Joint Temporal and Spatial Error Concealment for Multiple Description Video Coding

Wen-Jiin Tsai and Jian-Yu Chen

***Abstract*—Transmission of compressed video signals over error-prone networks exposes the information to losses and errors. To reduce the effects of these losses and errors, this paper presents a joint spatial-temporal estimation method which takes advantages of data correlation in these two domains for better recovery of the lost information. The method is designed for the hybrid multiple description coding which splits video signals along spatial and temporal dimensions. In particular, the proposed method includes fixed and content-adaptive approaches for estimation method selection. The fixed approach selects the estimation method based on description loss cases, while the adaptive approach selects the method according to pixel gradients. The experimental results demonstrate that improved error resilience can be accomplished by the proposed estimation method.**

***Index Terms*—Lost description estimation, multiple description coding, spatial segmentation, temporal segmentation.**

I. Introduction

THE DEMAND for transmitting video signals over wireless channels or over Internet protocol-based networks increases as bandwidth and storage of computer networks grow. Unfortunately, these environments are error-prone. During data transmission, packets may be dropped or damaged, due to channel errors, congestion, and buffer limitation. Moreover, the data may arrive too late to be used in real-time applications. In the case of transmission of compressed video sequences, this loss may be devastating and result in a completely damaged stream at the decoder side. For real-time applications, since retransmission is often not acceptable, error resilience (ER) and error concealment (EC) techniques are required for displaying a pleasant video signal despite the errors and for reducing distortion introduced by error propagation.

Several ER methods have been developed, such as forward error correction [1], intra/inter coding mode selection [2], layered coding [3], and multiple description coding (MDC) [4]. This paper is concerned with MDC. MDC is a technique that encodes a single video stream into two or more equally important sub-streams, called *descriptions*, each of which can be decoded independently. Different from the traditional single description coding (SDC) where the entire video stream (single description) is sent in one channel; in MDC, these multiple descriptions are sent to the destination through different channels, resulting in much less probability of losing the entire video stream (all the descriptions), where the packet losses of all the channels are assumed to be independently and identically distributed. Due to effectiveness in providing ER, a variety of MDC approaches had been proposed. These approaches can be intuitively classified through the stage where they split the signal, such as frequency [5], [6], spatial [7], [8], and temporal [9], [10] domains. In our previous works [11], a hybrid MDC method has been proposed, which splits the video signal along two dimensions, spatial and frequency domains. The hybrid method applies MDC first in spatial domain to split motion-compensated (MC) residual data, and then in frequency domain to split quantized coefficients.

In case of packet loss, EC techniques can be used to recover the lost information. There are many existing EC algorithms, such as spatial interpolation [12], frequency domain interpolation [13], [14], and temporal compensation based on inter-frame correlation [15]. There are several methods for spatial recovery of a lost block, which differ in the amount of neighboring pixels used, in their location and distance from the lost pixel, and in their relative weights in the concealment process. Frequency domain interpolation is based on spatial smoothness. It assumes high correlation between spatially adjacent blocks and uses discrete cosine transform (DCT) coefficients of neighboring blocks to reconstruct DCT coefficients of the missing block. As for temporal concealment methods, although replacing a lost block with the co-located block in the previous frame seems to be the easiest and fastest approach, it suffers from large distortion in case of fast motion in the block area. Thus, some methods based on motion compensation have been proposed, which replace the lost block with the one from previous frame that is shifted to compensate the estimated motion and minimizes boundary error to the correctly received adjacent blocks. Most of these methods estimate the lost motion vectors (MVs) from correctly received neighbors.

The aforementioned EC algorithms are originally designed for SDC and most of them assume that only a few slices or macroblocks (MBs) in a frame are lost. Although some temporal concealment methods, such as motion vector extrapolation (MVE) [16], [17], have been proposed to combat

Manuscript received October 31, 2009; revised April 20, 2010; accepted June 8, 2010. Date of publication October 18, 2010; date of current version January 22, 2011. This work was supported in part by the National Science Council of Taiwan, under Contract NSC98-2221-E-009-085. This paper was recommended by Associate Editor J. Ridge.

W.-J. Tsai is with the Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: wjtsai@cs.nctu.edu.tw).

J.-Y. Chen is with the military service in Taiwan (e-mail: abu.cs96g@g2.nctu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2010.2087816

the loss of a whole frame, only received past frames can be accessed to interpolate and estimate the lost MVs. In MDC, however, it is possible to access both past and future frames to interpolate the lost frame if the video stream is split in temporal domain. Moreover, in SDC, since pixels belonging to the same MBs can only be transmitted in the same packets, they will be missing all together once the packet is lost. Spatial EC in SDC can only utilize neighboring received MBs to estimate the inner pixels of the lost MB. By taking advantages of MDC, however, consecutive pixels of a video frame can be transmitted in different packets if they are split into different descriptions, and spatial EC has more alternatives to interpolate the missing pixels using neighboring pixels.

In the scope of this contribution, we will focus on the EC approaches in MDC. Although there have been some researches for this topic, most of them reconstruct the lost blocks by extrapolating the signal from correctly received areas either in spatial or in temporal direction. For the former one, only information from the spatial neighborhood of the lost block is used for extrapolating the signal and therewith concealing the loss, and, for the latter one, only information from temporally adjacent frames is used to extrapolate the lost block. To cope with this shortcoming, we propose a joint spatial-temporal estimation method using fixed and content-adaptive techniques to reconstruct lost descriptions. In particular, our method is designed in the context of hybrid MDC methods. Although the hybrid MDC in [11] has shown promising error-resilient results by splitting the video stream along spatial and frequency domains, its EC method utilizes data correlation within individual frame only. Data correlation between frames is not explored. This paper is concerned with the hybrid MDC method which segments the video along spatial and temporal dimensions. The results of experiments confirms that better estimation of lost descriptions can be achieved by taking advantages of data correlation in these two dimensions.

This paper is organized as follows. Section II introduces the hybrid MDC model upon which our method is based, and Section III presents the proposed estimation methods for description reconstruction. Experimental results are shown and discussed in Section IV. Concluding remarks are given in Section V.

II. Hybrid Model

The proposed hybrid model (called Hybrid) is designed to explore both temporal correlation between successive frames and spatial correlation between adjacent MC residual pixels. In this section, the Hybrid encoder is presented first, and then we present the Hybrid decoder.

A. Hybrid Encoder

The Hybrid encoder architecture is illustrated in Fig. 1, where the encoder has a two-level splitting process: 1) temporal splitter, and 2) residual splitter; the former splits the video sequence in temporal domain before motion estimation, while the latter splits the MC residual data in spatial domain.

The first level splitting, temporal splitter, splits a sequence along temporal dimension into two subsequences: one for all

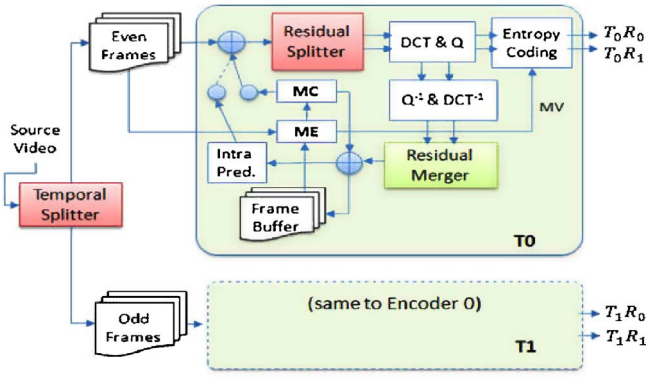

Fig. 1. Encoder architecture of Hybrid MDC.

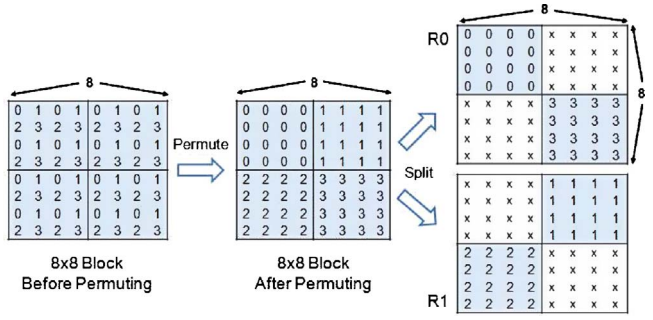

Fig. 2. Polyphase permuting and splitting of an 8×8 residual block.

the even frames and the other for all the odd frames. Even frames are predicted from even ones, and odd frames from odd ones, resulting in two motion-estimation prediction loops. We refer to one of the prediction loops as T_0 and the other as T_1. After motion estimation and compensation in each loop, the second level splitting, residual splitter, is performed on an 8×8-block basis using polyphase permuting and splitting in the residual domain. Each MC 8×8 residual block is first polyphase permuted inside the block and then split to two blocks, as shown in Fig. 2. The permuting mechanism is that the pixels in the 8×8 residual-block are first labeled with numbers ranging from 0 to 3, where for every 2×2 pixels, 0 is labeled on top-left pixel, 1 on top-right pixel, 2 on bottom-left pixel, and 3 on bottom-right pixel, and then label-0 pixels are re-arranged to the top-left 4×4 block, label-1 pixels to the top-right 4×4 block, and so on, as illustrated in the middle of Fig. 2. Note that there are four 8×8 residual blocks in each MB, all of them are permuted in the same way. The purpose of permuting pixels before splitting is to take into account the estimation method of lost description, which will be discussed in the next section.

