Fuzzy Logic Based Temporal Error Concealment for H.264 Video

Pei-Jun Lee and Ming-Long Lin

In this paper, a new error concealment algorithm is proposed for the H.264 standard. The algorithm consists of two processes. The first process uses a fuzzy logic method to select the size type of lost blocks. The motion vector of a lost block is calculated from the current frame, if the motion vectors of the neighboring blocks surrounding the lost block are discontinuous. Otherwise, the size type of the lost block can be determined from the preceding frame. The second process is an error concealment algorithm via a proposed adapted multiple-reference-frames selection for finding the lost motion vector. The adapted multiple-reference-frames selection is based on the motion estimation analysis of H.264 coding so that the number of searched frames can be reduced. Therefore the most accurate mode of the lost block can be determined with much less computation time in the selection of the lost motion vector. Experimental results show that the proposed algorithm achieves from 0.5 to 4.52 dB improvement when compared to the method in VM 9.0.

Keywords: Error concealment, H.264/MPEG-4 AVC, video transmission, fuzzy logic.

Manuscript received Dec. 12, 2005; revised Aug. 01, 2006.

Pei-Jun Lee (phone: +886 49 2910960 ext. 4985, email: pjlee@ncnu.edu.tw) was with Department of Electronic Engineering, St. John's University, Taipei, Taiwan, and is currently with Department of Electrical Engineering, National Chi Nan University, Puli 545, Taiwan.

Ming-Long Lin (email: 93M12058@student.sju.edu.tw) is with Department of Electronic Engineering, St. John's University, Taipei, Taiwan.

I. Introduction

In video communication, video data are compressed to reduce the bit rate for transmission over a channel. The international video coding standard MPEG-4 AVC/JVT/H.264 has been approved recently by the ITU-T Video Coding Experts Group (VCEG) [1] as the recommended H.264 and Joint Video Team (JVT) of ISO/IEC MPEG-4 as International Standard MPEG-4 part 10 Advanced Video Coding (AVC) [2]. It can reduce the bit rate up to 50% more than the previous video coding standard. However, a bit in high compression coding presents much more information than it does in low compression coding. Any single bit lost in high compression coding often results in the loss of the whole block and seriously degrades the visual quality of the decoded image at the receiver. In order to resolve this problem, H.264 has adopted flexible network adaptation to enhance the error robustness [3], [4]. For instance, automatic retransmission request (ARQ) and forward error control (FEC) coding techniques [5] aim to combat channel errors; however, both of them have some practical limitations. Retransmission is not a feasible option for applications such as conversational and streaming services with constraints on real-time delay. FEC has error recovery ability limitations. If an error cannot be removed by error control methods, the decoded image is still impaired. Thus, an error concealment algorithm is essential for video transmission.

H.264 contains some new coding schemes. One of the major differences between H.264 and the previous coding standards is the motion estimation scheme. In H.264, the motion estimation needs variable block sizes and multiple reference frames to improve the compression performance. Figure 1 shows the motion estimation prediction with multiple reference frames, in which seven different block division modes (16×16,

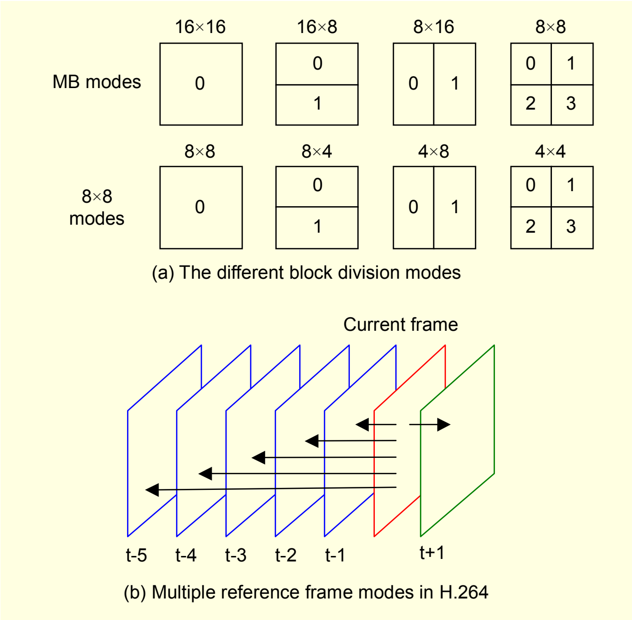

Fig. 1. H.264 motion estimation with multiple reference frames.

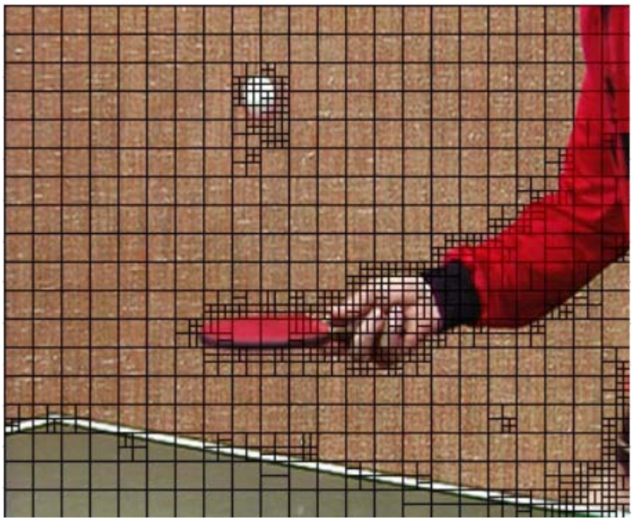

Fig. 2. Block sizes chosen after H.264 RDO.

