

Content-Adaptive Refined Error Concealment Schemes for H.264/AVC Video Coding

Santosh V. Chapaneri
Department of Electronics and
Telecommunication Engineering,
St. Francis Institute of Technology,
University of Mumbai

ABSTRACT

In this paper, we propose content-adaptive refined error concealment schemes for video transmission over packet lossy networks. The proposed spatial and temporal error concealment algorithms improve the reconstructed video quality at the common loss rates encountered on the networks. To reduce the computational complexity, the proposed schemes make use of the existing information of Intra prediction modes from the coded bit-stream for spatial error concealment. Neighboring macroblocks mode information is used to adapt the partition size of the lost macroblock to enhance the temporal error concealment performance. In addition, overlapped motion compensation is used to avoid the spatial discontinuities. A hybrid spatio-temporal concealment scheme is also proposed to use the appropriate algorithm for the lost macroblocks in the Inter frames. Experimental results demonstrate that the proposed schemes improve the concealment performance and reduce the structural degradations.

General Terms

Image Processing, Video Processing

Keywords

Error concealment, H.264/AVC, Intra prediction, OBMC, SSIM, Video quality index

1. INTRODUCTION

The recent video coding standards achieve efficient compression by using the techniques of motion estimation and compensation, discrete cosine transform, and variable-length coding. H.264 or Advanced Video Coding is the international video coding standard jointly contributed by the ITU-T Video Coding Experts Group and the ISO/IEC Moving Pictures Expert Group [1]. Compared to the previous standards, H.264/AVC adopted new coding tools such as Intra prediction, loop-filtering, tree structured motion estimation and compensation and so on. This enables the coded bitstream to have more information such as the parameter sets, Intra prediction modes, and flexible slicing information.

When the compressed video bit-stream is transmitted over the error-prone channels such as a packet lossy network, it is possibly corrupted by the channel noise. All information contained in the lost packets will not be available at the decoder and thus the video reconstruction is of a poor quality. Many error resilience techniques such as resynchronization marker insertion and forward error correction (FEC) have been proposed to add side information in the coded bit-stream at the encoder side to make it more resilient to channel errors. H.264/AVC also employs several new error resilience tools to

combat the channel errors. However, they still cannot guarantee to compensate the channel errors with good reconstruction quality. To reduce the quality degradations and improve the video quality, error concealment techniques are used at the decoder side as a post-processing module.

Error concealment has the advantages of not consuming extra bandwidth as FEC and not introducing retransmission delays as automatic retransmission request (ARQ). An error detection mechanism is required prior to error concealment for the location of the error region in the decoded video [2]. In this work, we assume that the error region is located and the focus is on the reconstruction of the lost video data using the error concealment techniques. We propose content-adaptive refined spatial and temporal error concealment algorithms, which conceal the lost region adaptively using the existing information available in the coded bit-stream and the neighboring correctly received regions.

2. PREVIOUS WORK ON ERROR CONCEALMENT

2.1 Spatial Error Concealment (SEC)

SEC methods assume that the video frames are spatially smooth in nature, and thus the lost content is reconstructed by interpolation from the neighboring pixels as proposed in [3], and implemented in the reference software of the H.264/AVC standard [4]. However, this method does not consider the effect of edges in the vicinity of the lost content and therefore, blurring artifacts are observed in the edge regions of the frames. To resolve this problem, it was proposed in [5] to find the edges first and then interpolating along the edge direction. Robie et al. [6] used Hough transform to determine the best operation for either directional interpolation or filtering. The spatial interpolation can also be applied to the spectral domain such as DCT, as proposed in [7]. An iterative procedure called projection onto convex sets (POCS) was proposed in [8] to impose an additional directional constraint in addition to the smoothness constraint. Markov random fields based error concealment method was proposed in [9]. Xu et al. [10] proposed the use of gradient filters to determine the dominant edge direction for the missing content. Best neighborhood matching method was proposed in [11], which exploits block-wise similarities within the frame to replace the missing content. Directional entropy of the neighboring edges was computed in [12] to switch between the directional and bilinear interpolation methods for the concealment of the missing content.

However, most of these existing techniques improve the concealment performance at the cost of high computational

complexity. In Section 3, we propose an algorithm for SEC with a reduced computational complexity and improve the concealment by reducing the structural distortions.

2.2 Temporal Error Concealment (TEC)

TEC methods use the temporal information from the previous or the next frame to conceal the loss of the current frame, and assume the video content to be smooth or continuous in time. Most approaches estimate the motion vectors (MVs) associated with the lost macroblock (MB) and perform motion compensation for the lost MB using the estimated MV. The simplest way to conceal a lost MB is to use a temporal replacement (TR) [5] method, which replaces the lost MB by the spatially corresponding MB in the previous frame. But TR fails when the motion in the video sequence is fast. Lam et al. proposed a boundary matching algorithm (BMA) that selects the reconstructed MV satisfying the best spatial coherence criterion from a set of candidates coming from the MVs of the neighboring MBs [13]. The reference implementation of the H.264/AVC standard uses BMA for temporal concealment based on the spatial smoothness property [4]. Zheng et al. [14] introduced Lagrange interpolation schemes for concealment of the corrupted H.264/AVC video streams. Refined temporal concealment (RTC) using weighted boundary match was presented by Xu et al. [10], in which a lost MB is divided into four blocks and the set of candidate MVs is increased. To exploit the temporal correlation between frames in the estimation of MV, the decoder motion vector estimation (DMVE) algorithm was proposed in [15]. The outer boundary matching algorithm (OBMA), which is a variant of BMA, was studied in [16] with the conclusion that OBMA performs better than BMA for temporal concealment at the same level of complexity. Chen et al. [17] proposed an approach that combines the overlapped motion compensation [18] and the side match distortion criterion to make the effect of lost MVs imperceptible. Kim et al. [19] proposed an adaptive block size (ABS) method that selects the concealment mode of the lost MB using the neighboring MB modes information.

