

An object-based error concealment technique for H.264 coded video

Shih-Hsuan Yang · Chi-Wen Chang · Chih-Chieh Chan

Received: 29 January 2014 / Revised: 8 July 2014 / Accepted: 18 July 2014 /

Published online: 31 July 2014

© Springer Science+Business Media New York 2014

Abstract The H.264 video coding standard has become popular owing to its excellent compression efficiency. However, the H.264-coded video is very vulnerable to data loss. Conventional error concealment techniques interpolate the lost data in units of rectangular blocks, which limit the performance because a visual object is not equivalent to an image block. In this paper, we propose a new error concealment technique that uses visual objects as concealment units. The H.264 error-resilience tool FMO (Flexible Macroblock Ordering) is also incorporated at the encoder side for utilizing the spatial correlation. A lost region is concealed at the decoder side in three steps, namely object segmentation, object matching, and region-based patching. Objects are formed in the reference pictures based on color similarity, and adjacent objects of small area or the same motion are grouped as a unity. Motion estimation is performed on detected objects to find the associated motion vector. A lost region is concealed by the object in the reference picture with the best boundary-matching score. The proposed method provides considerably higher PSNR (peak signal-to-noise ratio) than conventional block-based approaches, especially for traditionally difficult cases and high-quality videos.

Keywords Error concealment · Motion estimation · Object segmentation · H.264/AVC

1 Introduction

The H.264 standard, also known as MPEG-4 AVC (Advanced Video Coding) [1, 6, 7], has been widely employed in digital TV, mobile video, video streaming, and Blu-ray discs. H.264 produces video of high visual quality with much fewer bits, as compared to previous video-coding standards such as MPEG-2. The highly compressed H.264 video, however, is very sensitive to data loss because of the extensive use of predictive coding and variable-length codes. Various error protection mechanisms have been proposed to alleviate degradation of decoded video. For example, forward error correction or error detection with retransmission may be implemented in the network transport layer. Error resilience tools, such as inserting

S.-H. Yang (✉) · C.-W. Chang · C.-C. Chan
Department of Computer Science and Information Engineering, National Taipei University of Technology,
1, Sec. 3, Zhong-Xiao E. Rd., Taipei, Taiwan 10608
e-mail: shyang@ntut.edu.tw

resynchronization markers, data partitioning, reversible variable-length coding, and insertion of intra-blocks or intra-frames, may be used at the encoder to confine the damage of data impairment. In conjunction with appropriate error concealment at the decoder, these error-resilience tools may improve the performance of the overall system. Error concealment relieves the visual degradation by interpolating the lost or erroneous samples from spatially or temporally correlated samples. Spatial error concealment estimates a pixel of a lost block as a weighted average of correctly received neighboring pixels. However, spatial error concealment suffers from blurring and artifacts. In contrast, temporal error concealment estimates the motion vector (MV) of a lost region from correlated blocks and restores the lost region by motion compensation.

The H.264 standard adopts frame-based coding to achieve high compression efficiency unlike the MPEG-4 Part 2 [4] that inherently adopts the object-based coding where the basic coding unit is an arbitrary-shape object. In the current JM (Joint Model) reference software of H.264 [7], the motion vectors surrounding the lost macroblock (MB) and the zero MV are collected as the candidate MVs. The missing MB is restored by the MV with the smallest boundary-matching cost. This error concealment method fails to give satisfactory performance when no reliable MV exists in the candidate set, or the matching criterion is inappropriate, or multiple objects with different motion coexist in a lost macroblock. Many possible improvements such as more accurate MV estimation [12], better interpolation algorithm [9], classification of motion regions [5], and finer block classifications [8, 3], have been proposed in the literature. All of these approaches notably use rectangular blocks as basic units for concealment, following the basic coding structure of H.264. Furthermore, no error-resilience techniques are explicitly employed in these methods.

The H.264 standard includes some new application-layer error-resilience tools. One such tool is the FMO (Flexible Macroblock Ordering), which reorganizes the image blocks in a prioritized manner for better exploiting spatial correlation or facilitating unequal error protection. In this paper, we activate FMO at the encoding stage for object matching. At the decoding side, the restriction on rectangular blocks is removed and concealment is performed on objects of arbitrary shape. Objects are first segmented in the reference frame based on color similarity. Neighboring objects of small area or with consistent motion are grouped as a whole. Motion estimation is then performed on the detected objects to find the object motion vector around a refined search range. The object motion vectors associated with a lost macroblock are collected in a candidate set along with the conventional block-based motion vectors. A lost region is concealed by the object that incurs the smallest error in boundary matching. Experimental results show that the proposed object-based method achieves superior concealment results in terms of objective PSNR evaluations.

The rest of this paper is organized as follows. In Section 2, the relevant error concealment algorithms in the literature are reviewed. The proposed object-based error concealment method is presented in Section 3. Experimental results and analyses are given in Section 4, followed by the conclusion.

2 Previous work

Temporal error concealment requires the true motion vector of objects for perfect restoration. However, it is generally difficult and complicated to realize object-based true-motion estimation. Part of the reason is that the needed information (pixels, MVs) may be deficient in data-loss situations. Block-based methods that independently match and patch a missing block are usually used instead. In the current H.264 Joint Model (JM) reference software, the MVs of top

and left blocks of the current frame, the MV of the collocated block in the previous frame, and the zero MV, are collected as the MV candidates of a missing macroblock (MB) [7]. The MV with the smallest cost in BMA (Boundary Match Algorithm) is chosen as the MV for error concealment. BMA finds the sum of absolute differences between the boundary of a block (inside pixels) and the adjacent blocks (outside pixels), as shown in Fig. 1. Zhang *et al.* [12] modified the error concealment algorithm to include more vectors in the set of MV candidates and use EBMA (external BMA) as the distortion criterion. EBMA evaluates the distortion as the sum of absolute differences between the outside pixels of a candidate block in the reference frame and the successfully received outside pixels of the current frame. A hardware-efficient modification of [12] was proposed in [9], which saved considerable computation and memory bandwidth with slight visual degradation by reducing the number of MV candidates and reusing data and computational results. In [5], a motion characteristic differentiated error concealment method based on motion field transfer was proposed. Different concealing methods are applied to different regions according to their motion characteristics with the aid of FMO used at the encoder.

