International Journal of Recent Engineering Science (IJRES),
ISSN: 2349-7157, *volume1issue1 Jan to Feb 2014*

Recovery of missing blocks in image using alternating projections

R.Sangari, A.Susi
Computer Science and Engineering
Coimbatore Institute of Technology
Coimbatore

Abstract

This paper handles the missing block recovery algorithms based on projections for block based image coding. The algorithms consider here are Alternating Projections method, Hamming Inpaint method and Euclidean Inpaint method. For corruption of blocks occurring at the time of file sending via network can also be restored using those techniques. The alternating projection method is working based on orthogonal projection onto constraint sets in a Hilbert space. The basic idea behind in Hamming Inpaint method and Euclidean Inpaint method are to fill the missing regions with available information from their surroundings. The attributes considered for the comparative study are Peak signal to noise ratio, and Mean square error . Experimental results demonstrate that the Alternating projection method provides effective performance compared to other methods.

1. Introduction

Transmission of still images and video over lossy packet networks requires a reconstruction problem at the decoder. The corrupted digital images are recovered by filling the proper information from surrounding data.The Alternating projection method is working based on orthogonal projections onto constraint sets in a Hilbert space. The Hamming inpaint method and Euclidean inpaint method implements image retrieval by using the inpaint concept. The basic idea behind Hamming inpaint method is to fill-in missing or modified regions with available information from their surroundings. This information can be automatically detected from the inpaint image. To find the distance of matching we use Hamming method in this algorithm. The Euclidean inpaint method fills the lost-blocks using the available information from their surroundings. The Euclidean distance measure method is used to find out the matching measurement.

2. Alternative Projection Method

This method presents a spectrally robust interpolative image-restoration method based on projections onto convex sets and onto a line in Hilbert space defined by the best-matched adjacent N×N pixels.. The algorithm enables restored blocks to sustain the spectral and edge structure of the surrounding blocks and, consequently, to have striking continuity with neighboring pixels.

line detection and vector forming

Image condition for missing-pixel interpolation is illustrated in Fig. 2.1. In Fig. 2.1(a), a missing block, **M**, with its surrounding neighborhood, **A,** is shown. The orientation of edges in the adjacent surrounding neighborhood, **A,** is assumed to expand its 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. To restore the missing block, **M**, two recovery vectors including correctly received pixels and estimated missing pixels are formulated.

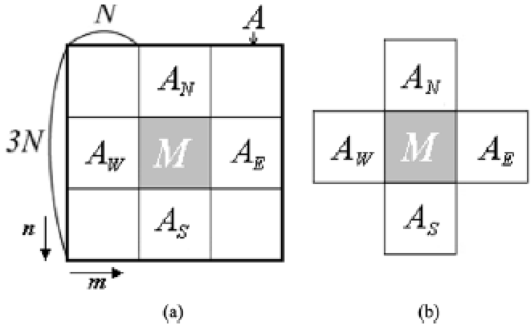

Figure 2.1 Missing block (M) with surrounding neighborhood blocks of correctly received data

Line Orientation Detection

A line detector in the spatial domain is applied to surrounding blocks to determine the line orientation of the area. The line masks L_v and L_h are applied to the surrounding blocks,

$$L_h = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad L_v = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

Corresponding responses R_h and R_v at coordinates m, n are

$\mathbf{R_h}$ = 2 ($x_{m,n-1} + x_{m,n} + x_{m,n+1}$) – ($x_{m-1,n-1}$ $x_{m-1,n} + x_{m-1,n+1} + x_{m+1,n-1} + x_{m+1,n} + x_{m+1,n+1}$) and

$\mathbf{R_v}$ = 2 ($x_{m-1,n} + x_{m,n} + x_{m+1,n}$) – ($x_{m-1,n-1} + x_{m,n-1} + x_{m+1,n-1} + x_{m-1,n+1} + x_{m,n+1} + x_{m+1,n+1}$). The magnitude of responses R_h and R_v at all m, n coordinates in the four surrounding blocks (A_E, A_W, A_N, and A_S) are computed as

$$E_h = \sum_{A_E,A_W,A_N,A_S} |R_h|, \quad E_v = \sum_{A_E,A_W,A_N,A_S} |R_v|.$$

Edge orientation is determined by E_h and E_v. If E_h is larger than E_v , the missing block is considered a horizontal line-dominating block.

