Proceedings of the International MultiConference of Engineers and Computer Scientists 2012 Vol I,
IMECS 2012, March 14 - 16, 2012, Hong Kong

An H.264/AVC Error Concealment Technique Enhanced by Depth Correlation

Shih-Hsuan Yang, Chia-Ling Chu, and Chi-Wen Chang

***Abstract*—The H.264 video coding standard, owing to its excellent compression efficiency, has become the key technology of many video applications. Coded video transmission over mobile channels or Internet, however, is 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. At the decoder, the motion vectors gathered from received H.264 slices are used to estimate the depth value of a lost block. The motion vectors, estimated 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 the external boundary matching algorithm. Experimental results show that the proposed method is effective in improving the quality of received video without generating and transmitting depth maps. Compared to the conventional method without incorporating the depth correlation, the proposed method increases the PSNR by 1 to 3 dBs.**

***Index Terms*—depth information, error concealment, H.264/AVC, multimedia communication**

I. 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 may offer the environment for ubiquitous visual communication, the transport environment of these networks is not always reliable. Under an IP/UDP/RTP real-time protocol suite, packet losses may frequently occur because of network congestion. In the case of wireless communication where FEC (Forward Error Correction) is applied at lower layers of the system, severe channel impairment such as deep fade may cause erasures of data frames. The extensive use of prediction and variable-length coding further makes the H.264/AVC video very vulnerable to transmission errors. To combat channel impairment, the H.264/AVC standard incorporates several error resilience tools, including FMO (Flexible Macroblock Ordering) as shown in Figure 1. In conjunction with appropriate error concealment at the decoder, these error resilience tools may improve the error performance of the overall system.

Manuscript received December 8, 2011. This work was supported in part by the National Science Council, Taiwan, under the Grant NSC 100-2219-E-027-002.

Shih-Hsuan Yang, Chia-Ling Chu, and Chi-Wen Chang are with the Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan 10608 (phone: +886-2-27722171 ext. 4211; fax: +886-2-87732945; e-mails: shyang@ntut.edu.tw, bow76512@gmail.com, calvinds0122@hotmail.com).

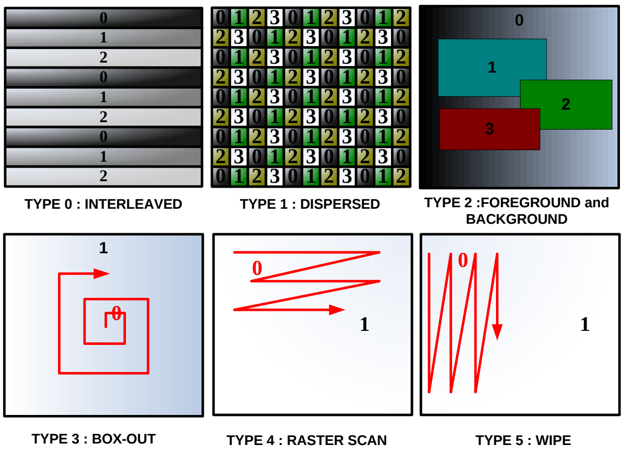


Fig. 1. Types of H.264/AVC FMO (Flexible Macroblock Ordering).

An error concealment technique manages to relieve the visual degradation by interpolating the lost or erroneous samples from its spatially (intra) or temporally (inter) correlated blocks [2][3]. Spatial error concealment estimates a pixel of a lost block as a weighted average of correctly received neighboring pixels. Drawbacks of spatial error concealment include the blurring of interior pixels, and creation of artifacts and wrong edges. Temporal error concealment estimates the motion vector (MV) of a lost block from correlated blocks and restores the lost block by motion compensation. Three essential issues are involved in temporal error concealment, namely the size of a block, the formation of candidate motion vectors, and the selection of the best motion vector. Smaller and adaptive block-size selection generally gives better concealment results with added complexity [4]. Typical candidate MVs are the zero motion vector, MVs of spatially neighboring blocks, and MVs of the collocated blocks. The collocated block refers to a block at the same spatial location as the current block in previous or next reference frames. 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.

