

Adaptive Texture-based Spatial Error Concealment with Data Hiding for H.264

Kun-Chang Liu, Long-Wen Chang

Department of Computer Science
National Tsing Hua University
Hsin-Chu, Taiwan 300
E-mail: lchang@cs.nthu.edu.tw

Abstract—The coding standard such as H.264/AVC or MPEG4 which have a high compression rate made it possible to transmit video data and even real-time video payload over network. However, the influence of error propagation and the sequential data loss in the spatial domain also occur as the side effect of this high compression rate. In order to improve the image quality degraded by transmission errors error concealment techniques are useful. By using the spatial relativity of the image after compression, it is feasible for the decoder to accomplish this job alone without consuming any extra network bandwidth.

In this paper, we propose an efficient method for reconstructing multi-edge cases for error concealment. By manipulating each edge unit separately, deriving the angle of the gradient for each unit by a similar procedure of directional interpolation, forming the best connection pattern with proper judging criterion, and filling the rest of the region by bilinear interpolation, we reconstruct the error blocks. Besides, the estimation of the best error concealment algorithm, which is processed in the decoder, is done in the encoder. Then we embed the estimated result into the image by data hiding techniques. Once the error appeared during the transmission, the decoder can efficiently conceal the loss by the information embedded by the encoder without incurring extra network payload.

2. INTRODUCTION

Error concealment (EC)[1-8,10-13] is necessary for video delivery for H.264[14] over unreliable channels such as the Internet and wireless network to recover corrupted image regions. According to the relation of interaction between the encoder and the decoder, the EC techniques can be divided into three categories. Forward error concealment adds information in the bit streams of the source in the encoder to enhance the ability of error recovery in the decoder. Post processing error concealment carries on the concealment task at the decoder based on the properties of the frame sequences. Interactive error concealment refers to recover the unusable data depending on the dialogue between the codec, such as automatic retransmission request (ARQ). Most of the researches about EC fall into the second category because

post processing error concealment brings the least overhead to the system by solving the problem in the decoder solely without involving relevant operations of modifying elementary codec structure. It does not impose any increase of bit rate, too. However, with the data hiding scheme employed, we propose a forward EC method which can take use of the surrounding macro block (MB) information more efficiently by predetermining MB properties relating to its neighbors in the encoder side without perceptible quality degradation and increase of bit rate, although there are few overheads for data hiding scheme.

There are mainly two domains to implement EC. That is, temporal EC (TEC) and spatial EC (SEC). These two ways are used in P frames and I frames respectively in default structure. TEC utilize the temporal relativity between the current frame and the reference frame to estimate the lost part while SEC reconstructs the blocks by the spatial relativity between the lost MB and its neighbors. Generally speaking, TEC reaches a better quality due to the fact that the temporal domain has more correlation information than the spatial domain when there are no scene change occurs. However, P frames have the effects of error propagation when transmission errors take place. TEC produces an unsatisfactory result when there are scene changes. With this reasons, I frames must be contained in video sequences and SEC must be employed to optimize the EC process in some situation inevitably.

The default algorithm adopted in H.264 is bilinear interpolation (BI). To reach a higher smoothness of error-concealed result without annoying visual perception, BI interpolates each lost pixels in both vertical and horizontal directions with closer reference pixels having a larger weights in the interpolation. When the diagonal consecutive block loss, the output of the edge components was blurred by interpolation

In order to overcome the problem of BI, one of the most popular refined algorithms proposed by Xu and Zhou [6][7] is "Directional Interpolation" (DI). DI separates the interpolation directions into 8 main directions. Each main direction has one direction counter and was partitioned with an equal radian interval $\pi/8$. Each pixel in the nearby MBs is

convolved by an edge operator such as the Prewitt operator to get the gradient, and the angle and the amplitude of the gradient. If the line formed by the current pixel coordinate and its slope (angle) passed the lost MB, the amplitude value will be added to the direction counter corresponding to the closest one of the main directions. As shown in Figure 1, the lost pixel marked with blue rectangle was interpolated along the main direction $\pi/4$ which has maximum counter value. P1 and P2 were chosen to be the reference pixels along the angle $\pi/4$ when DI has been performed.

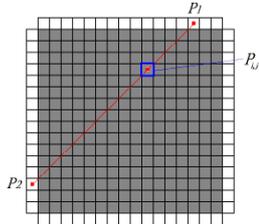


Figure 1: DI with the maximum direction counter of $\pi/4$.

Usually, for the lost MB passed by edges with a certain direction, DI has superior results of EC than BI's. However, there are some situations that DI can not do well. Such cases include multiple edges with obviously different directions passing the lost MB. Since DI selects a single direction to perform the interpolation, it can only connect smoothly the edges with the angle as the main direction of maximum counter value. Furthermore, when there are irregular objects that are not with straight edge shapes in the lost MB, DI can not do well, too.

