Error Resilient Video Coding Using B Pictures in H.264

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

***Abstract*—Since the quality of compressed video is vulnerable to errors, video transmission over unreliable Internet is very challenging today. Multi-hypothesis motion-compensated prediction (MHMCP) has been shown to have error resilience capability for video transmission, where each macroblock is predicted by a linear combination of multiple signals (hypotheses). B picture prediction is a special case of MHMCP. In H.264/AVC, the prediction of B pictures is generalized such that both of the two predictions can be selected from the past pictures or from the subsequent pictures. The multiple reference picture framework in H.264/AVC also allows previously decoded B pictures to be used as references for B picture coding. In this paper, we will discuss the error resilience characteristics of the generalized B pictures in H.264/AVC. Three prediction patterns of B pictures are analyzed in terms of their error-suppressing abilities. Both theoretical models (picture level error propagation) and simulation results are given for the comparison.**

***Index Terms*—Error propagation, error ratio, error resilience, multi-hypothesis motion-compensated prediction.**

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 the block-based video coding method, if we use INTER prediction mode, each macroblock (MB) is predicted from a previously decoded frame by motion compensation. 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 the 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], [7]. 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], [8], [9]. 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 independently-encoded 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 [10], [11]. In addition to MDC, multi-hypothesis motion-compensated prediction (MHMCP) has also been proven to have error resilience capability, where each MB is predicted by a linear combination of multiple signals (hypotheses) [12], [13]. It is shown in [14] 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 was originally developed to improve the compression efficiency of video coding [15], [16]. Its error resilience property is analyzed in [12], where two-hypothesis MCP is used. In this approach, each picture (except I picture and the first P picture) is predicted from its previous two pictures. The error propagation model at the decoder side is analyzed, which is combined with the encoder predictor to strike a balance between compression efficiency and error resilience capability. In [14], the authors extend two-hypothesis MCP by

Manuscript received November 22, 2006; revised June 15, 2008. First version published July 7, 2009; current version published September 30, 2009. This work was supported in part by the Innovation and Technology Commission of the Hong Kong Special Administrative Region, China, under project numbers GHP/033/05 and GHP/048/08, and by the Hong Kong Innovation Technology Fund, project number ITS/013/08. This paper was recommended by Associate Editor P. Topiwala.

M. Ma and O. C. Au are with the Multimedia Technology Research Center, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China (e-mail: myma@ust.hk; eeau@ust.hk).

S.-H. Gary Chan is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China. He is also with Microsoft Research Asia, Beijing, China (e-mail: gchan@ust.hk).

L. Guo is with Thomson Corporate Research, Princeton, NJ 08540 USA (e-mail: eeglw@ust.hk).

Digital Object Identifier 10.1109/TCSVT.2009.2026795

1051-8215/$26.00 © 2009 IEEE

utilizing the concept of reference picture interleaving and data partitioning. The parameters for each hypothesis, including the motion vectors and the reference picture indices, are separated into the bitstream to reduce the impact of a single picture loss. In [14], the error propagation effect in 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.

We will focus on two-hypothesis MCP (2HMCP) in this paper, to make a balance between the compression efficiency and error resilience capability. In addition, 2HMCP can be supported by the newest video coding standard H.264/AVC, whereas three-hypothesis MCP is not standard compatible [17]. In the older video coding standards, such as MPEG-1/2/4, bidirectional temporal prediction is used to encode B pictures. In this prediction mode, a block in a current picture is predicted from a block in a previous picture using forward prediction, as well as a block in a future picture using backward prediction. Then a weighted average of these two blocks is subtracted from the current block to generate the residue [18]. In H.264/AVC, the prediction of B pictures is generalized such that both of the two predictions can be selected from the past pictures or from the subsequent pictures, in addition to using the existing bidirectional prediction mode in other standards such as MPEG-1/2/4. The multiple reference picture framework in H.264/AVC also allows previously decoded B pictures to be used as references for B picture coding [19]. Due to its generalized form, using B picture to implement 2HMCP is H.264/AVC standard compatible, and thus convenient for the application of error resilient video coding.

In this paper, we will extend the previous works in [12] and [14] to a more generalized one, i.e., the reference pictures can be some distance from the current picture, instead of only the immediately preceding ones. This is analogous to the use of a long-term reference frame in multiple-reference-frame video coding, which has been shown to have the ability to improve both the compression efficiency and error robustness of the encoded videos [20]–[22]. The authors in [20] consider a rate-constrained motion-estimation framework to select the temporal delay parameter (reference frame) and motion vectors. In [21], the error propagation process in multiframe motion compensation is modeled as a multiple Markov chain. The authors use Markov chain analyses to derive a rule to randomize the selection of the reference frame and thus improve the robustness of the codec. In [22], a dual-frame buffer, where one short-term frame and one long-term frame are maintained as reference frames, is used together with an INTRA/INTER mode switching algorithm to improve the encoder's compression efficiency and the robustness of the compressed bitstream. In all these multiple-reference-frame coding algorithms only one of the references (single hypothesis) will be used to predict a MB, whereas a linear combination of multiple signals (hypotheses) will be used to predict a MB in MHMCP. We will focus on the latter one in this paper. Three types of prediction patterns are proposed and implemented by B pictures in H.264/AVC. In the case of a single picture loss during the transmission, the induced error propagation is analyzed at the picture level. Pixel level analyses can be found in the literature, such as [23]–[25]. In these works, the packet loss rate (P) is assumed to be available at the encoder side, and the expected distortion in the decoded video is estimated for each pixel.

This paper is organized as follows. In section II, we derive the error propagation models of the three types of prediction patterns for the case of a single picture loss. Simulation results are given in section III to verify the correctness of these models. The compression efficiencies and error resilience capabilities of the three prediction types are also compared. Section IV is the conclusion.

II. Three Prediction Patterns of B Pictures for Error Resilience

A. Background

Suppose the picture at time m is $\psi(m)$. In [12], each picture has two hypotheses and picture $\psi(m)$ is predicted by

$$\hat{\psi}(m) = h_1\tilde{\psi}(m-1) + h_2\tilde{\psi}(m-2) \qquad (1)$$

where $m \in [2, \infty)$ and $h_1 + h_2 = 1$. $\hat{\psi}(m)$ is the motion-compensated prediction of $\psi(m)$, and $\tilde{\psi}(m-i)$ is the hypothesis from the reconstructed picture at time $(m-i)$, $i = 1, 2$. The first two pictures $\psi(0)$ and $\psi(1)$ can be compressed as the traditional I picture and P picture, respectively. Note that if $h_1 = 1$, the predictor in (1) becomes a conventional prediction scheme. If $h_1 = 0$, this is the same as the odd/even subsampling method used in temporal MDC [26]. In the following analyses, we restrict h_1 and h_2 to be within the range (0, 1) for 2HMCP.

Consider the case of a single picture loss during the transmission. By using some error concealment technique, this picture can be reconstructed at the decoder side with some error. Due to the use of motion compensation, this error will propagate to the subsequent pictures. Suppose the loss occurs at time m_0. We define error $\epsilon(n)$ to be the difference between the reconstructed $(m_0+n)^{th}$ picture at the decoder and the reconstructed $(m_0+n)^{th}$ picture at the encoder. For the special case of all the motion vectors (MVs) being zero, motion compensation is just a copying process. Using the predictor in (1), we can obtain

$$\begin{cases} \epsilon(1) = h_1\epsilon(0), \\ \epsilon(n) = h_1\epsilon(n-1) + h_2\epsilon(n-2), \qquad n \in [2, \infty). \end{cases} \qquad (2)$$

From (2), we can get the error propagation model

$$\epsilon(n) = \frac{1-(h_1-1)^{n+1}}{2-h_1}\epsilon(0). \qquad (3)$$

When n goes to infinity and $h_1 \in (0, 1)$, $\epsilon(n)$ will decrease and converge at last. We define error ratio R_0, to be the ratio of converged value to the first error. So the error ratio of predictor in (1) for a single picture loss is

$$R_0 = \lim_{n\to\infty}\frac{\epsilon(n)}{\epsilon(0)} = \frac{1}{2-h_1}. \qquad (4)$$

One advantage of using 2HMCP is that the error ratio is smaller than one, i.e., $\frac{1}{2} < R_0 < 1$ for $h_1 \in (0, 1)$, which means it can reduce the propagated error.

The above analysis does not consider the effect of spatial filtering, which can be introduced by the deblocking filters, or as a side effect of sub-pixel motion compensation with linear interpolation [27], [28]. Spatial filtering can also attenuate the propagated error energy. This effect is analyzed in [27] for the case of traditional video coding, i.e., single hypothesis MCP, and approximated by

$$D(n) = \frac{D(0)}{1 + \gamma n} \tag{5}$$

where $D(n)$ is the variance of $\epsilon(n)$ and named the decoder distortion in [12]. γ is a parameter describing the efficiency of the loop filter to attenuate the error energy; typically $\gamma \in (0, 1)$. By combining (5) and previous analysis, we can obtain the decoder distortion for the two hypotheses prediction

$$D(n) = \left(\frac{1 - (h_1 - 1)^{n+1}}{2 - h_1}\right)^2 \frac{D(0)}{1 + \gamma n}. \tag{6}$$

In (1), each picture is predicted from its previous two pictures. A straightforward idea is to extend this prediction pattern to a more generalized one

$$\hat{\psi}(m) = h_1 \tilde{\psi}(m - a) + h_2 \tilde{\psi}(m - b) \tag{7}$$

where $b > a \geq 1$. But it turns out that getting the error propagation model for (7) is very complicated. So we only extend the prediction in (1) to the following three 2HMCP prediction patterns:

$$\text{2HMCP Type1:} \quad \hat{\psi}(m) = h_1 \tilde{\psi}(m - c) + h_2 \tilde{\psi}(m - 2c); \tag{8}$$

$$\text{2HMCP Type2:} \quad \hat{\psi}(m) = h_1 \tilde{\psi}(m - 2c) + h_2 \tilde{\psi}(m - 3c); \tag{9}$$

$$\text{2HMCP Type3:} \quad \hat{\psi}(m) = h_1 \tilde{\psi}(m - c) + h_2 \tilde{\psi}(m - 3c). \tag{10}$$

Here c is a constant, $c \geq 1$, $h_1, h_2 \in (0, 1)$ and $h_1 + h_2 = 1$. If $m < 2c$ in 2HMCP Type1, or $m < 3c$ in 2HMCP Type2 and Type3, $\psi(m)$ is predicted by (1). Note that the predictor in (1) is a special case of 2HMCP Type1 with $c = 1$. As all these three prediction patterns select two hypotheses from the past pictures, they can be implemented by the generalized B pictures in H.264/AVC. We define the error propagation function $\epsilon(n)$ for 2HMCP Type1, Type2, and Type3 to be $\epsilon_1(n)$, $\epsilon_2(n)$, and $\epsilon_3(n)$, respectively. Their error propagation models will be analyzed and compared in the subsequent subsections.