Depth information is essential in 3D video applications. The 2D-plus-depth representation saves bits for storing 3D data in contrast to the conventional planar (2D) image signals of all views. A pixel in a depth map (shown in Figure 2) indicates the distance from an object in a 3D scene to the viewer (camera). Because all pixels of an object should have similar depth values, we may use depth to identify objects.

Fig. 2. Video-plus-depth representation; left: texture image, right: depth image.

Ali *et al.* [5] proposed a spatial error concealment technique for 2D-plus-depth video. The depth map provides an indication of object boundaries that assist the concealment process. The segmented foreground objects and background are separately processed, by spatial-domain frequency interpolation and extrapolation. For temporal-domain error concealment, it is noted that the candidate motion vectors found by conventional approaches may fail to give satisfactory results when the true motion vector lies outside the candidate set. 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 coded and transmitted, *we estimate the depth map at the decoder directly from the MVs of the received 2D video*. These depth MVs are added to the set of candidate MVs. Compared to the conventional motion compensated error-concealment techniques without depth information, the proposed method provides much better PSNR results. Compared with the approaches where the depth map is explicitly coded and transmitted [6][7], the proposed method suffers only minor PSNR degradation.

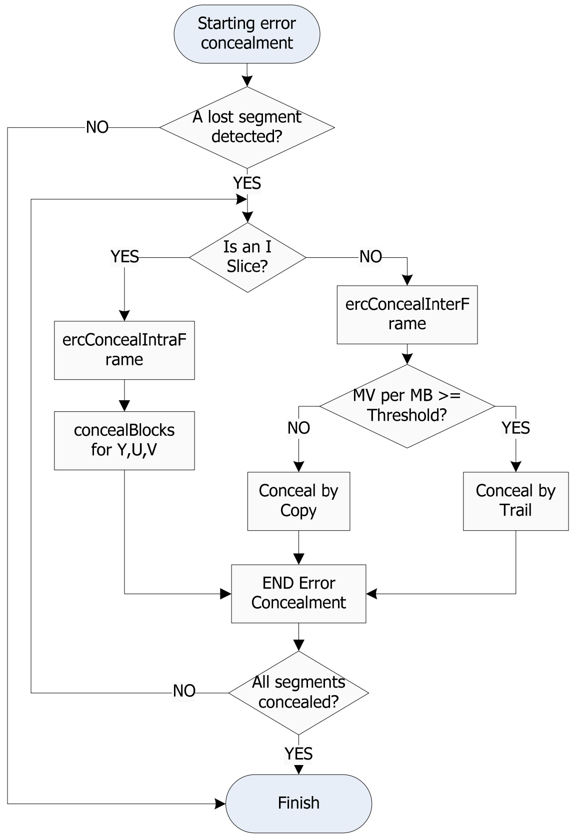
The rest of this paper is organized as follows. Section 2 introduces the error concealment algorithm in the current JM reference software, and other temporal-domain depth-aided error concealment techniques. The proposed depth-enhanced error concealment method is presented in Section 3. Experimental results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

II. Previous Work

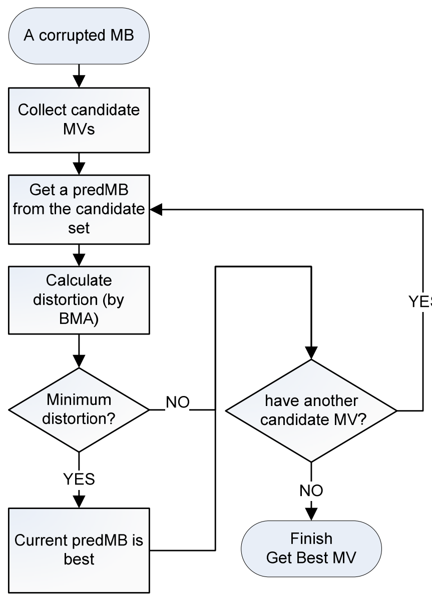
A. Error Concealment in JM

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 the relevant object [8]. However, it is generally difficult and complicated to implement a true motion estimation algorithm in the error situations due to possibly deficient information (pixels, MVs). In the following, we introduce the block-matching methods that find MVs in the distortion-minimization sense.

