BLOCK LOSS RECOVERY IN DCT IMAGE ENCODING USING POCS

Jiho Park†, Dong-Chul Park‡, R.J. Marks II†, M.A. El-Sharkawi†

†Dept. of Electrical Engineering, University of Washington, Seattle, WA
‡Dept. of Information Engineering, MyongJi University, Kyunggido, Korea

ABSTRACT

An image recovery algorithm based on alternating projections onto convex sets (POCS) is proposed for block-based coding. In the proposed algorithm, vectors are extracted from the surrounding neighborhood of a missing block to make a convex hull for the reference of recovery. The recovery vectors including both known and lost pixels are used for the projections to find optimal spectrum coefficients for the missing pixels. The convex hull and two other convex sets are formulated for application of POCS. Simulation results show that the proposed algorithm recovers spatial structure both at low and high frequencies.

Index Terms–image recovery, convex hull, POCS, JPEG, MPEG

1. INTRODUCTION

Many transform coding schemes such as JPEG and MPEG standards are based on block coding technique [1][2]. We consider the interpolation restoration of a plurality of blocks of pixels lost due to transmission error or other reason.

Many spatial interpolation techniques for restoring missing blocks of images have been proposed. Wang *et al.* [3] propose an optimization technique using the smoothness property of images, which recovers damaged blocks by minimizing the differences of the blocks and boundaries, and subsequently generates smooth images. Hemami and Meng [4] propose an image reconstruction algorithm exploiting interblock correlation. Sun and Kwok [5] suggest use of a spatial interpolation algorithm using *projections onto convex sets* (POCS) [6]. The performance of Sun and Kwok's algorithm depends significantly on the result of an edge orientation detector. It also limits recovery of certain spectral bands due to the use of a band pass filter. A fast DCT-based spatial domain interpolation technique is reported by Alkachouh and Bellanger [7] wherein a missing block and border pixels are transformed by DCT, high frequency coefficients are set to zero, and a missing block is subsequently restored.

Building on this previous work, we propose a spectrally robust interpolative image restoration method based on POCS. A convex hull is formed from surrounding pixel vectors. Two additional convex sets are defined for the constraints of dynamic range and gradient. Vectors which include both known and missing pixels are projected onto the vertices of the convex hull in the DCT domain and onto convex sets in the spatial domain. Missing pixels are restored iteratively using POCS. The algorithm enables restored pixels to sustain spectrum and edge structures of surrounding blocks, and consequently to have continuity, similarity, and smoothness with neighborhood pixels.

2. LINE DETECTION AND VECTOR FORMING

In this section, image conditioning for missing pixel interpolation is discussed. In Figure 1 (a), a missing block, M, with surrounding neighborhood, S, is shown. The orientation of edges in the surrounding neighborhood, S, is assumed to extend its spatial structure to the missing block, M. The structure in the missing block is dictated by the orientation of lines and edges in the surrounding pixels. Vectors are extracted from the surrounding blocks, and two recovery vectors are calculated in the surrounding and missing blocks.

2.1. Line Detection

Line detection in the spatial domain is applied to determine edge orientation. Line masks are applied to the surrounding blocks, S_E, S_W, S_N, and S_S, shown in Figure 1 (a). The responses R_h and R_v are computed at all i, j coordinates in the blocks. Line masks [8] are defined as

$$L_h = \begin{vmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{vmatrix}, \quad L_v = \begin{vmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{vmatrix}. \quad (1)$$

The magnitudes of responses R_h and R_v at all i, j coordinates in the blocks are computed as

$$T_h = \sum_{S_E, S_W, S_N, S_S} |R_h|, \quad T_v = \sum_{S_E, S_W, S_N, S_S} |R_v|. \quad (2)$$

If T_h is larger, the missing block is considered as a horizontal line dominating block. Otherwise, it is considered as a vertical line dominating block. In the proposed algorithm, line detection is related to setting the initial point of projections.

