International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 13 (2015) pp 33677-33679
© Research India Publications. http://www.ripublication.com

An Error Concealment Technique Based on Directional Interpolation

Seung-jong Kim
Dept. of Computer Science & Information Systems, Hanyang Women's University
200 Salgoji-gil, Seongdong-gu, Seoul, 133-817, South Korea

Abstract- Block-based error concealment techniques using directional interpolation is proposed to repair damaged blocks. In the proposed method, the edge directions are determined by finding the maximum correlation coefficients across the boundary pixels of the lost block. Once the edge orientation is found, the lost block is interpolated linearly along the determined edge direction. When applied to compressed images, the proposed method shows superior subjective and objective quality to conventional methods.

Keywords- Directional interpolation, concealment, correlation, edge orientation.

1. Introduction

Transmission of compressed image over Internet may introduce cell losses into the transmitted data and it degrades severely the quality of the received image [1-2]. For decades, various error concealment techniques have been proposed [3-8]. Sun et al. [3] proposed the projection onto convex sets (POCS) for error concealment. They used spatially correlated edge information obtained from a large number of surrounding pixels to recover the lost blocks. Wang et al. [4] proposed a best neighborhood matching (BNM) algorithm to recover lost blocks. Li and Orchard [5] introduced an orientation adaptive interpolation algorithm with sequential recovery. Zhao et al. [6] introduced an error concealment technique based on optimized directional decision and extrapolation in the spatial domain.

This paper proposes the method which conceals errors effectively that occurs due to bit errors during the transmission of still image or I-picture of video using the block-based compression technique. The proposed method considers the edge direction in the spatial domain. First, the edge direction is estimated by measuring the largest correlation between boundary pixels of the neighbor blocks of error, and then error block is reconstructed by interpolating linearly toward that direction.

2. Proposed Algorithm

Since edge information plays an important role in the spatial error concealment method, the edge orientation of the lost block is estimated by calculating correlation coefficients of the image blocks (top, bottom, left and right) around the lost block as shown in Fig. 1(b). For the determined direction, the proposed directional interpolation is carried out along the direction to repair the pixels of the lost block as shown in Fig. 1(d).

$$\rho_{P_k,\,Q_k} = \frac{\mathrm{cov}(P_k, Q_k)}{\sigma_{P_k}\sigma_{Q_k}} \quad (1)$$

In Eq. (1), cov(·) is the covariance, σ is the standard deviation, and ρ is the correlation coefficient. Two sets $\{P_k\}$ and $\{Q_k\}$ used in calculating correlation coefficient, have the various elements according to edge direction k. That is, when k is 0 or 4, only 8 boundary pixels of the adjacent block (top, bottom, left and right) are needed to calculate and decide the exact horizontal or vertical edge direction of the lost block. However it is necessary to increase the number of boundary pixels relatively when k is neither 0 nor 4 as shown in Fig. 1(b), since the 8 boundary pixels are not sufficient to determine the exact edge direction of the lost block. Fig. 1(b) represents the boundary pixels taken for the edge direction k=1 (22.5 degree), and the points of the boundary pixels where each line intersects the vertical line with x=0 and the horizontal line with y=N+1, are $\{P_1\}$. Also the points of the boundary pixels where each line intersects the horizontal line with y=0 and the vertical line with x=N+1, are $\{Q_1\}$. If there is no original boundary pixel on the points where each line meets x-axis and y-axis on, the linear interpolation using neighboring boundary pixels is carried out. The proposed directional interpolation method is performed in 8 steps, and the overall algorithm is shown as follows (See Fig. 1).