B. Error Propagation Model for 2HMCP Type1 Prediction

From the predictors in (8), (9), and (10), we can recursively define the error propagation functions $\epsilon_1(n)$, $\epsilon_2(n)$ and $\epsilon_3(n)$ for the condition of a single picture loss, similar as the one in (2). Deriving the closed-form expressions of these functions is useful in analyzing the error resilience characteristics of 2HMCP. For example, they can help us to estimate the converged values of the propagated errors, so as to compare the error suppressing abilities of different prediction schemes. Actually as shown in this and the subsequent subsections, the derived error propagation functions can be used as long as error occurs within c consecutive pictures for 2HMCP Type1, Type3 and within $2c$ consecutive pictures for 2HMCP Type2. This is more general than the case of a single picture loss. The following work is inspired by the famous problem of Fibonacci numbers, which form a recursively defined sequence F_n. After two starting values, $F_0 = 0$ and $F_1 = 1$, each number of this sequence is calculated by the sum of the two preceding numbers, i.e., $F_n = F_{n-1} + F_{n-2}$ for $n \in [2, \infty)$ [29]. Before deriving the closed-form expressions of $\epsilon_1(n)$, $\epsilon_2(n)$, and $\epsilon_3(n)$, we first give and prove two lemmas, both of which aim to derive the closed-form expression of a recursively defined function

Lemma 1 Suppose a, b, and d are constant parameters; $a, b, d \in \mathbb{R}$ and a function $f(n)$ is recursively defined on $n \in \mathbb{N}^0$.[1]

$$\begin{cases} f(0) = 1, \\ f(1) = d, \\ f(n) = a\,f(n-1) + b\,f(n-2), & n \in [2, \infty). \end{cases} \tag{11}$$

Then the closed-form expression of $f(n)$ is $f(n) =$

$$\begin{cases} \dfrac{(a+\mu)^n(2d - a + \mu) - (a-\mu)^n(2d - a - \mu)}{2^{n+1}\mu}, & \Delta > 0 \\[2ex] \dfrac{(a+\nu)^n(2d - a + \nu) - (a-\nu)^n(2d - a - \nu)}{2^{n+1}\nu}, & \Delta < 0 \\[2ex] \dfrac{(2dn - an + a)a^{n-1}}{2^n}, & \Delta = 0 \end{cases} \tag{12}$$

where $\Delta = a^2 + 4b$, $\mu = \sqrt{\Delta}$ and $\nu = \sqrt{-\Delta}j$.

Proof:

First we define a function $g(n)$ on $n \in \mathbb{N}^0$ as

$$\begin{cases} g(0) = 1 \\ g(1) = a \\ g(n) = a\,g(n-1) + b\,g(n-2), & n \in [2, \infty). \end{cases}$$

With the definition, function $g(n)$ generates a series of numbers recursively, which can be considered as a generalization of the *Fibonacci* sequence [30], [31]. Suppose x_1 and x_2 are the two solutions to equation $x^2 - ax - b = 0$. We can express x_1 as

$$\begin{cases} x_1^2 = ax_1 + b \\ x_1^3 = x_1(ax_1 + b) = a(ax_1 + b) + bx_1 = (a^2 + b)x_1 + ab \\ x_1^4 = x_1((a^2 + b)x_1 + ab) = (a^2 + b)(ax_1 + b) + abx_1 \\ \quad = (a^3 + 2ab)x_1 + (a^2 + b)b \\ \ldots . \end{cases}$$

Similar expressions can be obtained for x_2. Then the relation between x_1, x_2, and $g(n)$ for $n \geq 1$ can be written as

$$\begin{cases} x_1^{n+1} = g(n)x_1 + b\,g(n-1) \\ x_2^{n+1} = g(n)x_2 + b\,g(n-1) \end{cases}$$

which can be inductively proved using the definition of $g(n)$. Subtracting these two equations, we can obtain $g(n) = (x_1^{n+1} - x_2^{n+1})/(x_1 - x_2)$.

We define another function $h(n)$ on $n \in \mathbb{N}^0$ as $h(n) = f(n + 1) - g(n + 1)$. It is easy to prove that $h(n) = (d - a)g(n)$ for

[1] $\mathbb{N}^0$ is the set of nonnegative integers.

$n \in \mathbb{N}^0$. So the relation between function $f(n)$ and $g(n)$ for $n \geq 1$ is $f(n) = (d-a)g(n-1) + g(n)$. From the closed-form expression of $g(n)$, we have

$$f(n) = (d-a)\frac{x_1^n - x_2^n}{x_1 - x_2} + \frac{x_1^{n+1} - x_2^{n+1}}{x_1 - x_2}.$$

By incorporating the solutions of x_1 and x_2 into this equation, we can get the result in (12).

Lemma 2

Suppose a, b, c, and d are constant parameters; $a, b, d \in \mathbb{R}$, $c \in \mathbb{N}$ and a function $f(n)$ is recursively defined on $n \in \mathbb{N}^0$

$$\begin{cases} f(n) = s_n, & n \in [0, c-1] \\ f(n) = d\,f(n-c), & n \in [c, 2c-1] \\ f(n) = a\,f(n-c) + b\,f(n-2c), & n \in [2c, \infty) \end{cases} \tag{13}$$

where $s_0, s_1, \ldots, s_{c-1}$ are known.

Then the closed-form expression of $f(n)$ is $f(rc+k) =$

$$\begin{cases} \dfrac{(a+\mu)^r(2d-a+\mu) - (a-\mu)^r(2d-a-\mu)}{2^{r+1}\mu}s_k, & \Delta > 0 \\ \dfrac{(a+\nu)^r(2d-a+\nu) - (a-\nu)^r(2d-a-\nu)}{2^{r+1}\nu}s_k, & \Delta < 0 \\ \dfrac{(2dr-ar+a)a^{r-1}}{2^r}s_k, & \Delta = 0 \end{cases} \tag{14}$$

where $r, k \in \mathbb{N}^0$, $k \in [0, c-1]$, $\Delta = a^2 + 4b$, $\mu = \sqrt{\Delta}$ and $\nu = \sqrt{-\Delta}\,j$.

Proof:

Based on the definition of function $f(n)$, the range of $f(n)$ can be divided into c cosets, i.e., coset k contains $f(k)$, $f(c+k)$, $f(2c+k)\ldots$, $k \in [0, c-1]$. Following from the definition, the calculations of $f(n)$ in different cosets are uncorrelated. In other words, for $r_1, r_2 \in [0, \infty)$ and $k_1, k_2 \in [0, c-1]$, $f(r_1c+k_1)$ and $f(r_2c+k_2)$ can be independently determined if $k_1 \neq k_2$. Within a given coset k, $k \in [0, c-1]$, we can write $f(n)$ as $f(rc+k) =$

$$\begin{cases} s_k & r = 0 \\ d\,s_k & r = 1 \\ a\,f((r-1)c+k) + b\,f((r-2)c+k) & r \in [2, \infty). \end{cases}$$

Then lemma 1 can be used to derive the expression of $f(rc+k)$ for $r \in \mathbb{N}^0$.

For the 2HMCP Type1 predictor in (8), we can obtain its error propagation function $\epsilon_1(n)$ as

$$\begin{cases} \epsilon_1(n) = e_n & n \in [0, c-1] \\ \epsilon_1(n) = h_1\epsilon_1(n-c) & n \in [c, 2c-1] \\ \epsilon_1(n) = h_1\epsilon_1(n-c) + h_2\epsilon_1(n-2c) & n \in [2c, \infty) \end{cases}$$

where the first c errors $e_0, e_1, \ldots, e_{c-1}$ are assumed to be known, i.e., $e_n = \epsilon_1(n)$ for $n \in [0, c-1]$. Note that $\epsilon_1(n)$ is in the form of function $f(n)$ in lemma 2, with $d = a = h_1$, $b = h_2 = (1-h_1)$, and $(a^2+4b) = (2-h_1)^2 > 0$. Using (14) we can obtain the closed-form expression of $\epsilon_1(n)$

$$\epsilon_1(rc+k) = \frac{1-(h_1-1)^{r+1}}{2-h_1}e_k \tag{15}$$

where $r, k \in \mathbb{N}^0$ and $k \in [0, c-1]$. From this result, we can observe that the range of $\epsilon_1(n)$ can be divided into c cosets, i.e., coset k contains $\epsilon_1(k)$, $\epsilon_1(c+k)$, $\epsilon_1(2c+k)\ldots$, $k \in [0, c-1]$. Using (15), the converged error in coset k can be obtained as

$$\lim_{r\to\infty}\epsilon_1(rc+k) = \frac{e_k}{2-h_1}. \tag{16}$$

For the same e_k, as $h_1 \in (0,1)$, a smaller h_1 leads to a smaller converged error. Note that if $h_1 \in (0,1)$, the base (h_1-1) of the exponent part in (15) is less than zero, which means that there will be some oscillations in the propagated error. We can notice these phenomena from the simulation results of section III. If h_1 is allowed to be 1, then $\epsilon_1(rc+k) = e_k$ for $r \in [0, \infty)$. In this case, error will just propagate within the corresponding coset and will not decrease. If h_1 is allowed to be 0, then $\epsilon_1(2rc+k) = e_k$ and $\epsilon_1((2r+1)c+k) = 0$ for $r \in [0, \infty)$. In this case, erroneous pictures and correct pictures will be alternately displayed, i.e., the propagated error will not converge.

