

Journal of Zhejiang University SCIENCE A
 ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
 www.zju.edu.cn/jzus; www.springerlink.com
 E-mail: jzus@zju.edu.cn



An adaptive preprocessing algorithm for low bitrate video coding*

LI Mao-quan^{†1,2}, XU Zheng-quan²

⁽¹⁾*School of Electronic Information, Wuhan University, Wuhan 430079, China*

⁽²⁾*Multimedia Communication Engineering Center, State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China*

[†]E-mail: limaoquan@263.net

Received Mar. 15, 2006; revision accepted July 10, 2006

Abstract: At low bitrate, all block discrete cosine transform (BDCT) based video coding algorithms suffer from visible blocking and ringing artifacts in the reconstructed images because the quantization is too coarse and high frequency DCT coefficients are inclined to be quantized to zeros. Preprocessing algorithms can enhance coding efficiency and thus reduce the likelihood of blocking artifacts and ringing artifacts generated in the video coding process by applying a low-pass filter before video encoding to remove some relatively insignificant high frequent components. In this paper, we introduce a new adaptive preprocessing algorithm, which employs an improved bilateral filter to provide adaptive edge-preserving low-pass filtering which is adjusted according to the quantization parameters. Whether at low or high bit rate, the preprocessing can provide proper filtering to make the video encoder more efficient and have better reconstructed image quality. Experimental results demonstrate that our proposed preprocessing algorithm can significantly improve both subjective and objective quality.

Key words: Blocking artifact, Quantization parameter, Video preprocessing, Bilateral filtering

doi: 10.1631/jzus.2006.A2057

Document code: A

CLC number: TN919.8

INTRODUCTION

Currently, main video coding algorithms, such as H.26x and MPEG families are all based on block discrete cosine transform (BDCT). At low bit rate, all these BDCT-based algorithms suffer from visible blocking artifacts, which are easily noticeable as the discontinuities between neighboring blocks, particularly in flat and low activity homogeneous areas of the image because high frequency DCT coefficients are inclined to be quantized to zeros due to the high quantization parameters.

Many algorithms have been proposed to solve the problems. Most of them can be generally grouped into two categories: postprocessing and preprocessing. Most postprocessing algorithms can be divided into two steps: blocking artifact detection and smoothing

of blocks in the time domain (ITU-T, 1998; Webb, 1996; Yang *et al.*, 2005) or in the frequency domain (Chen *et al.*, 2001). At low bit rate, however, it is difficult to determine whether the discontinuities between neighboring blocks are exact edges or blocking artifacts, so postprocessing sometimes produces excessive smoothing of the image textures and edges.

In contrast, preprocessing algorithms improve coding efficiency and thus, reduce the likelihood of artifacts generation in the video coding process by removing some relatively insignificant high frequency components before encoding. Even if several preprocessing algorithms have been proposed, they are still far from perfection. The filters of most preprocessing algorithms are independent of bitrate and the characteristic of the pictures. They promote quality of reconstructed pictures at low bitrate while making reconstructed pictures lose high frequency details in high bitrate, so they are not fit for video

* Project (No. 2006CB303104) supported by the National Basic Research Program (973) of China

coding systems at a wide bitrate range. For example, Pham and van Vliet (2005) proposed a fixed parameters bilateral preprocessing filtering; Kimata *et al.* (2001) described a co-worked edge preserving pre- and post-filtering which are only adaptive to the edge detection; Huang (1997) proposed a four stage preprocessing system which is only simply adaptive to bitrate.

The best way to exploit preprocessing is to adjust the filtering intensity according to the complexity of the images and the bitrate of video encoding. When a significant number of bits are allocated to an image, little filtering is enough. When few bits are allocated, more filtering is necessary. So we developed a new bitrate adaptive preprocessing method, which employs an improved bilateral filter to provide adaptive edge preserving low-pass filtering. As shown in Fig.1, the filtering intensity is adjusted according to the quantization parameter. Whether at low or high bitrate, the preprocessing can provide proper filtering to make the video encoder more efficient and have higher visually image quality. Experimental results demonstrate that our proposed preprocessing method produces superior results at all tested bitrates in both subjectively perceived image quality and peak signal-to-noise ratio (PSNR).