16×8, 8×16, 8×8, 8×4, 4×8, and 4×4) can be selected. It indeed improves the compression rate and the quality of the reconstructed image. However, such features and functionalities also increase the complexity in encoding and decoding.

For an error concealment scheme, it is known that most traditional motion vector recovery algorithms [5]-[7] designed for 16×16 marcoblocks are not suitable for the variable block size of H.264 standard decoding. Therefore, Wang and others [8] have presented a weighted pixel value average algorithm to conceal the damaged block in the intra picture and have used a modified boundary matching algorithm with the neighborhood motion vectors to conceal the 16×16 macroblock motion vector in the inter pictures. This algorithm has been implemented in the test model of the draft ITU-T video coding standard H.26L. The refined motion-compensated temporal concealment (RMT) in [9] employed the block characteristics to define a matching criterion by which we can choose the most suitable motion vector to conceal the impaired block with size 8×8. Unfortunately, those methods investigated error concealment algorithms for lost blocks of fixed size.

In order to present object edge completeness and reduce the block effect, providing a method to estimate the size type of the lost block is helpful to the consequent error concealment work. It is well known that in a natural video sequence a large area with smooth motions is likely to be coded by using a large block size, and the area containing the boundaries of motion objects is likely to be coded by using a small size. Figure 2 shows an example frame “Table Tennis” from the QCIF sequence, in which different block modes determined by RDO in the H.264/AVC are presented. It can be seen that the edges of the hand and the area of the ball are coded by using a small block size, but the other smooth background is coded by using a large block size. Based on this observation, an error concealment algorithm with block size selection is proposed for the coding standard of H.264 in this paper. The rationale of the size type selection for the lost block is as follows. If the motion vectors of the neighboring blocks surrounding the lost block are continuous, the size type of the lost block can be determined from the block at the same location of the previous frame. Otherwise, it needs a method to select the size type of the lost block from the neighboring blocks of the lost block in the current frame. Because a fuzzy logic technique has been widely used in control systems design and decision making [10], it is adequate to be used here for the type selection of the lost block.

On the other hand, in order to increase the accuracy of the re-estimated motion vector, the error concealment by adapted multiple-reference-frames selection based on the motion estimation analysis of H.264 coding is implemented in this paper. It should be emphasized that the proposed algorithm can determine the most accurate mode of the lost block and significantly reduce the computational complexity in the selection of the lost motion vector. Experiment results have demonstrated that the image recovery performance can be significantly improved.

The rest of this paper is organized as follows: Section II presents the selection of the size type for the lost block. Section III describes the proposed error concealment algorithm for the lost block. Experimental results are shown in section IV. Finally, a conclusion is drawn in section V.

II. The Lost Block Size Selection by Fuzzy Logic

In this section, four types of blocks, namely, 16×16, 16×8, 8×16, and 8×8, are considered in the selection of the lost block type. The selection depends on the degree of reliability of the neighboring motion vector, defined as

$$RD = \sum_{i=A,B,C,D} \left| \left(\frac{1}{4} \sum_{i=A,B,C,D} MV_i\right) - MV_i \right|, \quad (1)$$

where MV_i is the neighboring block motion vector of the lost block and i denotes a neighboring block of the lost block shown in Fig. 3. The degree of reliability (RD) defined in (1) represents the degree of difference among four motion vectors $MV_A, \cdots, MV_D$. A threshold (T) for RD is pre-assigned to determine the lost block type from either the current or the previous frames. If $RD < T$, the motion vector of the lost block is similar to that of the neighboring block. The size type of the lost block is the same as that in the previous frame. If $RD \geq T$, the motion vector of the neighboring blocks surrounding the lost block is discontinuous. In other words, there is probably an edge crossing through the lost block. The previous frame is not reliable to estimate the lost block type; therefore, the block size type needs to be estimated from the current frame.

It is known that if there is an edge crossing through the lost block, the correct selection of the lost block size will improve the reconstructed image, resulting in good block effect elimination and boundary reservation. The boundary information in the surrounding blocks A, B, C, and D should be helpful to correctly select the size type of the lost block. How to select the size type of the lost block as correctly as possible with the boundary information in the surrounding blocks is therefore essential. If there is a vertical edge crossing through the upper or the bottom block (A or C in Fig. 3) of the lost block, it is possible that there is a vertical direction edge of an object through the lost block. Similarly, if there is a horizontal edge crossing through block B or D in Fig. 3, there is probably a horizontal direction edge of an object through the lost block. The above concept will be implemented by if-then rules based on fuzzy logic inference.