The above approaches either estimate only one MV for the lost MB, or divide the MB into four blocks to estimate different MVs for different blocks of the lost MB. In the former, noticeable blocking artifacts are observed when the lost MB is in the regions containing fast or rotational movements, zooming, or incoherent motion between different parts of the video sequence. In the latter, division of the lost MB into four blocks may not be helpful for all types of video content. For homogeneous regions, it may be better to use the whole MB size for better concealment. The ABS method restricts the concealment mode of the lost MB to four different block sizes and does not consider the spatial continuity of the concealed MB with its neighbors. In Section 4, we propose an algorithm for TEC in which the MB partitioning size is decided adaptively depending on the neighboring MB mode information.

3. PMEC: PROPOSED SEC ALGORITHM

Intra prediction in H.264/AVC exploits the spatial correlation between the pixels of the neighboring MBs [1]. For the luminance (luma) samples, the prediction can be formed for each 4×4 sub-block or a full 16×16 MB. There are a total of nine optional *pmodes* (0 to 8) for each 4×4 sub-block and four optional *pmodes* (0 to 3) for a full 16×16 MB. For the

chrominance (chroma) samples, each 8×8 chroma block has four optional *pmodes* (0 to 3).

For the concealment of the lost MB in Intra frames, the proposed prediction modes error concealment (PMEC) algorithm uses the information of prediction modes (*pmodes*) of the neighboring MBs already available in the coded bit-stream. The *pmodes* implicitly describe the orientation of the edges, which help in determining the dominant prediction mode (*dpm*) of the lost MB. The *dpm* corresponds to the dominant edge direction of the lost MB. The proposed PMEC algorithm for damaged Intra frames is illustrated using a flowchart in Fig. 1.

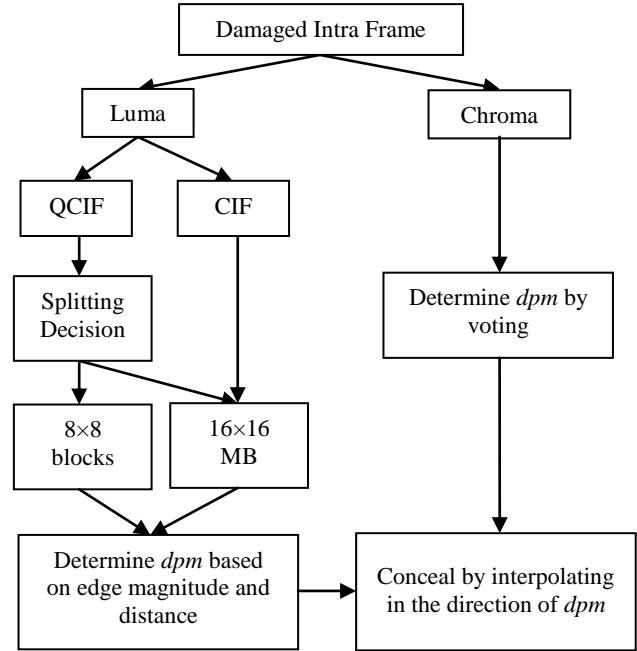


Fig 1: Flowchart of the PMEC Algorithm

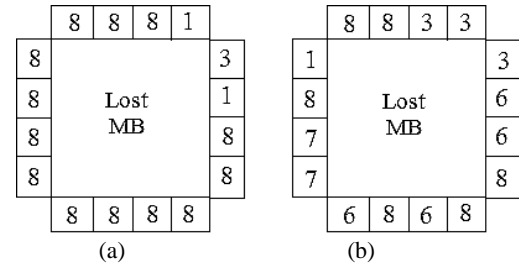


Fig. 2. Splitting decision. (a) Smooth spatial activity, no splitting of the lost MB. (b) High spatial activity, lost MB is split into four 8×8 blocks.

A splitting decision is taken for a lost MB if the video sequence is of the QCIF (176×144) format since splitting a 16×16 MB into four 8×8 blocks improves the concealment performance, especially at the boundaries of the lost MB. If the video sequence is of the CIF (352×288) format, then the concealment is done on the full 16×16 lost MB. The splitting decision is based on the observation of spatial activity of the neighboring MBs. The adjacent *pmodes* of the four 4×4 sub-blocks corresponding to each above, left, below and right neighbor of the lost MB are compared, making a total of twelve

comparisons. If the adjacent *pmodes* differ more than six times, the lost MB is split into four 8×8 blocks. For example, in Fig. 2 (a), we observe a smooth spatial region as the majority of the adjacent *pmodes* are identical, whereas in Fig. 2 (b), we decide to split the lost MB into four 8×8 blocks since the *pmodes* are different for most of the neighboring 4×4 sub-blocks. We avoid splitting the full MB into sixteen 4×4 sub-blocks as it increases the noise sensitivity and also consumes more time for concealment.

