J. Vis. Commun. Image R. 19 (2008) 437–449

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Alternate motion-compensated prediction for error resilient video coding

Mengyao Ma [a,*], Oscar C. Au [b], Liwei Guo [b], S.-H. Gary Chan [a], Xiaopeng Fan [b], Ling Hou [b]

[a] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
[b] Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong

ARTICLE INFO

Article history:
Received 25 October 2007
Accepted 11 July 2008
Available online 23 July 2008

Keywords:
Error resilience
Error concealment
Two-hypothesis motion-compensated prediction
Alternate motion-compensated prediction
Error ratio
Error energy
Temporal interpolation

ABSTRACT

Since the quality of compressed video is vulnerable to errors, video transmission over unreliable Internet is very challenging today. Two-Hypothesis Motion-Compensated Prediction (THMCP) has been shown to have Error Resilience (ER) capability for video transmission, where each macroblock is predicted from its previous two frames. In this paper, we propose a novel ER approach named Alternate Motion-Compensated Prediction (AMCP). In addition to two-hypothesis prediction, one-hypothesis prediction is alternately used. We use some schemes to determine which kind of prediction should be used, so that in some cases of loss, the propagated error can be first decreased to some extent before it spreads to the subsequent frames. As a result, the expected converged error is less than that obtained from THMCP with fixed weights (THMCPF). Both analysis and simulation results are given to show that AMCP performs better than THMCPF, in terms of both compression efficiency and ER capability.

© 2008 Elsevier Inc. All rights reserved.

1. 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]. If data loss occurs during the transmission, the corresponding frame will be corrupted, and this error will propagate to the subsequent frames because of INTER-prediction, until the next INTRA-coded frame is correctly received. For example, a simple bit error in VLC can cause desynchronization; as a result, all the following bits cannot be used until a synchronization code arrives. Due to these facts, it is useful to develop some schemes to improve the Error Resilience (ER) capability of the compressed video.

Several error resilience methods have been developed for video communication. One such method is Forward Error Correction (FEC) coding, typically applied at channel coding stage. In this method, FEC codes are added to the video stream by the encoder, and the decoder uses these codes to correct some bit errors. FEC techniques can be jointly used with other error resilience methods such as data partitioning [3] and subband-based coders [4,5]. It can also be used to protect a region of interest using the new error resilience tools provided by H.264/AVC [6–8]. Another method for error resilience is Layered (Scalable) Coding (LC). LC refers to partitioning the video stream into more than one layer. The base layer is protected and transmitted with higher priority; it contains the most important information for the video and can be used to provide acceptable video quality. Each enhancement layer incrementally improves the video quality [5,9,10]. LC provides different video qualities according to channel bandwidth, but the layers have to be obtained incrementally, leading to inflexibility and low video quality when a lower layer is lost. Contrary to this approach, Multiple Description Coding (MDC) divides the video stream into equally important *streams* (*descriptions*). These descriptions are sent to the destination through different channels. If error occurs during the transmission, only a subset of the descriptions will be received by the decoder, which can be used to reconstruct the video with lower but acceptable quality [11–13]. In addition to MDC, Multi-Hypothesis Motion-Compensated Prediction (MHMCP) has also been proven to have error resilience capability, where each macroblock (MB) is predicted by a linear combination of multiple signals (hypotheses) [14,15]. It is shown in [16,17] that MHMCP can suppress the short-term error propagation more effectively than the intra-refreshing scheme. Our proposed ER method is motivated by this approach.

MHMCP is originally developed to improve the compression efficiency of video coding [18,19]. Its error resilience property is analyzed in [14], where a special case of Two-Hypothesis Motion-Compensated Prediction (THMCP) is used. In this approach, each frame (except INTRA-frame and the first INTER-frame) is predicted from a weighted average of its previous two frames, and the weight is fixed for each prediction (hypothesis). The error propagation model at the decoder side is analyzed, which is combined with

* Corresponding author. Fax: +852 23350194.
E-mail addresses: myma@ust.hk, myma@cse.ust.hk (M. Ma), eeau@ust.hk (O.C. Au), eeglw@ust.hk (L. Guo), gchan@ust.hk (S.-H. Gary Chan), eexp@ust.hk (X. Fan), eileenzb@ust.hk (L. Hou).

1047-3203/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2008.07.003

the encoder predictor to strike a balance between compression efficiency and error resilience capability. We name this special case of two-hypothesis approach THMCPF (THMCP with fixed weights). In [15], the authors extend THMCPF by utilizing the concept of reference picture interleaving and data partitioning. The parameters for each hypothesis, including the motion vectors and the reference frame indices, are separated into the bitstream to reduce the impact of a single frame loss. In [16,17], the error propagation effect in MHMCP coder is investigated and the rate-distortion performance in terms of the hypothesis number and hypothesis coefficients is analyzed. It is shown that a hypothesis number no larger than three is suitable at low bit rates.

If one frame is corrupted during the transmission, error will propagate to all the subsequent frames. Without considering the effect of spatial filtering caused by sub-pixel motion compensation, this error will converge at last to some percentage (determined by the prediction weight) of the initial error [20]. We define error ratio to be the ratio of converged value to the first error. One advantage of using MHMCP is that the error ratio is smaller than one, which means the propagated error can be decreased. Motivated by this, we propose a new ER approach named Alternate Motion-Compensated Prediction (AMCP) in [21]. The novelty is that in addition to two-hypothesis prediction, one-hypothesis prediction is alternately used. We use some schemes to determine whether two-hypothesis or one-hypothesis should be used, such that in some cases of loss, the propagated error can be first decreased to some extent before it spreads to the subsequent frames. As a result, the expected error ratio at the decoder is less than that obtained from two-hypothesis MCP. In this paper, we will give more details to AMCP, and the effect of spatial filtering is considered in the analysis of error propagation [20,22]. The appropriate error concealment methods are also discussed. We propose to use temporal interpolation since it can be well combined with temporal subsampling ER methods, such as multiple description coding (MDC) and AMCP [23].

The remainder of this paper is organized as follows: In Section 2, the proposed approach AMCP is introduced. We will analyze the error propagation in the case of a single frame loss, and derive the closed-form expression for error energy. The expected error ratio is also obtained. Then AMCP is extended to layered coding, which can be applied in the multicast of video over the Internet. The error concealment approach for AMCP is discussed in Section 3, to further enhance the reconstructed video quality when error occurs. In Section 4, simulation results are given to compare AMCP and THMCP with fixed weights. A comparison between AMCP and the Loss-Aware Rate-Distortion Optimized MB mode decision algorithm (LARDO) used in H.264/AVC is also presented [24,25]. Section 5 gives a conclusion.

2. Alternate motion-compensated prediction (AMCP)

In this section we will introduce our proposed error resilience method alternate motion-compensated prediction (AMCP), where each frame can be predicted using one or two hypotheses, with an alternate pattern. We will analyze the error propagation in the case of a single frame loss, and derive the closed-form expression for error energy. If the effect of spatial filtering caused by sub-pixel motion compensation is not considered, the propagated error will converge to some percentage (determined by the prediction weight) of the initial error [20]. Define error ratio to be the ratio of converged value to the first error. We prove that the expected error ratio of AMCP is smaller than that of THMCP with fixed weights (THMCPF), which means AMCP can make the propagated error smaller. At last, we extend AMCP to layered coding, as a way for the multicast of real-time video over the Internet.

2.1. Introduction to the proposed AMCP

Suppose the video at time n is $\psi(n)$, which is an $H \times W$ matrix. In THMCPF [14], each frame has two hypotheses and frame $\psi(n)$ is predicted by

$$\hat{\psi}(n) = h_1\dot{\psi}(n-1) + h_2\dot{\psi}(n-2), \tag{1}$$

where $n \geqslant 2$ and $h_1 + h_2 = 1$. $\dot{\psi}(n-i)$ is a motion-compensated prediction from the ith previous reconstructed frame and h_i is a constant weighting parameter, $i = 1, 2$. Note that if $h_2 = 0$, this becomes a conventional predictor. And if $h_2 = 1$, this is the same as the odd/even sub-sampling method used in temporal MDC [23].

Consider the case of a single frame loss during the transmission. Suppose the frame loss happens at time l_0 and the lost frame is $\psi(l_0)$. Define error $\zeta(k)$ to be the difference between the reconstructed $(l_0 + k)$th frame at the decoder and that at the encoder. $\zeta(k)$ is an $H \times W$ matrix and we define $\zeta(k, x, y)$ to be the matrix element at the yth row and xth column, $x \in [0, W-1]$ and $y \in [0, H-1]$. For the special case of all the motion vectors (MVs) being zero, motion compensation is just a copying process. Using the predictor in Eq. (1), the error at time k becomes

$$\begin{cases} \zeta(1, x, y) = h_1\zeta(0, x, y), \\ \zeta(k, x, y) = h_1\zeta(k-1, x, y) + h_2\zeta(k-2, x, y), \quad k \in [2, \infty), \end{cases} \tag{2}$$

for $x \in [0, W-1]$ and $y \in [0, H-1]$ [14]. From Eq. (2), the error propagation model can be obtained as

$$\zeta(k, x, y) = \frac{1 - (-h_2)^{k+1}}{1 + h_2}\zeta(0, x, y). \tag{3}$$

When k goes to infinity and $h_1 \in (0, 1)$, $\zeta(k, x, y)$ will decrease and converge to $\frac{1}{1+h_2}\zeta(0, x, y)$. We define error ratio (R) to be the ratio of converged value to the first error. So the error ratio of predictor in Eq. (1), R_1, for a single frame loss is

$$R_1 = \lim_{k\to\infty}\frac{\zeta(k)}{\zeta(0)} = \lim_{k\to\infty}\frac{\zeta(k, x, y)}{\zeta(0, x, y)} = \frac{1}{1 + h_2}, \tag{4}$$

for any $x \in [0, W-1]$ and $y \in [0, H-1]$. Note that in this paper, error ratio is considered for the special case of all the motion vectors (MVs) being zero. One advantage of using THMCPF is that the error ratio is smaller than one, i.e. $\frac{1}{2} < R_1 < 1$ for $h_2 \in (0, 1)$, which means it can reduce the propagated error. On the other hand, error will propagate to all the subsequent frames, as in the conventional codec.

Motivated by this, we propose a new error resilience approach named alternate motion-compensated prediction (AMCP) in this paper, as shown in Fig. 1b. The basic idea comes from the error propagation property of the approach in Fig. 1a: each even frame is predicted from its previous two frames using (1) and each odd frame is predicted from its previous odd frame. Suppose a single frame is lost. The good thing of this predictor is that if the lost frame is an even frame (with $\times$), only the subsequent even frames are affected and the propagated error will decrease and converge to zero quickly. If the lost frame is an odd frame (with Δ), error will propagate to all the subsequent frames. In this case, although error will converge at last, the converged value is the same as the first one at the lost frame.

