

Error Concealment for Frame Losses in MDC

Mengyao Ma, *Student Member, IEEE*, Oscar C. Au, *Senior Member, IEEE*, Liwei Guo, *Student Member, IEEE*, S.-H. Gary Chan, *Senior Member, IEEE*, and Peter H. W. Wong, *Member, IEEE*

Abstract—Multiple description coding (MDC) is an effective error resilience (ER) technique for video coding. In case of frame loss, error concealment (EC) techniques can be used in MDC to reconstruct the lost frame, with error, from which subsequent frames can be decoded directly. With such direct decoding, the subsequent decoded frames will gradually recover from the frame loss, though slowly. In this paper we propose a novel algorithm using multihypothesis error concealment (MHC) to improve the error recovery rate of any EC in the temporal subsampling MDC. In MHC, the simultaneous temporal-interpolated frame is used as an additional hypothesis to improve the reconstructed video quality after the lost frame. Both subjective and objective results show that MHC can achieve significantly better video quality than direct decoding.

Index Terms—Error concealment, error propagation, error resilience, MDC, multihypothesis, temporal interpolation.

I. INTRODUCTION

DELIVERING video of good quality over the Internet or wireless networks is very challenging today, due to the use of predictive coding and *variable length coding* (VLC) in video compression [1], [2]. In block-based video coding, if INTER prediction mode is used, each macroblock (MB) is predicted from a previously decoded frame by *motion compensation*. One conventional approach is illustrated in Fig. 1(a), where each P-frame is predicted from its previous frame. Although the compression efficiency of this approach is high, it is vulnerable to errors in the transmission channel. If one frame is lost or corrupted (for example: P_4), the error in the reconstructed frame will propagate to the remaining frames until the next I-frame (I_{11}) is received. Thus, *error resilience* (ER) and *error concealment* (EC) techniques are developed to control and recover from the errors in video transmission.

Several ER methods have been developed, such as *forward error correction* (FEC) [3], *layered coding* [4], and *multiple description coding* (MDC) [5]. This paper is concerned with MDC. Different from the traditional *single description coding* (SDC), MDC divides the video stream into multiple equally important *streams* (*descriptions*), which are sent to the destination through different channels. Suppose the packet losses of all the channels are independently and identically distributed with

probability p . If we use SDC, the entire description is sent in one channel resulting in a loss probability of p . If we use MDC with M descriptions and send them in M channels, the probability of losing the entire description is p^M , which is much less than p . One simple but common implementation of MDC is the odd/even temporal subsampling approach: an even (odd) frame is predicted from the previous even (odd) frame, as illustrated in Fig. 1(b). Since the reference frames are farther in time, the prediction in such approach is not as good as in conventional SDC and the compression efficiency is lower. On the other hand, since each stream is encoded and transmitted separately, the corruption of one stream will not affect the other. As a result, the decoder can simply discard the corrupted stream until the next resynchronization point, and display the error-free video stream ($P_5P_7P_9\dots$) at half of the original frame rate. It can also reconstruct the corrupted frame by some appropriate error concealment (EC) method, and directly decode the subsequent frames.

There are many existing EC algorithms, such as spatial interpolation using some smoothness measure [6] and temporal compensation based on inter-frame correlation [7]. Many EC methods assume that only a few MBs or slices in a video frame are lost. However, in low bit-rate transmission applications, one frame is usually transmitted in one data packet in order to save transmission overhead. As a result, the loss of one packet will lead to the loss of one entire frame [8]. Therefore some EC algorithms assume whole frames are lost. Most methods estimate the lost motion vectors (MVs) at pixel or block level for a lost frame, based on the assumption of translational motion, and use the recovered MVs to fill the lost frame by copying pixels from the previous frame [8]–[10]. However, the methods are designed mainly for SDC, in which only past frames can be accessed in EC. To error-conceal a lost frame in MDC, a *temporal interpolation* method is more suitable, as MDC provides access to both past and future frames. As in the example in Fig. 1(c), when frame P_4 of stream 2 is corrupted during the MDC transmission, its surrounding frames (P_3 and P_5) would still be correct if stream 1 is error-free and they can be used to temporally interpolate P_4 .

Temporal interpolation was originally used to generate one or more frames between two received frames so as to improve the effective frame rate, while keeping smooth object motions in the video. As SDC is typically assumed, both forward and backward motion estimations are performed usually to track motions of the objects between adjacent received frames [11]. This leads to high computational complexity. In [12], *unidirectional motion compensated temporal interpolation* (UMCTI) is used, which performs only forward motion estimation and thus saves half of the computation time. By minimizing the prediction error variance between the original frame and the interpolated frame, the authors in [13] propose an optimal temporal interpolation filter

Manuscript received August 27, 2007; revised May 26, 2008. Current version published December 10, 2008. This work was supported in part by the Innovation and Technology Commission (Projects ITS/122/03 and GHP/033/05) of the Hong Kong Special Administrative Region, China. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Baochun Li.

The authors are with the Multimedia Technology Research Center (MTrec), Hong Kong University of Science and Technology (e-mail: myma@ust.hk; ecau@ust.hk; eeglw@ust.hk; gchan@ust.hk; eepeter@ust.hk).

Digital Object Identifier 10.1109/TMM.2008.2007282

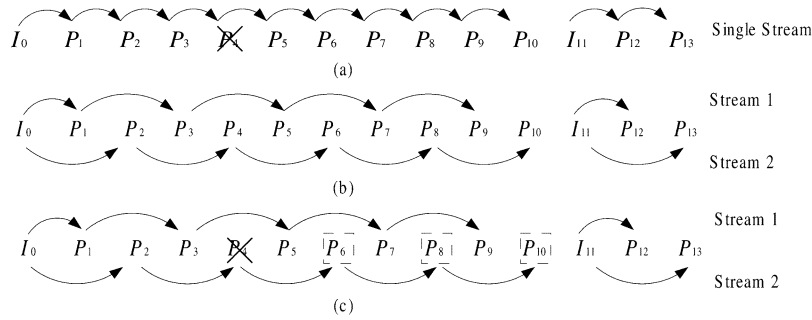


Fig. 1. Illustration of different approach for video coding. (a) Conventional video coding; (b) odd/even subsampling MDC; (c) error occurs in (b).

in which the interpolation filter taps are adapted based on motion vector reliability. When temporal interpolation is applied in MDC as an EC method, the MVs of the lost frame can be estimated by applying appropriate linear or nonlinear filtering along the motion trajectories from the future frame to the past frame [14], [15]. Based on these recovered MVs, corresponding blocks in the past or the future frame, or both of them, are used to recover the blocks in the lost frame. To further improve the concealed frame quality, some smoothness criteria can be imposed upon the neighboring MVs and/or the block boundaries, as in [16].

In conventional EC algorithms, only the corrupted (lost) frames are error-concealed and the subsequent frames are decoded directly leading to error propagation due to motion compensation. Nevertheless, the propagated error is known to reduce over time due to error suppression effects of the bilinear interpolation used in subpixel motion compensation and deblocking filters [17]. However, experiments show that the propagation error reduction rate (or error recovery rate) is low. While viewers might notice errors in the error-concealed frame, the error would be more pronounced with the low error recovery rate. Therefore it is desirable to develop some scheme to increase the error recovery rate. In this paper we propose a novel *multihypothesis error concealment* (MHC) algorithm, in which a number of the video frames after the lost one are error-concealed instead of decoded directly. A simultaneous temporal-interpolated frame is used as an additional hypothesis to improve the reconstructed video quality. The advantage of MHC is that it can be used to enhance many existing EC algorithms for MDC by increasing the error recovery rate and thus both objective and subjective video quality. In this paper, we choose one existing temporal interpolation algorithm, the UMCTI, and develop our proposed MHC around it. It should be noted that the proposed MHC can work with any other EC methods for frame losses.

The rest of this paper is organized as follows. In Section II, we describe the proposed MHC algorithm for MDC. Simulation comparisons between MHC and the direct decoding method are given in Section III. Section IV is the conclusion.

II. MULTIHYPOTHESIS ERROR CONCEALMENT FOR MDC

In this section, we will first introduce the proposed *multihypothesis error concealment* (MHC) algorithm in Section II-A. To control the emphasis on each hypothesis, weighting parameters (h_1 and h_2) are used. We use CMHC to denote MHC with

constant weights and AMHC to denote MHC with adaptively determined weights. We will then extend MHC to AMHC in Section II-B, based on the linear minimum mean square error (LMMSE) criterion. One control parameter (λ) is needed in AMHC, and it will be trained by experimental results in Section II-C.