We first explain the error concealment algorithm in the current JM (Joint Model) reference software [9]. The overall flowchart for JM slice error concealment is shown in Figure 3 (a). For a lost I slice, intra concealment is used. That is, 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) of size 16×16 plus the zero MV as the candidate concealed MVs. The MV with the smallest cost by the BMA (Boundary Match Algorithm) is chosen as the concealed MV, as illustrated in Figure 3(b). BMA calculates the block difference as the sum of absolute differences from inside pixels of a block to be concealed, as shown in Figure 4. Note that when the number of available MVs is few and no reliable MV exists, the found concealment block may cause severe degradation.



(a)



(b)

Fig. 3. (a) Flowchart of the JM error concealment for a lost slice, (b) evaluation of a candidate concealed MV.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

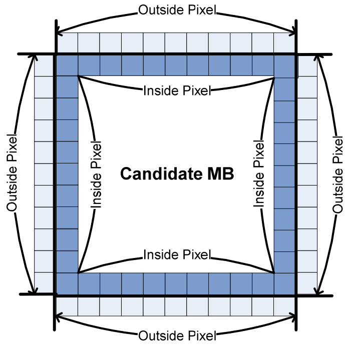


Fig. 4. BMA (Boundary Match Algorithm).

B. 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 the spatially neighboring blocks and collocated block, a new MV derived from depth-map motion estimation is incorporated in the set of candidate MVs. Also, the MVs of neighboring MBs with dissimilar depth values to the depth of the lost MB will be excluded from the candidate set. The above argument is based on the observation that blocks within an object should have similar MVs. In the case that both texture and depth MBs are lost, the MVs of the neighboring MBs in the depth map will be used instead. The best MV for concealment is finally found by BMA. The simulation scenario in [6] assumes random loss of MBs, which may be impractical for current H.264 application where slices are packetized into NAL units.

In [7], another depth-based temporal 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 candidate set of initial MVs. The distortion associated with an MV is the sum of the SAD (sum of absolute differences) resulting from the 2D (texture) image and the SAD resulting from the depth image. Furthermore, a lost MB is classified as either homogeneous or boundary, according to the depth features of received depth maps. A homogeneous MB will be concealed as a whole. A boundary MB, however, will be segmented into foreground and background, and then individually concealed. Since DTEC distinguishes foreground and background that potentially have different motion, it can provide better concealment performance. However, DTEC will fail to give good results when the depth information is missing at the same location as the 2D video. Note that the above depth-based error concealment methods [6][7] require the depth map to be encoded and transmitted, which is not applicable for pure 2D-video sequences. It should also be noted that the generation of depth maps in the 2D-plus-depth scenario is usually very involving [10].

III. Proposed Method

The major contribution of this paper is the inclusion of depth MVs without explicit depth maps and complicated depth estimation. This algorithm yields good visual quality with minimal increase of computational complexity. The flowchart of the proposed method is shown in Figure 5. Three major modifications are highlighted and will be detailed in the following subsections. The first stage calculates the depth value in units of 4x4 blocks from the known MVs. The second stage finds new candidate motion vectors (depth MVs) from the depth maps. The third stage finds the best MV from the candidate set, which is the union of JM MVs and depth MVs.

