IMPROVING DEBLOCKING ALGORITHMS BY USING STEP RESPONSE OF LPF IN THE LOW ACTIVITY REGION

Jung Won Kwak

School of Electrical Eng., Seoul National Univ.
Seoul 151-742, KOREA
곽정원
서울대학교 전기공학부

Nam Ik Cho

School of Electrical Eng., Seoul National Univ.
Seoul 151-742, KOREA
조남익
서울대학교 전기공학부

ABSTRACT

An algorithm for the reduction of blocking artifacts is proposed. Instead of filtering the pixels by a digital filter, the proposed algorithm considers the compressed image as a continuous-time signal and replaces it by a pre-designed lowpass filter outputs. The algorithm shows better performance than the conventional methods in low activity region, and comparable or a little worse objective performance in high activity region. Hence, by applying the proposed algorithm to the low activity region and the conventional ones to the high activity region, the objective and subjective performances are improved in most cases.

1 Introduction

Block discrete cosine transform (BDCT) is widely used in image and video compression standards. However, since the DCT is applied to 8×8 or 16×16 blocks, it introduces severe degradation around the block boundaries at low bit rates, which is known as "blocking artifacts" [1]. Hence, many postprocessing algorithms have been proposed for reducing the blocking artifacts at the decoder end .

In this paper, we propose a postprocessing algorithm based on the lowpass filtering. But, instead of handing the digital samples, the algorithm considers the scan lines of the image as continuous-time signals. The signal around the block boundary is modeled as a continuous-time step function, and it is just replaced by a step response of lowpass filter. More specifically, given the boundary difference and duration of step function, we can find the continuous-time step response without actual filtering. Then, the output pixels are obtained by sampling the step response.

Intensive simulation shows that the proposed algorithm provides better objective and subjective performance than conventional algorithms in low activity region, to which the human visual system is more sensitive. Since the block artifacts appear more like a step function in the low activity region, the algorithm performs better in this region than in the high activity area. In other words, the proposed model is better fitted in the low activity region. Hence, by incorporating the proposed and conventional algorithm, the deblocking performance is improved. More precisely, the blocks of the images are classified to be in the low or high activity region. Then, the proposed algorithm is applied to the low activity region, and the conventional ones to high activity region. By the incorporated algorithm, the objective performance of the conventional algorithm is improved more than 1.0 dB in some cases, and the subjective performance is also improved.

This paper is organized as follows. In section 2, we introduce an algorithm for the reduction of blocking artifact. Section 3 presents the incorporated algorithm and its performance. Finally, concluding remarks are given in section 4.

2 Proposed Algorithm for the Reduction of Blocking Artifacts

2.1. A deblocking algorithm

Fig. 1 shows typical shapes of scan lines at the block boundaries of compressed images. The circle represents the pixel, and the figure on the circle is the pixel value. The vertical broken lines represent the block boundary, and the lines represent the imaginary continuous-time signals, from which the pixels are sampled.

The conventional algorithms perform the filtering of the pixels with digital lowpass FIR filters. But, because the signal duration is very short and the shape of input varies from boundary to boundary, it is very difficult to determine the appropriate filter coefficients and tap lengths. The strong lowpass filter blurs the image too much, and the weak lowpass filter does not provide satisfying result in low activity area. Hence, instead of direct filtering, we consider the scan line as a continuous-time signal as shown in Fig. 1. Then the output of the filter is estimated with respect to the shape

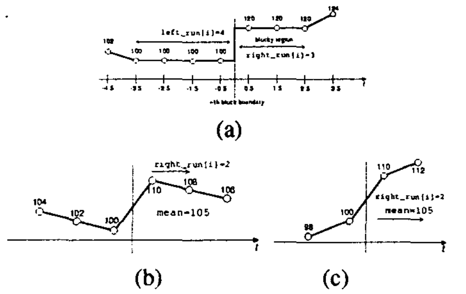


그림 1: Typical shapes of scan lines at the block boundary. (right_run[i] : length of the blocky region to the right. mean = (left[1]+right[1])/2 where left[j] is the j-th pixel value to the left of the boundary, and right[j] to the right.)

of the input without actual filtering.

