Spatial Error Concealment for H.264 Using Sequential Directional Interpolation

Myounghoon Kim, Hoonjae Lee, and Sanghoon Sull

Abstract — *Error concealment at the decoder restores erroneous macroblocks (MBs) caused by channel errors. In this paper, we propose a novel spatial error concealment algorithm based on prediction modes of intra-blocks which are included in a H.264-coded stream and highly correlated to the direction of local edge within the block. The key contribution is to sequentially interpolate each pixel in a lost MB by utilizing edge directions and strengths efficiently estimated from the neighboring blocks, preserving local edge continuity for more visually acceptable images. The proposed scheme is simple to implement and more reliably recover high-detailed content in corrupted MBs. The experimental results shows the proposed method achieves reduction in speed by 14%~39% as compared to existing method, and outperforms them in PSNR by 0.5~1dB as well as in subjective visual evaluation*[1].

Index Terms — Spatial error concealment, Intra prediction modes, Sequential Directional interpolation, Block-loss recovery, H.264.

I. INTRODUCTION

For video transmission over bandwidth-limited networks, digital video compression standards such as H.264/AVC [1] use a combination of various advanced features to improve the compression. However, compressed video streams are vulnerable to transmission errors due to both of packet losses and delays that are almost inevitable in video transmission over wireless channels as well as the Internet. Therefore, most video encoders adopt various methods to achieve error resilience for transmission over noisy channels so as to avoid or reduce the possible visual distortion.

There have been numerous studies on error resilient compression and decompression methods in an attempt to achieve robust video transmission. One approach [2] is to use a feedback channel to request retransmission or adjust the encoding modes according to channel conditions. Another approaches [3]-[5] are to insert redundant information into compressed video streams, which are efficient in stopping error propagation but may not be acceptable in some interactive applications due to extra delays and more computation power required.

A variety of spatial interpolation techniques for restoring an erroneous macroblock (MB) at the decoder side have been proposed, whose main objective is to estimate the lost information usually occurred due to transmission errors by utilizing the correctly received information. One of the typical spatial error concealment (SEC) methods [6], [7] is to interpolate each pixel in a lost MB from intact pixels in adjacent MBs. This linear interpolation scheme is a simple yet effective method for smooth images. More advanced approaches were proposed to adaptively recover the lost MB to improve quality of concealed frame. The algorithm [8] proposed the multi-directional interpolation (MDI) scheme which performs pixel domain interpolation along eight possible edge directions and considers the cases of both single edge and multiple edges. Kim *et al* proposed [9] a block loss recovery method based on fine directional interpolation (FDI), which extracts spatial direction vectors from the edge information of the neighboring images and then the spatial direction vectors are adaptively applied to interpolate lost pixels. These methods work well on smooth and regular blocks. However, a blurring effect is often seen in the concealed image if the lost MB contains low frequency components since the lost pixels are restored by an average of pixel values along multiple directions. Also they require a lot of computation on edge detection and reconstruction. Zeng and Liu [10] introduced the geometric-structure-based (GSB) SEC using a spatial directional interpolation scheme, which utilizes the local geometric information extracted from the surroundings. The two nearest surrounding layers of pixels of a missing block are converted into a binary pattern to reveal the local geometrical structure. The missing pixels are interpolated in a way to preserve the local geometrical structures. However, the directional interpolation could be sensitive due to the use of the angle information when the transition points are connected. Thus, the retrieved edges may not be faithful to the original ones. Park *et al.* [11] suggested the recovery of the image blocks using a method of alternating projections. Other approaches based on fuzzy logic reasoning [12], [13], are also widely used. They recover both the low and high frequency components by using a vague similarity relationship between a lost block and its neighboring blocks. However, the main drawback of these methods is that they have heavy computational load to be used for real-time applications due to its iterative procedures.

[1] This work was supported in part by the Korea University Grant.

Myounghoon Kim is with the Department of Electronics and Computer Engineering, Korea University, 1, 5-ka, Anam-dong, Sungbuk-ku, Seoul 136-701, Korea. (Telephone: +82-2-3290-3805, e-mail: mhkim@mpeg.korea.ac.kr).

Hoonjae Lee is with the Department of Electronics and Computer Engineering, Korea University, 1, 5-ka, Anam-dong, Sungbuk-ku, Seoul 136-701, Korea. (Telephone: +82-2-3290-3805, e-mail: hoonjae@mpeg.korea.ac.kr).

Sanghoon Sull is with the Department of Electronics and Computer Engineering, Korea University, 1, 5-ka, Anam-dong, Sungbuk-ku, Seoul 136-701, Korea. (Telephone: +82-2-3290-3805, e-mail: sull@mpeg.korea.ac.kr).

Corresponding Author: Sanghoon Sull

Contributed Paper
Manuscript received August 4, 2008

In this paper, we propose a low computational method of preserving edge continuity to obtain more visual acceptable reconstructed images: First, a set of the dominant edge directions (DED) and their strengths for the lost MB are efficiently estimated from the prediction modes and selected pixels of the neighboring blocks. Then, the pixels in the lost MB are interpolated sequentially along each DED in the set in the order of edge strength. In other words, the pixels along the most DED are first interpolated and then the pixels along the second most DED are interpolated by using the pixels in the neighboring blocks as well as the interpolated pixels in the lost MB. This process is repeated until the pixels along all DEDs are interpolated and then the remaining pixels not along the DEDs are interpolated.

