

Efficient Multiple-Reference Temporal Error Concealment Algorithm Based on H.264/AVC

Chen Yu^(✉), Chao Cheng, Hai Jin, and Jingli Zhou

Services Computing Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
yuchen@hust.edu.cn

Abstract. Temporal error concealment at the video communication receiver recovers damaged blocks using temporal information redundancy. To enhance the quality of reconstructed image, a multiple-reference temporal error concealment algorithm based on H.264/AVC was proposed. Firstly, a pre-concealment is implemented to improve video frame; then the fractional pixel boundary matching algorithm and search algorithm are used to evaluate the candidate motion vectors from a number of reference frames, and the motion vector (MV) giving the minimum boundary matching distortion (MBMD) is selected as optimal motion vector; finally it is used to recover the damaged block and the video frame quality can be improved further. For various video sequences and different macro-block loss ratio, experimental results show that the proposed algorithm is significantly better than traditional temporal error concealment algorithms such as BMA and BM from both objectively and subjectively. At the loss ratio of 10 %, improvements of the proposed algorithm are more than 0.8 ~ 3.7 dB compared to that of other algorithms.

Keywords: Temporal domain · Multiple-reference · Fractional pixel · Boundary matching · Search algorithm

1 Introduction

With the rapid development of multimedia and mobile communication technology, the application of video communication has become an inevitable trend. However, in wireless mobile channels, code error and data loss are always difficult to avoid, and compressed video data is very sensitive to the code error, so a small amount of errors may lead to a large number of codes' incorrect decoding. To control and reduce the effect of transmission errors, error concealment technology is used (H.264/AVC) to maximize the recovery of the damaged blocks [1, 2].

Error concealment technique at the video receiver recovers damaged blocks using temporal and spatial information redundancy in video transmission signal. When error occurs during the transmission, receiver uses the correct decoding information to improve image quality via reconstructed damaged block. Common error concealment algorithms are divided into temporal and spatial error concealment algorithms [3]. The former makes use of temporal information redundancy to estimated MV of the

damaged block and this MV is used to conceal the damaged block. The latter recovers damaged block using its neighboring blocks.

The critical step of temporal error concealment is to correctly estimate MV of the damaged block. The commonly used technique is to replace the damaged motion vector with (0, 0) [4]. This technique is referred to as temporal replacement (TR). Another technique is to replace the damaged motion vector with the median of neighboring vectors [5]. To choose the proper MV among candidate vectors, a boundary matching algorithm (BMA) has been suggested in [6]. A concealment algorithm based on block matching (BM) is proposed in [7].

In H.264/AVC video standard, a more precise and flexible inter-frame prediction method is introduced [8]. E.g., multiple-reference takes the place of single-reference, accuracy prediction of 1/4 pixel and 1/8 pixel replace that of integer pixel, 4×4 integer transform replaces discrete cosine transform (DCT), etc. In [9], a multiple-reference concealment algorithm based on motion field interpolation (MFI) is suggested. In this method, all the pixels in the damaged block are interpolated one by one, but calculation of each pixel need to know the MV of all the neighboring blocks and it may causes long delay, so this method is not suitable for real-time mobile channel.

According to the new features of H.264/AVC, a multiple-reference temporal error concealment algorithm is proposed. Firstly, a pre-concealment is implemented to improve video frame; then the weighted boundary matching algorithm based on fractional pixel and search algorithm are used to evaluate the candidate motion vectors from a number of reference frames (In this paper, reference frame number is 5), and the candidate MV giving the minimum boundary matching distortion is selected as the optimal MV; finally this MV is used to recover the damaged block and the video frame quality can be improved further.

Part II discusses the multiple-reference temporal error concealment algorithm. Part III presents the experiments and discussion and the conclusions are given at last.

2 Multiple-Reference Temporal Concealment

2.1 Pre-concealment of the Damaged Macro-Block

As described previously, the BMA, BM and MFI algorithms respectively mentioned in [6, 7, 9] only use the neighboring available blocks to do boundary matching, and the concealed blocks are used to provide candidate concealments. The disadvantage is that it doesn't make full use of the information of neighboring blocks, and there is likely to cause big boundary matching distortion and unsmooth matching between the final motion compensated block (MCB) and the available blocks. To resolve the problem, a pre-concealment method is implemented to improve video frame quality and then the improved boundary matching algorithm is used to evaluate candidate MVs of the damaged block.

