# AN ERROR RESILIENT CODING SCHEME FOR H.264 VIDEO TRANSMISSION BASED ON DATA EMBEDDING+

*Li-Wei Kang and Jin-Jang Leou*

Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan
Email: {lwkang, jjleou}@cs.ccu.edu.tw

**ABSTRACT**

For entropy-coded H.264 video frames, a transmission error in a codeword will not only affect the underlying codeword but also may affect subsequent codewords, resulting in a great degradation of the received video frames. In this study, an error resilient coding scheme for H.264 video transmission is proposed. At the encoder, for an H.264 I frame, the important data for each macroblock (MB) are extracted and embedded into the next frame by the proposed MB-interleaving slice-based data embedding scheme for I frames. For an H.264 P frame, two types of important data for each MB are extracted and embedded into the next frame by the proposed MB-interleaving slice-based data embedding scheme for P frames. At the decoder, if the important data for a corrupted MB can be correctly extracted, the extracted important data for the corrupted MB will facilitate the employed error concealment scheme to conceal the corrupted MB; otherwise, the employed error concealment scheme is used to conceal the corrupted MB. Based on the simulation results obtained in this study, the proposed scheme can recover high-quality H.264 video frames from the corresponding corrupted video frames up to a video packet loss rate of 20%.

## 1. INTRODUCTION

For entropy-coded H.264 video frames [1], a transmission error in a codeword will not only affect the underlying codeword but also may affect subsequent codewords, resulting in a great degradation of the received video frames. To cope with the synchronization problem, each of the two top layers of the H.264 hierarchical structure [1], namely, picture and slice, is ahead with a fixed-length start code. After the decoder receives any start code, the decoder resynchronizes regardless of the preceding slippage. However, a transmission error may affect the underlying codeword and its subsequent codewords within the corrupted slice. Moreover, because of the use of motion-compensated interframe coding, the effect of a transmission error may be propagated to the subsequent video frames.

In general, error resilient approaches include three categories [2], namely, (1) the error resilient encoding approach, (2) the error concealment (EC) approach [3]-[4], and (3) the encoder-decoder interactive error control approach. Recently, several error resilient coding approaches based on data embedding are proposed [5]-[7], in which some important data useful for EC performed at the decoder can be embedded into video frames at the encoder. At the decoder, the embedded data for the corrupted MBs are extracted and used to facilitate EC performed at the decoder. This paper is organized as follows. Error concealment for H.264 video transmission is given in Section 2. The proposed scheme is addressed in Section 3. Simulation results are included in Section 4, followed by concluding remarks.

## 2. ERROR CONCEALMENT FOR H.264 VIDEO TRANSMISSION

In this study, the spatial EC algorithm for I frames in H.264 [4] is employed, in which each pixel value in a corrupted MB can be concealed by a weighted sum of the closest boundary pixels of the selected four-connected neighboring MBs. The weight associated with each boundary pixel is relative to the inverse distance between the pixel to be concealed and the closest boundary pixel.

In this study, the two employed EC schemes for P frames are based on the motion-compensated best neighborhood matching (BNM) algorithm [3]. Each corrupted block of size $M \times M$ is extracted from a video frame together with its neighborhood as a range block of size $(M+m)\times(M+m)$. Within a range block, all the pixels in the corrupted region belong to the lost part and the others belong to the good part. After a range block is extracted, an $H\times L$ searching range block centralized with

+This work was supported in part by National Science Council, Republic of China under Grants NSC 91-2213-E-194-025 and NSC 92-2213-E-194-038.

the motion-compensated block in the reference frame is generated. Here, the motion vector (MV) of a corrupted block is recovered as the average of the MVs of the "believable" (correctly received or concealed) neighboring blocks of the corrupted block. Each $(M+m)\times(M+m)$ block in the searching range block may be a candidate domain block to recover the lost part of the range block (the corrupted block). For each candidate domain block, the mean absolute error (MAE) between the good part of the range block and the corresponding good part of the candidate domain block is evaluated. The candidate domain block with the minimum MAE (the best domain block) is then used to conceal the lost part of the range block by copying its corresponding central part to the lost part of the range block.

The first employed EC scheme is the Zero-MV BNM algorithm, in which $S$ candidate domain blocks with zero MVs in the $S$ reference frames of a corrupted P frame are evaluated. The second EC scheme is employed by applying the motion-compensated BNM algorithm [3] to $S$ reference frames.

## 3. PROPOSED SCHEME

In the proposed scheme, the following issues will be addressed: (1) what kind of important data for the MBs within a video frame should be extracted and embedded, (2) where should the important data be embedded, (3) how to embed the important data to the corresponding "masking" slices, and (4) how to extract and use the important data to facilitate the employed EC schemes to conceal the corrupted MBs at the decoder.