The sixteen *pmodes* of the 4×4 sub-blocks corresponding to the neighboring MBs are used to determine the *dpm*. If any neighboring MB is encoded as full 16×16, then each constituent 4×4 sub-block is considered to have the same *pmode* as that of the full MB. This is because the *pmodes* 0 to 3 of the 4×4 sub-blocks case are identical to the *pmodes* 0 to 3 of the full 16×16 MB [1].

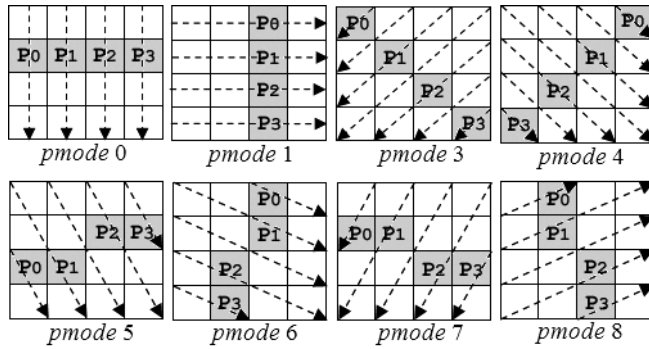


Fig. 3. Selection of four pixel values for computing the edge magnitude of eight *pmodes*.

For the luma component of the lost Intra MB, the *dpm* is determined by computing the edge magnitude of the *pmodes* for all neighboring MBs. The edge magnitude of each 4×4 available neighboring sub-block is estimated as follows:

- Four pixels P_0 to P_3 are selected in a direction perpendicular to that indicated by the corresponding *pmode* as shown in Fig. 3.
- The maximum (max) and the minimum (min) pixel intensities among these four pixels are determined.
- A distance d is determined as the Euclidean distance of the pixels having the max and min values.

The edge magnitude (EdgeMag) is then obtained as

$$\text{EdgeMag} = \frac{\max(P) - \min(P)}{d} \quad (1)$$

where $P = \{P_0, P_1, P_2, P_3\}$ are the local luma pixel intensities. The edge magnitude for *pmode* 2 is zero as this mode does not define any edge. For each *pmode*, we initialize a counter to zero and increase it with the corresponding edge magnitudes of the neighboring sub-blocks having that *pmode*. Thus, we obtain the *dpm* for the luma component as the *pmode* having the maximum total edge magnitude, i.e. the maximum counter value.

For the chroma component of the lost Intra MB, the *dpm* is selected as the *pmode* that occur the most number of times among the neighboring chroma MB components.

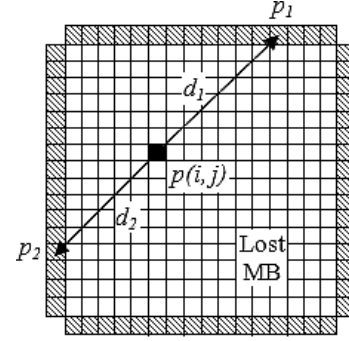


Fig. 4. Directional interpolation by extending the detected edge direction specified by *dpm* for concealment.

Traveling along the dominant edge direction as specified by the *dpm*, the detected edge is extended from each pixel in the lost MB to the neighboring pixels on the one-pixel-wide outer boundary of the lost MB as shown in Fig. 4. The concealment of lost pixels is then done using interpolation [20] along the extended edge as

$$p(i, j) = \frac{p_1 d_2 + p_2 d_1}{d_1 + d_2}; \quad \forall (i, j) \in \text{lost MB} \quad (2)$$

where p_1 and p_2 are the boundary pixel values and d_1 and d_2 are the corresponding distances from the desired pixel $p(i, j)$.

4. CAMP: PROPOSED TEC ALGORITHM

The proposed content-adaptive macroblock partitioning (CAMP) algorithm first determines the partition type of the lost MB to ensure a smoother concealment. Each partition is then temporally concealed with OBMA, as shown in Fig. 5, as the distortion measure.

The candidate set of eight motion vectors (MV_0, MV_1, \dots, MV_7) of the 8×8 blocks corresponding to the four neighboring MBs and the zero motion vector (ZMV) is used in the distortion computations. The candidate MB in the reference frame specified by the MV that yields the maximum boundary smoothness is selected for the temporal concealment of the lost MB. OBMA is preferred over BMA since it preserves the edge continuity between the lost MB and its neighbors [16]. After this initial concealment of the lost MB, post-processing is done using overlapped block motion compensation to remove the spatial discontinuities.

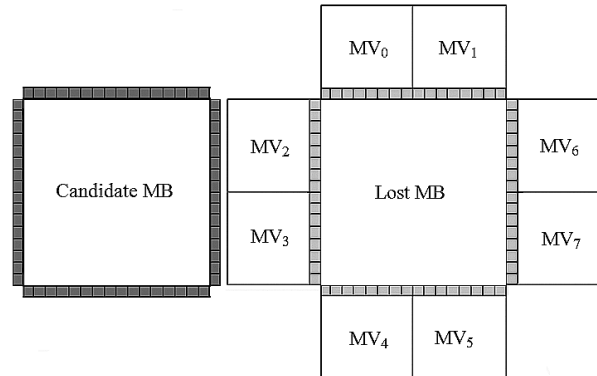


Fig. 5. OBMA technique for the lost MB using eight MVs of the four neighboring MBs. The candidate MB is obtained from the reference frame using these MVs.