After polyphase permuting, the splitting process is performed to split each 8×8 block into two 8×8 blocks, called residual 0 (R0) and residual 1 (R1), each carries two 4×4 blocks chosen in diagonal: top-left and bottom-right. 4×4 blocks belong to one 8×8 block, while top-right and bottom-left ones belong to the other 8×8 block. For each 8×8 block, the remaining two 4×4 blocks with all pixels labeled with "x" in Fig. 2 are given all-zero residual pixels. The encoder needs only little bits to encode these two blocks because their coefficients are all zero.

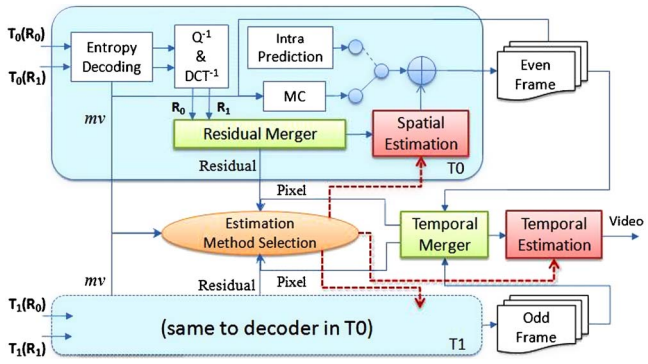

Fig. 3. Decoder architecture of Hybrid MDC.

Briefly, the encoding path of Hybrid MDC is split into two after temporal splitter, and then four after residual splitter, as shown in Fig. 1. That is, it splits every two frames into four descriptions using two prediction loops, where the resulting four descriptions are T_0R_0, T_0R_1, T_1R_0, and T_1R_1, respectively.

The second level splitting, residual splitter, uses a single prediction loop for two descriptions. To construct reference frames for prediction, residual merger is used. As shown in Fig. 1, after dequantization and inverse transformation, R0 and R1 are obtained and then residual merger is applied, which first discards the all-zero 4×4 blocks in R0 and R1, combines the resulting R0 and R1 into 8×8 blocks, and then performs polyphase inverse permuting to reconstruct the 8×8 blocks as reference. The four 8×8 blocks in a MB are all processed in this way. Actually, the residual merger is the reverse of residual splitter because it performs polyphase permuting and splitting in a reversed way.

B. Hybrid Decoder

Hybrid decoder architecture is depicted in Fig. 3, where the four input descriptions are T_0R_0, T_0R_1, T_1R_0, and T_1R_1. These descriptions are separately entropy decoded, dequantized, and inversely transformed, and then residual merger is applied to merge every two descriptions from the same prediction loops. The residual merger adopts residual merging and polyphase inverse permuting in the same way as illustrated in the encoder side. After motion compensation on each prediction loop, the temporal merger is applied to reconstruct the whole sequence. As shown in Fig. 3, lost descriptions (if any) can be spatially estimated after residual merger is performed, or temporally estimated after temporal merger is done. This is controlled by the estimation selection module. The details of the estimation methods are illustrated in the next section.

Note that the proposed Hybrid MDC is not fully compatible with H.264 standard because it requires a residual splitter and a residual merger inside the prediction loop as well as a temporal splitter and a temporal merger outside the prediction loop. For other functional blocks, they are compatible with H.264/AVC. To support bidirectional prediction (i.e., B-frame coding) in this Hybrid model, it is clear that no change is needed for the temporal splitter and merger because they are outside the prediction loop. There is also no need to change residual splitter and merger because they can perform splitting and merging on the MC residue of B frames just in the same way on that of P frames. To support B frames, only temporal estimation methods (which are beyond the scope of H.264 standard) need to be modified because more MVs can be utilized to have better estimation of the lost descriptions.

III. Estimation of Lost Description

If the decoder does not receive all the descriptions intact, then temporal or (and) spatial estimation methods for lost-description reconstruction are adopted to reconstruct the lost data. We first describe the temporal and spatial estimation methods in the context of the proposed Hybrid MDC, and then the criterion for estimation method selection is presented.

A. Temporal Estimation Method

Temporal estimation method can be applied to recover a whole frame or part of a frame. Here two bidirectional temporal estimation methods are proposed: one uses pixel-based motion vector interpolation (B-PMVI) and the other uses pixel-based motion vector extrapolation (B-PMVE).

1) *Whole Frame Estimation with B-PMVI:* When two descriptions from the same prediction loop are lost, it will result in the whole-frame loss. Since the proposed Hybrid method splits consecutive frames into different prediction loops, the lost frame can be estimated by its previous frame and next frame in the other prediction loop, as illustrated by the example in Fig. 4(a), where assumed frame n, from the prediction loop T_1, is lost. Since the MVs of all the MBs in frame n are lost, the lost motion information can be simply replaced by zero, i.e., each missing pixel is estimated by the co-located pixel value in the previous decoded frame. This works well for stationary areas, but fails for moving area. MVE [16] is another method combating the frame loss. In this method, the MVs of MBs are extrapolated from the last decoded frame to the missing frame. This method can overcome the disadvantage of incorrect MB displacement, but the block-based MV is easy to cause block-artifacts. To overcome this problem, Chen [17] proposed a method called PMVE, which extended the MVE to pixel level and improved the performance in large motion scenes. However, since PMVE is designed in the context of SDC, it only utilizes the pixels in the previous decoded frame for EC. In this paper, we take advantage of the proposed Hybrid MDC method and propose a bidirectional pixel-B-PMVI method, which replaces the pixels of the lost frame with the average of pixels at MC locations in two frames coming from the other prediction loop. Let $mv_{i,j}$ denote the MV pointing to frame j from frame i. In Fig. 4(b), by interpolating the $mv_{n+1,n-1}$, which is obtained from prediction loop T_0, an interpolated block on the missing frame n, and its two interpolated MVs, $mvi_{n,n-1}$ and $mvi_{n+1,n}$, can be obtained. (Note that the interpolation is done at pixel level, so the interpolated block is unnecessary to be aligned on MB positions.) By inversing $mvi_{n+1,n}$, we yield two estimated MVs for each pixel of the interpolated block: one is forward vector, $(f_x, f_y) = mvi_{n,n-1}$, and the other is backward vector, $(b_x, b_y) = -mvi_{n+1,n}$. The interpolation is performed for every

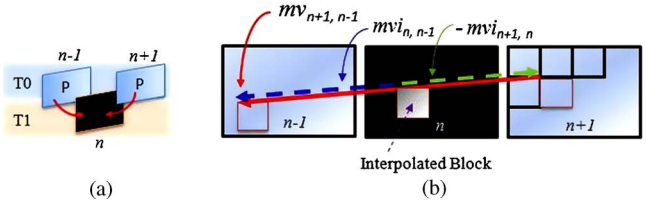

Fig. 4. Whole frame estimation with B-PMVI. (a) Bidirectional estimation. (b) MV interpolation.

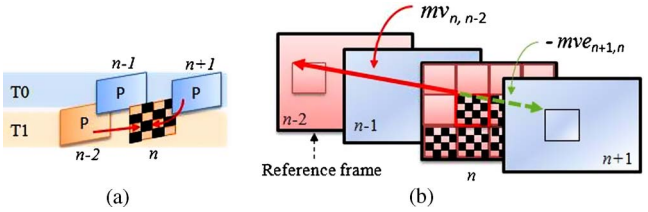

Fig. 5. Partial frame estimation with B-PMVE. (a) Bidirectional estimation. (b) MVE.

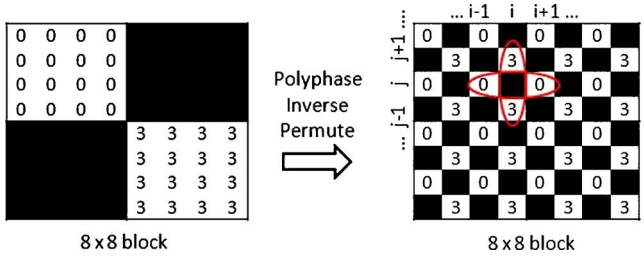

Fig. 6. Spatial estimation by bilinear interpolation.

MV in frame $n+1$, and the pixels in the lost frame n can be divided into two parts as follows.