The paper is organized as follows: In the next section, we briefly explain the intra prediction modes and SEC in the H.264. Section 3 introduces the proposed method. In order to show the effectiveness of the proposed method, the experimental results are provided in Section 4. Section 5 concludes the paper.

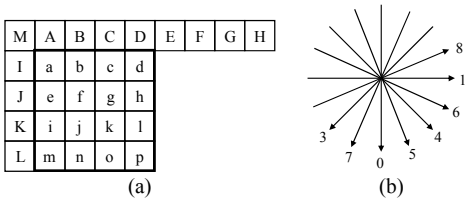

Fig. 1. Intra4×4 prediction coding is conducted for samples a-b of a block using samples A-Q. (b) Eight "prediction directions" for Intra4×4.

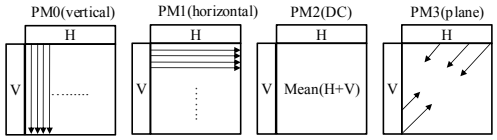

Fig. 2. Four prediction modes of intra16×16.

II. Prediction Modes and Spatial EC in H.264/AVC

This section presents an overview of the intra prediction modes and spatial error concealment in H.264/AVC.

A. Intra prediction modes in H.264

To encode a MB in intra-frame by prediction in spatial domain, the H.264 video coding standard defines three intra modes of different block size; intra4×4, intra8×8, and intra16×16. For luma samples, prediction block is formed for each 4×4 block, 8×8 block or 16×16 MB. For the prediction of a 4×4 block, there are 9 directional modes (*i.e.,* coding modes) consisting of a DC prediction (Mode 2) and 8 directional modes as shown in Fig. 1. The arrows in this figure indicate the direction of prediction in each mode. For prediction modes 3-8, the predicted samples are formed from a *weighted average* of the prediction samples A-L. The intra4×4 is well suited for coding of significant detail parts (*i.e.*, texture or high

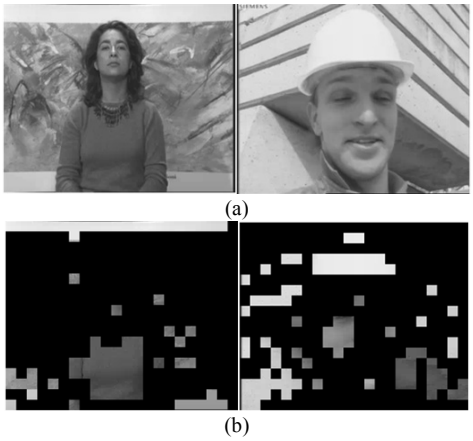

Fig. 3. (a) Frames in *Silent* and *Foreman* CIF sequences. (b) MBs coded by intra16×16 modes.

Fig. 4. An example of luma 4×4 block with neighboring pixels (A-Q)

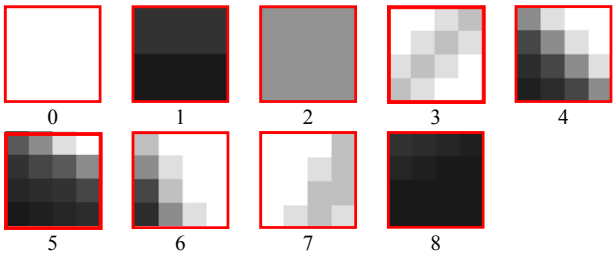

Fig. 5. Prediction blocks for luma 4×4 block in Fig. 4.

detail regions) in a frame. The intra8×8 modes are the same as those of 4×4 block, which has nine prediction modes and is defined in high profile. For regions with less spatial details (*i.e.*, flat regions), H.264 supports 16×16 intra-coding: in which one of four prediction modes is chosen for the prediction of the entire luma component of the MB as illustrated in Fig. 2. The arrows in this figure indicate the direction of prediction in each mode. Fig. 3 shows the first frames of *Silent* and *Foreman* CIF sequences and MBs coded by intra16×16 prediction modes. The MBs presented in Fig. 3(b) are in homogeneous area in an image. On the contrary, the MBs are coded by an intra4×4 prediction modes when the MB is in edge or irregular region.

To achieve the highest coding efficiency for H.264 intra prediction, the encoder tries all the possible modes and chooses the best one in terms of least rate-distortion cost (RD cost) [14]. Typically, the prediction of best coding mode is predicted by using those neighboring pixels that are in the same direction of the edge because the pixels along the direction of local edge normally have similar values. For example, the nine prediction modes (0-8) are calculated for the 4x4 block shown in Fig. 4 and the resulting prediction blocks are shown in Fig. 5. In this case, the best match to the actual current block is given by mode 4 (diagonal down-right)

TABLE I
RELATIONS BETWEEN PREDICTION MODE AND DIRECTION OF LOCAL EDGE

Sequence	N_I	N_{II}	N_{II}/N_I (%)
Foreman	75616	66775	88.3
Coastguard	162660	137871	84.7
Akiyo	58007	51003	87.9

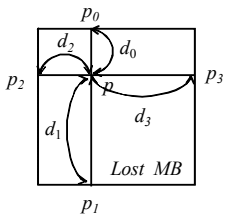

Fig. 6. Spatial interpolation: weight averaging.

because this mode gives the smallest RD cost and a visual comparison shown that the block appears quite similar to the original block. Table I shows the relations between best coding mode and direction of local edge. N_I is the number 4×4 edge blocks in MBs coded by intra4×4. Among edge blocks, N_{II} is the number of blocks whose direction of prediction mode is similar to the direction of local edge. It can be seen from Table I that prediction direction of best coding mode has high correlation with direction of local edge. Thus the proposed method uses the prediction modes included in a H.264 coded stream as edge direction of 4×4 blocks without edge detection.

