering effect between adjacent frames while suppressing the blocking artifacts within each frame to achieve the best coherent visual quality.

In this paper, we propose a novel postprocessing method to address both of the above two problems: video deflicker and deblocking. The proposed method exploits two types of redundant information in natural video sequences: motion coherence and self-similarities of local image structures. Motion information is used to efficiently reduce the flickering effect between adjacent frames, in the meanwhile the self-similarities are utilized to greatly suppress the blocking artifacts. Specifically, the self-similarities are explored in a spatio-temporal domain and the deblocking problem of removing the blocking artifacts from a stack of grouped patches is converted into a low rank matrix completion problem, which is solved by minimizing the nuclear norm of the matrix with linear constraints.

The rest of this paper is organized as follows: Section 2 presents the proposed deflicker and deblocking algorithm in details. Section 2.1 presents an adaptive multi-scale motion filtering to reduce the flickering effect between adjacent frames. In Section 2.2, a deblocking algorithm utilizing the self-similarities is presented in a low rank matrix completion framework. Experimental results demonstrated in Section 3 show the superior performance of the proposed method. The blocking artifacts are significantly reduced while the motion coherence between adjacent frames is well maintained, which

*Corresponding author

This work was supported by National Natural Science Foundation of China under contract No. 61071082, National Basic Research Program (973 Program) of China under contract No. 2009CB320907 and National Development and Reform Commission High-tech Program of China under Grant No. [2010]3044.



(a) Frame 12 (b) Frame 13

Fig. 1. Illustration of the mode configuration saltation of adjacent frames in Video *Akiyo*. (a) Mode configuration of 12$^{\text{th}}$ frame; (b) Mode configuration of 13$^{\text{th}}$ frame.

makes the overall outputs with better visual experience. Finally, Section 4 concludes this paper.

2. VIDEO DEFLICKER AND DEBLOCKING

For block-coded videos, two types of artifacts may be introduced due to the independent transformation and quantization of each block. Firstly, the similar blocks on the motion trajectory between adjacent frames may be dissimilarly quantized as different modes may be selected for each independent block coding, as shown in the marked regions in Fig.1. It will introduce the flickering artifacts into the decoded video and severely degrade the visual experience of the overall decoded video. Secondly, adjacent blocks in spatial positions within the same frame will have obvious grid noise along the block boundaries, which is referred to as blocking artifacts. In this section, we propose a novel postprocessing method to address both of the flicker and blocking artifacts in the decoded video.

2.1. Deflicker by adaptive multi-scale motion filtering

The flickering effect can be well reduced by filtering the similar patches along the motion trajectory between adjacent frames. However, finding the "true" motion trajectory is not a trivial task because the pixel intensities of the underlying similar blocks on the motion trajectory may be different, which is caused by the independent quantization of blocks. In the meanwhile, the motion filtering will smooth the underlying object boundaries and textures structure, which will reduce the visual quality of each processed frame. Therefore, it needs some special treatment to deal with the flickering artifacts while maintaining the visual quality.

In this paper, we adopt an adaptive multi-scale motion filtering techniques to address the above issues. Let $B_{curr\times32}$ be a 32×32 block image of the current processing frame. Let $B_{forw\times32}$ and $B_{back\times32}$ be the most similar blocks to $B_{curr\times32}$ in the forward and backward adjacent frames, respectively. We check the Sum of Absolute Difference (SAD) between these two blocks and $B_{curr\times32}$. Let us take the forward block $B_{forw\times32}$ as an example. If $SAD(B_{forw\times32}, B_{curr\times32}) > B_{thresh}$ where B_{thresh} is a predefined threshold that controls the degree of block similarity, the large-scale 32×32 block is divided into four small-scale 16×16 blocks. Then, for each small-scale block, we search its most similar block in the forward frames. The founded 16×16 blocks in the adjacent frame are assembled into a new 32×32 block, denoted as $B'_{forw\times32}$. Similar process is applied to the backward block matching.