A. Introduction to MHC

In temporal odd/even subsampling MDC, typically two descriptions are used, which are sent to the decoder through different channels. Let $\psi(n)$ be the original frame at time n , which is an $H \times W$ matrix. Consider the case of a single frame loss during the transmission and let n_0 be the time when the frame loss happens. In conventional EC algorithms, only the lost frame is error-concealed. Without loss of generality, suppose the lost frame $\psi(n_0)$ belongs to description 1 (D1) and $\psi_t(n_0)$ is the corresponding error-concealed frame. We will assume the error concealment is done by some temporal interpolation method. Based on $\psi_t(n_0)$, the subsequent frames can be decoded as usual, with the error in $\psi_t(n_0)$ propagated in D1 due to motion compensation. In [17], it was pointed out that spatial filtering, such as the bilinear interpolation for subpixel motion compensation and the deblocking filtering, can help to attenuate the propagated error energy. However, the error reduction rate is low as verified in our experiments. As low error recovery rate can greatly degrade the overall subjective video quality, it is desirable to develop some scheme to increase the error recovery rate.

As the video frames in description 2 (D2) are correctly received, they can be used to obtain an additional estimation for frame $\psi(n_0+2), \psi(n_0+4), \psi(n_0+6), \dots$, using some temporal interpolation method, most likely the same one as that used for $\psi(n_0)$. Based on this observation, we propose a *multihypothesis error concealment* (MHC) algorithm, in which the simultaneous temporally interpolated frame is used as an additional hypothesis to improve the reconstructed video quality of the frames after $\psi(n_0)$ in D1. The flowchart of MHC is shown in Fig. 2.

Consider frame $\psi(n_0 + 2k)$ for $k = 1, 2, \dots, N$. Suppose $\psi(n_0 + 2k - 2)$ has been previously reconstructed by temporal interpolation ($k = 1$) or the proposed MHC ($k > 1$) and let $\psi_r(n_0 + 2k - 2)$ be the corresponding reconstructed frame. As the data for frame $\psi(n_0 + 2k)$ (such as motion vectors and DCT coefficients, etc.) are correctly received, $\psi(n_0 + 2k)$ can be decoded directly using $\psi_r(n_0 + 2k - 2)$ as the reference frame. Let $\psi_d(n_0 + 2k)$ be that decoded frame. We also apply temporal interpolation on the two neighboring frames, $\psi(n_0 + 2k - 1)$ and

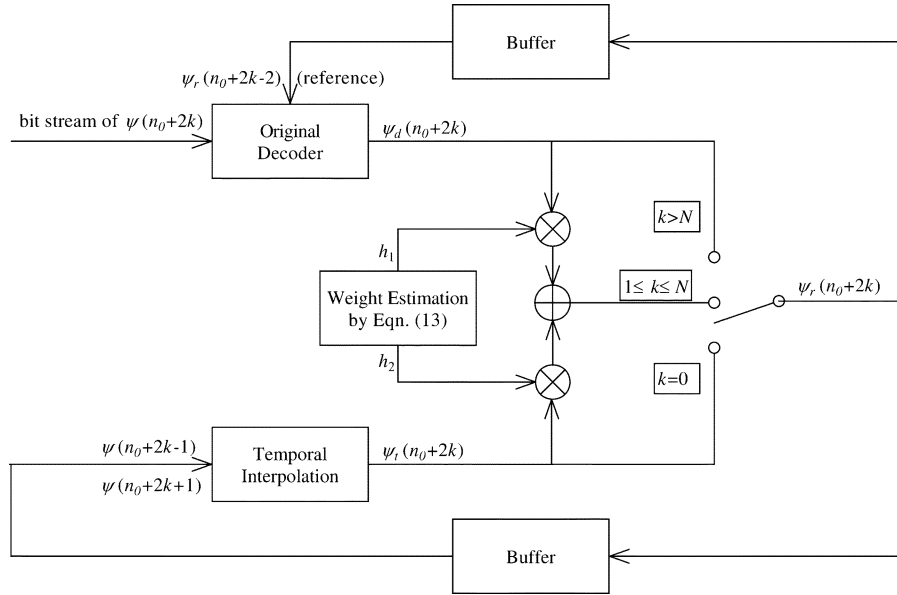


Fig. 2. Illustration for multihypothesis reconstruction of frame $\psi(n_0 + 2k)$.

$\psi(n_0 + 2k + 1)$, to get an additional estimate $\psi_t(n_0 + 2k)$ for $\psi(n_0 + 2k)$. We then form a linear combination of $\psi_t(n_0 + 2k)$ and $\psi_d(n_0 + 2k)$ to reconstruct $\psi(n_0 + 2k)$ by

$$\psi_r(n_0 + 2k) = h_1 \psi_d(n_0 + 2k) + h_2 \psi_t(n_0 + 2k) \quad (1)$$

where $k = 1, 2, \dots, N, h_1, h_2 \in [0, 1]$ and $h_1 + h_2 = 1$. The multihypothesis reconstruction is applied only for a limited *time interval* N immediately after $\psi(n_0)$. For $k = 0$, we have $\psi_r(n_0) = \psi_t(n_0)$. Note that if we set $h_1 = 1$ in (1) or use zero *time interval* ($N = 0$), MHC becomes a conventional decoder. We will discuss the values for parameter h_1 and h_2 in Section II-B. In order to obtain $\psi_t(n_0 + 2k)$, frame $\psi(n_0 + 2k + 1)$ needs to be decoded first. Thus a one-frame delay is introduced here. However, as demonstrated by the simulation results in Section III, the reconstructed video quality can be greatly improved at the cost of such a short delay.

B. MHC With Adaptive Weights (AMHC)

For simplicity, the weights h_1 and h_2 in (1) can be constant for $k \in [1, N]$. In this paper, we propose to have them adaptively determined based on the linear minimum mean square error (LMMSE) criterion. For better illustration, the important symbols used to derive the weighting parameters are listed in Table I.

1) *Deriving h_1 and h_2 Based on LMMSE*: Let $\tilde{\psi}(n_0 + 2k)$ be the original reconstructed frame of $\psi(n_0 + 2k)$ at the encoder side. We define the error of the decoded frame $\psi_d(n_0 + 2k)$, the temporal-interpolated frame $\psi_t(n_0 + 2k)$ and the reconstructed frame $\psi_r(n_0 + 2k)$ by MHC with respect to $\tilde{\psi}(n_0 + 2k)$ to be $\epsilon_d(k)$, $\epsilon_t(k)$ and $\epsilon_r(k)$, respectively.

$$\begin{aligned} \epsilon_d(k) &= \psi_d(n_0 + 2k) - \tilde{\psi}(n_0 + 2k), \\ \epsilon_t(k) &= \psi_t(n_0 + 2k) - \tilde{\psi}(n_0 + 2k), \\ \epsilon_r(k) &= \psi_r(n_0 + 2k) - \tilde{\psi}(n_0 + 2k). \end{aligned}$$