B. Spatial EC of in the reference software of H.264

The SEC algorithm used in the H.264 reference software (JM 10.1 [15]) is based on the weighted averaging interpolation (WAI) [6] of the values of the four pixels located at the vertically and horizontally neighboring boundaries of a lost MB. Thus, as shown in Fig. 6, the pixel value p of the lost MB is interpolated with the weighted average of four boundary pixel values as follows:

$$p = (p_0 \cdot d_1 + p_1 \cdot d_0 + p_2 \cdot d_3 + p_3 \cdot d_2) / \sum_{i=0}^{3} d_i , \qquad (1)$$

where p_0, p_1, p_2 and p_3 are the edge pixels of the four neighboring MBs of the lost MB, and d_0, d_1, d_2 and d_3 are the distances between the pixel to be restored in the lost MB and its four adjacent pixels, respectively. As shown in (1), the SEC method for Intra-frame in H.264 takes only the weighted average of the vertically and horizontally neighboring boundary pixels regardless of the edge characteristics such as the local edge directions and strengths. This method performs well when the lost MB is smooth or the area including vertical/horizontal straight edges. However, when the lost MB has irregular region, it produces visual artifacts within the region.

III. PROPOSED METHOD

Spatial-domain EC algorithms usually interpolate the lost MB by using decoded information of spatially surrounded

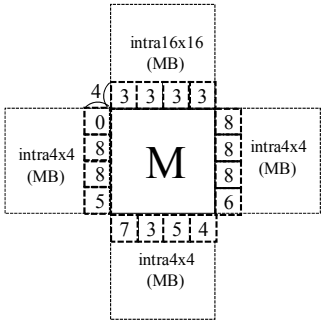

Fig. 7. Lost MB M with its surrounding neighborhood MBs coded by intra16×16 mode or intra4×4 mode.

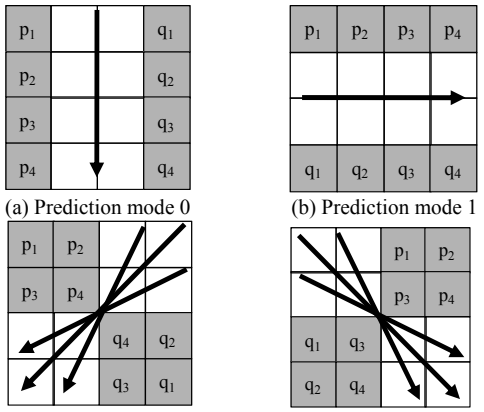

Fig. 8. Eight pixel values located at the perpendicular direction of the prediction direction in each prediction mode for calculating the edge magnitudes of 4×4 blocks.

image data. Especially, edges play an important role in human visual perception of images due to their sharp changes or discontinuities in pixel value. Therefore, the corrupted MB should be recovered along the edge directions in order to preserve edge structure. Fig. 7 shows an example of a corrupted MB surrounded by correctly decoded MBs. Three of the neighboring MBs are coded by intra4×4 modes and the other neighboring MB above the lost MB is coded by intra16×16 mode. The numbers described in the 4×4 blocks represent prediction modes. When neighboring MBs are coded by intra16×16 or intra8×8, we consider that a prediction mode of a 4×4 block is repeated. For example, if a MB is coded by intra16×16 using prediction mode 3, we assume that all 4×4 blocks in this MB have prediction mode 3 as shown in Fig. 7. It is because the prediction directions of prediction mode 0 to 3 in intra16×16 are identical with those in intra4×4 (see Figs. 1 and 2).

A. Estimation of the Dominant Edge Direction (DED)

The prediction modes of intra-blocks located around a lost MB are able to be exploited efficiently for the purpose of estimating the edge directions of the lost MB, because each prediction mode should be determined according to the edge direction of the corresponding block as described in section II. However, these prediction modes do not provide any information about the edge magnitude. Therefore, in order to accurately estimate the edge directions of a lost MB, the edge

magnitude of each block as well as prediction modes is to be considered. In typical methods [8] and [9], edge features of lost MB are detected by directly applying the Sobel operator to boundary pixels in neighboring blocks. However, the extra computation load is still high because Sobel operator should be applied on each pixel and the edge direction histograms also need to be calculated.

To calculate the edge magnitude (EM) of neighboring 4x4 blocks, we decide eight pixels according to the prediction mode k, as illustrated in Fig. 8. Then, the edge magnitude of block having prediction mode k is calculated as follow:

$$EM_{k,j} = \frac{1}{4}\sum_{i=1}^{4}\left|p_i - q_i\right|. \tag{2}$$

where j is the index of the neighboring blocks.