Let the row (column) of one-pixel width in the blocks A and C (B and D) adjacent to the border of the lost block be denoted by α and γ (β and ω), respectively, as shown in Fig. 3(b). The edge detection by the Sobel operator is applied to all pixels of α, β, γ, and ω to check the edge crossing. The pixel located on the intersection between the edge and α, β, γ, and ω is called the "crossing pixel" ($\hat{\alpha}$, $\hat{\beta}$, $\hat{\gamma}$, and $\hat{\omega}$). The locations of the crossing pixels will be the inputs to the designed fuzzy rules to determine the size type of the lost block. The fuzzy rule base is established in the following steps.

Step 1. *Fuzzy membership functions setting*

To establish a fuzzy rule base, we have to define fuzzy sets for premise variables and consequent variables for each rule first.

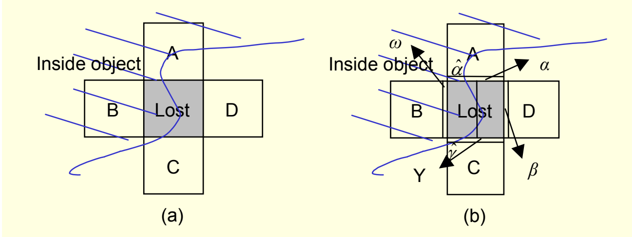

Fig. 3. Illustration of the size determination for the lost block.

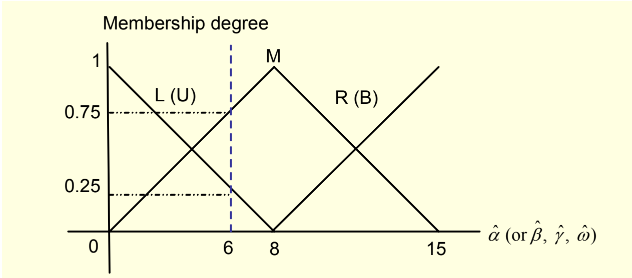

Fig. 4. Fuzzy sets of the pixel locations on each border.

Set three fuzzy sets L (left), M (middle), and R (right) for the crossing pixel locations in α and γ, and U (upper), M (middle), and B(bottom) for the crossing pixel locations in β and ω, respectively, as shown in Fig. 4. For convenience, let $\hat{\alpha}$ (or $\hat{\beta}$, $\hat{\gamma}$, $\hat{\omega}$) denote the crossing pixel location in α (or β, γ, ω), which will be the premise variables of the fuzzy rules. For example, in Fig. 4, $\hat{\alpha} = 6$ means the crossing pixel is the sixth pixel in α from left to right. It belongs to the fuzzy set "middle" with a membership degree of 0.75 and belongs to the fuzzy set "left" with a membership degree of 0.25.

Furthermore, two fuzzy sets S (small) and G (big) for the consequent variables P_v and P_h, respectively, are designed as shown in Fig. 5, where P_v (or P_h) is the probability of the vertical (or horizontal) edge of an object through the lost block.

Step 2. *Fuzzy rule bases establishment*

Rule 1: If $\hat{\alpha}$ is L and $\hat{\gamma}$ is L, then P_v is S.
Rule 2: If $\hat{\alpha}$ is R and $\hat{\gamma}$ is R, then P_v is S.
Rule 3: If $\hat{\alpha}$ is L and $\hat{\gamma}$ is R, then P_v is G.
Rule 4: If $\hat{\alpha}$ is R and $\hat{\gamma}$ is L, then P_v is G.
Rules 5 to 9: Otherwise, P_v is G.

For instance, Rule 3 means that if an edge crosses the left side of α and right side of γ, then the probability of the vertical edge of an object crossing through the lost block is high. The above-mentioned nine fuzzy rules for the vertical edge crossing are presented in Table 1(a) as a lookup table. Similarly, there are nine rules (rules 10 to 18) for $\hat{\beta}$ and $\hat{\omega}$ with output P_h to estimate the horizontal boundary path in the lost block. Table 1(b) presents the nine fuzzy rules for the horizontal edge crossing.

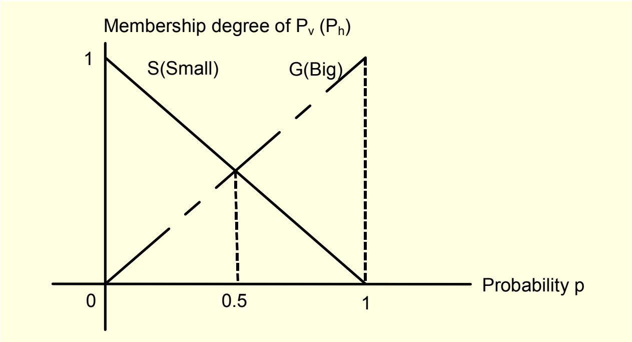

Fig. 5. Fuzzy sets of outputs P_v and P_h.

Table 1. Lookup table of fuzzy rules for (a) P_v and (b) P_h.