2.2. Surrounding Vector

Since the surrounding neighborhood of a missing block is assumed to have spatial and spectral similarity with the missing block, the neighborhood area can be segmented into several blocks each of which has its own spatial and spectral characteristics. The segmentation of the neighborhood and the extraction of vectors are made by shifting an $N \times N$ window on every grid of pixels in the surrounding neighborhood, S, in Figure 1 (a). This yields an $N \times N$ vector, $\mathbf{s}_k$, on that position, as shown in Figure 1 (b). We thereby generate $\mathbf{s}_k = \{x : x(i,j), \quad (i,j) \in W\}$, where W is an $N \times N$ window. The possible number of the surrounding vectors, $\mathbf{s}_k$, is $8N$, where N is the dimension of the missing block.

2.3. Recovery Vector

To restore missing blocks, a recovery window and vectors, $\{\mathbf{r}_i | i = 1, 2\}$, are introduced. The vectors include both known and missing pixels. Two positions for the recovery vectors are possible according to the dominating line in the surrounding blocks as shown in Figures 1 (c) and (d). The recovery windows in Figure 1 (c) are for a vertical line dominating area, and those in Figure 1 (d) are for a horizontal line dominating area. In each case, two $N \times N$ recovery vectors are made from the windows, and each vector consists of $(N-1) \times N$ known and $1 \times N$ missing pixels. After the missing pixels in a recovery vector are restored, each recovery window slides towards the other to recover next $1 \times N$ missing pixels as shown by the arrows in Figures 1 (c) and (d). We thereby generate $\mathbf{r}_k = \{x : x(i,j), \quad (i,j) \in W\}$, where W is a $N \times N$ window and $k = 1, 2$.

3. PROJECTIONS AND CONVEX SETS

In this section, projection operators and corresponding convex sets are formulated. Recovery vectors are alternately projected onto vertices of a convex hull and convex sets. Missing pixels are thereby restored iteratively.

3.1. Projection Operator P_1

The surrounding vectors, $\mathbf{s}_j$, extracted from surrounding blocks, are used to form a convex hull in an $N \times N$ dimensional space. Recovery vectors, $\mathbf{r}_i$, then, are projected onto the closest vertex of the convex hull in the DCT domain in mean-square sense.

Let $\mathbf{r}_i$ and $\mathbf{s}_j$, $(1 \leq i \leq 2,\ 1 \leq j \leq 8N)$, be recovery and surrounding vectors, respectively. The $\mathbf{s}$ vectors form a convex hull and, since there are $8N$ points in a N^2 dimensional space where $N \geq 8$, each vector, $\mathbf{s}_j$, becomes a vertex of the convex hull.

The closest vertex, $\{\mathbf{s}_{d_i} | i = 1, 2\}$, of the convex hull to the recovery vectors, $\{\mathbf{r}_i | i = 1, 2\}$, are found in the mean-square sense as $d_i = \arg\min_j ||\mathbf{r}_i - \mathbf{s}_j||$ for $1 \leq i \leq 2,\ 1 \leq j \leq 8N$. Or equivalently, $d_i = \arg\min_j ||\mathbf{R}_i - \mathbf{S}_j||$ for $1 \leq i \leq 2,\ 1 \leq j \leq 8N$, where $\mathbf{R}_i = \mathbf{T}\mathbf{r}_i$ and $\mathbf{S}_j = \mathbf{T}\mathbf{s}_j$, and where $\mathbf{T}$ is 2-D DCT kernel.

The recovery vectors in the DCT domain, $\{\mathbf{R}_i | i = 1, 2\}$ are then orthogonally projected onto the selected vertex as

$$P_{\mathbf{S}_{d_i}}(\mathbf{R}_i) = \frac{< \mathbf{R}_i,\ \mathbf{S}_{d_i} >}{\| \mathbf{S}_{d_i} \|^2} \cdot \mathbf{S}_{d_i} \qquad i = 1, 2 \qquad (3)$$

where $< \cdot, \cdot >$ is the inner product of two vectors, and $|| \cdot ||$ is the ℓ_2 vector norm.

