JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 31, 229-246 (2015)

DOI:10.1688/JISE.2015.31.1.12

Depth-Enhanced Error Concealment for H.264 Video*

SHIH-HSUAN YANG, CHIA-LING CHU AND CHI-WEN CHANG
Department of Computer Science and Information Engineering
National Taipei University of Technology
Taipei, 106 Taiwan

The H.264 video coding standard, owing to its excellent compression efficiency, has become the key technology of many video applications. But the H.264/AVC video is very vulnerable to data errors or packet losses. In this paper, we propose a new H.264 error concealment technique that exploits the depth relevance between spatially or temporally neighboring blocks. The motion vectors gathered from received H.264 slices are used to estimate the depth values and construct the depth maps. New motion vectors derived from depth maps are then added to the set of candidate motion vectors for concealing a lost block. The best motion vector is determined by an external boundary matching algorithm. Experimental results show that the proposed method is effective in improving the quality of received video without explicitly generating and transmitting the depth maps. Compared to the conventional method without incorporating the depth relevance, the proposed method increases the PSNR by 0.5 to 3.6 dBs.

Keywords: depth information, error concealment, H.264/AVC, quality of experience, multimedia communication

1. INTRODUCTION

The H.264 standard, also known as MPEG-4 AVC (Advanced Video Coding) [1], is one of the most important video coding standards. Owing to its superior rate-distortion performance, H.264 is widely employed in digital TV, mobile video, video streaming, and Blu-ray discs. Although Internet and wireless networks pave the way for ubiquitous visual communication, the transport environment of these networks is not always reliable. Packet loss may occur due to noise, interference, fading, or traffic congestion. For wireless communication where forward error correction is applied at the link layer, severe channel impairment such as deep fade may cause erasures of data frames. For real-time multimedia applications on packet networks with the IP/UDP/RTP protocol suite, an overdue packet will be dropped. The extensive use of prediction and variable-length codes further makes error control an important issue for H.264/AVC. The H.264/AVC standard has thus incorporated several error resilience tools to alleviate or confine the damage of packet loss. These tools include picture segmentation, Intra placement (placement of Intra macroblocks, intra slices, and intra pictures), reference picture selection, data partitioning, redundant slices, and flexible macroblock ordering (FMO). FMO assigns macroblocks to slices in an order other than the scan order, and so the image blocks can be re-organized in a prioritized manner for better exploiting the spatial correlation or unequal error protection. In conjunction with appropriate error concealment at the de-

Received March 5, 2013; revised June 22, 2013; accepted August 1, 2013.
Communicated by Yung-Yu Chuang.
* This work is supported in part by the National Science Council, Taiwan, under the Grant NSC 102-2219-E-027-002. A preliminary version of this paper was presented in the IAENG International Conference on Imaging Engineering, March 14-16, 2012, Hong Kong [13].

coder, 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 (Intra) or temporally (Inter) correlated samples [2, 3]. 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. On the other hand, temporal error concealment estimates the motion vector (MV) of a lost block from correlated blocks and restores the whole block by motion compensation. Three essential issues are involved in temporal error concealment, namely the size of a block, the selection of possible candidate MVs, and the criterion of choosing the best MV. Smaller and adaptive block-size generally gives better concealment results at the expense of added complexity [4]. Typical candidate MVs include the zero motion vector, MVs of spatially neighboring blocks, and MVs of the temporally collocated blocks. The collocated block refers to a block at the same spatial location in the previous or next reference frames as the current block. Appropriateness of an MV is usually verified using boundary matching, which assumes the smoothness and continuity across block boundaries. We will focus on temporal-domain error concealment in this paper.

Fig. 1. Video-plus-depth representation; left: texture image, right: depth image.

3D video is becoming popular owing to the advances of multi-view cameras, displays, coding algorithms, and rendering techniques. More and more attractive 3D contents further expedite the development of 3D video. In contrast to using planar (2D) image signals of all views, the 2D-plus-depth representation (2D images of selected views along with some depth maps) saves bits for storing and transmitting 3D data. A pixel in a depth map (shown in Fig. 1) indicates the distance between an object in a 3D scene and the viewer (camera). Usually all the pixels of an object have similar depth values, and so the depth information can be used to identify objects in the process of error concealment. Ali *et al.* [5] proposed a spatial error concealment technique for 2D-plus-depth video. The depth map provides an indication of object boundaries. The segmented foreground objects and background are separately processed, by spatial-domain frequency interpolation and extrapolation. For temporal-domain error concealment, it has been shown in [6, 7] that motion vectors derived from the depth maps (simply called the depth MVs) can improve the error concealment for 2D-plus-depth video. In this paper, we propose a new depth-enhanced error concealment technique *for mono-view H.264 video sequences*. Different from the previous work that requires the depth map to be encoded and transmitted, *we estimate the depth information at the decoder directly from the MVs of the received 2D video sequence without a separate depth map*. These MVs derived from depth information are added to the set of candidate MVs. Compared to the conventional motion-compensated error concealment techniques without depth information, the proposed method renders decoded video of better visual quality.

The rest of this paper is organized as follows. In section 2, the error concealment algorithm in the current JM reference software and temporal-domain depth-aided error concealment techniques are reviewed. The proposed depth-enhanced error concealment method is presented in section 3. Experimental results and analyses are given in section 4, followed by the conclusion.