(a)

$\hat{\alpha}$ \ $\hat{\gamma}$	L	M	R
L	S	G	G
M	G	G	G
R	G	G	S

(b)

$\hat{\beta}$ \ $\hat{\omega}$	U	M	B
U	S	G	G
M	G	G	G
B	G	G	S

By the max.-min. inference engine [10] as shown in Fig. 6, we can perform the union operation of weighted outputs $\tilde{P}_v = \bigcup_{i=1}^{9} \tilde{P}_{vi}$, where $\tilde{P}_{vi}$ is the weighted output fuzzy membership function of the i-th rule and $\cup$ means the union operation. Then, by the center of gravity defuzzification (or any other defuzzification methods) [10], the final output of Rules 1 to 9 can be obtained as (2) and (3).

$$p_v^* = \frac{\int \tilde{P}_v(p) p \, dp}{\int \tilde{P}_v(p) \, dp}, \tag{2}$$

where p is the argument of the membership function P_v(or P_h), p_v^* stands for the final probability of the vertical edge crossing into the lost block. Similarly, for Rules 10 to 18, the final output is

$$p_h^* = \frac{\int \tilde{P}_h(p) p \, dp}{\int \tilde{P}_h(p) \, dp}, \tag{3}$$

where p_h^* stands for the final probability of the horizontal edge crossing into the lost block.

For instance, suppose we have $\hat{\alpha} = 6$ and $\hat{\gamma} = 10$, the max.-min. inference engine is shown in Fig. 6, where only four rules are triggered, and the other five un-triggered rules are not shown.

In H.264, the error concealment algorithm is implemented from left to right and from top to bottom after all frame data are received. If all neighboring blocks of the error concealing block are lost, blocks A and B in Fig. 3 can always be used to determinate the size type of the lost block in the proposed fuzzy rules by setting $\hat{\gamma} = \hat{\beta} = 0$. If the interleaving slice mode of an error resilience tool is adopted, C block is also usable in the fuzzy rules with $\hat{\beta} = 0$.

Step 3. *Determination of the size type of the lost block*

Table 2 shows the determination of the lost block size based on the fuzzy logic selection. The first row reveals that when the possibilities of both the vertical and horizontal edge crossing into the lost block is higher than 0.5 as listed in Table 1, then the mode of the lost block is most likely to be 8×8. The other rows reveal information in a similar way. As a result, the most likely modes of the lost block can be determined.

After determining the size type of the lost block, we proceed to predict the motion vector so that the error concealment can be completed.

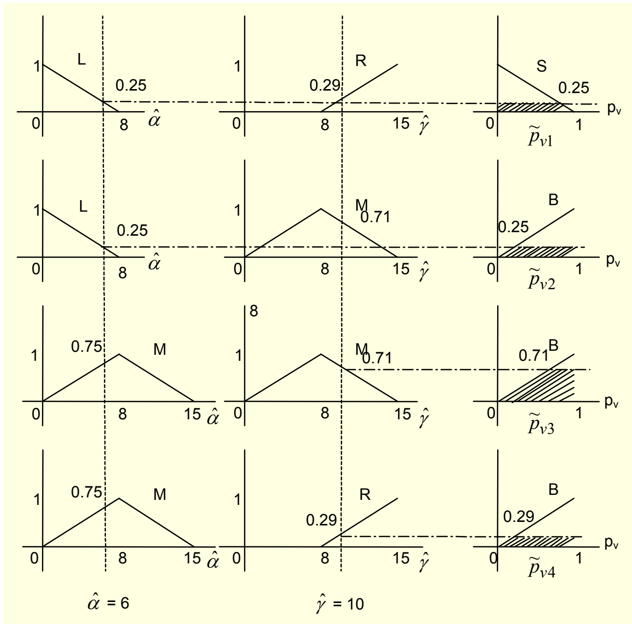

Fig. 6. Fuzzy max.-min. inference engine.

Table 2. Determination of the lost block size.

p_v^*	p_v^*	The size of lost block
≥ 0	≥ 0.5	8×8
≥ 0	< 0.5	8×16
< 0.5	≥ 0.5	16×8
< 0.5	< 0.5	16×16

III. Error Concealment by Multiple Reference Frame

In this section, the motion vector (MV) candidate of the lost block will be predicted by using the MV of neighboring blocks. Then, the predicted MV candidate will be used to find the most matched lost block from adapted multiple previous frames to conceal the lost block.

1. Motion Vector Prediction and Smoothness Criterion Definition

If the motion between spatially neighboring areas are highly correlated, then the motion of the lost block can be predicted from a spatial neighborhood block's motion vector. Figure 7(a) shows four types of candidate selection of the predicted motion vector for the lost block. Fig. 7(b) illustrates the prediction of the motion vector (MV^{pred}) in Fig. 3(b). From the neighborhood of the lost block, the motion vectors of the left 16×16 block, the upper 8×8 (the left bottom block of A in Fig. 7(b)), and the lower 16×16 block are the MV^{pred} candidates of the left 8×16 block in the lost block. (Similarly, the MV^{pred} candidates of the right 8×16 block are the MVs of the right block D, the lower block C, and the right bottom block of A).