AMCP shown in Fig. 1b is an improvement of the approach in Fig. 1a. The video sequence is divided into periodic Intervals $\{I_0, I_1, \ldots\}$, which start after the INTRA-frame. Suppose only the first frame is encoded as INTRA-frame. The frame index within an Interval (Interval Index) goes from 0 to $(2N+1)$, and the $(2N+1)$th frame of an Interval is the 0th frame of the next Interval. In other words, two consecutive Intervals overlap at one frame, as illustrated in Fig. 1b. Here N is a positive integer and $(2N+1)$ is the length of an Interval. Within an Interval, each of the odd frames is

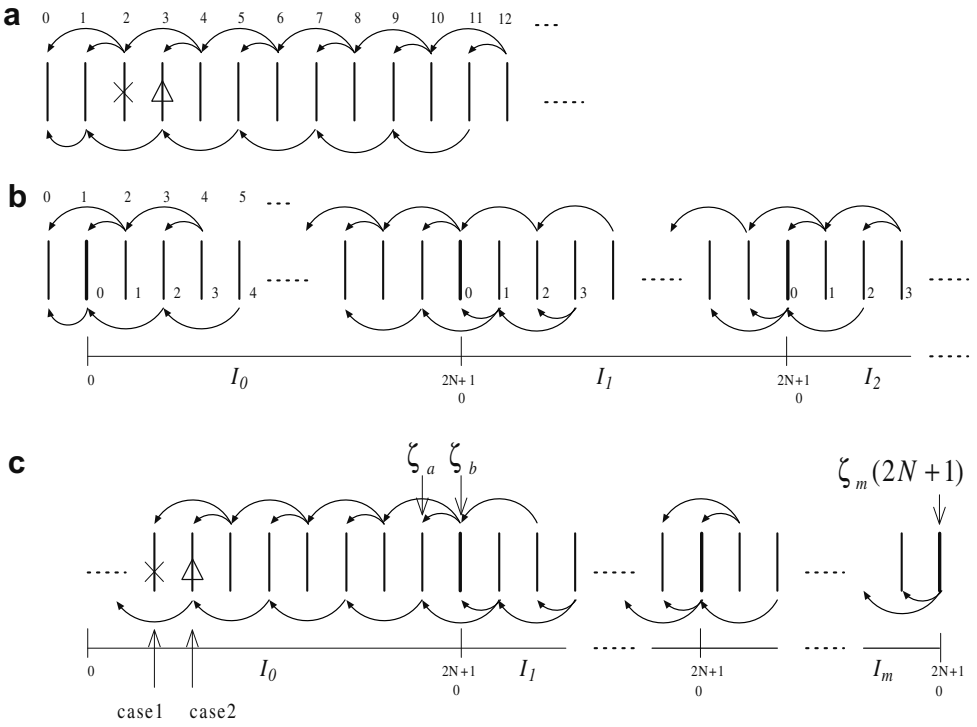


Fig. 1. Illustration of AMCP and its error propagation (the arrow means that the previous frame is used as the reference for the latter frame). (a) A basic approach, (b) proposed AMCP, (c) error propagation analysis of AMCP.

encoded by Two-Hypothesis Prediction (2HP), i.e. predicted from its previous two frames using Eq. (1). Each of the even frames is encoded by One-Hypothesis Prediction (1HP), i.e. predicted from its previous even frame. Here the odd or even frame is defined by its Interval Index, instead of the index in the video sequence (Time Index). One exception is the 0th frame of each Interval. It is the $(2N+1)$th frame of the previous Interval and thus is predicted from its previous two frames. In addition, the 0th frame of I_0 is predicted from the INTRA-frame. A special case of AMCP is $N=0$, which makes the predictor the same as THMCPF. And in addition to THMCPF, AMCP can also be considered as a special case of THMCP by setting one of the weighting parameters of THMCP to zero.

2.2. Error propagation model of AMCP

Consider the case of a single frame loss ($\psi(l_0)$) in AMCP. Without loss of generality, suppose $\psi(l_0)$ belongs to I_0 and the first error (at time l_0) is $\zeta(0)$. We want to obtain the error propagation model $\zeta(k)$. In the previous analyses, i.e. for Eqs. (2)–(4), all the motion vectors (MVs) were assumed to be zero, which may not be true in real situations. In addition, the effect of spatial filtering was not considered, which can be introduced by deblocking filters, or as a side effect of sub-pixel motion compensation with linear interpolation [20,22]. It is known that spatial filtering can attenuate the propagated error energy. In [22], this effect is analyzed and approximated by a separable average loop filter. In this work, we use a similar approximation. Assume the loop filter to be f, which is time invariant. Then the propagated error can be recursively calculated as

$$\begin{cases} \zeta(k,x,y) = \zeta(k-2,x,y)^* f(x,y), & \text{for 1HP,} \\ \zeta(k,x,y) = h_1\zeta(k-1,x,y)^* f(x,y) & \\ \qquad\qquad +h_2\zeta(k-2,x,y)^* f(x,y), & \text{for 2HP,} \end{cases} \tag{5}$$

After applying Fourier transform to these two equations, we have

$$\begin{cases} Z(k,u,v) = Z(k-2,u,v)F(u,v), & \text{for 1HP,} \\ Z(k,u,v) = h_1 Z(k-1,u,v)F(u,v) & \\ \qquad\qquad +h_2 Z(k-2,u,v)F(u,v), & \text{for 2HP,} \end{cases} \tag{6}$$

where $F(u,v)$ and $Z(k,u,v)$ are the Fourier Coefficients of f and $\zeta(k)$, respectively, $u \in [0, W-1]$ and $v \in [0, H-1]$.

Similar to the simple approach in Fig. 1a, the loss of a frame with an odd Interval Index (case 1) or with an even Interval Index (case 2) in AMCP will lead to different error propagations, as illustrated in Fig. 1c. We mark these two cases with $\times$ and Δ, respectively. Suppose $\zeta_\times(k)$ is the error at the (l_0+k)th frame for case 1 and $\zeta_\Delta(k)$ is the error at the (l_0+k)th frame for case 2. Their frequency domain representations are $Z_\times(k,u,v)$ and $Z_\Delta(k,u,v)$, respectively, $u \in [0, W-1]$ and $v \in [0, H-1]$. Based on Eq. (6), we can get the error propagation in I_0 for the two cases:

Case 1. A frame with an odd Interval Index is lost and the error propagation within I_0 is

$$\begin{cases} Z_\times(2n,u,v) = h_2^n Z(0,u,v)F^n(u,v), & n \geqslant 0, \\ Z_\times(2n+1,u,v) = 0, & n \geqslant 0. \end{cases} \tag{7}$$

Case 2. A frame with an even Interval Index is lost and the error propagation within I_0 is

$$\begin{cases} Z_\Delta(2n,u,v) = Z(0,u,v)F^n(u,v), & n \geqslant 0, \\ Z_\Delta(2n+1,u,v) = (1-h_2^{n+1})Z(0,u,v)F^{n+1}(u,v), & n \geqslant 0. \end{cases} \tag{8}$$

As we can see from Eq. (7) that case 1 has an advantage in suppressing the propagated error with $h_2 < 1$. Since usually F acts as a low pass filter and $F(u,v) \leqslant 1$, $Z_\times(2n,u,v)$ exponentially decreases with increasing n. Even if $F(u,v) > 1$, $Z_\times(2n,u,v)$ can be a decreasing function of n as long as $h_2F(u,v) < 1$. The case of $h_2F(u,v) \geqslant 1$

happens rarely. So we can expect that the energy of error $Z_{\times}(2n)$ decreases rapidly in Interval I_0. Although error $Z_{\Delta}(2n+1,u,v)$ may increase in case 2, it can be compensated by the great benefits from case 1 on average. We will give more discussions about this in the next subsection.

To analyze the error propagation in the subsequent Intervals, we define $\zeta_m(r)$ to be the error in the rth frame of Interval m, with frequency domain representation $Z_m(r,u,v)$, $r \in [0,2N+1]$. In addition, we have $\zeta_{m+1}(0) = \zeta_m(2N+1)$ and $Z_{m+1}(0) = Z_m(2N+1)$. Using similar deriving process as Eqs. (7) and (8), we can obtain the error propagation in Interval m ($m \geqslant 1$):

$$\begin{cases} Z_m(2n,u,v) = Z_{m-1}(2N+1,u,v)F^n(u,v), \\ Z_m(2n+1,u,v) = ((1-h_2^{n+1})Z_{m-1}(2N+1,u,v) \\ \qquad\qquad +h_2^{n+1}Z_{m-1}(2N,u,v))F^{n+1}(u,v) \end{cases} \tag{9}$$

for $n \in [0,N]$, $u \in [0,W-1]$ and $v \in [0,H-1]$. This is true for both case 1 and case 2.

Define the errors of the last two frames in I_0 to be $\zeta_a = \zeta_0(2N)$ and $\zeta_b = \zeta_0(2N+1)$, whose frequency domain representations are $Z_a(u,v)$ and $Z_b(u,v)$, respectively. By some manipulations of Eq. (9) we can get

$$\begin{cases} Z_m(2N,u,v) = \frac{(\omega+\mu)^{m-1}(2\kappa-\omega+\mu)-(\omega-\mu)^{m-1}(2\kappa-\omega-\mu)}{2^m\mu}Z_b(u,v)F^N(u,v), \\ Z_m(2N+1,u,v) = \frac{(\omega+\mu)^{m}(2\kappa-\omega+\mu)-(\omega-\mu)^{m}(2\kappa-\omega-\mu)}{2^{m+1}\mu}Z_b(u,v) \end{cases} \tag{10}$$

for $m \geqslant 1$, where $\omega = (1-\alpha)F^{N+1}(u,v)$, $\alpha = h_2^{N+1}$, $\kappa = \frac{(1-\alpha)Z_b(u,v)+\alpha Z_a(u,v)}{Z_b(u,v)}$ $F^{N+1}(u,v)$ and $\mu = \sqrt{\omega^2+4\alpha F^{2N+1}(u,v)}$.