2. RELATED WORKS

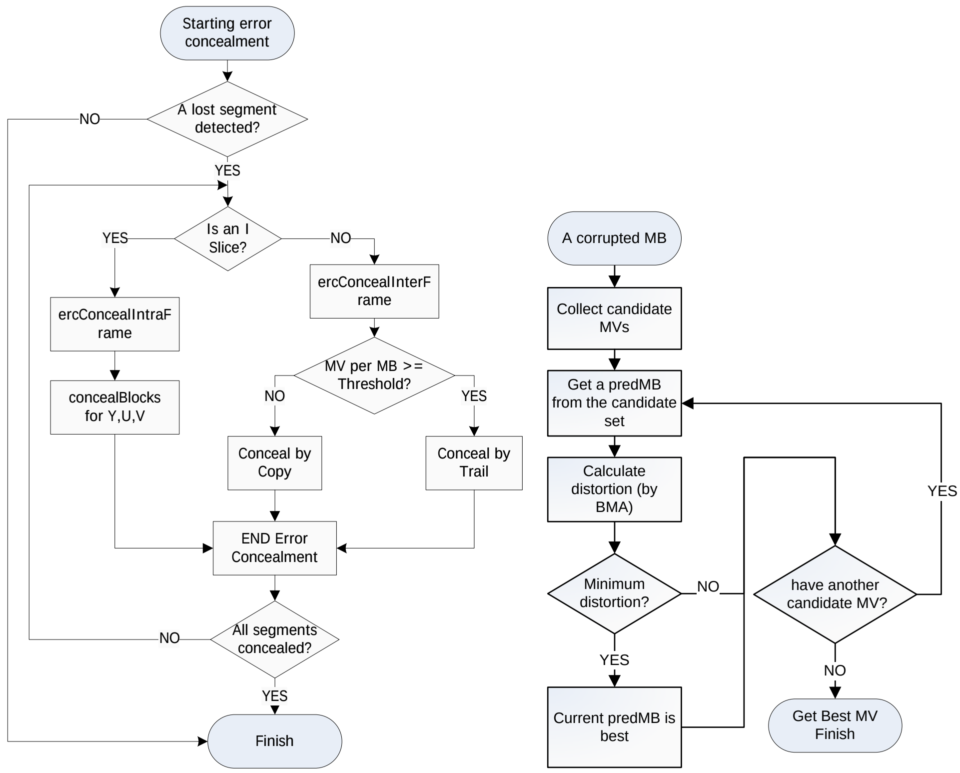
2.1 Error Concealment in JM

The true motion vector, which mimics the motion trajectory of an object, is desired for temporal error concealment. True motion can be estimated based on the spatial and temporal coherence of the motion field. The obtained true motion of a block should be close to the global motion of a relevant object [8]. However, it is generally difficult and complicated to implement a true motion estimation algorithm in the error situations because the information (pixels, MVs) required for motion estimation may be deficient. In the following, we introduce the block-matching methods that find MVs minimizing the local distortion.

The flowchart of the error concealment algorithm in the current JM (Joint Model) reference software [9] is shown in Fig. 2 (a). For a lost I slice, intra concealment is used. To be more precise, a pixel in a lost macroblock is estimated using bilinear interpolation of already received or concealed bordering pixels. For P slices, JM collects the motion vectors surrounding the lost macroblock (MB) plus the zero MV as the candidate MVs to be used for concealment. The MV that incurs the smallest cost in the BMA (Boundary Match Algorithm) cost is chosen, as illustrated in Fig. 2 (b). BMA calculates the block difference as the sum of absolute differences between the inside pixels of a block and the outside bordering pixels of adjacent blocks, as shown in Fig. 3. The obtained patching block fails to give satisfactory results if no reliable MV exists in the candidate set.

2.2 Depth-Based Error Concealment Techniques

Yan [6] presented a depth-based BMA (DBMA) method for 2D-plus-depth video. In addition to the zero vector and MVs of spatially neighboring blocks and the collocated block, a new MV derived from depth-map motion estimation is incorporated in the set of candidate MVs. Furthermore, a neighboring MB with a depth value dissimilar to that of the lost MB will be excluded from the candidate set. The above argument is based on the observation that the blocks associated with the same object should have similar MVs. In the case that both the texture MBs and the depth MBs are lost, the MVs of the neighboring MBs in the depth map will be used instead. The best MV for concealment is determined by BMA. The simulation scenario in [6] assumes random loss of MBs, which may be impractical for typical H.264 applications where a slice (consisting of contiguous MBs) is packetized into NAL units.



(a) (b)

Fig. 2. (a) Flowchart of the JM error concealment for a lost slice; (b) determination of the best MV from the candidate set.

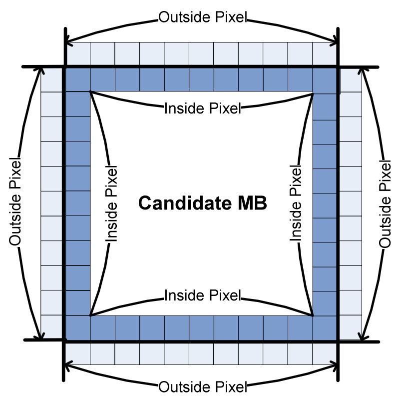


Fig. 3. BMA (Boundary match algorithm).

In [7], another depth-based temporal-domain error concealment (DTEC) algorithm was proposed for 2D-plus-depth video. Besides the available MVs of the MBs surrounding a lost MB, the MV of the corresponding MB in depth map is also included in the set of candidate MVs. The distortion associated with an MV is computed as the sum of two components, the SAD (sum of absolute differences) resulting from the texture image and the SAD resulting from the depth image. Furthermore, a lost MB is classified as either homogeneous or boundary, according to the features in the depth map. A homogeneous MB will be compensated as a whole. A boundary MB, on the other hand, will be segmented into the foreground region and the background region, and these two regions will be independently compensated. DTEC can provide better concealment performance when foreground and background have different motion. However, DTEC will fail to give good results if both the texture and the depth information at the same location are missing. Note that the above depth-based error concealment methods [6, 7] require the depth map to be separately encoded and transmitted, which is not applicable for pure 2D video sequences.