1) For the pixel covered by at least one interpolated block, its two MVs are estimated by averaging the corresponding MVs of all overlapped blocks.
2) For the pixel not covered by any interpolated block, its two MVs are set to zero, i.e., $(f_x,\ f_y)=(b_x, b_y)=(0,\ 0)$.

As a consequence, for a pixel $(x,\ y)$ in the lost frame n, with its two MVs, $(f_x,\ f_y)$ and (b_x, b_y), its value $\mathrm{P}_n(\mathrm{x},\ \mathrm{y})$ can be estimated as follows:

$$\mathrm{P_n}(x,\ y) = w \times \mathrm{P_{n-1}}(x+f_x,\ y+f_y)+(1-w)\times \mathrm{P_{n+1}}(x+b_x,\ y+b_y) \tag{1}$$

where w is used to adjust the weights of forward and backward MC pixels. In this paper, we simply average the two candidates, i.e., $w = 0.5$. Note that since the proposed B-PMVI method relies on the MVs in the nearest available succeeding P frames in the other description, it is applicable to whole frame loss, regardless of whether the lost frame is a P frame or an I frame.

2) *Partial Frame Estimation with B-PMVE:* When only one description is missing, it will result in partial frame loss. Since the second level splitting of the Hybrid MDC splits a frame in the residual domain of the same prediction loop, the resulting two descriptions will have the same MVs. Thus, when one of them is lost, its missing MVs can be recovered from the other one and thus, motion compensation still can be done. As for the lost residual data, its reference frame and the next frame in the other prediction loop are used as depicted in Fig. 5(a), where assumed one description of frame n is lost. For a lost residual pixel on frame n, since its MV pointing to its reference frame is available (i.e., $mv_{n,n-2}$), the only problem is to find its motion information on frame $n+1$. We use MVE, as depicted in Fig. 5(b). By extrapolating $mv_{n,n-2}$, an extrapolated MV, $mve_{n+1,\mathrm{n}}$, can be obtained. That is, for a lost residual pixel $(x,\ y)$ on frame n, we have its forward MV as $(f_x,\ f_y) = mv_{n,n-2}$, and the backward vector as $(b_x, b_y) = -mve_{n+1,\mathrm{n}}$. With the two MVs, its value $\mathrm{P}_n(\mathrm{x},\ \mathrm{y})$ can be estimated as follows:

$$\mathrm{P_n}(x,\ y) = w \times \mathrm{P_{n-2}}(x+f_x,\ y+f_y)+(1-w)\times \mathrm{P_{n+1}}(x+b_x,\ y+b_y) \tag{2}$$

where w is again the weights of forward and backward MC pixels.

Note that the proposed B-PMVE method relies on the availability of MVs of the lost pixels from the other description. However, this may fail if the lost pixels belong to intra-coded blocks which have no MVs. In this case, spatial estimation method will be applied, which is described in the next section.

B. Spatial Estimation Method

Spatial estimation method explores the spatial correlation between MC residual pixels to estimate the lost description in residual domain. It requires at least one of the two descriptions from the same prediction loop that is correctly received, namely, it is only adopted for the case of partial frame loss. As described in the previous section, a lost description can obtain its missing MVs from its counterpart in the same prediction loop and thus, motion compensation still can be done. To recover the lost residual data, either temporal or spatial method can be used. The aforementioned B-PMVE is the temporal method for this; the spatial method is presented here and the mechanism for estimation method selection is described in the next section. Assume that ${T_0}^n R_0$ and ${T_0}^n R_1$ are two descriptions split from frame n belonging to prediction loop $\mathrm{T_0}$, and that ${T_0}^n R_1$ is lost during transmission. In this case, the motion compensation of ${T_0}^n R_1$ can be done by using the MVs from ${T_0}^n R_0$. As for the missing residual of ${T_0}^n R_1$, they are spatially estimated from the residual of ${T_0}^n R_0$. By polyphase inversely permuting the residual pixels of ${T_0}^n R_0$, they are distributed like a checkerboard within a MB as shown in Fig. 6, where for each lost residual pixel, four neighboring residual pixels are available. The spatial method uses bilinear interpolation to estimate the lost residual pixels, as shown in (3) where $\tilde{f}_{j,i}$ is the estimated value of the residual pixel in column i and row j. Since neighboring pixels have high spatial correlation, spatial estimation should be efficient as follows:

$$\tilde{f}_{j,i} = (f_{j+1,i}+f_{j-1,i}+f_{j,i+1}+f_{j,i-1})/4. \tag{3}$$

C. Estimation Method Selection

The proposed Hybrid MDC segments a video sequence into four descriptions. There are 16 states of the four descriptions

TABLE I
SUMMARY OF ESTIMATION METHODS IN THE CORRESPONDING CASES

Estimation Methods		Descriptor(s) in T0			
		R0+R1	R0	R1	Loss
Descriptior(s) in T1)	R0+R1	N/A	A	A	T
	R0	A	A	A	S → T
	R1	A	A	A	S → T
	Loss	T	S → T	S → T	N/A

as listed in Table I, where the columns describe the four possible cases for the two descriptions split from prediction loop T_0, while the rows describe those for T_1. The estimation method to be applied for each case are also shown in this table, where "T" denotes the temporal estimation and "S" the spatial estimation. The "S→T" denotes that spatial method will be performed first and then temporal method is applied, and the "A" indicates that either temporal or spatial method will be applied but the choice of the method adaptively depends on the content of the video. The "N/A" means that no estimation method will be applied. As can be seen in the table, "S→T" is applied only for the cases of three-description loss, while "T" is applied only when two descriptions split from the same prediction loop are lost and the other two are received. For these cases, the estimation methods are selected statically according to description status. For other cases (labeled with "A"), the choice of the estimation method is dynamically determined according to the video content.

1) *Static Choice of Estimation Methods:* Since Hybrid splits every two frames into four descriptions using two prediction loops (say T_0 and T_1), for consecutive two frames, n and $n+1$, we refer to the two descriptions split from frame n as ${T_0}^n R_0$ and ${T_0}^n R_1$, while the other two from frame $n+1$ as ${T_0}^{n+1} R_0$ and ${T_1}^{n+1} R_1$. To illustrate the cases that "S→T" will be applied, Fig. 7(a) depicts one of the four possible cases that three descriptions are lost. The descriptions marked with "(x)" mean that they are lost. In this case, since ${T_0}^n R_0$ from prediction loop T_0 is received, spatial estimation can be applied to reconstruct its counterpart, ${T_0}^n R_1$, as indicated by the dotted arrow labeled with "S." After merging ${T_0}^n R_0$ and ${T_0}^n R_1$, the reconstructed frame ${T_0}^n$, together with the frame ${T_0}^{n+2}$, are used by temporal method B-PMVI to recover the lost frame ${T_1}^{n+1}$, as indicated by dotted arrow with "T." Fig. 7(b) shows how the "S→T" is performed.

To illustrate the cases where "T" will be applied, Fig. 8 depicts two cases stating that two descriptions from the same prediction loop are lost. In each case, spatial estimation cannot be applied since the lost description has no counterpart in the same prediction loop available for spatial estimation. For these cases, temporal method of B-PMVI will be applied for whole frame estimation. For example, in Fig. 8(a), after merging and polyphase inverse permuting, the full frame $n+1$ from prediction loop T_1 can be obtained, which is then adopted by temporal estimation to recover the lost frame n belonging to prediction loop T_0.

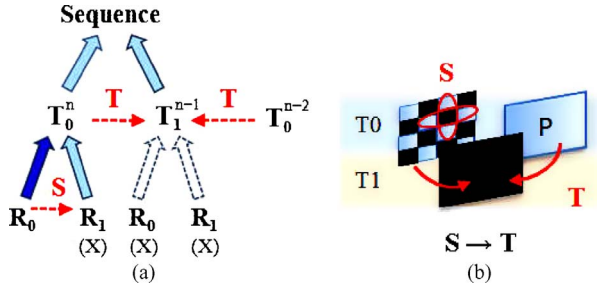

Fig. 7. "S→T" for three missing descriptions. (a) Three description loss. (b) Spatial and then temporal estimation.

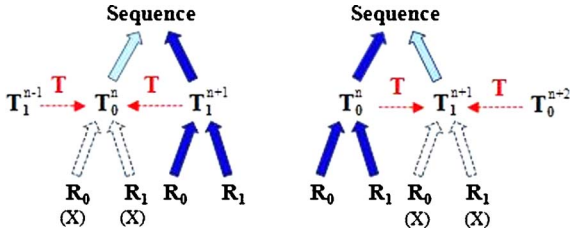

Fig. 8. Temporal estimation for two missing descriptions from the same prediction loop. (a) Two descriptions of T1 are lost. (b) Two descriptions of T0 are lost.