Once the edge magnitudes and prediction modes of sixteen neighboring 4×4 blocks around a lost MB are obtained, we define the edge strength (ES). The ES represents the characteristic of an edge in each of the eight directions as shown in Fig. 1(b). The ES_k for a direction represented by prediction mode k can be also defined as

$$ES_k = W(k)\sum_{j=0}^{N_k} EM_{k,j},$$
$$W(k) = \begin{cases} 0 & k = 2 \\ \lambda N_k & otherwise \end{cases}, \tag{3}$$

where j is the index of the neighboring blocks, N_k is the total number of blocks having prediction mode k, $W(k)$ is a weight function, and scale factor λ. For prediction mode 2, we assume that there is no edge, and the weight function returns 0.

After completely compute the ES values for each direction, the DEDs are selected among the eight directions. The direction having a strong ES can represent real edge direction, but the direction having a weak ES is considered to have insignificant ones. The set of DEDs can be defined as

$$S_{DED} = \left\{k \mid ES_k > \beta \cdot max(ES)\right\}, \tag{4}$$

where $max(\cdot)$ is a function to select the strongest ES among ones for eight directions (ES_k) and β is a scale factor whose value is lower than one.

Edge Recovery

In this step, the neighboring 4×4 blocks are classified into the edge block or non-edge block, and then the edges in the lost MB are restored by using edge features of edge blocks as illustrated in Fig. 9. In order to select edge block, we use the edge magnitude and prediction mode. Among sixteen 4×4 blocks around a lost MB, we select the blocks where their prediction mode are one of DEDs and edge magnitudes are larger than a pre-specified threshold value T as follows

$$Edge\ Block = \left\{j \mid EM_j > T \text{ and } k_j \in S_{DED}\right\},$$
$$j \in Neighboring\ blocks. \tag{5}$$

where j is the index for neighboring blocks, EM_j and k_j denote edge magnitude and prediction mode of block j, respectively.

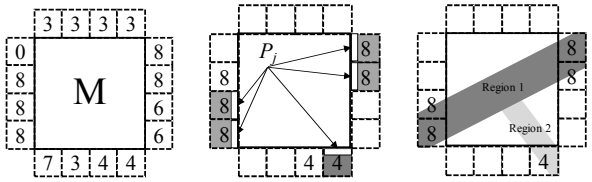

Fig. 9. Edge recovery process. (a) Corrupted MB and surrounding 4×4 blocks with prediction modes presented. (b) Edge blocks. (c) Block matching and linking.

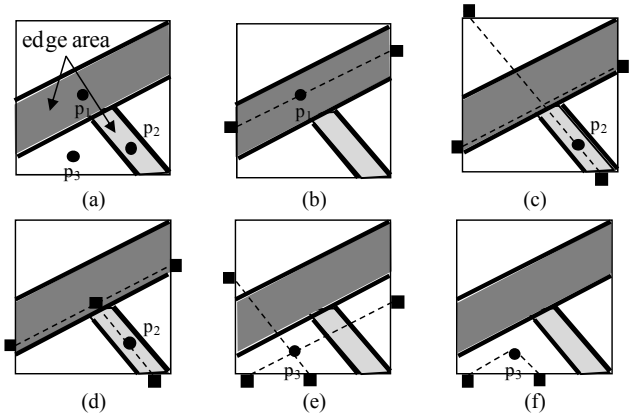

Fig. 10. Sequential directional interpolation.

Fig. 9 (b) shows an example of the selection of edge block in case that DEDs are chosen as prediction mode 8 and 4.

It is assumed that there are two types of edges. The first type is that an edge in 4×4 block is connected to one in another block, as region 1 in Fig. 9(c). The second type is that the edge in 4×4 block meets anther edge within the MB, as region 2 in Fig. 9(c). Based on this assumption, we compare the edge blocks to find the matched pairs. The attribute vector $a(j)$ of an edge blocks j is defined as

$$\alpha(j) = (\theta_j, P_j, E_j^h, E_j^v), \tag{6}$$

where θ_j represents the angle of prediction mode and P_j is the four boundary pixels in Fig. 9 (b). Each element in $a(j)$ gives similar contribution for an edge block. So, by normalizing each term by one, a simple attribute distance between two edge blocks j_1 and j_2 can be calculated via

$$d\left(\alpha(j_1) - \alpha(j_2)\right) = \left|E_{j_1}^h - E_{j_2}^h\right| + \left|E_{j_1}^v - E_{j_2}^v\right| +$$
$$\left|\theta_{j_1} - \theta_{j_2}\right| + \left|\theta_{j_1,j_2} - \theta_{j_1}\right| + \left|\theta_{j_1,j_2} - \theta_{j_2}\right| + E(j_1, j_2), \tag{7}$$
$$E(j_1, j_2) = \frac{1}{4}\sum_{i=1}^{4}\left|P_{j_1}^i - P_{j_2}^i\right|,$$

where $\theta_{j1, j2}$ is the slant angle of the line connecting the center points of these blocks and E is the sum of absolute difference of boundary four pixels for two blocks. The blocks are deemed to be matched if their attribute distance is the smallest among all. The same process is performed iteratively until all blocks

are matched or the attribute distance between two blocks is still above a certain threshold. Finally, corner points of each matched blocks are linked together to recover edges in lost MB. After edge linking of all matched blocks, if there is still unmatched edge block, it is extended into the lost MB along the prediction direction of prediction mode until it reaches an edge line as shown in Fig. 9 (c).