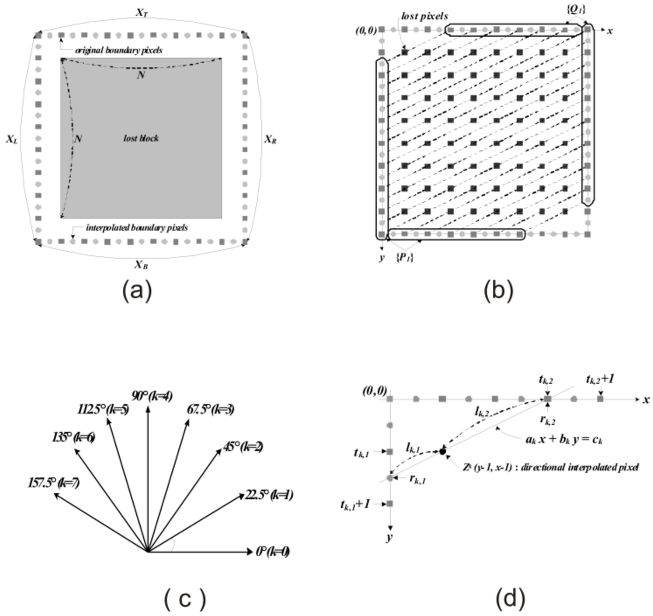

Fig. 1. Graphical representation of the proposed algorithm. (a) Set of the boundary pixels (b) Set of the boundary pixels taken for the edge direction k=1 (22.5 degree) (c) Edge orientations (d) Directional interpolation scheme

• **Step 1**: Derive the linear equation for the edge orientation k:

$$a_k x + b_k y = c_k, \quad x, y = 1, 2, ..., N \tag{2}$$

• **Step 2**: Calculate a_k for coefficient of x and b_k for coefficient of y.

• **Step 3**: Calculate the intercept c_k of line in Fig. 1(d). Intercept c_k depending on k is derived from Eq. (2).

• **Step 4**: Calculate the coordinates $t_{k,1}$ and $t_{k,2}$ of boundary pixels required during the directional interpolation. It depends on k and range of intercept c_k. The $t_{k,1}$ and $t_{k,2}$ are intersects with line x-axis and y-axis respectively. This helps step 5 to decide which boundary pixels need to be used in order to carry out the directional interpolation.

• **Step 5**: From the Fig. 1, there is no boundary pixel where the x-axis and the y-axis meet. Therefore, the boundary pixel value of any straight line that meets x-axis and the y-axis is needed, and this can be obtained by linear interpolation of the neighboring boundary pixels. The interpolated boundary pixels are $r_{k,1}$ and $r_{k,2}$, which can be obtained from k and c_k.

• **Step 6**: Obtain the slope $m_{k,1}$ and $m_{k,2}$ from Eq. (3). The derived slope is used for calculating the length required during directional interpolation in step 7.

$$m_{k,1} = a_k / b_k, \quad m_{k,2} = b_k / a_k \tag{3}$$

• **Step 7**: For the given k and c_k, the distances $l_{k,1}$ and $l_{k,2}$ can be obtained as shown in Table 3. These distances are used as weights in Eq. (4) to consider the higher correlation of closer boundary pixel value.

• **Step 8**: Interpolate with Eq. (4) to conceal the lost block. For the given edge direction k, the interpolated boundary pixels $r_{k,1}$, $r_{k,2}$ and the distances $l_{k,1}$, $l_{k,2}$ from step 6 and 7, are used to conceal the lost block.

Table 1. Coordinates $t_{k,1}$ and $t_{k,2}$ (N=8)

k	a_k	b_k	c_k	$t_{k,1}$	$t_{k,2}$
0	0	1	$a_k N \le c_k \le b_k N$	c_k	c_k
1, 7	1	2	$c_k \le a_k(N+1)$	c_k / b_k	c_k / a_k
			$a_k(N+1) < c_k \le b_k(N+1)$	c_k / b_k	$(c_k - a_k(N+1))/b_k$
			$c_k > b_k(N+1)$	$(c_k - b_k(N+1))/a_k$	$(c_k - a_k(N+1))/b_k$
2, 6	1	1	$c_k \le a_k(N+1)$	c_k / b_k	c_k / a_k
			$c_k > b_k(N+1)$	$(c_k - b_k(N+1))/a_k$	$(c_k - a_k(N+1))/b_k$
3, 5	2	1	$c_k \le b_k(N+1)$	c_k / b_k	c_k / a_k
			$b_k(N+1) < c_k \le a_k(N+1)$	$(c_k - b_k(N+1))/a_k$	c_k / a_k
			$c_k > a_k(N+1)$	$(c_k - b_k(N+1))/a_k$	$(c_k - a_k(N+1))/b_k$
4	1	0	$b_k N \le c_k \le a_k N$	c_k	c_k