In the above techniques, MBs of size 16×16 luma pixels are taken as the units for concealment. Better error concealment results can be achieved if the restriction on the single-size block interpolation can be removed. A variable-block-size error concealment technique was proposed in [8]. The MV of a missing MB is first estimated using boundary matching. The 16×16 MB will be divided into 16×8 or 8×16 blocks if such division provides smaller side-match distortion. Further division into 8×8 or smaller blocks will be performed under similar conditions. The authors also introduced a spatial-temporal boundary matching algorithm to increase the temporal coherence. In [11], a variable-block-size error concealment technique based on the coding modes was proposed. The mode (SKIP or not) and block size (16×16 , 16×8 , 8×16 , or 8×8) of the lost block to be concealed are determined from the coded modes of surrounding MBs. If the surrounding MBs are mostly of the SKIP mode or M types,

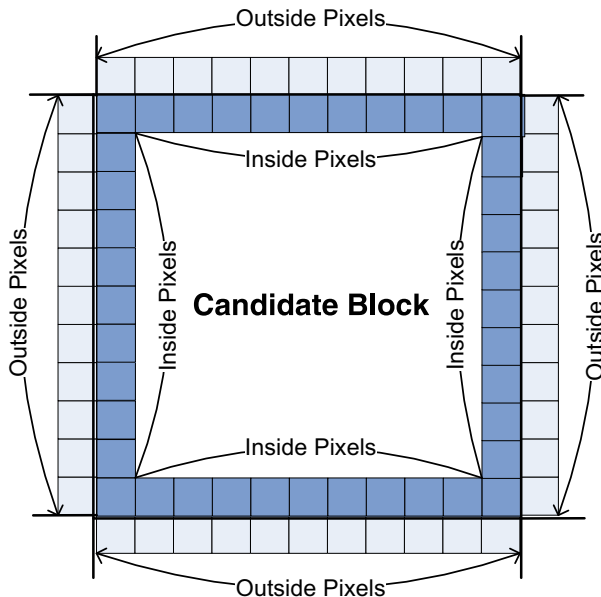


Fig. 1 BMA (Boundary Match Algorithm)

the lost MB is assigned the same type, respectively. The lost block is concealed by the motion vector that incurs the smallest EBMA cost. In [10] a hybrid motion extrapolation (HMVE) algorithm was proposed. The algorithm is hybrid in the sense that motion estimation is performed in blocks but error concealment is individually done for each pixel. Pixels in a missing block are concealed by extrapolated blocks from reference pictures assuming a constant-speed motion model. HMVE classifies pixels to be concealed into three categories, as shown in Fig. 2. Category A ($\{1,2,3,4,5,6,7,8,11,12,13\}$ in the Concealed Block 1 of Fig. 2) refers to pixels covered by at least one extrapolated 4×4 block. Category B ($\{9,10,14,15,16\}$ in the Concealed Block 1 of Fig. 2) refers to pixels not covered by any of the extrapolated blocks. The other pixels where its resident block is isolated with extrapolated blocks belong to Category C (Concealed Block 2 in Fig. 2). For Categories A and B, the dominant MV (with the largest overlapping area) and the average MV (weighted by overlapping area) are included in the set of candidate MVs. Category A also incorporates the MVs of the other overlapped extrapolated blocks in the candidate set. To remove the outliers in Category A, the MVs that are distant to the others will be discarded. The final MV used for concealment is obtained by averaging the remaining valid MVs. For Category C, the MV of the collocated pixel in the previous frame is used. It should be noted that HMVE assumes that the video was coded in frames (i.e., one slice per frame) and thus the error-resilience tools such as FMO were not employed.

3 Proposed method

The above H.264 object-based error concealment techniques [8, 2] are conducted in full frames, assuming that a frame is completely received or totally lost when an error occurs. However, better concealment results are expected if partial spatial information of the current frame can be used. In this paper, we employ the FMO tool at the encoding stage and

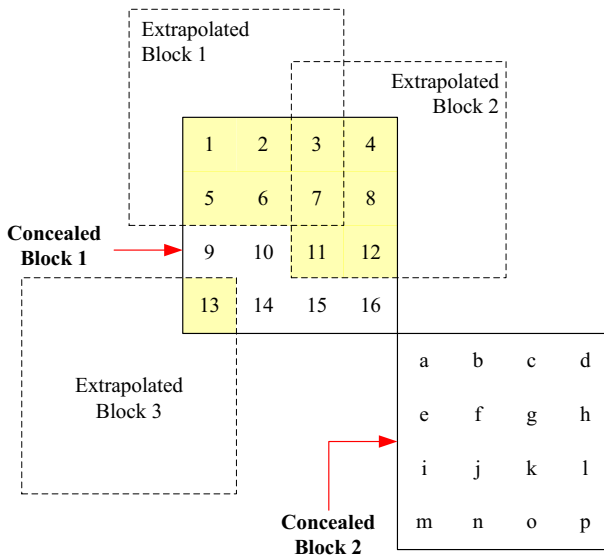


Fig. 2 Pixel classification for the HMVE algorithm [10]

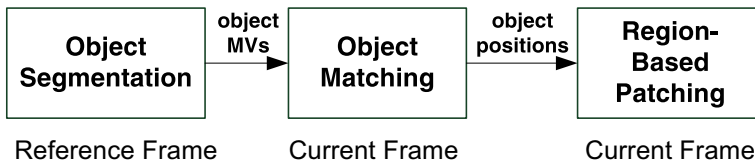


Fig. 3 Block diagram of the proposed method

consequently the spatial information from successfully received slices can be used in object matching. The block diagram of the proposed object-based error concealment technique at the decoder is shown in Fig. 3. Unlike the previous methods, the restriction on rectangular blocks for motion estimation is removed. Three major stages, object segmentation, object matching, and region-based patching, are involved in the algorithm. We explain each of them in the following subsections.