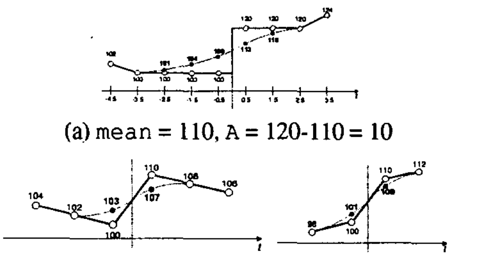
By using the quarter sine wave model which starts from boundary and ends at the end of blocky region, the strength of the filter is automatically adjusted according to the shape of the input. More precisely, the output changes slowly for wide blocky region, and vice versa.

Modeling the output by a sine wave means that

$$y(t) = \begin{cases} \text{mean} + \text{A}\sin\left(\frac{\pi t}{2(\text{right\_run}-0.5)}\right), & t \geq 0 \\ \text{mean} - \text{A}\sin\left(\frac{-\pi t}{2(\text{left\_run}-0.5)}\right), & t < 0 \end{cases}, \quad (1)$$

where mean is the mean of two pixels around the boundary, and A *is the magnitude of step input.*

For example, In Fig. 1(a), the blocky region is up to the 3rd pixel to the right, and up to the 4th pixel to the left. This parameter is denoted as right_run = 3, and left_run = 4 as shown in Fig. 2(a). By sampling the nearest interger of $y(t)$ at $t = -2.5, -1.5, \cdots, 0.5, 1, 5$, the ouput is obtained as shown in Fig. 2(a).



(a) mean = 110, A = 120-110 = 10

(b) mean = 105, A = 3 (c) mean = 105, A = 7

그림 2: Signal models for the filter outputs.

Let us consider the decision of the parameters right_run and left_run, which represent the length of blocky region. The block boundaries are classified into three classes : *no-operation*, *flat* and *nonflat* region. No postprocessing is required in *no-operation* region, where there is no pixel difference at the boundary. This region is indexed by setting the parameter to -1, which indicates that no filtering should be performed. The example of *flat* region is shown in Fig. 1(a), where there is no pixel differences up to several pixels other than the boundary. The rest are *nonflat* regions, *eg.* Fig. 1(b) and (c). Fig. 3 shows the flow chart for determining the length of the blocky region to the right (denoted as right_run in Fig. 1(a)) at the i-th boundary. The left_run is also similarly determined.

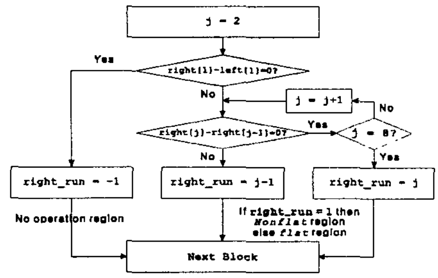


그림 3: Flow chart for the decision of right_run

As described in Fig. 3, if there is difference in the first and second pixel, right_run = 1 and the region is defined to be *nonflat*. However, if right_run = 1, no filtering is performed by the eq.(1) , and the blocking artifacts are not removed at all. Hence, for smoothing the *nonflat* region by the modeled output, if the difference of the first and second pixel is smaller than half the difference of the block boundary pixels, then the right_run[i] (or left_run[i]) is forced to become 2 as shown in the example of Fig. 1(b) and (c). * There is no more test for the pixels above the 3rd in the *nonflat* region, in order not to blur the high frequency region too much.

In some of the low frequency regions of highly compressed images, the sum of right_run[i] and left_run[i+1] may become larger than 8, as shown in Fig. 4(a). In this case, the result of filtering at the i-th boundary will be overridden by the filtering of $(i+1)$-th boundary as shown in Fig. 4(b). For preventing this undesirable result, we need to test this case and set right_run[i] = left_run[i+1] = 4, which will result in Fig. 4(c).

2.2. Performance of the proposed algorithm

The proposed algorithm is compared with the conventional filtering and POCS methods for still images. Table. 1 shows

*But, they are still classified as *nonflat* region.