3. PROPOSED METHOD

The major contribution of this paper is the inclusion of depth MVs for error concealment at the decoder without explicitly transmitting the depth maps at the encoder. The proposed method also dispenses with complicated depth estimation usually required in the generation of 2D-plus-depth videos [10]. The flowchart of the proposed method is shown in Fig. 4, where the major modifications to the error concealment method in JM are highlighted and will be explained in details in the following subsections. In the first

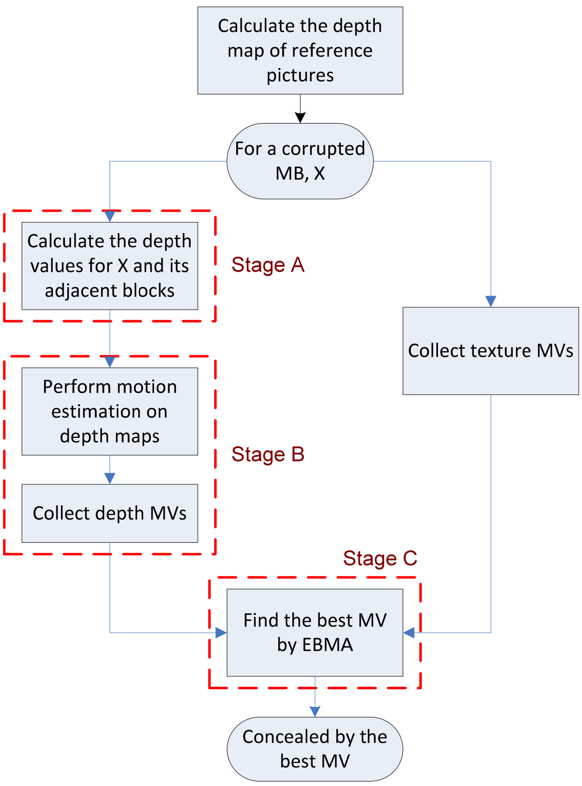


Fig. 4. Flowchart of the proposed depth-enhanced error concealment.

stage (Stage A), the depth values in units of 4×4 blocks are calculated from the received MVs and form the depth maps. In the second stage (Stage B), new depth MVs are derived from depth maps and incorporated into the set of candidate MVs additional to the conventional texture MVs. In the third stage (Stage C), the best MV in the candidate set is selected using the EBMA (Extended Boundary Matching Algorithm).

3.1 Generation of Depth Maps

We exploit the motion parallax to estimate the depth value of a block [11]. It is known that the motion displacement usually decreases with the distance of the object to the viewer, *i.e.*, an object closer to an observer will have a larger displacement. This assumption is especially true for the case of a moving camera with static scenes. Note that for H.264-coded video the MVs of temporally predictive blocks (P-blocks) have been generated by the encoder and transmitted to the receiver. Therefore, for the purpose of error concealment we may directly use the MVs embedded in the received bitstream to estimate the depth value of a block.

In H.264, an MV is given for every 4×4 luma P-block (a larger prediction unit assumes the same MV for all its constituent blocks). The depth of a block (in reference pictures) at location (i, j), $D(i, j)$, is simply estimated in this paper as [11]

$$D(i, j) = c \cdot \sqrt{MV(i, j)_x^2 + MV(i, j)_y^2} \tag{1}$$

where $MV(i, j)$ is the motion vector, the subscripts x and y represent the horizontal and vertical components, respectively, and c is a scaling constant (we choose $c = 1$ in this paper). The depth values of all the blocks for all the received pictures are calculated and stored as depth maps. For a block in a missing slice, the depth of a lost block is estimated from the previous and next pictures as

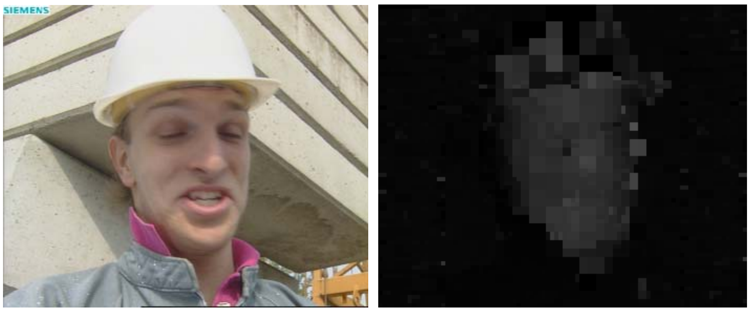
$$D(i, j) = c \cdot \sqrt{\frac{MV_{n-1}(i, j)_x^2 + MV_{n-1}(i, j)_y^2 + MV_{n+1}(i, j)_x^2 + MV_{n+1}(i, j)_y^2}{2}} \tag{2}$$

where the subscript n–1 and n+1 refer to the previous and next pictures.

Examples of depth maps derived from the described algorithm are shown in Fig. 5. Figs. 5 (a) and (c) have static background, and the moving foreground objects (head in Fig. 5 (a), hands and faces in Fig. 5 (c)) can thus be identified. Camera motion exists in Fig. 5 (b) (slow) and Fig. 5 (d) (fast), and consequently objects in the scene will be segmented into parts according to their relative motion to the camera. Although the depth information estimated by the coded motion vectors is not very accurate, it still helps error concealment especially on object boundaries.

3.2 Finding Depth MVs

Motion estimation is performed on the depth maps in units of macroblocks to obtain depth MVs. Recall that a point in the depth map corresponds to a 4×4 luma block. Therefore, there are sixteen 4×4 depth values within a macroblock. The depth SAD of a macroblock with top-left location (i, j) is calculated as



(a) Foreman, frame number = 79.

(b) Mobile, frame number = 15.

(c) Paris, frame number = 45.