For the sake of simplicity, we have omitted n_0 in the indices of $\epsilon_d(k)$, $\epsilon_t(k)$ and $\epsilon_r(k)$. Here $\epsilon_d(k)$, $\epsilon_t(k)$ and $\epsilon_r(k)$ are all $H \times W$ matrices and we define $\epsilon_d(k, x, y)$, $\epsilon_t(k, x, y)$ and $\epsilon_r(k, x, y)$ to be their corresponding matrix element at the y th row and x th column, respectively, $x \in [0, W - 1]$ and $y \in [0, H - 1]$. Assume $\epsilon_d(k, x, y)$ and $\epsilon_t(k, x, y)$ are two independent variables with zero mean, i.e., $E[\epsilon_d(k, x, y)] = E[\epsilon_t(k, x, y)] = 0$ which implies $E[\epsilon_r(k, x, y)] = 0$ by (1). Although this is not necessarily a very accurate assumption, especially when both descriptions are corrupted by frame losses, it can greatly simplify the analysis and implementation, and provide satisfactory results. Let $\sigma_i^2(k, x, y)$ be the variance of $\epsilon_i(k, x, y)$, where i represents d, k or r . Note that $\epsilon_i(k, x, y)$ is a random variable and it is difficult to estimate its variance $\sigma_i^2(k, x, y)$. Let $\bar{\sigma}_i^2(k)$ be the mean square values of $\epsilon_i(k, x, y)$ over $x \in [0, W - 1]$ and $y \in [0, H - 1]$, i.e., $\bar{\sigma}_i^2(k) = (1/HW) \sum_{x \in [0, W-1]} \sum_{y \in [0, H-1]} \epsilon_i^2(k, x, y)$. For simplicity, we assume $\bar{\sigma}_i^2(k, x, y)$ are the same for all x and y within the same frame. Applying this assumption, it is reasonable for us to use $\bar{\sigma}_i^2(k)$ to estimate $\sigma_i^2(k, x, y)$. In other words, we have $\sigma_d^2(k, x, y) \simeq \bar{\sigma}_d^2(k)$, $\sigma_t^2(k, x, y) \simeq \bar{\sigma}_t^2(k)$ and $\sigma_r^2(k, x, y) \simeq \bar{\sigma}_r^2(k)$. We want to find the optimal h_1 and h_2 in (1) to minimize $E[(\psi_r(n_0 + 2k, x, y) - \tilde{\psi}(n_0 + 2k, x, y))^2]$, or $E[\epsilon_r^2(k, x, y)]$. This becomes a linear minimum mean square error (LMMSE) problem and the solution is well known to be

$$h_1 = \frac{\bar{\sigma}_t^2(k)}{\bar{\sigma}_d^2(k) + \bar{\sigma}_t^2(k)}, \quad h_2 = \frac{\bar{\sigma}_d^2(k)}{\bar{\sigma}_d^2(k) + \bar{\sigma}_t^2(k)}. \quad (2)$$

With these optimal h_1 and h_2 values, the mean square error of the MHC reconstructed frame becomes

$$\bar{\sigma}_r^2(k) = \frac{\bar{\sigma}_d^2(k) \times \bar{\sigma}_t^2(k)}{\bar{\sigma}_d^2(k) + \bar{\sigma}_t^2(k)}, \quad k \in [1, N]. \quad (3)$$

2) *Estimating $\bar{\sigma}_d^2(k)$ and $\bar{\sigma}_t^2(k)$* : In order to reconstruct frame $\psi(n_0 + 2k)$ by (1) and (2), we need to estimate the values of $\bar{\sigma}_d^2(k)$ and $\bar{\sigma}_t^2(k)$ first. As stated previously, spatial filtering can

attenuate the propagated error energy. It can be introduced by deblocking filters, or as a side effect of subpixel motion compensation with linear interpolation [17]. In [18], this effect is analyzed and approximated by a separable average loop filter. In this paper, we use a similar approximation. Assume the loop filter to be f , which is time invariant. Then error in the decoded frame $\psi_d(n_0 + 2k)$ can be calculated as

$$\epsilon_d(k, x, y) = \epsilon_r(k - 1, x, y) * f(x, y), \quad k \in [1, N] \quad (4)$$

where $*$ denotes linear convolution, $x \in [0, W - 1]$ and $y \in [0, H - 1]$. Suppose the power spectral density (PSD) of $\epsilon_r(k)$ is $H_r(k)$ and $F(\omega_x, \omega_y)$ is the frequency representation of f . Based on the *Wiener-Khinchine relation* and *Fourier transform theory* [19] we have

$$\bar{\sigma}_d^2(k) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \|F(\omega_x, \omega_y)\|^2 \times H_r(k - 1, \omega_x, \omega_y) d\omega_x d\omega_y \quad (5)$$

$k \in [1, N]$. As $F(\omega_x, \omega_y)$ works like a lowpass filter, we approximate it to have a Gaussian shape, i.e.,

$$\|F(\omega_x, \omega_y)\|^2 = \exp[-(\omega_x^2 + \omega_y^2) \sigma_f^2] \quad (6)$$

where σ_f^2 can be used to control the strength of the filter. In addition to the Gaussian approximation for $\|F(\omega_x, \omega_y)\|^2$, we also approximate the PSD of $\epsilon_r(k)$ to be Gaussian:

$$H_r(k, \omega_x, \omega_y) = 4\pi \bar{\sigma}_r^2(k) \sigma_g^2(k) \exp[-(\omega_x^2 + \omega_y^2) \sigma_g^2(k)]. \quad (7)$$

Here $\bar{\sigma}_r^2(k)$ is the mean square value of $\epsilon_r(k)$ and the parameter $\sigma_g^2(k)$ determines the shape of the PSD. As stated previously that we only apply multihypothesis error concealment to a few frames after the lost one ($\psi(n_0)$), i.e., $k \in [1, N]$ in (1). With a moderate value of N , we expect the shape of the PSD to be similar for $k \in [0, N]$, i.e., $\sigma_g^2(k) \approx \sigma_g^2$. Then (7) can be rewritten as

$$H_r(k, \omega_x, \omega_y) = 4\pi \bar{\sigma}_r^2(k) \sigma_g^2 \exp[-(\omega_x^2 + \omega_y^2) \sigma_g^2]. \quad (8)$$

Based on the approximation in (6) and (8), we can solve (5) and get

$$\bar{\sigma}_d^2(k) = \frac{\bar{\sigma}_r^2(k - 1) \times \sigma_g^2}{\sigma_g^2 + \sigma_f^2} = \frac{\bar{\sigma}_r^2(k - 1)}{\lambda + 1}, \quad k \in [1, N] \quad (9)$$

with $\lambda = \sigma_f^2 / \sigma_g^2$, $\lambda > 0$. In addition, we have $\bar{\sigma}_r^2(0) = \bar{\sigma}_t^2(0)$ as the reconstructed frame at time n_0 is the temporally interpolated frame. By incorporating (3) into (9), we can get

$$\bar{\sigma}_d^2(k) = \left[\sum_{j=1}^k \frac{(\lambda + 1)^j}{\bar{\sigma}_t^2(k - j)} \right]^{-1}, \quad k \in [1, N]. \quad (10)$$

As typically the same temporal interpolation method is applied to $\psi_t(n_0 + 2k)$ and $\psi_t(n_0)$, the error energy of $\psi_t(n_0 + 2k)$ should be approximately the same as that of $\psi_t(n_0)$, i.e.,

$$\bar{\sigma}_t^2(k) \approx \bar{\sigma}_t^2(0) = \bar{\sigma}_t^2, \quad k \in [1, N]. \quad (11)$$

Then (10) can be simplified to

$$\bar{\sigma}_d^2(k) = \frac{\lambda \bar{\sigma}_t^2}{(\lambda + 1)[(\lambda + 1)^k - 1]}, \quad k \in [1, N]. \quad (12)$$

3) *Summary for AMHC*: With the approximations for $\bar{\sigma}_t^2(k)$ in (11) and $\bar{\sigma}_d^2(k)$ in (12), (2) can be simplified to be

$$h_1 = \frac{(\lambda + 1)^{k+1} - (\lambda + 1)}{(\lambda + 1)^{k+1} - 1}, \quad h_2 = \frac{\lambda}{(\lambda + 1)^{k+1} - 1}, \quad (13)$$

which can be used to reconstruct frame $\psi(n_0 + 2k)$ as in (1), $k \in [1, N]$.¹ Note that only one parameter λ is needed in this algorithm to control the variation of h_1 and h_2 . As $\lambda = \sigma_f^2 / \sigma_g^2$, it is related to the strength of the loop filter f and the error concealment method used. The value of λ will be trained using some training video sequences in Section II-C. From (13) we notice that h_1 (h_2) is an increasing (decreasing) function of the time offset k . This is reasonable as the propagated error can be reduced by motion compensation and thus the error energy in the decoded frame, $\bar{\sigma}_d^2(k)$, usually decreases over time. On the other hand, the error energy of the temporal-interpolated frames, $\bar{\sigma}_t^2(k)$, remains approximately the same because the same interpolation algorithm is used to reconstruct these frames. When k approaches infinity, we have $\bar{\sigma}_d^2 = 0$, $\bar{\sigma}_r^2 = 0$, and $h_2 = 0$ in (13) which is reasonable because, when the error of the reconstructed frame approaches zero, there is no need to refine it with the hypothesis obtained by temporal interpolation.

