A PRACTICAL REAL-TIME POST-PROCESSING TECHNIQUE FOR BLOCK EFFECT ELIMINATION

Sung-Wai Hong
enhswai@hkpu01.polyu.edu.hk

Yuk-Hee Chan
enyhchan@hkpucc.polyu.edu.hk

Wan-Chi Siu
enwcsiu@hkpucc.polyu.edu.hk

Dept. of Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong
FAX : (852)2362-8439
TEL : (852)2766-6224

ABSTRACT

In this paper, a non-iterative post-processing method is proposed to restore the images encoded with block-based compression standards such as JPEG. This method classifies small local boundary regions according to their intensity distribution and then select appropriate linear predictors to estimate the corresponding boundary pixels in the regions. This approach is easy to be implemented and we found in our simulations that its restoration performance was very respectable compared with the reported post-processing methods.

1. INTRODUCTION

The Joint Photographic Experts Group (JPEG) image compression standard [1] has been widely used as an industrial standard for still image compression. In this standard, source images are first divided into non-overlapped blocks of size 8×8 each. Each block is then transformed with discrete cosine transform (DCT). Transform coefficients are quantized and encoded with arithmetic or Huffman coding technique. Since it is basically a block-based process, none of the inter-block correlation is exploited during the compression and therefore it causes visible discontinuities among adjacent blocks. This blocking effect is generally considered to be the most noticeable artifact in the reconstructed image.

Two common strategies have been used to reduce the blocking effect. One tries to retain some information of the inter-block correlation in the transmitted message. This can be done by embedding inter-block correlation in the transform coefficients [2-3] or by directly sending extra information to the receiver via additional channel bandwidth [4, 5]. Some of these methods can achieve a remarkable performance in eliminating the blocking effect. However, since these methods generally have to modify the basic transform coding structure of the existing JPEG standard, it is difficult or even impossible to integrate them with the current industrial standard.

The other strategy is to restore or enhance the image quality by using post-processing techniques. Typical examples include block-boundary filtering [5] and convex projection (POCS) [6-10]. This strategy has appeal since it only requires the decompressed image and is fully compatible with the existing JPEG standard.

However, though most of them were proved to be very effective to eliminate the blocking effect, there are still some limitations. For instance, projection methods are not suitable for real-time application since they are basically iterative algorithms which typically requires exhaustive computation and takes time for the estimate to converge to a desirable solution [6-10]. On the other hand, some methods aim at the enhancement of the image quality instead of the restoration of the image and therefore the distortion may even be enlarged after the process [5, 7].

In this paper, a non-iterative post-processing method is proposed to restore the JPEG-encoded images. This method classifies small local boundary regions according to their intensity distribution and then select appropriate linear predictors to estimate the corresponding boundary pixels in the regions. This approach is easy to be implemented and we found in our simulations that its restoration performance is very respectable compared with the reported post-processing methods [5-10].

2. ALGORITHM

To a certain extent, the elimination of the blocking effect can be viewed as a process of image restoration. This restoration process involves the modification of the JPEG-encoded image. When there is no extra information about the original image, any modification is solely based on assumptions. In other words, post-processing could cause degradation instead of improvement in the image quality. To minimize the possibility of the occurrence of degradation, it is reasonable to modify as few image pixels as possible during the post-

This work was supported by The Hong Kong Polytechnic University Grant A/C No. 350.255.A3.420

0-7803-3258-X/96/$5.00 © 1996 IEEE

processing process. Since the blocking effect appears as the discontinuity among adjacent blocks, it would be a good idea to smooth the discontinuity by modifying the intensity of the boundary pixels. However, as for the intra-block pixels, it is better to leave them without modification in order to minimize the disturbance to the image pixels.

It is well-known that the blocking effect is a consequence of ignoring the inter-block correlation during the compression. Therefore, one of the best ways to restore the original image is to make use of the residue inter-block correlation in the JPEG-encoded image to recover the block boundary pixels. In the proposed method, we investigate the connection between the residue inter-block correlation in the encoded image and the boundary pixels in the region concerned, and then make use of the statistics obtained to predict the original pixel intensity. Since there is typically a considerable amount of residue inter-block correlation remaining in the JPEG-encoded images, a good prediction result can be obtained based on the statistics even though only a simple linear prediction is performed.