3.1 Object segmentation

When packet loss occurs, object segmentation is performed on the whole reference frame according to the color and motion consistency of pixels during decoding. Initially, the luma components (namely Y, 256 grey levels of 8 bits) of a reference frame are uniformly quantized into 8 levels (by taking the first three most significant bits). The number of quantized levels, eight, is chosen for reaching a good result of noise reduction and complexity efficiency. The initial segmentation is found by grouping connected-components of the same color. As illustrated in Fig. 4a and b, the initial segmentation may contain many small objects. These tiny objects typically amount up to 90 % of the objects found by color quantization and connected components. We join those tiny or fragile objects according to the object size and motion consistency. First, a segmented object with no more than 10 pixels is merged to its neighboring object with the most similar grey level. It is also observed that a visual object (with the same motion) such as balls may contain several color segments. Therefore, neighboring objects with the same motion are grouped as a whole. After the above merging process, those fragile objects will be properly joined, as illustrated in Fig. 4c. The motion vector of a pixel (called the pixel MV) is found as the MV of its resident block, and the MV of an object (called the object MV) is calculated as the average of its constituent pixel MVs, which will be used in the next stage. Note that for H.264-coded video the MV of a macroblock has been generated by the encoder and is then transmitted to the receiver. Although these MVs (associated with successfully received macroblocks) are calculated based on rate-distortion

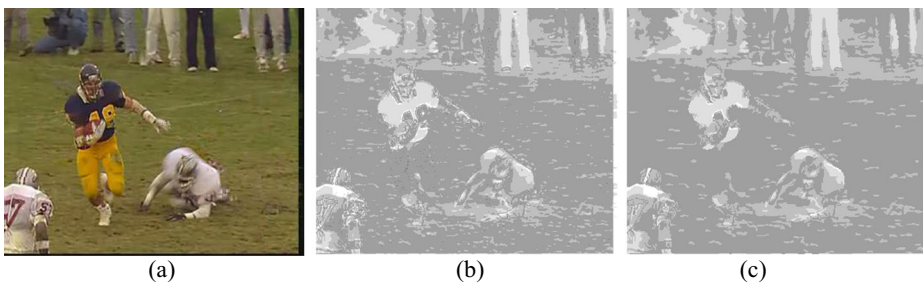


Fig. 4 a Decoded reference frame (b) initial segmentation result by color quantization and connected components (c) final object segmentation after the merging process

optimization and thus do not always represent the true motion, they are used in this paper to estimate the pixel MV and object MV for avoiding complicated true motion estimation.

3.2 Object matching

In this stage, the best match (motion vector) between the object in the reference frame and a lost region in the current frame is derived. The object MV, denoted as (d_x, d_y) , obtained in the reference frame by object segmentation is taken as the initial guess. To measure the difference between two objects, a region-based mean absolute error (MAE) is defined as follows

$$\text{MAE}(i, j) = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |\text{R}(p+m, q+n) - \text{F}(p+m+d_x+i, q+n+d_y+j)|}{\text{No. of valid pixels in the summation}} \quad (1)$$

In Eq. 1), $\text{F}(x,y)$ and $\text{R}(x,y)$ are the pixel's quantized grey level at position (x,y) of the current frame and reference frame, respectively; (p,q) is upper-left position of the bounding box of the object, and M and N are the sizes of the bounding box. Assuming that the motion of an object is relatively stable without abrupt change, we constrain the final object position in the current frame to ± 3 pixels predicted by the initial MV (i.e., $-3 \leq i, j \leq 3$ in Eq. (1)). If a matching pixel in the current frame is lost, this pixel is not counted as a valid pixel in Eq. (1). By evaluating all the MAEs in the search range, the position of an object in the current frame is found as follows

$$\text{top-left position of an object} = (p + d_x, q + d_y) + \arg \min_{-3 \leq i, j \leq 3} \text{MAE}(i, j) \quad (2)$$

These object positions will be recorded for use in the next stage.

3.3 Region-based patching

The process of the proposed object-based patching is shown in Fig. 5. We say that an object in the reference frame covers a region in the current frame if the extrapolated object obtained by object matching overlaps the region. First, the number of covering objects for a missing macroblock is counted. If one to four objects cover a missing macroblock, the proposed object-based patching will be used; otherwise the conventional block-based patching same as JM will be used. We limit the maximal number of covering objects to four because a large number of covering objects is usually the result of a lack of image features. Then for object-based patching, if a region is covered by more than one object (i.e., collision), the object with the smallest extended boundary matching score (EBMS) is used for concealment. The EBMS is calculated based on the outside bordering pixels of a block, as shown in Fig. 6. The object-based MV (obtained by object matching) will compete with the block-based MV (same as JM), and the one with a smaller EBMS will be selected as the final MV used for error concealment. For a hole region (pixels in a lost block without matching objects), the conventional block-based MV will be used.