To compare an algorithm with AMHC and without AMHC, we define *error reduction ratio* (R),

$$R(k) = \frac{\bar{\sigma}_r^2(k - 1)}{\bar{\sigma}_r^2(k)}, \quad k \geq 1 \quad (14)$$

to measure the relative error reduction from frame to frame. A larger R indicates a faster error reduction speed which is desirable. Using (9) and (12), we can get the error reduction ratio for AMHC.

$$R_{\text{amhc}}(k) = \lambda + 1 + \frac{\lambda}{(\lambda + 1)^k - 1}. \quad (15)$$

If the lost frame is error-concealed without the proposed MHC applied to the subsequent frames, we have $\bar{\sigma}_r^2(k) = \bar{\sigma}_d^2(k)$ in (9). The corresponding error reduction ratio is

$$R_{\text{no_mhc}}(k) = \frac{\bar{\sigma}_r^2(k - 1)}{\bar{\sigma}_r^2(k)} = \lambda + 1. \quad (16)$$

It is easy to verify that $R_{\text{amhc}}(k) > R_{\text{no_mhc}}(k)$ with $\lambda > 0$, i.e., AMHC can help to increase the error reduction speed.

C. Estimating Parameter λ for AMHC

As discussed in Section II-B, the parameter λ is needed to control the rate of increase (decrease) of h_1 (h_2) in AMHC. Its value is related to the strength of the loop filter f and the

¹Theoretically the estimation of $\bar{\sigma}_d^2(k)$ by (10) is more accurate than that by (12). However, as it is difficult to estimate $\bar{\sigma}_t^2(k)$ accurately from the received data, the algorithm using (10) performs similar to that using (12). So in this paper we will use the simplified one, (12). The advantage of using this equation is that the estimation of $\bar{\sigma}_t^2$ can be left out as it can be cancelled in (13).

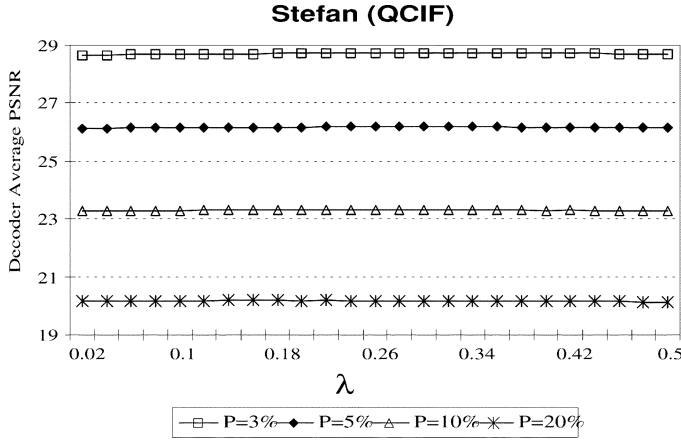


Fig. 3. Sensitivity study of AMHC with different values of λ . The random frame loss rate is $P = 3\%$, 5% , 10% , or 20% .

TABLE I
LIST OF SYMBOLS USED IN SECTION II-B

Symbols	Meanings
$\psi(n_0 + 2k)$	The video frame at time $(n_0 + 2k)$, which is an $H \times W$ matrix
$\tilde{\psi}(n_0 + 2k)$	The encoder reconstructed frame $\psi(n_0 + 2k)$
$\psi_d(n_0 + 2k)$	The directly decoded frame $\psi(n_0 + 2k)$
$\psi_t(n_0 + 2k)$	The error-concealed frame $\psi(n_0 + 2k)$ by temporal interpolation
$\psi_r(n_0 + 2k)$	The finally reconstructed frame $\psi(n_0 + 2k)$ by MHC
$\epsilon_d(k)$	Error in frame $\psi_d(n_0 + 2k)$
$\epsilon_t(k)$	Error in frame $\psi_t(n_0 + 2k)$
$\epsilon_r(k)$	Error in frame $\psi_r(n_0 + 2k)$
$\bar{\sigma}_d^2(k)$	The mean square error in frame $\psi_d(n_0 + 2k)$
$\bar{\sigma}_t^2(k)$	The mean square error in frame $\psi_t(n_0 + 2k)$
$\bar{\sigma}_r^2(k)$	The mean square error in frame $\psi_r(n_0 + 2k)$
f	The loop filter to approximate the deblocking filter and sub-pixel motion compensation
F	The frequency representation of f
$H_r(k)$	The power spectral density (PSD) of $\epsilon_r(k)$

temporal interpolation method used. In this paper, we simply use an existing algorithm, *unidirectional motion compensated temporal interpolation* (UMCTI), to interpolate the lost frame [12]. UMCTI is chosen because it gives concealed frames with good visual quality and requires low computational complexity. Actually other temporal interpolation algorithms can also be used. To find a reasonable λ , we make a simplifying assumption that λ is the same for all sequences under all coding conditions. To estimate λ , we observe that (12) can be simplified to $\lambda = \bar{\sigma}_t^2 / \bar{\sigma}_d^2(1) - 1$, with $N = 1$ and $k = 1$. Training values of λ can be obtained by performing temporal interpolation to MDC-encoded training sequences. At each possible location n_0 in a training sequence, we perform UMCTI to generate the concealed frame $\psi_t(n_0)$ using the neighboring frames. Comparing $\psi_t(n_0)$ with the error-free decoded frame $\tilde{\psi}_t(n_0)$, we can calculate $\bar{\sigma}_t^2$. Then we decode directly the frame at $(n_0 + 2)$ using $\psi_t(n_0)$ as reference frame, and calculate $\bar{\sigma}_d^2(1)$. With both $\bar{\sigma}_t^2$ and $\bar{\sigma}_d^2(1)$, we obtain one training value of λ . We repeat this process for all the possible n_0 in an MDC-encoded training sequence to obtain many training values of λ . By averaging the training values, we then obtain an estimated λ for that sequence.

TABLE II
AVERAGE λ FOR THE CIF AND QCIF SEQUENCES WITH DIFFERENT QP

Average λ of CIF sequence						
QP	22	26	30	34	38	42
<i>Akiyo</i>	0.183	0.141	0.095	0.069	0.052	0.036
<i>Mobile</i>	0.233	0.210	0.183	0.147	0.110	0.082
<i>News</i>	0.348	0.346	0.335	0.350	0.273	0.246
<i>Coastguard</i>	0.236	0.205	0.179	0.143	0.109	0.077
<i>Weather</i>	0.413	0.235	0.162	0.110	0.088	0.066
Average	0.283	0.227	0.191	0.164	0.126	0.101
Average λ of QCIF sequence						
QP	22	26	30	34	38	42
<i>Sales</i>	0.318	0.287	0.217	0.192	0.136	0.123
<i>Bus</i>	0.117	0.115	0.118	0.136	0.135	0.118
<i>Carphone</i>	0.188	0.152	0.227	0.219	0.222	0.173
<i>Miss_Am</i>	0.190	0.149	0.116	0.086	0.062	0.056
<i>Stefan</i>	0.149	0.145	0.142	0.170	0.193	0.268
Average	0.192	0.170	0.164	0.161	0.150	0.148

We used ten training sequences. Among them, five are CIF sequences encoded at 30 fps (*Akiyo*, *Mobile*, *News*, *Coastguard* and *Weather*), and five are QCIF sequences encoded at 15 fps (*Sales*, *Bus*, *Carphone*, *Miss_Am* and *Stefan*). All of them have 300 frames except *Miss_Am* and *Bus*, which have 150 frames. Each sequence is encoded with a fixed QP. Six common QP values are used, $QP \in \{22, 26, 30, 34, 38, 42\}$. The average λ values are shown in Table II. All the λ values are greater than zero, as expected. In the experiment, they range from 0.036 to 0.413, with an overall average of 0.173. For any sequence, the estimated λ appears to be different for different QP and the range of λ appears to be larger in CIF sequences than in QCIF sequences. This contradicts our assumption that λ is the same for all sequences under all coding conditions. In light of this, we performed a sensitivity study by simulating the proposed AMHC with a fixed λ for all the training sequences. Twenty-five values of λ are studied ranging from 0.02 to 0.5 with a step of 0.02. The typical results are shown in Fig. 3, with QP = 30 and random frame loss rate $P = 3\%$, 5% , 10% , or 20% . We find that the performance of AMHC is not very sensitive to the choice of λ . The PSNR typically changes slowly with λ , with an average PSNR fluctuation (i.e., max-min) of 0.066 dB. Among all the 25 values, $\lambda = 0.02$ and $\lambda = 0.5$ typically gives the worst PSNR. The PSNR of $\lambda = 0.2$ appears to be close to the optimal PSNR most of the time. As a result, for the sake of simplicity and abiding by our assumption, we will choose $\lambda = 0.2$ for all sequences under all simulation conditions.