Consequently, the projection operator P_1 translated to the DCT domain is

$$P_1 \cdot \mathbf{R}_i(u,v) = \begin{cases} P_{\mathbf{S}_{d_i}}(\mathbf{R}_i(u,v)) & \text{for } u, v \neq 0 \\ \mathbf{R}_i(u,v) & \text{otherwise} \end{cases} \qquad (4)$$

To preserve the DC level, the DC value in the recovery vectors, $\{\mathbf{r}_i | i = 1, 2\}$, is not changed.[1]

3.2. Projection Operator P_2

The convex set for the second projection operator P_2, onto the set C_2, is

$$C_2 = \left\{ \begin{array}{ll} \mathbf{f} : F_{min} \leq \mathbf{f}_n \leq F_{max} & \text{for } n \in L \\ \mathbf{f} : f_o & \text{otherwise} \end{array} \right\} \qquad (5)$$

where n is the pixel index, L is the set of missing pixels, f_o is original known value, and F_{min} and F_{max} are the minimum and maximum intensity of an image, respectively. Corresponding projection operator P_2 is,

[1] Rigorously, P_1 is not a projection onto a convex set. To truly project, the hull's skin must be computed and the corresponding projection onto it evaluated. Not only does P_1 require less computation, our experience shows the results are superior to the case where the true projection onto the convex hull is used.

$$P_2 \cdot x_k(i,j) = \begin{cases} F_{min} & \text{for } x_k(i,j) < F_{min},\ (i,j) \in L \\ F_{max} & \text{for } x_k(i,j) > F_{max},\ (i,j) \in L \\ x_k(i,j) & \text{otherwise } and\ (i,j) \in L \\ c_k(i,j) & \text{otherwise} \end{cases} \quad (6)$$

where (i,j) is the pixel index, $c_k(i,j)$ is the known pixel value, and L is the missing pixels of the recovery vectors, $\{\mathbf{r}_k | k = 1, 2\}$.

3.3. Projection Operator P_3

We consider a range constraint for the continuity with surroundings neighborhood of a restored block, described similar as [9]. Let $\vec{f_m}$ be the vector of missing pixels in a recovery vector, $\vec{f_0}$ be the vector of adjacent pixels to the missing line in the same vector, and $\vec{g}$ be $1 \times N$ vector of $\vec{f_m} - \vec{f_0}$. Vector $\vec{g}$ is illustrated as $\vec{g} = [(f_{m,0} - f_{0,0}), ..., (f_{m,N} - f_{0,N})]$. By setting the vector $\vec{g}$ as a bounded signal with a constant, α, the convex set for the third projection operator P_3 is obtained as

$$C_3 = \{\ \vec{g} :\ |\vec{g}_n| \leq \alpha\ \} \quad (7)$$

where n is the pixel index and α is a constant. The value of γ can be set to the maximum value of differences between pixels which are adjacent to the missing block in the surrounding neighborhood. Consequently, the projection operator P_3 is,

$$[P_3 \cdot f]_{m,n} = \begin{cases} f_{0,n} - \alpha & \text{for } \vec{g}_n < -\alpha,\ m \in L \\ f_{0,n} + \alpha & \text{for } \vec{g}_n > \alpha,\ m \in L \\ f_{m,n} & \text{otherwise} \end{cases} \quad (8)$$

where n and m are the pixel indices, and L is the missing pixels of the recovery vectors, $\{\mathbf{r}_k | k = 1, 2\}$.

3.4. Iterative Algorithm for Pixel Interpolations

Missing pixels in recovery vectors are restored by iterative algorithm of alternating projections [6], $f_{i+1} = P_1 \cdot P_2 \cdot P_3 \cdot f_i$, where i is the iteration index, f is restored signal at iteration i, and P_j is j^{th} projection operator. Our experience shows $i = 2$ suffices for robust results.

4. SIMULATION RESULTS

To illustrate the effectiveness, the proposed algorithm is tested on the "Lena" (512 × 512 pixels) and first frame of "Foreman" (176 × 144 pixels). A block size of 8 × 8 pixels are applied to "Lena" for JPEG image and the size of 16 × 16 pixels are applied to "Foreman" for MPEG. Throughout the simulation, the initial point, f_0, of the missing pixels of recovery vector, $\mathbf{r}$, is set to the adjacent value of the known pixels.