2) *Content-Adaptive Choice of Estimation Methods:* Adaptive estimation method is applied when two descriptions from different prediction loops are lost or when there is only one-description loss, as those labeled by "A" in Table I. In these cases, there are partially lost frames needed to be recovered. Fig. 9(a) depicts one out of four possible cases that one description ${T_0}^n R_1$ is lost, and Fig. 9(b) shows one of four cases that two descriptions ${T_0}^n R_1$ and ${T_0}^{n+1} R_0$ from different prediction loops are lost. In these cases, since each lost description can recover the lost MVs from its counterpart of the same prediction loop, motion compensation is able to be performed. Thus, only lost residual data needs to be estimated. For these cases, adaptive method which could be either spatial estimation or temporal estimation will be applied. As illustrated in Fig. 9(a), the missing residual of ${T_0}^n R_1$ can be predicted either from ${T_0}^n R_0$ by using spatial estimation [see left-side of Fig. 9(a)], or from two frames, ${T_0}^{n-2}$ and ${T_1}^{n+1}$, by using temporal estimation method B-PMVE [see right-side of Fig. 9(a)]. In Fig. 9(b), the lost residual of ${T_0}^n R_1$ and ${T_0}^n R_0$ can also be predicted by either spatial or temporal method in a similar way aforementioned.

Intuitively, it is more beneficial to adopt spatial estimation if it is a simple textured and high-motion video, and to apply temporal estimation if it is a slow-motion and complex textured video. To effectively select appropriate estimation methods for the above cases, a content-adaptive method is designed for the decoder, which measures the pixel gradient along the spatial and temporal dimensions to determine the characteristic of the video content and then makes the choice. The spatial gradient (GS) of a lost residual pixel is calculated as the average of the absolute differences between its two adjacent residual pixels in vertical and horizontal directions. Let ${r^n}_{(i,j)}$ denote a residual pixel at (i, j) of frame n. The GS

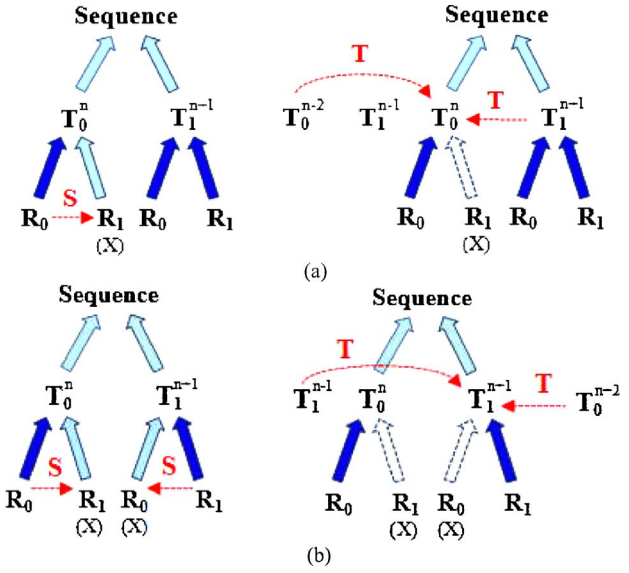

Fig. 9. Adaptive selection of estimation methods. (a) One description is lost. (b) Two descriptions from different prediction loops are lost.

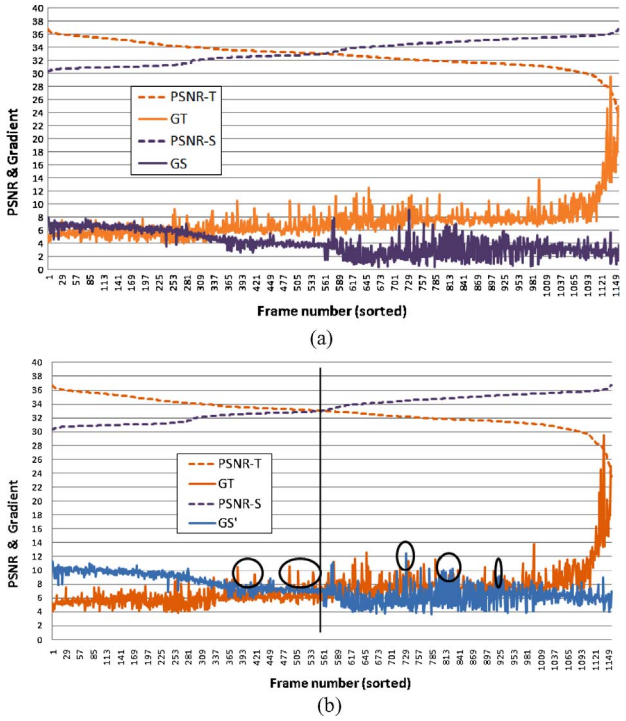

Fig. 10. Relation between PSNR and gradient value. (a) Statistical results (b) after lifting up the GS curve.

of this pixel is defined as follows:

$$GS^n_{(i,j)} = {}^1\!/_2\{|r^n_{(i,j+1)} - r^n_{(i,j+1)}| + |r^n_{(i+1,j)} - r^n_{(i-1,j)}|\}. \quad (4)$$

The temporal gradient (GT) of a lost residual pixel is defined as the absolute difference between the MC pixel in reference frame and the pixel at extrapolated location in the next frame, where pixel values, instead of residual-pixel values, are used in the calculation. For a lost residual pixel at (x, y) of frame n, assume its forward and backward MVs are (f_x, f_y) and (b_x, b_y), respectively, obtained by using B-PMVE. The GT of this residual pixel is then defined as follows:

$$GT^n_{(i,j)} = |Pixel^{n-2}_{(x+f_x,y+f_y)} - Pixel^{n+1}_{(x+b_x,y+b_y)}| \quad (5)$$

where $Pixel^k_{(i,j)}$ denotes the pixel value at (i, j) of frame k, $n-2$ denotes the reference frame, and $n+1$ denotes the next frame in the other prediction loop. To explore the relation between the estimation methods and the gradient values, experiments were conducted for 1160 frames from four different quarter common intermediate format sequences. All frames are encoded using the proposed Hybrid MDC and simulated with one-description loss. The lost description is reconstructed using temporal estimation on a per-frame basis without error propagation. The peak signal-to-noise ratio (PSNR) (denoted by PSNR-T) results of all frames are sorted in an ascending order and depicted in Fig. 10(a), where the average GT of each frame is also shown. Similar experiments were also conducted for spatial estimation method, and the average GS and PSNR (denoted by PSNR-S) are also presented in Fig. 10(a). As expected, the PSNR-T increases as GT decreases and the PSNR-S increases as GS decreases. The difference between PSNR-S and PSNR-T of the same frame can be up to more than 10 dB or down to equivalent, confirming that, to obtain the best PSNR for each frame, the choice of estimation methods is important. Besides, it is also observed that there is a single intersection for the two PSNR curves, where on each side of the intersection, one curve is always above the other one. Similar phenomenon also happens on the two gradient curves. By lifting up the GS curve about 3.3 units, the two intersection points will happen on the same frame as indicated by the vertical line shown in Fig. 10(b), where GS′ means GS + σ with $\sigma = 3.3$. Then, except some few frames circled in Fig. 10(b), almost all the frames with GS′ lower than GT will have higher PSNR-S than PSNR-T, indicating that spatial estimation is preferred for these frames. On the contrary, for those frames with GT lower than GS′, temporal estimation is preferred. Let e(A) denote the estimation method selected by adaptive method. Then, for a lost residual pixel at (x, y) in frame n, its e(A) is determined as follows:

$$e(A^n_{(x,y)}) = \begin{cases} \text{S}, & \text{if } GS^n_{(x,y)} + \sigma \le GT^n_{(x,y)} \\ \text{T}, & \text{if } GS^n_{(x,y)} + \sigma > GT^n_{(x,y)} \end{cases} \quad (6)$$

where σ is 3.3 for Fig. 10(b) in which QP = 28 is used. To explore the relation between σ and QP, we encode the same 1160 frames with 11 different QPs ranging from 18 to 38. For each QP, the σ is determined by sorting the PSNR results of the corresponding encoded 1160 frames just in the same way as Fig. 10. By depicting the σ value as a function of QPs in Fig. 11, we found that the relation between σ and QP can be modeled using a quadratic equation as follows:

$$\sigma = 0.017\text{QP}^2 - 0.525\text{QP} + 4.135. \quad (7)$$

With the σ determined by (7), the estimation method adopted can be selected by (6). Since the selection is on a pixel basis, different lost pixels on a frame might be reconstructed by using different estimation methods.