4.1 Steps of the CAMP Algorithm

The proposed algorithm for temporal error concealment is described in the following steps:

1. Adaptively select the partition type of the lost MB (one out of eight possible types) depending on the macroblock mode information of the available neighboring MBs.
2. Conceal the individual partitions with OBMA as the distortion measure. This gives the initial concealed MB.
3. Split the initial concealed MB into four 8×8 blocks.
4. Post-process each of these 8×8 blocks using OBMC to avoid the spatial discontinuities. This gives the final concealed MB.
5. Repeat steps 1 to 4 for all lost MBs of the Inter frames.

4.2 Macroblock Partitioning

In H.264/AVC, a MB of the Inter frame can be encoded in the following modes: 16×16, 16×8, 8×16, and 8×8 [1]. If the 8×8 block is further partitioned, then the ‘split-and-merge’ technique is used to determine its MV, i.e. the MV of the 8×8 block is the mean of the MVs of its partitioned sub-blocks. Using the MB mode information of the four available neighboring MBs, we determine the most suitable partition type of the lost MB from the eight possible types shown in Fig. 6.

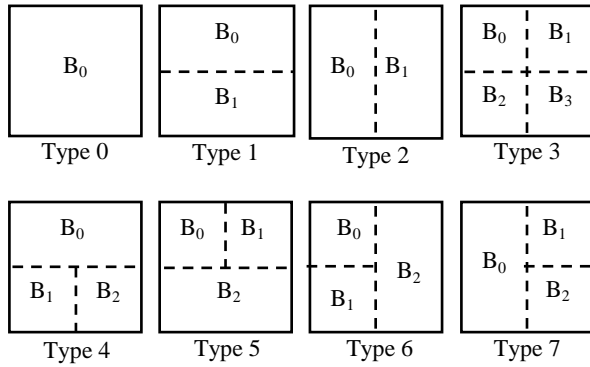


Fig. 6. Eight MB partitions selected adaptively.

The eight partition types for a lost 16×16 MB are determined adaptively as follows:

Type 0: If the modes of both above and below neighboring MBs are not 8×8 or 8×16, and the modes of both left and right neighboring MBs are not 8×8 or 16×8, then Type 0 is selected and the lost MB is concealed as one 16×16 block. This type is identical to the JM test model [4]. The MV candidates used for this type to conceal the lost MB are

$$B_0: \{MV_0, MV_1, \dots, MV_7, ZMV\} \quad (3)$$

Type 1: If the modes of both above and below neighboring MBs are not 8×8 or 8×16, and the modes of both left and right neighboring MBs are 8×8 or 16×8, then Type 1 is selected and the lost MB is concealed as two 16×8 blocks. The MV candidates used for this type to conceal the lost MB are

$$\begin{aligned} B_0: \{MV_0, MV_1, MV_2, MV_6, ZMV\} \\ B_1: \{MV_3, MV_4, MV_5, MV_7, ZMV\} \end{aligned} \quad (4)$$

Type 2: If the modes of both above and below neighboring MBs are 8×8 or 8×16, and the modes of both left and right

neighboring MBs are not 8×8 or 16×8, then Type 2 is selected and the lost MB is concealed as two 8×16 blocks. The MV candidates used for this type to conceal the lost MB are

$$\begin{aligned} B_0: \{MV_0, MV_2, MV_3, MV_4, ZMV\} \\ B_1: \{MV_1, MV_5, MV_6, MV_7, ZMV\} \end{aligned} \quad (5)$$

Type 3: If the modes of both above and below neighboring MBs are 8×8 or 8×16, and the modes of both left and right neighboring MBs are 8×8 or 16×8, then Type 3 is selected and the lost MB is concealed as four 8×8 blocks. The MV candidates used for this type to conceal the lost MB are

$$\begin{aligned} B_0: \{MV_0, MV_2, ZMV\} \quad B_1: \{MV_1, MV_6, ZMV\} \\ B_2: \{MV_3, MV_4, ZMV\} \quad B_3: \{MV_5, MV_7, ZMV\} \end{aligned} \quad (6)$$

Type 4: If the mode of above neighboring MB is not 8×8 or 8×16, and the mode of below neighboring MB is 8×8 or 8×16, and the modes of both left and right neighboring MBs are 8×8 or 16×8, then Type 4 is selected and the lost MB is concealed as one 16×8 block and two 8×8 blocks. The MV candidates used for this type to conceal the lost MB are

$$\begin{aligned} B_0: \{MV_0, MV_1, MV_2, MV_6, ZMV\} \\ B_1: \{MV_3, MV_4, ZMV\} \quad B_2: \{MV_5, MV_7, ZMV\} \end{aligned} \quad (7)$$

Type 5: If the mode of above neighboring MB is 8×8 or 8×16, and the mode of below neighboring MB is not 8×8 or 8×16, and the modes of both left and right neighboring MBs are 8×8 or 16×8, then Type 5 is selected and the lost MB is concealed as one 16×8 block and two 8×8 blocks. The MV candidates used for this type to conceal the lost MB are

$$\begin{aligned} B_0: \{MV_0, MV_2, ZMV\} \quad B_1: \{MV_1, MV_6, ZMV\} \\ B_2: \{MV_3, MV_4, MV_5, MV_7, ZMV\} \end{aligned} \quad (8)$$