C. Error Propagation Model for 2HMCP Type2 Prediction

Based on the predictor in (9), we can obtain the error propagation function of 2HMCP Type2

$$\begin{cases} \epsilon_2(n) = e_n & n \in [0, 2c-1] \\ \epsilon_2(n) = h_1\epsilon_2(n-2c) & n \in [2c, 3c-1] \\ \epsilon_2(n) = h_1\epsilon_2(n-2c) + h_2\epsilon_2(n-3c) & n \in [3c, \infty) \end{cases}$$

where the first $2c$ errors $e_0, e_1, \ldots, e_{2c-1}$ are assumed to be known, i.e., $e_n = \epsilon_2(n)$ for $n \in [0, 2c-1]$. Note that for $n \in [3c, \infty)$, function $\epsilon_2(n)$ can be expressed as $\epsilon_2(n) - \epsilon_2(n-c) = (-1)[\epsilon_2(n-c) - \epsilon_2(n-2c)] + (-h_2)[\epsilon_2(n-2c) - \epsilon_2(n-3c)]$. We define an auxiliary function $\delta(n) = \epsilon_2(n+c) - \epsilon_2(n)$. Then $\delta(n)$ can be written as

$$\begin{cases} \delta(n) = e_{n+c} - e_n & n \in [0, c-1] \\ \delta(n) = \dfrac{h_1e_{n-c} - e_n}{e_n - e_{n-c}}\delta(n-c) & n \in [c, 2c-1] \\ \delta(n) = (-1)\delta(n-c) + (-h_2)\delta(n-2c) & n \in [2c, \infty). \end{cases}$$

By expanding $\epsilon_2(n)$ as $\epsilon_2(n) = \epsilon_2(n-c) + \delta(n-c) = \epsilon_2(n-2c) + \delta(n-2c) + \delta(n-c) = \ldots = \epsilon_2(n-rc) + \sum_{i=1}^{r}\delta(n-ic)$, we can obtain the relation between $\epsilon_2(n)$ and $\delta(n)$

$$\epsilon_2(rc+k) = \sum_{i=0}^{r-1}\delta(ic+k) + e_k \tag{17}$$

for $r, k \in \mathbb{N}^0$ and $k \in [0, c-1]$. By using lemma 2, the closed-form expression of $\delta(ic+k)$ can be derived for $i \in [0, \infty)$, $k \in [0, c-1]$. Substituting this result for $\delta(ic+k)$ into (17), $\epsilon_2(rc+k)$ can be derived, i.e., $\epsilon_2(rc+k) =$

$$\begin{cases} \left(\frac{(1-\rho_2^r)(2\alpha+1+\rho)}{1-\rho_2} - \frac{(1-\rho_3^r)(2\alpha+1-\rho)}{1-\rho_3}\right)\frac{e_{c+k}-e_k}{2\rho} + e_k \\ \text{if } h_1 \in (\frac{3}{4}, 1), \\ \left(\frac{(1-\tau_2^r)(2\alpha+1+\tau)}{1-\tau_2} - \frac{(1-\tau_3^r)(2\alpha+1-\tau)}{1-\tau_3}\right)\frac{e_{c+k}-e_k}{2\tau} + e_k \\ \text{if } h_1 \in (0, \frac{3}{4}), \\ \left((-\frac{1}{2})^r((12\alpha+6)r-4\alpha-8)+4\alpha+8\right)\frac{e_{c+k}-e_k}{9} + e_k \\ \text{if } h_1 = \frac{3}{4} \end{cases} \tag{18}$$

where $r, k \in \mathbb{N}^0$, $k \in [0, c-1]$, $\alpha = (h_1 e_k - e_{c+k})/(e_{c+k} - e_k)$, $\rho = \sqrt{4h_1 - 3}$, $\rho_2 = (-1+\rho)/2$, $\rho_3 = (-1-\rho)/2$, $\tau = \sqrt{3-4h_1}j$, $\tau_2 = (-1+\tau)/2$ and $\tau_3 = (-1-\tau)/2$. Note that in 2HMCP Type2, there are three cases, depending on the range of h_1. Although τ is a complex number when $h_1 \in (0, \frac{3}{4})$, it is easy to prove that the imaginary part of (18) is zero, i.e., $\epsilon_2(rc+k)$ is a real number.

Similar to 2HMCP Type1, the range of $\epsilon_2(n)$ can be divided into c cosets, i.e., coset k contains $\epsilon_2(k), \epsilon_2(c+k), \epsilon_2(2c+k)\ldots$, $k \in [0, c-1]$. By (18), the converged error in coset k can be obtained (the same for the three cases) as

$$\lim_{r\to\infty} \epsilon_2(rc+k) = \frac{e_k + e_{c+k}}{3-h_1}. \tag{19}$$

For the same e_k and e_{c+k}, as $h_1 \in (0, 1)$, a smaller h_1 leads to a smaller converged error. If h_1 is allowed to be 1, then $\epsilon_2(2rc+k) = e_k$ and $\epsilon_2((2r+1)c+k) = e_{c+k}$ for $r \in [0, \infty)$. In this case, error will just propagate within the corresponding coset and will not decrease. If h_1 is allowed to be 0, then $\epsilon_2(3rc+k) = e_k$, $\epsilon_2((3r+1)c+k) = e_{c+k}$ and $\epsilon_2((3r+2)c+k) = 0$ for $r \in [0, \infty)$. In this case, erroneous pictures and correct pictures will be alternately displayed, i.e., the propagated error will not converge.

D. Error Propagation Model for 2HMCP Type3 Prediction

Based on the predictor in (10), the error propagation function of 2HMCP Type3 can be obtained as

$$\begin{cases} \epsilon_3(n) = e_n & n \in [0, c-1] \\ \epsilon_3(n) = h_1\epsilon_3(n-c) & n \in [c, 3c-1] \\ \epsilon_3(n) = h_1\epsilon_3(n-c) + h_2\epsilon_3(n-3c) & n \in [3c, \infty) \end{cases}$$

where the first c errors $e_0, e_1, \ldots, e_{c-1}$ are assumed to be known, i.e., $e_n = \epsilon_3(n)$ for $n \in [0, c-1]$. Note that for $n \in [3c, \infty)$, function $\epsilon_3(n)$ can be written as $\epsilon_3(n) - \epsilon_3(n-c) = (-h_2)[\epsilon_3(n-c) - \epsilon_3(n-2c)] + (-h_2)[\epsilon_3(n-2c) - \epsilon_3(n-3c)]$. We define an auxiliary function $\dot{\delta}(n) = \epsilon_3(n+c) - \epsilon_3(n)$. Then using lemma 2 and a similar deriving process to 2HMCP Type2, we can get the closed-form expression of $\dot{\delta}(n)$. Note that for $h_2 \in (0, 1)$, ${h_2}^2 - 4h_2 < 0$, so the condition of $(a^2 + 4b) < 0$ is satisfied in (14). In addition

$$\epsilon_3(rc+k) = \sum_{i=0}^{r-1} \dot{\delta}(ic+k) + e_k$$

for $r, k \in \mathbb{N}^0$ and $k \in [0, c-1]$. By combining this with the expression of $\dot{\delta}(ic+k)$, we can get

$$\epsilon_3(rc+k) = \left(\frac{(1-\omega_2^r)(h_1+1+\omega)}{1-\omega_2} - \frac{(1-\omega_3^r)(h_1+1-\omega)}{1-\omega_3}\right) \times \frac{(h_1-1)e_k}{2\omega} + e_k \tag{20}$$

where $r, k \in \mathbb{N}^0$, $k \in [0, c-1]$, $\omega = \sqrt{3 - 2h_1 - {h_1}^2}j$, $\omega_2 = (h_1 - 1 + \omega)/2$ and $\omega_3 = (h_1 - 1 - \omega)/2$. Similarly as 2HMCP Type2, although ω is a complex number, it can be proved that the imaginary part of (20) is zero, i.e., $\epsilon_3(rc+k)$ is a real number.

The range of $\epsilon_3(n)$ can also be divided into c cosets, i.e., coset k contains $\epsilon_3(k), \epsilon_3(c+k), \epsilon_3(2c+k)\ldots$, $k \in [0, c-1]$. The converged error in coset k is

$$\lim_{r\to\infty} \epsilon_3(rc+k) = \frac{e_k}{3-2h_1}. \tag{21}$$

For the same e_k, as $h_1 \in (0, 1)$, a smaller h_1 leads to a smaller converged error. If h_1 is allowed to be 1, then $\epsilon_3(rc+k) = e_k$ for $r \in [0, \infty)$. In this case, error will just propagate within the corresponding coset and will not decrease. If h_1 is allowed to be 0, then $\epsilon_3(3rc+k) = e_k$, $\epsilon_3((3r+1)c+k) = 0$ and $\epsilon_3((3r+2)c+k) = 0$ for $r \in [0, \infty)$. In this case, erroneous pictures and correct pictures will be alternately displayed, i.e., the propagated error will not converge.

E. Error Ratios and Decoder Distortions for the Three Prediction Patterns

From (16), (19), and (21), we can see that the converged error can be represented as a function of the initial error(s) and parameter h_1 for all the three prediction patterns. With fixed initial error(s), when h_1 decreases, the converged error decreases. This means that we can decrease h_1 to improve the error resilience property of the compressed video. On the other hand, when h_1 decreases, the compression efficiency will decrease, due to the larger weighting parameter for the long-distance reference frame. We need to select a proper h_1 to make a balance between the compression efficiency and error resilience capability. More discussions for the value of h_1 will be given in Section III.

As in Section II-A, suppose only one picture is lost and e_0 is the initial error introduced by the lost picture $[\psi(m_0)]$. By the definitions of 2HMCP Type1 and Type3, we can see that the predictions of the subsequent $(c-1)$ pictures are not related to $\psi(m_0)$, as a result $e_k = 0$ for $k \in [1, c-1]$. Similarly, $e_k = 0$ for $k \in [1, 2c-1]$ in 2HMCP Type2. This means that in all of these prediction patterns, the error incurred by a single picture loss will only affect part of the subsequent pictures: one erroneous picture followed by $(c-1)$ correct pictures, with a cycle equal to c pictures. One exception is picture $\psi(m_0+c)$ in 2HMCP Type2, which may not be erroneous as it is not predicted from $\psi(m_0)$.

Since the constant c makes the error propagation periodic, we will use $c = 1$ in the following analyses. Based on (16), (19), and (21), we can calculate the error ratios for the three types of prediction patterns (R_1, R_2, R_3)

$$R_1 = \frac{1}{2-h_1};\ R_2 = \frac{1}{3-h_1};\ R_3 = \frac{1}{3-2h_1}. \tag{22}$$

When $h_1 \in (0, 1)$, it can be easily proved that $R_2 < R_3 < R_1$. This means 2HMCP Type2 is better than Type1 and Type3, in terms of the error suppressing ability. We will verify this by the simulation results in Section III. To help to explain why 2HMCP Type2 can achieve a smaller error ratio, we list the errors in the first three pictures from the time the loss occurs (with the same initial error e_0 and parameter h_1)

$$\begin{array}{lllll}
\text{2HMCP Type1}: & e_0, & h_1 e_0, & (h_1^2 + h_2) e_0, & \ldots \\
\text{2HMCP Type2}: & e_0, & 0, & h_1 e_0, & \ldots \\
\text{2HMCP Type3}: & e_0, & h_1 e_0, & h_1^2 e_0, & \ldots .
\end{array}$$

There is an error-free picture in 2HMCP Type2 after the loss position, which will be used as the reference for the subsequent pictures and thus can help to suppress the propagated errors.

The previous analyses assume that all the motion vectors are zero. Suppose $D_i(n)$ is the variance of $\epsilon_i(n)$, $i = 1, 2, 3$. By incorporating the spatial filtering in (5) into (15), (18), and (20) and setting $c = 1$, we can get the decoder distortion functions for a single picture loss in (23).