3. PROPOSED METHOD

In order to deal with the multi-edge MBs, we propose an algorithm to accomplish the task by manipulating each edge unit separately. By computing the penalty scores of all kinds of link patterns formed by edge units, the most suitable linkage between them will be chosen by two judging criterion. After the scoring stage, the most suitable link pattern will be drawn. The rest pixels which had not been drawn would then be interpolated by regional BI. The algorithm can be divided into the following three steps;

1. Find the edge units.
2. Consider all possible connection patterns and find the most suitable one with judging scores.
3. Draw the pattern found and perform BI with pixels that have not been drawn yet.

The main concept of the proposed edge extension (EX) algorithm is to extend the edges surrounding the lost MB into the lost region, and connect the extended edges in a suitable way. Figure 2(a) is a lost MB with its surrounding pixels. The algorithm detects that there are three edges in the surrounding area and then extends each of them as Figure 2(b) shows. The extending edges form their connections properly as Figure 2(c). After the connection, every isolated region performs BI

separately, and the error concealment for this lost MB is finished.

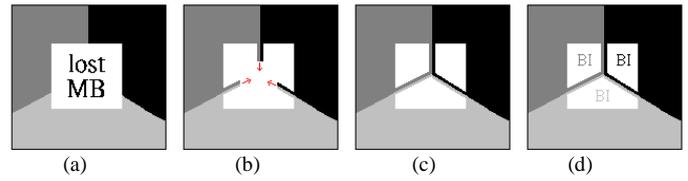


Figure 2: illustration for edge extension.

The first step of EX is to find the edge points that we use to extend into the lost MB. The boundary pixels around the lost MB were numbered from 0 to 63 in the order shown in Figure 3. Five points of edge pixels are viewed as an edge unit for extending the edges pixels in the lost MB for error concealment.

Consider the five pixels $P_{(0+k) \bmod 64}$, $P_{(1+k) \bmod 64}$, $P_{(2+k) \bmod 64}$, $P_{(3+k) \bmod 64}$ and $P_{(4+k) \bmod 64}$ with $k \in \mathbb{Z}, 0 \leq k \leq 63$, which have not belonged to any edge unit yet. The edge difference for the edge unit is defined as

$$edge_difference = |(P_{(0+k) \bmod 64} + P_{(1+k) \bmod 64}) - (P_{(3+k) \bmod 64} + P_{(4+k) \bmod 64})| \quad (1)$$

where p_x is the pixel value in the outer boundary of the lost MB with the index k . If these pixels have the maximum edge difference defined in the equation (1), where the edge difference represents the visibility of an edge, among all possible k , then they were chosen as the edge points for the newly added edge. After we have chosen these five pixels to be the edge points, the rest of the outer boundary pixels of the lost MB form a new pixel set. By finding the new edge points with maximum edge difference within the new set again, we can find another edge points. By repeating this process, we can continue to find new edge points until the edge difference of the newly chosen edge points is smaller than a predefined threshold (in the experiment, 20). The procedure of finding edge points is finished. Note that by choosing the edge points in this way, each boundary pixel only belongs to one edge at most. To attenuate the effect of noise, the boundary pixels can be filtered by a one dimension median filter before finding the edge points. The filtered pixels will be used to replace the original pixels to extend into the lost region.

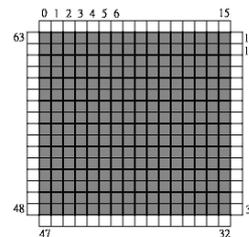


Figure 3: The surrounding pixels around the lost MB.

Now we begin to define the direction of the extension for each edge points. The edge points within the same edge will

be extended in the same direction. The direction of the extension of an edge is defined by the following procedure which is similar to the procedure of finding the interpolation direction of DI.

1. For the 5 x 5 block which is adjacent to all the edge points that we want to get its direction as in Figure 4 . Each pixel in the 5 x 5 block was convolved by a 3 x 3 Prewitt operator except for the location whose convolving filter is overlapped with lost pixels or with the positions out of the frame boundary.
2. The edge tendency was classified into eight main directions as in DI. The amplitude of each pixel in the region computed from step 1 was added to one of the main directions' own counters if this counter's direction has the closest radian to the angle of the current pixel.
3. The direction with largest counter value will be treated as the direction of the edge unit.

The procedure described above is the similar to the process of DI except that the area in which each pixel is convolved by a 3 x 3 Prewitt operator was limited to a specific region. If the edge points were crossing the corner, the above 5 x 5 block for computing the direction of extension will be replaced by a union region of two 5 x 5 blocks as shown in Figure 5, where the union region is the joint area of block 1 and block 2. To derive the direction for this corner edge, we convolve each pixel with a 3 x 3 Prewitt operator in the union region instead for a single 5 x 5 block in step 1.

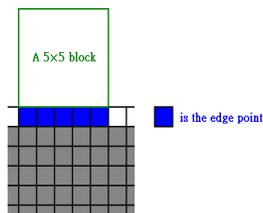


Figure 4: The example of the 5x5 block for the computation of the direction of an edge.