### 3.1. Error resilient coding for H.264 I frames

In this study, similar to [5], for an H.264 I frame, the edge direction within an MB is extracted as its important data. For a pixel $f(i, j)$ in an MB, the two gradient components, $G_x(i, j)$ and $G_y(i, j)$, of $f(i, j)$ is computed by the Sobel operators [8]. The magnitude, $|\nabla f(i,j)|$, and the direction, $\theta(i, j)$, of the gradient can be computed as

$$|\nabla f(i,j)| = |G_x(i,j)| + |G_y(i,j)|, \quad (1)$$

$$\theta(i,j) = arctan(G_y(i,j)/G_x(i,j)). \quad (2)$$

If $|\nabla f(i,j)|$ is greater than a predefined threshold, $t$, $f(i,j)$ is determined as an edge pixel. If an MB contains no edge pixels, it is determined as a "smooth block;" otherwise, it is determined as an "edge block." For an edge block, the direction for each edge pixel is quantized to one of the $E$ equally-spaced directions between $0^\circ$ and $180^\circ$ and the dominated edge direction is used as its edge direction. Hence, one bit is used to denote the MB type (smooth or edge) and $\lceil log_2 E \rceil$ bits are used to denote its edge direction, i.e., the important data for a "smooth" MB contain only 1 bit and the important data for an "edge" MB contain $1 + \lceil log_2 E \rceil$ bits.

The important data for each MB in an I frame will be embedded by using the proposed MB-interleaving slice-based data embedding scheme, in which the important data for all the MBs in a slice will be embedded into two corresponding "masking" slices in the next video frame. Because a slice and its masking slices may be corrupted simultaneously, the important data for a slice should not be completely embedded into only one masking slice. Hence, for example, the important data for the even-number (odd-number) MBs of the first slice and those for the odd-number (even-number) MBs of the third slice can be interleaved and concatenated to a mixed bitstream, which is embedded into one (another) "masking" slice in the next video frame.

To perform data embedding in an I frame, the odd-even data embedding scheme [5] is employed and applied on the non-zero quantized integer DCT coefficients in the corresponding masking slice. If the data bit to be embedded is "0," the DCT coefficient will be forced to be an even number, whereas if the data bit to be embedded is "1," the DCT coefficient will be forced to be an odd number. If the data bit to be embedded is $b_j$, the DCT coefficient $C_i$ of the odd-even data embedding scheme is determined as

$$C_i = \begin{cases} C_i + 1 & if\ C_i\ mod\ 2 \neq b_j,\ and\ C_i > 0, \\ C_i - 1 & if\ C_i\ mod\ 2 \neq b_j,\ and\ C_i < 0, \\ C_i & otherwise. \end{cases} \quad (3)$$

In this study, the flexible macroblock ordering (FMO) capability in H.264 [1] is enabled, in which the MBs in a slice can not be in a raster scan order so that the neighboring MBs of a corrupted MB may not be corrupted simultaneously, i.e., within a video frame, the even-number MBs in a row and the odd-number MBs in another row can form a slice. For example, the even-number MBs in the first row and the odd-number MBs in the second row can form the first slice.

At the decoder, the masking slice for a corrupted MB is determined, and its important data will be extracted if the masking slice is correctly received. Then, if the corrupted MB is an edge block, it will be concealed by using the bilinear interpolation with the edge direction and the two corresponding boundary pixels in the neighboring MBs. But if (1) the corrupted MB is a smooth block, (2) any of the two boundary pixels is not available, or (3) its masking slice is also corrupted, the corrupted MB will be concealed by the spatial interpolation algorithm in H.264 [4].

### 3.2. Error resilient coding for H.264 P frames

For an MB in a P frame, two types (Type-I and Type-II) of important data will be extracted and embedded. The Type-I data for an MB contains the coding mode, the reference frame(s), and the MV(s) for the MB, whereas the Type-II data for the MB includes the best EC scheme among 15 evaluated EC schemes for the MB.

For the Type-I data of an MB, one bit is used to denote whether the Type-I data will be embedded. If the Type-I data will be embedded, one bit is used to denote its coding mode (*skip* or *inter-coded*). Note that the rarely-used intra-coded mode is ignored here. For an inter-coded MB, 2 bits are used to denote which inter-coded mode (*inter-16×16*, *inter-16×8*, or *inter-8×16*) is used. Any of the other inter-coded modes containing many MVs is ignored here. For an MB (*inter-16×16*) or subblock (*inter-16×8* or *inter-8×16*), 2 bits are used to denote the reference frame (3 reference frames are used here, i.e., $S = 3$) and one bit is used to denote whether the MV is a zero MV. For an MB or subblock with a non-zero MV, 16 bits are used to denote one MV with the search range set to ±16 pixels (quarter pixel accuracy). Note that for an ignored intra-coded MB or an ignored inter-coded MB, only one bit denoting that no Type-I data is embedded.