Then the energy of error $\zeta_m(r)$ for $m \geqslant 1$ can be approximated as

$$\mathbf{E}_m(r) = \sum_{v=0}^{H-1}\sum_{u=0}^{W-1}|Z_m(r,u,v)|^2. \tag{11}$$

And $Z_m(r,u,v)$ can be obtained using Eqs. (9) and (10). The energy of the error $\zeta_{\times}(k)$ ($\zeta_{\Delta}(k)$) in the first Interval I_0 can be obtained similarly.

As we have discussed previously that if the frame loss is case 1, error can decrease much quicker than those of case 2, which lead to lower propagated error energy in the subsequent Intervals. In addition, we can get the following results with $F(u,v) = 1$ and fixed u, v, m ($m \geqslant 1$):

- If $Z_{m-1}(2N,u,v) < Z_{m-1}(2N+1,u,v)$ Within Interval m, $Z_m(2n,u,v)$ remains unchanged and $Z_m(2n+1,u,v)$ increases with increasing n. For the last two frames, we have $Z_m(2N,u,v) > Z_m(2N+1,u,v)$. In detail, the error propagations in Interval m are enclosed by two envelopes. The upper one is a horizontal line, and the bottom one increases with time.
- If $Z_{m-1}(2N,u,v) > Z_{m-1}(2N+1,u,v)$ Within Interval m, $Z_m(2n,u,v)$ remains unchanged and $Z_m(2n+1,u,v)$ decreases with increasing n. For the last two frames, we have $Z_m(2N,u,v) < Z_m(2N+1,u,v)$. In detail, the error propagations in Interval m are enclosed by two envelopes. The bottom one is a horizontal line, and the upper one decreases with time.
- If $Z_{m-1}(2N,u,v) = Z_{m-1}(2N+1,u,v)$ Based on Eq. (9), we have $Z_m(2n,u,v) = Z_m(2n+1,u,v)$ for $n \in [0,N]$. This is also true for the subsequent Intervals. So error will remain unchanged after Interval $m-1$.

We define the difference between the errors in the last two frames of Interval m to be D_m, i.e.

$$D_m(u,v) = |Z_m(2N+1,u,v) - Z_m(2N,u,v)|, \tag{12}$$

for $u \in [0,W-1]$ and $v \in [0,H-1]$. If $Z_{m-1}(2N,u,v) \neq Z_{m-1}(2N+1,u,v)$, from Eq. (9) we can obtain the relation between $D_m(u,v)$ and $D_{m-1}(u,v)$:

$$\frac{D_m(u,v)}{D_{m-1}(u,v)} = \frac{|Z_m(2N+1,u,v) - Z_m(2N,u,v)|}{|Z_{m-1}(2N+1,u,v) - Z_{m-1}(2N,u,v)|} = \alpha, \tag{13}$$

with $\alpha = h_2^{N+1}$ and $F(u,v) = 1$. For $h_2 \in (0,1)$, we have $D_m(u,v) < D_{m-1}(u,v)$. In summary, error propagates with some oscillations at the beginning; within an Interval, the oscillation decreases with time; from an Interval to the next Interval, the minimum oscillation difference, i.e. D_m, decreases with factor $\alpha = h_2^{N+1}$; in the long run, the oscillation becomes neglectable and error remains unchanged.

Fig. 2 illustrates how error propagates in case 1 and case 2 for a single frame loss. The Interval parameter N is 5. Since with a larger h_2, error can converge slower and thus have a more obvious propagation pattern, we use $h_2 = 0.875$ in this figure. Frames in I_0 with Interval Index 3 (case 1) and 4 (case 2) are selected to be lost, one at a time. The x-axis is the frame number after the lost frame, and the y-axis is the DC of the error at time k, i.e. $Z(k,0,0)$. The DC of the initial error, $Z(0,0,0)$, is normalized to be 1. We set $F(0,0) = 1$ for a clearer illustration. From the figure we can see that in case 1, the error can decrease quickly in the first Interval before it propagates to the subsequent frames. This can make the overall error energy much smaller than that of case 2. The characteristics of error propagation in Interval m, as listed above, can also be verified in this figure.

Fig. 3 is the verification of the error energy model in Eq. (11). We use the JVT reference software version 11.0 for the simulation [25]. Foreman sequence (QCIF, 300 frames) are encoded at 30 fps and only the first frame is INTRA-frame. B frame is used to implement the frame with two references. Fixed QP (QP = 30) is used for all the frames. In order to analyze error propagation, INTRA-MB is not used in P and B frames. We approximate $F(u,v)$ by a filter with a Gaussian shape, i.e.

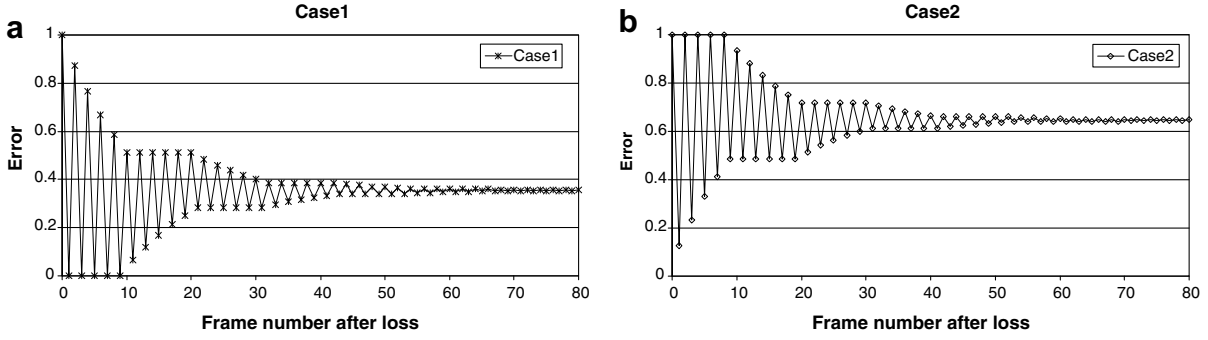


Fig. 2. Comparison between case 1 and case 2 for a single frame loss, $N = 5$, $h_2 = 0.875$ and $F(0,0) = 1$. The x-axis is the frame number after the lost frame, and the y-axis is the DC of the error at time k, i.e. $Z(k,0,0)$. The initial error $Z(0,0,0)$ is normalized to be 1.

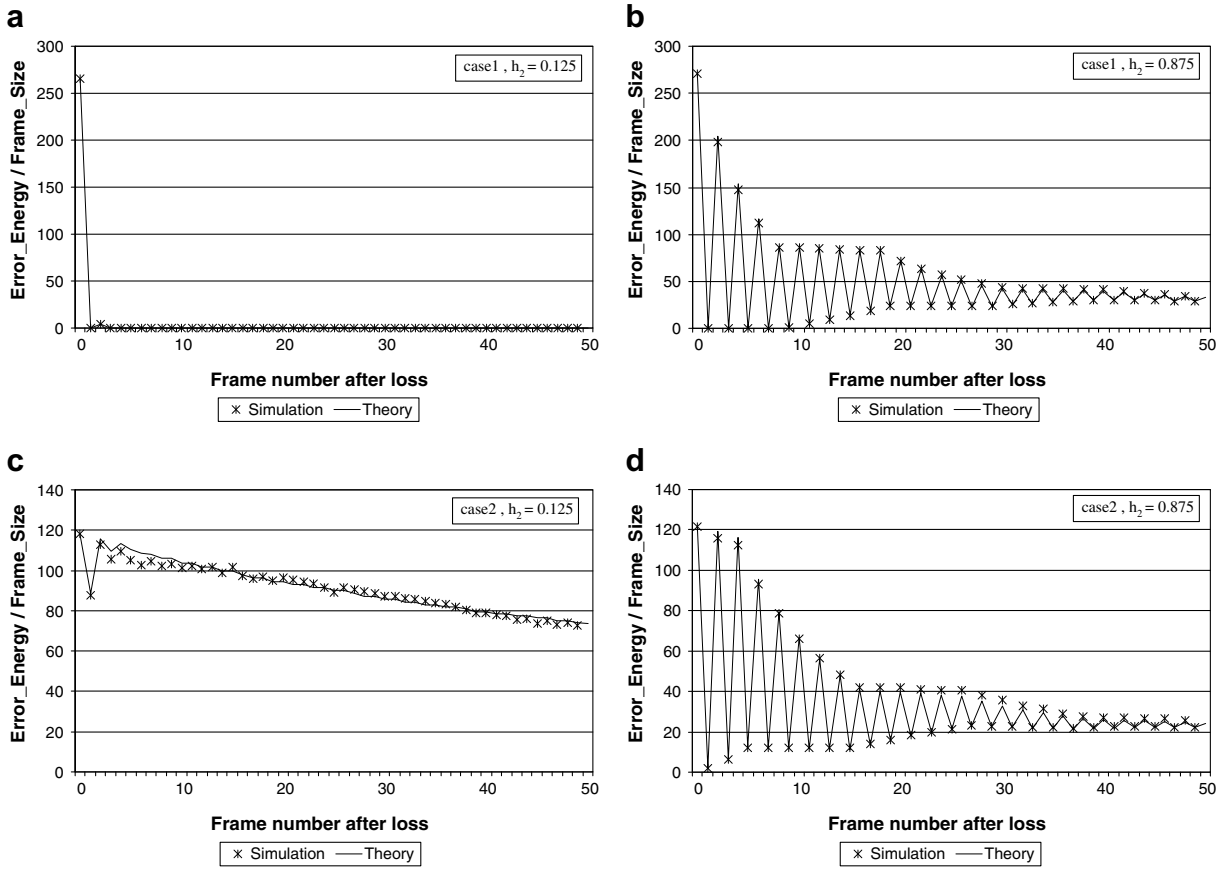


Fig. 3. Verification of the error energy model in Eq. (11) (Foreman QCIF, $N = 5$). (a and b) for case 1 and (c and d) for case 2. $h_2 = 0.125$ in (a) and (c) and $h_2 = 0.875$ in (b) and (d). The loop filter $F(u, v)$ is approximated to have a Gaussian shape, with $\sigma_F = 260$ in Eq. (14) for both $h_2 = 0.125$ and $h_2 = 0.875$. The y-axis is the error energy divided by the frame size (176×144 for QCIF), and the x-axis is the frame number after the lost frame.

$$F(u,v) = \mathrm{e}^{-(u^2+v^2)/(2\sigma_F^2)}. \tag{14}$$

The standard deviation σ_F describes the efficiency of the filter to attenuate the error energy. The Interval parameter is $N = 5$ and the weighting parameter is $h_2 = 0.125$ or $h_2 = 0.875$. We randomly select one frame to be lost for case 1 (Fig. 3a and b) and another one for case 2 (Fig. 3c and d). The error energy of the kth frame after loss is calculated and plotted, using

$$\mathbf{E}_T(k) = \sum_{y=0}^{H-1}\sum_{x=0}^{W-1}(\bar{\psi}(l_0+k,x,y) - \tilde{\psi}(l_0+k,x,y))^2, \tag{15}$$

where $\tilde{\psi}(l_0+k,x,y)$ is the pixel value of the encoder reconstructed frame, and $\bar{\psi}(l_0+k,x,y)$ is its reconstructed value at the decoder side. The energy calculated by the theory model in Eq. (11) is also plotted, where the initial error is assigned to be the decoder error at the loss position, i.e. $\zeta(0) = \bar{\psi}(l_0) - \tilde{\psi}(l_0)$. σ_F in Eq. (14) is trained to be 260 for both $h_2 = 0.125$ and $h_2 = 0.875$. We can see that the theory model can approximate the decoder error energy very closely. Similar as the THMCPF method, a larger h_2 leads to more severe oscillation of error propagation. For a smaller h_2, different case of loss generates different error propagation in AMCP; while in THMCPF, the smaller h_2 is, the larger the average error energy is. We will give more details about the comparison between THMCPF and AMCP in Section 4.