III. SIMULATION RESULTS

In the simulation, we compare the performance of the proposed MHC with that of direct decoding (DD), which is equivalent to MHC with $N = 0$ or $h_1 = 1$. In DD, only the lost frames are concealed with temporal interpolation and the subsequent frames are decoded directly. Both MHC with constant weights (CMHC) and MHC with adaptively determined weights (AMHC) are simulated. UMCTI [12] is used as the temporal interpolation method in both MHC and DD. In the special case of consecutive frame losses, *copying-previous* is used to reconstruct the lost frames.

We use the H.264/AVC reference software version 8.2 (baseline profile) for the simulation [20]. The first 300 frames of video

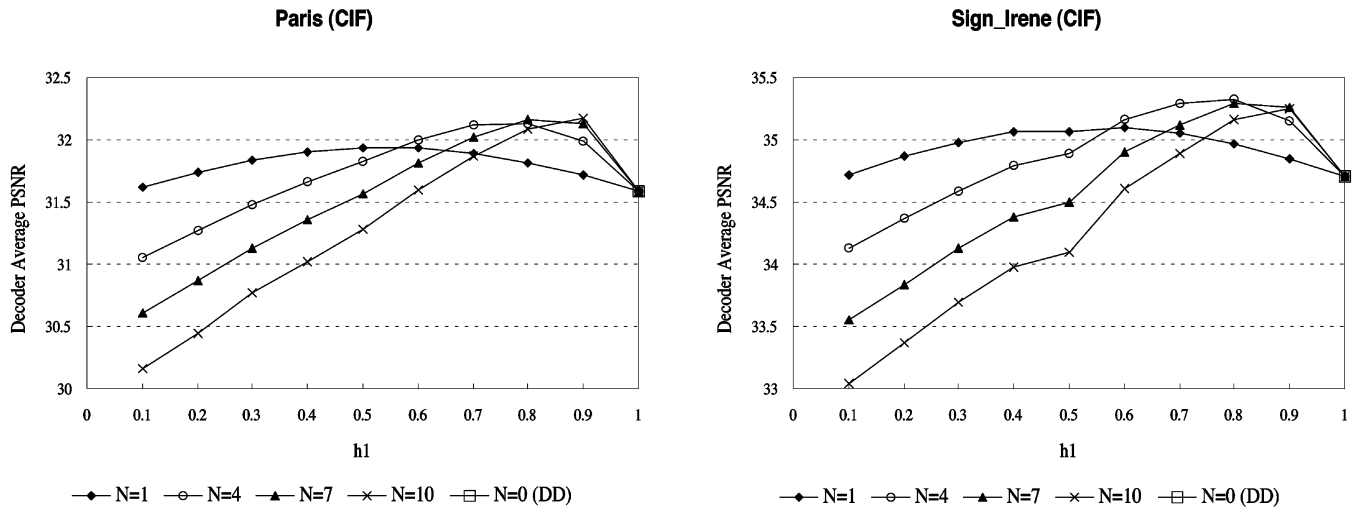


Fig. 4. Average PSNR at the decoder side for CMHC with different h_1 . The packet loss rate is $P = 3\%$. The corresponding encoder PSNRs (in the error-free case) are 33.82 dB for *Paris* and 37.18 dB for *Sign_Irene*.

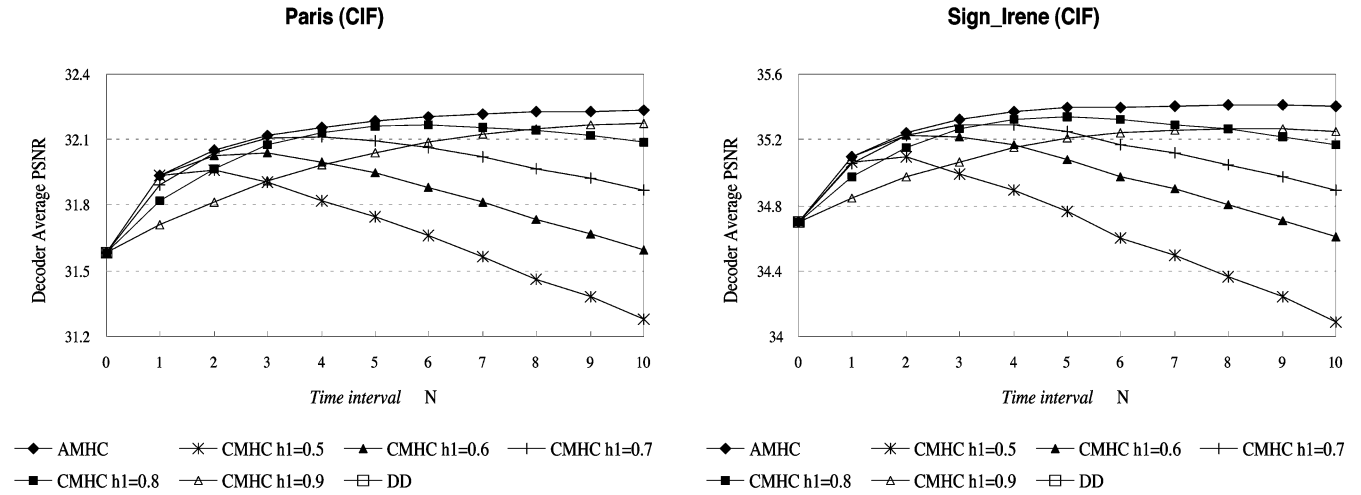


Fig. 5. Comparison of CMHC and AMHC for different *time interval* N . Parameter $\lambda = 0.2$ for AMHC. The packet loss rate is $P = 3\%$. The error-free PSNRs are 33.82 dB for *Paris* and 37.18 dB for *Sign_Irene*.

sequences *Paris* (CIF, 30 fps), *Sign_Irene* (CIF, 30 fps), *Foreman* (QCIF, 15 fps) and *Hall_Monitor* (QCIF, 15 fps) are encoded for the testing. Note that these are different from the sequences used for training λ . For each sequence, if not stated explicitly, only the first frame is encoded as I-frame, and all the subsequent ones are encoded as P-frames. To generate two descriptions, *ref_idx_10* is adjusted to simulate the odd/even subsampling MDC. One fixed QP is used to encode a whole sequence, and its value is adjusted to achieve different bit rate. The search range for motion estimation is $[-32, 32]$ for CIF sequences and $[-16, 16]$ for QCIF sequences. In the simulation, the two descriptions are transmitted through two channels with independent packet losses. One packet contains the information of one frame, and the loss of one packet will lead to the loss of one entire frame. The simulated packet loss patterns are obtained from [21], with loss rates $P = 3\%, 5\%, 10\%$ and 20% . Given a packet loss rate P , the video sequence is transmitted 40 times,

and the average PSNR for the 40 transmissions is calculated at the decoder side.

A. Comparison Between CMHC and AMHC

We first test the effect of the weighting parameter h_1 on the performance of CMHC. Four different *time intervals* (N) are used, i.e., $N = 1, 4, 7$, or 10 , and the packet loss rate is $P = 3\%$. The video sequences are encoded with a fixed QP ($QP = 30$), and the average PSNR at the decoder side is plotted in Fig. 4. The result of DD is also shown for the sake of comparison. From the figure we can see that there is an optimal h_1 for any N to give maximum PSNR in the corresponding curve, and the optimal h_1 tends to increase with N . When CMHC is applied to only one frame ($N = 1$) after the frame loss with the simple choice of $h_1 = 0.5$, there is a meaningful PSNR gain of 0.35 dB in *Paris* and 0.36 dB in *Sign_Irene* compared to DD. When N is larger and CMHC is applied to more frames, more PSNR gain