Consider an 8-pixel line segment which is perpendicular to the block boundary as shown in figure 1. This segment is composed of two block boundary pixels and six neighboring pixels in line with them, which forms a pixel vector. Without losing the generality, we assume that the pixel vector obtained after JPEG-encoding is $\vec{v} = (v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7)$. Our objective is to predict the original boundary pixels p_3 and p_4 as accurately as possible. This can be done with linear prediction by making use of the intensity correlation among the original boundary pixels and their neighboring pixels in the JPEG-encoded image. Assume that the predicted value $\hat{p_3}$ is given by

$$\hat{p_3} = \sum_{j=0}^{7} a_j v'_j + m \tag{1}$$

where a_j's are scalar prediction filter coefficients, $v'_j = v_j - m$ and $m = \frac{1}{8}\sum_{j=0}^{7} v_j$. The selection of the filter coefficients should be based on the least-squares minimization criteria. In other words, one has to minimize the prediction error $J = \mathbf{E}[e_3^2] = \mathbf{E}[(p_3 - \hat{p_3})^2]$ in an average least-squares sense, where $\mathbf{E}[\cdot]$ is the expectation operator. The minimization of J can be achieved by satisfying the criteria $\frac{\partial J}{\partial a_j} = 0$ for all a_j, which results in the following formulation.

$$\begin{bmatrix} \mathbf{E}[v'_0 v'_0] & \mathbf{E}[v'_1 v'_0] & \cdots & \mathbf{E}[v'_7 v'_0] \\ \mathbf{E}[v'_0 v'_1] & \mathbf{E}[v'_1 v'_1] & \cdots & \mathbf{E}[v'_7 v'_1] \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{E}[v'_0 v'_7] & \mathbf{E}[v'_1 v'_7] & \cdots & \mathbf{E}[v'_7 v'_7] \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_7 \end{bmatrix}$$

$$= \mathbf{R}_{v'v'} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_7 \end{bmatrix} = \begin{bmatrix} \mathbf{E}[(p_3 - m)v'_0] \\ \mathbf{E}[(p_3 - m)v'_1] \\ \vdots \\ \mathbf{E}[(p_3 - m)v'_7] \end{bmatrix} \tag{2}$$

Prediction coefficients a_j's can then be determined provided that $\mathbf{R}_{v'v'}^{-1}$ exists.

The prediction filter for p_4 can also be developed in a similar way. However, to avoid doubling the number of filters required, pixel p_4 can be predicted by processing the reversed vector $\vec{v_r} = (v_7, v_6, v_5, v_4, v_3, v_2, v_1, v_0)$ instead of $\vec{v}$ with the same prediction filter.

To assure the robustness of the prediction, pixel vectors are classified into different classes such that dedicated prediction filter can be designed based on the statistics of particular class of pixel vectors by making use of the above technique.

As most of the inter-block correlation can be reflected by the smoothness of the intensity distribution across the block boundary, pixel vectors should be classified according to their intensity variation. One of the simplest but most effective way to achieve this objective is to make use of the concept of mean removal vector quantization.

Consider the case that there is a codebook containing N codewords $C = \{\vec{c_i} : i = 1, 2, \cdots, N\}$, where N is the number of classes we are going to classify the pixel vectors. For any given mean-removed input pixel vector, classification is carried out by performing a VQ encoding procedure. That means one has to select a codeword which is closest to the given mean-removed pixel vector. The selection is based on the minimum Euclidean distance criterion, where the Euclidean distance measure is defined as $d = \sum_{j=0}^{7}(c_{i,j} - v'_j)^2$ and $c_{i,j}$ is the jth element of codeword $\vec{c_i}$. The index $i \in$

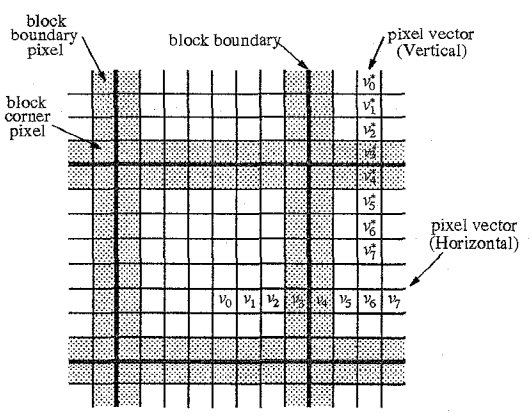

Figure 1: Pixel vectors generated from images under JPEG coding.