2.3. Expected error ratio of AMCP

From the analysis of the previous subsection, we can see that the values of ζ_a and ζ_b are determined by the position of the first error, using Eq. (7) for case 1 or Eq. (8) for case 2. Consider the condition of a single frame loss in I_0. Its Interval Index can be $i = 0, 1, \ldots, 2N$, each with equal probability. $(2N+1)$ is not included since it can be counted as the first one in the next Interval. Suppose all the motion vectors (MVs) are zero and there is no spatial filtering, i.e. $F(u,v) = 1$ for $u \in [0, W-1]$ and $v \in [0, H-1]$. Then using similar analyses as in Sections 2.1 and 2.2, we have

$$\zeta_m(2N+1,x,y) = \frac{\alpha + (-\alpha)^{m+1}}{1+\alpha}\zeta_a(x,y) + \frac{1-(-\alpha)^{m+1}}{1+\alpha}\zeta_b(x,y), \tag{16}$$

where $\alpha = h_2^{N+1}$. $\zeta_m(2N+1,x,y)$, $\zeta_a(x,y)$ and $\zeta_b(x,y)$ are the matrix elements of $\zeta_m(2N+1)$, ζ_a and ζ_b, respectively, $x \in [0, W-1]$ and $y \in [0, H-1]$. The expected value of $\zeta_a(x,y)$ and $\zeta_b(x,y)$ can be calculated as

$$E[\zeta_a(x,y)] = \sum_{n=0}^{N-1}\frac{1}{2N+1}\zeta_\times(2n+1,x,y) + \sum_{n=0}^{N}\frac{1}{2N+1}\zeta_\Delta(2n,x,y) = \frac{N+1}{2N+1}\zeta(0,x,y), \tag{17}$$

$$E[\zeta_b(x,y)] = \sum_{n=1}^{N}\frac{1}{2N+1}\zeta_\times(2n,x,y) + \sum_{n=0}^{N}\frac{1}{2N+1}\zeta_\Delta(2n+1,x,y) = \frac{N-\alpha+1}{2N+1}\zeta(0,x,y). \tag{18}$$

Combining Eqs. (17), (18) with (16), we can obtain the expected converged error of AMCP:

$$\lim_{m\to\infty} E[\zeta_m(2N,x,y)] = \lim_{m\to\infty} E[\zeta_m(2N+1,x,y)] = \frac{N+N\alpha+1}{(1+\alpha)(2N+1)}\zeta(0,x,y), \quad (19)$$

for $x \in [0, W-1]$ and $y \in [0, H-1]$. So the expected error ratio is

$$R_2 = \frac{N+N\alpha+1}{(1+\alpha)(2N+1)}, \quad (20)$$

where $\alpha = h_2^{N+1}$, $h_2 \in (0,1)$.

Define $\Delta(N,h_2) = R_2 - R_1$. When N is fixed and h_2 is continuous in range $(0,1)$, it is easy to verify that $\frac{\partial \Delta}{\partial h_2} > 0$. So Δ is a monotone increasing function of h_2 when $h_2 \in (0,1)$. Its range is $(-\frac{N}{2N+1}, 0)$. Based on the previous analysis, we can make the following conclusions with $F(u,v) = 1$:

- For fixed N, when $0 < h_2 < 1$, we have $\Delta < 0$, and therefore $R_2 < R_1$. In other words, the expected converged error using AMCP is smaller than that using THMCPF.
- When h_2 approaches 0, AMCP performs much better than THMCPF. On the other hand, a small value of h_2 makes both R_1 and R_2 large. If $h_2 = 0$, THMCPF becomes a conventional codec and $\zeta(0)$ will propagate to all the subsequent frames. Similar result for case 2 of AMCP. If the loss is case 1, only one frame is corrupted and the rest stream remains correct!
- When h_2 approaches 1, AMCP and THMCPF perform similar, both with decreasing error ratio. A special case is $h_2 = 1$, both methods converge to the odd/even sub-sampling method in temporal MDC, thus the propagated error cannot converge [23].

2.4. Extending AMCP to layered coding

In the environment of multicast of real-time video to the clients over the Internet, layered coding is widely used to adapt to the heterogeneous channel bandwidth as it provides the way to incrementally enhance the video quality. In layered coding, the base layer is transmitted to low bandwidth users, and high bandwidth users can further enhance the video quality with enhancement layers.

Due to the alternate prediction pattern, i.e. two-hypothesis or one-hypothesis prediction, AMCP can be easily extended to layered coding, as illustrated in Fig. 4. The video sequence is divided into two layers: the base layer (BL) has the frames indicated by the dot line, and the enhancement layer (EL) has the frames with solid line. Each frame is encoded using the same prediction as Fig. 1b. For the frames with Interval Index 0, we encode them twice: one is predicted from its previous frame and sent to BL; another is predicted from its previous two frames and set to EL. Note that by such prediction schemes, the generated EL stream itself is not H.264/AVC standard compatible, while the BL stream is still H.264/AVC compatible. As BL can be decoded independently from EL, it can be used for the users with low bandwidth. For the users with high bandwidth, they can take the packets from both BL and EL, and combine them to reconstruct the video with high quality. To avoid mismatch between the decoder and encoder, only the information from EL should be used to decode the frames with Interval Index 0.

3. Error concealment for AMCP

In the previous section, we analyze the error propagation property of AMCP. Note that there are two ways to decrease the converged error: one is to reduce the error ratio, and the other one is to decrease the error at the loss position ($\zeta(0)$). In this section, we will focus on the latter one and discuss the appropriate Error Concealment (EC) method for AMCP, i.e. temporal interpolation. Temporal interpolation was originally used to generate one or more frames between two received frames so as to improve the effective frame rate, and make the object motions in the video smoother. Usually both forward and backward motion estimations are performed to track motions of the objects between adjacent received frames [26]. This leads to high computational complexity. In [27], Unidirectional Motion-Compensated Temporal Interpolation (UMCTI) is used, which performs only forward motion estimation. UMCTI can be well combined with temporal sub-sampling ER methods, such as multiple description coding (MDC) and AMCP, since the motion vector preserved in the neighboring frames can be used for the interpolation, thus eliminating the exhaustive motion estimation [23].

Our error concealment algorithm for AMCP is based on UMCTI, named UMCTI–AMCP. In the following two subsections, we will first introduce our EC algorithm for INTER-frame losses, and then present how the INTER-frames are reconstructed in the special case of INTRA-frame loss.

3.1. EC for the lost INTER frames

As illustrated previously in Fig. 1c, in the case of a single frame loss (the nth frame), there are two kinds of conditions, leading to two different ways for the concealment:

Case 1. The two neighbors of the lost frame are correctly received, so the original UMCTI method can be applied, using the motion vector preserved in the $(n+1)$th frame. For an INTRA block in the $(n+1)$th frame, its motion vector is estimated from the neighboring blocks using median filter.

Case 2. Although the $(n+1)$th frame is received, it can not be correctly decoded due to the wrong reference frame. The solution is to reconstruct the lost frame first, i.e. copying the previous one, and use it to decode the $(n+1)$th frame. Then UMCTI can be used to reconstruct the lost frame by the preserved motion vector from the $(n+1)$th frame to the $(n-1)$th frame. After the interpolation, we can reconstruct the $(n+1)$th frame using the new reference. To further improve the reconstructed video quality, one may propose to repeat the reconstruction of the lost frame and the $(n+1)$th frame, until no much improvement can be achieved. However, experimental results show

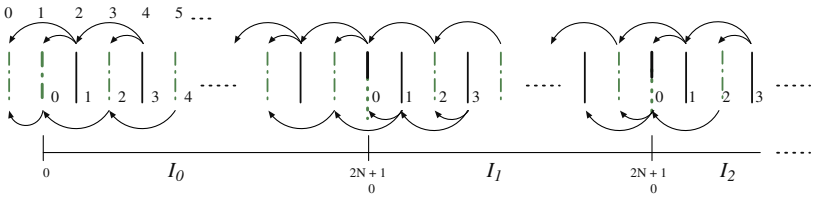


Fig. 4. Extending AMCP to layered coding.

that decoding the lost frame twice gives very little improvements, yet costs double computation time. One-time interpolation is enough to achieve acceptable quality.

The previous algorithm only works for the condition of a single frame loss. For continuous losses, as far as we know, there is no good error concealment method in the literature. In such cases, copying previous frame (Freeze) can be used to reconstruct the video.

3.2. INTER-frame reconstruction in the special case of INTRA-frame loss

In the special case of INTRA-frame loss, the reconstructed video quality is very bad as all the subsequent frames are encoded depending the INTRA-frame. In order to reduce the propagated error, Random INTRA-MB Refresh (RIR) can been jointly used with AMCP at the encoder side, where INTRA-coded MBs are randomly inserted into the bitstream to remove artifacts caused by error and INTER-prediction drift. When the INTRA-frame is lost, these received INTRA-MBs can be used to improve the quality of the subsequent frames.