The idea of pre-concealment is to do simple motion vector estimation and motion compensation for every damaged block. The current damaged block is denoted by MB_{current} and the block in previous frame that has the same position with MB_{current} is denoted by MB_{ahead} . MB_{up} and MB_{down} denote the blocks that are above and under

$MB_{current}$. (1) If the movement of MV of MB_{ahead} is not more than 8 pixels at x and y direction, the estimated MV of $MB_{current}$ is assigned to that of MB_{ahead} . (2) Otherwise, if MB_{up} is available and isn't boundary block of the current frame, the estimated MV of $MB_{current}$ is assigned to that of MB_{up} . (3) Otherwise, if MB_{down} is available and isn't boundary block, the estimated MV of $MB_{current}$ is assigned to that of MB_{down} . (4) Otherwise, the MV of $MB_{current}$ is assigned to that of MB_{ahead} .

2.2 Set of Candidate Motion Vectors

In this paper, as shown in Fig. 1, the coding modes of the neighboring vertical and horizontal blocks.

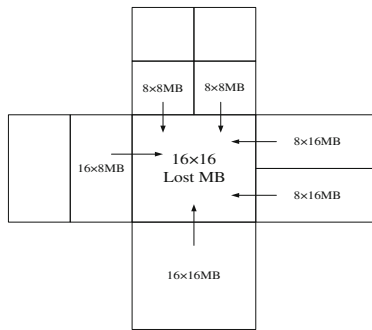


Fig. 1. Candidate motion vectors

The set of candidate motion vectors can be obtained by calculation, which is denoted by C . The motion vectors of the vertical and horizontal blocks, e.g.: the motion vectors of 8×8 , 16×16 , 16×8 and 8×16 coding mode as shown in Fig. 1.

- Zero motion vector [4].
- Mean motion vector [5], that is:

$$\begin{cases} mv_x = \frac{1}{N} \sum_{i=0}^{N-1} mv_x^i \\ mv_y = \frac{1}{N} \sum_{i=0}^{N-1} mv_y^i \end{cases} \quad (1)$$

- Median motion vector [5], that is:

$$\begin{cases} mv_x = \text{Median}(mv_x^i, i = 0, 1, \dots, N-1) \\ mv_y = \text{Median}(mv_y^i, i = 0, 1, \dots, N-1) \end{cases} \quad (2)$$

Among them, mv_x^i and mv_y^i are the coordinates x and y of the motion vectors, and N is the total number of the motion vectors of the available blocks.

2.3 1/4 Pixel Interpolation

The accuracy of the MV in H.264 is 1/4 pixel. In order to make full use of it, 1/4 pixel interpolation can be used for the reference frames. The rules of interpolation methods and H.264 standard decoding process are same [8], that is, half-pixel interpolation of the luminance signal uses six-order FIR filter and quarter-pixel interpolation of the luminance signal uses linear interpolation. Because the change of color difference signal is lower than that of luminance signal and humans' eyes are not sensitive on the color difference signal, we only use luminance signal for boundary matching. To reduce the complexity of the algorithm, the color difference signal doesn't use 1/4 pixel interpolation. It can be directly concealed from the reference frame that hasn't been interpolated by using the motion vectors of integer-pixel accuracy.

2.4 Proposed Boundary Matching Algorithm

When there exist edges or spatial sampling ratio is not high enough to cope with the rapid gray level change in the original image, the boundary matching algorithm (BMA) [6] and side matching algorithm (SMA) don't work properly when pixel values change abruptly. To overcome the problem, we proposed an improved, robust and efficient boundary matching algorithm.