Type 6: If the modes of both above and below neighboring MBs are 8×8 or 8×16, and the mode of left neighboring MB is 8×8 or 16×8, and the mode of right neighboring MB is not 8×8 or 16×8, then Type 6 is selected and the lost MB is concealed as one 8×16 block and two 8×8 blocks. The MV candidates used for this type to conceal the lost MB are

$$\begin{aligned} B_0: \{MV_0, MV_2, ZMV\} \quad B_1: \{MV_3, MV_4, ZMV\} \\ B_2: \{MV_1, MV_5, MV_6, MV_7, ZMV\} \end{aligned} \quad (9)$$

Type 7: If the modes of both above and below neighboring MBs are 8×8 or 8×16, and the mode of left neighboring MB is not 8×8 or 16×8, and the mode of right neighboring MB is 8×8 or 16×8, then Type 7 is selected and the lost MB is concealed as one 8×16 block and two 8×8 blocks. The MV candidates used for this type to conceal the lost MB are

$$\begin{aligned} B_0: \{MV_0, MV_2, MV_3, MV_4, ZMV\} \\ B_1: \{MV_1, MV_6, ZMV\} \quad B_2: \{MV_5, MV_7, ZMV\} \end{aligned} \quad (10)$$

After selecting the most suitable partition type for the lost MB using the neighboring MB modes information, each partition of the lost MB is concealed using different candidate set of MVs. This initial concealed MB is further processed with OBMC to avoid the spatial discontinuities.

4.3 Overlapped Block Motion Compensation

Annex F of the H.263 standard [21] describes the overlapped block motion compensation (OBMC) technique, which can be used to retain the spatial smoothness in the concealment of the lost MB. The initial concealed MB is split into four 8×8 blocks, and each of these blocks is processed individually.

For each 8×8 block, the best estimated motion vector MV_e has been determined in the first step of adaptive partitioning as explained above. If the partition type is 16×16, or 16×8, or 8×16, then each constituent 8×8 blocks of the partition will have the same MV as that of the partition. For example, if the 16×8 partition has a particular motion vector, then each of its two 8×8 blocks will also have the same motion vector. Besides the best estimated motion vector, two additional motion vectors are required: MV_{ab} , the motion vector of the block at the adjacent above or below side of the current block, and MV_{lr} , the motion vector of the block at the adjacent left or right side of the current block. Then, three predicted 8×8 blocks P_e , P_{ab} , and P_{lr} are formed using these three motion vectors. The block P_e is the concerned 8×8 block of the initial concealed MB. The blocks P_{ab} and P_{lr} correspond to the 8×8 top-left pixel area of the MBs in the reference frame pointed by MV_{ab} and MV_{lr} , respectively. Each pixel of the final concealed 8×8 block is modified by a weighted sum of three prediction values as

$$P(i, j) = [P_e(i, j)H_e(i, j) + P_{ab}(i, j)H_{ab}(i, j) + P_{lr}(i, j)H_{lr}(i, j) + 4] \gg 3$$

$$\forall (i, j) \in 8 \times 8 \text{ block of initial concealed MB} \quad (11)$$

where H_e , H_{ab} , and H_{lr} are the weighting matrices shown in Fig. 7, P is the final concealed block for the lost MB, and “ \gg ” is the right-shift bit operator.

4 5 5 5 5 5 5 4	2 2 2 2 2 2 2 2	2 1 1 1 1 1 1 2
5 5 5 5 5 5 5 5	1 1 2 2 2 2 1 1	2 2 1 1 1 1 2 2
5 5 6 6 6 6 5 5	1 1 1 1 1 1 1 1	2 2 1 1 1 1 2 2
5 5 6 6 6 6 5 5	1 1 1 1 1 1 1 1	2 2 1 1 1 1 2 2
5 5 6 6 6 6 5 5	1 1 1 1 1 1 1 1	2 2 1 1 1 1 2 2
5 5 6 6 6 6 5 5	1 1 2 2 2 2 1 1	2 2 1 1 1 1 2 2
5 5 5 5 5 5 5 5	2 2 2 2 2 2 2 2	2 1 1 1 1 1 1 2
4 5 5 5 5 5 5 4		

(a) H_e (b) H_{ab} (c) H_{lr}

Fig. 7. OBMC weighting matrices.

In the case where a neighboring MB is not Inter coded, then the following modifications apply. If any neighboring MB of the lost MB was not coded or skipped (due to zero motion) during encoding, then its MV is set to zero. If any neighboring MB was Intra coded, then its MV is replaced by the best estimated motion vector MV_e to determine the predicted block from the reference frame. Also, if the current block is at the border of the frame so that the neighboring MB is not available, then MV_e is used to predict the neighboring block for the purpose of weighted concealment.

5. ADAPTIVE HYBRID CONCEALMENT SCHEME

A hybrid spatio-temporal content-adaptive refined error concealment (CAREC) scheme using the proposed algorithms is discussed in this section.

For the lost MBs in the Intra frames, the PMEC algorithm is

used to conceal the errors. We switch between the PMEC and the CAMP algorithms for the lost MBs in the Inter frames. For each lost MB of the Inter frame, we examine its four neighboring MBs to determine their encoding modes. If at least two neighboring MBs are not correctly received or decoded, then the concealed neighboring MBs are used for the concealment of the lost MB. The neighboring MBs could be coded as Inter with the relevant MVs, or coded as Intra with the prediction modes, or skipped due to zero motion. If any neighboring MB is skipped, then it is replaced with the MB located in the corresponding spatial position in the previous frame.