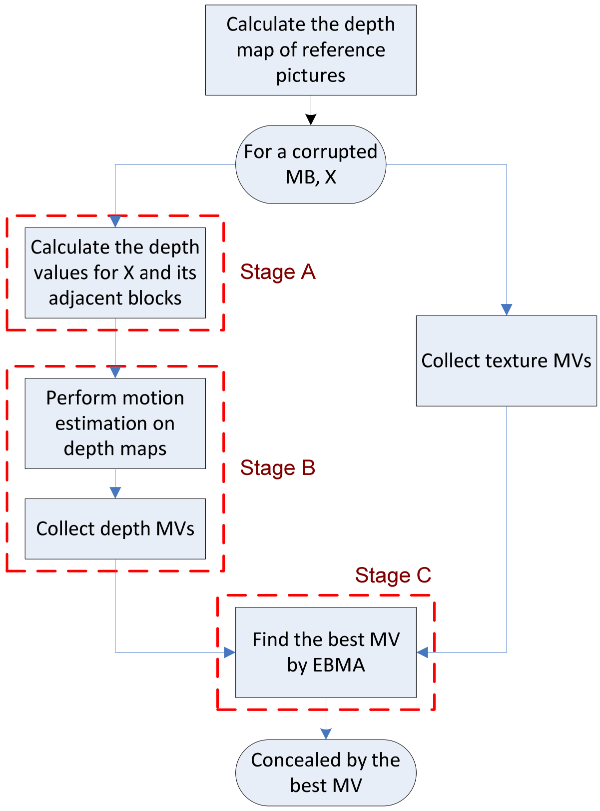


Fig. 5. Flowchart of the proposed depth-enhanced error concealment.

A. Depth Calculation

We exploit the motion parallax for depth calculation, similar to what is done in [11]. The motion displacement usually decreases as the distance of the object to the viewer increases. That is, an object close to an observer will have a larger displacement, which is especially true for the case of a moving camera with static scenes. Note that the MVs have been estimated at encoding and transmitted to the receiver for H.264 video. Therefore, we may directly use the MVs in the received reference pictures for depth calculation.

In H.264, the MV is given for every 4x4 block (a larger prediction unit assumes the same MV for all its constituent blocks). The depth of a block (in reference pictures) is found by

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

where $MV(i, j)$ is the motion vector at location (i, j), c is a scaling constant, and the subscript x and y represent the horizontal and vertical components, respectively [11]. The depth maps of received pictures will be calculated and stored. For a block in a missing slice, the depth of a lost block is estimated as

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

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

where the subscript *t*-1 and *t*+1 refer to the previous and next reference pictures. This method for depth estimation is very simple.

B. Finding and Collecting Depth MVs

We use motion estimation in the depth maps to obtain depth MVs. The SAD of depth values is used as the distortion measure, as shown in (3) and (4),

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

$$SAD(i,j)=\sum_{k=-R}^{R-1}\sum_{l=-R}^{R-1}\left|D_n(i,j)-D_{n+1}(i+k,j+l)\right| \tag{4}$$

where $D_k(i, j)$ denotes the depth value at location (i, j) for frame k and R is the search range. In equations (3) and (4), we assume that a slice in frame n is lost. Thus, $D_n(i, j)$ is calculated by (2) and $D_{n-1}(i, j)$ and $D_{n+1}(i, j)$ are calculated by (1). The first attempt is to find depth MVs from Eq. (3). Eq. (4) will be used instead if the previous frame is lost but the next frame is available.

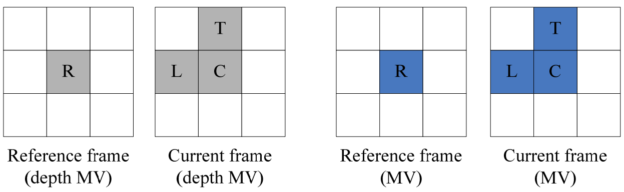


Fig. 6. Candidate set of motion vectors.

To increase the matching accuracy, we include the MV of the collocated block, and MVs to the left and on the top of the lost block in the candidate set, as shown in Figure 6. Therefore, there are at most eight motion vectors in the candidate set, the zero MV, 3 JM MVs, and 4 depth MVs. The JM MVs are the MVs associated with top and left blocks of the current frame and the collocated block in the previous frame. The depth MVs are obtained from depth maps with the same positions as JM MVs, but also adding the MV associated with the missing block (whose depth is estimated by (2)). Note that 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 assume better availability of adjacent blocks, we employ the FMO Type 1 (dispersed) as the error resilience tool. Four slice groups are formed within each frame, as shown in Figure 7.