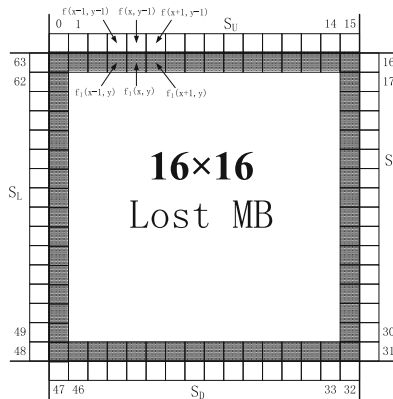


Fig. 2. Optimized boundary matching algorithm

The four neighboring blocks of damaged 16×16 block, including available blocks, concealed blocks and pre-concealed blocks, have 64 boundary integer-pixels. According to a clockwise sequence, assume that their ID are 0, 1, ..., 63 from the upper-left corner. As shown in Fig. 2, the variations between the current image block

and the one above it is denoted by S_U , where $f_1(x - 1, y)$, $f_1(x, y)$ and $f_1(x + 1, y)$ are the boundary integer pixel value of inside and $f(x - 1, y - 1)$, $f(x, y - 1)$ and $f(x + 1, y + 1)$ are the boundary integer pixel values of outside. S_U is defined as:

$$S_U = \sum_{i=0}^{15} \min\{Diff_1^i, Diff_2^i, Diff_3^i, Diff_4^i, Diff_5^i, Diff_6^i, Diff_7^i\} \tag{3}$$

In Eq. (3), $Diff_1$, $Diff_2$ and $Diff_3$ are the difference between inside and outside boundary integer pixel values and they are respectively defined as:

$$\begin{cases} Diff_1 = |f_1(x, y) - f(x - 1, y - 1)| \\ Diff_2 = |f_1(x, y) - f(x, y - 1)| \\ Diff_3 = |f_1(x, y) - f(x + 1, y - 1)| \end{cases} \tag{4}$$

$Diff_4$, $Diff_5$, $Diff_6$ and $Diff_7$ are the difference between the integer pixel value and sub-pixel value or the integer pixel value and quarter-pixel value. They are respectively defined as:

$$\begin{cases} Diff_4 = |f_1(x, y) - f_{\frac{1}{2}}(x - 1, y)| \\ Diff_5 = |f_1(x, y) - f_{\frac{1}{2}}(x + 1, y)| \\ Diff_6 = |f_1(x, y) - f_{\frac{1}{4}}(x - 1, y)| \\ Diff_7 = |f_1(x, y) - f_{\frac{1}{4}}(x + 1, y)| \end{cases} \tag{5}$$

In Eq. (5), $f_{1/2}(x-1,y)$ and $f_{1/2}(x + 1,y)$ are sub-pixel value and $f_{1/4}(x-1,y)$ and $f_{1/4}(x + 1,y)$ are the quarter-pixel value, they can be calculated by Eq. (6).

$$\begin{cases} f_{\frac{1}{2}}(x - 1, y) = \frac{1}{4} [f(x - 1, y - 1) + f(x, y - 1) \\ \quad + f_1(x - 1, y) + f_1(x, y)] \\ f_{\frac{1}{2}}(x + 1, y) = \frac{1}{4} [f(x + 1, y - 1) + f(x, y - 1) \\ \quad + f_1(x + 1, y) + f_1(x, y)] \\ f_{\frac{1}{4}}(x - 1, y) = \frac{1}{8} [3f(x - 1, y - 1) + 3f(x, y - 1) \\ \quad + f_1(x - 1, y) + f_1(x, y)] \\ f_{\frac{1}{4}}(x + 1, y) = \frac{1}{8} [3f(x + 1, y - 1) + 3f(x, y - 1) \\ \quad + f_1(x + 1, y) + f_1(x, y)] \end{cases} \tag{6}$$

Similarly, the variations between the current block and its left and right blocks and the one below it are denoted by S_D, S_L and S_R (as shown in Fig. 2), which are defined as:

$$S_R = \sum_{i=16}^{31} \min\{Diff_1^i, Diff_2^i, Diff_3^i, Diff_4^i, Diff_5^i, Diff_6^i, Diff_7^i\} \quad (7)$$

$$S_D = \sum_{i=32}^{47} \min\{Diff_1^i, Diff_2^i, Diff_3^i, Diff_4^i, Diff_5^i, Diff_6^i, Diff_7^i\} \quad (8)$$

$$S_L = \sum_{i=48}^{63} \min\{Diff_1^i, Diff_2^i, Diff_3^i, Diff_4^i, Diff_5^i, Diff_6^i, Diff_7^i\} \quad (9)$$

