Spatial Error Concealment: A Novel Exemplar-Based Approach Using Segmentation

Mani Ranjbar and Shohreh Kasaei[1]

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

Abstract

In this paper, the problem of spatial error concealment for real-time applications is addressed. The proposed method can be categorized in exemplar-based error concealment approaches. In this category, a patch of corrupted pixels are replaced by another patch of the image that contains correct pixels. For splitting the erroneous block to different patches, a novel context-dependent exemplar-based algorithm based on a previously proposed segmentation method is proposed. The capability of the proposed method for concealment in diverse image regions is depicted. Our detailed conducted experiments show that the proposed method outperforms the state-of-the-art spatial error concealment methods in terms of output quality.

Key words

Spatial Error Concealment, Image Completion, Inpainting, Exemplar-Based Error Concealment, Real-Time Concealment, Context-Dependent Splitting, Video Compression.

1. Introduction

Over the last decades, great improvements have been made in video communication by a growing demand for efficient video compression. Bandwidth limitations and the huge amount of data in raw image sequences were the very first problems in this type of communication. Therefore, a wide range of video compression methods was proposed for addressing this problem. Among different video compression methods, block-based

approaches have been exploited extensively in video coding standards. In block-based approaches, a video frame is first split to nonoverlapping blocks and then processed block-wise. Compressed blocks are then transmitted to the receiver through an error prone network (Internet, wireless, etc.) and decompressed to reconstruct the original data. During data transmission through the network, the transmitted bitstream might be affected by different types of error. Two possible error control methods have been studied in the literature for recovering erroneous blocks in the receiver.

The first error control method is based on informing the transmitter and requesting for retransmission of the entire (or a part) of the erroneous frame [1]. This method suffers from the imposed retransmission delay that might be impractical in real-time applications. The second method performs by recovering the corrupted block using correctly received data. In the latter method, two general approaches have been proposed in the literature; error resilient and error concealment. Error resilient methods add some redundant information to the original data in the transmitter side and employ them in the receive side when error occurs [2-5]. For better resiliency, more added redundant data is needed that is in contrast with compression purposes. Error concealment methods, on the other hand, employ the inherent spatial and temporal redundancy of the received data to conceal the error [6, 7]. In contrast with retransmission-based and resilient-based methods, these methods add no delay and no redundant data. However, they might suffer from higher computational complexity at the receiver side and less reconstructed image quality.

In this paper, a novel spatial error concealment method is presented that outperforms the state-of-the-art approaches in terms of reconstructed image quality and performs almost in real-time. The rest of this paper is organized as follows. In Section 2, previous works on error concealment are shortly introduced. The focus of this section is on spatial methods. The proposed method is presented in detail in Section 3. The conducted experimental results in diverse situations and in comparison with other existing methods are presented in Section 4. The time complexity of the method is also presented in this section. In section 5, conclusion and future work are drawn.

2. Previous Work

Numerous error concealment techniques have been reported in the literature. These can be categorized into two main groups: spatial and spatiotemporal. Spatial error concealment methods employ the correctly received data of the current frame to conceal the erroneous data. Spatiotemporal methods, on the other hand, exploit the data of neighboring frames in addition to the data of the current frame to conceal the corrupted data. For instance, in [8] a global motion is assumed for the object which is compensated using previous time instances and refined by a local motion. The method proposed in [9] utilizes a *self*

organizing map (SOM) to predict the corrupted *motion vectors* (MVs). A greedy suboptimal approach accompany with Kalman filtering technique is employed in [10] to conceal the corrupted motion vectors. In [24], at first a boundary matching approach is exploited to find the best MV candidate in the reference frame, and then a *partial differential equation* (PDE)-based method is employed to reduce the blocking artifacts caused by directly copying the corresponding *macro block* (MB) from the reference frame. In contrast with the proposed approach that separates the erroneous block according to the image context, the whole MB is concealed at once in this method.

Spatial error concealment methods can be categorized in four main types: interpolation-based, stochastic-based, tensor voting-based and exemplar-based.

Interpolation-based methods conceal each corrupted pixel by interpolation; using the correctly received neighboring pixels. In [11] and [12] for instance, directional interpolation methods are proposed that utilize edge directions for interpolation. The method proposed in [12] outperforms the method in [11] by taking into account some complex situations such as edge intersections inside the erroneous block. The weakness of interpolation-based methods is that although they perform very fast, texture preservation is not their intrinsic characteristic and should be imposed separately. In [25] an interpolation-based method is proposed to conceal the alpha planes in an object-based image or video. This method employs B-Spline curves to conceal the boundaries of the objects. The object of this method is similar to the object of the proposed method in interablock boundary recovery section. However, the proposed method computes the contours of different region and fills the erroneous regions with appropriate textures.

Stochastic-based methods consider each pixel of the frame as a random variable and employ the Bayes rule to approximate the best pixel values of the corrupted block. For example, methods proposed in [13] and [14] utilize the *maximum a posteriori* (MAP) to estimate the best approximation for the corrupted pixels. In [15], the main assumption is that the input image is a stationary Gaussian random process and satisfies the N^{th} order Markov property. Considering these assumptions, the MAP estimation boils down to a linear minimum mean-squared error estimation problem. The main disadvantage of these methods is their stationary assumption for input images.

In tensor voting approaches, a tensor vector (containing the intensity and gradient information of each pixel) is calculated for each correctly decoded pixel. Using a voting scheme, tensor vectors of the corrupted pixel are approximated and translated to the intensity values [16]. The main drawback of tensor voting approaches is their high computational complexity that is not affordable in real-time applications.

Error concealment task in exemplar-based methods is performed by cloning the best patches from the correctly received region of the image to the corrupted block. As an example, in [17] each block is considered as one patch and is copied from the most similar patch of the correctly received region. In this method, the similarity is measured by the sum-squared difference of the outer boundaries. In [18], each block is split by a predefined structure and then each subblock is concealed by the method proposed in [17]. In this approach, the subblocks near the edge are concealed first. While texture preservation is an intrinsic characteristic of exemplar-based approaches, obtaining a proper region boundary is not guaranteed.

In this paper, a novel exemplar-based approach is proposed. To achieve an appropriate region boundary, an efficient segmentation algorithm is employed [19]. High quality concealed images accompanied by fast concealment performance introduce the proposed method as a suitable option for intra coded frames (I frames) concealment in advanced video coders. Since the number of transmitted I frames per second in real scenarios such as video conferencing, is rarely more than 2, the optimized implementation of the proposed method can conceal the intra coded frames in real-time.