(d) Stefan, frame number = 90.

Fig. 5. Sample images in the test video sequences and the corresponding depth maps.

$$\text{SAD}_{i,j}(p,q)=\sum_{k=0}^{3}\sum_{l=0}^{3}\left|D_n(i+k,\, j+l)-D_{n-1}(i+k+p,\, j+l+q)\right| \tag{3}$$

where $D_k(i, j)$ denotes the depth value at location (i, j) for frame k and (p, q) is the displacement from frame k–1 to frame k, assuming that a slice in frame n is lost. Thus, $D_n(i, j)$ is calculated by (2) and $D_{n-1}(i, j)$ is calculated by (1). The depth motion vector is found by evaluating $\text{SAD}_{i,j}(p, q)$ over a search range identical to that for texture MVs, and taking the displacement (p, q) that incurs the minimal SAD. If the previous frame is lost but the next frame is available, the depth MVs will be found from the next frame instead. No depth MV will be used if both the previous frame and the next frame are lost.

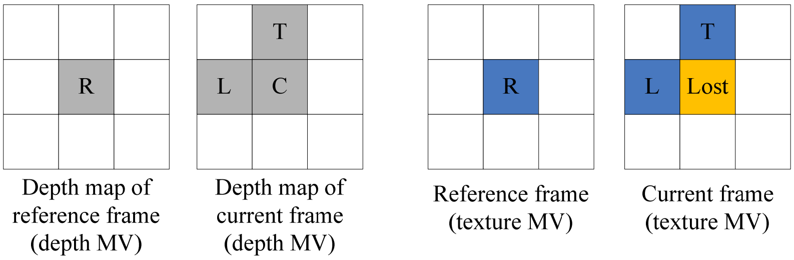


Fig. 6. Set of candidate motion vectors.

To increase the matching accuracy, MVs of adjacent MBs are included in the candidate set. These MBs are the collocated MB, and the MBs to the left and on the top of the lost MB, as shown in Fig. 6. Therefore, there are at most eight motion vectors in the candidate set, the zero MV, 3 texture MVs, and 4 depth MVs. The depth MVs are obtained from depth maps by the procedure explained in the last paragraph. The texture MVs are obtained through the decoding of the received bitstream, the same as what is done in the error concealment scheme of JM. Some of these candidate MVs may be the same, some of them may be lost, and some of them may be unavailable due to lost reference data. To facilitate better availability of adjacent blocks, we employ the FMO Type 1 (dispersed pattern) as the error resilience tool. Four slice groups are formed within each frame, as shown in Fig. 7.

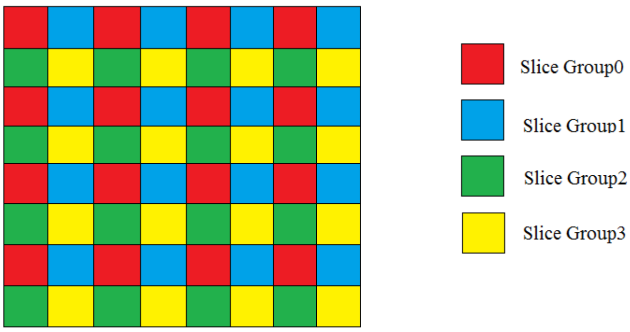


Fig. 7. Slice group pattern of the disperse FMO used in this paper.

3.3 Determining the Best MV

To determine the best MV in the candidate set for a lost MB, the EBMA (Extended Boundary Matching Algorithm) cost is evaluated for each of the candidate MVs. EBMA calculates the distortion as the sum of absolute differences between the outside pixels of MBs in the current frame and the outside pixels of a matching MB in the reference frame, as shown in Fig. 8 [12]. The lost MB is concealed by motion compensation with the MV that incurs the minimal EBMA cost.

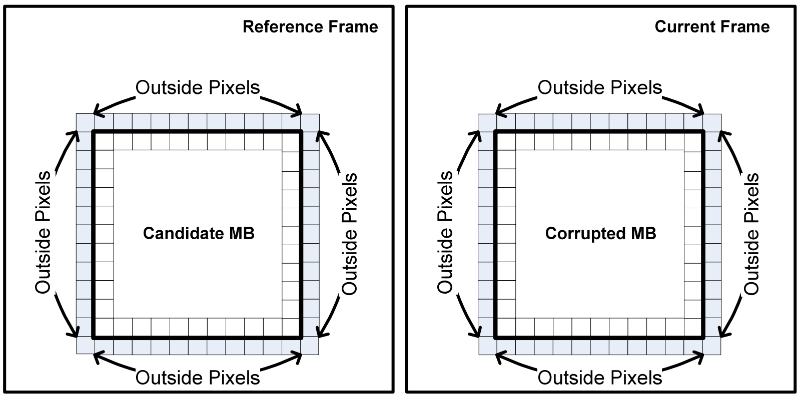


Fig. 8. EBMA (Extended Boundary Matching Algorithm).

4. EXPERIMENTAL RESULTS AND ANALYSES

The experimental settings of the H.264/AVC codec under investigation are shown in Table 1. The first 100 frames of the four test sequences (Foreman, Mobile, Paris, Stefan) in CIF resolution are used for simulation. The Baseline Profile (IPPP… structure) is used with the dispersed FMO of four slice groups. Five QP (quantization parameter) values (20, 24, 28, 32, 36) that correspond to a wide range of image quality are tested.

Table 1. H.264/AVC encoding parameters in this paper.