To make full use of the correctness of the neighboring blocks, a weighted method is introduced in boundary matching. The total variation SAD is defined as:

$$SAD = W_U S_U + W_D S_D + W_L S_L + W_R S_R \quad (10)$$

Among them, W_U , W_D , W_L and W_R are the weighted values of the boundary matching variations. According to status of the neighboring blocks, W_U , W_D , W_L and W_R can be assigned to one of the three values, which are 1, 1/4 and 1/8. (1) If it comes from correctly decoded information block, chooses 1, (2) If the concealed block, chooses 1/4; (3) If the pre-concealed block (Sect. 2.1), chooses 1/8.

2.5 Proposed Search Algorithm

To find the MV that makes SAD obtain the minimum value in a number of reference frames more accurately and quickly, we need to set up the search range of boundary matching and take advantage of suitable search algorithm. Under ordinary conditions, search range is selected as rectangular window and full search (FS) algorithm mentioned in is used to give the global optimum solution to the motion estimation. In order to reduce the search time further, the cross search and diamond search algorithms are respectively proposed.

To improve the efficiency of searching motion vector, an efficient cross-diamond search algorithm for fast block matching motion estimation based on multiple-reference is presented in this paper. As shown in Fig. 3, four kinds of search patterns such as cross search, small diamond search, horizontal diamond search and vertical diamond search need to be used in the proposed algorithm.

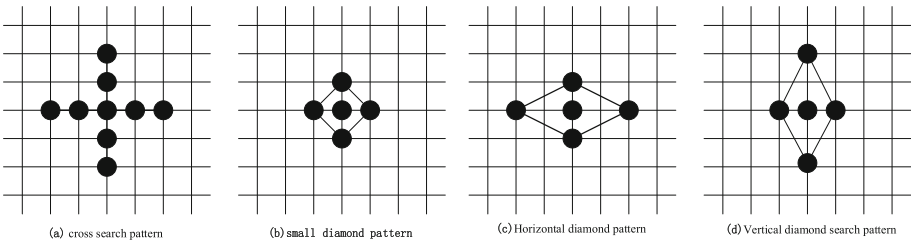


Fig. 3. Four kinds of search patterns

Assume that the set of candidate motion vectors $C = \{mv_1, mv_2, \dots, mv_n\}$, and $mv_i = (mv_x^i, mv_y^i)$, $1 < i < n$, and the upper-left coordinate of the 16×16 damaged block is (x_0, y_0) .

Step1. For each (mv_x^i, mv_y^i) of set C , the original coordinate (x_0, y_0) changes into $(4x_0 + mv_x^i, 4y_0 + mv_y^i)$ after 1/4 pixel interpolation and take it as the upper-left coordinate, we set up 16×16 block which has the same size with the damaged block. The total variations SAD between this block and the neighboring blocks of the damaged block can be work out by Eq. (10) and take the MV which makes SAD obtain the minimum value in the set of candidate motion vectors C as the candidate optimal MV of every reference frame, denoted by MV_C .

Step2. Take the pixel point of MV_C as the center point (origin of coordinate $(0, 0)$) and test it and its 8 nearby points which are combined as a cross (cross search pattern). Calculating boundary matching distortion by using Eq. (10), as shown in Fig. 4(a), if the minimum boundary matching distortion (MBMD) point is located at the center position, MV_C is selected as the candidate optimal MV of this current reference frame, and go to end; otherwise go to step3.

Step3. Test the two points selected from the set $\{(1, 1), (1, -1), (-1, 1), (-1, -1)\}$ and have the shortest distance with MBMD point. Then compare the two points with the MBMD point which generated in the previous step, as shown in Fig. 4(b), if the new MBMD point of the three is located at one of the coordinates of $\{(0, 1), (0, -1), (1, 0), (-1, 0)\}$, take the MV of MBMD point as the candidate optimal MV of this current reference frame, and go to end; otherwise go to step 4.