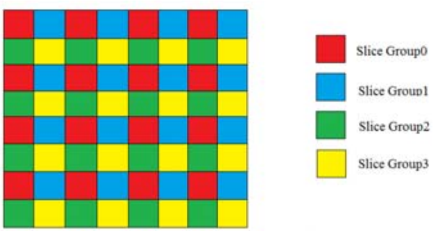


Fig. 7. FMO used in the simulation

C. Determining the best MV

The final MV used for concealing the lost block is obtained by EBMA (Extended Boundary Matching Algorithm). EBMA evaluates the distortion by the sum of absolute differences from the outside pixels of blocks in the current frame and reference frame, as shown in Figure 8 [12].

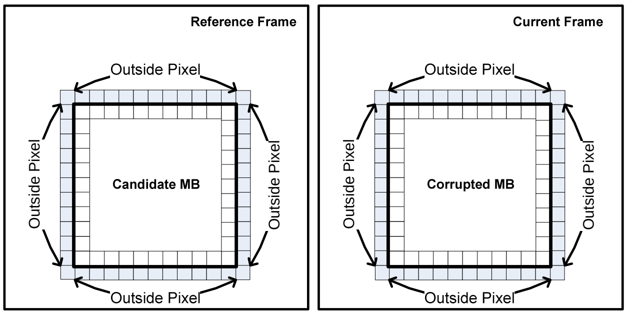


Fig. 8. EBMA (Extended Boundary Matching Algorithm).

IV. Experimental Results and Analyses

The experimental setups of the H.264/AVC codec under investigation are shown in Table 1. The first 100 frames of test sequences are used for simulation. The Baseline Profile (IPPP… structure) is used with five different QP values (20, 24, 28, 32, 36). The dispersed FMO with four slice groups are formed and each slice group is packetized into an NALU/RTP packet. We assume the packet loss rate (PLR) to be 5%, 10%, and 15%. Twenty random patterns are tested for each PLR and the average PSNR is taken as the final result.

TABLE I
H.264/AVC Encoding Parameters

Coding Parameters	Values
CODEC	JM16.2
Profile	Baseline
Frame Rate	30.0
Intra Period	30
Enable IDR	enabled
Number of Frames to Encode	100
Number of Reference Frames	1
Motion Vector Resolution	Full Pixel
Search Range	16
Inter Partition	16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4
Intra Partition	16x16, 4x4
RDO	off
Fast Motion Estimation	on
Fast Mode Decision	on
Entropy Coding	CAVLC
FMO TYPE	Dispersed, 4 slice groups
QP	20, 24, 28, 32, 36

We first compare our method with the JM reference software, and the quantitative results are shown in Table 2. It is observed that the proposed method provides consistent better results by 1 to 3 dB gain margins. The gap is more significant for smaller QP values (high-quality video). A larger PLR causes a little increase in PSNR gap. In Figure 9, the concealed frames are shown for comparison. 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 for faces and hands that have significant movement. For the Foreman and Stefan sequences, notable improvement is found for the proposed method in regions containing edges. The added depth information successfully identifies the foreground objects in these cases.

TABLE II
PSNR PERFORMANCE (IN DB). THE AVERAGE IN LAST ROWS ARE 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

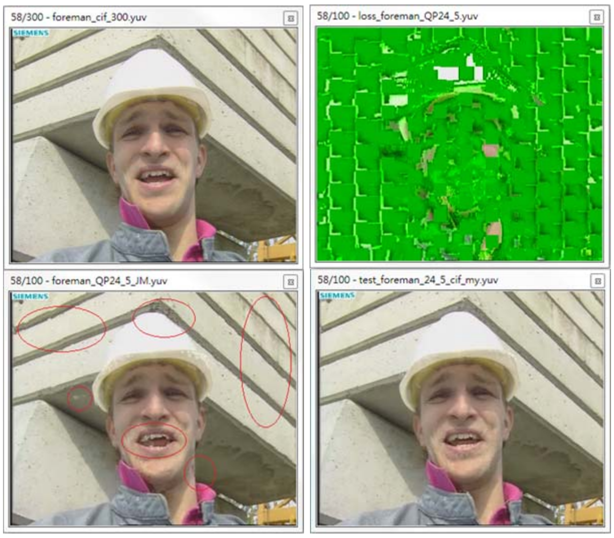
QP	Algorithm	PSNR under PLR		
		5%	10%	15%
20	JM	29.73	26.60	24.07
	Proposed	33.06	30.03	27.50
24	JM	28.92	26.03	23.55
	Proposed	32.07	29.21	27.11
28	JM	28.00	25.54	23.17
	Proposed	30.71	28.37	26.46
32	JM	26.99	24.55	22.54
	Proposed	28.83	27.09	25.53
36	JM	25.26	23.38	21.57
	Proposed	26.56	25.40	24.24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