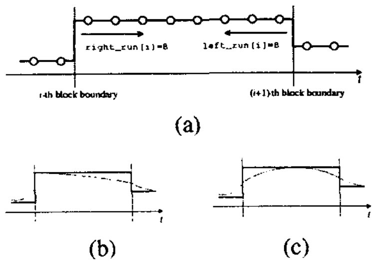


그림 4: (a) Shape of the blocky region where `right_run[i] + left_run[i+1]` > 8. (b) Result of filtering from left to right. (c) Result of filtering after setting `right_run[i] = left_run[i+1] = 4`.

the comparison of objective quality in PSNR when the algorithms are applied only to *flat* regions. The result shows that the proposed algorithm provides better performance in *flat* region. When the algorithms are applied to the *nonflat* region, the propose algorithm shows comparable or worse objective performance as shown in Table. 2. In summary, the proposed algorithm performs better in low activity region than in the high activity area. This means that the proposed model is better fitted in the low activity region. Improvement in the low activity region is more important than in the high activity area, because the human visual system is more sensitive to the artifacts in the low activity region.

표 1: Performance of the algorithms when applied to only *flat* regions (dB).

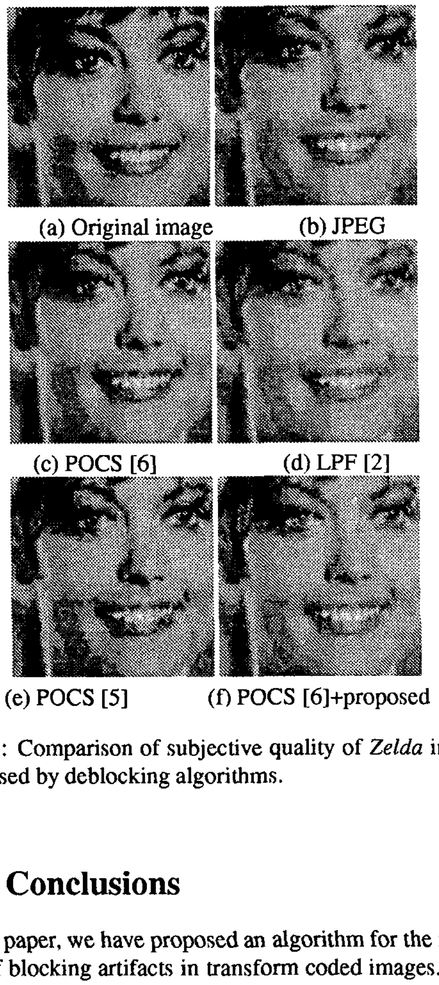
	bpp	JPEG	proposed	LPF [3]	POCS [6]	LPF [2]	POCS [5]
zelda	0.0552	25.33	27.16	26.69	26.69	26.15	26.23
	0.0736	27.18	28.55	28.23	28.24	27.98	27.85
peppers	0.0707	24.29	25.42	25.16	25.18	25.04	24.94
	0.0807	24.83	25.79	25.54	25.56	25.43	25.31

표 2: Performance of the algorithms when applied to only *nonflat* regions (dB).

	bpp	JPEG	proposed	LPF [3]	POCS [6]	LPF [2]	POCS [5]
barbara	0.1803	24.35	24.62	24.81	25.00	23.85	24.73
	0.2074	24.76	24.97	25.23	25.44	24.06	25.15
baboon	0.3120	22.48	22.70	22.71	22.76	22.26	22.83
	0.3519	23.16	23.27	23.36	23.41	22.55	23.47

3 Incorporating the proposed algorithm with conventional ones

As stated in the previous section, the proposed algorithm has advantages in the *flat* region, which usually corresponds to the low activity region. Hence, by applying the proposed algorithm to the *flat* region and the conventional algorithms to the *nonflat*, objective and subjective performance of conventional algorithms are improved in most cases. Table 3 shows the results of the conventional algorithms employing the proposed method in the *flat* region. The result shows that the performance is improved more than 1.0 dB in some cases. Fig. 5 shows the results for *Zelda* image for subjective comparisons. It can be observed that the artifacts in the *flat* region is less objectionable when the proposed algorithm is employed.