The founded blocks $B'_{forw\times32}$ and $B'_{back\times32}$ contain the redundant motion information of the current block $B_{curr\times32}$. We can simply apply a motion filtering by

$$B'_{curr\times32} = \gamma_1 B'_{forw\times32} + \gamma_2 B'_{back\times32} + \gamma_3 B_{curr\times32}, \quad (1)$$

where $0\le\gamma_1,\gamma_2,\gamma_3\le1$ and $\gamma_1+\gamma_2+\gamma_3=1$.

However, the direct motion filtering on the 32×32 block will smooth the underlying edge and texture structures and degrade the deblocking performance described in the following subsection. Therefore, we adopt an adaptive filtering method that classifies the overlapping small patches (8×8 patch size) in the 32×32 large-scale block into two categories: smooth patch and feature patch. Smooth patches with low variances will be filtered by (1) while feature patches will not be processed and leave them in the following deblocking procedure.

2.2. Deblocking by low-rank matrix completion

Following the deflicker step, we propose to deblocking each frame of the filtered video by a low-rank matrix completion method. The proposed method firstly construct a patch group for each patch in the spatio-temporal domain. Then low-rank matrix completion solved by minimizing the nuclear norm of the matrix with linear constraints is performed on each patch group to reduce the blocking artifacts. After deblocking, each single patch in the group will be relocated to its corresponding position in the video sequence to produce the overall deblocked video.

2.2.1. Spatio-temporal patch grouping

Considering an image patch $y_{j,k}$ of size $\sqrt{n}\times\sqrt{n}$ centered at position j in frame Y_k. We set this patch as a reference patch and search m patches that are similar to $y_{j,k}$ within the spatio-temporal volume of video $\mathcal{Y}$. The index of these patches are grouped into $G_{j,k}$, which is $G_{j,k} = \left\{(s,t)|T\ge\|y_{s,t}-y_{j,k}\|_F^2\right\}$, where the parameter $T>0$ controls the minimum degree of similarity among patches and $\|\cdot\|_F^2$ denotes the Frobenius norm of matrix. If we represent each patch $y_{j,k}$ as a vector $\mathbf{y}_{j,k}\in\mathbb{R}^n$ by concatenating all columns of the patch, a $n\times m$ matrix $Y_{G_{j,k}}$ can be defined by

$$Y_{G_{j,k}} = \left(\mathbf{y}_{G_{j,k}(1)}, \mathbf{y}_{G_{j,k}(2)}, \cdots, \mathbf{y}_{G_{j,k}(m)}\right). \quad (2)$$

2.2.2. Low-rank matrix completion on patch groups

We consider the deblocking problem as a low-rank matrix completion problem from the noisy and incomplete observation $Y_{G_{j,k}}$. For the sake of compact description, we omit the subscript $\{j,k\}$. The set of missing elements of Y_G have two subsets: one is the pixel collection along the block boundaries caused by the block-based video coding. The other one is the pixel set in which the pixel intensity differs from the mean of the corresponding row vector by an amount larger than a predefined threshold. The set of all the remained pixels of Y_G is denoted as Ω.

The low-rank completion problem can be solved by the following optimization

$$\min_{X_G} \text{rank}\,(X_G), \text{s.t.}\ \|\mathcal{P}_\Omega(X_G - Y_G)\|_F^2 < m \cdot \hat{\epsilon}^2, \tag{3}$$

where $\hat{\epsilon}$ is the estimated standard deviation of noise, which is obtained by calculating quantization noise variance, and $\mathcal{P}_\Omega$ is the projection operator of Ω that extracts only the elements in Ω and sets other elements equal to zero.