B. Sequential directional interpolation

After edges are recovered, the corrupted MB can be divided into several regions that are composed of edge and flat area as shown in Fig. 9. In this step, pixel values in edge areas are restored by using the DEDs, and then the pixel values in flat areas are recovered. Fig. 10 depicts the example of the classification of the edge/flat area in case that directions of prediction mode 8 and 4 are chosen as DEDs. The pixels p_1, p_2 and p_3 in the corrupted MB are interpolated using boundary pixels in the same region to smoothly recover the lost information. For each lost pixels p_1, p_2 and p_3, we find its reference pixels to be used in interpolation process. In order to recover corrupted pixel p_1 which is in the region of connected edge area, the two reference pixels are obtained along the DED as shown in Fig. 10(b). In case of pixel p_2 which is in the region of edge area, this edge meets another one within corrupted MB. Therefore, reference pixels are obtained along two DEDs, as shown in Fig. 10(c). Note that reference pixels within same region as p_2 are reliable due to discontinuities caused by edges. Thus, three pixels in same region are selected among the reference pixels as shown in Fig. 10(d). In case of pixel p_3, although the region including this pixel is not edge area, it is necessary that pixel values in this region are recovered along the surrounding edge directions to smoothly restore the lost information. Therefore, reference pixels along the DEDs are determined as shown in Fig. 10(e). Then the pixels close to lost pixel p_3 among them are selected as shown in Fig. 10(f). After selecting reliable reference pixels, the corrupted pixel p can be directionally interpolated via

$$f(p) = \sum_k \frac{p_k}{d_k} \Big/ \sum_k \frac{1}{d_k}, \tag{8}$$

where p_k is the kth reliable reference pixel, and d_k represents the distance from p_k to p. If matched blocks cannot be found in previous step then no reference pixel is available. In such case, the lost pixels are restored by MDI method.

SIMULATION RESULTS

A. Environment of the Implementation

To evaluate the performance of the proposed method, our method is implemented into H.264/AVC reference software JM 9.8 [14] and various test sequences are encoded. The frame size is QCIF and CIF, and the frame rate 30 frames/s. The baseline profile is employed, and all frames are intra-coded. We do not consider P or B-frames, since the target in the proposed method is intra-frame concealment. The number of encoded frame of each run is

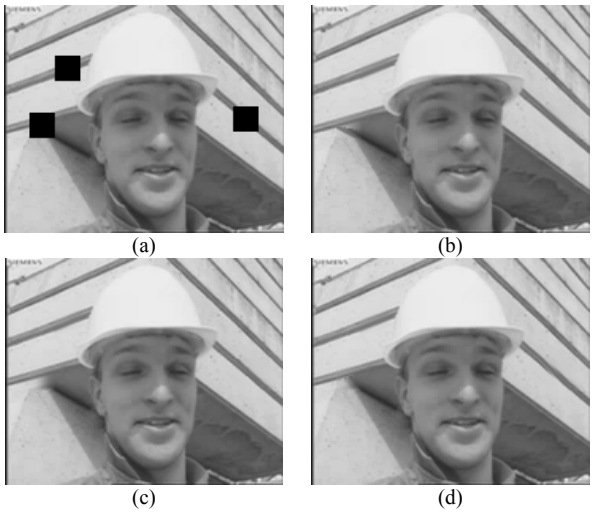

Fig. 12. Frame concealment results when 3 MBs are lost. (a) Error locations (18.56 dB). (b) GSB (37.52dB). (c) FDI (37.85dB) (d) Proposed algorithm (39.16 dB).

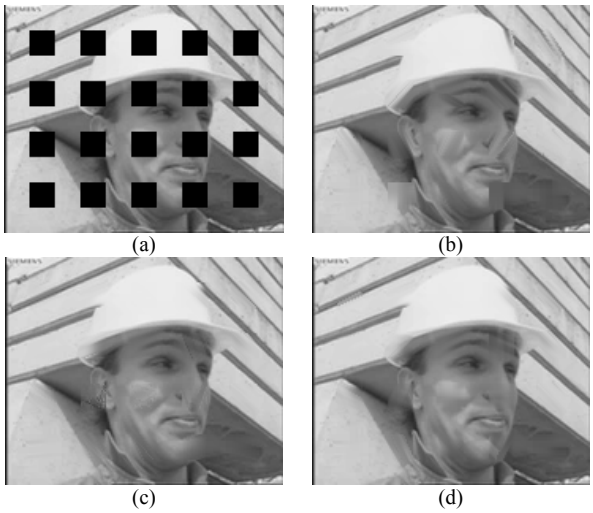

Fig. 13. Frame concealment results BER with 20% (QCIF) (a) Error locations (10.38 dB) (b) GSB (30.39 dB) (c) FDI (33.79 dB) (d) Proposed algorithm (34.02 dB)

30. Also, in order to increase the error robustness, we select "dispersed map" which modifies the assignment of MBs to a slice. When one packet is corrupted, the corrupted data can be concealed more effectively using the information in the remaining slice groups. The number of MBs is calculated according to the block error rate (BER) given by

$$BER = \frac{\#\,of\ corrupted\ MBs}{\#\,of\ total\ MBs}. \tag{9}$$

To simulate error conditions, the BER is set to 5%, 10% and 20%, respectively. Two conventional methods FDI [9] and GSB [10] are compared with the proposed method. We present the average PSNR for Y performance. The PSNR's for U and V components exhibit similar tendencies. In our experiment, λ and β are set to 1 and 0.4, respectively.