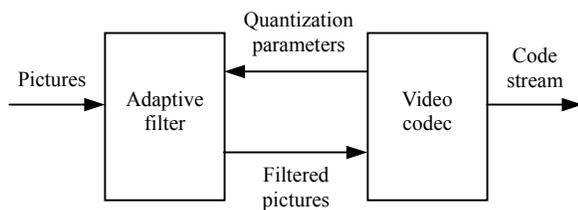


Fig.1 Flow chart of preprocessing

PRE-FILTERING

In an image, the most important information is in the low-frequency. Our response to an image is based upon a low-frequency ‘over-view’ of the image with important details added by higher frequencies in a few significant areas of the image (Richardson, 2002). In a broadcast environment the video source is not guaranteed to be noise free especially for low-cost cameras or under poor lighting condition. Noise in video signal does not only reduce video quality but also costs extra bits to code (Shen, 1999).

By attenuating the high-frequency components,

preprocessing algorithms reduce the complexity of a video sequence, less difference need to be transformed, so the quantization parameters are reduced. Thus, the quality of the reconstructed images is improved and they are less prone to errors such as blocking artifacts and so on. However, improvement in compression is at the expense of loss of details in the reconstructed images, such as edge features and sharp lines. At high bitrate, too much low-pass filtering is not acceptable due to the loss of details. So when a significant number of bits are allocated to an image, little filtering is needed. When few bits are available, more filtering is beneficial. A good preprocessing algorithm can exploit this improvement by adjusting the intensity of filtering. The selecting of filter and determination of the filtering intensity are therefore critical in designing preprocessing systems.

Bilateral filter is a kind of edge-preserving filter that employs both geometric closeness and photometric similarity of neighboring pixels to construct its filter kernel. When used as a low-pass filter, the bilateral filter produces excellent results because of its ability to keep edge features not to be attenuated as the high-frequency components.

BILATERAL FILTER

Bilateral filtering smoothes images while preserving edges, by means of a nonlinear combination of nearby image values. It does a weighted average of local samples, in which higher weights are given to samples that are closer in both space and intensity to the center sample (Tomasi and Manduchi, 1998; Pham and van Vliet, 2005). The weighted average is done over a neighborhood S around the center sample $s_0 = \{x_0, y_0\}$, whose intensity is $X(s_0)$:

$$Y(s_0) = \frac{\sum_{s \in S} X(s) S(s - s_0) T(X(s) - X(s_0))}{\sum_{s \in S} S(s - s_0) T(X(s) - X(s_0))}, \quad (1)$$

where $S(s - s_0)$ and $T(X(s) - X(s_0))$ are spatial and tonal weights of a pixel around s_0 . They are both Gaussian functions:

$$S(x) = \frac{1}{\sigma_s \sqrt{2\pi}} e^{-x^2 / 2\sigma_s^2}, \quad (2)$$

$$T(x) = \frac{1}{\sigma_T \sqrt{2\pi}} e^{-x^2/2\sigma_T^2}. \quad (3)$$

Fig.2 shows the weights of a bilateral filter centered at a pixel nearby an edge. The spatial weights $S(x)$ depicts the low-pass nature of the bilateral filter. The tonal weights $T(x)$ suppresses the contributions of pixels from the other side of the edge. It is responsible for the truncation of Gaussian bell weights. As can be seen from Figs.2c and 2e, after the filtering, the edge is not diffused and the high-frequency noise and insignificant components are suppressed.

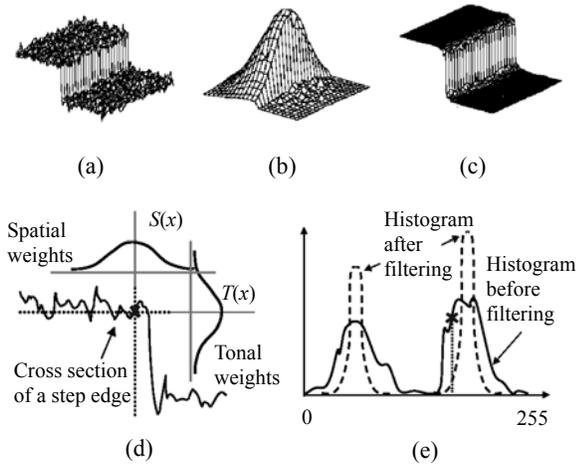


Fig.2 Bilateral filtering as a mode seeking process (Pham and van Vliet, 2005). (a) Noisy edge; (b) Bilateral filter; (c) Filtered output; (d) Spatial and tonal filters; (e) Thinning histogram

