Selective Pixel Interpolation for Spatial Error Concealment

Yi Ge #, Bo Yan #, Kairan Sun #, Hamid Gharavi *

School of Computer Science, Fudan University
Shanghai, PRC 200433

* *National Institute of Standards and Technology*
100 Bureau Drive, Gaithersburg, MD USA

***Abstract*—This paper proposes an effective algorithm for spatial error concealment with accurate edge detection and partitioning interpolation. Firstly, a new method is used for detecting possible edge pixels and their matching pixels around the lost block. Then, the true edge lines can be determined, with which the lost block is partitioned. Finally, based on the partition result, each lost pixel can be interpolated with correct reference pixels, which are in the same region with the lost pixel. Experimental results show that the proposed spatial error concealment method is obviously superior to the previous methods for different sequences by up to 4.04 dB.**

***Index Terms*—spatial error concealment, edge detection, partitioning interpolation.**

I. Introduction

In video transmission, the highly compressed video signal is vulnerable to channel errors. Error concealment thus becomes a good way to conceal these errors without incurring much overhead and delay. Spatial error concealment techniques are performed in the spatial domain by interpolating the corrupted pixels with the surrounding pixels, which are normally applied in situations where information from neighboring frames is not available (INTRA coded frames or images), hence having to rely entirely on the information from intact neighboring macroblocks (MBs).

For this purpose, numerous methods [1]–[11] have been proposed by researchers, which can detect the smoothness and directional structure within the processed area. However, when there are more than one edges in the corrupted image regions and/or the corrupted regions consist of complex textures, the methods discussed above may not have the desired capability of detecting several edges robustly. As a result, the recovery quality of lost pixel information in corrupted regions may be poor.

In this paper, we propose a new solution for spatial error concealment, which offers a better method for multiple edges' detection and geometric interpolation based on the partition of the lost block. Firstly, for the correctly received pixels around the lost block, we apply a new method to detect the possible edge pixels and their matching pixels. Then the detected edge lines are used for partitioning the lost block. Finally, based on the partition results, each lost pixel can be interpolated with the correct reference pixels, which are in the same region as the lost pixel.

This paper is organized as follows. In Section II, we give a review of existing methods. In Section III, we propose a new spatial error concealment algorithm. Then we evaluate the performance of the proposed algorithm by simulations in Section IV. Finally, in Section V we draw conclusions.

II. Conventional Methods

Recently, many spatial error concealment methods have been proposed in order to conceal channel errors for video transmission. Among them, a typical method [6], uses a dynamic method for the texture estimation and geometric interpolation. The nuclear idea of [6] is a variant of dynamic programming to analyze the matching points of two sequences, which could be applied to estimate the texture of the layers outside the lost blocks. As shown in Fig.1, [6] uses the information in two-pixel-thick layers around the lost block, labeled as the outer layer and the inner layer.

In [6], four pairs of outer layers are firstly selected, including the outer layer pixel sequences from the upper and lower of the lost region, the pixel sequences from the left and the right of the lost region, the pixel sequences from the upper and the left of the lost region, and pixel sequences from the lower and the right of the lost region. Then [6] uses these four pairs of outer layers to interpolate the inner layers, and compares the PSNR between the interpolated inner layers and the actual received inner layers. The pair of inner layers with the highest PSNR will be selected to make the interpolation process. [6] believes that this selection provides the best pairs of inner layers, which have the same texture as the lost block. After finding the pair of inner layers, geometric interpolation could be executed by using only the selected pair of inner layers and thus the whole lost block could be recovered.

This method is efficient and effective for some cases. However, for the lost block that has multiple edges, the result is still not satisfying. In addition, when the lost blocks have strong edges and both sides of the edge are the flat area, the performance is not good. This is because that the selection process mentioned above will choose the inner layers in the flat areas instead of the edge areas. Moreover, the final step

This work is supported by NSFC (Grant No.: 61073067).