4 Experimental results

We conducted experiments based on JM 16.2 with the settings listed in Table 1. To facilitate error resilience, videos are encoded with the Baseline profile (IPPP structure and one reference

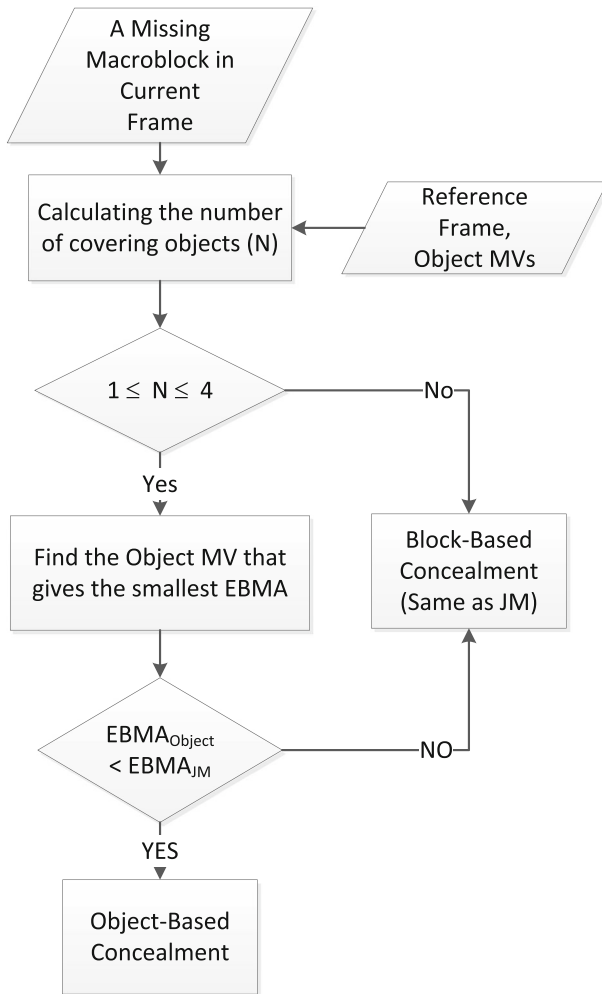


Fig. 5 Process of object-based patching

frame for P frames). The disperse FMO is activated with 6 slice groups per frame, as shown in Fig. 7. We activated the built-in fast motion estimation algorithm in the H.264 JM, namely the UMHexgonS algorithm (Unsymmetrical-cross Multi-Hexagon-grid Search) [2] to find the motion vectors of Inter MBs. Each slice is encapsulated in one packet and independently transmitted. Three packet loss rates (PLR) of 5, 10, or 15 % with independent packet losses are tested. Specifically, packet arrivals (success or loss) are regarded as independent random variables, and the loss probability is assigned as the specified packet loss rate. The random seed was determined by the current system time. All of the evaluated methods will assume the same loss pattern during a random experiment. The MV resolution for error concealment is 1/4 pixel.

The proposed method is first compared to the error concealment schemes implemented on JM 16.2 [7], and Ref. [10] (HMVE). In this experiment, one Intra frame is inserted for every 10 frames (Intra period = 10). Six standard CIF video sequences (frame rate equal to 30 Hz) of

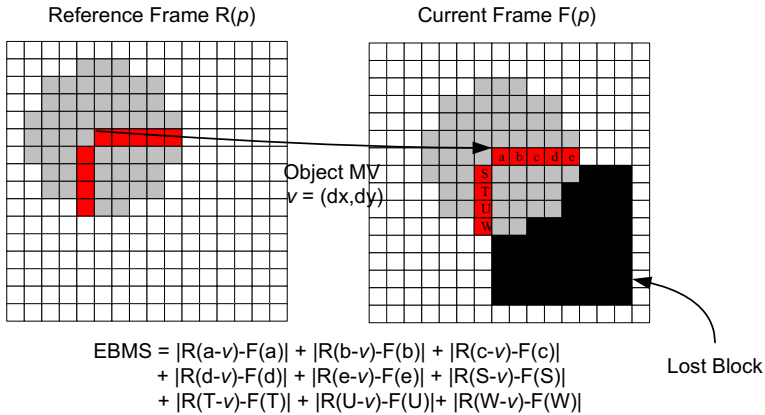


Fig. 6 Calculation of EBMS and region-based patching

different visual characteristics,¹ Football, News, Mobile, Foreman, Paris, and Stefan, are tested, where the whole image sequence is used for simulation. Simulation is performed with three different QP (quantization parameter) values, 20, 28, and 36 that correspond to high-quality, medium-quality, and low-quality video, respectively. Table 2 lists the PSNR values of the three evaluated methods. Although the proposed object-based method differs from the method in JM only on blocks that contain multiple objects, it achieves notably better performance (0.40 to 1.41 dB gain averaged over three QPs) for the investigated video sequences. A larger PSNR gap is observed for higher-quality (smaller QP) and more complex video sequences (such as Mobile). Note that the proposed method reduces to the JM block-based full-frame method [7] when FMO and slice groups are not employed. For the comparison with HMVE, consistently better improvement is observed since the proposed method incorporates the spatial information into object-based error concealment. Note that the HMVE algorithm was implemented and tested under the slice-loss scenario with the same simulation conditions (with FMO). Simulation results on higher-resolution standard videos² (“Crew” 480p and “Ducks_take_off” 720p) are given in Table 3. The proposed method provides better PSNR performance in all the evaluated cases. Note that the performance of Ref. [10] becomes worse. It is conjectured that the candidate MVs obtained by motion extrapolation are less reliable for high-resolution cases, because as compared to the CIF cases fewer significant image features reside in a macroblock for high-resolution videos.

In Table 4, the proposed method is compared with Ref. [12] and Ref. [9] in terms of the differential PSNR relative to the error concealment method implemented on JM. The PSNR results are averaged over the same number of trials as that used in the reference methods. Recall that Ref. [12] has a broader set of MV candidates and better boundary matching algorithm (EBMA) than the error concealment scheme in JM, and Ref. [9] realizes a corresponding hardware-efficient implementation. It can be seen that the proposed method achieves reliably good results and outperforms Ref. [12] (and thus Ref. [9]) in most cases. More

¹ <http://trace.eas.asu.edu/yuv/>

² <https://media.xiph.org/video/derf/>

Table 1 H.264 encoder settings in this paper

Parameter	Value
Profile	Baseline
Picture structure	IPPP
FMO TYPE	Disperse, 6 slice groups
Search Range	16
Fast motion estimation	on
Fast mode decision	on
MC motion vector resolution	1/4 pixel

significant difference is observed on the Football and Stefan sequences that contain fast moving objects and are regarded more difficult for error concealment.