Table 2. Interpolated boundary pixels $r_{k,1}$ and $r_{k,2}$ (N=8)

k	c_k	$r_{k,1}$	$r_{k,2}$
0	$a_k N \le c_k \le b_k N$	$X_L(t_{k,1})$	$X_R(t_{k,2})$
1, 7	$c_k \le a_k(N+1)$	$[(b_k - 1)X_L(t_{k,1}) + X_L(t_{k,1}+1)]/b_k$	$[(a_k - 1)X_T(t_{k,2}) + X_T(t_{k,2}+1)]/a_k$
	$a_k(N+1) < c_k \le b_k(N+1)$	$[(b_k - 1)X_L(t_{k,1}) + X_L(t_{k,1}+1)]/b_k$	$[(b_k - 1)X_R(t_{k,2}) + X_R(t_{k,2}+1)]/b_k$
	$c_k > b_k(N+1)$	$[(a_k - 1)X_B(t_{k,1}) + X_B(t_{k,1}+1)]/a_k$	$[(b_k - 1)X_R(t_{k,2}) + X_R(t_{k,2}+1)]/b_k$
2, 6	$c_k \le a_k(N+1)$	$[(b_k - 1)X_L(t_{k,1}) + X_L(t_{k,1}+1)]/b_k$	$[(a_k - 1)X_T(t_{k,2}) + X_T(t_{k,2}+1)]/a_k$
	$c_k > b_k(N+1)$	$[(a_k - 1)X_B(t_{k,1}) + X_B(t_{k,1}+1)]/a_k$	$[(b_k - 1)X_R(t_{k,2}) + X_R(t_{k,2}+1)]/b_k$
3, 5	$c_k \le b_k(N+1)$	$[(b_k - 1)X_L(t_{k,1}) + X_L(t_{k,1}+1)]/b_k$	$[(a_k - 1)X_T(t_{k,2}) + X_T(t_{k,2}+1)]/a_k$
	$b_k(N+1) < c_k \le a_k(N+1)$	$[(a_k - 1)X_B(t_{k,1}) + X_B(t_{k,1}+1)]/a_k$	$[(a_k - 1)X_T(t_{k,2}) + X_T(t_{k,2}+1)]/a_k$
	$c_k > a_k(N+1)$	$[(a_k - 1)X_B(t_{k,1}) + X_B(t_{k,1}+1)]/a_k$	$[(b_k - 1)X_R(t_{k,2}) + X_R(t_{k,2}+1)]/b_k$
4	$b_k N \le c_k \le a_k N$	$X_T(t_{k,1})$	$X_B(t_{k,2})$

Table 3. Distances of $l_{k,1}$ and $l_{k,2}$ (N=8)

k	c_k	$l_{k,1}(y,x)$	$l_{k,2}(y,x)$
0	$a_k N \le c_k \le b_k N$	x	$N+1-x$
1, 7	$c_k \le a_k(N+1)$	$\sqrt{x^2 + (m_{k,1}x)^2}$	$\sqrt{y^2 + (m_{k,2}y)^2}$
	$a_k(N+1) < c_k \le b_k(N+1)$	$\sqrt{x^2 + (m_{k,1}x)^2}$	$\sqrt{(N+1-x)^2 + (m_{k,1}(N+1-x))^2}$
	$c_k > b_k(N+1)$	$\sqrt{(N+1-y)^2 + (m_{k,2}(N+1-y))^2}$	$\sqrt{(N+1-x)^2 + (m_{k,1}(N+1-x))^2}$
2, 6	$c_k \le a_k(N+1)$	$\sqrt{x^2 + x^2}$	$\sqrt{y^2 + y^2}$
	$c_k > b_k(N+1)$	$\sqrt{(N+1-y)^2 + (N+1-y)^2}$	$\sqrt{(N+1-x)^2 + (N+1-x}$
3, 5	$c_k \le b_k(N+1)$	$\sqrt{x^2 + (m_{k,1}x)^2}$	$\sqrt{y^2 + (m_{k,2}y)^2}$
	$b_k(N+1) < c_k \le a_k(N$	$\sqrt{(N+1-y)^2 + (m_{k,2}(N+1-y))^2}$	$\sqrt{y^2 + (m_{k,2}y)^2}$
	$c_k > a_k(N+1)$	$\sqrt{(N+1-y)^2 + (m_{k,2}(N+1-y))^2}$	$\sqrt{(N+1-x)^2 + (m_{k,1}(N+1-x))^2}$
4	$b_k N \le c_k \le a_k N$	y	$N+1-y$

$$Z^k_{m,n}(y-1, x-1) = \frac{r_{k,1} \times l_{k,2}(y,x) + r_{k,2} \times l_{k,1}(y,x)}{l_{k,1}(y,x) + l_{k,2}(y,x)}, \quad x, y = 1, 2, ..., N \tag{4}$$

4. Experimental Results

The performance of the proposed algorithm is evaluated on the 512×512 "Lena" image. Fig. 2(a) shows the original image. The corrupted image is shown in Fig. 2(b). It shows that there are 225 blocks losses with 16×16 block size. Fig. 2(c) shows the restored image with Aign's algorithm [7].