$\{1, 2, \cdots, N\}$ of the chosen codeword is then equivalent to the class number with which the pixel vector associates.

The codewords have to be carefully chosen such that they are representatives of their corresponding classes of pixel vectors. As usual, the codebook has to be designed with a set of training vectors prior to its application. Typical JPEG-encoded images can be used to generate training vectors. In particular, all vertical and horizontal mean-removed pixel vectors of the images, except those containing block corner pixels, are used as training vectors. The reversed vector of each training vector is also used as a training vector to double the size of the training set. Then, the LBG algorithm [11] is used to generate the optimal codebook with the set of training vectors. To assure the effectiveness of the post-processing method, the training set should be large enough and able to represent the statistics of the images to be coded.

The design of the codebook and the prediction filters could be computation-intensive, but their realization can be off-line and no second realization is required once it is done unless the nature of input images is greatly changed. In real-time application, the proposed method consist of two stages. Figure 2 shows the block diagram of the proposed post-processing method. After a JPEG-encoded image is decoded, pixel vectors of the decoded image are mean-removed and then classified with a VQ encoder. Based on the output of the VQ encoder, an appropriate prediction filter is selected for each mean-removed pixel vector. This vector is then fed into the filter to get a prediction value. The boundary pixel predicted will then be assigned the rounded result of the prediction value if it is not a block corner pixel. Note the reversed vector of the processing pixel vector is also a pixel vector and it gives another boundary pixel when it goes through the same procedures mentioned above. As for those block corner pixels, they are restored at the end of the process. Since there will be two prediction values (One from a horizontal pixel vector and one from a vertical pixel vector), the reconstructed pixel value is assigned to be the rounded result of the average value. Note that the pixel vectors used for restoring block corner pixels are composed of the restored boundary pixels.

3. SIMULATIONS

Computer simulations have been carried out to evaluate the restoration performance of the proposed method. The simulations are based on a set of 256 level grayscale digital images of size 256 × 256 pixels. The blocky versions of five standard images "Lenna", "House", "Peppers", "Airplane" and "Sailboat" were used to generate a training set to obtain codebooks of size N, where N = 8, 16 and 32. These blocky versions were generated by coding the original images with the JPEG scheme at a compression rate around 0.27 bit per pixel. Training pixel vectors were then classified into N classes with the corresponding codebook. Prediction filters of various classes were then designed with training vectors of their corresponding classes by using the approach mentioned in the previous section.

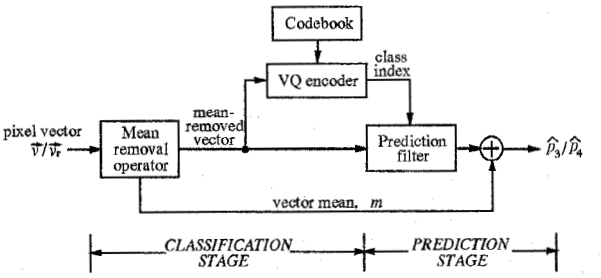

Figure 2: Block diagram of the post-processing method.

The performance is evaluated in terms of the improvement in signal-to-noise ratio which is defined as

$$\Delta\text{SNR} = 10 \log_{10} \frac{\|F_B - F_I\|^2}{\|F_R - F_I\|^2} \text{ dB} \qquad (3)$$

where F_R, F_B and F_I are the restored , the JPEG-encoded and the original images respectively. Some other post-processing methods [5-7] for blocking effect elimination are also evaluated for comparison. The termination criterion for iterative methods is chosen to be $\frac{\|F_{R,k} - F_{R,k-1}\|^2}{\|F_{R,k}\|^2} \leq 10^{-8}$, where $F_{R,k}$ is the restored image obtained at the kth iteration.

Table 1 shows the restoration performance of various methods. One can see the superior restoration performance of the proposed method compared with other post-processing methods [5-7]. In general, methods using low-pass filtering [5, 7] may reduce the SNR since what they concern is to enhance the subjective quality by smoothing the block boundary. The modified JPEG decoder proposed in ref. [12] is efficient and practical, but its performance was much more sensitive than that of the proposed method to the choice of the training set. This can be seen from the fact that, with the same small number of training images, its restoration performance fluctuated from training images to non-training images while the proposed method could provide a much more stable performance in our simulation.

Table 1 also shows the performance of the proposed method with various number of classes. More classes

Table 1: Restoration performance of various JPEG post-processing methods in terms of SNR improvement.