If the number of Inter coded MBs is greater than or equal to the number of MBs coded with the other two types, then the CAMP algorithm is used to utilize the temporal information in the concealment process. Otherwise, the PMEC algorithm is used for concealing the lost MB of the Inter frame.

6. ANALYSIS AND EXPERIMENTAL RESULTS

The PMEC and CAMP algorithms were introduced within JM ver. 18 of the H.264/AVC baseline profile reference software [4]. To evaluate the performance of the proposed algorithms, we used six standard video sequences, namely CIF (352×288) resolution sequences *Foreman*, *Stefan*, *Hall*, and QCIF (176×144) resolution sequences *Carphone*, *Table*, and *Salesman*. We encoded the first 100 frames of each of these sequences at 128 kbps and 20 fps using dispersed FMO [1]. The GOP length was set to 25, i.e. an Intra frame was coded after every 25 frames of the video sequence. To simulate an error-prone environment, we dropped MBs at the common loss rates of 5%, 10%, and 20% [22]. Since the error concealment is a post-processing module in the decoder, we use the error-free reconstructed video sequence as the reference for quality calculations instead of the original uncompressed video to accurately determine the concealment performance. The CAREC scheme using the proposed algorithms was evaluated against the algorithms implemented in JM [4] and Xu’s method of Refined Spatial and Temporal Concealment (RSTC) [10]. The hybrid concealment scheme of CAREC is similar to that of RSTC.

Two quality metrics are used to evaluate the performance of the proposed algorithms: Peak Signal-to-Noise Ratio (PNSR) and the video quality index (Q) proposed in [23]. Q is based on the concept of structural similarity index (SSIM) [24], which takes into account the luminance, contrast and structural degradations in the distorted image. To determine Q , weighted SSIM values are obtained for the luminance component of each frame in the test video sequence. These SSIM values are then averaged over the sequence by a frame weight adjustment, which considers the texture characteristics of each frame and the effects of motion between adjacent frames. To do this, a simple block-based motion estimation method is used to determine the motion vectors. The maximum value of Q is 1, and a value closer to 1 indicates that the concealed frame quality is closer to the quality of the error-free frame. The video quality index Q and SSIM have good correlation with the perceived video quality and the mean opinion score obtained from the subjective quality assessments [23], [24].

Fig. 8 shows the visual analysis of a *Table-Tennis* Inter frame concealed with the RSTC and CAREC schemes.



Fig. 8. Concealment of Table-Tennis frame at 10% loss. (a) Concealment with RSTC (PSNR=31.8165 dB, $Q=0.8674$). (b) Concealment with CAREC (PSNR=33.0837, $Q=0.9578$).

The RSTC scheme determines the dominant edge orientation using a gradient filter whereas the CAREC scheme uses the existing Intra prediction modes of the neighboring MBs. This reduces the computational complexity of CAREC significantly, since the concealment is done using the existing information in the coded bit-stream. Table I shows the computational complexity required to conceal a 16×16 lost MB, considering all four neighboring MBs are available. The JM reference software [4] requires six additions, four multiplications and one division for concealment of each pixel of the lost MB. Thus, to conceal all 256 pixels of the lost MB, there are 1536 additions, 1024 multiplications and 256 divisions.

TABLE I. Computational Complexity of CAREC

Operations	JM Test Model [4]	Xu's Method [10]	Proposed Algorithm
Additions	1536	1280+512*	544+512*
Multiplications	1024	640	512
Divisions	256	320	272
Comparisons	0	7	104
Shifts	0	0+512*	0+512*

*Extra computations for half-pixel boundary values.

For Xu's method, edge detection is done using gradient computations which require five additions for horizontal and five additions for vertical Prewitt mask, two multiplications and one addition for squared gradient magnitude and one division for finding the slope of the edge. Since the gradient is computed on the sixteen boundary pixels of all the four neighbors, there are 704 additions, 128 multiplications and 64 divisions for gradient computations. For each of the 64 boundary pixels, one addition is required to increment the counter with the gradient magnitude for the corresponding edge direction. Seven comparisons of the counter are required to determine the dominant edge direction out of eight possible directions for the lost MB. Further, the directional interpolation requires two additions, two multiplications and one division for concealment of each pixel of the lost MB. Thus, to conceal all 256 pixels of the lost MB, there are a total of 1280 additions, 640 multiplications, 320 divisions and 7 comparisons. Additionally, if the dominant edge direction is not an integer so that the detected edge falls on half-pixel boundary locations, then an

extra two additions and two shifts are required to compute the two half-pixel boundary values. In this case, there are an extra 512 additions and 512 shifts to conceal the whole MB.

The proposed algorithm of PMEC first sorts the four pixel intensities to find the maximum and minimum values, thus requiring six comparisons for each *pmode*. To compute the edge magnitude for each *pmode*, one addition and one division are required as shown in eq. (1). Thus, for all sixteen *pmodes*, there are 96 comparisons, 16 additions and 16 divisions. One addition is required for incrementing the counter corresponding to each *pmode*, so 16 more additions are required. Eight comparisons decide the *dpm* out of nine possible *pmodes* for the lost MB. Directional interpolation requires two additions, two multiplications and one division for concealment of each pixel of the lost MB. Thus, to conceal all 256 pixels of the lost MB, there are a total of 544 additions, 512 multiplications, 272 divisions and 104 comparisons. Additionally, if the *dpm* is either 5, 6, 7 or 8, then an extra 512 additions and 512 shifts are required to conceal the whole MB. Overall, the proposed algorithm of PMEC/CAREC is faster compared to these two existing techniques due to its lower computational complexity.