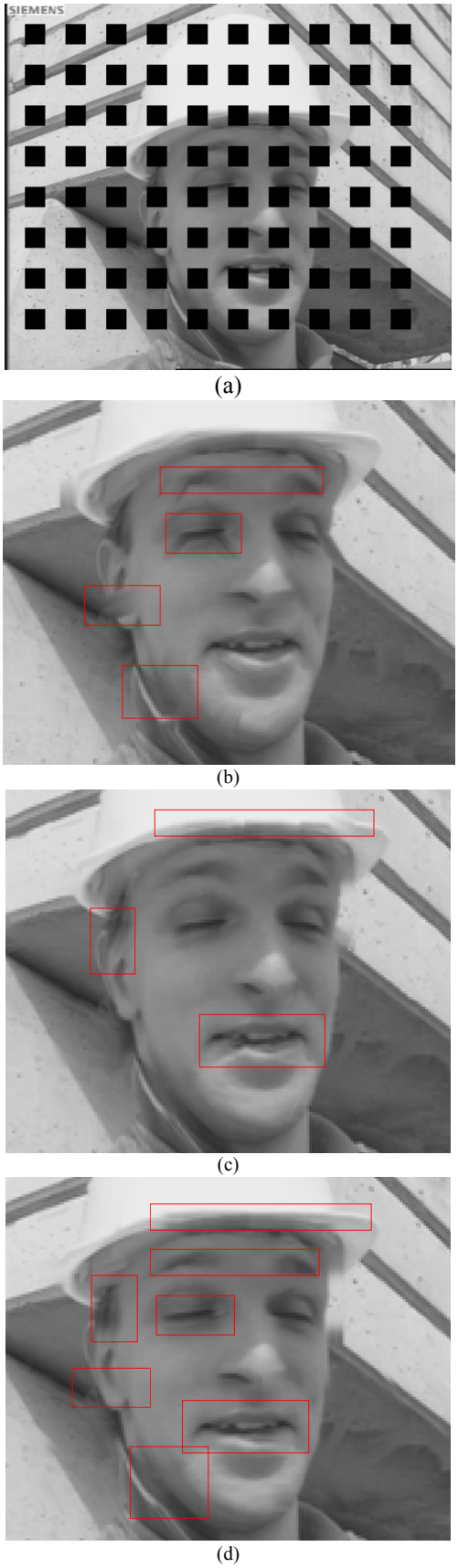

Fig. 14. Frame concealment results BER with 20% (CIF). (a) Error locations (10.56 dB). (b) GSB (33.07dB). (c) FDI (33.80dB). (d) Proposed algorithm (34.86 dB).

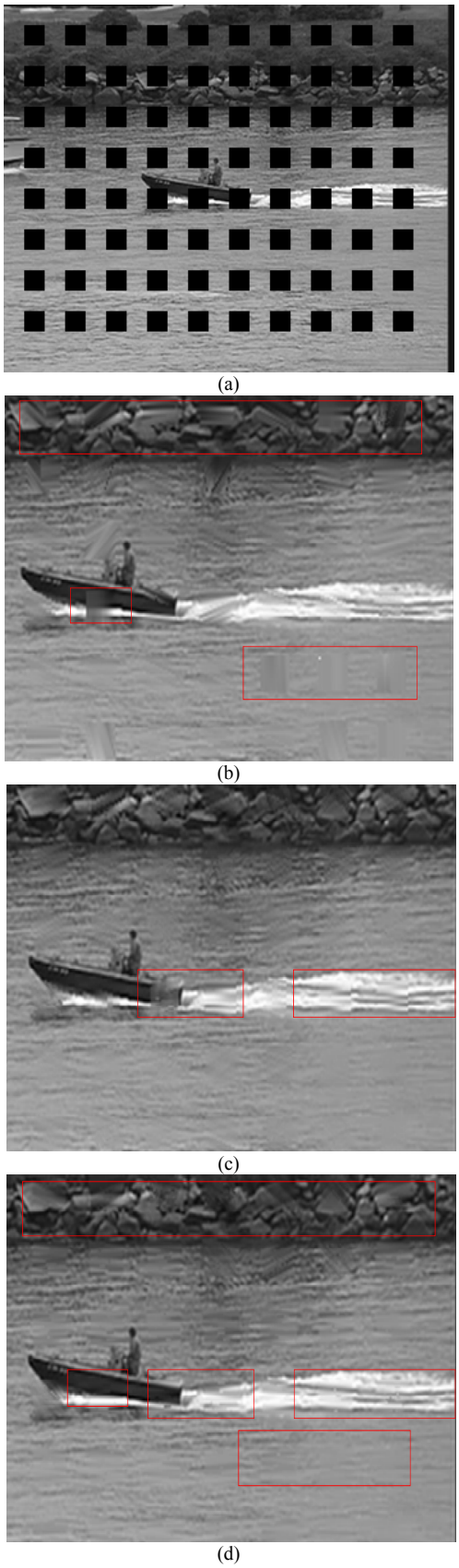

Fig. 15. Frame concealment results BER with 20% (CIF). (a) Error locations (10.50 dB). (b) GSB (29.49dB). (c) FDI (29.75dB) (d) Proposed algorithm (30.73 dB).