Coding Parameters	Values
Codec	JM 15.1
Profile	Baseline (IPPP)
Frame Rate	30.0
Intra Period	30
Enable IDR	enabled
Number of Reference Frames	1
Motion Vector Resolution	quarter pixel
Search Range	16
RDO	on
Fast Motion Estimation	on
Fast Mode Decision	on
FMO TYPE	dispersed, 4 slice groups

Each slice group is packetized into a packet and independently transmitted across a packet network with packet loss rate (PLR) equal to 5%, 10%, or 15%. For each QP value and each PLR, the PSNR is obtained by averaging the results against twenty random patterns.

Table 2. PSNR Performance (in dB). The average in last rows is taken for the five selected QP values.

(a) Foreman

QP	Algorithm	PSNR under PLR		
		5%	10%	15%
20	JM	35.30	32.60	30.15
	Proposed	37.51	34.30	32.18
24	JM	34.46	31.88	29.74
	Proposed	36.12	33.69	31.66
28	JM	33.33	31.22	29.13
	Proposed	34.73	32.65	30.96
32	JM	32.27	30.61	28.73
	Proposed	33.07	31.23	29.87
36	JM	30.92	29.58	27.82
	Proposed	31.38	30.10	28.86
Avg.	**JM**	**33.26**	**31.18**	**29.11**
	Proposed	**34.56**	**32.39**	**30.71**

(b) Mobile

QP	Algorithm	PSNR under PLR		
		5%	10%	15%
20	JM	29.73	26.60	24.07
	Proposed	33.06	30.03	27.50
24	JM	28.92	26.03	23.55
	Proposed	32.07	29.21	27.11
28	JM	28.00	25.54	23.17
	Proposed	30.71	28.37	26.46
32	JM	26.99	24.55	22.54
	Proposed	28.83	27.09	25.53
36	JM	25.26	23.38	21.57
	Proposed	26.56	25.40	24.24
Avg.	**JM**	**27.78**	**25.22**	**22.98**
	Proposed	**30.25**	**28.02**	**26.17**

(c) Paris

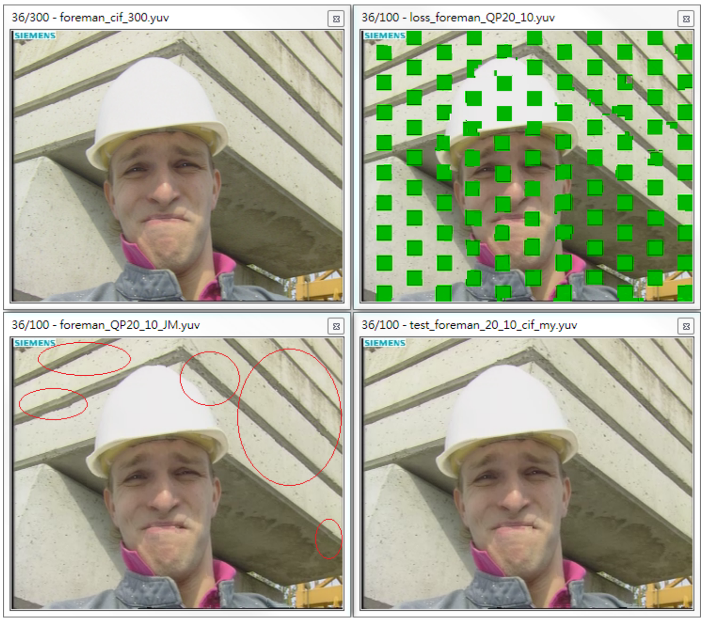
QP	Algorithm	PSNR under PLR		
		5%	10%	15%
20	JM	33.25	30.16	28.14
	Proposed	36.12	33.46	31.09
24	JM	32.53	30.01	27.91
	Proposed	34.88	32.69	30.43
28	JM	31.40	28.92	27.16
	Proposed	33.47	31.64	29.60
32	JM	30.19	28.04	26.48
	Proposed	31.51	30.23	28.45
36	JM	28.07	26.66	25.47
	Proposed	28.98	28.15	27.08
Avg.	**JM**	**31.09**	**28.76**	**27.03**
	Proposed	**32.99**	**31.23**	**29.33**

(d) Stefan

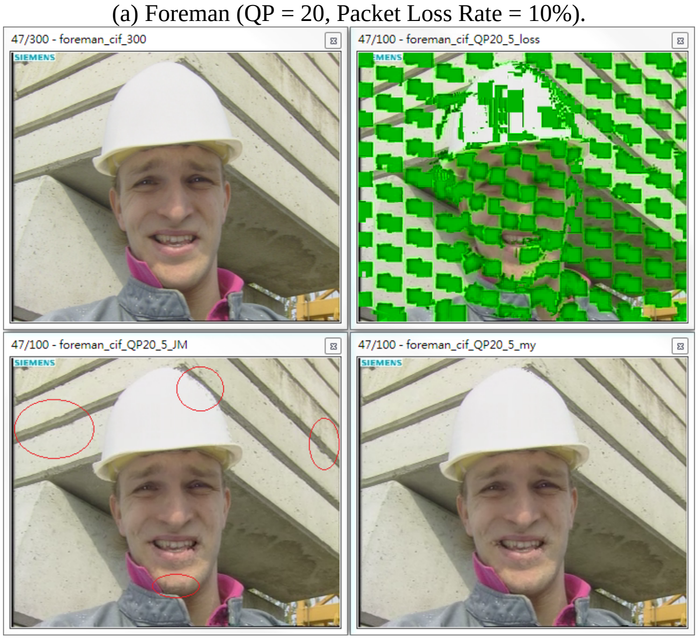
QP	Algorithm	PSNR under PLR		
		5%	10%	15%
20	JM	31.41	28.45	26.33
	Proposed	33.06	30.10	27.74
24	JM	30.99	28.06	25.83
	Proposed	31.91	29.48	27.27
28	JM	29.53	27.18	25.23
	Proposed	30.87	28.83	26.69
32	JM	28.30	26.38	24.70
	Proposed	29.23	27.61	25.80
36	JM	26.88	25.32	23.76
	Proposed	27.52	26.28	24.92
Avg.	**JM**	**29.42**	**27.08**	**25.17**
	Proposed	**30.52**	**28.46**	**26.48**