III. Simulation Results

In the simulation, we compare both the compression efficiencies and error resilience abilities of the three types of prediction patterns, using the H.264/AVC reference software version 10.2 (main profile) [32]. The video sequences *Carphone* (QCIF, 300 frames), *Foreman* (QCIF, 300 frames), *Mother_Daughter* (QCIF, 300 frames), *Mobile* (CIF, 300 frames), and *News* (CIF, 150 frames) are used for the testing, encoded at 30 frames/s. Due to its generalized form, we use B picture in H.264/AVC to implement 2HMCP Type1, Type2, and Type3 with $c = 1, 2, 3, 4$. In particular, the first picture of a video sequence is encoded as an I picture, and the second one is encoded as a P picture; then all the subsequent pictures are encoded as B pictures. During the motion estimation/compensation process, a weighted average of two hypotheses is used to predict each block in a B picture, and the reference picture is selected according to the prediction type and the value of distance parameter c. The search range in motion estimation is $[-32, 32]$ for QCIF sequences and $[-64, 64]$ for CIF sequences. The weighted prediction mode for B picture is enabled to carry out different weighting parameter h_1, which goes from 0.125 to 0.875. In Table I, we list the major parameter values used for the encoder configuration. Note that although *Random INTRA-MB Refresh* is not used, additional INTRA-MB can be encoded if it has a lower RD cost in the encoder mode-decision procedure.

TABLE I
The Major Parameter Values in the Encoder Config. File

Parameter Name	Value	Comment
NumberReferenceFrames	2–12	Number of previous frames used for INTER motion search
RandomIntraMBRefresh	0	Number of forced INTRA-MBs per picture
SearchRange	32/64	Maximum search range for ME
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)
SymbolMode	1	Entropy coding method is CABAC
PicInterlace	0	Frame coding
UseConstrainedIntraPred	0	INTER pixels can be used for INTRA-MB prediction
LoopFilterDisable	0	Enable loop filter in slice header

A. Compression Efficiency

Table II lists the comparisons between the three types of encoders, with $c = 1$ and different weighting parameter h_1. Fixed QPs are used to encode the sequences: 28 for I picture and 30 for P (B) pictures. The average PSNR and the bit rate (kbit/s) are listed. To analyze the compression efficiency, we can use the method proposed in [33] and approximate the relation between ΔPSNR and ΔBitrate by $0.05\,\text{dB} \simeq 1\%$. From the table we can see that the compression efficiency of 2HMCP Type3 is closer to Type2 than to Type1 for a small h_1 (i.e., $h_1 = 0.125$), and closer to Type1 than to Type2 for a large h_1 (i.e., $h_1 = 0.875$). In most cases, 2HMCP Type1 compresses best and 2HMCP Type2 compresses worst. This is reasonable since 2HMCP Type1 uses the closest two pictures as the references so that the prediction is better. On the other hand, the references of 2HMCP Type2 are farther than those of 2HMCP Type1 and Type3, which can lead to a worse prediction and larger residue. The common characteristic for these three methods is that increasing h_1 can lead to a better compression efficiency. One exception is the *Mobile* sequence, in which 2HMCP Type3 compresses better than Type1 for a larger h_1, i.e., $h_1 \geq 0.5$, and increasing h_1 does not always improve the compression efficiency. This is mainly due to the sub-pixel motion vectors of the pixels in the calendar. In other words, if the motion estimation/compensation of such a pixel is performed in the immediately previous reference picture, interpolation is needed to get the reference pixel at a sub-pixel location. On the other hand, if the motion estimation/compensation is performed in a farther reference picture, the reference pixel may lie on an integer-pixel location and thus have a better quality than the interpolated one. In such cases, using a long-distance reference picture may have a better compression efficiency than using a neighboring one.

B. Oscillation and Transition Time of the Propagated Errors

As explained in Section II that when $c = 1$, the propagated error due to a single picture loss will decrease and converge at last, if the effect of spatial filtering is not considered. In this section, we will discuss the fluctuation in the reconstructed video quality caused by this error propagation. Due to the use of 2HMCP, the propagated error will stabilize after some oscillations. We define the time needed for this stabilization to be *Error Transition Time* and compare it for the three prediction patterns. As we know, the perceptible quality of a video sequence is bad when good pictures and bad pictures are alternately displayed, even when the average PSNR is high. So a shorter *Error Transition Time* helps to improve the visual quality of the decoded video. The *Decoder Distortion* models

$$
\begin{cases}
D_1(n) = \left(\dfrac{1-(h_1-1)^{n+1}}{2-h_1}\right)^2 \dfrac{D_1(0)}{1+\gamma n}. \\[2ex]
D_2(n) = \begin{cases}
\left(1 - \dfrac{(1-\rho_2^n)(1-2h_1+\rho)}{2\rho(1-\rho_2)} + \dfrac{(1-\rho_3^n)(1-2h_1-\rho)}{2\rho(1-\rho_3)}\right)^2 \dfrac{D_2(0)}{1+\gamma n} & \text{if } h_1 \in (\frac{3}{4}, 1), \\[2ex]
\left(1 - \dfrac{(1-\tau_2^n)(1-2h_1+\tau)}{2\tau(1-\tau_2)} + \dfrac{(1-\tau_3^n)(1-2h_1-\tau)}{2\tau(1-\tau_3)}\right)^2 \dfrac{D_2(0)}{1+\gamma n} & \text{if } h_1 \in (0, \frac{3}{4}), \\[2ex]
\left(\dfrac{4+(-\frac{1}{2})^n(3n+5)}{9}\right)^2 \dfrac{D_2(0)}{1+\gamma n} & \text{if } h_1 = \frac{3}{4},
\end{cases} \\
\text{where } \rho = \sqrt{4h_1-3},\ \rho_2 = (-1+\rho)/2,\ \rho_3 = (-1-\rho)/2,\ \tau = \sqrt{3-4h_1}\,j,\ \tau_2 = (-1+\tau)/2 \\
\text{and } \tau_3 = (-1-\tau)/2. \\[2ex]
D_3(n) = \left(1 + \dfrac{h_1-1}{2\omega}\left(\dfrac{(1-\omega_2^n)(h_1+1+\omega)}{1-\omega_2} - \dfrac{(1-\omega_3^n)(h_1+1-\omega)}{1-\omega_3}\right)\right)^2 \dfrac{D_3(0)}{1+\gamma n}, \\
\text{where } \omega = \sqrt{3-2h_1-{h_1}^2}\,j,\ \omega_2 = (h_1-1+\omega)/2 \text{ and } \omega_3 = (h_1-1-\omega)/2.
\end{cases}
\tag{23}
$$

TABLE II

COMPARISON BETWEEN THE ENCODERS

	PSNR							Bit Rate						
h_1	0.125	0.25	0.375	0.5	0.625	0.75	0.875	0.125	0.25	0.375	0.5	0.625	0.75	0.875
Capphone (QCIF)														
Type1	35.01	34.93	34.92	34.84	34.84	34.85	34.82	152.02	146.52	139.96	139.24	133.82	133.46	131.60
Type2	35.09	35.06	35.05	34.97	34.98	34.99	34.99	173.32	168.62	163.34	162.26	159.08	160.56	160.58
Type3	35.10	35.04	35.00	34.88	34.86	34.85	34.86	168.34	160.94	151.76	147.88	140.58	138.74	134.80
Foreman (QCIF)														
Type1	34.41	34.39	34.39	34.30	34.30	34.27	34.24	147.06	142.26	135.26	135.06	128.46	127.18	123.78
Type2	34.47	34.47	34.49	34.44	34.45	34.42	34.38	175.26	170.10	164.56	163.04	158.88	158.88	157.62
Type3	34.51	34.48	34.46	34.35	34.34	34.28	34.23	171.54	163.48	154.10	149.54	140.46	136.12	129.52
Mother_Daughter (QCIF)														
Type1	36.16	36.04	36.08	35.86	35.95	35.95	35.97	51.18	49.80	46.66	45.96	43.68	43.06	40.50
Type2	36.25	36.20	36.24	36.04	36.15	36.10	36.15	64.82	63.36	60.64	59.82	58.36	58.48	56.86
Type3	36.26	36.17	36.14	35.88	35.96	35.89	35.92	62.10	59.58	55.02	52.72	49.42	47.64	43.70
Mobile (CIF)														
Type1	31.94	31.91	31.88	31.87	31.83	31.85	31.84	1288.39	1216.22	1151.94	1170.46	1139.85	1182.20	1230.24
Type2	32.02	32.02	32.00	31.98	31.97	31.96	31.96	1369.35	1296.19	1235.00	1252.10	1224.31	1268.36	1322.16
Type3	32.01	31.98	31.95	31.91	31.87	31.86	31.85	1344.46	1252.63	1173.43	1174.81	1133.86	1169.88	1217.73
News (CIF)														
Type1	37.20	37.15	37.15	36.99	37.07	37.05	37.07	242.71	233.91	222.58	219.81	208.30	203.74	195.07
Type2	37.25	37.22	37.24	37.09	37.18	37.17	37.20	287.53	280.11	271.73	270.89	263.05	262.57	259.87
Type3	37.25	37.20	37.20	36.98	37.08	37.01	37.02	279.28	265.76	250.61	242.54	225.60	217.06	203.82

and *Error Ratios* derived in section II-E are also verified in this section. *Carphone* (QCIF) sequence is used for the simulation, compressed as in Table II with $c = 1$.

To verify the *Decoder Distortion* models, the condition of a single picture loss at time m_0 is simulated. The lost picture is error concealed by copying the previous correctly decoded picture. Then the *Mean Square Errors* (MSEs, compared to the error-free decoded pictures) at the decoder after that picture are obtained and shown in Fig. 1, Fig. 2, where $m_0 = 20$. The distortion calculated by the theoretical model in (23) is also plotted, where $D_i(0)$ is assigned to be the decoder MSE at the loss position (time m_0), $i = 1, 2, 3$. Parameter γ is trained to fit the curves, which is in the range of $[0.03, 0.04]$. From the figures we can see that the theoretical model can approximate closely to the decoder distortion. To give more illustrations, the differences between the theoretical distortions and the actual decoder MSEs are calculated for the 50 pictures after loss. Their mean and variance (VAR) are obtained and listed in Table III. Except for 2HMCP Type3 with $h_1 = 0.75$, the absolute value of the mean difference is always less than 1. The variance of the difference is also small, around 2.18, 0.43, and 1.12 for 2HMCP Type1, Type2, and Type3, respectively.