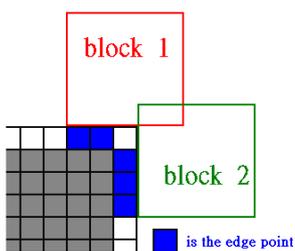


Figure 5 : The union region of the corner edge: The union region is the joint area of block 1 and block 2.

There are three possible ways for two edges to connect with each other. In 6 (a), edge 1 and edge 2, edge 1 has the direction of $\pi/4$, and edge 2 has the direction of $3\pi/4$. Two edges have an intersection in their preceding route. We define that two edges have an intersection if and only if their central edge points have an intersection along their own edge

direction. Suppose that p1, p2, p3, p4, and p5 are the five edge points of Edge 1 and q1, q2, q3, q4, and q5 are the five edge points of Edge 2. Figure 6(b), Figure 6(c), and Figure 86(d) are the three possible ways for the connection between the two edges, edge 1 and edge 2.

In Figure 6 (b), the two edges connect with each other. As a result, p1, p2, p3, p4, and p5 connect with q5, q4, q3, q2, and q1 along their edge direction, respectively.

In Figure 6 (c), edge 2 sinks into edge 1. For this situation, p1 and p2 connect with q5 and q4, respectively as Figure 6(b) does. However, p3, p4, and p5 will continue to extend along their direction. Simultaneously, q1 and q2 respectively substitute p1 and p2 to extend along the preceding route of p1 and p2 along p1 and p2's direction. In this situation of linkage, the value of p1 and p2 in the latter computation of scoring stage will be substituted by the value of q1 and q2, respectively.

In Figure 6(d), edge 1 sinks into edge 2 while edge 2 continues proceeding. It is similar to Fig. 6(c) that p4 and p5 respectively replace the extension of q4 and q5 while q1, q2, and q3 keep their route of extension when edge 2 meets edge 1. The value of p4 and p5 will replace the value of q4 and q5 in the latter computation of scoring stage. At the same time, p1 and p2 connects with q5 and q4, respectively.

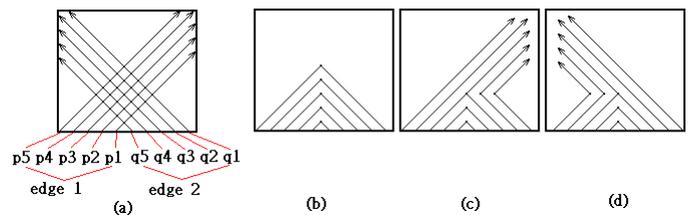


Figure 6: Three possible ways for linkage of two edges. (a) The two edges emitting from their edge points along their directions (b) Two edges end to each other (c) edge 2 end to edge 1 and edge 2 continue to extend along its direction (d) edge 1 end to edge 2 and edge 1 continue to extend along its direction

The connection between two edges can be formed only by the three ways of linkage shown in Figure 6. However, the connection among n edges, $n > 2$, has many possibilities. For example, Figure 7 (b) and Figure (c) are the two possible ways for the connection of all the four edges of Figure 7 (a) Note that the five pixel-width edges in the figures are represented by a single line .

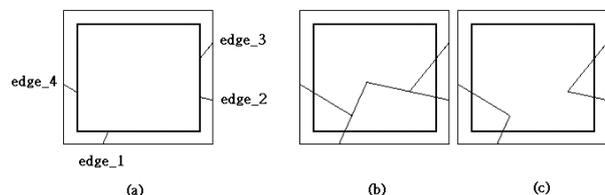


Figure 7: The variations of connection between all edges.

To choose the most suitable connection pattern among all the possible connections of all edges surrounding the lost MB, we use two counters to achieve this goal. The first counter is

linking penalty counter (LPC) which records the penalty score for a connection pattern. LPC defines the degree of smoothness of connection formed by all the links between edge point pairs. A connection pattern with small LPC value is considered better than the pattern with large value. The other counter is visual penalty counter (VPC). Generally, we will choose the pattern in which the edge with high edge difference dominates the edge with low edge difference. It means that, we want the edge with low edge difference to sink into the edge with high edge difference. When there are multiple connection patterns with the same LPC value, we will choose the pattern with smallest VPC to be the final result of connection.

The computation of the LPC for a connection pattern is as the following. When there is a linkage between two edge points, the distance between the gray values of the two edge points is added into LPC except that the two edge points are the center points of their own edge.

The computation of the VPC takes place only when one edge sinks into the other edge. In the three ways of linkage shown in Figure 6, there are only Figure 6(c) and Figure 6(d) that need to modify VPC value of the current pattern. When this two situations occurs, and suppose edge A sinks into edge B, we add the edge difference of edge A into VPC and subtracts VPC value from the edge difference of edge B.