978-1-4577-1434-4/11/$26.00 ©2011 IEEE MMSP 2011

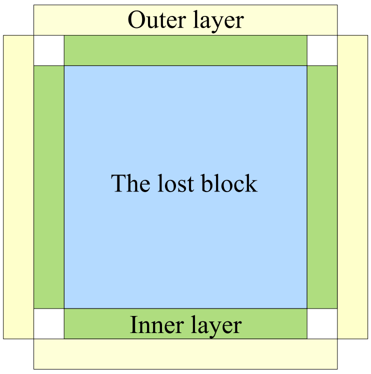


Fig. 1. The inner and outer layers of the lost block.

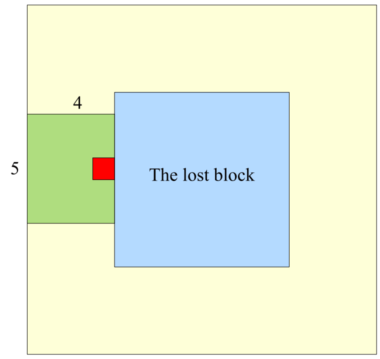


Fig. 2. The detection of the matching pixel.

is not based on area partitioning and is simply geometrical interpolation due to the lack of edge detection. Therefore, in order to solve these problems, it is necessary to propose an improved spatial error concealment method.

III. Our Proposed Method

In this section, we propose a new method for spatial error concealment. We assume that the erroneous blocks are exactly 8×8 blocks. We will use the information contained in the correctly received part to conceal the erroneous block.

A. Edge Detection

We compute the luminance difference of each pixel $p(i, j)$ in the layer outside of the 8×8 lost block in both X-direction (the above pixel) and Y-direction (the left pixel). The difference is calculated as:

$$\begin{cases} D_x = p(i,j) - p(i-1,j) \\ D_y = p(i,j+1) - p(i,j) \end{cases} \quad (1)$$

For those pixels in the bottom layer of the erroneous block, we will use the pixel beneath pixel $p(i, j)$ because the above pixel is lost. Thus D_x is computed as:

$$D_x = p(i+1,j) - p(i,j) \quad (2)$$

Similarly, for those pixels in the right layer of the erroneous block, we will use the pixel to the right of pixel $p(i, j)$. As a result, D_y is derived as:

$$D_y = p(i,j+1) - p(i,j) \quad (3)$$

Thus, the luminance difference of pixel $p(i, j)$ is:

$$\begin{cases} |D| = \sqrt{D_x^2 + D_y^2} \\ D_\theta = \tan^{-1}(D_y/D_x) \end{cases} \quad (4)$$

Once we get all the luminance differences of n pixels in the outer layer of the erroneous block, named as $D_1, D_2, ..., D_n$, we can compute the threshold of difference Th_{dif} as:

$$Th_{dif} = \frac{Mean + Max}{2} \quad (5)$$

where

$$Mean = \frac{\sum_{k=1}^{n} |D_k|}{n} \quad (6)$$

$|D_k|$ is the module of luminance difference for each pixel in the outer layer. Max is the maximum value of $|D_k|$.

If the luminance difference of the pixel in the outer layer is larger than Th_{dif}, this pixel will be selected as the edge pixel.

B. Searching the Matching Pixels

Then we will look for the matching pixel of each edge pixel in order to determine the edge line. The matching pixel and the edge pixel are supposed to be on the same edge. Thus, if the matching pixel is detected, the edge will be determined.

Suppose that the pixel marked in red around the lost block in Fig.2 is an edge pixel. We set a 5×4 window in the correctly received part, as shown in Fig.2. The matching pixel is expected to be found in this window. In order to find the matching pixel, we first compare the luminance between the edge pixel and all other pixels in the window. Since the matching pixel and the edge pixel should have similar luminance due to being in the same region, we select five pixels which have the most similar luminance as the candidate edge pixel.