For temporal concealment, the candidate set of MVs in RSTC is increased by including the mean, median and diagonal MVs of the neighboring MBs, and the lost MB is always split into four 8×8 blocks. Due to this, certain blocking artifacts are observed as shown in Fig. 8 (a) as the splitting of lost MB into four blocks is not always helpful and the spatial continuity is not retained. In CAREC, the lost MB is partitioned adaptively into different block sizes and each partition is concealed with the different sets of MVs giving a smoother concealment. The proposed scheme also uses OBMA for distortion computation instead of BMA as used in JM and weighted BMA in RSTC, and thus accurate MVs are used to determine the best matching candidate MB from the reference frame.

Table II gives a comparative analysis of PSNR with the three concealment algorithms at different MB loss rates for all the test video sequences examined in this work. Average PSNR was computed for the error-free sequence and for the concealed sequence with the JM implementation ($PSNR_{JM}$). The PSNR improvements with the RSTC and CAREC schemes with respect to JM were obtained as

$$\Delta PSNR_{RSTC} = PSNR_{RSTC} - PSNR_{JM} \quad (12)$$

$$\Delta PSNR_{CAREC} = PSNR_{CAREC} - PSNR_{JM} \quad (13)$$

These differences were averaged for the hundred frames of the test video sequence and their standard deviations (σ) were computed. CAREC scheme improves the PSNR performance at all MB loss rates for all the test video sequences under consideration compared to the JM implementation and the RSTC scheme. The maximum PSNR improvement with the CAREC scheme is 1.4 dB relative to RSTC and 2.9 dB relative to JM. A similar analysis with the video quality index is presented in Table III with

$$\Delta Q_{RSTC} = Q_{RSTC} - Q_{JM} \quad (14)$$

$$\Delta Q_{CAREC} = Q_{CAREC} - Q_{JM} \quad (15)$$

The CAREC scheme achieves higher quality values, with a

maximum improvement of 0.09 relative to RSTC and 0.19 relative to JM. Since the proposed algorithms in the CAREC scheme maintains the structural similarity of the concealed MB more closely with that of the neighboring MBs, the quality improvement with CAREC is higher compared to the RSTC scheme.

7. CONCLUSION

The proposed algorithms adapt the concealment mechanism of the lost MBs in the Intra and Inter frames depending on the neighboring available MB information. The computational complexity of the CAREC scheme is lower than the JM implementation and the RSTC scheme since the existing information of Intra prediction modes is used in the spatial error concealment. Due to the adaptive partitioning of the lost macroblocks in Inter frames, different candidate motion vectors are used for each partition making the temporal concealment smoother. The hybrid switching scheme improves the concealment of the damaged Inter frames by using the spatial concealment algorithm if no neighboring temporal information is available.

8. REFERENCES

- [1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 13, no. 7, pp. 560-576, Jul. 2003.
- [2] L. Superiori, O. Nemethova, W. Karner, and M. Rupp, "Cross-layer detection of visual impairments in H.264/AVC video sequences streamed over UMTS networks," *IEEE Intl. Workshop on Cross Layer Design*, pp. 96-99, Sep. 2007.
- [3] S. S. Hemami, and T. Meng, "Transform coded image reconstruction exploiting interblock correlation," *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 1023-1027, Jul. 1995.
- [4] H.264/AVC Software Coordination, "Joint Model software," *Joint Video Team*, ver. 18.0 [Online] Available: <http://iphome.hhi.de/suehring/tml/>.
- [5] J. W. Suh, and Y. S. Ho, "Error concealment based on directional interpolation," *IEEE Trans. Consum. Electron.*, vol. 43, no. 3, pp. 295-302, Aug. 1997.
- [6] D. L. Robie, and R. M. Mersereau, "The use of Hough transforms in spatial error concealment," *IEEE Intl. Conf. Acoustics, Speech and Signal Process.*, vol. 4, pp. 2131-2134, Jun. 2000.
- [7] Z. Alkachouh, and M. G. Bellanger, "Fast DCT-based spatial domain interpolation of blocks in images," *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 729-732, Apr. 2000.
- [8] G. S. Yu, M. M. K. Liu, and M. W. Marcellin, "POCS-based error concealment for packet video using multiframe overlap information," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 8, no. 4, pp. 422-434, Aug. 1998.
- [9] P. Salama, N. Shroff, and E. J. Delp, "Error concealment in MPEG video streams over ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1129-1144, Jun. 2000.
- [10] Y. Xu, and Y. Zhou, "H.264 video communication based refined error concealment schemes," *IEEE Trans. Consum. Electron.*, vol. 50, no. 4, pp. 1135-1141, Nov. 2004.
- [11] Z. Wang, Y. Yu, and D. Zhang, "Best neighborhood matching: An information loss restoration technique for block-based image coding systems," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 1056-1061, Jul. 1998.
- [12] D. Agrafiotis, D. R. Bull, and C. N. Canagarajah, "Enhanced error concealment with mode selection," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 16, no. 8, pp. 960-973, Aug. 2006.
- [13] W. M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," *IEEE Intl. Conf. Acoustics, Speech and Signal Process.*, vol. 5, pp. 417-420, Apr. 1993.
- [14] J. Zheng, and L. P. Chau, "A temporal error concealment algorithm for H.264 using Lagrange interpolation," *Proc. IEEE Intl. Symp. Circuits and Systems*, vol. 2, pp. 133-136, May 2004.
- [15] J. Zhang, J. F. Arnold, and M. R. Frater, "A cell-loss concealment technique for MPEG-2 coded video," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 10, no. 4, pp. 659-665, Jun. 2000.
- [16] T. Thaipanich, P. H. Wu, and J. C. C. Kuo, "Video error concealment with outer and inner boundary matching algorithms," *Applications of Digital Image Processing XXX, Proceedings of the SPIE*, vol. 6696, DOI: 10.1117/12.735998, Sep. 2007.
- [17] M. J. Chen, L. G. Chen, and R. M. Weng, "Error concealment of loss motion vectors with overlapped motion compensation," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 7, no. 3, pp. 560-563, Jun. 1997.
- [18] M. Orchard, and G. J. Sullivan, "Overlapped block motion compensation: An estimation theoretic approach," *IEEE Trans. Image Process.*, vol. 3, no. 9, pp. 693-699, Sep. 1994.
- [19] D. Kim, S. Yang, and J. Jeong, "A new temporal error concealment method for H.264 using adaptive block sizes," *IEEE Intl. Conf. Image Process.*, vol. 3, pp. 928-931, Sep. 2005.
- [20] O. Nemethova, A. Al-Moghrabi, and M. Rupp, "Flexible error concealment for H.264 based on directional interpolation," *IEEE Intl. Conf. Wireless Networks, Communications and Mobile Computing*, vol. 2, pp. 1255-1260, Jun. 2005.
- [21] ITU-T Video Coding Experts Group, "Video coding for low bit rate communication," ITU-T recommendation H.263, Jan. 2005.
- [22] S. Wenger, "Common conditions for wire-line, low delay IP/UDP/RTP packet loss resilient testing," ITU-T SG16 Doc. VCEG-N79r1, 2001.
- [23] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, special issue on "Objective Video Quality Metrics", vol. 19, no. 2, pp. 121-132, Feb. 2004.
- [24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error measurement to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 1, pp. 1-14, Jan. 2004.