However, the low-rank optimization problem in (3) is a NP-hard minimization problem. Therefore, it is relaxed to a convex optimization problem instead [7]:

$$\min_{X_G} \|X_G\|_* + \lambda \|\mathcal{P}_\Omega(X_G - Y_G)\|_F^2, \tag{4}$$

where $\|X_G\|_*$ represents the nuclear norm of matrix X_G and is defined as $\|X_G\|_* := \sum_i \sigma_i(X_G)$ in which $\sigma_i(X_G)$ denotes the i^{th} largest singular value of X_G.

There are many numerical optimization algorithms that can be utilized to solve the convex optimization problem in (4). In this paper, we choose the Singular Value Thresholding (SVT) algorithm in [8] mainly because of its simplicity and ease of implementation.

After each patch group is jointly deblocked, each single patch in the group will be relocated to its corresponding position in the video sequence, and the final deblocked video can be obtained by overlapping the patches and averaging the pixel values in the overlapped region.

3. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed method on video deflicker and deblocking, we conduct experiments on four typical test videos compressed at three different QPs by the H.264 encoder. These four test videos includes *Akiyo*, *City*, *Foreman* and *Stefan*. Three quantization steps, QP = $32, 37, 42$, are selected to produce different levels of blocking artifacts in the compressed videos, and the corresponding quantization steps are $Qstep = 26, 44, 80$, respectively. For estimating the compression noise variance $\hat{\epsilon}^2$ from the input videos for the deblocking algorithm in (3), we choose the empirical formula in [9]: $\hat{\epsilon}^2 = 0.69 \cdot (Qstep)^{1.3}$. Therefore, for the above three QPs, the estimated noise standard deviations are $\hat{\epsilon} = 6.9$, 9.7 and 14.3, respectively.

In the deblocking algorithm, we set the patch size to be 6×6, and overlap the patches with 5 pixels along each direction. For each reference patch, 40 similar patches are grouped to construct the data matrix in (2).

Table 1 shows the mean PSNR results of the four test videos for 20 video frames. We compare the proposed post-processing method to the standard deblocking filter in [2]. The results of only applying the low-rank matrix completion deblocking algorithm is denoted as *DB*, and the results of the final proposed method combining the deflicker and deblocking algorithm is denoted as *DF+DB*. In Fig.2, we plot the PSNR curves of different methods for 20 frames in details. For objective quality assessment in terms of PSNR, the proposed DF+DB algorithm is superior to both the H.264 filter [2] and the deblocking algorithm only (DB).

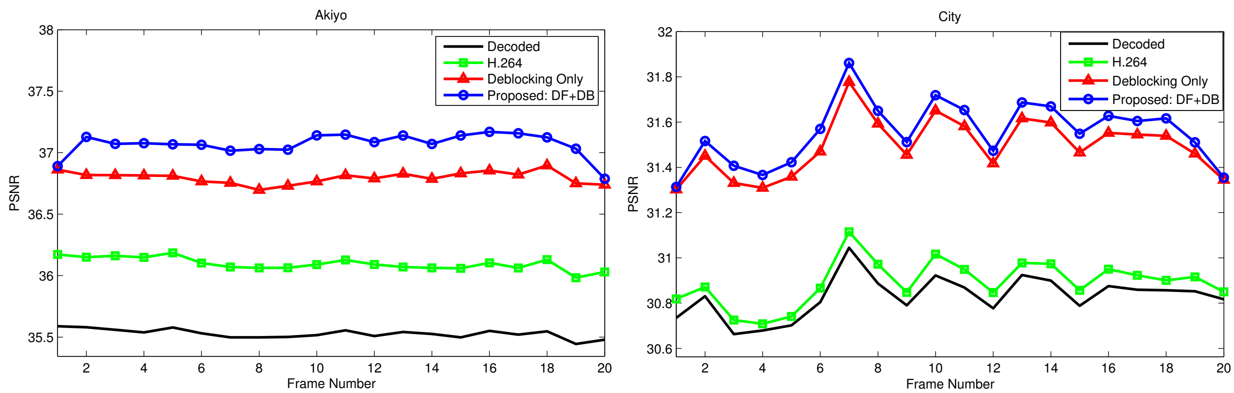
Table 1. Summary of average PSNR values (dB) of different methods on three quantization parameters (QP).