In detail, each pixel of the lost INTRA-frame is filled by 128, and each 4×4 block of a later received frame (the mth frame) is reconstructed according to its block type:

- *INTRA block*
  The block is decoded as in the conventional decoder.
- *INTER block predicted by one-hypothesis*
  Suppose the motion vector of this block is mv_0, pointing to the $(m-2)$th frame. We can approximate its motion vector to the $(m-1)$th frame by $\frac{1}{2}mv_0$, based on the assumption of linear translation. Then for each pixel p, we can find its reference pixel p_1 in the $(m-1)$th frame with $\frac{1}{2}mv_0$, and reference pixel p_2 in the $(m-2)$th frame with mv_0, as shown in Fig. 5a. If p_1 is within an INTRA-MB but p_2 is not, we set $p = p_1$; otherwise, p is decoded as in the conventional decoder. Fig. 5b gives the illustration for the EC of pixel p.
- *INTER block predicted by two hypotheses*
  Suppose the two motion vectors of this block are mv_1 pointing to the $(m-1)$th frame, and mv_2 pointing to the $(m-2)$th frame. For each pixel p, we can find its reference pixel p_1 in the $(m-1)$th frame with mv_1, and reference pixel p_2 in the $(m-2)$th frame with mv_2, as in Fig. 5a. If p_1 is within an INTRA-MB but p_2 is not, we set $p = p_1$; if p_2 is within an INTRA-MB but p_1 is not, we set $p = p_2$; otherwise, p is decoded as in the conventional decoder. Fig. 5c gives the illustration for this EC process.

To implement this algorithm, we only need to save the positions (indices) of the INTRA-MBs in the previous two frames, when the current frame is reconstructed. So the increase of memory is neglectable.

4. Simulation results

In the simulation, we compare both the compression efficiency and error resilience ability between THMCPF and AMCP, using the H.264/AVC reference software version 11.0 (main profile) [25]. The comparisons between AMCP and the loss-aware rate-distortion optimized MB mode decision algorithm (LARDO) are also given [24]. Video sequences News (QCIF, 300 frames, 10 fps), Foreman (QCIF, 300 frames, 7.5 fps) and Football (QCIF, 250 frames, 15 fps) are used in the simulation, compressed according to the specifications in [28]. All the block sizes from 4×4 to 16×16 are allowed for the motion estimation/compensation and the search range is $[-16, 16]$. For each sequence, only the first frame is encoded as an INTRA-frame, and all the subsequent ones are encoded as INTER frames. One fixed QP is used for the whole sequence, and its value is adjusted to achieve different bit rate.

Suppose the compressed video is transmitted though a packet loss channel, and one packet contains the information of one frame. So the loss of one packet will lead to the loss of one entire frame. The simulated packet loss patterns are obtained from [29], with loss rate P = 3%, 5%, 10%, or 20%. PSNR is used for the objective measurement, which is computed using the original (uncom-

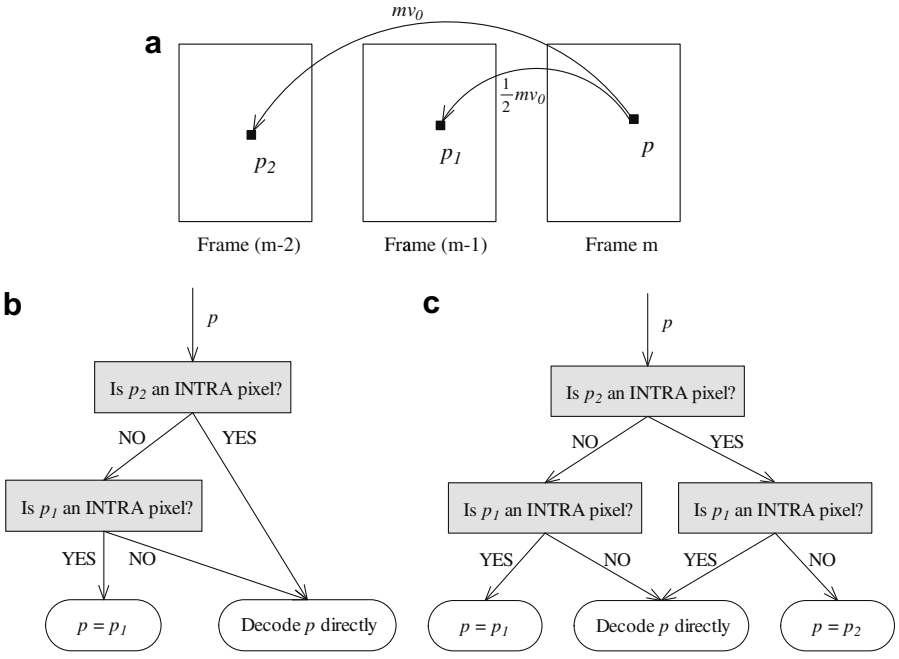


Fig. 5. The reconstruction of INTER pixels in the case of INTRA-frame loss.

Table 1
The parameter values in the config file of THMCPF and AMCP

Parameter name	Value	Comment
RandomIntraMBRefresh	0/2	Number of forced INTRA-MBs per picture
NumberReferenceFrames	2	Number of previous frames used for INTER motion search
UseFME	1	Use fast motion estimation (UMHexagonS)
SymbolMode	1	Entropy coding method is CABAC
RDOptimization	1	R/D optimization enabled
DirectModeType	1	Direct mode type (spatial)
BReferencePictures	1	B pictures are used as references
WeightedBiprediction	1	Weighted prediction for B picture is used (explicit mode)
UseConstrainedIntraPred	1	INTER pixels are not used for INTRA-MB prediction

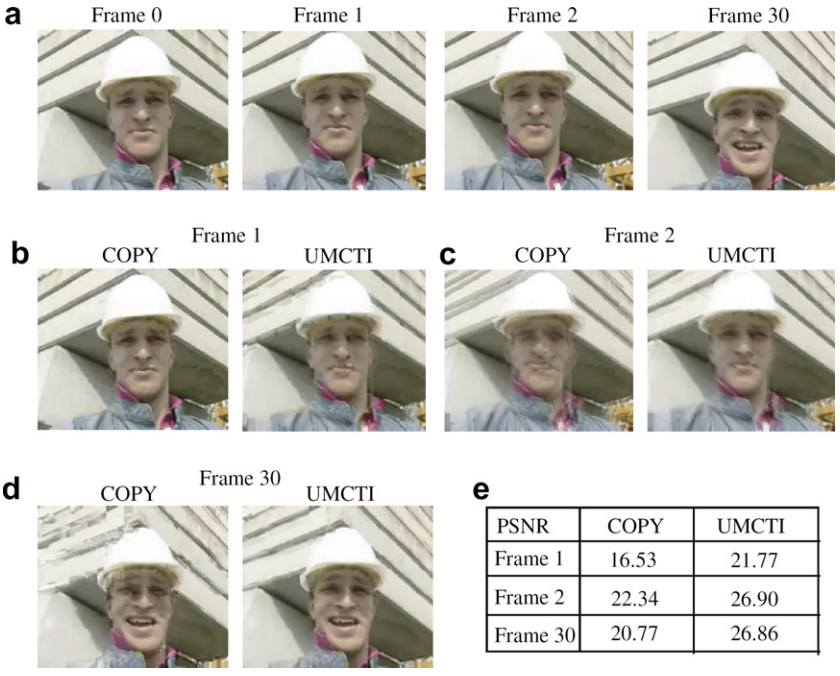
pressed) video as reference. Given a packet loss rate P, the video sequence is transmitted 60 times, and the average PSNR for the 60 transmissions is calculated at the decoder side.

4.1. Comparison between THMCPF and AMCP

In the reference software, the bidirectional prediction of B pictures is generalized to support forward/forward and backward/backward prediction pairs, in addition to the already used forward/backward pair [30]. B pictures can also be used as the reference for the following frame. Due to these generalized form of B pictures, we use B frame to implement the frame with two references (hypotheses) for both THMCPF and AMCP. The weighted prediction mode for B frame is enabled to carry out different weighting parameter h_1 (h_2), which goes from 0.125 to 0.875. For a frame predicted by one-hypothesis in AMCP, e.g. the frame with index n, the reference-list-reordering technique provided by H.264/AVC is used to make frame $(n-2)$ the default reference. And during the motion estimation process, ref_idx_l0 is restricted to be 0 for each MB in frame n. By using these encoding techniques, the generated bitstreams of THMCPF and AMCP are both H.264/AVC standard compatible. Since THMCPF is a special case of AMCP with $N = 0$, the algorithm in Section 3 is used to conceal the lost frames for both THMCPF and AMCP. In Table 1, we list the major parameter values used in the simulation. Note that in the config file of the reference software, parameter *RandomIntraMBRefresh* is used to represent the number of forced INTRA-MBs per picture. We will use the name *IntraRefresh* for short in this paper.

We first give an illustration about the performance of UMCTI in AMCP, with $N = 5$ and $h_2 = 0.5$. Foreman is used as the testing sequence. The frame-copy error concealment is also simulated for the comparison. Suppose a single frame is lost. The reconstructed video frames are shown in Fig. 6, with the frame indices indicated on the top. The first row shows the original encoded frames, with frame index 0, 1, 2 and 30. Suppose the 1th frame is lost, and it can be concealed at the decoder by copying the previous frame, or using the algorithms introduced in Section 3 (UMCTI–AMCP). The two reconstructed frames are shown in Fig. 6b, labelled COPY and UMCTI, respectively. Note that when the lost frame is concealed by UMCTI–AMCP, case 2 of the algorithm is satisfied to reconstruct frame 1 and frame 2. After the concealment, the subsequent frames can be decoded as usual, and we also show the corresponding 2th frames in Fig. 6c and the 30th frames in Fig. 6d for the illustration. The PSNRs of the six frames in the second and the third rows are listed in Fig. 6e. By using UMCTI–AMCP to conceal the lost frame, the video quality at the decoder side has been improved a lot.

Fig. 7 compares the RD curves of THMCPF and AMCP, under different packet loss rate, $P = 3\%$, 5%, 10% or 20%. The RD curves at the encoder side ($P = 0\%$) are also plotted, to give a comparison between the compression efficiencies. The weighting parameter h_2 is 0.5 for both THMCPF and AMCP, and the Interval parameter N for AMCP is 5. In Fig. 7a, i.e. the fist row, Random INTRA-MB Refresh is not used (*IntraRefresh* = 0); in Fig. 7b, i.e. the second row, the number of forced INTRA-MBs for each INTER-frame is 2 (*IntraRefresh* = 2). Note that in both Fig. 7a and b, additional INTRA-MB can be encoded, if it has a lower RD cost in the encoder mode decision procedure. As illustrated in the figure, the compression efficiency of AMCP is better



PSNR	COPY	UMCTI
Frame 1	16.53	21.77
Frame 2	22.34	26.90
Frame 30	20.77	26.86

Fig. 6. The visual results of applying different error concealment methods on Foreman for a single frame loss, $N = 5$ and $h_2 = 0.5$ in AMCP. The lost frame is the 1th one. (a) The original encoded frames without loss, (b) the lost frame is reconstructed by copying the previous one (the left figure) or by UMCTI–AMCP (the right figure), (c) the decoded 2th frame in the sequence, (d) the decoded 30th frame in the sequence, (e) the PSNRs of the frames in (b)–(d).

than THMCPF for all the tested cases. For example, in the two figures of News sequence, the PSNR of AMCP is about 0.34 dB higher than that of THMCPF at bit rate 48 kbps. When loss occurs, AMCP performs better than THMCPF for most of the cases. With *IntraRefresh* = 0 and loss rate $P = 20\%$ in Football, the average PSNR of AMCP is about 0.75 dB higher than that of THMCPF at bit rate 266 kbps. As the percentage of INTRA-MBs increases, i.e. *IntraRefresh* goes from 0 to 2, the gap between AMCP and THMCPF may decrease, especially for Foreman sequence. Since the randomly inserted INTRA-MBs help to suppress the error propagation for both algorithms, the gain of AMCP over THMCPF is reduced.