The algorithm was compared with the interblock correlation interpolation scheme by Hemami and Meng [4] and fast DCT-based spatial domain interpolation technique by Alkachouh and Bellanger [7]. For the objective measure of the restored image quality, the peak signal-to-noise ratio (PSNR) is used [10], given by $PSNR = 10log\left(\frac{N \cdot M \cdot 255^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}|f(i,j)-\hat{f}(i,j)|^2}\right)$, where f is the original image and $\hat{f}$ is the restored image of N x M pixels. Table 1 shows the PSNR of restored images by the different algorithms. The proposed restoration outperforms the other procedures here and in other examples we tried. Restored images in Figure 2 show that the proposed scheme restores both monotone and complex textures faithfully.

Table 1. PSNR comparison among algorithms for different cases

	Lena	Lena(row missing)	Foreman
Hemami	31.86	26.86	-
Alkachouh	31.57	-	25.65
Proposed	34.22	29.51	30.10

ACKNOWLEDGMENT

This work is supported by a grant from the National Science Foundation.

5. REFERENCES

[1] Joint Photographic Experts Group, ISO-IEC/JTC1/SC2/WP8, "JPEG technical specification, revision 8,", JPEG-8-R8, Aug. 1990.

[2] ISO/IEC 13818-2 MPEG-2 Video Coding Standard, "ISO/IEC 13818-2, Generic coding of moving pictures and associated audio information-Part 2: Video,", 1995. ISO.

[3] Y. Wang, Q. F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Trans. Commun.*, vol 41, pp. 1544-1551, Oct. 1993

[4] S. S. Hemami and T. H.-Y. Meng, "Transform coded image reconstruction exploiting interblock correlation," *IEEE Trans. Image Processing*, vol. 4, no. 7, pp. 1023-1027, Jul. 1995.

[5] H. Sun and W. Kwok, "Concealment of damaged blocks transform coded images using projections onto convex sets," *IEEE Trans. Image Processing*, vol. 4, pp. 470-477, Apr. 1995.

[6] R.J. Marks II, "Alternating Projections onto Convex Sets", in *Deconvolution of Images and Spectra*, edited by Peter A. Jansson, San Diego: Academic Press, 1997, pp. 476-501.

[7] Z. Alkachouh and M. G. Bellanger, "Fast DCT-based spatial domain interpolation of blocks in images," *IEEE Trans. Image Processing*, vol. 9, no. 4, pp. 729-732, Apr. 2000.

[8] R. Gonzalez and R. Woods, *Digital Image Processing*, Addison-Wesley, 1992.

[9] Y. Yang and N. P. Galatsanos, "Removal of compression artifacts using projections onto convex sets and line process modeling," *IEEE Trans. Image Processing*, vol. 6, no. 10, pp. 1345-1357, Oct 1997.

[10] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.