Take Fig. 8 as an example for illustration of the computation of LPC and VPC of a connection pattern. The corresponding gray value for each of the edge points is shown in Table 1. To compute the LPC value, we need to find the sum of the distance between the gray values of the edge points which form a linkage, except for the central point of the edge. The intersections of such links are marked with red rectangle as shown in the figure. Therefore, we merely need to add together the absolute value of the difference between the two terminals of each red rectangle-marked link. The result is:

$$|90-88|+|91-86|+|13-15| + |14-12|+|1-6|+|2-7|=21$$

Then, because edge_4 has no linkage with other edges, the distances of the gray values between its edge points (except for the central edge point) and their corresponding boundary points reached along edge_4's direction need to be added into LPC, too. The sum of the distances between these point pairs is computed as the following:

$$|4-3|+|5-4|+|7-61|+|8-62|=110$$

Hence, the final LPC value is 21+110.

Since VPC value needs to be considered only in the situation that one edge sinks into another edge, we have just to consider the link between edge_1 and edge_2. Since edge_2 sinks into edge_1, the VPC value is the edge difference of edge_2 subtracts the edge difference of edge_1. The answer is

$$|(13+14)-(88+86)|-|(90+91)-(1+2)| = 147-178=-31$$

It is possible that a pixel in the lost MB passed by multiple edge points from different edges. For this situation, the edge point with edge of largest edge difference will dominate the pixel, and the edge with large edge difference extends over the other edges with small edge difference.

For more accuracy of the edge direction, the direction for each edge should be modified by a line testing before the scoring stage. The testing is mainly for the edges passing throughout the MB in the original image. For each of the edges in this kind, the EC system will detect two edge units around the MB after loss. Since these two edge units are in the same object edge in the original image, their directions should be close to their direct connection's direction.

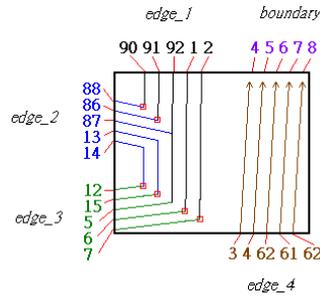


Figure 8 : The example for computation of VPC and LPC.

Table 1: The gray values for the edge points of the edges in Figure 8.

gray value	point_1	point_2	point_3	point_4	point_5
edge_1	90	91	92	1	2
edge_2	14	13	87	86	88
edge_3	7	6	5	15	12
edge_4	62	61	62	4	3
boundary	4	5	6	7	8

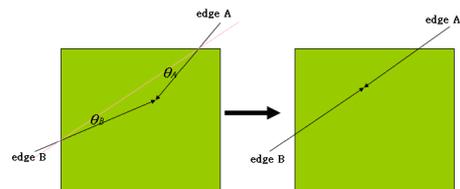


Figure 9: The illustration for modifying a line pair by line testing

By observing the angle and luma (gray) between every two edges in the MB, both the directions of these two edges deemed to be in the same side of object will be modified to point to each other as shown in Figure 9. To test whether two edge 'A' and 'B' are in the same side of the same object, the two quantities θ_T and G_T between two edges are used. θ_T is the angle of relation between edge 'A' and edge 'B'. θ_A is the angle between the direction of edge 'A' and the direct connection of edge 'A' and edge 'B'. θ_B is the angle between the direction of edge 'B' and the direct connection of

edge ‘A’ and edge ‘B’. If $\theta_A + \theta_B$ is a small value, edge ‘A’ and edge ‘B’ is probably on the same straight line of the same object in the MB before the loss. G_T is the difference of edge points’ value between edge ‘A’ and edge ‘B’. After finding the best connection between edges, the original lost MB is partitioned by the extended edges into several sub-regions. The remaining pixels in each of these sub regions will be interpolated by regional BI. In the interpolation, the pixels extended from the edges can also be viewed as the reference pixels.

Figure 10(a) is a possible connection between edges after the scoring stage. In the figure, the edge points and its extended portion are represented in the same color for each edge. Figure 10 (b) corresponds to Figure 10(a). The extended part of Figure 10(b) is represented by the gray color while the rest region is represented in the white color. Once the extension of the edge units is finished, the reconstruction of the extended part for each edge is also finished. The purpose of the regional BI is to assign values to the pixels in the region without passing by the extension of the edges. That is, the region in the white color in Figure 10(b). The only difference between BI and regional BI is that the reference pixels in BI must be on the boundary, while the reference pixels in regional BI could be the boundary pixels or the extended part of the edges. For example, the interpolation of the pixel marked by the blue rectangle in Figure 10(b) finds the reference pixels along the vertical direction and the horizontal direction as BI does, except that the extended part of edges is taken as the reference if it was met before the boundary pixel in the direction. In Figure 10 (b) the boundary references for the current interpolating pixel marked with blue rectangle are marked by gray rectangles, and the references on the extended edges are marked with red rectangles. The remaining procedure is the same as BI.