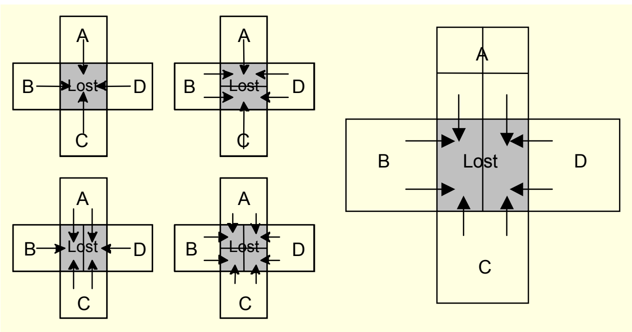

Fig. 7. Motion vector candidate selection.

Based on the MV^{pred}, a smoothness index in the search region is needed to find the most matched lost block. The smoothness index is defined as the minimum of the side match distortion d_{sm} as

$$d_{sm} = \sum_{j=1}^{N} \left| Y_j(MB_{in\_pre}^{MV^{pred}+SR}) - Y_j(MB_{out\_cur}) \right|, \qquad (4)$$

where $MB_{in}^{MV^{pred}+SR}$ is the location of the inside boundary pixel of the candidate block adopted from the predicted candidate MV (MV^{pred}) in the search region (SR) in previous frames. MB_{out} is the location of the outside boundary pixels of the lost block. The pixel value of the j-th location of MB_{in} or MB_{out} is $Y_j(MB_{out})$. The total number of pixels of MB_{in} or MB_{out} is N. If the candidate block has the minimum d_{sm}, the lost block is concealed by the value of the candidate block.

2. Error Concealment by Adapted Multiple Reference Frames

Conventionally, using the previous reference frame to find the impaired motion vector is popular. However, if the relation between the frame with the lost block and the reference frame is not significant or the reference frame is damaged, the error concealment using only one reference frame is not reliable. Therefore, some studies [12] proposed multiple reference frames instead of only one frame to find the MV candidate. Error concealment with multiple reference frames will improve the performance of image reconstruction; however, the computation complexity will increase considerably. In order to reduce the computational complexity, based on the analysis of a number of sequences for motion estimation of H.264 coding in [11], 76.82% of MBs need only one previous reference frame to get the optimal mode and 80% of optimal motion vectors determined by the nearest reference frame. In [12] it was also shown that if the mode 16×16 is selected, the optimal reference frame tends to be unchanged; and if smaller blocks are selected, searching more frames is helpful. Based on the above characteristics, it is known that if the predicted lost block size is 16×16, only one previous frame is needed to find the MV candidate. However, if the lost block size is not 16×16, the motion vector selected from the previous frame might not be adequate in frames with significant motion.

In order to reduce the computational complexity of MV searching for a lost block with a size other than16×16, we can obtain the value of (4) from the previous frame. If $min\ d_{sm} \leq T_d$, T_d is a threshold, the smoothness of the concealed image based on the candidate of the previous frame is acceptable. If $min\ d_{sm} > T_d$ (according to the above analysis), the side matching candidate block is selected from the previous five frames in the error concealment algorithm to decide the appropriate reference frame. However, to reduce the computational complexity of MV searching, it is not necessary to check all of the previous five frames. The following scheme (called adapted multiple reference frame selection) is intended to show the determination of the most adapted reference frame. Let d_{sm}^1 denote the minimum smoothness index from the first previous frame and d_{sm}^i denote the minimum smoothness index from the i-th previous frame. If $d_{sm}^1 < d_{sm}^2$, the first previous frame is the appropriate reference frame. Otherwise, check d_{sm}^2 and d_{sm}^3, and so on until the minimum d_{sm}^i appears; then stop the comparison. The adapted multiple reference frames selection is illustrated in Fig. 8. Finally, the adapted reference frame is obtained with minimum d_{sm}^i. For clarity, a flowchart is provided in Fig. 9 to better describe the proposed algorithm presented in sections II and III.

IV. Simulation

The proposed algorithm is implemented in MPEG-4/H.264 AVC reference software JM9.0 [13]. The sequences of "Foreman," "Stefan," and "Table" are evaluated with 300 frames under different bit rates and different packet error rates. The picture decoding type is set to I, P, P, P,···, where "I" and "P" represent the intra coding and inter coding, respectively. The frame rate is 30 frames per second. The slice-interleaving packetization mechanism is used. By adjusting the quantization parameter (QP), the videos are acquired with different bit rates.

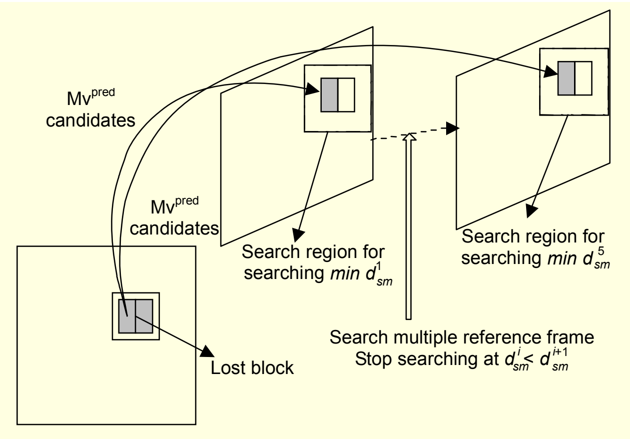

Fig. 8. The adapted multiple reference frames selection.