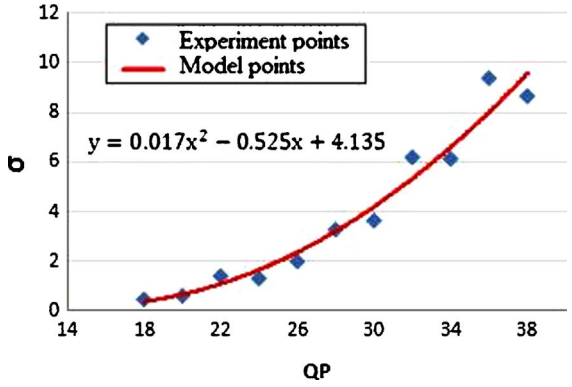

Fig. 11. Relation between σ and QP.

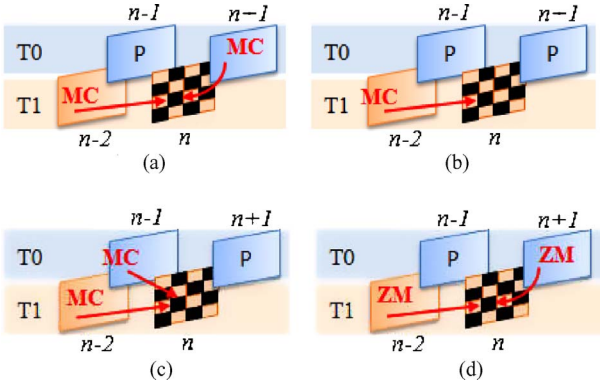

Fig. 12. Temporal estimation methods used for comparison. (a) B-PMVE. (b) 1FwdMC. (c) 2FwdMC. (d) Bi-ZM.

IV. Experimental Results

In this section, the performance results of the proposed Hybrid MDC are presented. We first examine the effects of temporal, spatial, and adaptive estimation methods used in the proposed Hybrid MDC, and then the performance of Hybrid MDC is examined in packet loss environments with various packet-loss rates. Rate-distortion (R-D) performance and frame-by-frame quality comparison are presented.

A. Performance of Estimation Methods

This section examines the performance of the estimation methods used by Hybrid MDC. Experiments were conducted for temporal, spatial, and adaptive methods, respectively.

1) *Temporal Estimation Method:* Here, we examine the performance of temporal estimation methods for partial frame loss. Since the second level splitter of Hybrid MDC produces two descriptions in the same prediction loop, when one of them is lost, its MVs can be found from the other to perform motion compensation. To estimate the lost MC residual, we compare the proposed B-PMVE with one-frame forward motion compensation (1FwdMC), two-frame forward MC interpolation (2FwdMC), and bidirectional zero-motion (Bi-ZM) interpolation. The four methods differ in the number of frames selected and the location of residual pixels used. As depicted in Fig. 12, 1FwdMC adopts the reference frame $(n-2)$ of the frame under estimation; 2FwdMC adopts two frames: one is the reference frame $(n-2)$ and the other is the nearest past frame $(n-1)$ of the other prediction loop; and both B-PMVE and Bi-ZM employ bidirectional frames: one is the reference frame $(n-2)$, and the other is the nearest next frame $(n+1)$ of the other prediction loop. In these methods, Bi-ZM uses zero-motion locations and the other three use MC locations for residual pixel selection. Since the decoder side has the MVs from frame n to $n-2$ only, the other MVs needed are either interpolated or extrapolated.

Experiments were conducted for the situations of partial frame loss, which include the cases of one-description loss and the cases that two descriptions from different prediction loops are lost, i.e., the eight cases labeled with "A" in Table I. (Note that in order to see the effects of temporal estimation, only temporal methods are adopted even though the proposed Hybrid MDC will apply adaptive method for these cases).

TABLE II
PSNR of Various Temporal Estimation Methods

	Foreman	*News*	*Mobile*	*Coastguard*
B-PMVE	**32.9**	**34.51**	**31.65**	**32.76**
1FwdMC	**32.9**	34.12	30.25	31.92
2FwdMC	**32.9**	34.51	30.98	**32.65**
Bi-ZM	32.34	**35**	**31.82**	32.29

Each lost case was tested independently on each frame without error propagation and four test sequences, *Foreman*, *News*, *Mobile*, and *Coastguard*, are used. The average PSNR of the eight loss cases from each frame is calculated and the results are shown in Table II, where the top two PSNR are highlighted for each sequence. It is observed that since 1FwdMC method uses only one frame for lost-residual estimation, it yields the worst performance among the four methods. 2FwdMC and Bi-ZM have complementary performance; one is better for *Foreman* and *Coastguard* sequences while the other is better for *News* and *Mobile* sequences. As for B-PMVE method, although it did not achieve the best estimation for all the sequences, it always performed as one of the top-two methods. Due to the superiority of B-PMVE in the estimation of partial frame loss, the proposed Hybrid MDC adopts it as the temporal estimation method.

2) *Spatial Estimation Method:* To examine the performance of spatial estimation methods, we compare the proposed spatial estimation method, Hybrid-S, with near neighbor replication (NNR), edge sensing (ES) [7], and ES-r, where NNR is a classical spatial estimation method which replicates the first correctly received pixel in the 8-pixel neighborhood of the current one, starting from the left and proceeding in a clockwise order; ES uses two gradients to detect horizontal and vertical edges around the processed pixel, and computes missing pixels while taking the edge orientation into account; and ES-r is a variation of ES, which, instead of applying estimation of lost data in the pixel domain as in the ES, applies the edge sensing algorithm on the residual data before motion compensation is performed.

Experiments were also conducted for the situations of partial frame loss, namely, the cases labeled with "A" in Table I. In order to see the effects of spatial estimation, we applied spatial

TABLE III
PSNR OF DIFFERENT SPATIAL ESTIMATION METHODS

	Foreman	*News*	*Mobile*	*Coastguard*
Hybrid-S	**34.33**	**35.57**	**31.19**	**32.74**
ES[7]	32.19	33.45	25.93	29.84
ES-r	**34.47**	**35.5**	**31.02**	**32.63**
NNR[7]	28.77	29.65	23.01	27.39

methods instead of adaptive method for these cases. Each lost case was tested independently on each frame without error propagation and the results of four test sequences are shown in Table III, where the average PSNR are presented and the top two PSNR are highlighted for each video sequence.

As the results show, both Hybrid-S and ES-r performed better than NNR and ES for all the sequences. It is due to that both Hybrid-S and ES-r apply estimation of lost data in residual domain before motion compensation; while NNR and ES in pixel domain. The results indicate that spatial estimation adopted in residual domain should have better performance, no matter what kind of sequences is used. Besides that, although Hybrid-S and ES-r have similar performance, the bi-linear interpolation method used in Hybrid-S is simpler than edge-sensing algorithm in ES-r which needs to calculate horizontal and vertical gradients to determine the direction of the interpolation. The results confirm that high performance can be achieved by Hybrid-S at low computational cost.

3) *Adaptive Estimation Method:* In order to see the effects of the proposed adaptive estimation method (called Hybrid-A), experiments were conducted for the eight cases of description-loss labeled with "A" in Table I. We compared the Hybrid-A with spatial estimation (Hybrid-S) and temporal estimation (Hybrid-T). The difference among the three methods is that Hybrid-A selects estimation method according to spatial and GTs as proposed, while Hybrid-S applies spatial estimation only, and Hybrid-T applies temporal estimation only. Each lost case was tested independently on each frame without error propagation. Four test sequences are used and the results are shown in Table IV, where the average PSNR of the eight description-loss cases are presented, and the top two PSNR are highlighted for each sequence.

It is observed that spatial method performed the best estimation for *Foreman* and *News* sequences, but worst for *Mobile* and *Coastguard* sequences, while temporal method performed the best estimation for *Mobile* sequence, but worst for *News* and *Foreman* sequences. Adaptive method, although it performed the best estimation only for the *Coastguard* sequence, it always performed as one of the top-two methods for each sequence. The reason why the performance of Hybrid-A is not always better than Hybrid-S and Hybrid-T can be explained from two aspects. First, although using pixel gradients as criterion is able to choose better estimation methods in most of the cases, it may fail sometimes. In Fig. 10(b), the frames marked with circles are the examples that wrong decisions will be made. Second, the σ in (6) is approximated by a quadratic curve; however, the model curve does not completely fit in with all the actual points as depicted in Fig. 11. Applying

TABLE IV
PSNR OF VARIOUS ADAPTIVE ESTIMATION METHODS

	Foreman	*News*	*Mobile*	*Coastguard*
Hybrid-A	**34.28**	**35.25**	**31.38**	**32.88**
Hybrid-S	**34.33**	**35.57**	31.19	32.74
Hybrid-T	32.9	34.51	**31.65**	**32.76**

TABLE V
COMPUTATIONAL OVERHEAD IN ADAPTIVE METHOD

	ADD	SUB	ABS	Shift	CMP
GS measure	1	2	2	1	0
GT measure	0	1	1	0	0
Decision making	1				1
Total	2	3	3	1	1

TABLE VI
PERCENTAGES OF SPATIAL AND TEMPORAL METHODS ADOPTED

		Foreman	*News*	*Mobile*	*Coastguard*
Hybrid-A	S	67.9%	49.2%	61.5%	57.8%
	T	32.1%	50.8%	38.5%	42.2%

model points in (6), Hybrid-A may make a wrong selection of estimation methods. As a consequence, Hybrid-A does not always perform better than Hybrid-S and Hybrid-T. Even though Hybrid-A does not always perform the best, it always performed as one of the top-two methods for each sequence, showing that with proper choice of estimation methods, the adaptive method can adapt to various types of video sequences.