Step4. Take the two points that test in previous step and another three points on the cross to compose a small diamond. If the MBMD point of the five is in the horizontal direction, horizontal diamond pattern is selected to search the new MBMD points; if the MBMD point of the five is in the vertical direction, the vertical diamond pattern is selected. Then we judge whether the new MBMD point is the center point, if it is, go to step 6, otherwise go to step 5.

Step5. Take the new MBMD point as the center point to search optimal point, if it is in the horizontal direction of diamond in previous step, horizontal diamond pattern is selected, otherwise vertical diamond pattern is selected. In this step, if the new MBMD point is the center point go to step 6, recursively repeat this step until MBMD point located in the center position.

Step6. Take the MBMD point which generated in step4 or step5 as the center point to do search by using small diamond pattern, and the point has the minimum boundary matching distortion is selected as the candidate optimal MV of the current frame and the search process is end. Figure 4(c) shows the whole search process of the cross-diamond search algorithm.

For the 5 reference frames, the candidate MV of every reference frame can be found by above-mentioned algorithm, and its motion compensated block is denoted by B_k ($1 \leq k \leq 5$). In these blocks, the one that makes SAD obtain the minimum value is used to recover the damaged block. $B' = \arg \min\{SAD(B_k)\}$

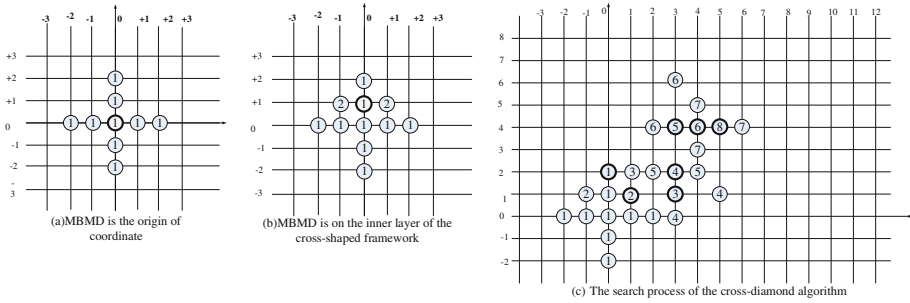


Fig. 4. The search process of the cross-diamond search algorithm.

2.6 The Concealment of Damaged Macro-Block

From the analysis above, the MV that corresponds to the block B' is the optimal MV in the whole search process and let (m_x, m_y) denote it. Assume that $Y_{M,FP}(x, y)$ is the Y component value at coordinates (x, y) after 1/4 pixel interpolation. $U_M(x, y)$ and $V_M(x, y)$ are the U and V component values at coordinate (x, y) when reference frame is not interpolated, so the concealed Y, U and V component values of the damaged block are as follows:

$$\begin{cases} Y(x, y) = Y_{M,FP}(4x + m_x, 4y + m_y) \\ U(x, y) = U_M(x + \frac{m_x}{4}, y + \frac{m_y}{4}) \\ V(x, y) = V_M(x + \frac{m_x}{4}, y + \frac{m_y}{4}) \end{cases} \quad (11)$$

In Eq. (11), x and y are in the range of $x_0 \leq x < x_0 + 16, y_0 \leq y < y_0 + 16$.

3 Simulation Results and Discussions

In the experiment, H.264/AVC standard test model JM15.0 was used to simulate the situation of packet loss of the video compression code streaming in the network transmission. In the same packet loss ratio, three temporal error concealment algorithms which are the proposed multiple-reference temporal error concealment (MTEC), boundary matching-based algorithm (BMA) and error concealment based on block matching (BM) had been compared on different sequences to show the effects.

Five different QCIF (176×144) coding sequences, each of 100 frames, were used: FOREMAN, MOBILE, MISS, CLAIRE and COASTGUARD. The flexible macro-block order (FMO) technique in H.264/AVC standard was used that the consecutive macro-block loss caused by packet loss was converted to dispersive macro-block loss. Each test sequence used IPPP frame format to encode and the macro-blocks in I-frame were not lost and the loss ratio of P-frames were as follows: 2 %, 5 % and 10 %. The loss ratio of 2 % means that 2 % of macro-blocks were damaged per frame. Quantization parameter (QP) is 28. We calculate the average peak signal to noise ratio (PSNR) values after damaged frames in the sequences had been concealed. We take the average value of the experimental results in 30 times as the ultimate objective evaluation criteria.