When we define *Error Ratio* in Section II-A, we assume all the motion vectors are zero, i.e., the spatial filtering effect is not considered, since the main focus here is on analyzing the error resilience characteristics introduced by different motion compensation schemes. Based on this assumption, error will converge after a long enough time, i.e., at time $(m_0 + n_c)$. Multiplying the initial error by the *Error Ratio*, we can get this converged error, which can be combined with the spatial filter in (5) to get the *Decoder Distortion* at time $(m_0 + n_c)$: $D_i(n_c) \simeq (R_i)^2 D_i(0)/(1+\gamma n_c)$, $i = 1, 2, 3$. From this result, we

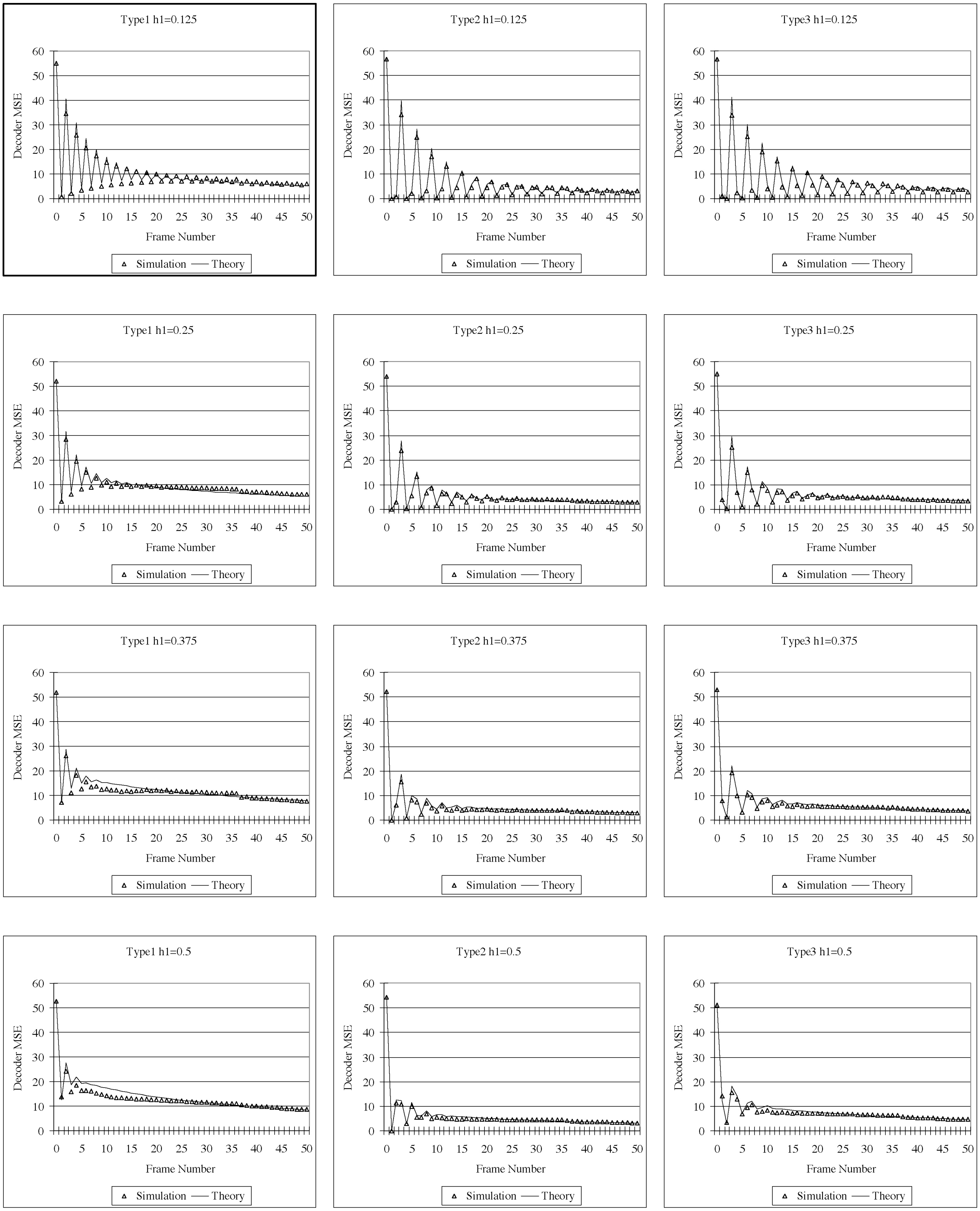


Fig. 1. Error propagation curve of *Carphone* for a single picture loss ($c = 1$). The three types of prediction patterns are listed in the three columns, and the range of the weighting parameter h_1 is [0.125, 0.875], from the top to the bottom. In each plot, the horizontal axis is the frame number after the loss position, and the vertical axis is the *Decoder Distortion*, represented as MSE. This figure only shows the plots for $h_1 \in [0.125, 0.5]$; the remaining ones are listed in Fig. 2.
]

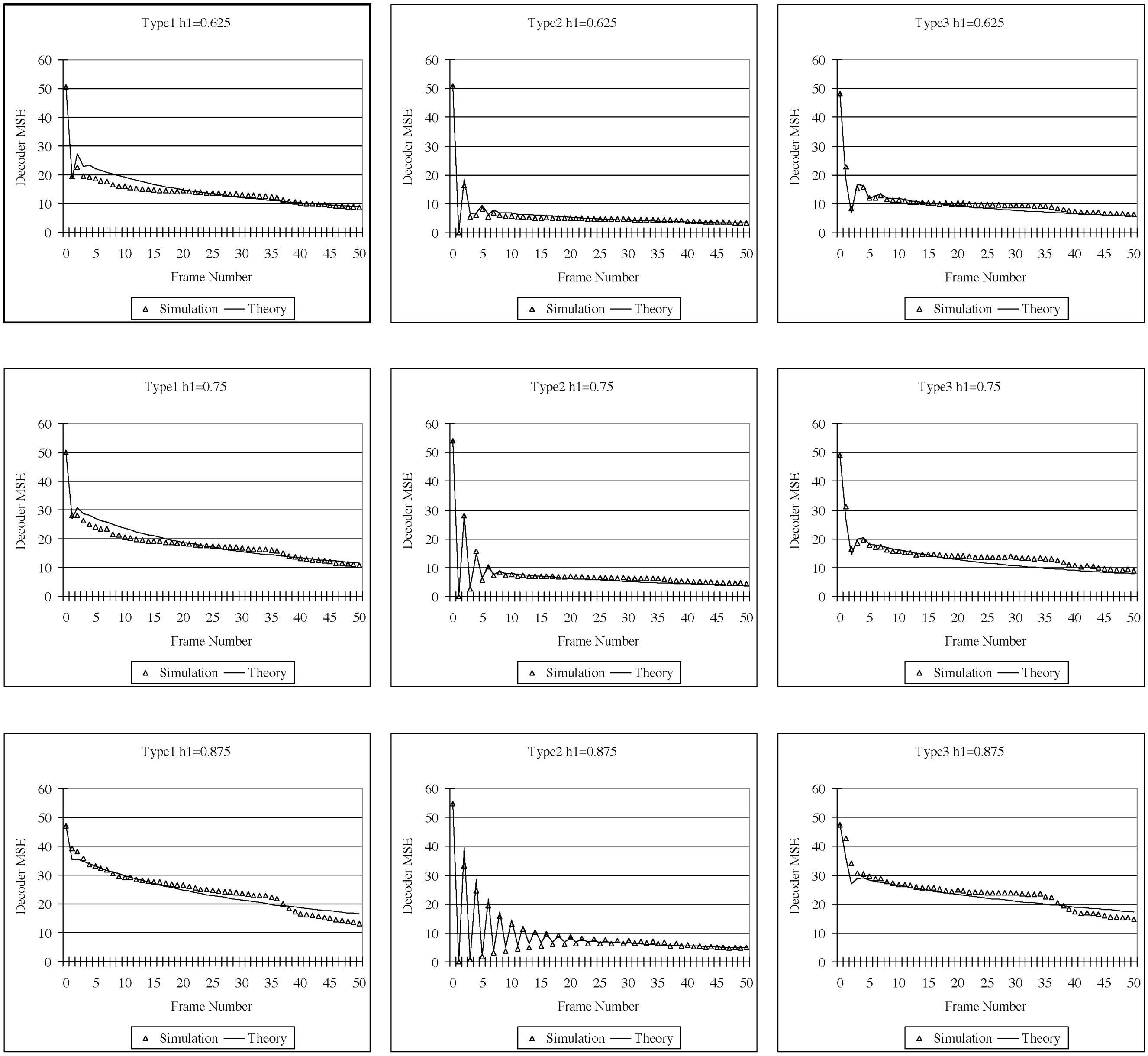


Fig. 2. To continue Fig. 1 for $h_1 \in [0.625, 0.875]$.

can roughly calculate the *Error Ratios* from the simulation data using $R_i = \sqrt{D_i(n_c)(1+\gamma n_c)/D_i(0)}$, $i = 1, 2, 3$. $D_i(0)$ and $D_i(n_c)$ are the decoder MSEs at time m_0 and $(m_0 + n_c)$, respectively. The obtained results are plotted in Fig. 3 with $\gamma = 0.035$ and $n_c = 50$. Theoretical *Error Ratios* calculated by (22) are also plotted, which are very close to those obtained from the simulation data. From (22) we can see that the *Error Ratios* of Type2 and Type3 are close for a small h_1 (i.e., $h_1 = 0.125$), while the *Error Ratios* of Type1 and Type3 are close for a large h_1 (i.e., $h_1 = 0.875$). In addition, increasing h_1 can make the *Error Ratio* larger. All these properties can be verified by the curves in Fig. 3.

A common characteristic of 2HMCP Type1, Type2, and Type3 is that the *Decoder Distortion* will decrease after some oscillations. To give a rough explanation for these phenomena, we suppose the state of each picture to be an element of set $\{G, B, M\}$, where G, B, and M represent good quality, bad quality, and median quality, respectively. Since the loss occurs at time m_0, picture $\psi(m_0)$ will be in state B and the past pictures are all in state G. Suppose the two hypotheses of picture $\psi(m)$ are $\tilde{\psi}_1(m)$ and $\tilde{\psi}_2(m)$, which may come from different reference pictures according to the predictors in (8), (9), or (10). If we know the states of $\tilde{\psi}_1(m)$ and $\tilde{\psi}_2(m)$, by using predictor $\hat{\psi}(m) = h_1\tilde{\psi}_1(m) + h_2\tilde{\psi}_2(m)$, we can roughly estimate the state of $\psi(m)$.

1) For small h_1 and large h_2, $\psi(m)$ would be in the same state as $\tilde{\psi}_2(m)$.
2) For large h_1 and small h_2, $\psi(m)$ would be in the same state as $\tilde{\psi}_1(m)$.
3) For similar h_1 and h_2, $\psi(m)$ would be in state M if $\tilde{\psi}_1(m)$ and $\tilde{\psi}_2(m)$ have different states; otherwise, $\psi(m)$ would be in the same state as $\tilde{\psi}_1(m)$.