3. Spatial Error Concealment

a. Overview

As illustrated in Fig. 1, the proposed algorithm is consisted of four main steps. In the first step, a patch around the corrupted block is cropped from the input image. The cropped patch is segmented in the second step. Segment boundaries are recovered in the third step. By recovering the segment boundaries, erroneous block is split to some subregions. In the fourth step, each subregion is inpainted, independently.

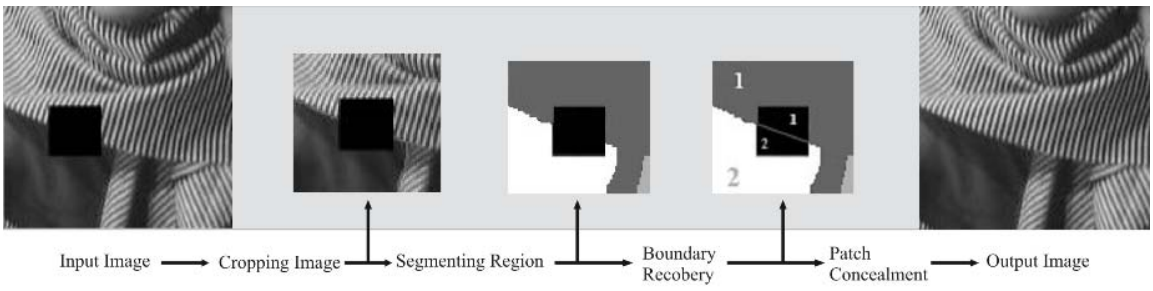


Figure 1. Overall block diagram of the proposed method.

b. Region Cropping

The main idea of the proposed method is based on the local similarity property in each image. According to this property, and considering each image as a Markov process with a limited order, one can take into account only a local window around the processing region to conceal that block. Moreover, on one hand, the high computational complexity of segmentation and inpainting algorithms imposes to process in a local window. On the other hand, using larger local windows leads to more image information and thus more chance for better concealment quality. Consequently, to achieve the best tradeoff between complexity and performance, in this work an adjustable rectangular patch is cropped around the erroneous block. The location of top left corner of the patch is obtained by

$$TL_x = \max(1, E_x - (\frac{P_w - E_w}{2})) \quad (1)$$

$$TL_y = \max(1, E_y - (\frac{P_h - E_h}{2})) \quad (2)$$

in which E_x, E_y, E_w, and E_h denote the horizontal and vertical location of the top left corner of the erroneous block and the width and height of the erroneous block, respectively. Also, P_w and P_h stand for the patch width and height, respectively. In a specific application, the tradeoff between quality and complexity can be controlled by adjusting P_w and P_h.

c. Segmentation

The objective of this process is to segment the input image into connected homogeneous regions that contain similar pixels. The similarity measure of segmentation process is discussed later in this section.

In order to segment the input image, a graph-based segmentation method [19] is adopted which is capable of preserving details in low variable image regions while ignoring details in high variable regions. The algorithm includes the following steps.

Each image pixel is considered as a region if it corresponds to a node (v in V) in overall image graph $G(V, E)$.

Neighboring pixels are connected by undirected edges (e in E). For each edge, a weight coefficient is computed according to dissimilarities among pixels.

Similar regions (A and B) are merged together to form a larger region if the following condition is held

$$Dif(A,B) \leq MInt(A,B) \tag{3}$$

where

$$Dif(A,B) = \min_{v_i \in A, v_j \in B, (v_i, v_j) \in E} w((v_i, v_j)) \tag{4}$$

here, E is the graph edge set and for each node, $w((v_i,v_j))$ is the weight between vertices v_i and v_j. Moreover

$$MInt(A,B) = \min(Int(A) + \tau(A), Int(B) + \tau(B)) \tag{5}$$

$$Int(A) = \max_{e \in MST(A,E)} w(e) \tag{6}$$

$$\tau(A) = \frac{k}{|A|} \tag{7}$$

where *MST* represents the *minimum spanning tree* of graph $G(V, E)$. Also, the value of 'k' defines the sensitivity level of segmentation algorithm (larger 'k' is preferred for larger regions). Note, however, that 'k' is not the minimum segment size. In fact, smaller segments are allowed when there is a sufficiently large difference among neighboring regions.

Fig. 2 illustrates the capability of the segmentation method in differentiating complex textures from smooth textures while maintaining similar textures at each segment.

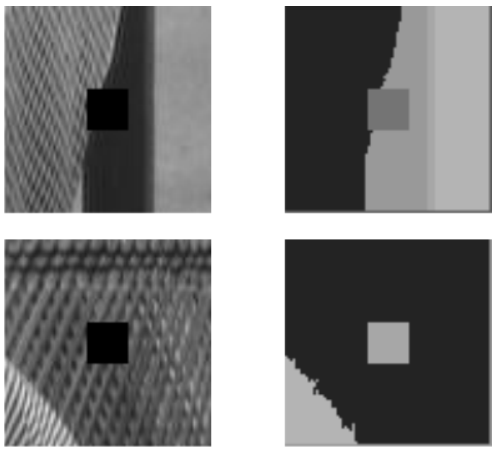


Figure 2. Segmentation results in two complex scenes:

(a) input patch, (b) segmented patch.

d. Intrablock Boundary Recovery

The purpose of this step is recovering the boundaries of segmented regions inside the corrupted block. For achieving this purpose, the following steps are performed.

1. Find the intersection between boundary points and erroneous block borders.
2. Approximate those boundaries that have intersection with corrupted block by a line.
3. Create a feature vector for each point found in step 1.
4. Find correspondence boundaries.
5. Calculate the parameters of approximated boundary lines inside the corrupted block.

After finding the intersection points $p_j=(x_j,y_j)$, $j=1..k$, where k is the number of intersection points, corresponding boundary points are gathered and a least squared line fitting method is utilized to calculate the parameters of the best fitted line to these points. The slope and y-intercept of the best fitted line are obtained by

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \times \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \tag{8}$$

$$b = \frac{\sum_{i=1}^{n} y_i - a\sum_{i=1}^{n} x_i}{n} \tag{9}$$

where, x_i and y_i are the x and y coordinates of the ith boundary point and n is the number of boundary points. After this step, each founded point in step 1 has a corresponding line. After employing the approximated lines, feature vectors Vj are created according to the intersection points p_j, by

$$V_j = \{(x_j, y_j), \tan^{-1}(a_j), C_{1_j}, C_{2_j}\}, \quad j \in \{1,...,k\} \tag{10}$$

As shown in (10), each vector consists of four elements: intersection point location (x_i , y_i), slope of the best fitted line $\tan^{-1}(a_j)$, class number of the region in one side of the intersection point C_{1j}, and class number of the region in the other side of the intersection point C_{2j},. The process of creating a sample feature vector is depicted in Fig. 3.