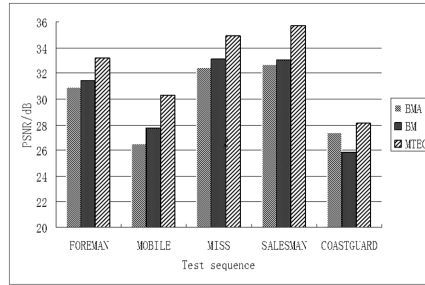


Fig. 5. The comparison of the average PSNR of three algorithms

Table 1 shows the average PSNR values of five different video sequences at the loss ratio of 2 %, 5 % and 10 %. And Fig. 5 shows the comparison of the average PSNR of three algorithms at the loss ratio 10 %. The performance of the boundary matching-based algorithm (BMA) and block matching algorithm (BM) are not as good as the proposed multiple-reference temporal error concealment algorithm (MTEC) for the damaged P-frame in different sequences. The proposed algorithm has better adaptability and it can recover the video information more clearly. For the sequence of simple scene and texture, such as MISS, the concealed PSNR of ASEC are 0.73 ~ 3.02 dB and 0.59 ~ 2.24 dB larger than that of BMA and BM respectively; for the sequence of more complex exercises as well as more details, such as MOBILE, the proposed algorithm has better effect of concealment. At the loss ratio of 10 %, compared with BMA, improvements are 1.22 ~ 3.75 dB and compared with BM the values are 0.34 ~ 2.53 dB.

Table 1. Simulation results for different macroblock loss ratio

Video sequences	Algorithms	PSNR/dB		
		2 %	5 %	10 %
FOREMAN	BMA	35.24	34.76	31.93
	BM	36.12	34.58	31.47
	MTEC	36.46	35.32	33.19
MOBILE	BMA	31.14	29.27	26.51
	BM	32.27	30.49	27.73
	MTEC	33.36	31.38	30.26
MISS	BMA	35.58	33.06	32.48
	BM	36.83	35.23	33.12
	MTEC	37.79	36.57	34.94
SALESMAN	BMA	39.56	37.85	32.69
	BM	40.14	38.42	33.03
	MTEC	40.73	39.27	35.71
COASTGUARD	BMA	32.67	29.74	27.35
	BM	32.42	28.22	25.84
	MTEC	33.59	30.56	28.08

For SALESMAN and MOBILE sequence, Figs. 6 and 7 depict PSNR of each concealed frame respectively when macro-block loss ratio of 5 % occurs. The concealed results of the three algorithms have been compared in the front 40 damaged P-frames. I-frame had been inserted in every 20 P-frames and it didn't have packet loss. The P-frames that are close to the I-frame have lower effects and they have higher PSNR values after concealment. The farther distance damaged P-frame is away from I-frame, the greater effects error proliferation causes, the lower PSNR values after concealment. As the number of the frames increases, the proposed algorithm can obtain better results of concealment.

Compared with SALESMAN sequence, MOBILE sequence has more complex exercises and intense scenes. And when the damaged P-frames are concealed by BMA, BM and the proposed MTEC algorithms respectively in different sequences. As shown in Figs. 6 and 7, for SALESMAN sequence, the downward trends of three algorithms are almost the same; and for MOBILE sequence, the decreasing rate of BMA and BM are larger than that of MTEC.

For MISS sequence, Fig. 8 compares the subjective quality of concealed image in the forty-second frame by using BMA, BM and MTEC algorithms respectively when macro-block loss ratio of 10 % occurs. As show in the Fig. 8, the proposed algorithm shows substantial improvement over the existing methods and the concealment of the subjective image quality is closest to the original image, but the quality of the three image are clear and the effect are not obvious.