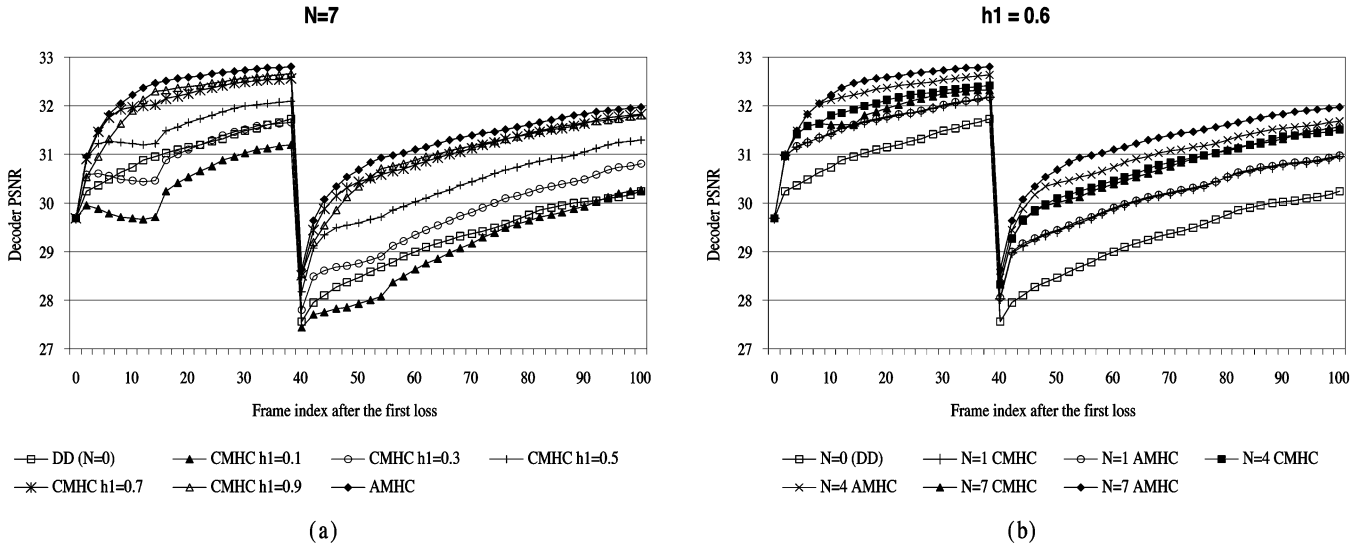


Fig. 6. Error propagation in *Paris* ($QP = 30$) for multiple frame losses. Only the PSNR of one description is plotted. The error-free PSNR is 33.77 dB.

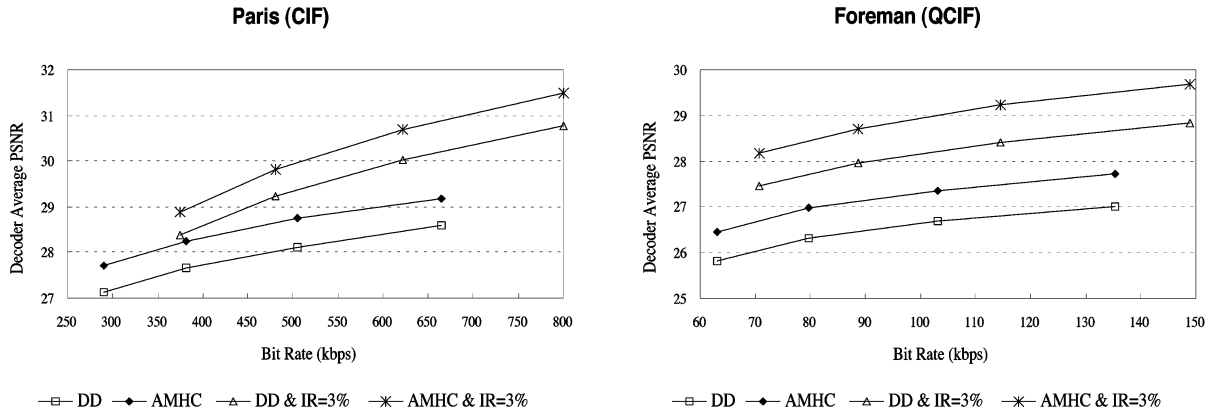


Fig. 7. RD curves of DD and AMHC ($\lambda = 0.2$, $N = 5$) under packet loss rate $P = 10\%$.

can be achieved with an optimal h_1 . Interestingly, the maximum PSNR of each curve appears to be achieved with $h_1 \geq 0.5$. This is reasonable because the propagated error can be decreased by motion compensation and deblocking filter and thus the error in the directly decoded frame should be less than that in the temporally interpolated one. In other words, we should give a larger weight to the decoded frame in (1), i.e., $h_1 \geq h_2$.

In Fig. 5, the comparison between AMHC and CMHC is given for different *time interval* N . The video sequences are compressed with $QP = 30$ and the packet loss rate is $P = 3\%$. For CMHC, only the results with $h_1 \geq 0.5$ are presented as its performance for $h_1 < 0.5$ is not good. From the figure we can see that the performance of AMHC is better than CMHC, i.e., the curve of AMHC lies above those of CMHC for all the N . There are also some interesting behavior of CMHC and AMHC in Fig. 5. For CMHC with a specific h_1 , there is an N at which maximum PSNR is achieved in the corresponding curve. The optimal N tends to increase with h_1 . On the other hand, for AMHC, the PSNR curve tends to increase monotonically with N , very fast at first and then levelling off for large N suggesting that most of the benefits are obtained at the early N . As the com-

plexity of AMHC increases with N due to the temporal interpolation, a moderate value of N is appropriate in reality. Thus, we will choose to use $N = 5$ for AMHC in our later simulations.

To further illustrate how error propagates over time in CMHC and AMHC, we plot the decoder PSNR in the case of multiple single frame losses in Fig. 6. Recall that n_0 is the time when the first frame loss happens. Here we call the description associated with n_0 as D1. Assume frames $(n_0 + 19)$ and $(n_0 + 40)$ are also lost, which belong to description D2 and D1 respectively. Each curve is the average behavior of CMHC/AMHC obtained by 40 simulations, each of which has different instance of n_0 . For clearer illustrations, only the PSNR of description D1 is plotted. In Fig. 6(a), we fix the *time interval* $N = 7$ and show the results for different weight (h_1) of CMHC. The curve of DD is also plotted for comparison. As expected, the PSNR of DD increases over time because the deblocking filter and the subpixel interpolation in DD can inherently reduce the propagated error energy. However, the error reduction is slow leading to relatively poor subjective user experience. Among all the curves, the AMHC is always the highest with the highest initial error reduction rate. Note that in addition to the frame losses in D1, error also oc-

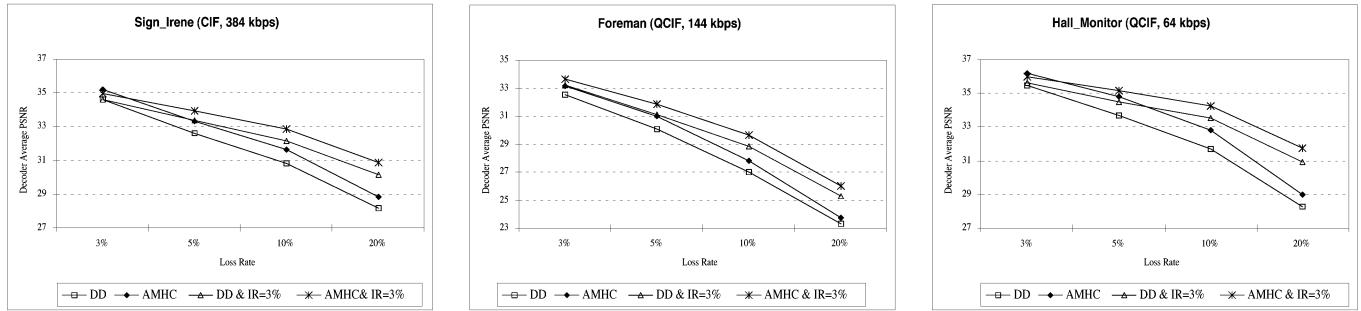


Fig. 8. Comparison of DD and AMHC ($\lambda = 0.2, N = 5$) for different packet loss rate P .