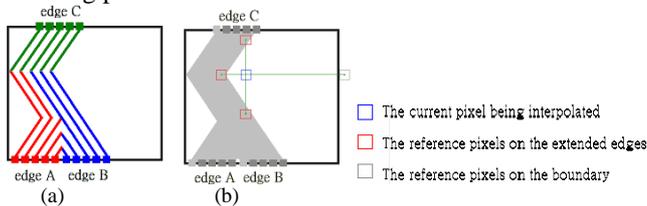


Figure 10: (a) The connection between edges. (b) Illustration for regional BI.

There are mainly two situations that EX can not perform well. The first situation is that, we want to reconstruct MB that is very complex and contains many objects of sophisticated shapes that can hardly be described by lines in it. The second situation is the reconstruction of a certain MB that, the 5×5 blocks in its surrounding area for computing the edge directions when EX performed can not provide enough information for EX to compute the correct direction for each edge. However, in the first situation, BI can usually reach a higher PSNR value than EX due to the interpolation which produces small square error in the reconstructed area. In the second situation, DI can usually perform better than EX because DI computes the direction of the lost MB from all

neighbor pixels, and thus can catch the entire edge trend (that has probably the same direction with the false-estimated 5×5 blocks considered in EX) which EX fails to do.

To overcome the two situations that EX can not perform well, we propose a hybrid scheme to conceal the error in the lost MB in the media sender by embedding the EC mode that indicates which one of the three algorithms (EX, BI, or DI) should be used to reconstruct the lost MB in the receiver side. For each MB in the frame, we can simulate its corresponding frame error by simulating the loss of the packet in which the current MB resides. Once the error pattern for this MB has been known, each of the three EC algorithms (EX, BI, and DI) was applied to reconstruct the lost MB. The mode of the algorithm that introduces the least MSE (mean square error) between the original MB and its reconstructed MB will be embedded in the frame.

Data hiding is performed right after the quantization of the coefficients produced by the DCT-based transform applied to the motion-compensated residual data. Since H.264 takes each 4×4 residual block as an unit for DCT-based transform, by modifying the coefficient in the position (1,1) shown in Figure 11 with “even-odd” algorithm which modifies the coefficient to even if “0” is hidden and modifies the coefficient to odd if “1” is hidden, each 4×4 residual block is capable of embedding one bit, so there are at most 16 bits that can be hidden in one MB. Since there are three methods to be chosen (EX, DI and BI), the first bit of the MB represents whether EX should be employed in the decoder. And the second bit will be modified only when EX had not been used. The meaning of the second bit is DI or BI should be used.

The information of EC mode of the current MB should be hidden in the MB that will be transmitted in different packet by the encoder to prevent that the MB and the hidden mode lost together. In the experiment, the mode for EC of the MB in the position (x, y) is hidden in the MB with the position:

$$\left(\left(x + \frac{image_width}{2} \right) \bmod (image_width), \left(y + \frac{image_height}{2} \right) \bmod (image_height) \right)$$

If the MB with data hiding for current MB is also lost, the concealing method used in neighbor MBs with most frequency will be employed instead since there is high correlation between nearby MBs that are possibly hidden with the same EC mode.

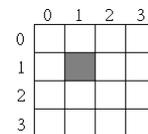


Figure 11: The position of data hiding in a 4×4 residual block.

The flowchart of the data hiding scheme is shown in Figure 12. Before the transmission of a frame, for the MBs that may possibly be lost in the error-prone channel, the encoder tries to reconstruct each MB with all the three methods and compute the MSE of each reconstructed MB with the original MB. The encoder then embeds the EC mode for the method

that produces least MSE, and begins the transmission of the current frame. Once the frame suffering an error during the transmission, the decoder will receive the frame with some MB lost. For each lost MB, the decoder tries to retrieve the embedded mode hidden by the encoder. If the MB for embedding survives during the transmission, then the hidden bits for the EC mode will be retrieved successfully and the hidden mode for EC can be used to determine the algorithm that the decoder should perform; otherwise, for the situation that the hidden bits for EC mode are also lost, the EC mode hidden for nearby MBs surrounding the lost MB will be used instead. After the retrieval, the decoder will reconstruct the lost MB and finish the EC process for the current block.

4. EXPERIMENT RESULTS AND DISCUSSION

In the experiment, we first show that proposed EX can provide better enhancement in multi-edge regions. For the image "Lena", Figure 13 (a) is the original image without error. Figure 13 (b) is the frame after data hiding from the original frame. Figure 13(c) is the frame with a lost MB. Figure 13 (d) is the reconstructed frame from the loss by BI. Figure 13 (e) is the reconstructed frame from the loss by DI. Figure 13 (f) is the reconstructed frame from the loss by the proposed EX algorithm with QP=24, and the threshold of the edge difference is set to 20. The corresponding zoom-in images for Figure 13 are shown in Figure 14. Note that the PSNR in the parentheses shown in Figure 13 are for the whole image whereas the PSNR in the parentheses shown in Figure 14 are for the reconstructed MB.