For the Type-II data of an MB, 15 simple EC schemes will be evaluated at the encoder and the best one among 15 is extracted as its important data (4 bits). The 15 EC schemes are as follows. (1) Three zero-MV techniques, in which the three MBs at the corresponding spatial locations in the three reference frames are copied to conceal the corrupted MB, respectively. (2) Three average-MV techniques. If the 8-connected spatially neighboring MBs of the corrupted MB are denoted by $B_i$, $1 \le i \le 8$ and 3 reference frames are used, i.e., $S = 3$, all the MVs of $B_i$'s will locate on the three reference frames. Then, the average MV, $mv_{s,av}$, on the $s$th reference frame of the MVs of the 8-connected spatially neighboring MBs of the corrupted MB can be denoted as $mv_{s,av}$, $s = 1, 2, 3$. The three motion-compensated MBs with the three MVs, $mv_{s,av}$, $s = 1, 2, 3$, are used to conceal the corrupted MB, respectively. (3) Assume that the coding mode of a corrupted MB is *inter-16×8* and the top (bottom) 16×8 subblock can be concealed by the three average-MV techniques with $mv_{s,av}$, $s = 1, 2, 3$, which are evaluated over the half top (bottom) of the 8-connected "believable" spatially neighboring MBs (subblocks), respectively. Hence, there are nine possible combinations, i.e., nine EC schemes. As a summary, the MAEs between the error-free MB and the 15 "pre-concealed" MBs using the 15 EC algorithms are evaluated. The best EC scheme with the minimum MAE is finally extracted as the Type-II data for the MB.

For an MB in a P frame, the important data will be embedded by using the proposed MB-interleaving slice-based data embedding scheme, in which the important data for all the MBs in a slice will be embedded into the four corresponding "masking" slices in the next frame. Hence, for example, the Type-I data for the even-number MBs of the first slice, the Type-I data for the odd-number MBs of the third slice, the Type-II data for the even-number MBs of the fifth slice, and the Type-II data for the odd-number MBs of the seventh slice are interleaved and concatenated to a mixed bitstream, which is embedded into its masking slice in the next frame. Here, the odd-even data embedding scheme [5] is also employed to embed the important data. For an MB in a P frame, the data size of the Type-I data is at most 42 bits, which is sometimes too large to be embedded. Hence, before embedding the Type-I data, the priority for an MB will be determined by the MAE between the error-free MB and the "pre-concealed" MB by using the corresponding best EC scheme. The larger the MAE is, the higher the priority of the MB will be. Then, the Type-I data for the two MBs with the lowest priorities will be ignored, i.e., replaced by one bit denoting that no Type-I data is embedded. However, the Type-II data for each MB will be always embedded.

At the decoder, the masking slice for a corrupted MB is determined and its important data will be extracted if its masking slice is correctly received. Then, if the coding mode is *skip*, the corresponding MB in the previous frame is used to conceal the corrupted MB. If the coding mode is *inter-coded*, the coding mode, reference frame(s), and MV(s) are together used to conceal the corrupted MB. If (1) no Type-I data is embedded or (2) the Type-I data cannot be correctly extracted and the Type-II data are available, the best EC scheme is used to conceal the corrupted MB. If no important data is available, the average MV, $MV_{av}$, of the MVs of the "believable" 8-connected neighboring MBs of the corrupted MB is computed. If the magnitude of $MV_{av}$, $|MV_{av}|$, is smaller than the predefined threshold, $T_{MV}$, the corrupted MB is concealed by the Zero-MV BNM algorithm (select the best one on 3 reference frames). Otherwise, the motion-compensated BNM algorithm with $mv_{s,av}$, $s = 1, 2, 3$ on 3 reference frames (select the best one) is used to conceal the corrupted MB. Note that before concealing a corrupted MB by using the Type-II data or the employed EC scheme for P frames, its 8-connected neighboring corrupted MBs can be concealed first if they can be concealed with the correctly extracted Type-I data. Then, the corrupted MB can be concealed by using the best or employed EC scheme with more neighboring MB information.

## 4. SIMULATION RESULTS

Several QCIF video sequences with a frame rate 30fps and different video packet loss rates (*VPLR*) are used to evaluate the performance of the proposed scheme. To prevent bit rate increasing due to data embedding, in the proposed scheme, the quantization parameter (*QP*) for P frames are set to 30 (default $QP = 28$). The average peak signal to noise ratio of an H.264 video sequence, denoted by $PSNR_{seq}$, is employed as the objective performance measure. In the employed EC scheme for P frames, $M$, $m$, $H$, and $L$ are set to 16, 8, 30, and 30, respectively. Additionally, $t$, $E$, and $T_{MV}$ are set to 300, 32, and 5, respectively. To evaluate the performance of the proposed scheme, five existing approaches for comparison [3]-[4], [6]-[7] are implemented in this study. They are: (1) zero-substitution (denoted by Zero-S); (2) the EC scheme in H.264 JM7.3 (denoted by JM7.3) [4]; (3) the motion-compensated BNM algorithm (denoted by BNM) [3]; (4) the data embedded video coding scheme (denoted by DEVCS) [7]; (5) the error resilient video coding scheme based on data embedding (denoted by ERDE) [6].