We first compare our method with the JM reference software [9], and the quantitative results are shown in Table 2. It is observed that the proposed method provides consistent better results with a 0.5 to 3.6 dB margin on average for a specific QP and PLR. The introduced depth MVs are more influential in the Mobile sequence that has complex scene and motion. The gap is more significant for smaller QP values (higher-quality video). A larger PLR also slightly increases the PSNR gap. To investigate the impact of the encoder's search range (SR) for motion vectors, we tested SR equal to 8 and 32 additional to the original search range (SR = 16). An increased SR does not always generate better error-concealment results for the proposed method. Furthermore, different SRs only slightly change the PSNR gap (less than 0.1 dB) between the proposed method and JM. Two reasons may account for this unusual behavior. First, motion estimation is performed in the rate-distortion optimization (ROD) sense that also counts the bits used for encoding the motion vector. Second, the depth MVs performed in units of 4×4 pixels is rough and thus it is less meaningful to use accurate motion vectors.

In Fig. 9, the proposed method and the error concealment method in JM are subject-

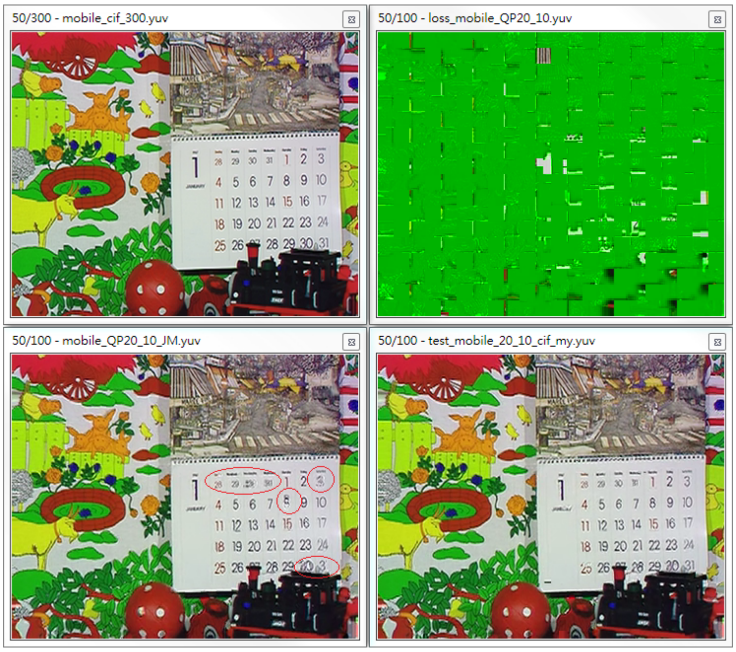


(a) Foreman (QP = 20, Packet Loss Rate = 10%).

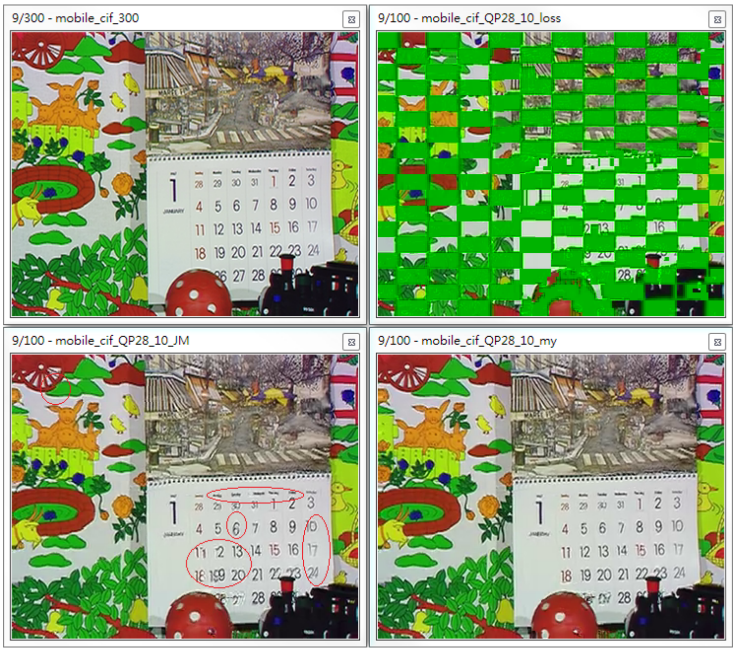


(b) Foreman (QP = 20, Packet Loss Rate = 5%).

Fig. 9. (a)-(b) Snap-shots of the error-concealment results. Pictures in each case are arranged as follows, top-left: original frame, top-right: decoded frame without error concealment, bottom-left: decoded frame concealed by JM, bottom right: decoded frame concealed by the proposed method.