Let's use E_θ to denote the slope angle of the edge determined by the edge pixel and its candidate matching pixel, which is derived as:

$$E_\theta = \frac{y_{edge} - y_{match}}{x_{edge} - x_{match}} \quad (7)$$

where (x_{edge}, y_{edge}) is the coordinate of the edge pixel and (x_{match}, y_{match}) is the coordinate of the candidate matching pixel.

We then use the sobel operator to compute the gradient of these five candidate pixels. Let's use G_θ denote the gradient direction of the candidate matching pixel. For these five pixels, we use a vector to evaluate their potential to be the matching pixel, which is

$$(\Delta Slope, \Delta Gradient, \Delta Lum/255)$$

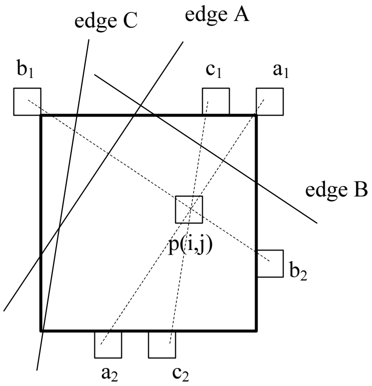


Fig. 3. The partitioning interpolation.

TABLE I
PSNR COMPARISON FOR DIFFERENT SEQUENCES.

Sequence	[3]	[6]	Proposed	Gain1	Gain2
Silent	25.56	27.59	29.35	3.79	1.76
Bus	21.65	21.88	23.41	1.76	1.53
Hall	23.22	24.72	27.26	4.04	2.54
Football	21.64	22.39	23.96	2.32	1.57
Coastguard	24.70	26.26	28.01	3.31	1.75
Flower	19.83	20.66	21.29	1.46	0.63
Lena	22.30	24.19	26.14	1.89	1.95

where

$$\begin{cases} \Delta Slope = \|D_\theta - E_\theta\| \\ \Delta Gradient = \|D_\theta - G_\theta\| \end{cases} \tag{8}$$

ΔLum is the luminance difference between the candidate matching pixel and the edge pixel, which is divided by 255 to ensure the three elements of the vector are in the same magnitude.

Finally, we compute the module of the vector for each of the five candidate pixels. Since the matching pixel has the closest characteristics to the edge pixel, we select the pixel with the minimal vector module as the matching pixel. Once the matching pixel is selected, we can get the slope of the edge. As a result, the edge is determined.

C. Partitioning Interpolation

After the possible edges are detected, we can use these edges to partition the erroneous block. Based on the partition result of the erroneous block, the lost pixel can be interpolated. In order to conceal the corrupted pixel $p(i,j)$, only those uncorrupted pixels in the same region with $p(i,j)$ will be used for interpolation.

As shown in Fig.3, suppose that there are three edges (A, B and C) in the erroneous block. Firstly, according to the edge slope of edge A and the position of the lost pixel $p(i,j)$, we can determine two reference pixels (a_1 and a_2) in the outer layer of the erroneous block for interpolation. Pixels a_1, a_2 and $p(i,j)$ are in the same line, whose slope is the same as that of edge A. In the same way, we can find reference pixel