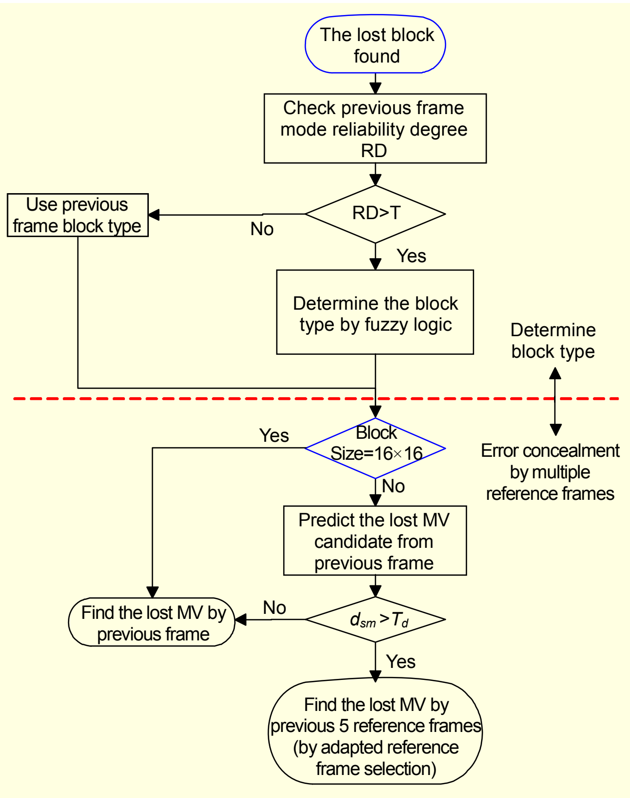

Fig. 9. Flowchart of the proposed algorithm.

Table 3 reveals the simulation results for different sequences and bit rates, with 3%, 5%, 10%, and 20% macroblock loss rates via the proposed algorithm. The algorithms used for comparison include NE (no error concealment adopted in the decoder), VM (the error concealment approaches used in H.264 [13]), RMT (refined and motion compensated temporal concealment [9]) and the proposed error concealment (error concealment of intra frame adopted bi-interpolation with neighboring pixels). The proposed algorithm is adopted in H.264; the inter frame is concealed with one reference frame with the threshold T=2 for RD, and T_d is set to 250. The results of the proposed algorithm seem to show much improvement over the others in a drastic movement sequence, such as the "Stefan" sequence. It can be seen that the proposed algorithm achieves a PSNR improvement of between 0.5 and 4.52 dB over the VM methods. As shown in Table 3, the proposed concealment scheme achieves a better performance not only with different bit rates but also with different packet error rates.

The multiple-reference-frames error concealment is evaluated in Fig. 10, which shows the PSNR comparison for the concealed "Foreman" sequence with QP=20 and 20% loss rate. Five approaches, including the error concealment adopted in H.264 (VM), the proposed fuzzy-logic-based algorithm and adapted reference frame selection (PFZ_A), the proposed fuzzy-logic-based algorithm and five reference frames (PFZ_5), and the proposed fuzzy-logic-based algorithm and one reference frame (PFZ_1), are compared in Fig. 10(a). For the average PSNR with 300 frames, there are 31.23 dB in PFZ_1, 32.38 dB in PFZ_A, and 32.71 dB in PFZ_5. PFZ_A is 1.15 dB better than PFZ_1, and PFZ_5 is only 0.33 dB better than PFZ_A. However, PFZ_A reduces computational complexity by at least 60% when compared to PFZ_5 (see Table 4). Thus, the increase in computational complexity is worth exploring. From the enlargement of frames 48-100 shown in Fig. 10(b), the performance of PFZ_A is close to that of PFZ_5. Figure 11 shows that significant errors occur in the wall edge on the 127th frame of the "Foreman" with 10% loss rate and a QP equal to 24. This is an indication of the improvement in reconstructed images by the proposed concealment schemes over other error concealment schemes. The minimum PSNR improvement resulting from the proposed algorithm is 0.5 dB and the maximum PSNR improvement is 4.52 dB. By using the proposed schemes, the image quality degradation of damaged video streams is apparently less severe.

Let the computational complexity be expressed in terms of milliseconds per processed frame (ms/f). The data shown in Table 4 are obtained by taking the average of the values for the sequences in Table 3. The simulation is performed on a PC with a 3.0 GHz CPU. The processing speed of the proposed

Table 3. The simulation results of the proposed algorithm.