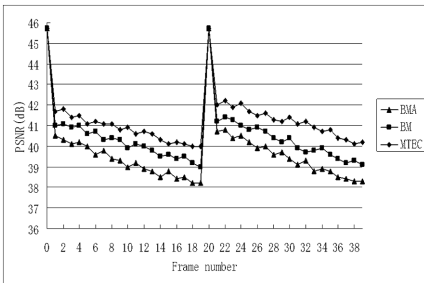


Fig. 6. PSNR of three temporal concealment algorithms for SALESMAN when macro-block loss ratio of 5 % occurs

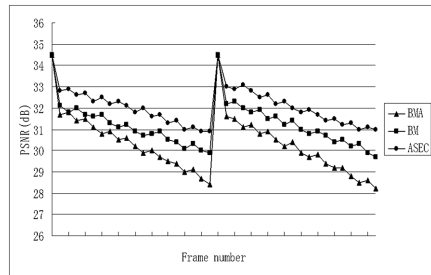


Fig. 7. PSNR of three temporal concealment algorithms for MOBILE when macro-block loss ratio of 5 % occurs

For MOBILE sequence, Fig. 9 compares the subjective quality of concealed image in the eighth frame by using BMA, BM and MTEC algorithms respectively when macro-block loss ratio of 10 % occurs. MOBILE sequence has the most complicated scenes and intense exercises, and it contains more rich information. In the Fig. 9, the difference of subjective quality of the three concealed image in MOBILE sequence is more obvious than that of MISS sequence. In MOBILE and MISS sequence, the PSNR of MTEC is 3.53 dB and 2.56 dB larger than that of BMA, while the PSNR of MTEC is 4.18 dB and 1.69 dB larger than that of BM. The proposed algorithm shows

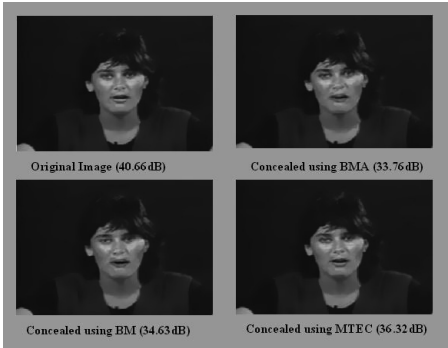


Fig. 8. The comparison of subjective quality of concealed image for MISS when macro-block loss ratio of 10 % occurs

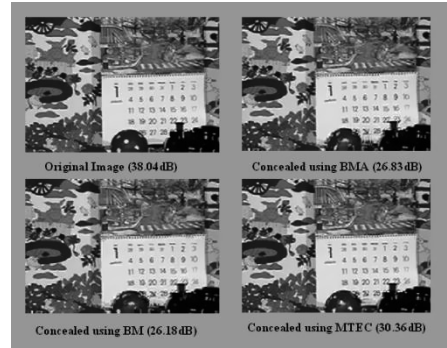


Fig. 9. The comparison of subjective quality of concealed image for MOBILE when macro-block loss ratio of 10 % occurs

improvement over the existing methods and the concealment of the subjective image quality is closest to the original image.

Table 2 shows the average decoding time of five different video sequences at the loss ratio of 10 % and these test sequences are respectively 50, 100 and 200 frames. Figure 10 shows the comparison of the decoding time when the length of test sequence is 100 frames. When the frame number is 50, the decoding time of MTEC is little longer than that of BMA and BM, while in some special cases such as MISS sequence of 50 frames, MTEC has the least decoding time; when the frame number increases to 100 and 200, the decoding time of MTEC increases more slowly and it is 0.4~0.9 s longer than the decoding time of BMA and BM.

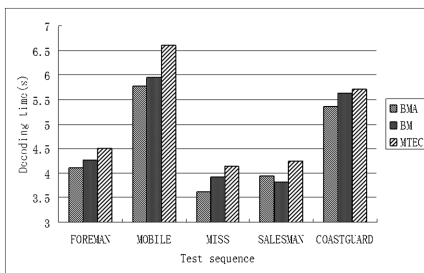


Fig. 10. The comparison of the decoding time of three algorithms

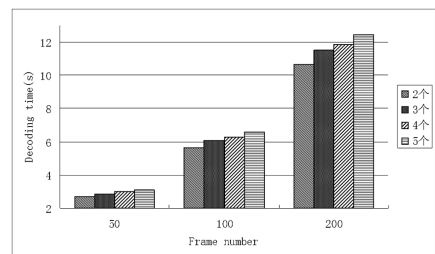


Fig. 11. The average decoding time of MTEC when using different reference number

Table 3 shows the average decoding time of two different video sequences such as MOBILE and SALESMAN when using different reference number of MTEC at the loss ratio of 10 % and these test sequences are respectively 50, 100 and 200 frames. Figure 11 shows the average decoding time of MTEC when using different reference