Fig. 8 gives the comparison between THMCPF and AMCP with different weighting parameter h_2. Fixed bit rate is used. The Interval parameter N for AMCP is 5, and the packet loss rate is $P = 3\%$ or $P = 10\%$. Random INTRA-MB Refresh is not used in the first column, and in the second column, the number of forced INTRA-MBs for each INTER-frame is 2. From the figure we can see that when h_2 increases, the compression efficiencies of both methods decrease, due to the larger weighting parameter for the long-distance reference frame. For the same bit rate R and parameter h_2, AMCP compresses better than THMCPF. One reason is that although two-hypothesis can improve the coding efficiency by a better prediction, the improvement can not compensate the overhead caused by sending double motion vectors. If packet loss occurs, as we can see from the figure, the average PSNR of AMCP is above THMCPF for most of the cases. The smaller h_2, the larger the gap is. For example, with *IntraRefresh* = 2, $h_2 = 0.125$, and loss rate $P = 10\%$, the average PSNR of AMCP is about 0.76 dB higher than that of THMCPF in News, and about 1.63 dB higher in Football. For a large h_2, the performances of these two methods are similar. This is consistent with the discussions in Section 2.3. To make a balance between the compression efficiency and error resilience capability, generally a moderate value of h_2 (e.g. $h_2 = 0.5$) can be used for both AMCP and THMCPF. When the loss rate is small, e.g. $P = 3\%$, we can observe from the figure that the decoder PSNRs of AMCP in News or Football are very close when h_2 goes from

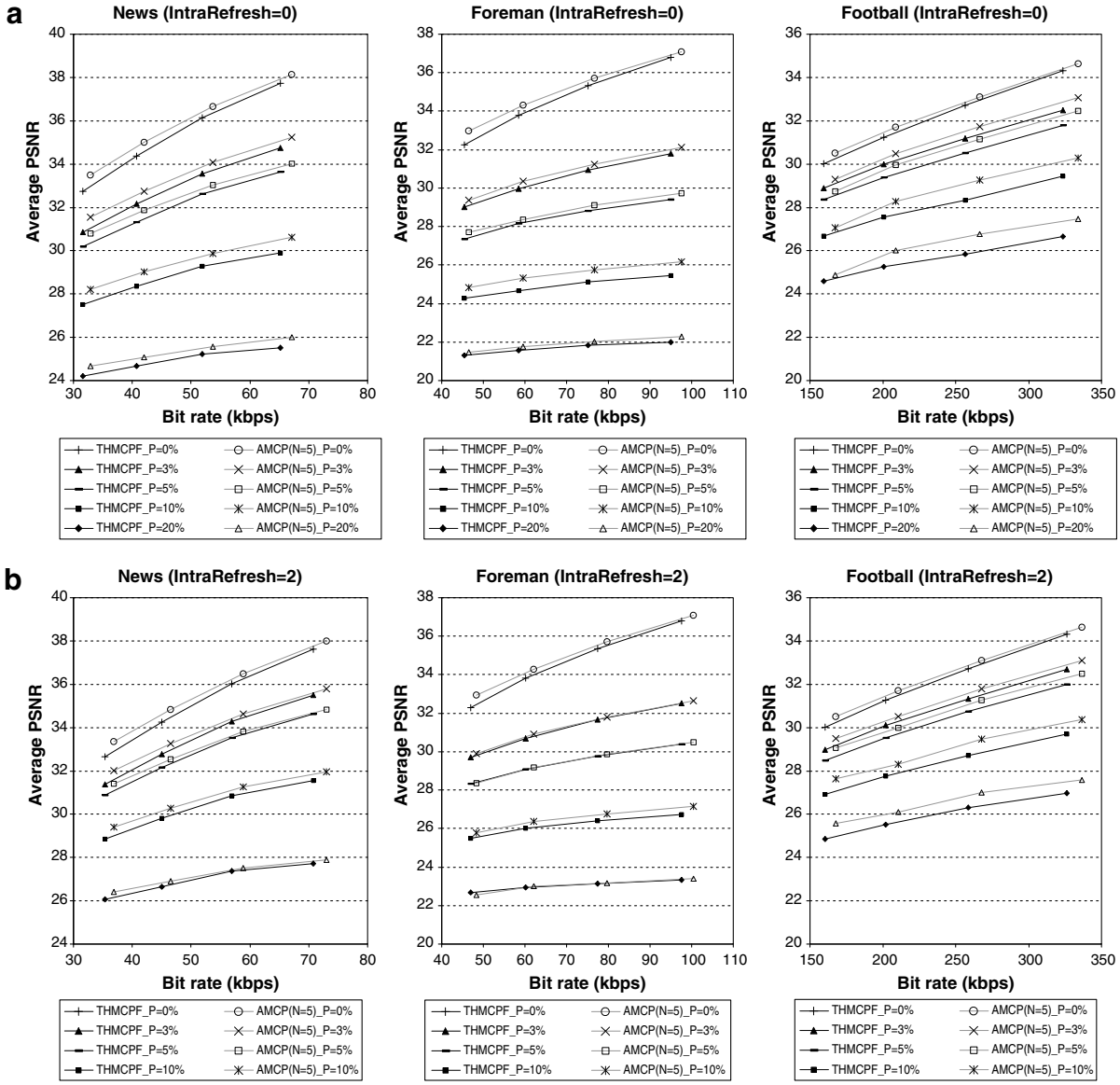


Fig. 7. The RD curves of THMCPF and AMCP for different packet loss rate ($P = 0\%$, $P = 3\%$, $P = 5\%$, $P = 10\%$ or $P = 20\%$). The weighting parameter h_2 is 0.5 and the Interval parameter N for AMCP is 5. In the first row, Random INTRA-MB Refresh is not used (*IntraRefresh* = 0); in the second row, the number of forced INTRA-MBs per frame is 2 (*IntraRefresh* = 2).

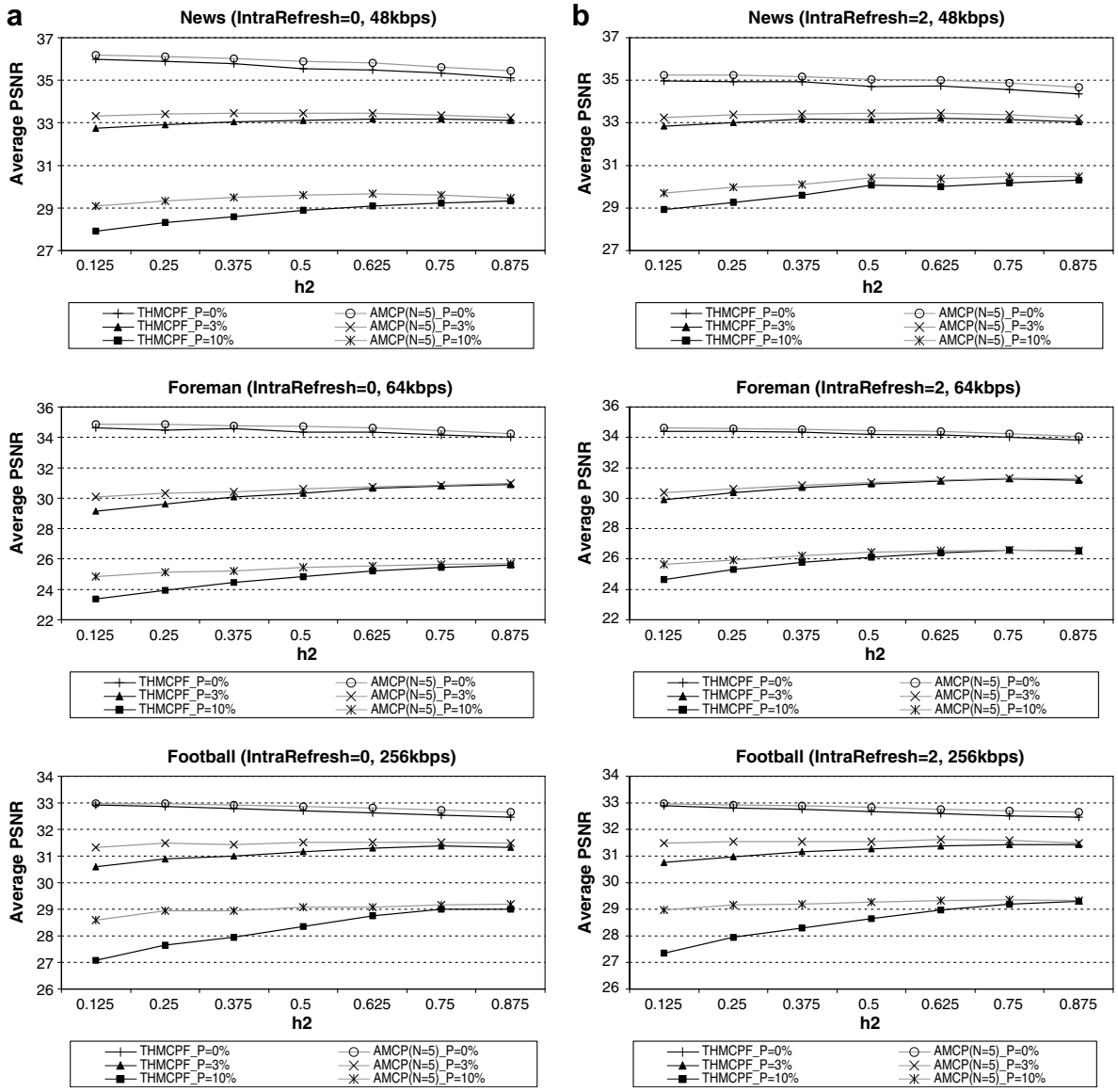


Fig. 8. Comparison between THMCPF and AMCP for different weighting parameter h_2, at a given bit rate. The Interval parameter N for AMCP is 5, and the packet loss rate is $P = 0\%$, $P = 3\%$ or $P = 10\%$. In the first column, Random INTRA-MB Refresh is not used (*IntraRefresh* = 0); in the second column, the number of forced INTRA-MBs per frame is 2 (*IntraRefresh* = 2).