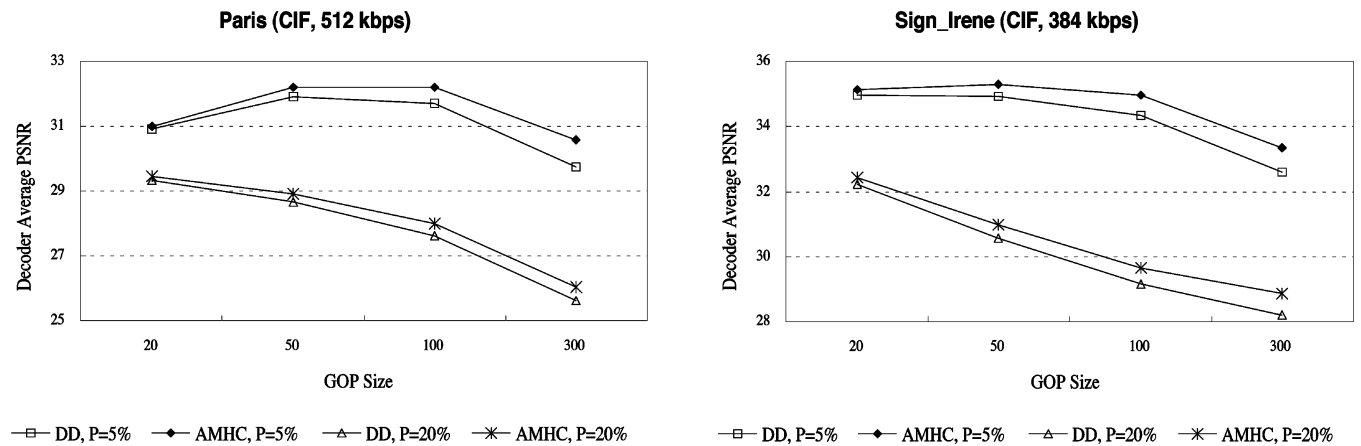


Fig. 9. Comparison of DD and AMHC ($\lambda = 0.2, N = 5$) for different GOP.

curs in description D2 at time $(n_0 + 19)$. When the frame is lost at $(n_0 + 40)$, temporal interpolation is applied to frame 39 and 41 which both have error propagated from $(n_0 + 19)$. Thus the PSNR of the error-concealed frame at $(n_0 + 40)$ is significantly lower than that at time n_0 . For DD, the PSNR drop is 2.12 dB which is very large. For AMHC, the PSNR drop is 1.05 dB, much smaller than DD because the propagated error of AMHC in D2 is much reduced at frame 39 and 41 compared with DD. In Fig. 6(b), parameter h_1 is chosen to be 0.6 for CMHC and three *time intervals* (N) are compared, $N \in \{1, 4, 7\}$. Once again AMHC achieves significantly better PSNR than DD, even for $N = 1$. When N goes from 1 to 4 and from 4 to 7, the PSNR of AMHC increases. In other words, it is good to use a larger N for AMHC, though the incremental gain becomes progressively smaller. For CMHC, $N = 4$ gives better performance than $N = 1, 7$. This is not surprising as similar behaviors can be observed in Figs. 4(a) and 5(a), although the simulation conditions are different.

B. Comparison Between DD and AMHC

As AMHC performs better than CMHC in suppressing the propagated errors, we will only compare DD with AMHC in this subsection. For both algorithms, the conditions with and without *random INTRA refresh* (RIR) are tested [22]. Four algorithms are compared.

- **DD**: Direct decoding. RIR is not used.
- **AMHC**: The AMHC algorithm without using RIR.

- **DD & IR = 3%**: The DD algorithm with RIR enabled. The percentage of forced INTRA-MBs for each P-frame, i.e., the INTRA-rate (IR), is 3%.
- **AMHC & IR = 3%**: The AMHC algorithm with RIR enabled, and the INTRA-rate is 3%.

Note that whether RIR is enabled or not, additional INTRA-MB can be encoded if it has a lower RD cost in the encoder mode-decision procedure.

Fig. 7 shows the RD curves of AMHC and DD under packet loss rate $P = 10\%$. From the figure we can see that, no matter whether RIR is used or not, the RD curve of AMHC is always higher than that of DD. Fig. 8 compares the performance of AMHC and DD under different packet loss rate. The bit rate is fixed. As before, the curve of AMHC is consistently higher than that of DD. The PSNR gain of AMHC over DD can be as high as 0.50 dB, 0.76 dB, 0.77 dB, 0.64 dB for $P = 3\%, 5\%, 10\%$, and 20% respectively.

In previous simulations, one sequence is encoded in one GOP. In Fig. 9, we compare the performance of AMHC and DD with different frequency of I-frames. Four GOP sizes (20, 50, 100, and 300) are compared with a fixed bit-rate. The simulated packet loss rate is $P = 5\%$ and $P = 20\%$. As before, the curve of AMHC is always higher than that of DD for different GOP size. However, the PSNR gain of AMHC over DD is smaller for smaller GOP size. This is because, with more I-frames, error propagation is less of a problem, and thus the advantage of AMHC over DD is smaller.

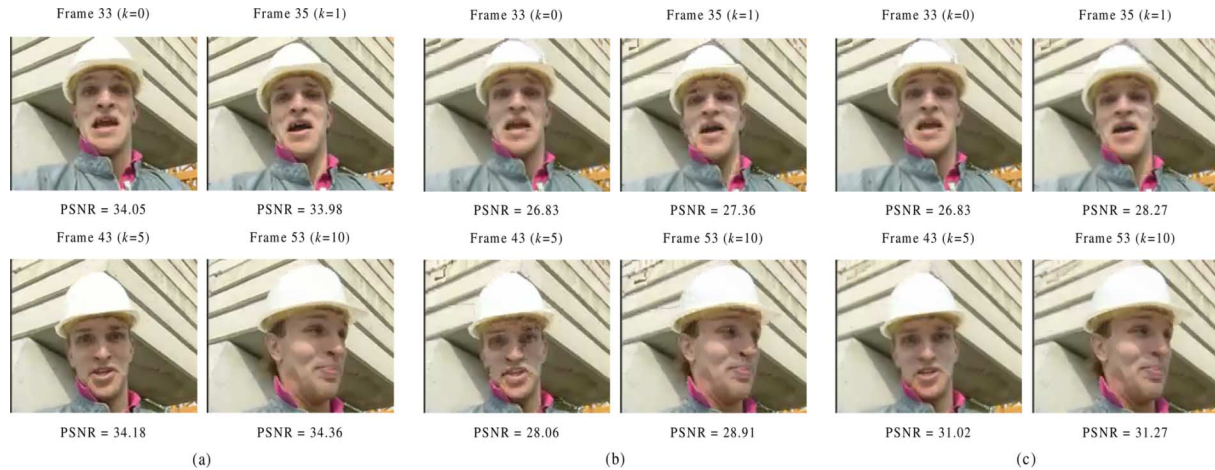


Fig. 10. Perceptual results of AMHC and DD on Foreman (QCIF, 15 fps, $QP = 30$). (a) Original error-free decoded frames; (b) reconstructed frames by DD; (c) reconstructed frames by AMHC with $\lambda = 0.2$, $N = 5$.

Fig. 10 illustrates the visual quality after applying DD and AMHC on *Foreman* for the case of a single frame loss (at frame 33). The four frames in each subfigure correspond to the lost frame ($k = 0$), the subsequent 1st, 5th and 10th frames in the same description, with the frame indices shown at the top. The lost frame concealed by UMCTI is the same for both AMHC and DD. From the figure, we can observe that both DD and AMHC can help to recover from the frame loss. And the proposed AMHC can recover much faster than DD.

IV. CONCLUSION

In this paper we propose a novel algorithm called *multihypothesis error concealment* (MHC) to improve the reconstructed video quality of any error concealment (EC) method for MDC by improving its error recovery rate. While existing EC methods apply concealment to the lost frames only, the proposed MHC applies temporal interpolation to some additional frames after the frame loss so as to reduce propagated error quickly. Simulation results show that MHC can effectively improve the error recovery rate of a traditional EC algorithm. In the current work, the weight of MHC is fixed for a whole frame. To further improve the reconstructed video quality, block or pixel level adaptation can be used to adjust the weight. We take this as a future work.