The effect of the Intra period has also been investigated, and the results for the Paris sequence with QP equal to 20 are shown in Table 5. Three different Intra periods (10, 30, and 240) are tested, which correspond to inserting one I frame every 1/3, 1, and 8 s, respectively. Due to error propagation, it can be seen that a long Intra period significantly worsens the performance of error concealment. Nevertheless, the proposed method provides consistently better results as compared to JM 16.2 and Ref. [10]. The PSNR difference becomes even bigger with a longer Intra period.

Subjective concealment results on single still frames of the proposed method, Ref. [7], and Ref. [10] are shown in Figs. 8, 9 and 10, which substantiate the superiority of the proposed method. For the Football sequence, better concealment results are observed on the player even though the body motion is fast and irregular. The distinctive image features of the football player provide good clues for object identification. The News sequence has a static MPEG-4 banner and slow-moving news reporters in the foreground, and fast-moving dancers and poles on the background screen. Both the reporter and the background screen are better concealed with the proposed method. The Mobile sequence has complex color distribution and object movement, and thus more objects are formed during the process of object segmentation. Distinguished concealment results are observed for the proposed method within and around

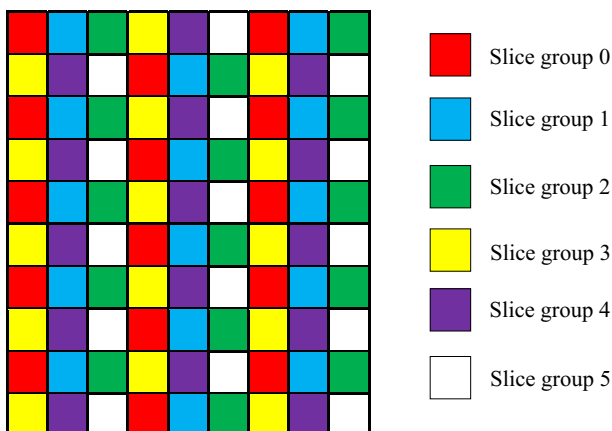
**Fig. 7** FMO of the dispersed mode with six slice groups

Table 2 PSNR comparison (in dBs) of the proposed method with JM16.2 [7] and Ref. [10] (Intra period = 10, the PSNR gain is averaged over PLR = 5, 10, 20% and QP = 20, 28, 36)

QP	PLR	Football			News			Mobile		
		JM 16.2	Ref. [10]	Proposed method	JM 16.2	Ref. [10]	Proposed method	JM 16.2	Ref. [10]	Proposed Method
20	0 %	43.41	43.41	43.41	44.07	44.07	44.07	42.80	42.80	42.80
	5 %	35.35	35.56(+0.21)	35.74(+0.39)	37.09	37.64(+0.56)	37.79(+0.70)	32.43	33.83(+1.40)	33.88(+1.45)
	10 %	29.48	29.77(+0.29)	30.07(+0.59)	31.89	32.63(+0.74)	32.85(+0.96)	25.60	27.67(+2.07)	27.73(+2.14)
	15 %	27.77	27.98(+0.21)	28.28(+0.51)	29.79	30.55(+0.75)	30.74(+0.94)	23.79	25.76(+1.97)	25.83(+2.03)
	28	0 %	37.21	37.21	37.21	38.93	38.93	38.93	35.26	35.26
36	5 %	32.14	32.30(+0.16)	32.46(+0.32)	34.30	34.63(+0.33)	34.71(+0.42)	29.08	30.16(+1.08)	30.20(+1.12)
	10 %	28.55	28.82(+0.27)	29.08(+0.53)	30.87	31.49(+0.62)	31.61(+0.75)	24.85	26.53(+1.68)	26.59(+1.74)
	15 %	27.16	27.37(+0.21)	27.64(+0.48)	29.12	29.76(+0.64)	29.89(+0.77)	23.31	25.06(+1.74)	25.11(+1.79)
	0 %	31.37	31.37	31.37	33.24	33.24	33.24	28.12	28.12	28.12
	5 %	28.98	29.09(+0.11)	29.18(+0.20)	30.86	31.09(+0.23)	31.13(+0.27)	25.66	26.14(+0.49)	26.16(+0.50)
Average gain	10 %	27.10	27.20(+0.10)	27.36(+0.26)	28.83	29.25(+0.42)	29.31(+0.48)	23.44	24.33(+0.89)	24.35(+0.91)
	15 %	26.04	26.16(+0.11)	26.34(+0.30)	27.73	28.18(+0.45)	28.24(+0.51)	22.23	23.24(+1.01)	23.27(+1.04)
	0 %	-	+0.19	+0.40	-	+0.53	+0.64	-	+1.37	+1.41
	28	0 %	37.05	37.05	36.66	36.66	36.66	36.66	36.67	36.67
	5 %	33.59	33.83(+0.23)	34.07(+0.48)	32.30	32.59(+0.29)	32.63(+0.33)	31.57	31.96(+0.38)	32.03(+0.45)
Foreman	10 %	30.68	31.11(+0.43)	31.51(+0.83)	28.77	29.22(+0.45)	29.27(+0.50)	27.73	28.27(+0.54)	28.39(+0.66)
	15 %	29.33	29.72(+0.39)	30.18(+0.85)	27.04	27.47(+0.43)	27.53(+0.49)	26.14	26.75(+0.61)	26.88(+0.74)
	0 %	31.93	31.93	31.93	30.32	30.32	30.32	29.98	29.98	29.98
	5 %	30.29	30.34(+0.05)	30.47(+0.18)	28.32	28.42(+0.10)	28.44(+0.11)	27.80	28.03(+0.23)	28.07(+0.27)
	28	0 %	37.05	37.05	37.05	36.66	36.66	36.66	36.67	36.67
Paris	5 %	36.91	37.29(+0.39)	37.64(+0.73)	35.39	35.87(+0.47)	35.93(+0.53)	34.94	35.42(+0.48)	35.51(+0.58)
	10 %	31.79	32.45(+0.66)	32.98(+1.20)	29.94	30.62(+0.68)	30.71(+0.77)	28.55	29.28(+0.72)	29.40(+0.84)
	15 %	30.08	30.72(+0.64)	31.28(+1.21)	27.89	28.50(+0.61)	28.58(+0.69)	26.88	27.55(+0.67)	27.67(+0.79)
	0 %	42.95	42.95	42.95	42.45	42.45	42.45	43.44	43.44	43.44
	5 %	36.91	37.29(+0.39)	37.64(+0.73)	35.39	35.87(+0.47)	35.93(+0.53)	34.94	35.42(+0.48)	35.51(+0.58)
Stefan	10 %	31.79	32.45(+0.66)	32.98(+1.20)	29.94	30.62(+0.68)	30.71(+0.77)	28.55	29.28(+0.72)	29.40(+0.84)
	15 %	30.08	30.72(+0.64)	31.28(+1.21)	27.89	28.50(+0.61)	28.58(+0.69)	26.88	27.55(+0.67)	27.67(+0.79)
	0 %	42.95	42.95	42.95	42.45	42.45	42.45	43.44	43.44	43.44
	5 %	36.91	37.29(+0.39)	37.64(+0.73)	35.39	35.87(+0.47)	35.93(+0.53)	34.94	35.42(+0.48)	35.51(+0.58)
	28	0 %	37.05	37.05	37.05	36.66	36.66	36.66	36.67	36.67