TABLE III
THE DIFFERENCE BETWEEN THE THEORETICAL DISTORTION AND THE ACTUAL DECODER MSE

h_1	0.125	0.25	0.375	0.5	0.625	0.75	0.875
Type1 μm	0.34	−0.18	0.56	0.97	0.81	0.63	−0.34
Type1 σ^2	2.18	1.47	1.73	1.95	2.97	2.38	3.74
Type2 μm	0.46	0.22	0.48	0.30	0.25	−0.51	0.39
Type2 σ^2	1.26	0.57	0.43	0.32	0.37	0.36	1.48
Type3 μm	0.70	0.30	0.33	0.36	−0.73	−1.36	−0.88
Type3 σ^2	1.93	0.67	0.47	0.70	1.12	1.92	4.52

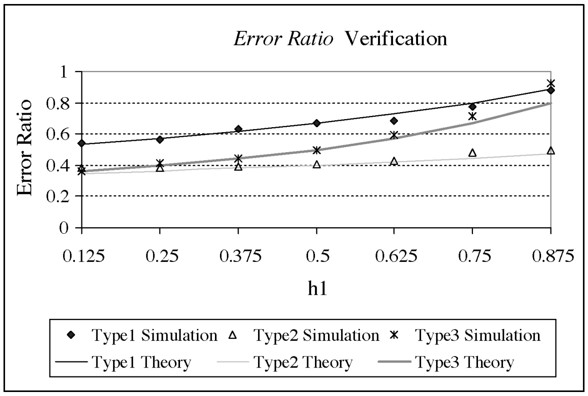


Fig. 3. Error ratio verification ($c = 1$).

Based on this, the states of the pictures from time m_0 can be obtained as:

Type1 with small h_1 : $BGBGBGB\ldots$
with large h_1 : $BBBBBBB\ldots$;
Type2 with small h_1 : $BGGBGGB\ldots$
with median h_1 : $BGMMMMM\ldots$
with large h_1 : $BGBGBGB\ldots$;
Type3 with small h_1 : $BGGBGGB\ldots$
with large h_1 : $BBBBBBB\ldots$.

An alternating occurrence of State B and State G leads to the oscillations in the propagated error, whereas a continual occurrence of the same state means stabilization. As illustrated in Figs. 1 and 2, for 2HMCP Type1 and Type3, when h_1 is larger, the oscillation is less severe, i.e., the *Error Transition Time* is shorter, but the converged error is larger. For 2HMCP Type2, the least severe oscillation occurs when h_1 is 0.625; the converged error is small for all the values of h_1. More illustrations for the relationship between the distortion and parameter h_1 can be found in Section III-C. As shown in each row of the figures, for a fixed h_1 and similar initial distortion $D(0)$, 2HMCP Type2 can always obtain a smaller distortion than Type1 and Type3 after the oscillations.

We also use some schemes to find the theoretical *Error Transition Time*, to give a clearer illustration for the oscillation effects. As stated previously, the loss occurs at time m_0 and $\epsilon(n)$ is the error at the $(m_0 + n)$th picture. Suppose N is a constant to specify a time interval with length $(2N + 1)$ and σ_n is the error variance in this period, i.e., $\sigma_n = \sum_{i=n-N}^{n+N}(\epsilon(i) - \mu_n)^2/(2N + 1)$ and $\mu_n = \sum_{i=n-N}^{n+N} \epsilon(i)/(2N + 1)$. Then we expect the *Error Transition Time* to be n if $\sigma_{n-1} > T_{\text{stable}}$ and $\sigma_n \leq T_{\text{stable}}$, where T_{stable} is a threshold. Based on (15), (18) and (20), the numerically obtained *Error Transition Time* is plotted in Fig. 4, where $c = 1$, $\epsilon(0) = 1$, $N = 2$ and $T_{\text{stable}} = 10^{-4}$. Similarly as the results in Figs. 1 and 2, a larger h_1 achieves a shorter *Error Transition Time* for 2HMCP Type1 and Type3, whereas the shortest *Error Transition Time* for 2HMCP Type2 is achieved when h_1 is around 0.625.

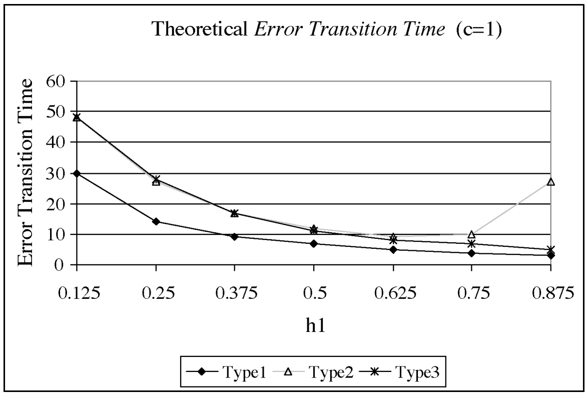


Fig. 4. Theoretical *Error Transition Time* ($c = 1$). The time is counted by the number of pictures.

C. Error Resilience Property

Suppose the compressed video is transmitted though a random packet loss channel and one packet contains the information of one picture. Then the loss of one packet will lead to the loss of one entire picture. Two error concealment algorithms are used to reconstruct the lost pictures.

1) *Copy*: The lost pictures are error concealed by copying the previous correctly decoded picture.
2) *OptiFlow*: The lost pictures are error concealed based on multiframe optical flow estimation [34]. In particular, the optical flow of the lost picture is estimated first, where the constant velocity model and median filter are used to provide both temporal and spatial regularization of the motion vector field. Then the previous picture is projected onto the lost one based on this estimated information. For more details please refer to [34].

PSNR is used as the objective video quality measurement, which is computed using the original (uncompressed) video as reference. Given a packet loss rate P, the video sequence is transmitted 300 times, and the average PSNR for the 300 transmissions is calculated at the decoder side.

In Fig. 5, we test the effect of weighting parameter h_1 on the performance of the three types of predictions, where h_1 goes from 0.125 to 0.875. *Carphone* (QCIF) and *Mobile* (CIF) are used as the testing sequences, compressed as in Table II. The packet loss rate is $P = 3\%$, and *OptiFlow* is used to error-conceal the lost pictures. We can observe from the figure that Type2 and Type3 perform similarly for a small value of h_1 (i.e., $h_1 = 0.125$), and Type1 and Type3 perform similarly for a large value of h_1 (i.e., $h_1 = 0.875$). This is consistent with the results in Fig. 3. In addition, decreasing h_1 can make the *Error Ratio* smaller, thus leading to a larger decoder PSNR. On the other hand, when h_1 decreases, the compression efficiencies of these prediction patterns may decrease accordingly, due to the smaller weighting parameter for the closer reference picture

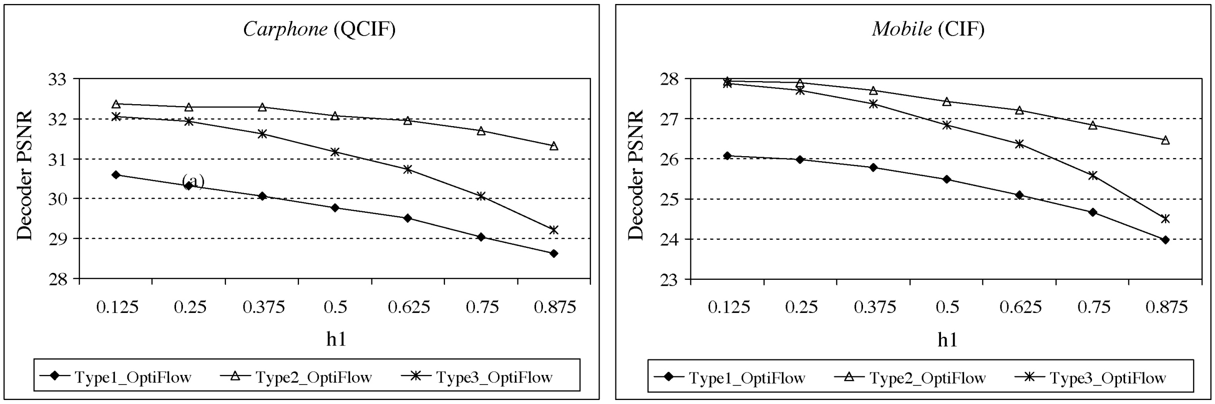


Fig. 5. Average PSNR at the decoder side with different weighting parameter h_1 (c = 1 and P = 3%).

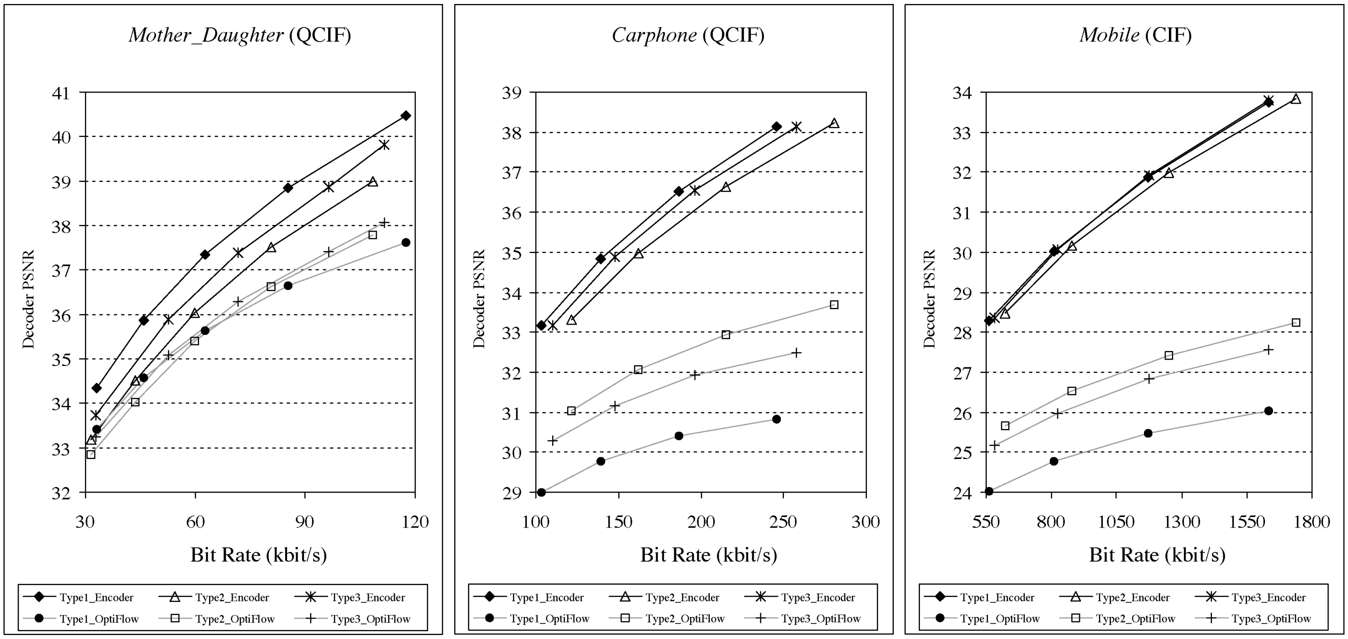


Fig. 6. R–D curves of the three prediction patterns (h_1 = 0.5, c = 1, and P = 3%).