Surrounding Vectors

The segmentation of the neighborhood area and corresponding vectors are formed by shifting an window N×N on every grid of pixels in the surrounding neighborhood A in Figure 2.1(a). This is illustrated in Figure 2.2. The process yields an N×N vector $\mathbf{s}_k$ on that position. We, thereby, generate

$$\mathbf{s}_k = \{x : x(m,n), (m,n) \in B\}$$

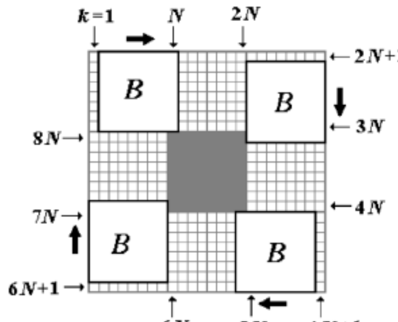

Figure 2.2 Missing block with surrounding neighborhood and N x N window B to make the surrounding vector s_i .

where B is an N × N window in A, m and n are pixel indices, and is an vector index. The number of the surrounding vectors $\mathbf{s}_k$ is 8N, and can be enumerated from 1 to 8N clockwise starting at the top-left corner, as shown in Fig.2.2 . If we define an N×N vector, $\mathbf{S}_k$ for,1 ≤ K ≤ 8N, which is the two-dimensional (2-D) DCT pair of the surrounding vector $\mathbf{s}_k$ then $S_k = T . S_k$ for $1 \le K \le 8N$, where T is 2-D DCT kernel.

Recovery Vectors

To restore a missing block, recovery vectors $\{\mathbf{r}_k | k = 1, 2\}$ are introduced. As shown in Fig.2.3 , according to the dominating line orientations in the surrounding blocks, two positions of the recovery vectors are employed. The position of recovery windows in Fig. 2.3(a) are for the vertical line-dominating area, while those in Fig. 2.3(b) are for the horizontal line-dominating area.. This is shown in Fig. 2.3. The gray in the windows indicates missing pixels, while the white-colored portion indicates correctly received pixels. We, thereby, generate

$$\mathbf{r}_k = \{x : x(m, n), (m, n) \in C\}$$

where C is an N × N window in A (for surrounding blocks) and M (for the missing block), m and n are pixel indices, and k is a vector index. Let the N×N vector , R_k for $1 \leq K \leq 2$, be the 2-D DCT pair of the surrounding vector $\mathbf{r}_k$,

$$\mathbf{R}_k = \mathbf{T} \cdot \mathbf{r}_k \text{ for } 1 \leq k \leq 2$$

After missing pixels in a recovery vector are restored, recovery windows slide in opposite directions to each other to extract a new recovery vector to restore the next N missing pixels. This is shown by the arrows in Fig. 2.3.

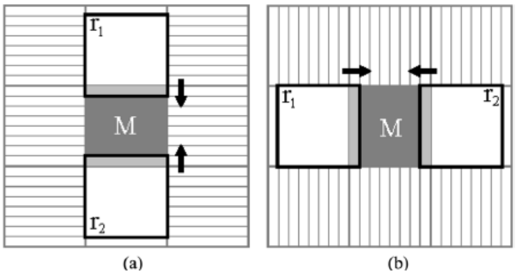

Figure 2.3.Missing block with surrounding neighborhood and two N x N recovery vectors r_i.

(a) Recovery vectors r_i for the vertical line-dominating area.

(b) Recovery vectors r_i for the horizontal line-dominating area.