Table 2 (continued)

10 %	28.77	28.87(+0.10)	29.13(+0.36)	26.52	26.69(+0.17)	26.72(+0.19)	25.76	26.08(+0.32)	26.15(+0.38)
15 %	27.80	27.93(+0.13)	28.24(+0.44)	25.39	25.59(+0.20)	25.62(+0.23)	24.80	25.16(+0.36)	25.24(+0.44)
Average gain		+0.34	+0.70	–	+0.38	+0.43	–	+0.48	+0.57

Table 3 PSNR comparison (in dBs) for high-definition videos (Intra period = 10, the PSNR gain is averaged over PLR = 5, 10, 15 % and QP = 20, 28, 36)

(a) Crew (480p, 300 frames, 30 Hz)

QP	PLR	JM 16.2	Ref. [10]	Proposed Method
20	0 %	43.46	43.46	43.46
	5 %	37.58	37.13 (-0.45)	37.99 (+0.41)
	10 %	33.41	32.79 (-0.62)	33.89 (+0.48)
	15 %	31.74	31.16 (-0.58)	32.18 (+0.44)
28	0 %	38.23	38.23	38.23
	5 %	34.72	34.38 (-0.24)	34.95 (+0.23)
	10 %	32.36	31.77 (-0.59)	32.77 (+0.41)
	15 %	30.93	30.31 (-0.62)	31.27 (+0.35)
36	0 %	33.66	33.66	33.66
	5 %	31.98	31.73 (-0.25)	32.10 (+0.37)
	10 %	30.60	30.17 (-0.43)	30.77 (+0.59)
	15 %	29.60	29.12 (-0.48)	29.78 (+0.65)
Average gain		–	-0.47	+0.43

(b) Ducks_take_off (720p, 500 frames, 50 Hz)

QP	PLR	JM 16.2	Ref. [10]	Proposed Method
20	0 %	42.45	42.45	42.45
	5 %	32.36	32.29 (-0.07)	32.55 (+0.19)
	10 %	28.30	28.26 (-0.04)	28.49 (+0.19)
	15 %	26.61	26.56 (-0.05)	26.79 (+0.18)
28	0 %	35.53	35.53	35.53
	5 %	30.45	30.43 (-0.03)	30.60 (+0.15)
	10 %	27.69	27.67 (-0.02)	27.86 (+0.17)
	15 %	26.22	26.20 (-0.02)	26.40 (+0.18)
36	0 %	29.15	29.15	29.15
	5 %	27.32	27.30 (-0.02)	27.39 (+0.07)
	10 %	25.92	25.90 (-0.02)	26.02 (+0.10)
	15 %	24.98	24.96 (-0.02)	25.10 (+0.12)
Average gain		–	-0.03	+0.15

Table 4 Differential PSNR comparison (in dBs) of the proposed method with Ref. [12] and Ref. [9] (QP = 28, Intra period = 10); $\Delta\text{PSNR}_{\text{Ref}[12]} = \text{PSNR}_{\text{Ref}[12]} - \text{PSNR}_{\text{JM10.2}}$, $\Delta\text{PSNR}_{\text{Ref}[9]} = \text{PSNR}_{\text{Ref}[9]} - \text{PSNR}_{\text{JM10.2}}$, $\Delta\text{PSNR}_{\text{proposed}} = \text{PSNR}_{\text{proposed}} - \text{PSNR}_{\text{JM16.2}}$. Note that Ref. [12] and Ref. [9] was implemented on JM10.2 while the proposed method was implemented on JM16.2

	Football		News		Mobile		Foreman		Paris		Stefan	
PLR	5 %	10 %	5 %	10 %	5 %	10 %	5 %	10 %	5 %	10 %	5 %	10 %
$\Delta\text{PSNR}_{\text{Ref}[12]}$	+0.05	+0.05	+0.25	+0.39	+1.11	+1.66	+0.29	+0.41	+0.71	+0.99	+0.15	+0.26
$\Delta\text{PSNR}_{\text{Ref}[9]}$	-0.25	-0.49	+0.06	+0.15	+1.10	+1.62	+0.07	+1.06	+0.62	+0.83	-0.17	-0.29
$\Delta\text{PSNR}_{\text{proposed}}$	+0.32	+0.53	+0.42	+0.75	+1.12	+1.74	+0.48	+0.83	+0.33	+0.50	+0.45	+0.66