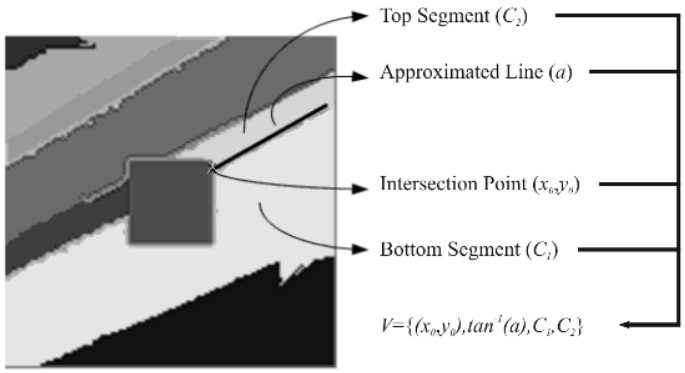


Figure 3. Proposed feature vector formation procedure.

For finding the entry and exit points of the corrupted boundary regions, every pair of intersection points p_i and p_j is examined through the following conditions. Each pair that satisfies all conditions is returned as a recovered region boundary.

The line between p_i and p_j should intersect with the corrupted block.

Feature vectors V_i and V_j should have at least one common class number.

The angle between the approximated line in p_i and the approximated line in p_j should be obtuse.

Above mentioned conditions are checked through (11) which returns 1 when points p_i and p_j are a correct pair, and 0 otherwise.

$$L(p_i,p_j)=\begin{cases} 1 & \begin{array}{ll} i\neq j \quad i,j\in(1,..,k) & \wedge \\ x_i\neq x_j \vee y_i\neq y_j & \wedge \\ \{C_{1_i},C_{2_i}\}\cap\{C_{1_j},C_{2_j}\}\neq\varnothing & \wedge \\ 90<\left|\tan^{-1}(a_i)-\tan^{-1}(a_j)\right|\times\dfrac{180}{\pi}<270 & \end{array} \\ 0 & \textit{Otherwise} \end{cases} \tag{11}$$

e. Segment Mask Creation

After recovering the boundaries of segments, the erroneous block is split into some patches (see Fig. 4).

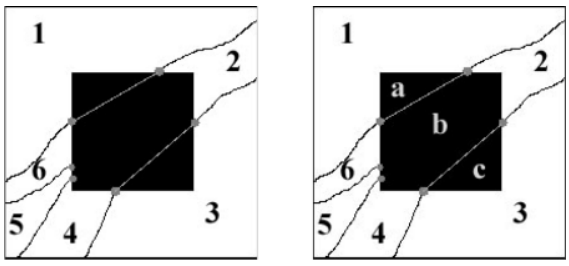


Figure 4. Splitting an erroneous block using recovered boundaries:

(a) boundary recovering result, (b) subregions creation result.

For each match, a particular mask is created and utilized in the next step for patch recovery. Before creating a mask for each patch, pixels of the corrupted block that belong to a specific patch should be distinguished. Therefore, an *ID* value for each pixel inside the corrupted block is calculated using

$$ID(x,y) = \sum_{i=0}^{n-1} 2^i \times U_{-1}(f_i(x) - y) \tag{12}$$

where

$$f_i(x) = \frac{y_{1_i} - y_{2_i}}{x_{1_i} - x_{2_i}}(x - x_{1_i}) + y_{1_i} \tag{13}$$

Here, x_1 and y_1 show the start point, and x_2 and y_2 show the end point of each recovered region boundary. In (12), Bit j of $ID(x,y)$ is equal to 1 if the pixel is above the j line, and 0 otherwise. Therefore, the *ID* value of each pixel determines the location of that pixel corresponding to these lines. Consequently, pixels with the same *ID* value belong to the same patch.

Each mask is then created through the following steps.

1. Create the smallest rectangle that encloses the patch.
2. Set pixels of the rectangle that correspond to the patch pixels to 1, and the rest to 0.
3. Extend the rectangle by d pixels in the sides that have boundary pixels equal to 1.

4. Set the extended pixels to -1 if their corresponding boundary pixel is 1, and 0 otherwise.

To better illustrate this process, the created mask for patch 'a' of Fig. 4-b is shown in Fig. 5-b. For easier manipulation, in the created mask, pixels with value 1 are the corrupted pixels, pixels with value -1 are correct pixels around the erroneous block that are employed for finding the best replacement patch, and pixels with value 0 are added to created a rectangular mask.

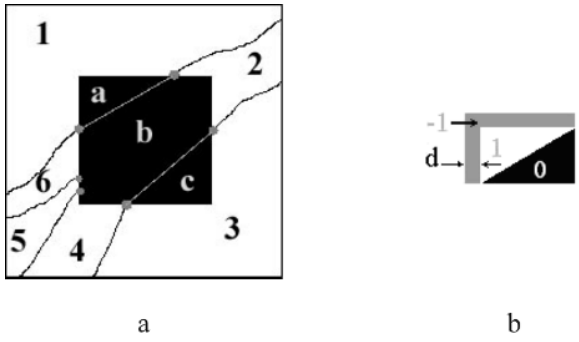


Figure 5. A sample mask creation. (a) Boundary recovered image containing three different patches; *a, b,* and *c.* (b) Created mask for patch *a*.

f. Patch Concealment

The created mask for each patch is then utilized to conceal that patch. First, a copy of this mask- called *sliding mask* (SM)- is created. The original mask which is located on top of the corrupted patch is called *reference mask* (RM). In the following, RM_1 denotes the location of those pixels of RM that have the value of 1 in the cropped image. Furthermore, $I_c(RM_1)$ is the set of pixel values in the cropped image for which the RM mask has the value of 1. Using these definitions, the best recovering patch is obtained through the following algorithm (see Fig. 6).

1. Find $\mu_{RM,}$ using

$$\mu_{RM} = \frac{1}{n}\sum I_c(RM_{-1}) \tag{14}$$

2. For every possible location of SM,
    a. If SM∩RM=Ø,
        i. Find μ_{SM}, using

$$\mu_{SM(x,y)} = \frac{1}{n}\sum I_c(SM_{-1}(x,y)) \tag{15}$$

ii. Find the matching cost, using

$$(bestX, bestY) = \arg\min_{x,y}\{w(x,y)\times\left|I_c(RM_{-1}) - \mu_{RM} - I_c(SM_{-1}\right. \tag{16}$$

iii. Update the best matching cost by

$$w(x,y) = \left(\frac{1}{\sqrt{E_w^{\,2} + E_h^{\,2}}}\sqrt{(x-\frac{P_w}{2})^2 + (y-\frac{P_h}{2})^2} + 1\right)^{-1} \tag{17}$$

where E_w and E_h are the width and height of the corrupted block and P_w and P_h are the width and height of the cropped patch. Weighting factor w, ensures the preference for closer matches.