Due to the tonal weights, a 2D bilateral filter cannot be separated into two 1D bilateral filters. It means that the complexity of computation is $O(MND^2)$, which is much more than common low-pass filters (where M and N are width and height of the image, D is the size of the bilateral filter). Fortunately, Pham and van Vliet (2005) proved that if two 1D bilateral filters are used instead, they still have almost the same noise reduction and edge preservation performance as the 2D bilateral filter. So the complexity of computation is down to $O(MND)$ which is at the same level of common low-pass filters.

ADAPTIVE PREPROCESSING FILTER

In this paper, we proposed a two-mode pre-

processing filter. When a significant number of bits are allocated, it works in noise removal mode. When few bits are available, it works in low-pass filtering mode. In each mode, the filtering intensity is adaptive according to quantization parameters. The target is to keep the pictures to be coded in a proper quantization parameter range.

The first step of adaptation is determining the filtering level $L(i)$ which is from 0 to L_{\max} , which means from no filtering to maximum filtering intensity.

When the i th picture is coded, the filtering level $L(i)$ is computed as follows:

- (1) For the first picture, $L(0)$ is predefined to L_{init} .
- (2) For the following pictures,

$$L(i) = \begin{cases} \max(0, L(i-1) - L_{\text{dec}}), & QP(i-1) < Q_L; \\ \min(L_{\max}, L(i-1) + L_{\text{inc}}), & QP(i-1) > Q_U, \end{cases} \quad (4)$$

where L_{inc} and L_{dec} are increment and decrement units of $L(i)$, $QP(i-1)$ is the average quantization parameter of the $(i-1)$ th picture. Because we cannot get the quantization parameters of a picture before it is encoded, we employ the quantization parameters of the last picture instead according to the rule of similarity between neighbor pictures. Q_U and Q_L are upper and lower thresholds of proper QP range.

When $L(i)$ is low, it means a significant number of bits are allocated to a picture, little or no filtering is enough. If $L(i)$ is less than a threshold L_{mode} ($0 < L_{\text{mode}} < L_{\max}$), preprocessing will work in adaptive de-noise mode to remove high-frequency random noise. When $L(i)$ is high, few bits are allocated and more filtering is necessary. If $L(i)$ is more than threshold L_{mode} , the preprocessing will work in low-pass filtering mode to discard the insignificant high-frequency components so that the rest of the picture can be encoded in less quantization parameters and get superior quality.

The second step is determining $\sigma_S(i)$ and $\sigma_T(i)$ from filtering level $L(i)$:

$$\sigma_S(i) = \begin{cases} \sigma_{S,\text{BASE}} \times V_S^{L(i)-L_{\text{mode}}}, & L(i) \leq L_{\text{mode}}; \\ \sigma_{S,\text{BASE}}, & L(i) > L_{\text{mode}}; \end{cases} \quad (5)$$

$$\sigma_T(i) = \begin{cases} \sigma_{T,\text{BASE}}, & L(i) \leq L_{\text{mode}}; \\ \sigma_{T,\text{BASE}} + V_T(L_{\text{mode}} - L(i)), & L(i) > L_{\text{mode}}; \end{cases} \quad (6)$$

where $\sigma_{S,\text{BASE}}$ and $\sigma_{T,\text{BASE}}$ are the bases of variable σ_S

and σ_T , V_S and V_T are the variable steps of σ_S and σ_T . They are all constants, and should be determined according to the type of video codec. When $L(i)$ is no more than L_{mode} , $\sigma_T(i)$ is constant and keeps the maximum edge preservation performance and $\sigma_S(i)$ is varied according to $L(i)$. The less $L(i)$ the less de-noise performance. When $L(i)$ is more than L_{mode} , $\sigma_S(i)$ is constant and keeps the maximum low-pass filtering intensity and $\sigma_T(i)$ is varied according to $L(i)$. The more $L(i)$, the more pixels are low-pass filtered. Eqs.(5) and (6) are designed to provide evenly distributed edge preservation performance and low-pass filtering intensity when in de-noise mode and low-pass filtering mode.