(a) Original image (b) JPEG
(c) POCS [6] (d) LPF [2]
(e) POCS [5] (f) POCS [6]+proposed

그림 5: Comparison of subjective quality of *Zelda* images processed by deblocking algorithms.

4 Conclusions

In this paper, we have proposed an algorithm for the reduction of blocking artifacts in transform coded images. Considering the scan line of a compressed image as a continuous time signal, it is modeled as a step function. The output of the lowpass filter for the step input is estimated with respect to its length and amplitude. Then the postprocessed pixels are obtained by sampling the continuous-time output signal. The proposed technique is adaptive in the sense that the strength of the filter and length of the signal to be filtered are automatically adjusted according to the shape of the input.

It has been shown that the algorithm performs better in the low activity region, because the proposed model is better

표 3: Comparison of objective quality in PSNR (dB). [·]+ means the incorporated algorithm, *i.e.*, the algorithm in [·] is applied to the *nonflat* region, and the proposed to the *flat*.

			POCS		LPF		LPF		LPF		POCS	
	bpp	JPEG	[6]	[6]+	[3]	[3]+	[7]	[7]+	[2]	[2]+	[5]	[5]+
zelda	0.0552	25.33	26.97	27.48	26.93	27.37	26.52	27.34	26.30	27.55	26.42	27.38
	0.0736	27.18	28.77	29.19	28.72	29.05	28.48	28.96	28.24	29.27	28.29	29.05
	0.1034	29.39	30.78	30.96	30.73	30.83	30.61	30.85	30.43	31.02	30.38	30.85
lena	0.0677	24.26	25.42	25.71	25.36	25.63	24.96	25.39	25.24	25.85	25.24	25.66
	0.0772	24.84	25.94	26.26	25.90	26.19	25.59	25.96	25.75	26.35	25.75	26.21
	0.0938	26.48	27.60	27.79	27.53	27.70	27.25	27.45	27.43	27.76	27.39	27.72
peppers	0.0707	24.29	25.65	25.93	25.53	25.82	25.07	25.64	25.26	25.97	25.35	25.86
	0.0807	24.82	26.07	26.35	25.97	26.25	25.68	26.10	25.72	26.38	25.78	26.29
	0.0980	26.03	27.16	27.39	27.05	27.25	26.90	27.17	26.88	27.33	26.88	27.29
barbara	0.0929	22.11	23.02	23.17	22.91	23.04	22.54	22.84	22.51	22.56	22.69	23.03
	0.1065	22.49	23.33	23.48	23.21	23.34	22.94	23.14	22.82	22.83	23.01	23.31
	0.1292	23.30	24.09	24.17	23.94	24.00	23.70	23.79	23.51	23.33	23.77	23.96

fitted in this region. Hence, by applying the proposed algorithm to the *flat* region, and the conventional ones to the *nonflat*, we have obtained improved results in most cases. Since the classification method and the proposed filtering algorithm are relatively simple, the incorporated algorithm requires less computational complexity than the original ones in many cases.

5 References

[1] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 3, no. 6, pp. 421 - 432, Dec. 1993.

[2] H. C. Reeves, and J. S. Lim, " Reduction of blocking effects in image coding," *Opt. Eng.*, vol. 23, pp. 34 -37, Jan./Feb., 1984.

[3] H. Paek, R.-C. Kim and S.-U. Lee, " A DCT-based spatially adaptive post-processing technique to reduce the blocking artifacts in transform coded images," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 1, pp. 36 - 41, Feb. 2000.

[4] R. Rosenholtz and A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 91-95, Mar.1992.

[5] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Projection-based spatially adaptive reconstruction of block-transform compressed images," *IEEE Trans. Image Processing*, vol.4, pp. 896-908, July 1995.

[6] H. Paek, R.-C. Kim and S.-U. Lee, "On the POCS-based postprocessing technique to reduce the blocking artifacts in transform coded images," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 3, pp. 358 - 367, Jun. 1998.

[7] ITU Telecom. Standardization Sector of ITU, "Video codec test model near-term, Version 8(TMN8), Release 0," H.268 Ad Hoc Group, Jun. 1997.