Table 2
The average PSNR of AMCP with different Interval parameter N ($h_2 = 0.5$)

	News (*IntraRefresh* = 0, 48 kbps)										News (*IntraRefresh* = 2, 48 kbps)									
	PSNR					Delta-PSNR					PSNR					Delta-PSNR				
P:	0%	3%	5%	10%	20%	0%	3%	5%	10%	20%	0%	3%	5%	10%	20%	0%	3%	5%	10%	20%
N = 0	35.54	33.10	32.18	28.98	25.04	0.00	0.00	0.00	0.00	0.00	34.71	33.16	32.51	30.06	26.84	0.00	0.00	0.00	0.00	0.00
N = 3	35.86	33.20	32.42	29.09	25.00	0.32	0.10	0.24	0.11	−0.04	35.05	33.23	32.65	30.10	26.68	0.34	0.07	0.14	0.04	−0.16
N = 5	35.88	33.44	32.48	29.47	25.34	0.34	0.34	0.30	0.49	0.30	35.05	33.44	32.70	30.41	26.98	0.34	0.28	0.19	0.35	0.14
N = 7	35.91	33.26	32.31	28.98	24.96	0.37	0.16	0.13	0.00	−0.08	35.06	33.29	32.55	30.04	26.70	0.35	0.13	0.04	−0.02	−0.14
	Foreman (*IntraRefresh* = 0, 64 kbps)										Foreman (*IntraRefresh* = 2, 64 kbps)									
N = 0	34.38	30.33	28.42	24.85	21.67	0.00	0.00	0.00	0.00	0.00	34.21	30.94	29.26	26.13	23.01	0.00	0.00	0.00	0.00	0.00
N = 3	34.66	30.62	28.89	25.34	21.93	0.28	0.29	0.47	0.49	0.26	34.47	31.06	29.60	26.39	23.10	0.26	0.12	0.34	0.26	0.09
N = 5	34.70	30.62	28.59	25.45	21.84	0.32	0.29	0.17	0.60	0.17	34.46	31.02	29.26	26.43	23.03	0.25	0.08	0.00	0.30	0.02
N = 7	34.78	31.10	28.74	25.32	21.85	0.40	0.77	0.32	0.47	0.18	34.55	31.48	29.41	26.33	23.06	0.34	0.54	0.15	0.20	0.05
	Football (*IntraRefresh* = 0, 256 kbps)										Football (*IntraRefresh* = 2, 256 kbps)									
N = 0	32.70	31.17	30.50	28.34	25.85	0.00	0.00	0.00	0.00	0.00	32.67	31.28	30.69	28.66	26.26	0.00	0.00	0.00	0.00	0.00
N = 3	32.84	31.43	30.89	28.92	26.44	0.14	0.26	0.39	0.58	0.59	32.82	31.44	30.96	29.10	26.63	0.15	0.16	0.27	0.44	0.37
N = 5	32.86	31.52	30.94	29.09	26.63	0.16	0.35	0.44	0.75	0.78	32.83	31.55	31.03	29.26	26.84	0.16	0.27	0.34	0.60	0.58
N = 7	32.88	31.49	30.95	28.96	26.58	0.18	0.32	0.45	0.62	0.73	32.83	31.51	31.07	29.13	26.84	0.16	0.23	0.38	0.47	0.58

Table 3
The parameter values in the config file of LARDO

Parameter name	Value	Comment
NumberReferenceFrames	2	Number of previous frames used for INTER motion search
UseFME	1	Use fast motion estimation (UMHexagonS)
SymbolMode	1	Entropy coding method is CABAC
RDOptimization	3	Loss-aware R/D optimization
LossRateA	$P \times 100$	Expected packet loss rate of the channel
NumberOfDecoders	30	Numbers of decoders used to simulate the channel
RestrictRefFrames	1	Does not allow reference to areas that have been INTRA updated in a later frame
UseConstrainedIntraPred	1	INTER pixels are not used for INTRA-MB prediction

0.125 to 0.875. In these situations, $h_2 = 0.125$ can be used for AMCP as it provides the highest compression efficiency.

We also study the effect of Interval parameter N ($N = 0, 3, 5, 7$) on the performance of AMCP, with and without Random INTRA-MB Refresh. The weighting parameter h_2 is 0.5 and the bit rate is fixed for a specific sequence. The average PSNRs are shown in Table 2, with packet loss rate $P = 3\%$, 5%, 10% or 20%. The encoder PSNR is also presented under $P = 0\%$ to show the compression efficiency. To give a clearer illustration, we present the difference between the PSNR with $N \neq 0$ ($N = 3$, 5 or 7) and the PSNR with $N = 0$ (THMCPF) for the same loss rate, as shown in the column named Delta-PSNR. From the table we can see that AMCP with $N = 7$ has a better compression efficiency than other ones. If loss occurs with loss rate P, AMCP with $N = 5$ achieves the highest PSNR in most of the cases for News and Football. For Foreman, the performance of AMCP is similar for $N = 3$, 5 or 7. We can increase N a little to improve the performance of AMCP, i.e. from $N = 0$ to $N = 3$ and $N = 3$ to $N = 5$ in the table. On the other hand, a larger Interval length can lead to a higher randomness for the packet loss position, thus making the performance of

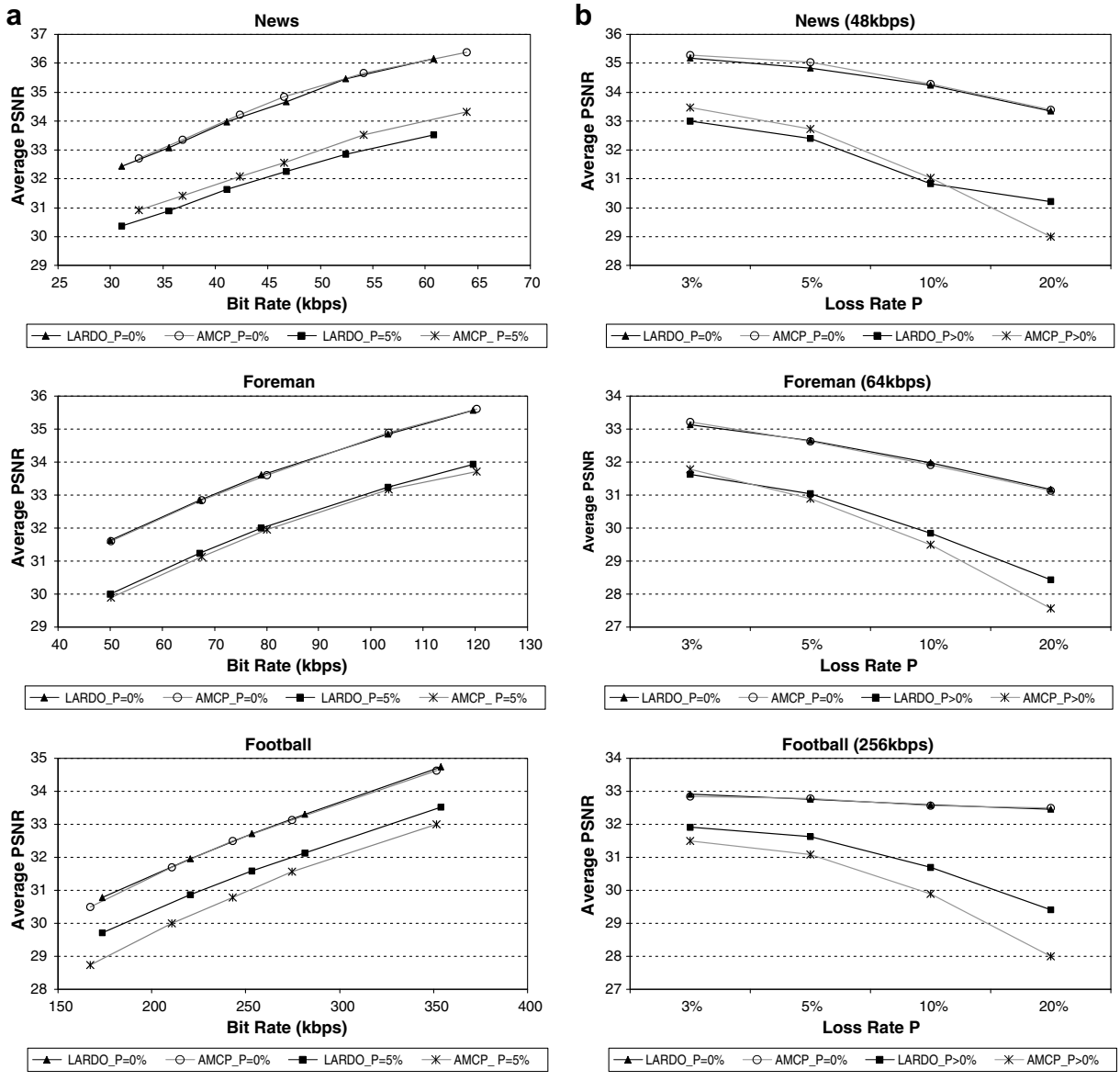


Fig. 9. Comparison between LARDO and AMCP ($N = 5$ and $h_2 = 0.5$). (a) The RD curves of LARDO and AMCP at the encoder side ($P = 0\%$), and those at the decoder side with packet loss rate $P = 5\%$. (b) The average PSNR for different packet loss rate ($P = 3\%$, 5%, 10% or 20%). The bit rate is fixed for a specific sequence, and the corresponding PSNRs at the encoder are also plotted ($P = 0\%$).

AMCP unstable. Due to these facts, a moderate Interval parameter, i.e. $N = 5$, is preferable.