TABLE II
PERFORMANCE COMPARISION IN PSNR WITH QP=20

Video Sequence	BER (%)	QCIF			Average
		Akiyo	Coastguard	Container	
GSB	5	38.94	35.20	31.10	**35.08**
	10	32.72	32.30	28.69	**31.23**
	20	30.00	30.79	28.45	**29.74**
FDI	5	39.72	35.52	31.68	**35.64**
	10	33.12	32.91	29.17	**31.73**
	20	30.15	31.76	28.57	**30.16**
Proposed	5	39.58	35.36	31.60	**35.51**
	10	33.32	32.79	29.17	**31.76**
	20	30.16	31.80	28.99	**30.31**

TABLE III
PERFORMANCE COMPARISION IN PSNR WITH QP=20

Video Sequence	BER (%)	CIF			Average
		Foreman	Football	Mobile	
GSB	5	40.75	37.69	28.96	**35.8**
	10	33.73	29.37	24.13	**29.07**
	20	30.72	27.31	21.35	**26.46**
FDI	5	41.80	38.66	29.84	**36.76**
	10	35.65	31.05	25.04	**30.58**
	20	32.87	27.60	21.84	**27.43**
Proposed	5	41.24	39.62	29.70	**36.85**
	10	36.22	31.25	24.99	**30.82**
	20	33.20	27.62	22.08	**27.63**

TABLE IV
PERFORMANCE COMPARISON IN PSNR WITH BER=20%

Video Sequence	QP	QCIF			Average
		Akiyo	Coastguard	Container	
GSB	20	30.00	30.79	28.45	**29.74**
	31	29.08	29.39	27.84	**28.77**
	39	27.63	27.17	26.13	**26.97**
FDI	20	30.15	31.76	28.57	**30.16**
	31	29.91	30.19	27.87	**29.32**
	39	27.63	27.49	26.04	**27.05**
Proposed	20	30.16	31.80	28.99	**30.31**
	31	29.26	30.05	28.00	**29.10**
	39	27.88	27.40	26.21	**27.16**

TABLE V
PERFORMANCE COMPARISON IN PSNR WITH BER=20%

Video Sequence	QP	CIF			Average
		Foreman	Football	Mobile	
GSB	20	30.72	27.31	21.35	**26.46**
	31	29.85	26.44	21.18	**25.82**
	39	27.51	24.86	20.19	**24.18**
FDI	20	32.87	27.60	21.84	**27.43**
	31	31.25	26.79	21.70	**26.58**
	39	28.33	25.01	20.68	**24.67**
Proposed	20	33.20	27.62	21.88	**27.56**
	31	31.19	26.63	21.72	**26.51**
	39	28.74	24.98	20.83	**24.85**

TABLE VI
TEST RESULT OF THE COMPUTATIONAL COMPLEXITY.

Average computation time (ms / frame)			
WAI	GSB	FDI	**Proposed**
25	66	47	**40**

B. Performance evaluation of the proposed method

To compare the subjective qualities, simulation results on a corrupted MB are shown in Figs. 12-15. They show the comparisons of the recovered images for the isolated block loss given by GSB, FDI and our proposed algorithm. Observation shows that our method has achieved noticeable improvements. The test image is the frame of the *Forman* QCIF and CIF sequence, which has 43.21-dB PSNR with error-free reconstruction. As shown Fig. 12, three blocks, containing object edges, are lost in this simulation. The two blocks contain parallel edges that are connected edges. These simple edges are detected and correctly concealed by the proposed method and conventional methods. Another block contains intersecting edge with another one. This edge meets another one within the block and does not exit the block. In FDI, this edge is not detected and the resulting interpolation yields a false edge in the concealed MB. In case GSB, this edge direction can be detected. However the resulting interpolation yields a derivative image quality. On the other hand, the proposed algorithm successfully detects the edge direction and provides a better reconstructed result. Figs. 13 and 14 show the result when BER is set 20%. In Figs. 13 (b) and 14 (b), the blurred areas and blocking artifacts can be seen at the wall of the building and face. Also the blocking artifacts are caused by FDI around the hat, the mouse, and the face in Figs 13(c) and 14(c). Compared with these results, the proposed method represses the artifacts around there and produces more acceptable concealed images as shown in Figs. 13(d) and 14(d). Fig. 15 shows the frame of *Coastguard* CIF sequence and the blurred areas can be seen at the boat and water of the reconstructed image by conventional methods. However, the concealed image by the proposed method has the best visual quality as shown in Fig. 15(d). The proposed method achieves not only higher PSNR performance but also better subjective visual quality as compared with the conventional methods.

Table II-V demonstrates the average PSNR of the concealed frames for all test sequence at various BERs and QP, and concealed by different EC methods. In these Tables, FDI objectively outperforms GSB for all test sequences. However, the most PSNRs of FDI are lower than proposed method because the blurred areas can be seen in the concealed MBs. In terms of objective video qualities, the proposed method have achieved about 0.52~0.91 dB higher PSNR's than the conventional methods. We see that the proposed algorithm reconstructs the edges more accurately and provides better image quality.

Next, we evaluate the computation complexity of each method. Comparison results of the computational complexity of all EC methods are summarized in Table VI. The simulation is performed on a PC with 3GHz Pentium process. Table VI shows that the average CPU time for concealing a corrupted frame with BER of 20%. Our method needs less computational time compared to GSB and FDI.