3. Replace $I_c(RM_1)$ with $I_c(SM_1(\text{best}X,\text{best}Y))$ after adjusting the mean value by

$$I_c(RM_1) = I_c(SM_1(bestX, bestY)) - \mu_{SM(bestX,bestY)} + \mu_{RM} \tag{18}$$

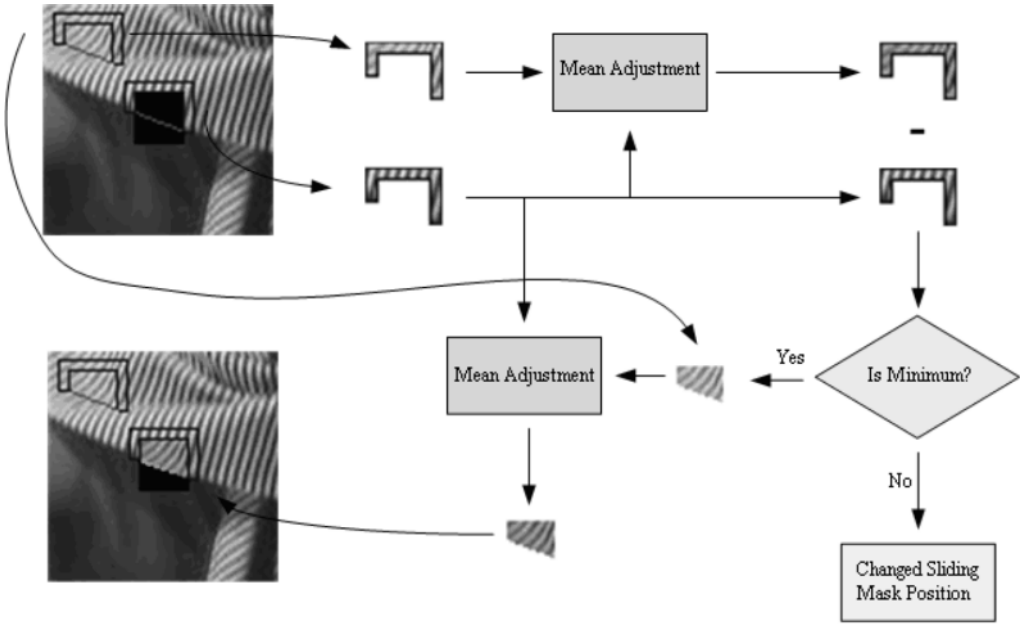
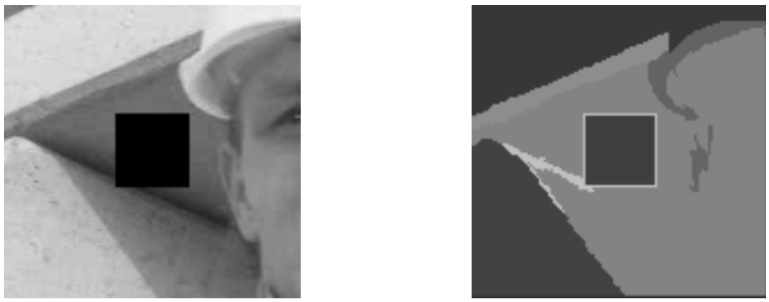


Figure 6. Block diagram of the proposed patch concealment process.

It is worth to mention that in some situations, the corrupted patch might be larger than the existing corrected data around the erroneous block and because the whole patch is concealed at once, poor concealment result might be obtained (see Fig. 7).



a b

Figure 7. An example of a situation in which the erroneous block is not split into smaller patches and thus no proper match is available for the corrupted block in the cropped image: (a) cropped image, (b) segmentation result.

To resolve this problem (when the cost of replacement for the best match is more than a predefined threshold) an alternative patch concealment algorithm is employed. This is explained next.

g. Alternative Patch Concealment

Previous step finds the best match for each patch and also returns a nonsimilarity measure (defined in the righthand side of (16)). If the nonsimilarity value divided by the number of compared pixels is more than a predefined threshold, the alternative patch concealment algorithm is employed. The method with less nonsimilarity value is selected for concealing the corrupted patch. The following steps describe the alternative method.

1. Split the erroneous block into four similar rectangular subblocks; by splitting the erroneous block in vertical and horizontal directions.
2. Calculate the gradient magnitude of outer boundary pixels of corrupted block.
3. First conceal the subblock with the maximum sum of gradient magnitudes on its outer boundary.

4. Iterate Step 3 until the erroneous block is concealed. (Note that the concealed subblock in the previous step is also utilized for concealing the current subblock.)

For all four subblocks, the maximum nonsimilarity value is considered as the nonsimilarity value of the alternative method. This value is compared to the nonsimilarity value obtained from the previous patch concealment method. The method with less nonsimilarity value is chosen for patch concealment.

h. Boundary Smoothing

In the patch concealment step, the only parameter that is utilized to adjust the intensity of the best match with the outer boundary pixels is the mean value. Therefore, it is not guaranteed to have a smooth change in erroneous block boundaries. The goal of this section is to reduce such discontinuities by smoothing along block boundaries. To achieve this goal, the inner and outer boundary pixels are updated by

$$I(P_2) = \frac{I(P_1) + I(P_3)}{2} \tag{19}$$

$$I(P_3) = \frac{I(P_2) + I(P_4)}{2} \tag{20}$$

Pixels P_1 to P_4 are illustrated in Fig. 8.

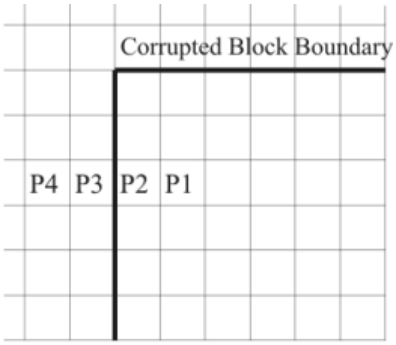


Figure 8. An example of inner and outer boundary points used in Eqs. (19) and (20).

4. Experimental Results

In this section, our conducted experimental results are presented. For obtaining the results, the proposed algorithm was implemented in C++ using the OpenCV functions on a computer with an Intel Core 2, 6300 @ 1.86 GHz CPU and 2GB RAM.