(hypothesis). This can be observed from the results in Table II. For example, with the same set of QP values in Type1 of *Carphone*, when h_1 decreases from 0.875 to 0.5 and from 0.5 to 0.125, the bit rate increases about 6% and 9%, respectively. To make a balance between the compression efficiency and error resilience capability, generally a moderate value of h_1 (e.g., h_1 = 0.5) can be used. One exception is the *Mobile* sequence, in which h_1 = 0.375 and h_1 = 0.625 tend to have a better compression efficiency than the other values of h_1, as discussed previously. In addition, we can observe from Fig. 5 that the prediction with h_1 = 0.375 also has a good error resilience ability. Using Type1 prediction as an example, the decoder PSNR with h_1 = 0.375 is 1.8 dB higher than that with h_1 = 0.875, but only 0.3 dB lower than h_1 = 0.125. This indicates that h_1 = 0.375 is a good choice for *Mobile*. For the sake of simplicity, we will just use h_1 = 0.5 for all the testing sequences in the remaining simulations.

Fig. 6 shows the *Rate–Distortion* (R–D) curves of the three prediction patterns under packet loss rate P = 3%, h_1 = 0.5, and c = 1. *OptiFlow* is used to error-conceal the lost pictures. The R–D curves at the encoder side are also plotted to give a comparison between the compression efficiencies. From the figure we can see that for sequences *Carphone* and *Mobile*, the decoder R–D curve of Type2 always lies above those of Type1 and Type3, and Type1 lies at the bottom. In addition, the gain of Type3 over Type1 is much obvious than that of Type2 over Type3. Note that in *Mobile*, the encoder R–D curves of Type1 and Type3 almost overlap with each other, whereas the decoder R–D curve of Type3 is more than 1dB higher than that of Type1. For a sequence with relatively simple and small motions, or with a large static background, such as *Mother_Daughter*, the decoder R–D curve of Type1 is higher than those of Type2 and Type3 at a low bit rate. When the bit rate increases, the curve of Type3 becomes higher than those of Type1 and Type2. The reason is that when packet loss occurs, error concealment can work well in such sequences and the propagated error is small in all these three prediction types. As a result, the advantage of Type3

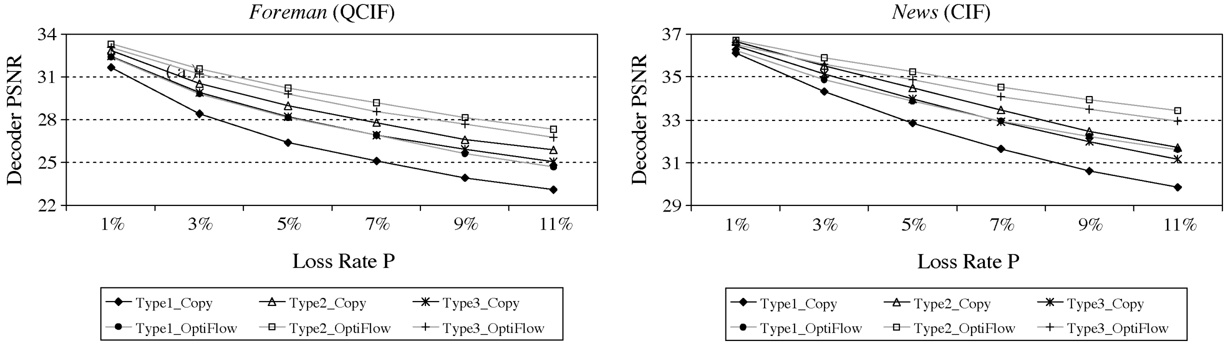


Fig. 7. Average PSNR at the decoder side with different loss rate P ($h_1 = 0.5$ and $c = 1$).

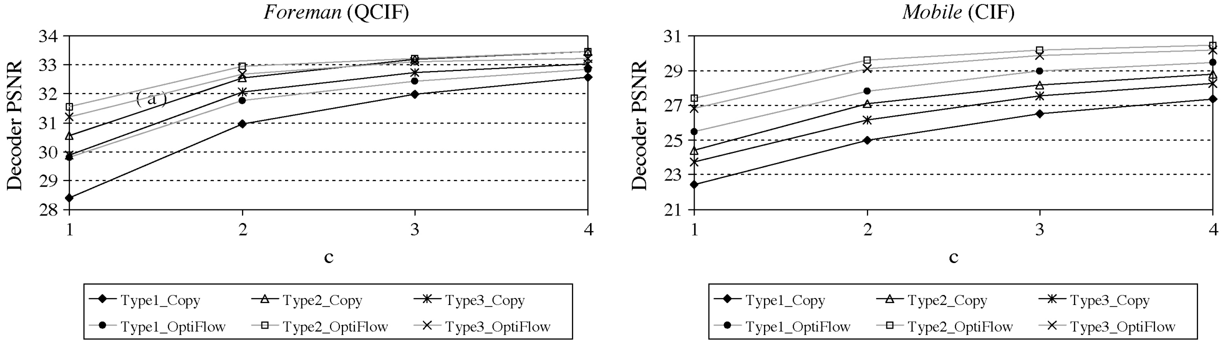


Fig. 8. Average PSNR at the decoder side with different distance parameter c ($h_1 = 0.5$ and $P = 3\%$).

(Type2) over Type1 (Type3), in terms of the error suppression ability, is not obvious in such situations, but the corresponding bit rate is much larger.

Fig. 7 compares the performance of the three prediction patterns under different packet loss rate, $P = 1\% \sim 11\%$. The testing sequences are *Foreman* (QCIF) and *News* (CIF), compressed as in Table II with $h_1 = 0.5$ and $c = 1$. From the figure we can see that no matter whether *Copy* or *OptiFlow* is used to error-conceal the lost pictures, the curve of 2HMCP Type2 always lies above Type3, and Type3 lies above Type1. This is consistent with the previous analysis that $R_2 < R_3 < R_1$ for $0 < h_1 < 1$. In addition, as the loss rate increases, the PSNR differences between the three prediction patterns increase accordingly. For example, when P goes from 1% to 11% in *Foreman* with *OptiFlow*, the gain of Type3 over Type1 increases from 0.71 dB to 2.07 dB, and the gain of Type2 over Type3 increases from 0.22 dB to 0.56 dB. Note that as shown previously in Table II, with $h_1 = 0.5$ and the same set of QPs in *Foreman*, the bit rate increases about 10.7% from Type1 to Type3 and 9% from Type3 to Type2. This indicates that with a similar percentage of bit rate increase, the gain of Type3 over Type1 is much larger than that of Type2 over Type3.

In Fig. 8, we investigate the effect of distance parameter c on the performance of 2HMCP. Sequences *Foreman* (QCIF) and *Mobile* (CIF) are encoded with $h_1 = 0.5$ and fix QPs (28 for I picture and 30 for P, B pictures), and then transmitted with packet loss rate $P = 3\%$. At the encoder side, the PSNRs for *Foreman* and *Mobile* are around 34.5 dB and 32 dB, respectively. When c increases from 1 to 2, the bit rate increases a lot, e.g., about 29% for 2HMCP Type3 of *Foreman* and 17% for *Mobile*. For a larger increment of c, the increase in bit rate is smaller, e.g., around 9% for *Foreman* when c goes from 3 to 4. Although increasing c leads to a larger bit rate, it effectively helps to suppress the propagated error, as shown in the figure. For example, when c goes from 1 to 2 for 2HMCP Type3 of *Mobile* with *OptiFlow*, a 17% redundancy in the bit rate can give us about 2.28dB gain at the decoder side. When c goes above 2, the increment in decoder PSNR is not as large as that when c goes from 1 to 2. As a result, it is reasonable to use a small c, i.e., $c = 1, 2$, to make a balance between the compression efficiency and error resilience capability.

In previous simulations, the encoded video stream is supposed to be transmitted through a random packet loss channel. In Table IV, we compare the performance of the three prediction patterns under the condition of burst errors. The two-state Markov chain model has been employed to simulate packet losses. The average burst length L_b (in terms of consecutive packet losses) goes from 1 to 5, and the packet loss rate is $P = 5\%$ and $P = 20\%$. *Foreman* (QCIF) and *News* (CIF) are used as the testing sequences, encoded with $h_1 = 0.5$ and fix QPs (28 for I picture and 30 for P, B pictures), $c = 1, 2, 3$. *Copy* is used to error-conceal the lost pictures. We can observe from the table that when the burst length increases, the PSNR gain of 2HMCP Type2 over Type3 will decrease on the whole, and Type2 may even become worse than Type3 with $c = 1$ and a large L_b. This is different from the previous results for random packet losses. As unlike prediction Type3, the references of Type1 and Type2 are two consecutive pictures when c equals 1, it is more likely that both of these reference pictures are lost when L_b increases. We can also observe from the table that the decoder PSNR usually decreases with increasing L_b, as expected. However, this is not always true, especially for $c = 1$. One reason is that with $c = 1$ and a large L_b, when a picture is received after a burst error, all its previous pictures

