

# Transactions Letters

## A Practical Postprocessing Technique for Real-Time Block-Based Coding System

Yuk-Hee Chan, Sung-Wai Hong, and Wan-Chi Siu

**Abstract**—In this paper, a noniterative postprocessing method is proposed to restore the images encoded with block-based compression standards such as Joint Photographic Experts Group (JPEG). This method classifies small local boundary regions according to their intensity distribution and then selects appropriate linear predictors to estimate the corresponding boundary pixels in the regions. This approach is easy to implement and we found in our simulations that its restoration performance was very respectable compared with the reported postprocessing methods.

**Index Terms**—Block-based coding, codec, deblocking, image restoration, JPEG, postprocessing technique, vector quantization.

### I. 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 nonoverlapped blocks of size  $8 \times 8$  each. Each block is then transformed with discrete cosine transform (DCT). Transform coefficients are quantized and encoded with an arithmetic or Huffman coding technique. Since it is basically a block-based process, little of the interblock 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 interblock correlation in the transmitted message. This can be done by embedding interblock correlation in the transform coefficients [2]–[4] or by directly sending extra information to the receiver via additional channel bandwidth [5], [6]. 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 postprocessing techniques. Typical examples include block-boundary filtering [6] and convex projection (POCS) [7]–[11]. 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 proven to be very effective in eliminating 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 require exhaustive computation and take time for the estimate to converge to a desirable solution [7]–[11]. 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 [6], [8].

In this paper, a noniterative postprocessing method is proposed to restore the JPEG-encoded images. This method classifies small local boundary regions according to their intensity distribution and then selects appropriate linear predictors to estimate the corresponding boundary pixels in the regions. This approach is easy to implement and we found in our simulations that its restoration performance is very respectable compared with the reported postprocessing methods [6]–[11].

### II. 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, postprocessing 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 postprocessing 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 intrablock pixels, it is better to leave them without modification in order to minimize the disturbance to the image pixels, especially when the compression rate is not too high.

It is well known that the blocking effect is a consequence of ignoring the interblock correlation during the compression. Therefore, one of the best ways to restore the original image is to make use of the residue interblock correlation in the JPEG-encoded image to recover the block boundary pixels. In the proposed method, we investigate the connection between the residue interblock correlation in the encoded image and

Manuscript received November 3, 1995; revised May 10, 1997. This paper was recommended by Associate Editor K.-H. Tzou. This work was supported by The Hong Kong Polytechnic University Grant A/C 350.255.A3.420.

The authors are with the Department of Electronic Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

Publisher Item Identifier S 1051-8215(98)01517-1.

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 interblock 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 eight-pixel line segment which is perpendicular to the block boundary as shown in Fig. 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

$$\hat{p}_3 = \sum_{j=0}^7 a_j v'_j + m \quad (1)$$

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 criterion [12]. 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  $\partial J / \partial a_j = 0$  for all  $a_j$ , which results in the following formulation:

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

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 a dedicated prediction filter can be designed based on the statistics of a particular class of pixel vectors by making use of the above technique.

As most of the interblock 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 ways to achieve this objective is to make use of the concept of mean removal vector quantization.

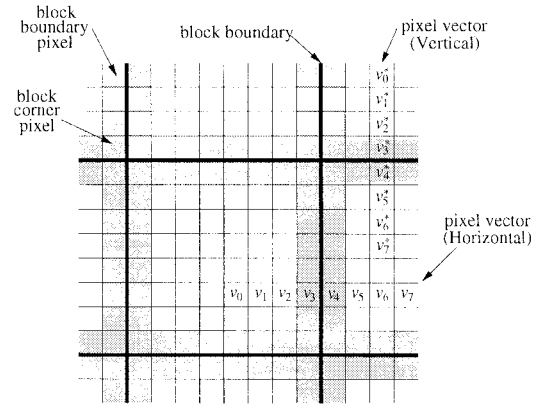


Fig. 1. Pixel vectors generated from images under JPEG coding.