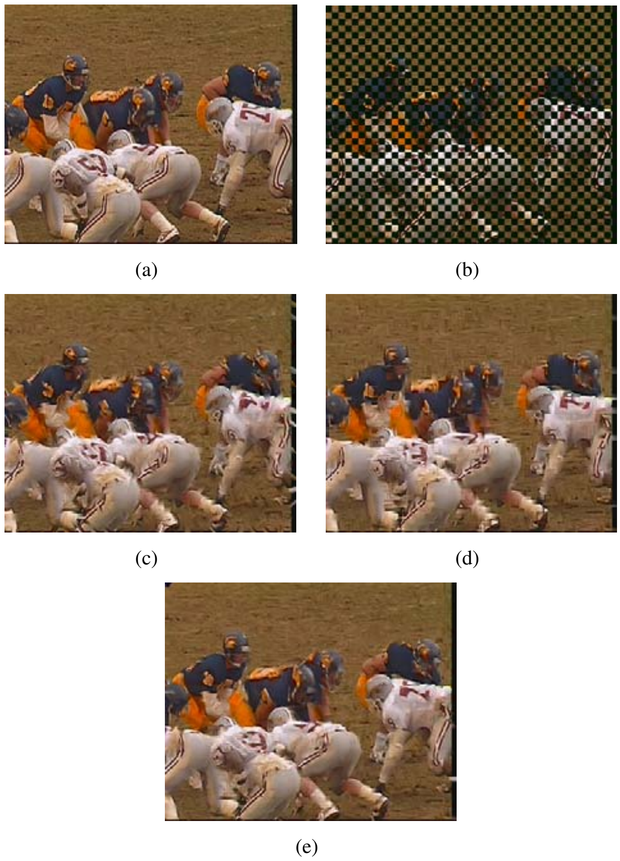


Fig. 4. The subjective results of spatial concealment for sequence "Football" with BLR=50%. (a) correct image; (b) corrupted image; (c) concealed with [3]; (d) concealed with [6]; (e) concealed with our proposed method.

b_1, b_2, c_1 and c_2 according to edge B and edge C respectively. Then we will check if each reference pixel is in the same side with $p(i,j)$ of all the edges. Only the reference pixel, which is in the same part with $p(i,j)$, is the valid reference for interpolation. As shown in Fig.3, reference pixel a_2, c_2 and b_2 are valid reference pixels to interpolate $p(i,j)$.

After all the valid reference pixels are obtained, $p(i,j)$ can be interpolated as:

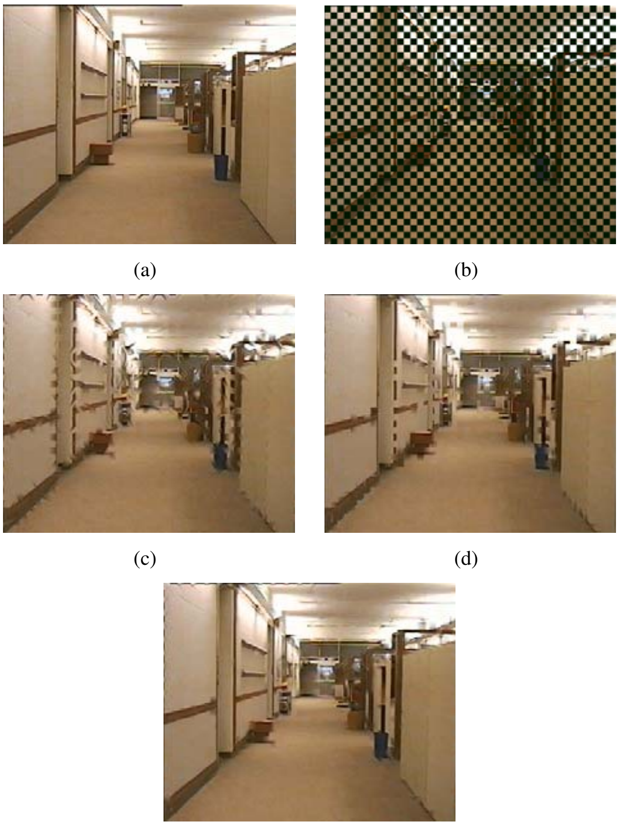
$$P_{i,j} = \frac{\sum_{k=1}^{n} P_k/d_k}{\sum_{k=1}^{n} 1/d_k} \tag{9}$$

where P_k is the luminance of the valid reference pixels for interpolation and d_k denotes the distance to $p(i,j)$.