4) *Complexity Analysis:* The proposed adaptive estimation method plays an important role in the Hybrid MDC because it is adopted for 8 out of 14 description-loss cases, as shown in Table I. However, the adaptive method suffers from computational overhead because it needs to measure and compare spatial and GTs (4)–(6) for each lost pixel before doing recovery. The overhead involves two addition, three subtraction, three absolution, one shift, and one comparison operations for each lost pixel as shown in Table V.

To see how the computational overhead affects the execution time of adaptive method, we have measured the execution time for the cases labeled with "A" in Table I, namely, the time of performing adaptive method. For comparison, we also applied spatial method and temporal method for these "A" cases and measured their respective execution times. In our experiments, packet loss rate (PLR) $P_{loss} = 5\%$ was used and error propagation was implemented. The time measured for each method was normalized by dividing with the execution time of spatial estimation method and was shown in Fig. 13. It is observed that, even with gradient measure overhead for each lost pixel, adaptive method still consumes less time than temporal method. This is due to that spatial method executes much faster than temporal method as shown in Fig. 13 and that adaptive method may choose to adopt spatial method after gradient measure. Table VI shows that about 49–68% of lost pixels in adaptive method were recovered by spatial estimation.

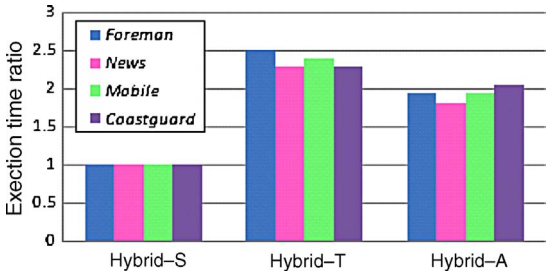

Fig. 13. Time comparison for different estimation methods.

B. R-D Performance

In this section, the proposed Hybrid MDC is examined in both error-free and packet-loss environments. We compare the Hybrid method with T4, PSS [7], H_SF [11], MSVC_RP [20], and H.264/AVC, where the T4 is a MDC method which splits video stream along temporal dimension only, the PSS is a MDC method which splits the video along spatial dimension only, the H_SF is a hybrid MDC which splits the video along spatial and frequency domains, the MSVC_RP is a MDC method which, instead of using sub-sampling as in the T4, PSS, and H_SF methods, uses redundant pictures to increase the resilience to loss, and the H.264/AVC is a SDC coder. The Hybrid, T4, PSS, H_SF, and MSVC_RP coders encode every video sequence into four descriptions, while H.264/AVC encodes every sequence as a single description. These methods are implemented based on H.264/AVC reference software, JM 13.2 [19].

Three common intermediate format (CIF) test sequences *Foreman*, *News*, and *Coastguard* are used for performance evaluation. Each sequence consists of 300 frames, the group of picture (GoP) size is 30 frames, the structure of GoP is IPPPP..., and the frame rate is 30 Hz. Fig. 14 shows the R-D performance for all the methods in error-free environment. It can be seen that H.264/AVC has the best R-D performance than all the MDC methods because it has the best coding efficiency. Among MDC methods, T4 performed the best, PSS performed the worst, and the two hybrid methods performed in between T4 and PSS, showing that temporal sub-sampling has better coding efficiency than spatial sub-sampling. Although Hybrid is less coding efficient than T4, the performance gaps between them are small. MSVC_RP is not a MDC based on sub-sampling, but its coding efficiency is close to T4.

The experiments were also conducted in a packet-loss scenario with the loss rates ranging from 1% to 10%. Bernoulli channel model was adopted which assumes that each packet is lost randomly and independently. To have a fair comparison, for each method, every packet consists of one-fourth information of one original frame. In other words, T4 which encodes every four frames into four descriptions uses four packets for each frame of each description, Hybrid which encodes every two frames into four descriptions uses two packets for each frame of each description, PSS and H_SF which sub-sample every single frame into four descriptions use one packet for each frame of each description, MSVC_RP which inserts redundant pictures for every frame to produce four descriptions uses four packets for each frame of each

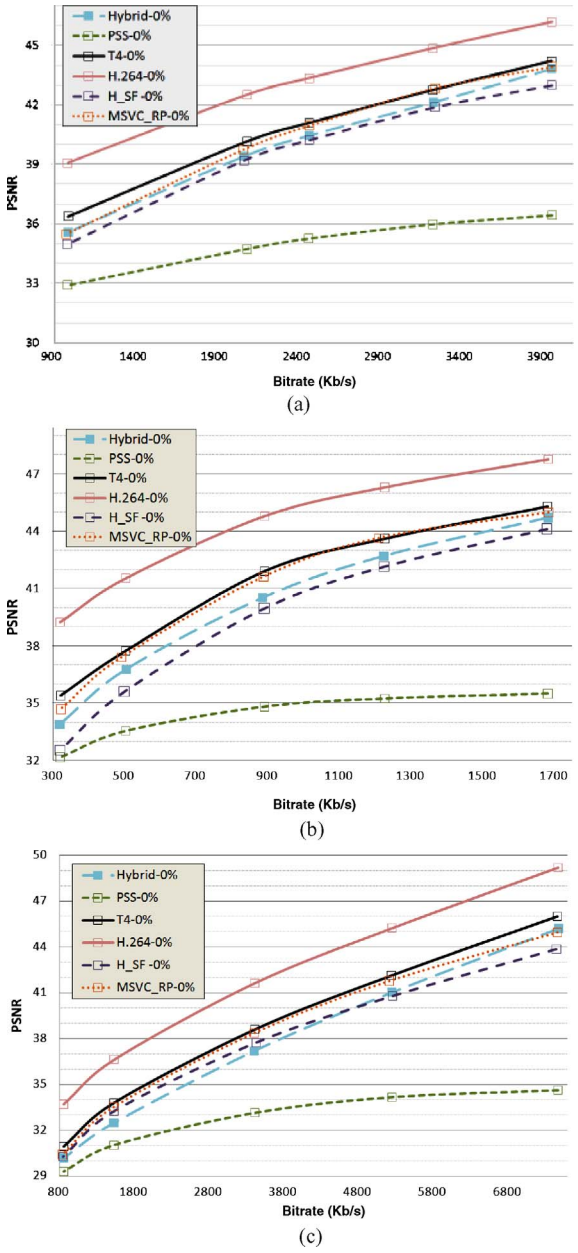

Fig. 14. Performance comparison in error-free environments. (a) *Foreman* sequence (CIF). (b) *News* sequence (CIF). (c) *Coastguard* sequence (CIF).

description, and H.264/AVC which encodes every frame as a single description uses four packets for each frame. In case of packet loss, T4 reconstructs the lost data by using B-PMVI which is the temporal estimation method used in Hybrid MDC, PSS adopted NNR and ES according to [7], H_SF adopted spatial and frequency estimation according to [11], MSVC_RP used discarding, replacement, and copying methods, respectively, in different loss cases for reconstruction [20], and H.264/AVC adopted basic EC described in [18].

The R-D performances with various PLRs, P_{loss}, for these methods are shown in Fig. 15, where the results are the averages of 100 independent simulation runs. It is observed that the R-D curve of H.264/AVC drops quickly as P_{loss} increases but the curves of the MDC methods drop gradually, confirming the advantage of MDC in improving the ER. Among five