IV. CONCLUSIONS

In H.264 video coding standard, the intra-frame is typically coded by intra-modes, and optimal prediction modes of each intra-MB are included in a H.264-coded bitstream. Moreover the prediction direction in each mode is strongly related with the direction of local edge direction. When the MBs within an intra-frame are damaged, prediction modes can be exploited to

restore the damaged MB efficiently. The proposed method estimates the edge direction of damaged MB by using prediction modes of neighboring intra-blocks. Also broken edges in the damaged MB are recovered. After that the damaged MB is partitioned into several regions, and each pixel in a region is directionally interpolated from the boundary pixels within the same region of lost pixel. Our proposed method reduces the computational complexity on edge detection and reconstruction because of using the predefined prediction mode, and provides the reliable capability of handling highly detailed image structures. The experimental results show that the proposed method outperforms other two existing techniques for intra-frames in terms of both subjective and objective video qualities.

ACKNOWLEDGMENT

This work was supported by a Korea University Grant.

REFERENCES

[1] ISO/IEC JTC1/SC29/WG11 (MPEG), "Coding of audio-visual objects – Part 10: Advanced Video Coding," *International Standard 14496-10*, ISO/IEC, 2004.

[2] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," *Proc. IEEE*, vol. 87, no. 10, pp. 1707–1723, Oct.1999.

[3] J. Hagenauer, and T. Stockhammer, "Channel coding and transmission aspects for wireless multimedia," *Proc. IEEE*, vol. 87, no. 10, pp. 1764–1777, Oct. 1999.

[4] Y. Wang, M. Orchard, V. Vaishampayan, and A. R. Reibman, "Multiple description coding using pairwise correlating transform,". *IEEE Trans. on Image Processing*, 10, 3 (March 2001), 351-366.

[5] L. W. Kang and J. Leou, "An error resilient coding scheme for H.264 video transmission based on data embedding," *Proc of 2004 IEEE Int Conf on Acoustics, Speech, and Signal Processing*, Montreal, QC, Canada, May 2004, vol. 3 pp 257-260.

[6] Y.-K Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta and M. Gabbouj, "The Error Concealment Feature in *the H.26L Test Model,*" *Proc. of ICIP* 2002, pp. II-729 - II-732.

[7] P. Salama, N. B. Shroff, and E. J. Delp, "Deterministic spatial approach," *Signal Recovery Techniques for Image and Video Compression and Transmission*, A. Katsaggelos and N. Galatsanos, Eds. Amsterdam, The Netherlands: Kluwer Academic, Jun. 1998, pp. 212–216.

[8] W. Kowok and H. Sun, "Mulit-directional interpolation for spatial error concealment," *IEEE Trans. On Consumer Electronics*, vol.39, Aug. 1993, pp. 455-460.

[9] W. Kim, J. Koo, and J. Jeong , "Fine Directional Interpolation for Spatial Error Concealment," *IEEE Trans. on Consumer Electronics*, vol. 52, no. 3, pp. 1050-1056, Aug. 2006.

[10] W. Zeng and B. Liu, "Geometric-structure-based error concealment with novel applications in block-based low-bit-rate coding," *IEEE Trans. Circuits Syst. Video Technol*., vol. 9, no. 4, pp. 648–665, Jun. 1999.

[11] J. Park, D.-C. Park, R. J. Marks, II, M. A. El-Sharkawi, "Recovery of image blocks using the method of alternating projections," *IEEE Trans. Image Process*., vol. 14, no. 4, pp. 461 – 474, Apr. 2005.

[12] ——, "Information loss recovery for block-based image coding techniques—a fuzzy logic approach," *IEEE Trans. Image Processing*, vol. 4, no. 3, pp. 259–273, Mar. 1995.

[13] X. Lee, Y.-Q. Zhang, and A. Leon-Garcia, "Image and video reconstruction using fuzzy logic," in Proc. *IEEE Global Telecommunications Conf.*, Dec. 1993, vol. 2, pp. 975–979.

[14] G. Sullivan, T.Wiegand, and K.-P. Lim, "Joint model reference encoding methods and decoding concealment methods," presented at the 9th JVT Meeting (JVT-I049d0), San Diego, CA, Sep. 2003.

[15] JVT Model reference software from http://iphome.hhi.de/suehring/tml/download/oldjm/jm98.zip.

Myounghoon Kim received the B.S. and M.S. degree in computer science from the Kookmin University, Seoul, Korea, in 2003 and 2005, respectively. He is currently working toward the Ph.D. degree in the Department of Electronics and Computer Engineering of the Korea University. His research interests include video indexing for content-based retrieval, image processing, video signal processing, digital broadcasting and other issues on image and video technologies.

Hoonjae Lee received the B.S. degree in Electronic Engineering from the Korea University, Seoul, Korea, in 2006. He is currently working toward the M.S. and Ph.D. joint degree in Electrical Engineering at Korea University. His research interests are image and video signal processing, digital broadcasting, and other issues on image and video technologies.

Sanghoon Sull (S'79-M'81) received the B.S. degree (with honors) in electronics engineering from the Seoul National University, Korea, in 1981, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology in 1983, and Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 1993. In 1983-1986, he was with the Korea Broadcasting Systems, working on the development of the teletext system. In 1994-1996, he conducted research on motion analysis at the NASA Ames Research Center. In 1996-1997, he conducted research on video indexing/browsing and was involved in the development of the IBM DB2 Video Extender at the IBM Almaden Research Center. He joined the School of Electrical Engineering at the Korea University as an Assistant Professor in 1997 and is currently a Professor. His current research interests include multimedia data management including search/browsing, image processing, Internet applications, and digital broadcasting.