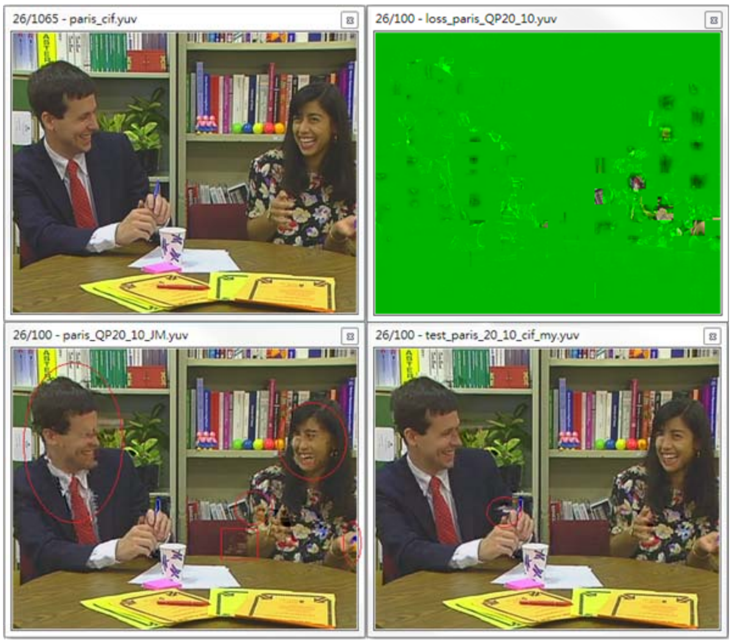


(c) Mobile (QP = 20, Packet Loss Rate = 10%).



(d) Mobile (QP = 28, Packet Loss Rate = 10%).

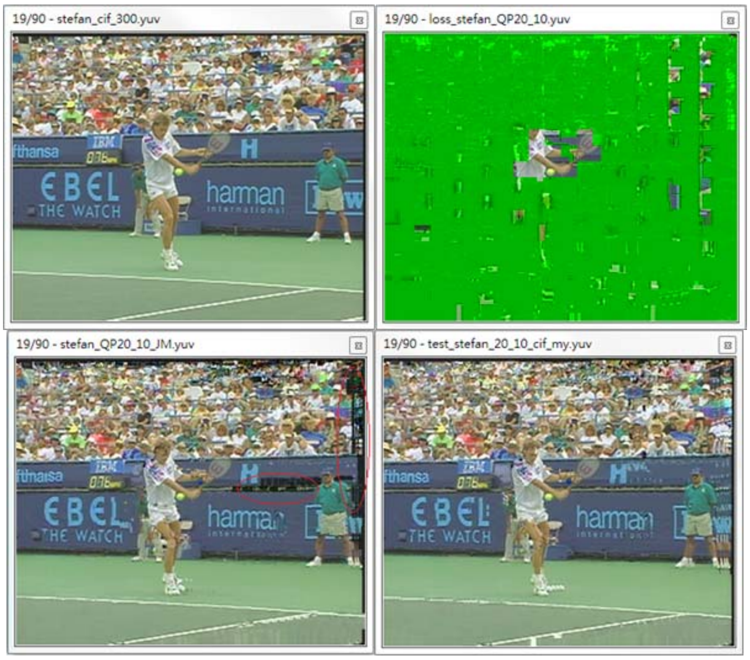
Fig. 9. (c)-(d) Snap-shots of the error-concealment results. Pictures in each case are arranged as follows, top-left: original frame, top-right: decoded frame without error concealment, bottom-left: decoded frame concealed by JM, bottom right: decoded frame concealed by the proposed method.



(e) Paris (QP = 20, Packet Loss Rate = 10%).

(f) Paris (QP = 28, Packet Loss Rate = 15%).

Fig. 9. (e)-(f) Snap-shots of the error-concealment results. Pictures in each case are arranged as follows, top-left: original frame, top-right: decoded frame without error concealment, bottom-left: decoded frame concealed by JM, bottom right: decoded frame concealed by the proposed method.



(g) Stefan (QP = 20, Packet Loss Rate = 10%).

(h) Stefan (QP = 28, Packet Loss Rate = 15%).

Fig. 9. (g)-(h) Snap-shots of the error-concealment results. Pictures in each case are arranged as follows, top-left: original frame, top-right: decoded frame without error concealment, bottom-left: decoded frame concealed by JM, bottom right: decoded frame concealed by the proposed method.

tively evaluated with sample frames taken from the decoded sequences. For the Foreman and Stefan sequences, notable improvement is found in regions containing edges. For the mobile sequence, the proposed method gives much better concealment results for the text-and-number regions. For the Paris sequence, the difference is magnificent in the face and hands that have significant movement. The depth information added in the proposed method successfully identifies the objects and their motion in these cases.

To explore the significance of depth information, we investigate the contribution of depth MVs in error concealment and the results are listed in Table 3. The percentage of the depth MVs selected as the best MV (different from the texture MVs) varies drastically from a few percent (for Paris) to thirty percent (for Mobile). Video sequences with complex scenes and high-motion characteristics benefit more from the introduced depth MVs. The QP value also affects the percentage of depth MVs to be selected, where a gradual decrease is found for an increased QP value. These observations are consistent with the results in Table 2.

The major expense of the proposed method is the complexity, required especially for generating and storing the depth maps and depth MVs. On the tested simulation platform (Microsoft Windows 7 on Intel Core 2 Quad CPU 2.67GHz with 4GB RAM), the total computational time for error concealment lies between 7 and 49 seconds for test sequences of 100 frames. The time complexity increases linearly with the packet loss rate. On the other hand, the error-concealment method in JM is relatively simple. It takes only 0.3 to 3 seconds under the same test conditions. So hardware acceleration is expected if the proposed method is considered for real-time applications.

Table 3. The percentage of depth MVs selected as the best MV.

