WPM P-1.12

Post-processing Using Computationally Efficient Adaptive Nonlinear Filters for HDTV

Yoon Kim*, Nam-Hyeong Kim*, Young Huh**,
Sung-Jea Ko*, *Senior Member, IEEE*, and Aldo W. Morales***

*Department of Electronics Engineering, Korea University, Seoul, Korea
**Applied Imaging Research Group, Korea Electrotechnology Research Institute, Changwon, Korea
***The Pennsylvania State University, DuBois, PA 15801, U.S.A.
Email: sjko@dali.korea.ac.kr

Abstract— **This paper presents a computationally efficient post-processing algorithm for HDTV. The proposed algorithm can reduce both blocking artifacts and mosquito noise while preserving the sharpness and naturalness of the reconstructed video signal. Performance improvements compared with other techniques are obtained according to simulation results.**

I. Introduction

Visual artifacts afflicting block DCT based image/video compression systems (e.g. JPEG, H.261/3, MPEG-1/2) are blocking effects (or blocking artifacts) and mosquito noise. Blocking effects are artificial structured artifacts due to discontinuities in the amplitude or slope of the reconstructed signal across the block boundaries. Mosquito noise is typically seen when a sharp edge separating two uniform regions occurs within a block. When only a small number of DCT coefficients are coded, the reconstructed block is littered with random noise or oscillatory distortion. Mosquito noise is accentuated because of its sharply defined square boundary. Various post-processing techniques for HDTV have been proposed to improve the quality of the decoded images by alleviation the blocking artifacts [1]-[4]. However, they are computationally and structurally complex.

In this paper, we present a computationally efficient post-processing algorithm for HDTV images which can reduce both blocking artifacts and mosquito noise while preserving the sharpness and naturalness of the reconstructed video signal. The proposed technique can reduce visual artifacts adaptively according to the local characteristics of images and the extent of blocking effects.

II. Proposed de-blocking method

For the measure of blocking effects we use the normalized summation of the absolute pixel value difference between the block and its four neighboring blocks. The measure of blocking effects, $B_d(i,j)$, is estimated along four boundaries of each block as follows:

$$B_l(i,j) = \frac{1}{N}\sum_{p=1}^{N}\left|\frac{X(Ni+1,Nj+p)-X(Ni,Nj+p)}{X(Ni+1,Nj+p)+X(Ni,Nj+p)}\right|,$$

$$B_r(i,j) = \frac{1}{N}\sum_{p=1}^{N}\left|\frac{X(N(i+1)+1,Nj+p)-X(N(i+1),Nj+p)}{X(N(i+1)+1,Nj+p)+X(N(i+1),Nj+p)}\right|,$$

$$B_t(i,j) = \frac{1}{N}\sum_{p=1}^{N}\left|\frac{X(Ni+p,Nj+1)-X(N(i+p),Nj)}{X(Ni+p,Nj+1)+X(N(i+p),Nj)}\right|,$$

$$B_b(i,j) = \frac{1}{N}\sum_{p=1}^{N}\left|\frac{X(Ni+p,N(j+1)+1)-X(Ni+p,N(j+1))}{X(Ni+p,N(j+1)+1)+X(Ni+p,N(j+1))}\right|, \quad (1)$$

where $B_l(i,j)$, $B_r(i,j)$, $B_t(i,j)$, and $B_b(i,j)$ denote the measures of blocking effects of the left, right, top, and bottom boundaries of the $(i,j)th$ $N \times N$ block, respectively, and $X(x,y)$ denotes the image pixel value at the (x,y) position (see Fig. 1).

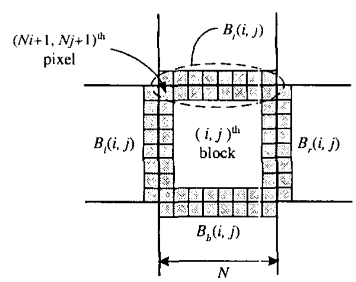

Fig. 1. Region of denoting the measure of blocking effects in the proposed method.

The output $Y(x,y)$ of the proposed filter is given by

$$Y(x,y) = \frac{\sum_{k=1}^{m}\sum_{l=1}^{m} w_{xy}(k,l)X(k,l)}{\sum_{k=1}^{m}\sum_{l=1}^{m} w_{xy}(k,l)}, \quad (2)$$

where $w_{xy}(k,l)$ is the adaptive weight. This adaptive weight is determined at each pixel using a trimming function $t_{xy}(k,l)$ and a distance weighting function $d_{xy}(k,l)$ as follows:

$$w_{xy}(k,l) = t_{xy}(k,l) \times d_{xy}(k,l),$$

$$t_{xy}(k,l) = \begin{cases} 1, & \text{if } |X(k,l) - X(x,y)| < T_1, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

$$d_{xy}(k,l) : Gaussian - like\ function.$$

Here, the $t_{xy}(k,l)$ is used to preserve image edges by excluding pixel values inside the window whose absolute difference from the central pixel value $X(x,y)$ is greater than threshold T_1. The distance weighting function $d_{xy}(k,l)$ is a Gaussian-like function which gives more weight to the central values inside the window.