Table 2. Decoding time of different temporal error concealment algorithms

Video sequence	Frame number	Decoding time/s		
		BMA	BM	MTEC
FOREMAN	50	1.825	1.839	1.901
	100	4.105	4.276	4.508
	200	8.547	8.396	9.204
MOBILE	50	2.845	2.937	3.109
	100	5.769	5.942	6.593
	200	11.566	11.935	12.463
MISS	50	1.809	1.876	1.781
	100	3.616	3.924	4.138
	200	7.428	7.873	8.069
SALESMAN	50	1.792	1.674	1.930
	100	3.939	3.817	4.252
	200	7.861	7.529	8.538
COASTGUARD	50	2.562	2.746	2.837
	100	5.348	5.623	5.714
	200	10.886	11.430	11.791

Table 3. Decoding time of MTEC algorithm when using different reference number

Video sequence	Frame number	Decoding time/s		
		2	3	4
MOBILE	50	2.684	2.860	2.985
	100	5.651	6.085	6.263
	200	10.682	11.524	11.839
SALESMAN	50	1.581	1.758	1.841
	100	3.594	3.774	4.048
	200	7.146	7.872	8.094

number. It can be seen that when the reference number decreases to 4, the decoding time can reduce 4 % ~ 5 % compared to that of the 5 reference frame; when decreases to 3, the decoding time can reduce 7 % ~ 9 %; when decreases to 2, the decoding time can reduce 14 % ~ 16 % and in this case, the decoding time of MTEC is less than that of BM and BMA in Table 2, but the cost is the PSNR of these concealed image are less than that of 5 reference number.

4 Conclusion

A temporal error concealment algorithm based on multiple-reference was proposed in this paper. The proposed algorithm uses H.264/AVC new features: multiple-reference frames, 1/4 pixel interpolation, etc. In addition, it optimized the boundary matching

algorithm and search method. The proposed algorithm can obtain better image quality. The experimental results show that it has strong adaptability and is significantly better than traditional H.264/AVC algorithms like BMA, SMA and BM from both objectively and subjectively.

Acknowledgement. The work is partly supported by NSFC (No. 61472149), the Fundamental Research Funds for the Central Universities (2015QN67) and the National 863 Hi-Tech Research and Development Program under grant (2015AA01A203).

References

1. Suh, J.W., Ho, Y.S.: Error concealment techniques for digital TV. *IEEE Trans. Broadcast.* **48**(4), 299–306 (2002)
2. Wiegand, T., Sullivan, G.J., Bjontegaard, G., et al.: Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 560–576 (2003)
3. Wang, Y., Zhu, Q.F.: Error control and concealment for video communication: a review. *Proc. IEEE* **86**(3), 974–997 (2003)
4. Aign, S., Fazel, K.: Temporal and spatial error concealment technique for hierarchical MPEG-2 video codec. *Proc. IEEE Int. Conf. Commun.* **3**, 1778–1783 (1995)
5. Haskell, P., Messerschmitt, D.: Resynchronization of motion compensated video affected by ATM cell loss. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 545–548 (2003)
6. Valente, T.S., Dufour, C., Groliere, F., Snook, D.: An efficient error concealment implementation for MPEG4 video streams. *IEEE Trans. Consum. Electron.* **47**(3), 568–578 (2001)
7. Kim, D., Yang, S., Jeong, J.: A new temporal error concealment method for H.264 using adaptive block sizes. In: *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. III-928–III-931 (2005)
8. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496–10 AVC), Geneva, Switzerland, May 2003
9. Chen, B.N., Lin, Y.: Temporal error concealment using selective motion field interpolation. *Electron. Lett.* **42**(24), 1390–1391 (2006)