Table 5 The effects of Intra interval (PSNR for Paris, CIF, 1065 frames, 30 Hz, QP = 20)

Intra Period	PLR	Method		
		JM 16.2	Ref. [10]	Proposed Method
10	0 %	42.45	42.45	42.45
	5 %	35.39	35.87 (+0.47)	35.93 (+0.53)
	10 %	29.94	30.62 (+0.68)	30.71 (+0.77)
	15 %	27.89	28.50 (+0.61)	28.58 (+0.69)
30	0 %	42.24	42.24	42.24
	5 %	33.30	34.05 (+0.75)	34.20 (+0.90)
	10 %	28.25	29.10 (+0.85)	29.27 (+1.02)
	15 %	26.56	27.35 (+0.79)	27.49 (+0.93)
240	0 %	42.18	42.18	42.18
	5 %	29.34	30.53 (+1.19)	30.91 (+1.58)
	10 %	25.47	26.36 (+0.89)	26.64 (+1.17)
	15 %	24.81	25.76 (+0.95)	26.08 (+1.27)
Average gain		–	+0.80	+0.98

the calendar. Also, generally more significant improvements are achieved by the proposed method for low QP values because correct object segmentation and motion vector estimation rely on the quality of received data.

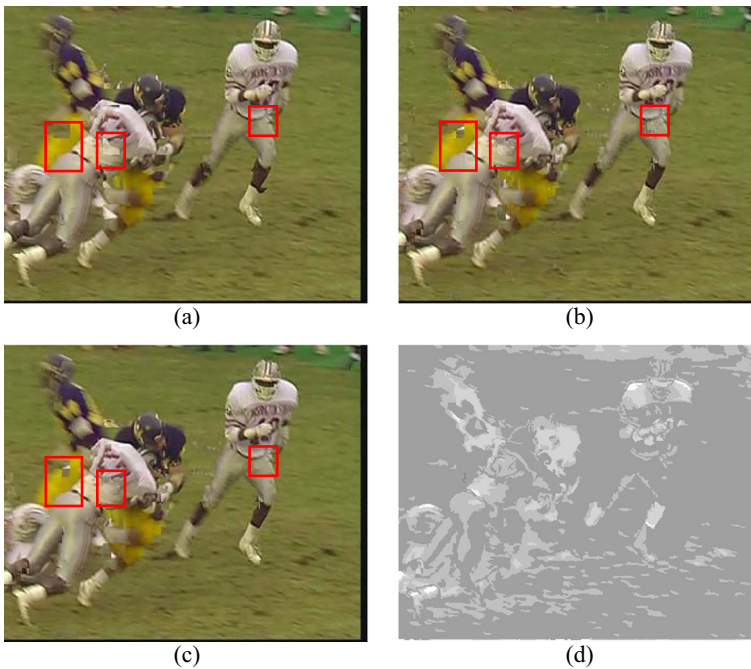


Fig. 8 Subjective evaluations (*Football*): **a** JM16.2 (27.79 dB), **b** Ref. [10] (29.24 dB), **c** the proposed method (29.75 dB), **d** object segmentation (QP = 28, PLR = 10 %, frame number = 174)

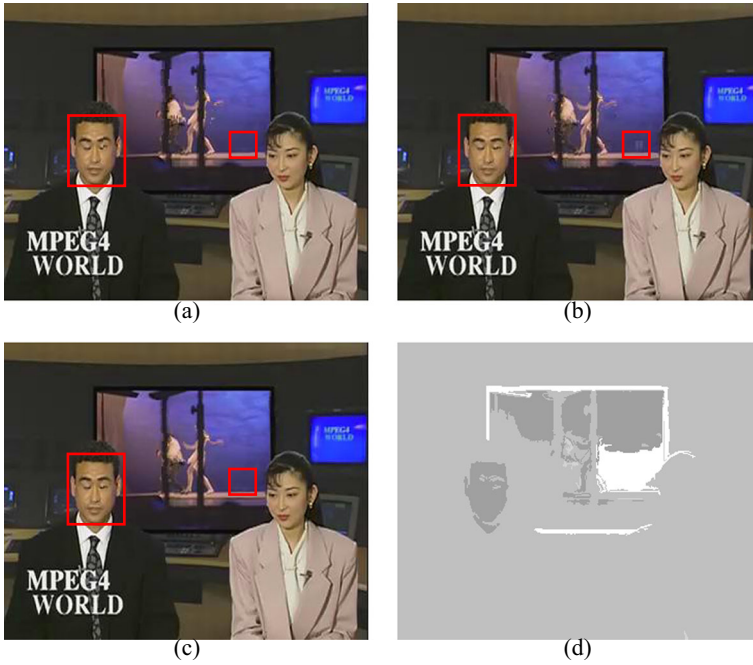


Fig. 9 Subjective evaluations (*News*): **a** JM16.2 (32.71 dB), **b** Ref. [10] (35.23 dB), **c** the proposed method (35.92 dB), **d** object segmentation (QP = 28, PLR = 10 %, frame number = 218)