REFERENCES

- [1] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Review of error resilient coding techniques for real-time video communications," *IEEE Signal Process. Mag.*, vol. 17, pp. 61–82, Jul. 2000.
- [2] Y. Wang and Q. F. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, pp. 974–997, May 1998.
- [3] A. Nafaa, T. Taleb, and L. Murphy, "Forward error correction strategies for media streaming over wireless networks," *IEEE Commun. Mag.*, vol. 46, pp. 72–79, Jan. 2008.
- [4] C.-M. Fu, W.-L. Hwang, and C.-L. Huang, "Efficient post-compression error-resilient 3D-scalable video transmission for packet erasure channels," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP'05)*, Mar. 2005, pp. 305–308.
- [5] Y. Wang, A. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proc. IEEE*, vol. 93, pp. 57–70, Jan. 2005.
- [6] 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, pp. 713–718, Oct. 1998.
- [7] S.-C. Hsia, S.-C. Cheng, and S.-W. Chou, "Efficient adaptive error concealment technique for video decoding system," *IEEE Trans. Multimedia*, vol. 7, pp. 860–868, Oct. 2005.
- [8] Y. Chen, K. Yu, J. Li, and S. Li, "An error concealment algorithm for entire frame loss in video transmission," in *Picture Coding Symp.*, Dec. 2004.
- [9] S. Belfiore, M. Grangetto, E. Magli, and G. Olmo, "Concealment of whole-frame losses for wireless low bit-rate video based on multiframe optical flow estimation," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 316–329, Apr. 2005.
- [10] Z. Wu and J. M. Boyce, "An error concealment scheme for entire frame losses based on H.264/AVC," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'06)*, May 2006, pp. 1463–1466.
- [11] C.-K. Wong and O. Au, "Fast motion compensated temporal interpolation for video," in *Proc. SPIE Visual Commun. Image Processing (VCIP'95)*, May 1995, pp. 1108–1118.
- [12] C.-W. Tang and O. Au, "Unidirectional motion compensated temporal interpolation," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'97)*, June 1997, pp. 1444–1447.
- [13] G. Dane and T. Nguyen, "Optimal temporal interpolation filter for motion-compensated frame rate up conversion," *IEEE Trans. Image Process.*, vol. 15, pp. 978–991, Apr. 2006.
- [14] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Proc. SPIE Visual Communications and Image Processing (VCIP'01)*, Jan. 2001, pp. 392–409.
- [15] Y. Lu, R. Zhou, H. Cui, and K. Tang, "Bi-directional entire frame recovery in MDC video streaming," in *Proc. IEEE Int. Symp. Communications and Information Technology (ISCIT'05)*, Oct. 2005, pp. 1058–1061.
- [16] M. Ma, O. C. Au, S.-H. G. Chan, L. Guo, and Z. Liang, "Three-loop temporal interpolation for error concealment of MDC," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS'06)*, May 2006, pp. 694–697.
- [17] B. Girod and N. Farber, M.-T. Sun and A. R. Reibman, Eds., "Wireless video," in *Compressed Video Over Networks*. New York: Marcel Dekker, 2000.
- [18] N. Farber, K. Stuhlmüller, and B. Girod, "Analysis of error propagation in hybrid video coding with application to error resilience," in *Proc. IEEE Int. Conf. Image Processing (ICIP'99)*, Oct. 1999, pp. 550–554.
- [19] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises and Solutions*, 3rd ed. New York: Wiley, 1997.
- [20] "JVT reference software, version 8.2. [Online].," Available: [Online]. Available: <http://iphome.hhi.de/suehring/tml/download/>
- [21] S. Wenger, "Error patterns for internet experiments," in *ITU-T SG16 Doc. Q15-I-16rl*, Oct. 1999.
- [22] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error resiliency schemes in H.264/AVC standard," *Elsevier J. Vis. Commun. Image Represent.*, vol. 17, no. 2, pp. 425–450, Apr. 2006.



Mengyao Ma (S'05) received the B.Sc. degree in computer science and technology from Peking University, Beijing, China, in 2003. She is currently pursuing her Ph.D. with the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. Her research interests include error resilient video compression, error propagation analyses and error concealment of video streams over packet loss channels.



Oscar C. Au (S'87–M'90–SM'01) received the B.A.Sc. degree from the University of Toronto, Toronto, ON, Canada, in 1986, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, in 1988 and 1991, respectively.

After being a Postdoctoral Researcher at Princeton for one year, he joined the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology (HKUST), in 1992. He is now an Associate Professor, Director of Multimedia Technology Research Center (MTrec), and Advisor

of the Computer Engineering (CEPG) Program in HKUST. His main research contributions are on video and image coding and processing, watermarking and light weight encryption, speech and audio processing. Research topics include fast motion estimation for MPEG-1/2/4, H.261/3/4 and AVS, optimal and fast suboptimal rate control, mode decision, transcoding, denoising, deinterlacing, post-processing, multiview coding, scalable video coding, distributed video coding, subpixel rendering, JPEG/JPEG2000 and halftone image data hiding, etc. He has published about 200 technical journal and conference papers. His fast motion estimation algorithms were accepted into the ISO/IEC 14496-7 MPEG-4 international video coding standard and the China AVS-M standard. He has three U.S. patents and is applying for 40+ more on his signal processing techniques. He has performed forensic investigation and stood as an expert witness in the Hong Kong courts many times.

Dr. Au has been an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART 1 (TCAS1) and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT). He is the Chairman of the Technical Committee on Multimedia Systems and Applications (MSATC) and a member of the TC on Video Signal Processing and Communications (VSPC) and the TC on DSP of the IEEE Circuits and Systems (CAS) Society. He served on the Steering Committee of IEEE TRANSACTIONS ON MULTIMEDIA (TMM) and the IEEE International Conference on Multimedia and Expo (ICME). He also served on the organizing committee of the IEEE International Symposium on Circuits and Systems (ISCAS) in 1997, the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) in 2003, the ISO/IEC MPEG 71st Meeting in 2004, International Conference on Image Processing (ICIP) in 2010, and other conferences.



Liwei Guo (S'06) received the B.E degree in 2004 from the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. He is currently pursuing the Ph.D. degree in the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. His current research interests include image/video processing, video compression and signal estimation.



S.-H. Gary Chan (S'89–M'98–SM'03) received the B.S.E. degree (Highest Honor) in electrical engineering from Princeton University, Princeton, NJ, in 1993, with certificates in applied and computational mathematics, engineering physics, and engineering and management systems, and the M.S.E. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1994 and 1999, respectively, with a minor in business administration.

He is currently an Associate Professor with the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology, and an Adjunct Researcher with Microsoft Research Asia, Beijing. He was a Visiting Assistant Professor in Networking with the Department of Computer Science, University of California, Davis, from 1998 to 1999. During 1992–1993, he was a Research Intern at the NEC Research Institute, Princeton. His research interests include multimedia networking, peer-to-peer technologies and streaming, and wireless communication networks.

Dr. Chan was a William and Leila Fellow at Stanford University during 1993–1994. At Princeton University, he was the recipient of the Charles Ira Young Memorial Tablet and Medal, and the POEM Newport Award of Excellence in 1993. He served as a Vice-Chair of IEEE COMSOC Multimedia Communications Technical Committee from 2003 to 2006. He is a Guest Editor for the *IEEE Communication Magazine* (Special Issues on Peer-to-Peer Multimedia Streaming), 2007 and Springer *Multimedia Tools and Applications* (Special Issue on Advances in Consumer Communications and Networking), 2007. He was the Co-Chair for the workshop on Advances in Peer-to-Peer Multimedia Streaming for the ACM Multimedia Conference (2005), and the Multimedia Symposia for IEEE GLOBECOM (2006) and IEEE ICC (2005, 2007).



Peter H. W. Wong (M'01) received B. Eng. degree (with first class honour) in computer engineering from the City University of Hong Kong in 1996, and the M.Phil. and Ph.D. degrees in electrical and electronic engineering from Hong Kong University of Science and Technology (HKUST) in 1998 and 2003 respectively.

He was a Postdoctoral Fellow at the Department of Information Engineering, Chinese University of Hong Kong (CUHK) from 2003 to 2005. He worked as at the Applied Science and Technology Research Institute Company Limited (ASTRI) as a Member of Professional Staff from 2005 to 2007. He was the Visiting Assistant Professor at the Department of Electronic and Computer Engineering, HKUST from 2007 to 2008. He is currently the R&D Director of VP Dynamics Labs (Mobile) Ltd. His research interests include digital data hiding and watermarking, time scale modification, fast motion estimation, video/image de-noising, audio coding, audio enhancement, auto white balancing, high dynamic range image processing and subpixel rendering.

Dr. Wong served on the organizing committee of the ISO/IEC MPEG 71st Meeting in 2004, the International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS) in 2005 and Pacific-Rim Conference on Multimedia (PCM) in 2007. Dr. Wong received the Schmidt award of excellence in 1998. He is a member of Sigma Xi.