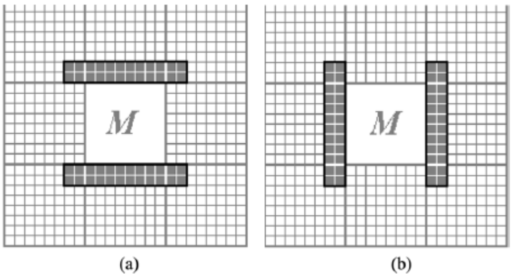

Figure 2.4.Areas for computing parameters α_1 and α_2. Upper and lower blocks in (a) are areas to compute α_1 and α_2, respectively. The left and right blocks in (b) are areas for α_1 and α_2, respectively. Pictured here are (a) the area for computing parameter α_i in the vertical line-dominating area and (b) the area for computing parameter α_i in the horizontal line-dominating area.

Projection Operator P_1 :

The vectors $\{\mathbf{s}_j | 1 \leq j \leq 8N\}$, extracted from the surrounding blocks,A, are used to form a convex hull in an N×N dimensional space. Recovery vectors, $\{\mathbf{r}_i | i = 1, 2\}$, are then projected in the DCT domain onto the line between closest1 vertex of the convex hull and the origin of the space. Let $\{\mathbf{r}_i | i = 1, 2\}$ and $\{\mathbf{s}_j | 1 \leq j \leq 8N\}$ be recovery and surrounding vectors, respectively.. Each vector S_j becomes a vertex of the convex hull. The closest vertices $\{\hat{\mathbf{s}}_i = \mathbf{s}_{d_i} | i = 1, 2\}$ of the convex hull to the $\{\mathbf{r}_i | i = 1, 2\}$ vectors are found in the mean-square sense

$$d_i = \arg\min_j ||\mathbf{r}_i - \mathbf{s}_j|| \text{ for } 1 \leq i \leq 2, \ \ 1 \leq j \leq 8N$$

or, equivalently

$$d_i = \arg\min_j ||\mathbf{R}_i - \mathbf{S}_j|| \text{ for } 1 \le i \le 2, \ \ 1 \le j \le 8N$$

where $\mathbf{R}_i = \mathbf{T} \cdot \mathbf{r}_i, \mathbf{S}_j = \mathbf{T} \cdot \mathbf{s}_j$, and T is a 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 $\hat{\mathbf{S}}_i$, as

$$P_{\hat{\mathbf{S}}_i}(\mathbf{R}_i) = \frac{\langle \mathbf{R}_i, \hat{\mathbf{S}}_i \rangle}{||\hat{\mathbf{S}}_i||^2} \cdot \hat{\mathbf{S}}_i \quad i = 1, 2$$

where $\langle \cdot, \cdot \rangle$ 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

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

Projection Operator P_2: Projection operator P_2 imposes constraints on the range on the restored pixel values. It operates in the spatial domain. The convex set C_2 for the projection operator P_2 is

$$C_2 = \{ f : F_{min} \le f_n \le F_{max} , \text{ for } n \in L \}$$

where L is the set of missing pixels and F_{min} and F_{max} are chosen minimum and maximum intensities of an image, respectively. The corresponding projection operator P_2 is a threshold.

$$P_2 \cdot f_n = \begin{cases} F_{\min}, & \text{for } f_n < F_{\min}, n \in L \\ F_{\max}, & \text{for } f_n > F_{\max}, n \in L \\ f_n, & \text{for } F_{\min} \le f_n \le F_{\max}, n \in L \\ c_n, & \text{otherwise} \end{cases}$$

Where " n " is the pixel index "cn" is the known pixel value and " L" is the missing pixel of the recovery vectors.

Projection Operators P_3:

A range constraint for continuity within the surroundings neighborhood of a restored block is imposed for smooth reconstruction of a damaged image. Let be the vector of missing pixels in a recovery vector, g be the vector of adjacent pixels to the missing line in the same vector, and h be $N \times 1$ vector of $\mathbf{f} - \mathbf{g}$. Define $\mathbf{h} = [(f_0 - g_0), \ldots, (f_N - g_N)] = \mathbf{f} - \mathbf{g}$. By setting the vector $\mathbf{h}$ as a bounded signal with a constant, α, the convex set for the third projection operator P_3 can be obtained as

$$C_3 = \{ \mathbf{h} : |h_n| \le \alpha \}$$

where n is the pixel index and α is a predetermined constant. The value of α 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

$$P_3 \cdot f_n = \begin{cases} g_n - \alpha, & \text{for } h_n < -\alpha \\ g_n + \alpha, & \text{for } h_n > \alpha \\ f_n, & \text{otherwise} \end{cases}$$

where $1 \le n \le N$.