Consider the case that there is a codebook containing  $N$  codewords  $C = \{\vec{c}_i : i = 1, 2, \dots, N\}$ , where  $N$  is the number of classes into which 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  $j$ th element of codeword  $\vec{c}_i$ . The index  $i \in \{1, 2, \dots, N\}$  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 [13] is used to generate the optimal codebook with the set of training vectors. To assure the effectiveness of the postprocessing 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 consists of two stages. Fig. 2 shows the block diagram of the proposed postprocessing 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

TABLE I  
RESTORATION PERFORMANCE OF VARIOUS JPEG POSTPROCESSING METHODS IN TERMS OF SNR IMPROVEMENTS (UNIT: dB)

JPEG encoded image	bpp	Iterative method		Non-iterative method							
		POCS [7]	CLS [7]	Filtering [6]	Modified decoder [15]	NID <sub>o</sub> decoder [16]	Proposed approach with various numbers of classes				
							1 class	8 classes	16 classes	32 classes	64 classes
† Lenna	0.26	0.356	0.136	-0.039	0.940	1.242	0.353	0.503	0.534	0.551	0.586
† House	0.28	0.008	0.149	-0.138	0.632	1.117	0.327	0.477	0.506	0.539	0.566
† Peppers	0.27	0.063	0.121	-0.022	0.917	1.277	0.314	0.465	0.496	0.520	0.544
† Airplane	0.29	0.161	0.122	-0.361	1.758	1.801	0.245	0.337	0.362	0.397	0.430
† Sailboat	0.29	-0.023	0.101	-0.219	0.940	1.094	0.213	0.297	0.317	0.332	0.343
Couple	0.20	0.189	0.122	0.021	-0.777	0.209	0.336	0.497	0.521	0.532	0.549
Girl	0.20	0.526	0.149	0.300	-0.593	0.261	0.413	0.574	0.595	0.609	0.614
Hat	0.28	0.713	0.210	0.586	-0.756	-0.001	0.664	0.828	0.854	0.880	0.896
Germany	0.20	0.297	0.112	0.119	-0.461	0.203	0.276	0.394	0.416	0.422	0.423
Tiffany	0.19	0.233	0.098	0.180	-0.639	-0.142	0.251	0.335	0.338	0.346	0.359

† Images that were used for training in the simulation of methods [15], [16] and the proposed method.

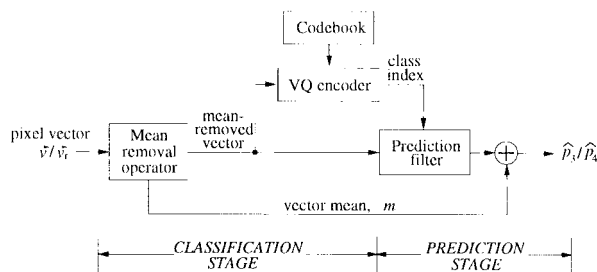


Fig. 2. Block diagram of the postprocessing method.

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.

The robustness of the prediction filter is guaranteed by the classification of pixel vectors and the following corresponding action. In fact, the classification process introduces a nonlinear nature into the system. Theoretically, the greater the number of classes, the better performance it can achieve. Naturally, the complexity gets higher too, but the burden can be significantly reduced by using the technique proposed in [14].

### III. 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  $256 \times 256$  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  $N = 8, 16, 32$ , and  $64$ . These blocky versions were generated by coding the original images with the JPEG scheme at a compression rate around 0.27 b/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.

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

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

where  $F_R$ ,  $F_B$ , and  $F_I$  are the restored, the JPEG-encoded, and the original images, respectively. Some other postprocessing methods [6]–[8] for blocking effect elimination are also evaluated for comparison. The termination criterion for iterative methods is chosen to be  $\|F_{R,k} - F_{R,k-1}\|^2 / \|F_{R,k}\|^2 \leq 10^{-8}$ , where  $F_{R,k}$  is the restored image obtained at the  $k$ th iteration.