(C) PARIS

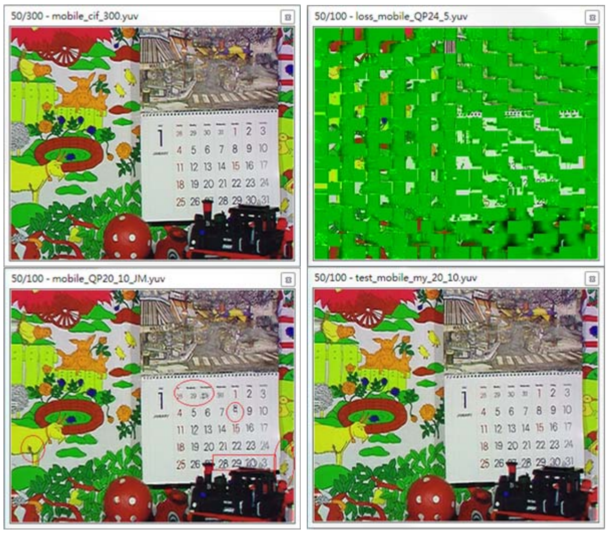
QP	Algorithm	PSNR under PLR		
		5%	10%	15%
20	JM	33.25	30.16	28.14
	Proposed	36.12	33.46	31.09
24	JM	32.53	30.01	27.91
	Proposed	34.88	32.69	30.43
28	JM	31.40	28.92	27.16
	Proposed	33.47	31.64	29.60
32	JM	30.19	28.04	26.48
	Proposed	31.51	30.23	28.45
36	JM	28.07	26.66	25.47
	Proposed	28.98	28.15	27.08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

(D) STEFAN

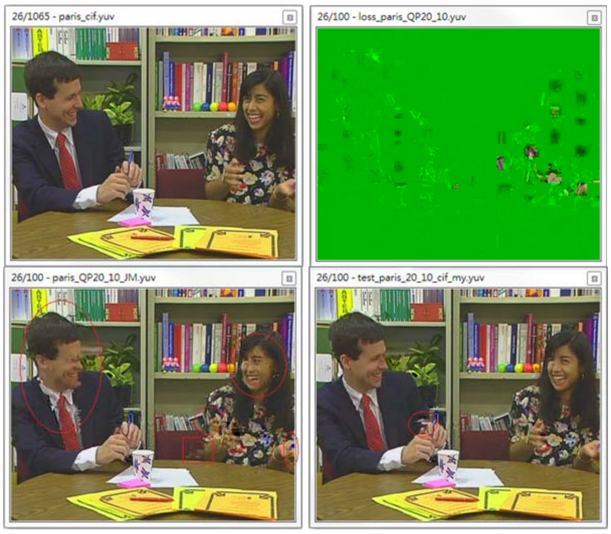
QP	Algorithm	PSNR under PLR		
		5%	10%	15%
20	JM	31.41	28.45	26.33
	Proposed	33.06	30.10	27.74
24	JM	30.99	28.06	25.83
	Proposed	31.91	29.48	27.27
28	JM	29.53	27.18	25.23
	Proposed	30.87	28.83	26.69
32	JM	28.30	26.38	24.70
	Proposed	29.23	27.61	25.80
36	JM	26.88	25.32	23.76
	Proposed	27.52	26.28	24.92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



(a) Foreman (Packet Loss Rate = 5%, QP = 24)



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



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

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

Fig. 9. Snap-shots of error-concealment results. Top-left: original frame, top-right: decoded frame without error concealment, bottom-left: concealed by JM, bottom right: concealed by the proposed method.