3. EUCLIDEAN INPAINT METHOD

In this missing block recovery method the matching measurement is done by Euclidean distance method.. The minimum matching value patch is desired as the best match patch. This best match is used to replace the target patch. Similarity measure is a key component in image matching. Traditionally, Euclidean distances are used to measure the similarity between the target patch and source patch. The smaller distance is more similar to the target patches. The formula used in this Euclidean distance measure is "The Sum Of Squared Difference" That is

DistValue= Σ (MissingBlockPixel(i,j) –SourcePatch(i,j)) ^ 2

4. HAMMING INPAINT METHOD

The Hamming distance method is also used to compute the similarity measurement between the target patch (Missing Block) and each source patch. This method produces a single value which represents the matching level of the two patches.The mathematical concept of this method is Sum of differences of squared values. This can be shown as

DistValue= Σ (Missing BlockPixel (i,j)^2 – SourcePatch(i,j) ^ 2)

5.Analysis

For analysis, we are considering the input image which is provided in fig.5.1

Fig5.1

Table 5.1 Analysis on peak signal to noise ratio(PSNR)

Recovery methods	PSNR value(db)
Alternating projection method	56.0085
Hamming Inpaint method	28.0479
Euclidean Inpaint method	29.3741

Table 5.2 Analysis on mean square error (MSE)

Recovery methods	MSE value
Alternating projection method	0.1630
Hamming Inpaint method	101.9282
Euclidean Inpaint method	75.1057

Table 5.1 gives the performance, based on peak signal to noise ratio(PSNR).Here, for a given input image, the alternating projection method is the highest PSNR value ,whereas the Hamming Inpaint method is the lowest PSNR value.

From table 5.2,it is known that the alternating projection method shows least error value , and so it is preferable method for retrieving missing blocks.

6.Conclusion

This paper handles the image block removal algorithms which are used to retrieve or retouch the lost-blocks of digital image photographs. The methods considered for this work produce expected outputs and the results are analyzed.

The Peak signal to noise ratio comparison identify that the “Alternating Projections method” has greater value and the “Hamming Inpaint method” has lower value. Mean square error comparison declares that “Hamming Inpaint method” produces high value and the “Alternating Projections method” produces low value. Based on the studies performed, it is concluded that the alternating projection method is the best method among the three methods.

7. Bibliography

[1] *ISO/IEC 13 818-2, Generic Coding of Moving Pictures and Associated Audio Information-Part 2: Video,* ISO/IEC 13 818-2 MPEG-2 Video Coding Standard, 1995.

[2] Y. Wang, Q. F. Zhu, and L. Shaw, “Maximally smooth image recovery in transform coding,” *IEEE Trans. Commun.*, vol. 41, no. 10, pp. 1544–1551, Oct. 1993.

[3] J.W. Park, J. W. Kim, and S. U. Lee, “DCT coefficients recovery-based error concealment technique and its application to the MPEG-2 bit stream error,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 8, pp. 845–854, Dec. 1997.
[4] X. Lee, Y.-Q. Zhang, and A. Leon-Garcia, “Information loss recovery for block-based image coding techniques-a fuzzy logic approach,” *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 259–273, Mar. 1995.
[5] S.S.Hemami and T.H.-Y.Meng, “Transform coded image reconstruction exploiting interblock correlation,” *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 1023–1027, Jul. 1995.
[6] Z. Alkachouh and M. G. Bellanger, “Fast DCT-based spatial domain interpolation of blocks in images,” *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 729–732, Apr. 2000.