EXPERIMENTS

We here take the H.263 video codec as an example. We define the 70% proportion center area of the pictures as the sensitivity area. For example, if picture sequences are CIF size, one can choose the 14 lines and 18 columns of macroblocks in the center of pictures as sensitivity area. After each picture is coded, the average of the top 10% quantization parameters of macroblocks in the sensitivity area is computed. The $L(i)$ is computed as follows: (1) For the first picture, $L(0)$ is set to 4; (2) For the following pictures,

$$L(i) = \begin{cases} \max(0, L(i-1) - 1), & QP(i-1) < 4; \\ \min(24, L(i-1) + (QP(i-1) - 6)/3), & QP(i-1) > 6. \end{cases} \quad (7)$$

Then get $\sigma_S(i)$ and $\sigma_T(i)$ as follows:

$$\sigma_S(i) = \begin{cases} 0.4 \times 1.1^{L(i)-4}, & L(i) \leq 9; \\ 0.4 \times 1.1^5, & L(i) > 9, \end{cases} \quad (8)$$

$$\sigma_T(i) = \begin{cases} 15, & L(i) \leq 9; \\ 15 + 6 \times (L(i) - 9), & L(i) > 9. \end{cases} \quad (9)$$

The kernel size of bilateral filter is 5 by 5.

This adaptive preprocessing algorithm has been tested on Foreman, Paris, Irene and Akiyo CIF size standard test sequences at bitrates of 128, 256, 512 and 768 kbps at 30 fps. Only the first frame is an I-frame.

In each case, the 128 pictures were encoded by a modified version of British Columbia's TMN 3.0 H.263 Reference Codec and TMN8 rate control is employed. QP of the I-frame is set to 13. The comparison of result data of the coding with and without adaptive preprocessing is listed in Table 1. The maximum increment of average luma PSNR is 4.33 dB (Irene sequence at 128 kbps). The minimum increment is 0.46 dB (Akiyo sequence at 768 kbps). The average increment of average luma PSNR is 1.88 dB.

Take the encoding parameters of the Paris sequence at 512 kbps as an example which is shown in Fig.3. Both the increment of PSNR and the decrement of QP are remarkable.

Fig.4 and Fig.5 compare a reconstructed picture and its segments of the 90th picture of the Paris sequence at 256 kbps. It can be seen the almost all blocking artifacts and ringing artifacts are removed.

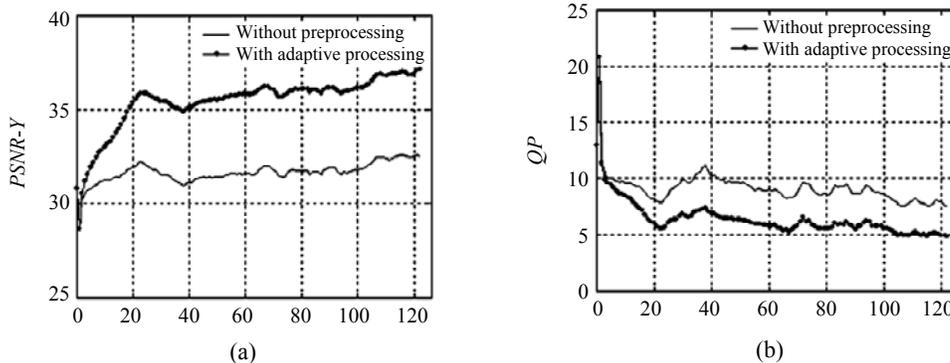


Fig.3 Comparing coding parameters of Paris sequence at 512 kbps between with and without preprocessing (a) PSNR-Y; (b) QP

Table 1 The statistics comparison between video encoding with and without adaptive preprocessing