Video sequence	QP	Bitrate (kbps)	Original PSNR (dB)	Method	PSNR (dB) of different packet loss rate			
					0.03	0.05	0.10	0.20
Foreman	24	32.72	38.46	NE	29.71	27.28	24.08	20.99
				VM	34.86	33.36	31.25	30.70
				RMT	37.20	33.95	32.13	31.13
				Proposed	37.32	34.36	32.45	31.46
	20	59.50	41.48	NE	29.68	23.29	24.07	20.98
				VM	37.13	36.21	32.98	30.63
				RMT	38.30	36.46	34.15	30.96
				Proposed	38.19	36.61	34.20	31.23
	16	106.96	44.76	NE	29.68	27.26	24.06	20.97
				VM	41.99	39.16	35.05	30.57
				RMT	44.28	40.78	37.08	33.02
				Proposed	44.43	41.11	37.37	33.25
Stefan	24	89.37	37.48	NE	31.31	29.01	25.97	22.91
				VM	35.09	31.96	28.53	25.04
				RMT	35.25	32.28	29.58	25.61
				Proposed	35.81	32.70	29.91	25.87
	20	139.62	40.97	NE	31.28	28.97	25.94	22.87
				VM	34.92	31.87	28.46	24.91
				RMT	35.08	32.25	29.56	25.65
				Proposed	35.83	32.72	30.02	25.92
	16	208.01	44.50	NE	31.27	28.96	25.92	22.85
				VM	34.97	31.86	29.15	24.86
				RMT	35.54	32.59	29.75	25.76
				Proposed	35.96	32.79	30.02	25.98
Table	24	27.47	38.32	NE	33.81	31.33	27.98	24.56
				VM	36.37	32.86	32.12	31.78
				RMT	34.64	33.90	32.65	32.84
				Proposed	37.51	37.17	35.15	33.43
	20	45.11	41.43	NE	34.05	31.28	27.93	24.52
				VM	36.10	33.31	32.04	31.73
				RMT	38.48	36.56	33.59	33.24
				Proposed	40.69	37.74	35.11	33.49
	16	74.01	44.85	NE	33.86	31.25	27.91	24.50
				VM	36.62	35.99	33.01	31.97
				RMT	39.04	35.49	34.55	32.87
				Proposed	41.14	37.48	36.18	33.54

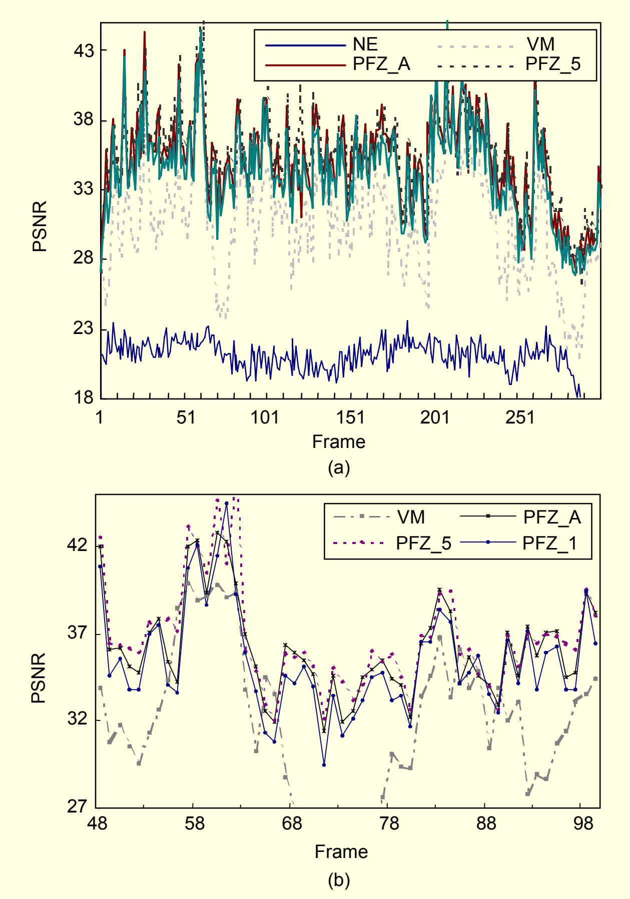

Fig. 10. The multiple-reference-frames error concealment evaluation.

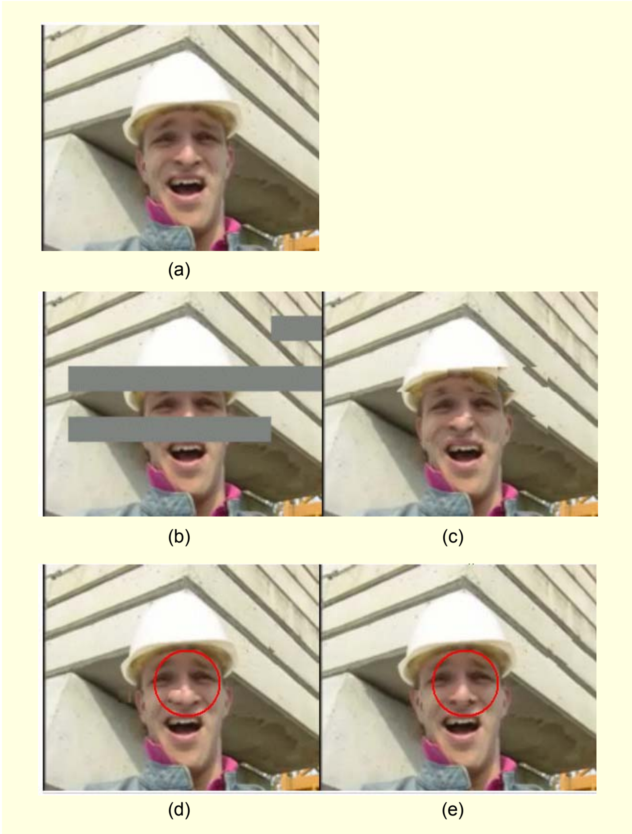

Fig. 11. Reconstructed image: (a) the original picture coded by QP=24; (b) the damaged picture with a 10% error rate. The recovered pictures by VM, RMT and the proposed algorithm are shown in (c), (d) and (e), respectively.