When a loss MB (in the red rectangle) has appeared just in the edge-crossing region as shown in Figure 13(c), we compare BI, DI, and EX in turn. First, by using BI, the loss in the reconstructed image has been concealed with the interpolation along both the vertical and the horizontal directions as shown in Figure 13(d). Clearly, the reconstructed region is blurred due to the fact that the BI interpolates the loss region in vertical and horizontal direction regardless of the edge tendency as is the results in most reconstructed edged MBs by using BI. Secondly, we reconstruct the lost MB by DI. The result of the MB reconstructed from Figure 13(c) by DI is shown in Figure 13 (e). With the gradients of pixels in the nearby MBs been taken into consideration in DI, DI interpolates the broken MB in the main direction of largest counter value which represents the sum of the amplitudes of the gradients along this direction. For this reason, DI can recover the broken edge with the direction of most visual significance. However, when there is more than one edge contained in the lost MB such as the situation in Figure 13(c), DI can not recover both edges simultaneously. Therefore, only the edge along $\pi/4$, which is the main direction of most gradient counter value, has been recovered in the figure whereas the other edge which is about $3\pi/4$ was not be considered by DI in the figure. Thus, the portion of the reconstructed result of the $3\pi/4$ edge was blurred since it is also interpolated in the same direction $\pi/4$ as the most

significant edge is. Finally, we examine the same lost MB reconstructed by the proposed EX algorithm. As the EX-reconstructed MB shown in Figure 13 (f), we first detect that there are mainly three edge units in the surrounding region of the lost MB. By judging the connection patterns with the visual score, edges with slight visibility shall not dominate the extending direction of each of the three main edges. In other words, when there is a slight edge with small edge difference meets a heavy edge with large edge difference, the slight edge shall sink in the heavy edge and stop extending, while the heavy edge continues to extend along its edge direction. Actually, this property is what we expect that the visual score should do. Unlike DI, we extend each edge unit separately and find the best connection pattern between them. Since each edge extends without affecting the directions of the other edges, each of the gradient for every edge unit will be maintained and form a clear topology of the edge connection. In fact, DI has better performance than BI, and EX has better performance than DI in most of the lost MBs that have multiple, simple and regular-connected edges.

For the purpose, diagonal consecutive loss was imposed on the first frame of "Foreman" sequence in Figure 15. In the experiment, we first reconstruct the image by using BI, DI, or EX for all the lost MB separately. Then the hybrid scheme with data hiding was applied to reconstruct the image and the result of concealment was compared with each of the three algorithms. In Figure 15, (a) is the original first frame of "foreman" sequence in QIF format. When 25% diagonal consecutive loss has been applied to (a) such as shown in Figure 15(c), the reconstructed frame by BI, DI, and EX is shown in Figure 15(d), (e), and (f), respectively. In the figure, most MBs in the background of the frame consisted of one single edge. Since BI has rarely the ability to recover edged MB, the result of Figure 15(d) has some obvious distortions. The result of DI in 15(e) is much better than BI because DI is suitable to recover the lost MB containing a single edge. Unfortunately, there are some noises in the left side of the frame. As can be seen, there is a straight and thin black line lying closely to the left side of the frame. Because the black color makes the edge with a heavy weight, DI misjudges many edge tendencies of the lost MBs near the left side of the frame to the angle of the noise black line $\pi/2$. The lost MBs in the intersection of the noise line and the background edge are in truth multi-edged MBs. Therefore, EX performs better than BI and DI in such a case with the result of concealment shown in Figure 15(f).

Now we examine the performance of the proposed hybrid scheme with data hiding for mode selection. By embedding the most suitable algorithm, which generates least MSE, derived from the comparison between BI, DI, and EX with the intact original MB each MB is reconstructed by the embedded algorithm. The frame after data hiding is shown in Figure 15(b), and the frame reconstructed by the embedded algorithm is shown in Figure 15 (g). Also, the embedded mode for each MB is shown in Figure 15(h) with BI represented by the color of light yellow, DI represented by the

color of light blue, and EX represented by the color of purple as shown in Table 2. Usually, nearby MBs have similar properties so they are embedded with the same algorithm. If the embedded bits also lost by the channel during the transmission, the modes embedded in nearby MBs of the lost MB can be used to substitute for the lost mode. Compared with the concealment using the same mode for all the MBs, the reconstructed frame of the hybrid scheme has both better visual quality and PSNR even with the overhead of data hiding. By repairing the lost regions with homogeneous texture and regions containing complex objects with BI, repairing the lost MBs with single tendency or regions containing complex shape variety along the edge direction with DI, and multi-edged MBs with EX, the hybrid scheme can choose the most suitable algorithm to conceal the error in the MBs with different properties.