In deducted experimental result only the isolated block loss situation is considered, because using the *flexible macroblock ordering* (FMO) of H.264 codec and the pattern proposed in [26], every burst error in a slice results in an isolated error pattern in the reconstructed image.

The results are classified into two sets. In the first set, the capability of the proposed method in concealing large erroneous blocks is shown. In the second set, in order to be comparable with other reported methods, the results of the proposed method are depicted in a situation identical to that of other reported methods and the obtained results are compared using the PSNR measure.

Unlike most of previously reported methods that use only 8×8 or 16×16 corrupted block sizes, the results of the proposed method (in some manually selected areas) for corrupted block sizes 32×32 are shown in Fig. 9 and Fig. 10. The selected areas are sorted from simple to complex situations. As illustrated in these figures, the proposed method has been able to properly conceal a region containing a flat region, sharp edges, complex texture, and a combination of them.

For performance comparison purposes, the scenario described in [15] is exploited. Fig. 11 and Fig. 12 demonstrate the result of the proposed method over Lena, Zelda, Pepper, and Baboon images. In Table 1, the results of some previous methods (obtained from [5]) accompanied with the results of the proposed method are listed using the PSNR measure. Note that the PSNR value is calculated only over the corrupted blocks (and not over the entire image). The results listed in Table 1 clearly show that the proposed method outperforms the previous approaches by at least 0.3 dB.

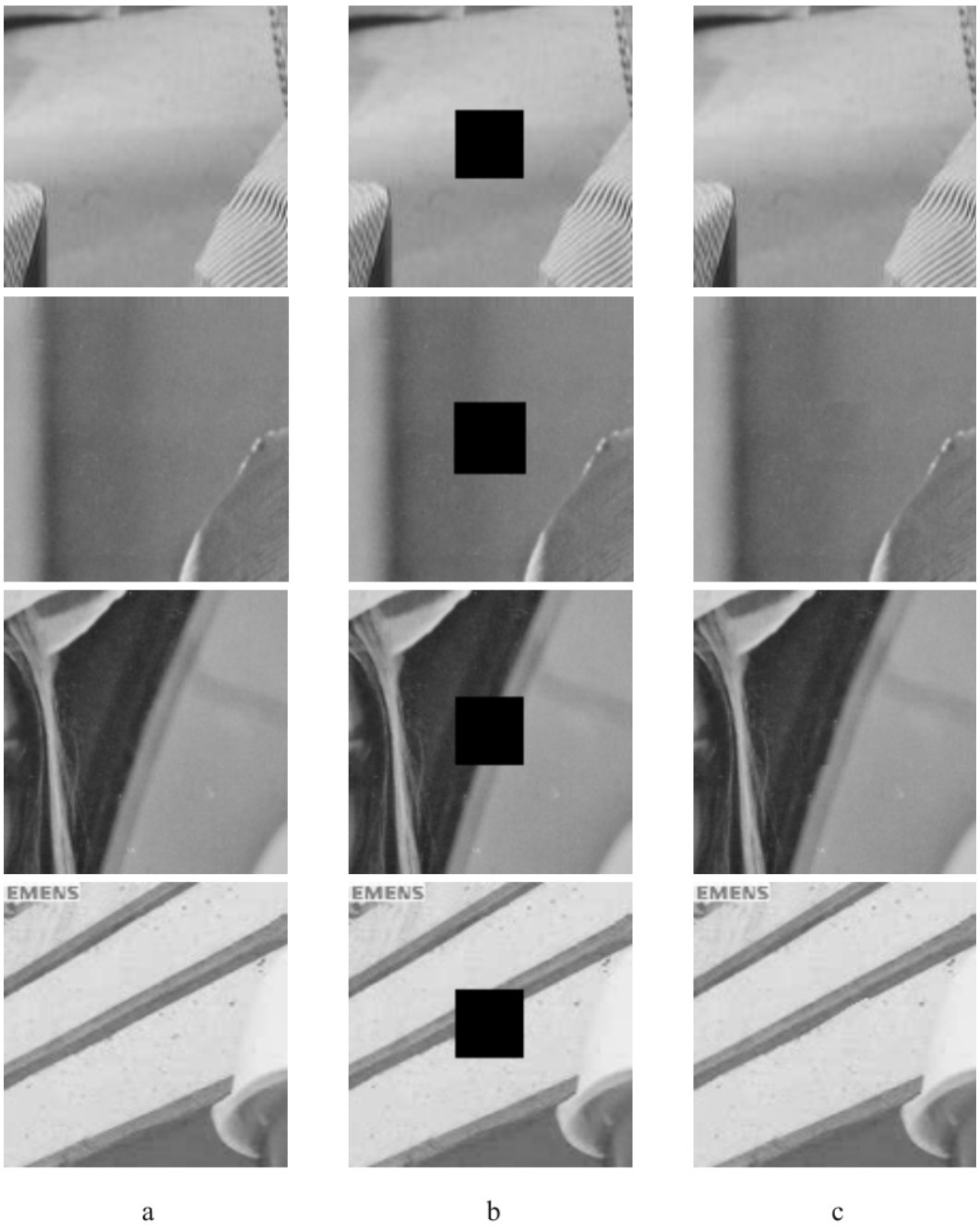


Figure 9. Results of the proposed concealment method in some selected regions (sorted from simple to complex): (a) original image, (b) corrupted image, (c) concealed image.