Sequences	Bitrate (kbps)	Average PSNR-Y (dB)			Average QP			Average filtering level
		Without pre-processing	With pre-processing	Increment	Without pre-processing	With pre-processing	Decrement	
Foreman	128	29.54	31.56	2.02	28.63	27.33	1.03	23.47
	256	32.26	35.44	3.18	13.18	10.29	2.89	23.37
	512	34.89	35.70	0.81	7.24	6.82	0.42	6.27
	768	36.37	36.79	0.42	5.35	5.19	0.16	0.89
Paris	128	26.65	28.47	1.82	28.23	23.97	4.26	23.45
	256	28.84	31.79	2.95	15.89	11.57	4.32	23.48
	512	31.68	35.41	3.73	9.10	6.30	2.80	22.81
	768	33.87	35.99	2.12	6.42	5.28	1.14	13.51
Irene	128	32.26	36.59	4.33	20.52	18.02	2.50	23.52
	256	34.79	38.06	3.27	10.43	8.01	2.42	22.68
	512	37.62	38.02	0.40	5.81	5.63	0.18	0.95
	768	39.40	39.88	0.48	4.06	3.88	0.18	0.22
Akiyo	128	37.39	40.18	2.79	8.65	6.48	2.17	20.45
	256	40.48	40.95	0.47	4.63	4.46	0.17	2.97
	512	43.42	44.09	0.67	2.54	2.37	0.17	0.10
	768	44.73	45.19	0.46	1.81	1.70	0.11	0.11



Fig.4 90th reconstructed picture of Paris sequences at 256 kbps. (a) Without preprocessing; (b) With preprocessing



Fig.5 Segments of 90th reconstructed picture of Paris sequences at 256 kbps. (a) Without preprocessing; (b) With preprocessing

CONCLUSION

In this article, we propose a two-mode adaptive preprocessing filter. Its filtering intensity varies according to the complexity of the pictures and the available bits. When a significant number of bits are allocated to an image, it works in adaptive de-noise mode. When few bits are allocated, it works in adaptive low-pass filtering mode. By removing high-frequency noise and some insignificant high-frequency components, the quantization parameters of video coding are reduced, thus enhancing the coding efficiency. Experimental results demonstrate that our proposed preprocessing method produces superior results in all tested bitrates in both subjective image quality and PSNR.

This preprocessing filter only requires quantization parameters of the last picture from the video codec. As a result, it is suitable for all BDCT based video codecs, such as H.26x and MPEG family codecs and it can become a universal preprocessing algorithm.

References

- Chen, T., Wu, H.R., Qiu, B., 2001. Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts. *IEEE Trans. on Circuits and Systems for Video Technology*, **11**(5):594-602. [doi:10.1109/76.920189]
- Huang, S.J., 1997. Adaptive Noise Reduction and Image Sharpening for Digital Video Compression. Proc. IEEE Int. Conf. Computational Cybernetics Simulation, **4**:3142-3147. [doi:10.1109/ICSMC.1997.633077]
- ITU-T, 1998. Video Coding for Low Bitrate Communication. ITU-T Recommendation H.263, Version 2.
- Kimata, H., Yashima, Y., Kobayashi, N., 2001. Edge Preserving Pre-post Filtering for Low Bitrate Video Coding. Proc. of IEEE ICIP 2001, Thessaloniki, Greece, **3**:554-557.
- Pham, T.Q., van Vliet, L.J., 2005. Separable Bilateral Filtering for Fast Video Preprocessing. Proc. of IEEE ICME 2005, Amsterdam, the Netherlands, p.454-457.
- Richardson, I.E.G., 2002. Video Codec Design. John Wiley & Sons, Ltd., p.136-137.
- Shen, K., 1999. Video Preprocessing Techniques for Real-time Video Compression-noise Level Estimation Techniques. Proc. of IEEE Int. Symposium on Circuits and Systems (ISCAS), **2**:778-781.
- Tomasi, C., Manduchi, R., 1998. Bilateral Filtering for Gray and Color Images. Sixth International Conference on Computer Vision, p.839-846. [doi:10.1109/ICCV.1998.710815]
- Webb, J.L.H., 1996. Postprocessing to Reduce Blocking Artifacts for Low Bit-rate Video Coding Using Chrominance Information. Proc. of IEEE ICIP 1996, Lausanne, Switzerland, **2**:9-12.
- Yang, X.K., Yao, S.S., Lim, K.P., Lin, X., Rahardja, S., Pan, F., 2005. An Adaptive Edge-preserving Artifacts Removal Filter for Video Post-processing. Proc. of IEEE Int. Symposium on Circuits and Systems (ISCAS), **5**:4939-4942. [doi:10.1109/ISCAS.2005.1465741]