In Figure 15(h), for MBs containing single edge tendency, EX matches edge units with similar gradient and gray values into pairs and modifies the direction of both the edges in the pairs to their relative position directly. On the other hand, DI interpolates the MBs by using one of the fixed eight main directions so sometimes the edge pair can not be aligned properly with the eight selectable fixed-angle directions. Hence EX may perform better in single-edged MBs. However, it is still necessary to reconstruct MB with DI in some situation. Because EX processes each edge separately, the gradient for each edge is derived from a limited region of its corresponding 5 x 5 block. If there is not enough information contained in the 5 x 5 block to provide the correct direction of the edge, the performance of DI can be better than EX. When there are complex and unpredictable edge variations along the edge direction, DI can perform well because of the interpolation which can alleviate the impact of errors caused by the variation along the edge direction. In general, the hybrid scheme reaches 1~3db PSNR enhancement in average. We compare the PSNR with [13] by imposing 25% isolated block loss and 50% diagonal consecutive block loss on 512 x 512 "Lena" image as shown in Table 3. The proposed method has more PSNR improvement than [13] in 25% isolated block loss. It is due to more information that can be used to reconstruct the lost MBs from their surrounding MBs. Since the proposed method utilize the surrounding information to conceal the error, the more information available we can catch from the surrounding pixels, the more PSNR gain the proposed method can approach.

5. CONCLUSION

We proposed a novel spatial EC method to extend each separate edge into the lost MB with proper judging criterion to form the linkage between edges, so as to recover the broken edges especially in multi-edged MBs. To alleviate the effect of the two false-determined situations in the proposed EX method, DI and BI was employed to deal with each situation separately. Consequently, a hybrid scheme was proposed. By hiding the EC mode introducing the least MSE for each MB into its corresponding MB in the encoder, the decoder would apply the hidden mode to perform the EC process if the MB

for the current mode survived during the transmission. With the data hiding scheme employed, each of the three EC methods in the hybrid scheme takes part in the recovery of appropriate MBs with dedicated properties. At the same time, for each MB, since the encoder know the loss pattern incurred by the loss of the packet in which the current MB resides, a more precise decision for EC mode selection could be made. Hence, better image quality would be achieved after EC.

6. REFERENCES

- [1] D. Agrafiotis, D. R. Bull, C. N. Canagarajah, "Enhanced Error Concealment With Mode Selection," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 8, pp. 960 – 973, 2006.
- [2] X. Xiu, L. Zhuo, L. Shen, "A hybrid error concealment method based on H.264 standard," *proceeding of ICOSP*, vol. 2, pp. 16-20, 2006.
- [3] M. J. Chen, W. W. Liao, M. C. Chi, "Robust temporal error concealment for H.264 video decoder," *proceedings of ICCE*, pp. 383 – 384, 2006.
- [4] K. Donghyung, Y. Siyoung, J. Jechang, "A new temporal error concealment method for H.264 using adaptive block sizes", *proceeding of ICIP*, vol. 3, no.3, pp. 928-931, 2005.
- [5] W. K. Li, J. L. Jin, "An error resilient coding scheme for H.264 video transmission based on data embedding", *proceeding of ICASSP*, vol. 3, no. 3, pp. 257-260, 2004.
- [6] Y. L. Xu, Y. H. Zhou, "H.264 video communication based refined error concealment schemes", *IEEE Trans. on Consumer Electronics*, vol. 50, no. 4, pp. 1135 – 1141, 2004
- [7] Y. L. Xu, Y. H. Zhou, "Refined video error concealment over wireless IP network", *proceedings of the IEEE 6th Circuits and Systems Symposium*, vol. 1, pp. 257 – 260, 2004
- [8] S. Wenger, "H.264 over IP," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645- 656, 2003
- [9] T. Wiegand, G. J. Sullivan, G. Bjntegaard, A. Luthra, "Overview of the H.264 video coding standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560- 576, 2003.
- [10] Wei-Ying Kung, Chang-Su Kim, C.-C. Jay Kuo, "Spatial and Temporal Error Concealment Techniques for Video Transmission over Noisy Channels" *IEEE Transaction on circuits and systems for video technology*, vo.6, No.7, July 2006, pp. 789-802
- [11] Salama P. Shroff N. and Delp E.J, "A fast suboptimal approach to error concealment in encoded video streams. "in *Proc.Inr. Con\$ Image Processing* vol. 2, pp. 101-104, Oct. 1997.
- [12] Shirani S. Kossentini F. and W.R, "An adaptive Markov random field based error concealment method for video communication in an error prone environment" in *Proc ICASSP'99*, vvl. 6, pp.31 173120, Mar 1999.
- [13] Adisom Sirikam, Wuttipong Kumwilaisak, "New spatial error concealment using dynamic texture estimation and geometric interpolation", *Multimedia and Expo, 2007 IEEE international conference on 2-5 July 2007*.