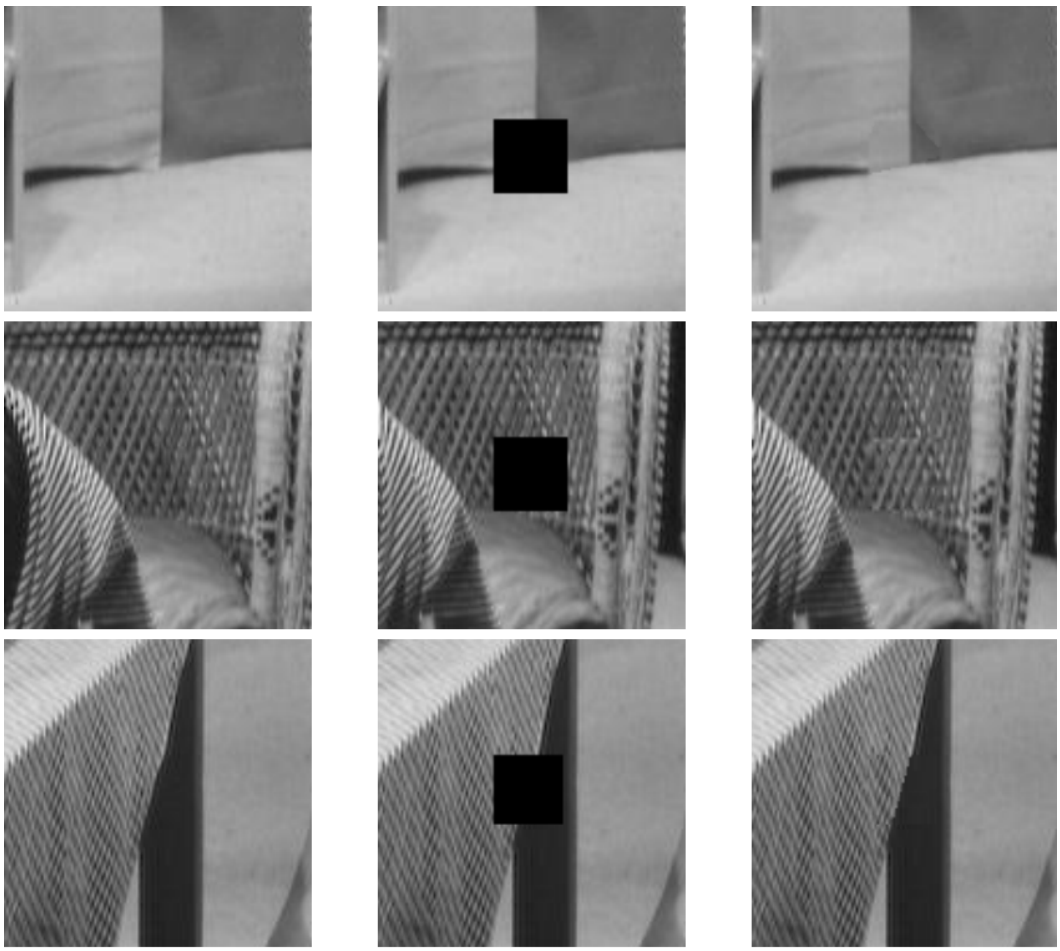
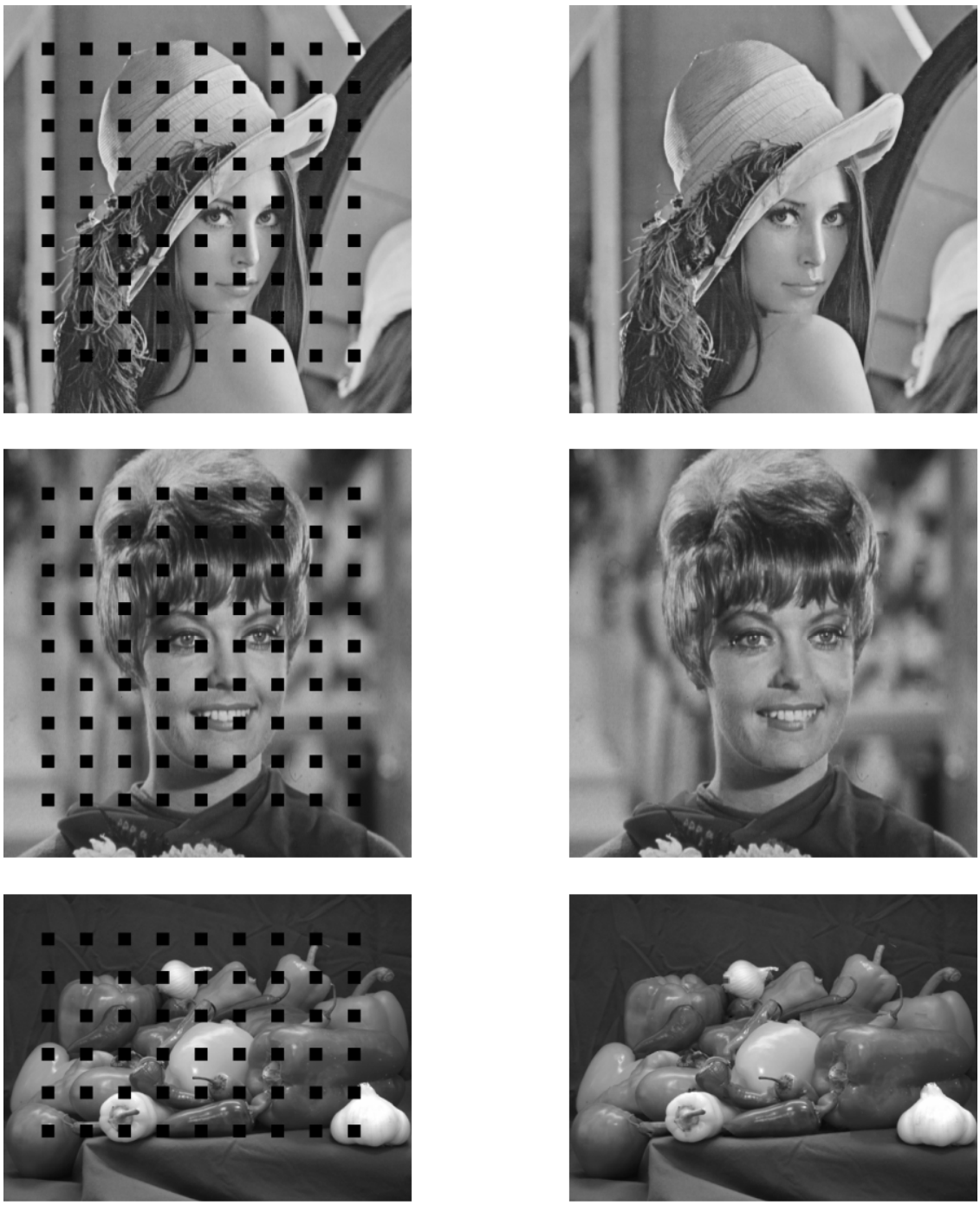
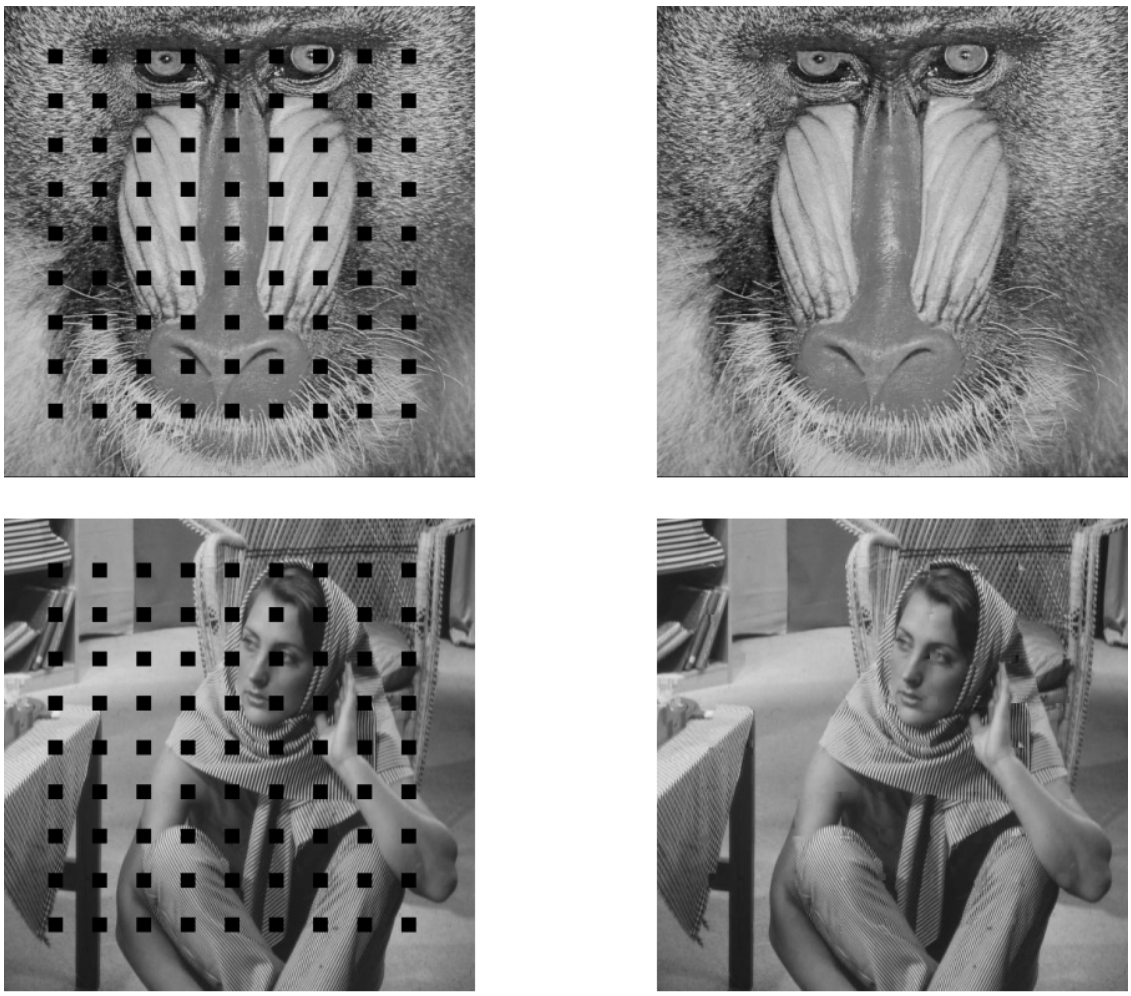