JPEG encoded image	bpp	Iterative method		Non-iterative method				
		POCS [6]	CLS1 [6]	* Filtering [5]	Modified decoder [12]	Proposed approach		
						8 classes	16 classes	32 classes
† Lenna	0.26	0.356dB	0.136dB	-0.039dB	0.940dB	0.503dB	0.534dB	0.551dB
† House	0.28	0.008dB	0.149dB	-0.138dB	0.632dB	0.477dB	0.506dB	0.539dB
† Peppers	0.27	0.063dB	0.121dB	-0.022dB	0.917dB	0.465dB	0.496dB	0.520dB
† Airplane	0.29	0.161dB	0.122dB	-0.361dB	1.758dB	0.337dB	0.362dB	0.397dB
† Sailboat	0.29	-0.023dB	0.101dB	-0.219dB	0.940dB	0.297dB	0.317dB	0.332dB
Couple	0.20	0.189dB	0.122dB	0.021dB	-0.777dB	0.497dB	0.521dB	0.532dB
Girl	0.20	0.526dB	0.149dB	0.300dB	-0.593dB	0.574dB	0.595dB	0.609dB
Hat	0.28	0.713dB	0.210dB	0.586dB	-0.756dB	0.828dB	0.854dB	0.880dB
Germany	0.20	0.297dB	0.112dB	0.119dB	-0.461dB	0.394dB	0.416dB	0.422dB
Tiffany	0.19	0.233dB	0.098dB	0.180dB	-0.639dB	0.335dB	0.338dB	0.346dB

† Images that were used for training in the simulation of method [12] and the proposed method.
* It is modified such that the filtering output is bounded. This is also the first iteration result of [7].

resulted in better performance. However, 8 to 16 classes would be appropriate since it made the design simpler while maintaining a reasonable restoration performance compared with other reported methods [5-10].

4. CONCLUSION

Based on linear prediction and vector quantization concepts, an effective post-processing technique for transform coding is proposed in this paper. This technique is fully compatible with the existing JPEG standard and enables real-time application. Computer simulations shown that the proposed method can achieve a better restoration performance in terms of SNR improvement compared with some other existing post-processing methods [5-8]. Moreover, the image quality can also be improved subjectively by eliminating most of the blocking effect.

5. REFERENCES

[1] W. B. Pennebaker and J. L. Mitchell, *JPEG still image data compression standard,* Van Nostrand Reinhold, New York, 1993.

[2] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects," IEEE Trans. Acoust., Speech, Signal Processing, 37(4), pp. 553-559, Apr. 1989.

[3] Y. H. Chan and W. C. Siu, "An approach to subband DCT image coding," J. Visual Comm. and Image Rep., 5(1), pp. 95-106, Mar. 1994.

[4] Y. Q. Zhang, R. L. Pickholtz and M. H. Loew, "A new approach to reduce the blocking effect of transform coding," IEEE Trans. Comm., 41(2), pp. 299-302, Feb. 1993.

[5] H. C. Reeve and J. S. Lim, "Reduction of blocking effect in image coding," Opt. Eng., 23(1), pp. 34-37, Jan./Feb. 1984.

[6] Y. Yang, N. P. Galatsanos and A. K. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," IEEE Trans. Cir. Sys. for Video Tech., 3(6), pp. 421-432, Dec. 1993.

[7] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," IEEE Trans. Cir. Sys. for Video Tech., 2(1), pp. 91-95, Mar. 1992.

[8] T. P. O'Rourke and R. L. Stevenson, "Improved image decompression for reduced transform coding artifacts," SPIE, 2182, Image & Video Processing II, pp. 90-101. 1994.

[9] S. Minami and A. Zakhor, "An optimization approach for removing blocking effects in transform coding," IEEE Trans. Cir. Sys. for Video Tech., 5(2), pp. 74-81, Apr. 1995.

[10] S. W. Hong, Y. H. Chan and W. C. Siu, "An adaptive constrained least square approach for removing blocking effect," Proc. IEEE ISCAS'95, 2, pp. 773-776, Apr./May 1995.

[11] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Comm., 28, pp. 84-95, 1980.

[12] S. W. Wu and A. Gersho, "Improved decoder for transform coding with application to the JPEG baseline system," IEEE Trans. Comm., 40(2), pp. 251-254, Feb. 1992.