Table 4. The computational complexity results by a PC with a 3.0 GHz CPU for various packet error rates.

Error rate	VM (ms/f)	RMT (ms/f)	PFZ_1 (ms/f)	PFZ_A (ms/f)	PFZ_5 (ms/f)
3%	9.17	10.88	11.18	15.59	38.73
5%	9.96	11.94	12.18	17.51	41.49
10%	10.51	13.75	14.18	20.44	47.13
20%	11.90	16.18	17.85	24.14	63.74

method is only 7.96 to 12.24 ms/f slower than its competing methods, but the PSNR performance of the proposed method is about 0.5 to 4.52 dB better than the other methods. Thus, the slight increase in the computational cost is well justified.

V. Conclusion

In this paper, a new error concealment algorithm for the coding standard H.264 has been presented. This algorithm uses the fuzzy logic method to determine the size type of the lost block from the current frame, and provides an adapted multiple-reference-frames selection method to find the lost motion vector. Fuzzy logic was used to determine the most accurate mode for the lost block under edge crossing. To find the most adequate motion vector of the lost block, the adapted multiple-reference-frames selection saves at least 60% more computation time than searching all five reference frames. The experimental simulation results have shown that the proposed algorithm indeed improves the quality of corrupted video images, especially, for edge preservation in drastic movement sequences.

References

[1] *Information Technology—Coding of Audio-Visual Objects—Part10: Advanced Video Coding*, Final Draft International Standard, ISO/IECFDIS 14496-10, Dec. 2003.

[2] Report of the Formal Verification Tests on AVC, Dec. 2003. ISO/IEC14496-10 ITU-T Rec. H.264 MPEG2003/N6231.

[3] T. Stockhammer, M.M. Hannuksela, and T. Wiegand, "H.264/AVC in Wireless Environments," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, July

2003, pp. 657-673.
[4] S. Wenger, "H.264/AVC over IP," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, July 2003, pp. 645-656.
[5] Y. Wang and Q.F. Zhu, "Error Control and Concealment for Video Communication: A Review," *Proc. of the IEEE*, vol. 86, May 1998, pp. 974-997.
[6] C.T. Hsu, M.J. Chen, W.W. Liao, and S.I. Lo, "High Performance Spatial and Temporal Error Concealment Algorithms for Block Based Video Coding Techniques," *Electronics and Telecommunication Research Institute Journal*, vol. 27, no. 1, Feb. 2005, pp.53-63.
[7] M.J. Chen, L.G. Chen, and R.M. Weng, "Error Concealment of Lost Motion Estimation Vectors with Overlapped Motion Compensation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 3, June 1997, pp.560-563.
[8] Y. Wang, M.M. Hannuksela, and V. Varsa, "The Error Concealment Feature in the H.26L Test Model," *Proc. of the Int. Conf. Image Processing*, vol. 2, Sep. 2002, pp. 729-732.
[9] Y. Xu and Y. Zhou, "H.264 Video Communication Based Refined Error Concealment Schemes," *IEEE Trans. Consumer Electronics,* vol. 50, no. 40, Nov. 2004, pp. 1135-1141.
[10] L.X. Wang, *A Course in Fuzzy Systems and Control*, Prentice-Hall, NJ, 1997.
[11] Y.W. Huang, B.Y. Hsieh, T.C. Wang, S.Y. Chien, S.Y. Ma, C.F. Shen, and L.G. Chen, "Analysis and Reduce of Reference Frames for Motion Estimation in MEG-4 AVC/JVT/H.264," *Proceeding of the Int. Conf. On Acoustics, Speech, & Signal Processing,* vol. 2, Sep. 2003, pp. 809-812.
[12] M. Flierl, T. Wiegand, and B. Girod, "Rate-Constrained Multihypothesis Prediction for Motion-Compensated Video Compression," *IEEE Trans. Circuits System Video Technology*, vol. 12, Nov. 2002, pp. 957-969.
[13] *Joint Video Team (JVT)* software JM9.0, http://iphome.hhi.de/suehring/tml/download/

Pei-Jun Lee received the PhD degree from National Taiwan University in electrical engineering. Since February 2006, she has been an Assistant Professor of the Department of Electrical Engineering, National Chi Nan University, Puli 545, Taiwan. From Aug. 2004 to February 2006, she was an Associate Professor of the Department of Electronic Engineering, St. John's University. Her research interests include image processing, video compression, video communication, error concealment and DSP/VLSI system application.

Ming-Long Lin received the BS degree in electronic engineering and MS degree in the Institute of Automation and Mechatronics from St. John's University, Taipei, in 2004 and 2006, respectively. His major research interests include H.264/AVC video coding, and associated system application.