4.2. Comparison between LARDO and AMCP

In the simulation, we also compare the performance of AMCP with the loss-aware rate-distortion optimized MB mode decision algorithm (LARDO), which has been implemented in the reference software [24]. In LARDO, when the mode and reference frame is selected for each MB, the expected decoder distortion is considered in the RD optimization process, instead of the distortion caused by encoder quantization only. In order to do this, the channel statistics are supposed to be known. In traditional hybrid video coding, the encoder includes a decoder for INTER-frame prediction. So in the encoder of LARDO, there are K copies of such decoder, which operate independently based on the channel statistics. Therefore, the expected distortion at the decoder can be estimated from these K decoders. As the loss rate of the channel increases, more MBs are tentatively to be encoded as INTRA to decrease the decoder distortion. For more details about LARDO, please refer to [24]. In Table 3, we list the major parameter values of LARDO used in the simulation. Suppose the LARDO encoder can know the true packet loss rate (P), and the lost frame is concealed by copying the previous one. The parameters used for AMCP is $N = 5$ and $h_2 = 0.5$.

Fig. 9a is the comparison between the RD curves of LARDO and AMCP at the encoder side ($P = 0\%$), and those at the decoder side with packet loss rate $P = 5\%$. For AMCP, the number of randomly inserted INTRA-MBs in the INTER-frame (*IntraRefresh*) is adjusted, so that the encoder RD curves of AMCP and LARDO can overlap on the whole. As illustrated in the figure, the RD curve of AMCP with $P = 5\%$ is about 0.35 dB higher than that of LARDO in News. For Foreman and Football, the average PSNR of AMCP is about 0.1 and 0.7 dB lower than that of LARDO, respectively. These results indicate that for a sequence with relatively simple and small motions, or with a large static background, such as News, AMCP is a better choice than LARDO. The reason is that when packet loss occurs, error concealment can work well in such sequences. In other words, temporal interpolation can reconstruct the frames with a similar quality as those using INTRA-MB refresh, but the corresponding bit rate is much lower.

Fig. 9b compares the performance of AMCP and LARDO under different packet loss rate, $P = 3\%$, 5%, 10% or 20%. The corresponding encoder PSNR ($P = 0\%$) is also presented to show the compression efficiency. The bit rate is fixed for a specific sequence. Different QPs are used to encode the video sequence three times. And the same scheme as in Fig. 8 is used to interpolate the PSNR for a given bit rate. We can see from the figure that for LARDO, the encoder PSNR ($P = 0\%$) is different for different loss rate P, since the encoder needs P to control the number of INTRA-MB inserted. As a result, a larger P will lower the compression efficiency. For AMCP, we also adjust the number of forced INTRA-MBs in the INTER-frame as in Fig. 9a, to make the encoder curves of AMCP close to those of LARDO. As shown in the figure, for a given loss rate in News, the average PSNR of AMCP is about 0.33 dB higher than that of LARDO, except for $P = 20\%$. In Foreman, the average PSNR of AMCP is 0.15 dB higher than that of LARDO at $P = 3\%$. In low loss rate conditions, most of the frames are correctly received and have a good quality. Thus they can help in temporal interpolation to reconstruct the lost frames with a good quality.

Although the decoder PSNR of LARDO is higher than that of AMCP for some simulated cases, AMCP still has some advantages over LARDO. It can be preferred in a sequence with slow motions and in the condition with low loss rate, as shown in Fig. 9. In addition, the complexity and memory requirement of AMCP are much lower than those of LARDO. In the encoder of LARDO, there are K independent decoders, so the complexity and the memory requirement are as K times as those of the traditional encoder. On the other hand, the complexity of AMCP is similar as that of the traditional one, and its memory requirement is the same as the one using two previous frames for INTER motion search. Note that in the simulation, we manually adjust the number of forced INTRA-MBs in the INTER-frame of AMCP, so that the encoder curves of AMCP and LARDO can overlap on the whole. Actually AMCP can be jointly used with other error resilience algorithms which are based on INTRA-MB refresh, such as [31,32]. However, this is out of the main focus of this paper.

5. Conclusion

In this paper, we propose an error resilience approach named alternate motion-compensated prediction (AMCP), where two-hypothesis and one-hypothesis prediction are combined, with an alternate pattern. The Interval parameter N and the weighting parameter h_2 can be combined to adjust its compression efficiency and error resilience capability. Extension of AMCP to layered coding is also given, as a way for multicast of video over the Internet. In addition to this, the appropriate error concealment method for AMCP is discussed, i.e. unidirectional motion-compensated temporal interpolation (UMCTI) [27]. It can be further used to enhance the reconstructed video quality.

Acknowledgments

This work has been supported in part by the Innovation and Technology Commission (project no. GHP/033/05) of the Hong Kong Special Administrative Region, China.

References

[1] Y. Wang, S. Wenger, J. Wen, A.K. Katsaggelos, Error resilient video coding techniques, IEEE Signal Proc. Mag. 17 (2000) 61–82.
[2] Y. Wang, Q.F. Zhu, Error control and concealment for video communication: a review, Proc. IEEE 86 (1998) 974–997.
[3] M. Ghandi, B. Barmada, E. Jones, M. Ghanbari, Unequally error protected data partitioned video with combined hierarchical modulation and channel coding, in: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06), 2006, pp. II-529–II-532.
[4] C.-M. Fu, W.-L. Hwang, C.-L. Huang, Efficient post-compression error-resilient 3D-scalable video transmission for packet erasure channels, in: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), 2005, pp. 305–308.
[5] M. Stoufs, A. Munteanu, P. Schelkens, J. Cornelis, Optimal joint source-channel coding using unequal error protection for the scalable extension of H.264/MPEG-4 AVC, in: Proc. IEEE International Conference on Image Processing (ICIP'07), 2007, pp. IV-517–IV-520.
[6] M. Hannuksela, Y.-K. Wang, M. Gabbouj, Isolated regions in video coding, IEEE Trans. Multimedia 6 (2004) 259–267.
[7] P. Baccichet, S. Rane, A. Chimienti, B. Girod, Robust low-delay video transmission using H.264/AVC redundant slices and flexible macroblock ordering, in: Proc. IEEE International Conference on Image Processing (ICIP'07), 2007, pp. IV-93–IV-96.
[8] E. Setton, P. Baccichet, B. Girod, Peer-to-peer live multicast: a video perspective, Proc. IEEE 96 (2008) 25–38.
[9] R. Aravind, M. Civanlar, A. Reibman, Packet loss resilience of MPEG-2 scalable video coding algorithms, IEEE Trans. Circ. Syst. Video Technol. 6 (1996) 426–435.
[10] Y. Guo, Y.-K. Wang, H. Li, Error resilient mode decision in scalable video coding, in: Proc. IEEE International Conference on Image Processing (ICIP'06), 2006, pp. 2225–2228.
[11] Y.-C. Lee, Y. Altunbasak, R. Mersereau, Coordinated application of multiple description scalar quantization and error concealment for error-resilient MPEG video streaming, IEEE Trans. Circ. Syst. Video Technol. 15 (2005) 457–468.
[12] F. Verdicchio, A. Munteanu, A. Gavrilescu, J. Cornelis, P. Schelkens, Embedded multiple description coding of video, IEEE Trans. Image Process. 15 (2006) 3114–3130.
[13] E. Akyol, A. Tekalp, M. Civanlar, A flexible multiple description coding framework for adaptive peer-to-peer video streaming, IEEE J. Select. Top. Signal Process. 1 (2007) 231–245.
[14] S. Lin, Y. Wang, Error resilience property of multihypothesis motion-compensated prediction, in: Proc. IEEE International Conference on Image Processing (ICIP'02), 2002, pp. 545–548.

[15] Y.-C. Tsai, C.-W. Lin, H.264 error resilience coding based on multihypothesis motion compensated prediction, in: Proc. IEEE International Conference on Multimedia and Expo (ICME'05), 2005, pp. 952–955.
[16] W.-Y. Kung, C.-S. Kim, C.-C. Kuo, Error resilience analysis of multihypothesis motion compensated prediction for video coding, in: Proc. IEEE International Conference on Image Processing (ICIP'04), 2004, pp. 821–824.
[17] W.-Y. Kung, C.-S. Kim, C.-C. Kuo, Analysis of multihypothesis motion compensated prediction (MHMCP) for robust visual communication, IEEE Trans. Circ. Syst. Video Technol. 16 (2006) 146–153.
[18] B. Girod, Efficiency analysis of multihypothesis motion-compensated prediction for video coding, IEEE Trans. Image Process. 9 (2000) 173–183.
[19] M. Flierl, T. Wiegand, B. Girod, Rate-constrained multi-hypothesis motion-compensated prediction for video coding, in: Proc. IEEE International Conference on Image Processing (ICIP'00), 2000, pp. 150–153.
[20] B. Girod, N. Farber, Wireless video, in: A.R. Reibman, M.-T. Sun (Eds.), Compressed Video over Networks, Marcel Dekker, 2000.
[21] M. Ma, O.C. Au, S.-H.G. Chan, A new motion compensation approach for error resilient video coding, in: Proc. IEEE International Conference on Image Processing (ICIP'05), 2005, pp. I-773–I-776.
[22] N. Farber, K. Stuhlmuller, B. Girod, Analysis of error propagation in hybrid video coding with application to error resilience, in: Proc. IEEE International Conference on Image Processing (ICIP'99), 1999, pp. 550–554.
[23] 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), 2001, pp. 392–409.
[24] T. Stockhammer, D. Kontopodis, T. Wiegand, Rate-distortion optimization for JVT/H.26L video coding in packet loss environment, in: Proc. International Packet Video Workshop, 2002.
[25] Jvt reference software, version 11.0. URL: http://iphome.hhi.de/suehring/tml/download/.
[26] C.-K. Wong, O. Au, Fast motion compensated temporal interpolation for video, in: Proc. SPIE Visual Communications and Image Processing (VCIP'95), 1995, pp. 1108–1118.
[27] C.-W. Tang, O. Au, Unidirectional motion compensated temporal interpolation, in: Proc. IEEE International Symposium on Circuits and Systems (ISCAS'97), 1997, pp. 1444–1447.
[28] Y.-K. Wang, S. Wenger, M.M. Hannuksela, Common conditions for SVC error resilience testing, in: ITU-T SG16 Doc. JVT-P206, 2005.
[29] S. Wenger, Error patterns for internet experiments, in: ITU-T SG16 Doc. Q15-I-16r1, 1999.
[30] M. Flierl, B. Girod, Generalized B pictures and the draft H.264/AVC video-compression standard, IEEE Trans. Circ. Syst. Video Technol. 13 (2003) 587–597.
[31] R. Zhang, S. Regunathan, K. Rose, Video coding with optimal inter/intra-mode switching for packetloss resilience, IEEE J. Select. Areas Commun. 18 (2000) 966–976.
[32] G. Cote, S. Shirani, F. Kossentini, Optimal mode selection and synchronization for robust videocommunications over error-prone networks, IEEE J. Select. Areas Commun. 18 (2000) 952–965.