Figure 10. Results of the proposed concealment method in various scenarios (sorted from simple to complex): (a) original image, (b) corrupted image, (c) concealed image.



a b

Figure 11. Results of the proposed method for concealing Lena, Zelda and Peppers images: (a) corrupted image, (b) concealed image.



a b

Figure 12. Results of the proposed method for concealing Baboon and Barbara image: (a) corrupted image, (b) concealed image.

Table 1. PSNR comparison between state-of-the-art and the proposed method. (Performance of other methods is quoted directly from [5].)

Image	[20]	[21]	[22]	[23]	[15]	Proposed Method
Lena	23.93	23.99	24.41	24.96	26.46	26.91
Peppers	22.19	23.69	24.06	24.48	27.25	27.56
Zelda	26.35	27.13	26.40	27.36	28.23	28.63
Baboon	17.46	18.98	19.02	17.42	NA	20.12
Average	22.48	23.44	23.47	23.55	NA	25.80

As mentioned above, the parameters P_w and P_h can be utilized to control the tradeoff between time complexity and output quality. In Fig. 13, the effect of these parameters is demonstrated by capturing the running time and the output quality (PSNR) for concealing corrupted Lena image shown in Fig. 11. Also, in Table 2, the running time for different steps of the algorithm is reported.

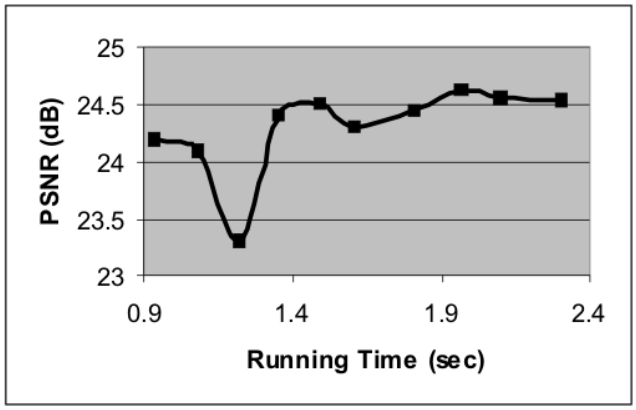


Figure 13. Output quality (PSNR) versus search elapsed time (left most point $P_w = P_h=62$, right most point $P_w=P_h=80$).

Table 2. Average elapsed times of different parts of the proposed method.

Method	Elapsed Time(ms)
Segmentation	507
Boundary Recovery	0.2
Patch Concealment	421
Boundary Smoothing	0.1

5. Conclusion

In this paper, a novel error concealment method that utilizes spatial information is proposed. Unlike interpolation-based, statistical-based, and tensor voting-based algorithms, texture preservation is the intrinsic characteristic of the proposed method. The novelty of this approach among previously proposed methods is in its context-dependent patch creating inside erroneous blocks. The performance of the proposed method is evaluated through a large number of experimental results in different scenes. Our proposed method achieves an improvement of at least 0.3dB in average in comparison with the state-of-the-art methods with a loss rate of about 10%. The computational cost can be further reduced using a more efficient implementation of the codes.

Acknowledgement

This work was in part supported by a grant from Iran Telecommunication Research Center (ITRC).

Reference

[1] B. Girod, and N. Farber. Feedback-based error control for mobile video transmission. In: Proceedings of the IEEE 1999;87(10):1707-17.

[2] J. Hagenauer, T. Stockhammer. Channel coding aspects for wireless multimedia. In: Proceedings of the IEEE 1999;87(10):1764-14.