(a) Foreman

QP	PLR = 5%	PLR = 10%	PLR = 15%
20	9.9%	9.2%	9.4%
24	8.6%	8.8%	8.7%
28	7.7%	7.8%	7.2%
32	5.3%	5.9%	5.7%
36	4.5%	4.4%	4.2%
Avg.	**7.2%**	**7.2%**	**7.0%**

(b) Mobile

QP	PLR = 5%	PLR = 10%	PLR = 15%
20	33.0%	32.0%	30.8%
24	33.1%	32.2%	30.8%
28	31.8%	30.6%	29.4%
32	29.2%	28.5%	27.6%
36	27.1%	27.6%	26.1%
Avg.	**30.8%**	**30.2%**	**28.9%**

(c) Paris

QP	PLR = 5%	PLR = 10%	PLR = 15%
20	3.4%	3.3%	3.3%
24	3.3%	3.4%	3.2%
28	2.7%	2.8%	2.7%
32	2.3%	2.3%	2.3%
36	2.0%	2.1%	2.0%
Avg.	**2.7%**	**2.8%**	**2.7%**

(d) Stefan

QP	PLR = 5%	PLR = 10%	PLR = 15%
20	12.0%	13.1%	12.2%
24	12.3%	12.2%	11.8%
28	12.2%	12.1%	11.2%
32	10.6%	10.4%	9.9%
36	9.0%	9.3%	9.3%
Avg.	**11.2%**	**11.4%**	**10.9%**

We also compare our method with DBMA [6] and DTEC [7], the two error concealment methods that explicitly encode and transmit the depth maps. To make fair comparison, the H.264 encoding parameters are adjusted to be the same as those benchmarking approaches. Both DBMA and DTEC consider the 2D-plus-depth scenario and so they receive additional depth maps besides the 2D (texture) images. Furthermore, it is assumed that the I frames are correctly received with proper error protection and the errors only occur in the P frames. DBMA assumes a random loss pattern for macroblocks from either texture images or depth maps. The comparison with DBMA is given in Table 4. It is observed that the proposed method achieves slightly better PSNR performance than DBMA. On the other hand, DTEC assumes that a slice (with contiguous macroblocks) is randomly lost, including slices containing the depth map. The comparison with DTEC is given in Table 5. The proposed method achieves close or slightly worse PSNR performance, and the PSNR gap increases with the packet loss rate. However, it should be commented again that the proposed method is applicable to pure 2D video sequences, which performs error concealment without explicit depth maps.

Table 4. PSNR comparison with DBMA [6] (test sequence: Interview (720×576), I slice QP = 28, P slice QP = 32, an I frame in every 10 frames). Note that DBMA requires a separate depth map.

Algorithm	PSNR (dB)			
	PLR = 10%	PLR = 20%	PLR = 30%	PLR = 40%
JM	36.12	35.42	34.81	34.34
DBMA [6]	36.33	35.67	35.05	34.63
Proposed Method	36.41	35.74	35.20	34.75

Table 5. PSNR comparison with DTEC [7] (test sequence: Orbi (720×576), I slice QP = 34, P slice QP = 36, an I frame in every 15 frames). Note that DTEC requires a separate depth map.

Algorithm	PSNR (dB)		
	PLR = 5%	PLR = 10%	PLR = 20%
JM	33.39	33.12	30.52
DTEC [7]	33.83	33.71	31.51
Proposed Method	33.85	33.62	31.27

5. CONCLUSION

A new depth-enhanced error concealment technique for 2D H.264 video sequences is presented, which incorporates depth MVs without explicitly transmitting the depth maps at the encoder. The proposed method first estimates the depth value of a block from the received motion vectors. These depth values are then used to derive extra candidate motion vectors for temporal error concealment. With the added depth information, moving objects are more easily identified and concealed. The proposed algorithm yields good visual quality with a minimal increase of computational complexity. Experimental results substantiate the superiority in visual quality for the proposed method over JM. The proposed method can also achieve close PSNR performance to the methods inherently incorporating the depth maps.

REFERENCES

1. *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264, 5th ed., 2009.

2. J. Wu, X. Liu, and K.Y. Yoo, "A temporal error concealment method for H.264/AVC using motion vector recovery," *IEEE Transactions on Consumer Electronics*, Vol. 54, 2008, pp. 1880-1885.
3. D. Kim, S. Yang, and J. Jeong, "A new temporal error concealment method for H.264 using adaptive block sizes," in *Proceedings of IEEE International Conference on Image Processing*, Vol. 3, 2005, pp. 928-931.
4. S.-H. Yang and J.-C. Tsai, "A fast and efficient H.264 error concealment technique based on coding modes," in *Proceedings of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2010, pp. 1-4.
5. A. Ali, H. A. Karim, N. A. M. Arif, and A. Sali, "Depth image-based spatial error concealment for 3-D video transmission," in *Proceedings of IEEE Student Conference on Research and Development*, 2010, pp. 421-425.
6. B. Yan, "A novel H.264 based motion vector recovery method for 3D video transmission," *IEEE Transactions on Consumer Electronics*, Vol. 53, 2007, pp. 1546-1552.
7. Y. Liu, J. Wang, and H. Zhang, "Depth image-based temporal error concealment for 3-D video transmission," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 20, 2010, pp. 600-604.
8. G. de Haan, P. W. Biezen, H. Huijgen, and O. A. Ojo, "True-motion estimation with 3-D recursive search block matching," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 3, 1993, pp. 368-379.
9. H.264 Software Coordination, http://iphome.hhi.de/suehring/tml/.
10. L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, 2004, pp. I-74-I-81.
11. I. Ideses, L. Yaroslavsky, and B. Fishbain, "Depth map manipulation for 3D visualization," in *Proceedings of 3DTV Conference: The True Vision – Capture, Transmission and Display of 3D Video*, 2008, pp. 337-340
12. S.-H. Yang, C.-L. Chu, and C.-W. Chang, "An H.264/AVC error concealment technique enhanced by depth correlation," in *Proceedings of IAENG International Conference on Imaging Engineering*, 2012, pp. 668-673.

Shih-Hsuan Yang (楊士萱) 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

Chia-Ling Chu (朱嘉玲) received the M.S. degree in Computer Science and Information Engineering from the National Taipei University of Technology in 2011.

Chi-Wen Chang (**張繼文)** received the M.S. degree in Computer Science and Information Engineering from the National Taipei University of Technology in 2012.