Table I and Fig. 3 show the restoration performance of various methods. One can see the superior restoration performance of the proposed method compared with other postprocessing methods [6]–[8]. In general, methods using low-pass filtering [6], [8] may reduce the SNR since they are concerned with the enhancement of the subjective quality by smoothing the block boundary. The modified JPEG decoders proposed in [15] and [16] are efficient and practical, but their 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 nontraining images, while the proposed method could provide much more stable performance in our simulation. As a final remark, we note that a few iterative methods such as [11] can occasionally achieve a slightly higher SNR improvement than the proposed method can. However, due to their iterative nature, they cannot be used for real-time applications.

Table I also shows the performance of the proposed method with various numbers of classes. It was found that, though more classes resulted in better performance, the marginal improvement decreased as the number of classes increased. From our simulation result, 8 to 16 classes would be appropriate since it made the design simpler while maintaining a reasonable restoration performance compared with other reported methods [6]–[11].

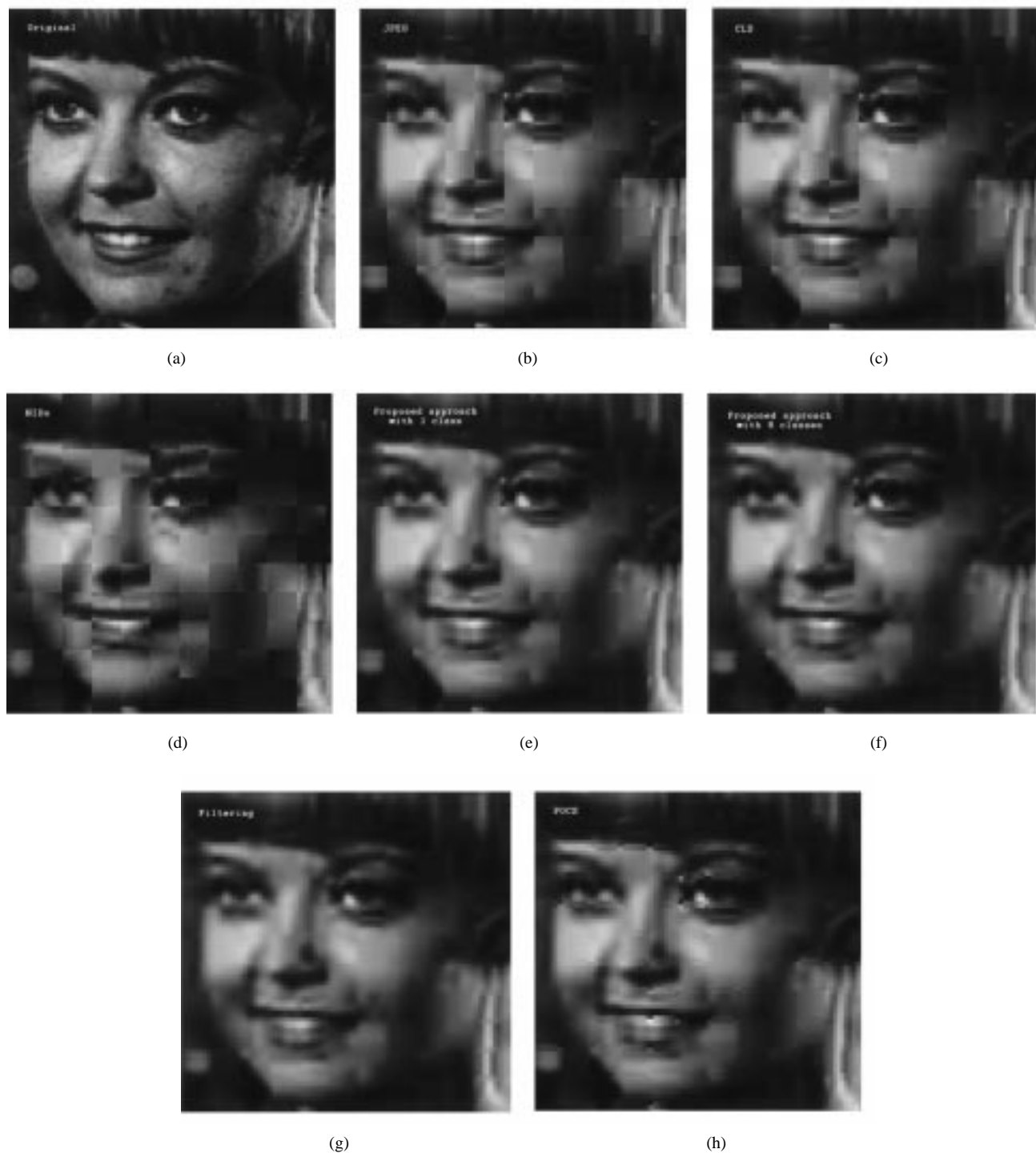


Fig. 3. Reconstructed images with various JPEG-compatible deblocking decoding methods: (a) original (b) JPEG (0.2 bpp), (c) CLS [7], (d)  $NID_0$  [16], (e) proposed approach with no classification, (f) proposed approach with eight classes, (g) filtering [6], and (h) POCS [7] (testing image: girl).

#### IV. CONCLUSION

Based on linear prediction and vector quantization concepts, an effective postprocessing 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 show that the proposed method could achieve a better restoration performance in terms of SNR improvement compared with some other existing postprocessing methods [6]–[9]. Moreover, the image quality could also be

improved subjectively by eliminating most of the blocking effect.