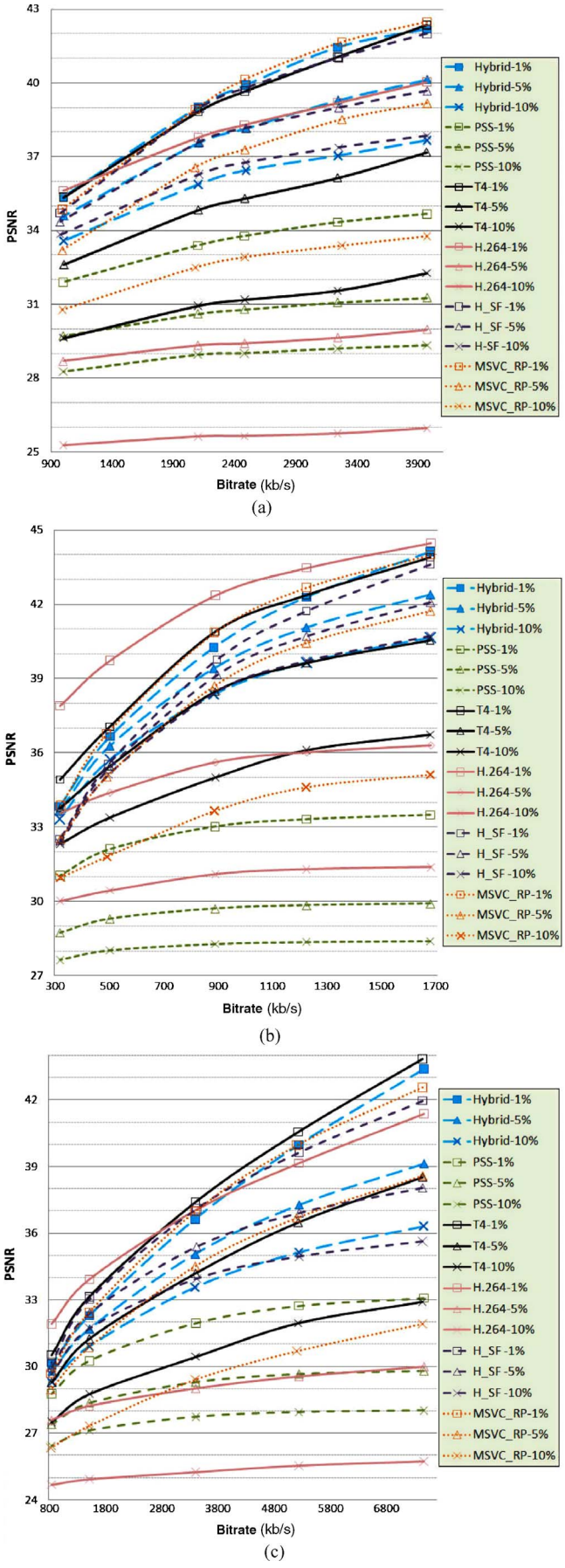

Fig. 15. Performance comparison in packet-loss environments. (a) *Foreman* sequence (CIF). (b) *News* sequence (CIF). (c) *Coastguard* sequence (CIF).

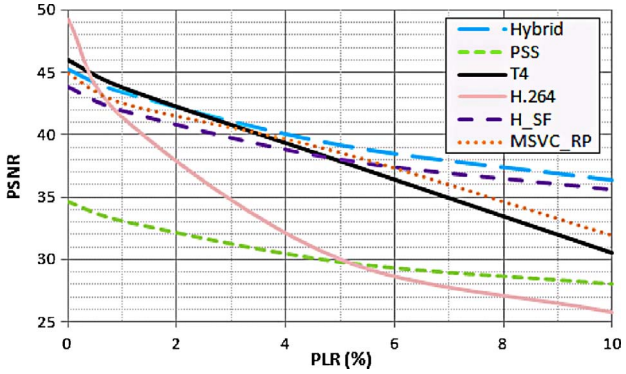

Fig. 16. PSNR as a function of PLR (*Coastguard* CIF at 7470 kb/s).

MDC methods, T4 is most sensitive to the increase of P_{loss}, and MSVC_RP is the second, even though both of them have the best performance in error-free environments (see Fig. 14). The performance gaps between T4 and Hybrid are large for $P_{loss} = 5\%$ and even up to 5 dB, 3.5 dB, and 3 dB in *Foreman*, *News*, and *Coastguard*, respectively, for $P_{loss} = 10\%$, while the gaps between MSVC_RP and Hybrid are up to 3.6 dB, 5.5 dB, and 4.2 dB in the three sequences, respectively, for $P_{loss} = 10\%$. The result indicates that the estimation methods adopted in Hybrid are superior to the methods used in T4 and MSVC_RP, and therefore, the PSNR gained from the estimation methods is able to compensate the loss of Hybrid in coding efficiency. In contrast to T4 and MSVC_RP, both H_SF and PSS are less sensitive to the increase of P_{loss}. However, due to inefficient coding, H_SF performs worse than Hybrid for low loss rates such as 0%, 1%, and 5%, and PSS performs worse than Hybrid for all the loss rates in all sequences. The result in Fig. 15 also shows that the proposed estimation methods for Hybrid are adaptive to various video sequences.

Fig. 16 presents the PSNR of various methods as a function of PLRs for *Coastguard* sequence (CIF). It can be seen that H.264 is most sensitive to the increase of PLR, T4 is the second, and MSVC_RP is the third. Even though they have high coding efficiency (i.e., high PSNR at PLR = 0%), their PSNRs drop quickly as the PLR increases. Compared to these three methods, the Hybrid, H_SF, and PSS are less sensitive to the increase of PLR. However, due to low coding efficiency, H_SF and PSS perform worse than Hybrid.

C. Frame-by-Frame Comparison

This section presents frame-by-frame PSNR comparison of different methods. Experiments were conducted for random packet loss with $P_{loss} = 5\%$ for *Foreman* CIF sequence at bitrate 2100 kb/s. The results are shown in Fig. 17, where the PSNR relative to error-free SDC are presented for the first 280 frames of *Foreman* sequence because these frames consist of high-activity and low-activity contents.

It is observed that when there is a dramatic PSNR drop for T4, it often drops periodically for every fourth frame. This is due to that T4 uses four prediction loops for every four consecutive frames. Since an error on one prediction loop will not propagate to the other three, the quality between successive frames is unbalanced for T4 method. Similarly, since Hybrid

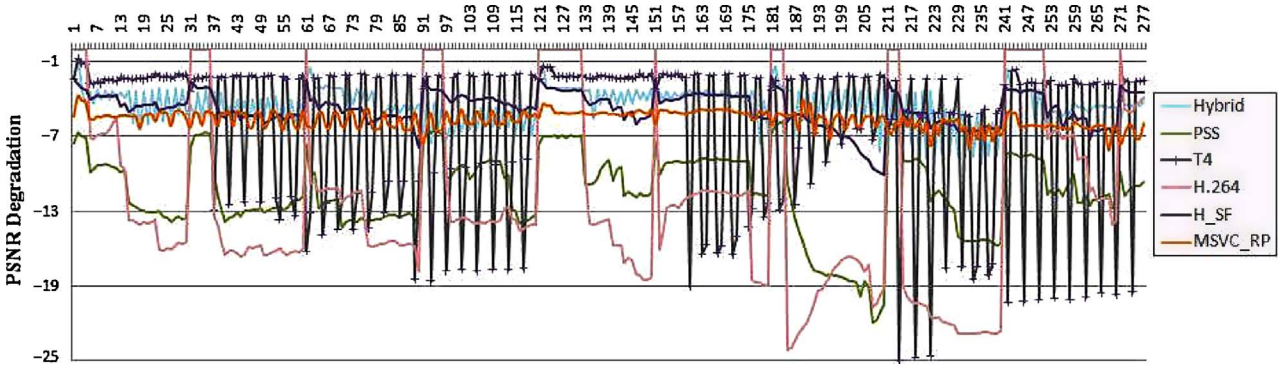

Fig. 17. Frame-by-frame PSNR for *Foreman* sequence (GoP size = 30) at 2100 kb/s with $P_{loss} = 5\%$.

adopts two prediction loops for every two consecutive frames, the quality is often degraded on every second frame when there is a packet loss. However, in comparison to T4, the PSNR variation is quite small. As for other four methods, since all frames use the same prediction loop(s), the error occurring on one prediction loop will be propagated to every succeeding frame. Hence, quality is degraded on every succeeding frame and the PSNR variation between successive frames is small.

In Fig. 17, it is also observed that even though Hybrid suffers from small PSNR variation between successive frames, it still has relatively stable and better overall performance among all methods. This is due to that the estimation methods adopted in Hybrid can effectively reconstruct lost data and thus suppress the error propagation.

D. Coding Efficiency Comparison

This section examines the coding efficiency of different MDC methods. The relative bit-rate, *R*, of a MDC method is defined as follows:

$$R = \frac{\text{Total bit-rates of four descriptions of MDC}}{\text{Bit-rates of SDC}} \quad (8)$$

where the SDC means the single description coding using H.264/AVC coder. Five QPs ranging from 18 to 34 are used and the results are shown in Table VII. As the table shows, T4 has the lowest R among the five MDC methods and Hybrid performs similar to T4. For *Foreman* and *Coastguard* sequences, T4 has lower R than Hybrid for small QPs, but Hybrid has lower R than T4 for large QPs. For *News* and *Mobile* sequences, although T4 performs better than Hybrid, the difference of R between them is quite small. Among the five MDC methods, PSS has the highest R and both H_SF and MSVC_RP have the R in between Hybrid and PSS. Furthermore, we define the relative center decoder quality, Q, of a MDC method as follows:

$$R = \frac{\text{PSNR of center performance of MDC method}}{\text{PSNR of SDC}} \quad (9)$$

where the center performance of a MDC method means the situation that all the descriptions of the MDC are received and correctly decoded without error. The center decoder qualities of the five MDC methods are shown in Table VIII. As the table shows, PSS has the lowest Q among all methods and the other four methods have similar Q, regardless of QPs and sequences.