Fig. 2(d) shows the result of the proposed algorithm. The simulation results show that the image edge can be indeed recovered. As compared with other spatial interpolation methods, the proposed algorithm can provide better performance not only in high PSNR values but also in good subjective image quality.

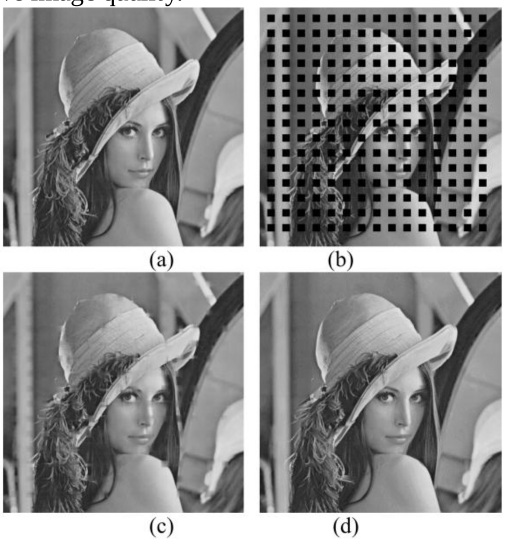

Fig. 2. Concealment results when 225 blocks (22%) are lost. (a) Original image (b) Lost blocks (12.20dB) (c) Aign's algorithm (29.57dB) (d) Proposed algorithm (34.14dB)

5. Conclusions

In this paper, an effective block-based error concealment method was proposed. The edge directions are determined by finding the maximum correlation coefficients across the boundary pixels of the lost blocks. Once the edge orientation is found, the lost blocks are interpolated linearly along the determined edge direction. Experimental results show the proposed method achieves an excellent concealment performance for lost blocks.

Acknowledgements

This research was supported by 2014-1 Hanyang Women's University Research Fund.

References

[1] Wei-Ying Kung, Chang-Su Kim and C.-C. Jay Kuo, Spatial and Temporal Error Concealment Techniques for Video Transmission over Noisy Channels, *IEEE Trans. on Circuits and Systems for Video Technology* (2006), Vol. 16, No. 7, pp. 789-802.

[2] Razieh K. and Ali A., Block and Region Based Image Error Concealment Using Fragile Watermarking in the Spatial Domain, *International Journal of Advanced Science and Technology* (2012), Vol. 46, No. 9, pp. 1-16.

[3] H. Sun and W. Kwok, Concealment of Damaged Block Transform Coded Images using Projection onto Convex Set, *IEEE Trans. on Image Processing* (1995), Vol. 4, pp. 470-477.

[4] Z. Wang, Y. Yu and D. Zhang, Best Neighborhood Matching: An Information Loss Restoration Technique for Block-Based Image Coding Systems, *IEEE Trans. on Image Processing* (1998), Vol. 7, No. 7, pp. 1056-1061.

[5] X. Li and M. T. Orchard, Novel Sequential Error-Concealment Techniques using Orientation Adaptive Interpolation, *IEEE Trans. on Circuits and Systems for Video Technology* (2002), Vol. 12, No. 10, pp. 857-864.

[6] Y. Zhao, Q. Chen, H. Chen and M. Rupp, Spatial Error Concealment Using Optimized Directional Decision and Extrapolation, *The 5th International Conference on Visual Information Engineering* (2008), China, pp. 658-661.

[7] S. Aign and K. Fazel, Temporal and Spatial Error Concealment Techniques for Hierarchical MPEG-2 Video Codec, *IEEE International Conference on Communication* (1995), Vol. 3, pp. 1778-1783.

[8] M. B. Seo, C. J. Lee, D. G. Kim, A Water Surface Detection Method by Correlation Analysis of Watermark Images with Time Interval, *Journal of the Korea Academia-Industrial cooperation Society* (2013), Vol. 14, No. 1, pp. 470-477.

Received: Month xx, 20xx