After the partitioning interpolation, most of the pixels will be interpolated successfully. However, there might be a small amount of pixels which are still not interpolated due to no valid reference pixels. In our method, these pixels will be labeled independently and interpolated with neighboring pixels.

IV. Simulation Results

In order to evaluate the performance of our proposed method, sequences "Bus", "Coastguard", "Flower", "Football", "Hall", "Silent" and image "Lena" are used in the simulations. The resolution of "Lena" is 256×256 and others



(a) (b) (c) (d) (e)

Fig. 5. The subjective results of spatial concealment for sequence "Hall" with BLR=50%. (a) correct image; (b) corrupted image; (c) concealed with [3]; (d) concealed with [6]; (e) concealed with our proposed method.

are 352×288. The block size is 8×8. Block loss rate (BLRs) of 50% is tested in our simulations. In the experiments, we select methods [3] and [6] for performance comparison with our proposed method.

The PSNR is computed for each method and the comparison results are listed as shown in Table I, where Gain1 and Gain2 denote the PSNR improvements of our proposed method over [3] and [6] respectively. From this table, we can observe that our algorithm provides significant PSNR improvement over previous methods by up to 4.04 dB.

Fig. 4 and Fig. 5 show the subjective comparison results for different sequences with different BLRs. (a) is the correct image and (b) is the image corrupted by channel errors. (c) to (e) are concealed by [3], [6] and our proposed method respectively. From these figures, we may observe that the image quality is consistent with the PSNR measurement as shown in Table I. The subjective performance of our proposed algorithm is perceptually superior to others, which can be particularly demonstrated around the edges in the missing blocks.

V. Conclusions

This paper presents a novel spatial domain error concealment algorithm. Compared with other existing methods, our proposed method is able to detect the edges more accurately. In addition, the final pixel interpolation of our algorithm is based on the partition of the lost block, which means that only the pixels in the same region as the lost pixel will be used for interpolation. Experimental results show that the proposed method is able to improve the performance significantly compared with previous methods in both subjective and objective measurements.

References

[1] J. Park, D.-C. Park, I. Marks, R.J., and M. El-Sharkawi, "Recovery of image blocks using the method of alternating projections," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 461 –474, Apr. 2005.

[2] M. Kim, H. Lee, and S. Sull, "Spatial error concealment for h.264 using sequential directional interpolation," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1811–1818, Nov. 2008.

[3] N. El Beheiry, M. El Sharkawy, M. Lotfy, and S. Elnoubi, "An adaptive fast and efficient spatial error concealment technique for block-based video coding systems," in *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS '09)*, Aug. 2009, pp. 663–668.

[4] M. Chen, M. Wu, and Y. Zheng, "Classification-based spatial error concealment for images," in *Proceedings of International Conference on Image Processing, 2003 (ICIP'03)*, vol. 3, 2003, pp. 675–678.

[5] H. Gharavi and S. Gao, "Spatial interpolation algorithm for error concealment," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2008 (ICASSP'08)*, 2008, pp. 1153–1156.

[6] A. Sirikam and W. Kumwilaisak, "New spatial error concealment using dynamic texture estimation and geometric interpolation," in *2007 IEEE International Conference on Multimedia and Expo*, July 2007, pp. 120–123.

[7] W. Zhu, Y. Wang, and Q.-F. Zhu, "Second-order derivative-based smoothness measure for error concealment in DCT-based codecs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 6, pp. 713 –718, Oct. 1998.

[8] W. Zeng and B. Liu, "Geometric-structure-based error concealment with novel applications in block-based low-bit-rate coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 648–665, June 1999.

[9] Y. Wang, Q.-F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Transactions on Communications*, vol. 41, no. 10, pp. 1544–1551, Oct. 1993.

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

[11] W. Kwok and H. Sun, "Multi-directional interpolation for spatial error concealment," *IEEE Transactions on Consumer Electronics*, vol. 39, no. 3, pp. 455–460, June 1993.