[14] Iain E.G. Richardson “H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia”, The Robert Gordon University, Aberdeen, UK

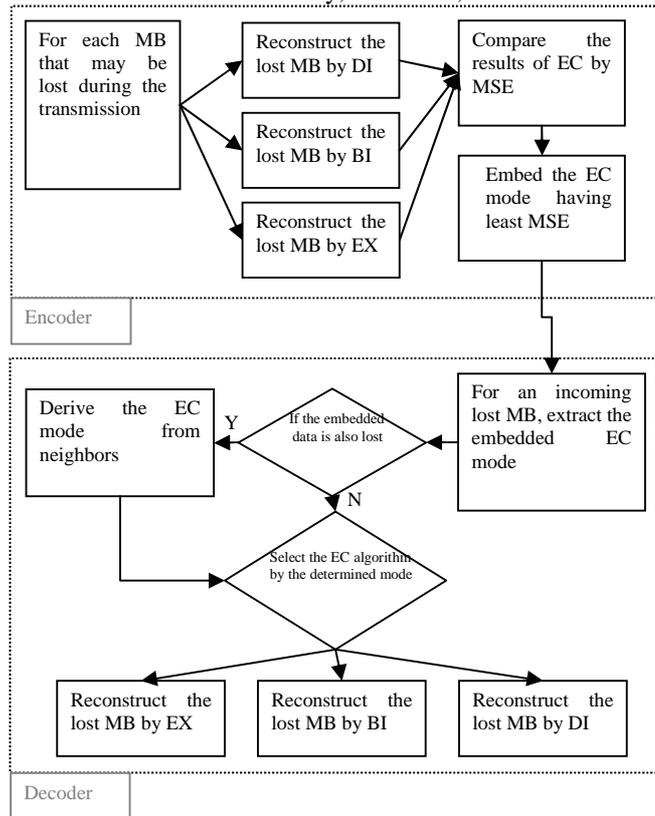


Figure 12: The flowchart of the data hiding scheme.



Figure 13 :Error concealment for Lena(512x512). (a)Original (b)With data hiding (53.16db) (c) A MB loss in the image of (b) (d)BI (48.62db) for the image of (c) (e)DI (47.59db) for the image of (c) (f)EX (51.70db) for the image of (c).

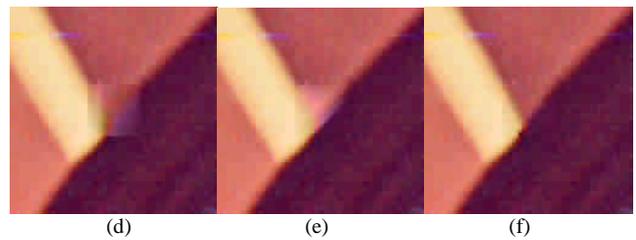
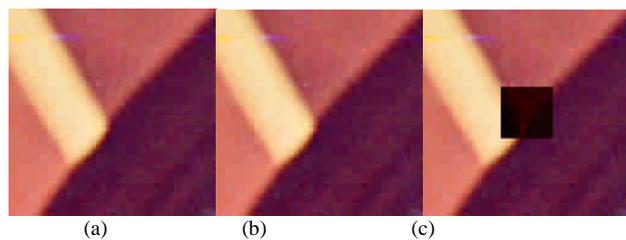


Figure 14: The zoomed in images of Figure. (a)The original image (b)The image after data hiding (PSNR = 58.41db) , (c)The image with a MB loss in the image of (b) , (d)The image reconstructed by BI (PSNR = 20.40db) for the image of (c) , (e) The image reconstructed by DI (PSNR = 19.40db) for the image of (c), (f) The image reconstructed by EX (PSNR = 27.06db) for the image of (c).

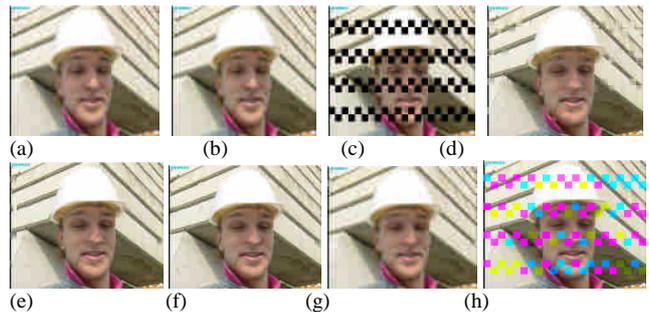


Figure 15: The proposed hybrid scheme for the first frame of foreman (a)The original frame , (b)The frame with data hiding (PSNR = 53.26db) , (c)The frame with a MB loss of the image (b) , (d)The frame reconstructed from the image (c) by BI (PSNR = 28.79db) , (e)The frame reconstructed from the image (c) by DI (PSNR = 30.78db), (f)The frame reconstructed from the image (c) by proposed EX (PSNR = 33.53db), (g) The frame reconstructed from the image (c) by proposed hybrid scheme with data hiding (PSNR = 35.06), (h) The algorithm selection for each MB in (f).

Table 2: The number of MB hidden with each algorithm in Figure15(g) .

	BI (MBs)	DI (MBs)	EX (M Bs)	Total (MBs)
Lena	288	357	379	1024
Foreman	78	138	180	396
Color in (g)	Light yellow	Light blue	Purple	

Table 3 PSNR comparison for “Lena” image with [13]

Block loss	[13] (db)	Proposed (db)
25% isolated block loss	34.12	34.75
50% diagonal consecutive block loss	30.72	30.98