Video	QP	Decoded	[2]	DB	DF+DB
Akiyo	32	38.59	39.03	39.97	**40.12**
	37	35.53	36.10	36.80	**37.07**
	42	32.28	32.84	33.26	**33.54**
City	32	34.09	34.13	**34.91**	34.90
	37	30.83	30.89	31.49	**31.55**
	42	27.83	27.95	28.30	**28.38**
Foreman	32	36.14	36.37	**37.30**	37.22
	37	33.29	33.64	34.54	**34.57**
	42	30.24	30.69	31.30	**31.39**
Stefan	32	34.81	34.94	**36.13**	36.12
	37	31.07	31.20	32.26	**32.30**
	42	27.30	27.41	28.17	**28.22**

To further demonstrate the deflicker performance of the proposed method, Fig. 3 shows five frames of the processed video results on *Akiyo* video with QP = 37. Although the H.264 filter and the deblocking only algorithm can greatly reduce the blocking artifacts, both of them ignore the motion coherence between adjacent frames. Please notice the flickering effect on the Akiyo's face especially on the nose region. In contrast, the proposed DF+DB algorithm can significantly mitigate both the flickering and blocking artifacts in the decoded video. More results are presented in our website: http://www.icst.pku.edu.cn/course/icb/Video-DFDB.html.

4. CONCLUSION

This paper aims to reduce both the flickering effect and blocking artifacts in the block-coded videos. Two types of redundant information including the motion coherence and self-similarities are utilized. An adaptive multi-scale motion filtering is presented to alleviate the flickering effect between adjacent frames while maintaining the underlying object structures. For deblocking, a low rank matrix completion based algorithm is proposed to fully explore the self-similarity property in the spatio-temporal domain of video. Experimental results demonstrate that the proposed approach can significantly reduce the blocking artifacts while maintaining preferable motion coherence between adjacent frames.



(a) *Akiyo* (b) *City*

Fig. 2. PSNR performance curves of video *Akiyo* and *City* (20 frames, QP = 37). (a) *Akiyo*; (b) *City*.

Fig. 3. Post-processing results of *Akiyo* video sequence (Frame No.1-5, QP = 37). From top to bottom: original frames, decoded frames, deblocked frames by the H.264 filter [2], deblocked frames by low-rank matrix completion deblocking method only and the deblocked frames by the proposed deflicker and deblocking method (DF+DB).

5. REFERENCES

[1] A.W.-C. Liew and Hong Yan, "Blocking artifacts suppression in block-coded images using overcomplete wavelet representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 4, pp. 450–461, Apr. 2004.

[2] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, July 2003.

[3] Yongyi Yang, N.P. Galatsanos, and A.K. Katsaggelos, "Projection-based spatially adaptive reconstruction of block-transform compressed images," *IEEE Trans. Image Processing*, vol. 4, no. 7, pp. 896–908, July 1995.

[4] Francois Alter, Sylvain Durand, and Jacques Froment, "Adapted total variation for artifact free decompression of jpeg images," *Journal of Mathematical Imaging and Vision*, vol. 23, no. 2, pp. 199–211, 2005.

[5] K.. Dabov, A.. Foi, V.. Katkovnik, and K.. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[6] X. Li, W. Dong, and G. Shi, "Nonlocal image restoration with bilateral variance estimation: a low-rank approach," *IEEE Trans. Image Processing*, vol. PP, no. 99, pp. 1, 2012.

[7] E. J. Candes and B. Recht, "Exact matrix completion via comvex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, Dec. 2009.

[8] Jian-Feng Cai, Emmanuel J. Cands, and Zuowei Shen, "A singular value thresholding algorithm for matrix completion," *Siam Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[9] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Processing*, vol. 16, no. 5, pp. 1395–1411, May 2007.