TABLE VII
BIT-RATES OF MDC METHODS FOR CIF SEQUENCES

R	*Foreman*					*News*				
QP	Hybrid	PSS	T4	H_SF	MSVC-RP	Hybrid	PSS	T4	H_SF	MSVC-RP
18	1.44	2.01	1.39	1.56	1.89	1.82	1.72	1.64	1.93	1.96
22	1.59	2.47	1.56	1.61	2.09	1.95	1.89	1.73	2.09	2.13
26	1.69	2.81	1.75	1.79	2.69	2.04	2.04	1.78	2.30	2.57
30	1.85	2.82	1.96	2.01	2.52	2.09	2.07	1.78	2.31	2.35
34	1.95	2.12	2.06	2.25	2.29	2.11	1.84	1.74	2.27	2.37
R	*Mobile*					*Coastguard*				
QP	Hybrid	PSS	T4	H_SF	MSVC-RP	Hybrid	PSS	T4	H_SF	MSVC-RP
18	1.31	1.99	1.14	1.62	1.34	1.57	1.72	1.38	1.70	1.70
22	1.34	2.26	1.10	1.57	1.33	1.68	1.87	1.50	1.64	1.84
26	1.36	2.71	1.07	1.44	1.35	1.77	2.08	1.64	1.54	2.09
30	1.34	3.40	1.09	1.36	1.44	1.75	2.08	1.79	1.49	1.75
34	1.38	3.14	1.15	1.65	1.76	1.66	1.86	1.92	1.67	2.01

TABLE VIII
CENTER DECODER QUALITY OF MDC METHODS FOR CIF SEQUENCES

Q	*Foreman*					*News*				
QP	Hybrid	PSS	T4	H_SF	MSVC-RP	Hybrid	PSS	T4	H_SF	MSVC-RP
18	0.99	0.84	1.00	0.99	1.00	0.99	0.78	1.00	0.99	1.00
22	0.98	0.88	1.00	0.98	1.00	0.98	0.85	1.00	0.98	1.00
26	0.97	0.91	1.00	0.97	1.00	0.97	0.90	1.00	0.98	1.00
30	0.96	0.93	1.00	0.96	1.00	0.97	0.93	1.00	0.97	1.00
34	0.96	0.94	1.00	0.96	1.00	0.97	0.92	1.00	0.97	1.00
Q	*Mobile*					*Coastguard*				
QP	Hybrid	PSS	T4	H_SF	MSVC-RP	Hybrid	PSS	T4	H_SF	MSVC-RP
18	1.00	0.65	1.00	0.99	1.00	1.00	0.78	1.00	0.99	1.00
22	0.99	0.71	1.00	0.99	1.00	0.99	0.84	1.00	0.98	1.00
26	0.98	0.79	1.00	0.98	1.00	0.97	0.89	1.00	0.97	1.00
30	0.98	0.88	1.00	0.97	1.00	0.96	0.92	1.00	0.96	1.00
34	0.97	0.90	1.00	0.97	1.00	0.96	0.94	1.00	0.96	1.00

To summarize, PSS suffers from a relatively high coding bit-rate, resulting in the worst R-D performance among the methods. T4 benefits from relatively high coding efficiency and hence has the best R-D performance in error-free environments. However, T4 has a dramatic quality degradation as the PLR increases, showing its weakness in ER capability. H_SF suffers from relatively low coding efficiency, compared to T4 and Hybrid. Although its strong capability in EC enables it to outperform T4 in packet-loss environments, its R-D performance is still worse than that of Hybrid at low loss rates. MSVC benefits from high coding efficiency, but its weakness in error-resilient capacity makes it suffer from dramatic performance drops in case of packet loss. By splitting along temporal and spatial domains, the proposed Hybrid has the coding efficiency very close to T4 and yields a much better error-resilient performance than all other methods.

V. Conclusion

This paper proposed a joint spatial and temporal estimation method which takes advantages of data correlation in these two domains for better estimation of lost descriptions. The proposed method included fixed and adaptive approaches for estimation method selection. The fixed approach adopted the estimation methods based on the situations of description loss, while the adaptive approach adopted the estimation methods according to spatial and GTs of the lost pixels and thus, was adaptive to different kinds of video sequences. The proposed approaches were designed in the context of the hybrid MDC which segmented the video in both temporal and spatial domains. With the proposed estimation methods, the Hybrid MDC was adaptive to various video sequences and PLRs.

References

[1] A. Nafaa, T. Taleb, and L. Murphy, "Forward error correction strategies for media streaming over wireless networks," *IEEE Commun. Mag.*, vol. 46, no. 1, pp. 72–79, Jan. 2008.

[2] R. Zhang, S. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.

[3] C.-M. Fu, W.-L. Hwang, and C.-L. Huang, "Efficient post-compression error-resilient 3-D-scalable video transmission for packet erasure channels," in *Proc. IEEE ICASSP*, Mar. 2005, pp. 305–308.

[4] Y. Wang, A. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proc. IEEE*, vol. 93, no. 1, pp. 57–70, Jan. 2005.

[5] V. A. Vaishampayan, "Design of multiple description scalar quantizers," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, pp. 821–834, May 1993.

[6] O. Campana and R. Contiero, "An H.264/AVC video coder based on multiple description scalar quantizer," in *Proc. IEEE ACSSC*, Oct.–Nov. 2006, pp. 1049–1053.

[7] R. Bemardini, M. Durigon, R. Rinaldo, L. Celetto, and A. Vitali, "Polyphase spatial subsampling multiple description coding of video streams with H.264," in *Proc. IEEE ICIP*, Oct. 2004, pp. 3213–3216.

[8] J. Jia and H. K. Kim, "Polyphase downsampling based multiple description coding applied to H.264 video coding," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. E89-A, no. 6, pp. 1601–1606, Jun. 2006.

[9] J. G. Apostolopoulos, "Error-resilient video compression through the use of multiple states," in *Proc. IEEE ICIP*, vol. 3. Sep. 2000, pp. 352–355.

[10] S. Gao and H. Gharavi, "Multiple description video coding over multiple path routing networks," in *Proc. ICDT*, 2006, p. 42.

[11] C. W. Hsiao and W. J. Tsai, "Hybrid multiple description coding based on H.264," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 1, pp. 76–87, Jan. 2010.

[12] W. Zhu, Y. Wang, and Q.-F. Zhu, "Second-order derivative-based smoothness measure for error concealment in DCT-based codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 6, pp. 713–718, Oct. 1998.

[13] S. Cen and P. C. Cosman, "Decision trees for error concealment in video decoding," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 1–7, Mar. 2003.

[14] M. Ancis, D. D. Giusto, and C. Perra, "Error concealment in the transformed domain for DCT-coded picture transmission over noisy channels," *Eur. Trans. Telecommun.*, vol. 12, no. 3, pp. 197–204, 2001.

[15] S.-C. Hsia, S.-C. Cheng, and S.-W. Chou, "Efficient adaptive error concealment technique for video decoding system," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 860–868, Oct. 2005.

[16] Q. Peng, T. Yang, and C. Zhu, "Block-based temporal error concealment for video packet using motion vector extrapolation," in *Proc. IEEE Int. Conf. Commun., Circuits Syst. West Sino Expo.*, vol. 1. Jun. 2002, pp. 10–14.

[17] Y. Chen, K. Yu, J. Li, and S. Li, "An error concealment algorithm for entire frame loss in video transmission," in *Proc. IEEE PCS*, Dec. 2004, pp. 389–392.

[18] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 13, no. 7, pp. 657–673, Jul. 2003.

[19] H.264/AVC Reference Software: JM 13.2 [Online]. Available: http://iphome.hhi.de/suehring/tml

[20] I. Radulovic, P. Frossard, Y. K. Wang, M. M. Hannuksela, and A. Hallapuro, "Multiple description video coding with H.264/AVC redundant pictures," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 20, no. 1, pp. 144–148, Jan. 2010.

Wen-Jiin Tsai received the Ph.D. degree in computer science from National Chiao-Tung University (NCTU), Hsinchu, Taiwan, in 1997.

She is currently an Assistant Professor with the Department of Computer Science, NCTU. Before joining NCTU in 2004, she was with Zinwell Corporation, Hsinchu, as a Senior Research and Development Manager for six years. Her current research interests include video coding, video streaming, error-concealment, and error resilience techniques.

Jian-Yu Chen received the B.S. and M.S. degrees in computer science from National Chiao-Tung University, Hsinchu, Taiwan, in 2007 and 2009, respectively.

He is currently in compulsory military service in Taiwan. His current research interests include video codec firmware development.