In this paper, we have considered the application of the proposed method in JPEG coding. However, the same technique can be applied to block-based transform video coding such as MPEG standard. Since video sequence typically contains a series of images with similar statistics, even few classes can achieve a respectable and consistent improvement in image quality when the proposed method is exploited. This is especially true in videophone application.

## REFERENCES

- [1] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.
- [2] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 553–559, Apr. 1989.
- [3] G. Aharoni, A. Averbuch, R. Coifman, and M. Israeli, "Local cosine transform—A method for the reduction of the blocking effect in JPEG," *J. Mathematical Imaging and Vision*, vol. 3, pp. 7–38, 1993.
- [4] Y. H. Chan and W. C. Siu, "An approach to subband DCT image coding," *J. Visual Commun., Image Represent.*, vol. 5, no. 1, pp. 95–106, Mar. 1994.
- [5] Y. Q. Zhang, R. L. Pickholtz, and M. H. Loew, "A new approach to reduce the blocking effect of transform coding," *IEEE Trans. Commun.*, vol. 41, pp. 299–302, Feb. 1993.
- [6] H. C. Reeve and J. S. Lim, "Reduction of blocking effect in image coding," *Opt. Eng.*, vol. 23, no. 1, pp. 34–37, Jan./Feb. 1984.
- [7] 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 Syst. Video Technol.*, vol. 3, pp. 421–432, Dec. 1993.
- [8] 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.
- [9] 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.
- [10] S. Minami and A. Zakhor, "An optimization approach for removing blocking effects in transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 74–81, Apr. 1995.
- [11] S. W. Hong, Y. H. Chan, and W. C. Siu, "An adaptive constrained least square approach for removing blocking effect," in *Proc. IEEE ISCAS'95*, Apr./May 1995, vol. 2, pp. 773–776.
- [12] S. J. Orfanidis, *Optimum Signal Processing*, 2nd ed. New York: McGraw-Hill, 1988.
- [13] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95, 1980.
- [14] Y. H. Chan and W. C. Siu, "In search of the optimal searching sequence for VQ encoding," *IEEE Trans. Commun.*, vol. 43, pp. 2891–2893, Dec. 1995.
- [15] S. W. Wu and A. Gersho, "Improved decoder for transform coding with application to the JPEG baseline system," *IEEE Trans. Commun.*, vol. 40, pp. 251–254, Feb. 1992.
- [16] ———, "Nonlinear interpolative decoding of standard transform coded images and video," in *SPIE Conf. Image Processing Algorithms and Techniques III*, San Jose, CA, Feb. 1992, pp. 88–99.