TABLE IV
THE AVERAGE DECODER PSNR FOR THE CONDITION OF BURST ERRORS

		P = 5%					P = 20%				
	L_b	1	2	3	4	5	1	2	3	4	5
Foreman (QCIF) c = 1	Type1	26.64	26.03	26.11	26.43	26.78	20.88	19.70	19.55	19.67	19.75
	Type2	29.30	27.48	27.00	27.13	27.33	23.93	21.22	20.62	20.46	20.39
	Type3	28.40	27.41	27.11	27.22	27.52	22.44	21.18	20.68	20.61	20.53
Foreman (QCIF) c = 2	Type1	29.69	28.32	27.97	27.89	28.23	24.29	22.02	21.24	21.07	20.92
	Type2	31.68	30.27	29.52	29.28	29.38	26.72	24.24	23.02	22.66	22.28
	Type3	31.07	29.78	29.17	29.03	29.05	26.05	23.72	22.83	22.51	22.21
Foreman (QCIF) c = 3	Type1	30.85	29.61	28.98	28.78	28.84	25.95	23.61	22.53	22.22	21.91
	Type2	32.52	31.49	30.70	30.38	30.32	28.64	26.21	24.84	24.27	23.76
	Type3	31.89	30.81	30.22	29.93	29.92	27.53	25.18	24.11	23.72	23.35
News (CIF) c = 1	Type1	32.97	32.66	32.55	32.94	33.08	27.58	26.88	26.80	27.17	27.19
	Type2	34.71	33.59	33.04	33.33	33.33	29.74	27.95	27.55	27.82	27.70
	Type3	34.12	33.51	33.08	33.29	33.29	28.72	27.85	27.54	27.81	27.73
News (CIF) c = 2	Type1	34.85	33.85	33.29	33.42	33.42	30.30	28.38	27.79	27.87	27.70
	Type2	35.95	35.05	34.24	34.14	34.02	32.38	30.13	29.18	28.92	28.52
	Type3	35.57	34.72	34.07	34.03	33.94	31.62	29.62	28.99	28.87	28.57
News (CIF) c = 3	Type1	35.54	34.63	33.94	33.87	33.80	31.58	29.53	28.68	28.57	28.15
	Type2	36.25	35.67	35.07	34.86	34.69	33.26	31.52	30.65	30.23	29.76
	Type3	35.98	35.34	34.84	34.72	34.62	32.56	30.86	30.27	30.08	29.72

have been seriously deteriorated. As the error in the decoded picture is very large, the reconstructed video quality can be equally bad for a small variation of L_b. On the other hand, with the same loss rate P, the frequency of burst errors will decrease when L_b increases. As a result, the average decoder PSNR can become larger with a larger L_b. Similar to the results in Fig. 8, increasing c can help to improve the reconstructed video quality, but the improvement from $c = 2$ to $c = 3$ is not as obvious as that from $c = 1$ to $c = 2$, especially for a small value of L_b. For a sequence with small motions such as *News*, 2HMCP with $c = 1$ is enough to suppress the propagated errors. For *Foreman*, $c = 1$ or $c = 2$ can be used, depending on the available bit rate budget.

IV. CONCLUSION

In this paper, we extend the two-hypothesis motion compensated prediction in [12] to three more generalized ones, which are H.264/AVC standard compatible and can be implemented by the generalized B pictures in H.264/AVC [19]. The error propagation models of these prediction patterns are derived for the case of a single picture loss. In addition to this, the convergence time of the propagated error and its oscillation effect are also discussed. Simulation results are given to compare both the compression efficiencies and error resilience abilities of these three prediction patterns.

REFERENCES

[1] Y. Wang and Q. F. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[2] 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, no. 4, pp. 61–82, Jul. 2000.

[3] M. Ghandi, B. Barmada, E. Jones, and M. Ghanbari, "Unequally error protected data partitioned video with combined hierarchical modulation and channel coding," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 2. May 2006, pp. II-529–II-532.

[4] C.-M. Fu, W.-L. Hwang, and C.-L. Huang, "Efficient post-compression error-resilient 3-D-scalable video transmission for packet erasure channels," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 2. Mar. 2005, pp. 305–308.

[5] M. Stoufs, A. Munteanu, P. Schelkens, and J. Cornelis, "Optimal joint source-channel coding using unequal error protection for the scalable extension of H.264/MPEG-4 AVC," in *Proc. IEEE Int. Conf. Image Process.*, vol. 6. Oct. 2007, pp. IV-517–IV-520.

[6] M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Isolated regions in video coding," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 259–267, Apr. 2004.

[7] P. Baccichet, S. Rane, A. Chimienti, and B. Girod, "Robust low-delay video transmission using H.264/AVC redundant slices and flexible macroblock ordering," in *Proc. IEEE Int. Conf. Image Process.*, San Antonio, TX, vol. 4. Oct. 2007, pp. IV-93–IV-96.

[8] R. Aravind, M. Civanlar, and A. Reibman, "Packet loss resilience of MPEG-2 scalable video coding algorithms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 5, pp. 426–435, Oct. 1996.

[9] Y. Guo, Y.-K. Wang, and H. Li, "Error resilient mode decision in scalable video coding," in *Proc. IEEE Int. Conf. Image Process.*, Atlanta, GA, Oct. 2006, pp. 2225–2228.

[10] F. Verdicchio, A. Munteanu, A. Gavrilescu, J. Cornelis, and P. Schelkens, "Embedded multiple description coding of video," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3114–3130, Oct. 2006.

[11] E. Akyol, A. Tekalp, and M. Civanlar, "A flexible multiple description coding framework for adaptive peer-to-peer video streaming," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 2, pp. 231–245, Aug. 2007.

[12] S. Lin and Y. Wang, "Error resilience property of multihypothesis motion-compensated prediction," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3. Jun. 2002, pp. 545–548.

[13] Y.-C. Tsai and C.-W. Lin, "H.264 error resilience coding based on multi-hypothesis motion-compensated prediction," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2005, pp. 952–955.

[14] W.-Y. Kung, C.-S. Kim, and C.-C. Kuo, "Analysis of multihypothesis motion-compensated prediction (MHMCP) for robust visual communication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 146–153, Jan. 2006.

[15] B. Girod, "Efficiency analysis of multihypothesis motion-compensated prediction for video coding," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 173–183, Feb. 2000.

[16] M. Flierl, T. Wiegand, and B. Girod, "Rate-constrained multi-hypothesis motion-compensated prediction for video coding," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2000, pp. 150–153.

[17] T. Wiegand, G. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[18] Y. Wang, J. Ostermann, and Y.-Q. Zhang, "Video coding standards," in *Video Processing and Communications*. Englewood Cliffs, NJ: Prentice-Hall, 2001, ch. 13.

[19] M. Flierl and B. Girod, "Generalized B pictures and the draft H.264/AVC video-compression standard," *IEEE Trans. Circuits Syst.*

Video Technol., vol. 13, no. 7, pp. 587–597, Jul. 2003.

[20] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 70–84, Feb. 1999.

[21] M. Budagavi and J. Gibson, "Multiframe video coding for improved performance over wireless channels," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 252–265, Feb. 2001.

[22] A. Leontaris and P. Cosman, "Video compression for lossy packet networks with mode switching and a dual-frame buffer," *IEEE Trans. Image Process.*, vol. 13, no. 7, pp. 885–897, Jul. 2004.

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

[24] C.-S. Kim, J. Kim, I. Katsavounidis, and C.-C. Kuo, "Robust MMSE video decoding: Theory and practical implementations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 39–51, Jan. 2005.

[25] Y. Wang, Z. Wu, and J. Boyce, "Modeling of transmission-loss-induced distortion in decoded video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 6, pp. 716–732, Jun. 2006.

[26] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Proc. Vis. Commun. Image Process.*, Jan. 2001, pp. 392–409.

[27] N. Farber, K. Stuhlmuller, and B. Girod, "Analysis of error propagation in hybrid video coding with application to error resilience," in *Proc. IEEE Int. Conf. Image Process.*, vol 2. Oct. 1999, Kobe, Japan, pp. 550–554.

[28] B. Girod and N. Farber, "Wireless video," in *Compressed Video Over Networks*, M.-T. Sun and A. R. Reibman, Eds. New York: Marcel Dekker, 2000.

[29] Fibonacci number. [Online]. Available: http://en.wikipedia.org/wiki/Fibonacci_number

[30] Fibonacci number. [Online]. Available: http://mathworld.wolfram.com/FibonacciNumber.html

[31] Fibonacci numbers spelled out. [Online]. Available: http://ulcar.uml.edu/∼iag/CS/Fibonacci.html

[32] JVT reference software, version 10.2. [Online]. Available: http://iphome.hhi.de/suehring/tml/download/

[33] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, ITU-T SG16 Doc. VCEG-M33, Apr. 2001, pp. 1–4.

[34] 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.

Mengyao Ma (S'05) received the B.Sc. degree in computer science and technology from Peking University, Beijing, China, in 2003, and the Ph.D. degree in computer science and engineering from the Multimedia Technology Research Center, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China, in 2009.

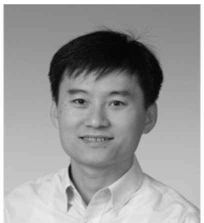
Currently, she is with Hong Kong Applied Science and Technology Research Institute Company Limited as a Senior Engineer. Her research interests include error resilient video compression, error propagation analyses, and error concealment of video streams over packet loss channels.

Dr. Ma was a recipient of the Best Paper Awards from the IEEE International Workshop on Signal Processing Systems, in 2007, and PACIFIC-RIM Conference on Multimedia, in 2007.

Oscar C. Au (S'87–M'90–SM'01) received the B.A.Sc. degree from the University of Toronto, Toronto, Ontario, 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 University for one year, he joined the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China, in 1992. He is now an Associate Professor, Director of Multimedia Technology Research Center, and Advisor of the Computer Engineering (CPEG) Program in Hong Kong University of Science and Technology. His main research contributions are on video and image coding and processing, watermarking and light weight encryption, speech and audio processing. His research interests 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, postprocessing, multiview coding, scalable video coding, distributed video coding, subpixel rendering, JPEG/JPEG2000 and halftone image data hiding, etc. He is the author of about 200 technical journal and conference papers published. 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 SYSTEM, Part 1 (TCAS1), the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and IEEE TRANSACTIONS ON IMAGE PROCESSING. He is the Chairman of the Technical Committee (TC) on Multimedia Systems and Applications and a Member of the TC on Video Signal Processing and Communications and the TC on DSP of the IEEE Circuits and Systems Society. He served on the Steering Committee of IEEE TRANSACTIONS ON MULTIMEDIA and the IEEE International Conference of Multimedia and Expo. He also served on the organizing committee of the IEEE International Symposium on Circuits and Systems, in 1997, the IEEE International Conference on Acoustics, Speech and Signal Processing, in 2003, the ISO/IEC MPEG 71st Meeting, in 2004, International Conference on Image Processing, in 2010, and other conferences.



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, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China, 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 Nippon Electric Company Research Institute, Princeton, NJ. His research interests include multimedia networking, peer-to-peer technologies and streaming, and wireless communication networks. He served as the Vice-Chair of the IEEE Communications Society Multimedia Communications Technical Committee from 2003 to 2006.

Dr. Chan is the 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 Association for Computing Machinery Multimedia Conference in 2005, and the Multimedia Symposia for IEEE Global Communications Conference, in 2006, and IEEE Communications Society, in 2005 and 2007. He 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.

Liwei Guo (S'05) received the B.E degree from the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, in 2004, and the Ph.D. degree from the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China, in 2008.

He is currently a Postdoctoral Researcher at Thomson Corporate Research, Princeton, NJ. His research interests include image/video processing, video compression, and signal estimation.

Dr. Guo is a recipient of the Best Paper Awards from the IEEE International Workshop on Signal Processing Systems in 2007 and PACIfIC-RIM Conference on Multimedia in 2007. He is a Member of Sigma Xi.