To reduce the visual difference between adjacent blocks, we selectively use the symmetric $d_{xy}(k,l)$ and asymmetric $d_{xy}(k,l)$ along block boundaries. The asymmetric $d_{xy}(k,l)$ imposes more weight on pixels of the adjacent block if $B_d(i,j)$ is greater than threshold T_2. That is, if $B_d(i,j) < T_2$, the corresponding boundary pixels are processed using the symmetric $d_{xy}(k,l)$ and $t_{xy}(k,l)$, and if $B_d(i,j) \geq T_2$, the asymmetric $d_{xy}(k,l)$ and $t_{xy}(k,l)$ are used. And the pixels inside the blocks are processed using the symmetric $d_{xy}(k,l)$ and $t_{xy}(k,l)$.

III. Experimental Results and Conclusions

For simulations, we used the symmetric and asymmetric distance functions for a window 5×5 as shown in Fig. 2(a) and (b).

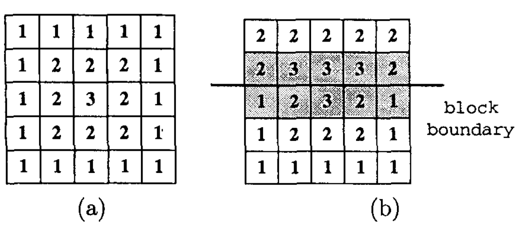

Fig. 2. Example of signal adaptive weighting function. (a) The symmetric $d_{xy}(k,l)$. (b) The asymmetric $d_{xy}(k,l)$.

For the subjective visual quality comparison, we applied the aforementioned post-processing techniques to real sequences. Fig. 3(a) shows the MPEG-2 decompressed "Lena" image at 0.2 bit/pixel (bps), and Fig. 3(b), (c), and (d), respectively, show results of the Ramamurthi's algorithm [3], the John's algorithm [4], and the proposed method. It is seen that the proposed method eliminates blocking effects as well as mosquito noise while preserving the original edge components of the image. The PSNR performance is summarized in Table I. It is shown that the PSNR of the proposed method is higher than that of the other algorithms. Experimental results indicate that the proposed method can be an useful alternative to existing post-processing methods.

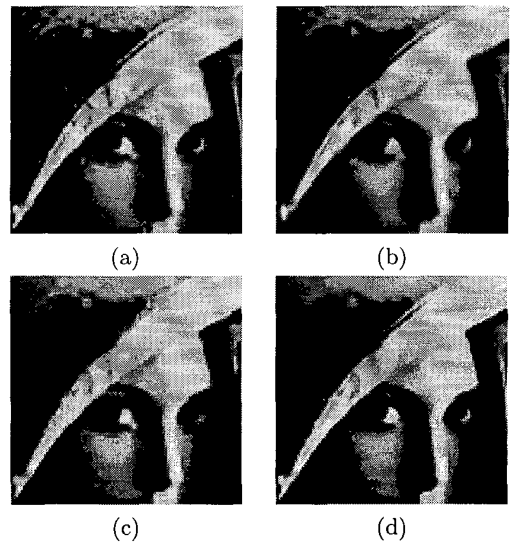

Fig. 3. Comparison of subjective quality on "Lena". (a) MPEG-2 decompressed image at 0.2 bpp. Post-processed images: (b)Ramamurthi's algorithm. (c) John's algorithm. (d) The proposed method.

TABLE I

PSNR FOR THE DIFFERENT DE-BLOCKING METHODS.

	Flower garden	Lena	Table tennis
MPEG	19.99	30.12	24.19
Ramamurthi's algorithm	18.67	28.85	23.93
John's algorithm	19.00	28.39	24.45
Proposed	20.00	30.98	25.35

REFERENCES

[1] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Projection-based spatially adaptive reconstrucion of block-transform compressed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 298-304, Aug. 1995.

[2] C. Avril and T. Nguyen-Trong, "Linear filtering for reducing blocking effects in orthogonal transform image coding," *J. Electron. Imaging*, vol. 1, pp. 183-191, Apr. 1992.

[3] B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 1258-1268, Oct. 1986.

[4] John G. Apostolopoulos and Nikil S. Jayant, "Postprocessing for very low bit-rate video compression," *IEEE Trans. on Image Processing.*, vol. 8, pp. 1125-1129, Aug. 1999.