In terms of $PSNR_{seq}$ (dB), the simulation results for the "*Carphone*" sequence with different *VPLR*s of the five existing approaches for comparison and the proposed scheme are listed in Table 1. As a subjective measure of the quality of the concealed video frames, the error-free and concealed frames by the five existing approaches for comparison and the proposed scheme for the "*Carphone*" sequence are shown in Fig. 1. Here, the bit rates for the "*Carphone*" sequence obtained by the original H.264 and the proposed scheme are 126.01kbps and 124.19kbps, respectively.

## 5. CONCLUDING REMARKS

Based on the simulation results obtained in this study, two observations can be found. (1) The concealment results of the proposed scheme are better than those of the five existing approaches for comparison. (2) The average degradation of the proposed scheme with data embedding is below 1 dB, which is comparable with those in [6]-[7]. The proposed scheme can recover high-quality H.264 video frames from the corresponding corrupted video frames up to a video packet loss rate of 20%.

## 6. REFERENCES


[1] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264|ISO/IEC 14496-10 AVC)," 2003.

[2] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Processing Magazine*, vol. 17, no. 4, pp. 61-82, July 2000.

[3] L. W. Kang and J. J. Leou, "A new hybrid error concealment scheme for MPEG-2 video transmission," in *Proc. of IEEE Int. Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin Islands, December 2002, pp. 29-32.

[4] Y. K. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," in *Proc. of IEEE Int. Conf. on Image Processing*, 2002, pp. 729-732.

[5] P. Yin, B. Liu, and H. H. Yu, "Error concealment using data hiding," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 2001, vol. 3, pp. 1453-1456.

[6] L. W. Kang and J. J. Leou, "A new error resilient coding scheme for H.263 video transmission," in *Proc. of IEEE Pacific-Rim Conf. on Multimedia*, Hsinchu, Taiwan, December 2002, pp. 814-822.

[7] J. Song, and K. J. R. Liu, "A data embedded video coding scheme for error-prone channels," *IEEE Trans. on Multimedia*, vol. 3, no. 4, pp. 415-423, 2001.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. 2nd Edition, New Jersey: Prentice Hall, 2002.


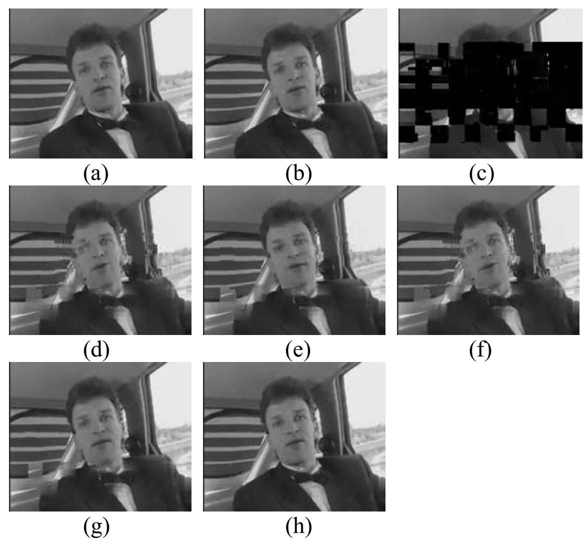


Fig. 1. The error-free and concealed H.264 video frames of a *P*-frame (the 14th frame) within the "*Carphone*" sequence with $VPLR = 10\%$: (a) the error-free frame; (b) the error-free frame with data embedding; (c)-(h) the concealed frames by Zero-S, JM7.3, BNM, DEVCS, ERDE, and the proposed scheme, respectively.

Table 1. The simulation results, $PSNR_{seq}$ (dB), for the "*Carphone*" sequence with different *VPLRs* of the five existing approaches for comparison and the proposed scheme.

| *VPLR* | Without data embedding | | | With data embedding | | |
|---|---|---|---|---|---|---|
| | Zero-S | JM7.3 | BNM | DEVCS | ERDE | Proposed |
| 0% | 38.47 | 38.47 | 38.47 | 37.46 | 37.72 | 37.51 |
| 10% | 8.95 | 30.45 | 31.79 | 30.53 | 33.35 | 35.45 |
| 15% | 8.87 | 29.53 | 30.89 | 29.62 | 32.63 | 34.27 |
| 20% | 8.60 | 27.95 | 29.72 | 27.94 | 31.49 | 33.20 |