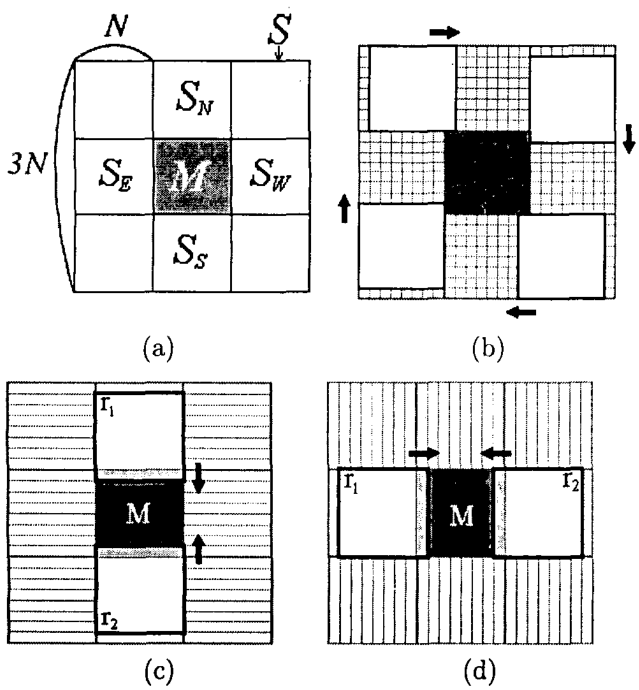

Fig. 1. (a) a missing block M, and surrounding blocks S. (b) $N \times N$ window move to make surrounding vectors. (c), (d) recovery window for vertical and horizontal edge orientation blocks, respectively.

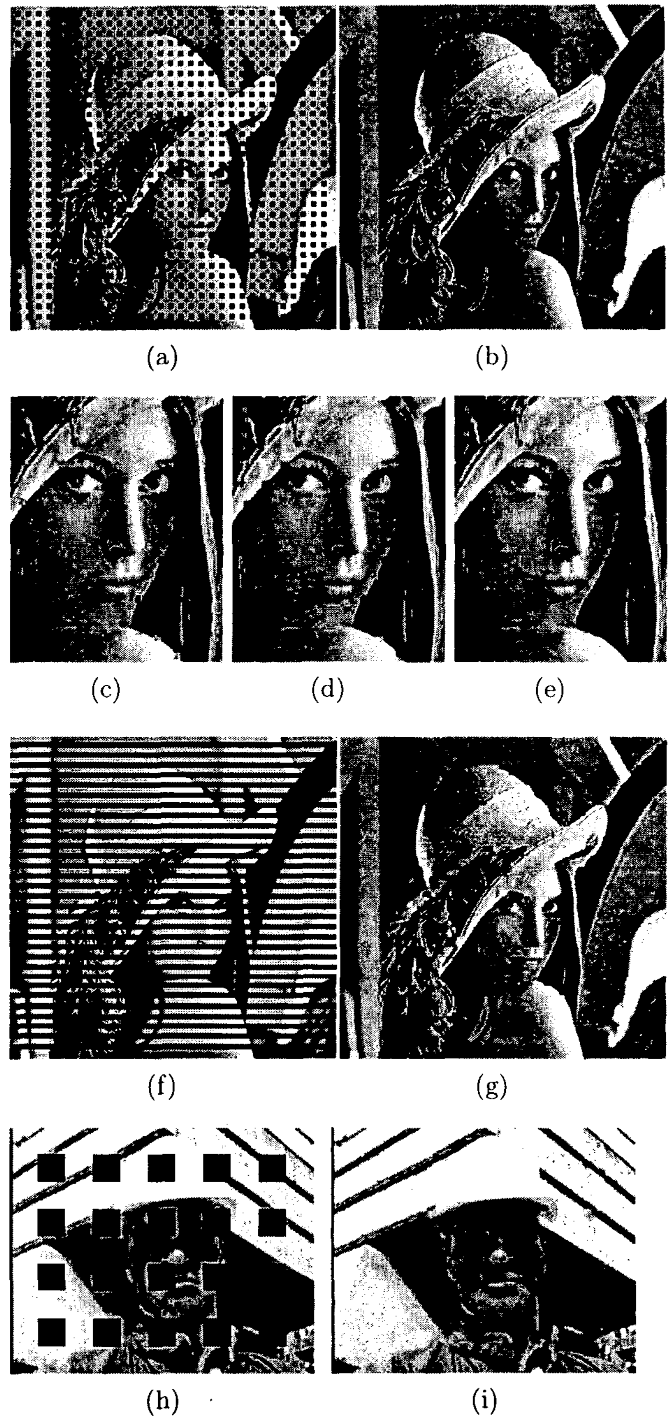

Fig. 2. (a), (f) damaged Lena.; (b), (e), and (g) restored Lena by the proposed algorithm.; (c), (d) restored Lena by Hemami's and Alkachouh's, respectively; (h), (i) damaged and restored Foreman by the proposed algorithm, respectively.