TABLE II. Simulation Results: PSNR

Test Sequence	Error-free PSNR (dB)	Loss Rate	PSNR _{JM} (dB)	Δ PSNR _{RSTC} (dB)		Δ PSNR _{CAREC} (dB)	
				Mean	σ	Mean	σ
Foreman	36.3874	5%	30.7202	1.1015	0.0686	1.8684	0.0439
		10%	28.7335	1.4231	0.0704	2.3502	0.1018
		20%	26.9476	1.3918	0.0571	2.7648	0.0714
Stefan	35.1793	5%	27.8336	1.3445	0.0531	1.8041	0.0647
		10%	24.3828	1.7142	0.0884	2.3829	0.0398
		20%	21.9611	1.5623	0.0684	2.9104	0.0648
Hall	37.1778	5%	31.0822	1.7201	0.0932	1.8917	0.0568
		10%	29.8872	1.6166	0.0881	2.2485	0.0562
		20%	27.0216	1.8637	0.0755	2.7181	0.0891
Carphone	36.7651	5%	31.6324	1.2868	0.0233	1.6814	0.0545
		10%	30.7521	1.7269	0.0813	2.2819	0.0891
		20%	28.5611	1.6369	0.0341	2.7245	0.0929
Table	35.0361	5%	28.6509	0.9998	0.0722	1.5371	0.0858
		10%	26.5743	1.5792	0.0596	2.4994	0.0983
		20%	24.4453	1.3668	0.0445	2.2675	0.0451
Salesman	34.9893	5%	31.7119	1.0993	0.0423	1.9151	0.0378
		10%	29.0715	1.2129	0.0122	2.1036	0.0596
		20%	26.6122	1.3158	0.0517	2.5146	0.0997

TABLE III. Simulation Results: Video Quality Index Q

Test Sequence	Error-free Q	Loss Rate	Q_{JM}	ΔQ_{RSTC}		ΔQ_{CAREC}	
				Mean	σ	Mean	σ
Foreman	0.9343	5%	0.8976	0.0155	0.0035	0.0246	0.0028
		10%	0.8766	0.0225	0.0023	0.0425	0.0032
		20%	0.8314	0.0577	0.0045	0.0949	0.0032
Stefan	0.9754	5%	0.8906	0.0224	0.0034	0.0742	0.0033
		10%	0.8285	0.0568	0.0058	0.1094	0.0064
		20%	0.7429	0.1006	0.0063	0.1945	0.0092
Hall	0.9496	5%	0.9124	0.0153	0.0029	0.0236	0.0025
		10%	0.8752	0.0348	0.0037	0.0618	0.0042
		20%	0.8048	0.0944	0.0037	0.1268	0.0046
Carphone	0.9655	5%	0.8949	0.0278	0.0027	0.0632	0.0036
		10%	0.8551	0.0389	0.0028	0.0966	0.0029
		20%	0.7623	0.1211	0.0062	0.1954	0.0025
Table	0.8882	5%	0.8065	0.0288	0.0031	0.0692	0.0048
		10%	0.7626	0.0643	0.0048	0.1129	0.0055
		20%	0.7196	0.0671	0.0044	0.1368	0.0074
Salesman	0.9573	5%	0.8798	0.0224	0.0065	0.0618	0.0032
		10%	0.8315	0.0537	0.0066	0.1104	0.0062
		20%	0.7806	0.1117	0.0057	0.1509	0.0086