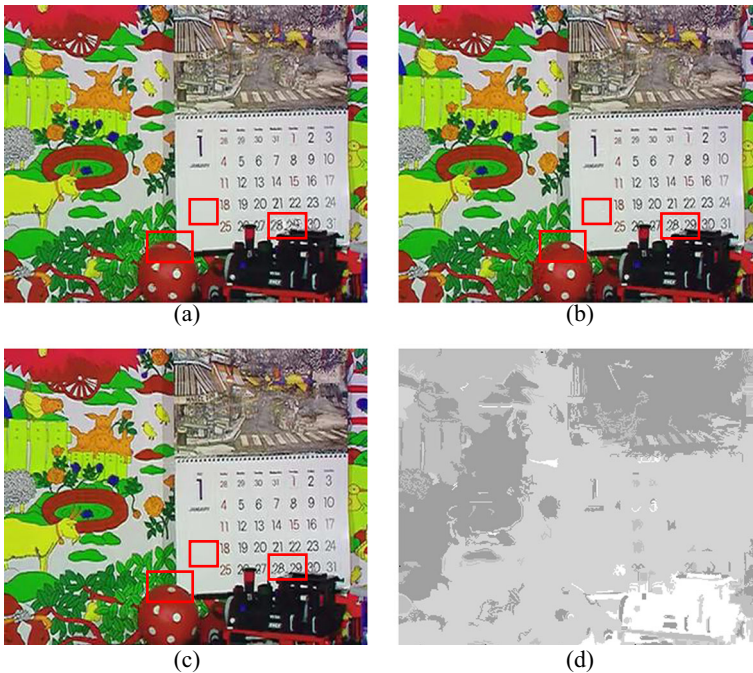


Fig. 10 Subjective evaluations (*Mobile*): **a** JM16.2 (24.62 dB), **b** Ref. [10] (28.26 dB), **c** the proposed method (28.42 dB), **d** object segmentation (QP = 28, PLR = 10 %, frame number = 58)

The proposed method has higher computational complexity, required especially for object segmentation and object matching. For our current version of un-optimized code, the computational complexity of the proposed method is approximately 100 times of JM and 10 times of Ref. [10] for CIF sequences. The decoding frame rate of the proposed method for CIF sequences on the used PC platform is about 1 frame/s. So hardware acceleration is expected if the proposed method is considered for real-time applications.

5 Conclusions

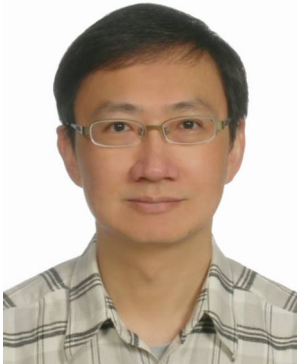
A new object-based error concealment technique is proposed for H.264-coded video with FMO in this paper. The proposed method exploits both the spatial and temporal information of successfully received slices. The visual objects within a frame are identified in the reference frame based on the color (grey-level) and motion consistency. The motion vector of an object is refined within a small search range around the initial object motion vector with a modified boundary-matching algorithm. Concealment is then performed in objects and the issue of multiple correspondences is properly resolved. The proposed method is evaluated in various encoding and transmission conditions such as different test sequences, different QPs, different Intra periods, and different packet loss rates. The proposed object-based method provides better performance than the conventional block-based approaches when multiple objects co-exist within a block. Compared with the methods in the literature, the proposed method provides considerably better objective visual quality especially for traditionally difficult cases and high-quality videos. The major drawback of the proposed method is its high computational complexity. Hardware acceleration is required for the use of the proposed method in real-time applications.

Acknowledgments This research is supported in part by the National Science Council, Taiwan, under the Grants NSC 101-2219-E-027-002 and NSC 102-2219-E-027-002.

References

1. Advanced Video Coding for Generic Audiovisual Services, ITU-T Rec. H.264, or ISO/IEC 14496–10 Information technology—Coding of audio-visual objects - Part 10: Advanced Video Coding (MPEG-4 AVC), Fifth Edition, May 2009
2. Fast Integer Pel and Fractional Pel Motion Estimation for JVT (JVT-F017.doc), ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, 6th Meeting, Awaji Island, Japan, Dec. 2002
3. Hsiao H-H, Leou J-J (2013) Background initialization and foreground segmentation for bootstrapping video sequences. *EURASIP J Image Video Proc* 12
4. ISO/IEC 14496–2 information technology—coding of audio-visual objects, Part 2: Visual, Amendment 1: visual extensions, July 2000
5. Li H, Zhong Y (2013) Motion characteristic differentiated error concealment. *Multimedia Tools Appl* 65: 297–320
6. Richardson IE (2010) The H.264 advanced video compression standard, 2nd edn. John Wiley & Sons Ltd
7. Wang YK, Hannuksela MM, Varsa V, Hourunranta A, Gabbouj M (2002) The error concealment feature in the H.26 L test model. In: *Proc. IEEE ICIP*, 729–736
8. Wang L, Wang J, Goto S, Ikenaga T (2007) Variable block size error concealment scheme based on H.264/AVC non-normative decoder. *International Symposium on Intelligent Signal Processing and Communication Systems*, China

9. Wu G-L, Chen C-Y, Wu T-H, Chien S-Y (2010) Efficient spatial-temporal error concealment algorithm and hardware architecture design for H.264/AVC. *IEEE Trans Circ Syst Vi Technol* 20(11):1409–1422
10. Yan B, Gharavi H (2010) A hybrid frame concealment algorithm for H.264/AVC. *IEEE Trans Image Process* 19(1):98–107
11. Yang S-H, Tsai J-C (2010) A fast and efficient H.264 error concealment technique based on coding modes. *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, Shanghai, China, March, pp. 1–4
12. Zhang J, Arnold JF, Frater MR (2000) A cell-loss concealment technique for MPEG-2 coded video. *IEEE Trans Circ Syst Vi Technol* 10(4):659–665



Shih-Hsuan Yang received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1987. He obtained the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of Michigan, Ann Arbor, USA, in 1990 and 1994, respectively. He joined the National Taipei University of Technology, Taipei, Taiwan, in 1994, where he is currently Professor of Computer Science and Information Engineering (CSIE). He was the chairman of the CSIE Department from 2004 to 2006. His major research interests include image and video coding, multimedia transmission, and smart TV. Dr. Yang is a senior member of the IEEE.