[3] S. B. Wicker. Error control systems for digital communication and storage. NJ: Prentice Hall, 1995.

[4] V. K. Goyal. Multiple description coding: compression meets the network. IEEE Signal Process. Mag. 2001;18(5):74-20.

[5] C. B. Adsumilli, M. C. Q. Farias, S. K. Mitra, M. Carli. A Robust Error Concealment Technique Using Data Hiding for Image and Video Transmission over Lossy Channels. IEEE Trans. on Circuits and Systems for Video Technology 2005;15(11).

[6] B. W. Wah, X. Su, and D. Lin. A survey of error-concealment schemes for real-time audio and video transmission over the internet. In: Proceedings of the Int. Symp. Multimedia Software Engineering 2000, December 2000, p. 17-8.

[7] P. Cuenca, L. Orozco-Barbosa, A. Garrido, F. Quiles, T. Olivares. A survey of error concealment schemes for MPEG-2 video communications over ATM network. In: Proceedings of the IEEE Can. Conf. Elect. and Comp. Eng. , vol. 1, May 1997; p. 25-4.

[8] L. D. Soares, F. Pereira. Temporal Shape Error Concealment by Global Motion Compensation with Local Refinement. IEEE Trans. on Image Processing 2006;15(6).

[9] Y. L. Huang, H. Y. Lien. Temporal Error Concealment for MPEG Coded Video Using a Self-Organized Map. IEEE Trans. on Consumer Electronics 2006;52(2).

[10] W. N. Lie, Z. W. Gao. Video Error Concealment by Integrating Greedy Suboptimization and Kalman Filtering Techniques. IEEE Trans. on Circuits and Systems for Video Technology 2006;16(8).

[11] Wenjun Zeng, Bede liu. Geometric-structure-based Error Concealment with Novel Applications in Block-based Low Bit Rate Coding. IEEE Trans. on Circuits and Systems for Video Technology, June 1999.

[12] Wei-Ying Kung, Chang-Su Kim, 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;16(7).

[13] P. Salama, N. B. Shroff, and E. J. Delp. A Bayesian Approach to Error Concealment in Encoded Video Streams. In: Proceedings of the International Conference on Image Processing, September 1996.

[14] Zhiliang XU, Shengli Xie. An Efficient Spatial Error Concealment for Video Transmission. In: Proceedings of the IEEE on Communications, Circuits and Systems 2005.

[15] Xin Li, Michael T. Orchard. Novel Sequential Error-Concealment Techniques Using Orientation Adaptive Interpolation. IEEE Trans. on Circuits and Systems for Video Technology 2002;12(10).

[16] Jiaya Jia, Chi-Keung Tang. Inference of Segmented Color and Texture Description by Tensor Voting. IEEE Trans. on Pattern Analysis and Machine Intelligence 2004;26(6).

[17] D. Zhang, Zh. Wang. Image Information Restoration Based on Long-Range Correlation. IEEE Trans. on Circuits and System 2002;12(5).

[18] Antonio Criminisi, Patrick Pérez, Kentaro Toyama. Region Filling and Object Removal by Exemplar-Based Image Inpainting. IEEE Trans. on Image Processing 2004;13(9).

[19] P. F. Felzenszwalb, D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. International Journal of Computer Vision 2004, vol. 59, September 2004, p. 167-15.

[20] H. Sun and W. Kwok. Concealment of damaged block transform coded images using projection onto convex sets. IEEE Trans. on Image Processing 1995;4(4):470-8.

[21] P. Salama, N. B. Shroff, E. J. Coyl, E. J. Delp. Error Concealment Techniques for encoded video streams. In: Proceedings of Int. Conf. on Image Processing, vol. 1, 1995, pp. 9-4.

[22] Y. Wang, Q. Zhu. Signal Loss Recovery in DCT-based Image and Video Codecs. In: Proceedings of SPIE Conf. on Visual Communication and Image Processing, vol. 1605, 1991, p. 667-12.

[23] J. W. Park, S. U. Lee. Recovery of Corrupted Data Based on the NURBS Interpolation. IEEE Trans. on Circuits and Systems for Video Technology 1999;9(10):1003-6.

[24] Y. Chen, Y. Hu, O. C. Au, H. Li, C. W. Chen. Video Error Concealment Using Spatio-Temporal Boundary Matching and Partial Differntail Equation. IEEE Trans. on Multimedia 2008;10(1):2-14.

[25] L. D. Soares, F. Pereira. Spatial Shape Error Concealment for Object-Based Image and Video Coding. IEEE Trans. on Image Processing 2004;13(4):586-14.

[26] Y. Dhondt, S. Mys, S. D. Zutter, R. V. D. Walle. An Alternative Scattered Patern for Flexible Macroblock Ordering in H.264/AVC. IEEE 8th International Workshop on Image Analysis for Multimedia Interactive Services 2007.

Mani Ranjbar was born in Semnan, Iran, in 1982. He received the B.Sc. degree in Computer Engineering from Sharif University of Technology, Tehran, Iran, in 2005. He joined the Image Processing Laboratory (IPL) at the Department of Computer Engineering, Sharif University of Technology in 2005 and received the M.Sc. degree in that department in 2007. He is currently working toward the Ph.D. degree in the Vision and Media Laboratory (VML), School of Computing Science, Simon Fraser University, Burnaby, British Columbia, Canada. His research interests include image processing, computer vision, and robotic vision.

Shohreh Kasaei received the B.Sc. degree from Department of Electronics, Faculty of Electrical and Computer Engineering, Isfahan University of Technology (IUT), Iran, in 1986. She worked as a Research Assistant in Amirkabir University of Technology (AUT) for three years. She then received the M.Sc. degree from Graduate School of Engineering, Department of Electrical and Electronic Engineering, University of the Ryukyus, Japan, in 1994, and the Ph.D. degree at Signal Processing Research Centre (SPRC), School of Electrical and Electronic Systems Engineering (EESE), Queensland University of Technology (QUT), Australia, in 1998. She was awarded as the best graduate student in engineering faculties of University of the Ryukyus, in 1994, the best Ph.D. students studied in overseas by the ministry of Science, Research, and Technology of Iran, in 1998, and as a distinguished researcher of Sharif University of Technology (SUT), in 2002, where she is currently an associate professor. She is the director of Image Processing Laboratory (IPL) at SUT. Her research interests are in image and video processing with primary emphasis on object–based video coding, video surveillance, video restoration, video resilient and concealment, video mosaicing, color image processing, hyperspectral change detection, motion estimation and active object tracking, fingerprint authentication/identification, image watermarking, content–based image retrieval, and virtual studios.