TABLE III
PSNR COMPARISON WITH REF. [6] (TEST SEQUENCE: INTERVIEW, I SLICE QP = 28, P SLICE QP = 32, PLR = 10%).

Algorithm	PSNR
JM	35.89
DBMA [6]	37.02
Proposed Method	36.78

TABLE IV
PSNR COMPARISON WITH REF. [7] (TEST SEQUENCE: ORBI, I SLICE QP = 34, P SLICE QP = 36).

	PSNR (PLR = 5%)	PSNR (PLR = 10%)
JM	33.66	32.10
DTEC [7]	33.83	33.71
Proposed Method	34.09	33.32

We also compare our method with the depth-map coded methods DBMA [6] and DTEC [7], and the results are given in Tables 3 and 4. The simulation condition is adjusted for the proposed method to be the same as those benchmarking approaches. Both DBMA and DTEC assume that a separate depth map is sent along with the 2D (texture) images. DBMA assumes that the macroblocks for texture images and depth maps are lost at random. DTEC assumes that slices are randomly lost, including the slices containing the depth map. Only small PSNR degradation (less than 0.4 dB) is observed for the proposed method. DTEC may notably have *worse* PSNR performance than the proposed method (at PLR = 5%). Since the additional depth map is not required in the proposed method, the small PSNR gap is reasonable and acceptable.

We also investigate the contribution of depth MVs in error concealment. The percentage of depth MVs to be selected as the best MV varies drastically from a few percent (for Paris) to forty percent (for Mobile). It is speculated that video sequences with complex scenes and high-motion characteristics will benefit more from incorporating depth MVs. The QP value also affects the percentage of depth MVs to be selected, where gradual percentage decrease is found for an increased QP value. These observations are consistent with the results in Table 2.

V. CONCLUSION

A new depth-enhanced error concealment technique for H.264 was presented. The proposed method estimates the depth value of a block from the received motion vectors. These depth values are then used to derive more candidate motion vectors for temporal error concealment. With the added depth information, foreground objects may be identified and better concealed. Experimental results substantiate the superiority of the proposed method over JM. Compared to the methods inherently incorporating the depth maps, the proposed method has minor PSNR gap but with much simpler implementations.

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] J. Wu, X. Liu, and K.Y. Yoo, “A temporal error concealment method for H.264/AVC using motion vector recovery,” *IEEE Trans. Consumer Electronics*, vol. 54, no. 4, pp. 1880-1885, Nov. 2008.

[3] D. Kim, S. Yang, and J. Jeong, “A new temporal error concealment method for H.264 using adaptive block sizes,” in *Proc. IEEE ICIP*, vol. 3, pp. 928-931, Sep. 2005.

[4] S.-H. Yang and J.-C. Tsai, “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 2010.

[5] A. Ali, H. A. Karim, N. A. Mohd Arif, and A. Sali, “Depth image-based spatial error concealment for 3-D video transmission,” *IEEE Student Conference on Research and Development*, pp. 421-425, Dec. 2010.

[6] B. Yan, “A novel H.264 based motion vector recovery method for 3D video transmission,” *IEEE Trans. Consumer Electronics*, vol. 53, no. 4, pp.1546-1552, Nov. 2007.

[7] Y. Liu, J. Wang, and H. Zhang, “Depth image-based temporal error concealment for 3-D video transmission,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 4, pp. 600-604, April 2010.

[8] G. de Haan, P. W. Biezen, H. Huijgen, and O. A. Ojo, “True-motion estimation with 3-D recursive search block matching,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 5, pp. 368-379, Oct. 1993.

[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* (CVPR), pp. I-74 - I-81, June-July 2004,.

[11] I. Ideses, L. Yaroslavsky, and B. Fishbain, “Depth map manipulation for 3D visualization,” *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pp. 337-340, May 2008.

[12] W.-M. Lam, A. R. Reibman, and B. Liu, “Recovery of lost or erroneously received motion vectors,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. V, pp. 417-420, April 1993.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)