Title

Video coding with fixed-length packetization for a tandem channel

Journal

IEEE Transactions on Image Processing, 15(2)

ISSN

1057-7149

Authors

Shen, Yushi
Cosman, P. C.
Milstein, L. B.

Publication Date

2006-02-01

DOI

10.1109/TIP.2005.860598

Peer reviewed

Video Coding With Fixed-Length Packetization for a Tandem Channel

Yushi Shen, Pamela C. Cosman, *Senior Member, IEEE*, and Laurence B. Milstein, *Fellow, IEEE*

***Abstract*—A robust scheme is presented for the efficient transmission of packet video over a tandem wireless Internet channel. This channel is assumed to have bit errors (due to noise and fading on the wireless portion of the channel) and packet erasures (due to congestion on the wired portion). First, we propose an algorithm to optimally switch between intracoding and intercoding for a video coder that operates on a packet-switched network with fixed-length packets. Different re-synchronization schemes are considered and compared. This optimal mode selection algorithm is integrated with an efficient channel encoder, a cyclic redundancy check outer coder concatenated with an inner rate-compatible punctured convolutional coder. The system performance is both analyzed and simulated. Last, the framework is extended to operate on a time-varying wireless Internet channel with feedback information from the receiver. Both instantaneous feedback and delayed feedback are evaluated, and an improved method of refined distortion estimation for encoding is presented and simulated for the case of delayed feedback.**

***Index Terms*—Mode switching, packet-switched networks, tandem channel, video compression, wireless internet.**

I. Introduction

PACKET video is becoming a significant portion of traffic over wireless and wireline networks. However, network congestion and wireless channel errors can yield tremendous packet loss and, thus, degrade the video quality. The transmitted bitstream should be organized to minimize the possible corruption and error propagation.

Motion compensation, or intercoding, is a basic and efficient approach for video coding. However, it may suffer from potentially severe error propagation, because a single error in a frame may corrupt all subsequent frames if intercoding is used repeatedly. Intracoding, by encoding the current macroblock (MB) by itself, can stop error propagation, but this mode is usually much more costly in bits than intercoding. Thus, it is desired to switch between intra- and intercoding intelligently according to channel conditions, to achieve the right balance between compression efficiency and robustness.

We are interested in using fixed-length packets over tandem channels, whereby we mean a channel that has both wireline and wireless links, and so experiences both packet erasures due to congestion on the wireline component, and bit errors due to noise and fading on the wireless component of the link. Video communications over tandem channels has been addressed in references such as [1]–[4].

The ROPE algorithm for inter/intramode selection was proposed in [5]; it used variable length packets and was designed for a packet erasure channel whose loss rate is fixed and known. Our work uses distortion estimation and mode switching in the style of the ROPE algorithm, but for more complex channels, so significant modifications are needed.

This paper is organized as follows. In Section II, we derive a modified ROPE algorithm for fixed-length packets with two different re-synchronization approaches. Both analysis and simulation results suggest that the performance of fixed-length packets is worse than that of variable-length packets. We also compare different re-synchronization approaches. In Section III, we study video coding over a constant tandem channel with both bit errors and packet erasures. By means of a well-designed concatenated channel coder, the tandem channel can dynamically be treated as a simple erasure channel by the source encoder; thus, the modified ROPE algorithm can be used. In Section IV, we extend our framework to the scenario where the channel has time-correlated variation, and a feedback channel is used to tell the encoder about the channel status. The performance is evaluated with both instantaneous and delayed feedback information. Conclusions are drawn in Section V.

II. Optimal Mode Switching With Fixed-Length Packets

In video compression, typically each frame is segmented into MBs of size 16×16 pixels. One horizontal row or slice of MBs is called a group of blocks (GOB). The encoding mode and the quantization step are selected for each MB individually in DCT-based video encoders such as MPEG-2 and MPEG-4. In a packetized transmission system, the compressed bit stream is then sent by either variable-length or fixed-length packets.

For variable-length packets, each GOB can be carried in a separate packet; a short packet header says which GOB is in the packet. One packet loss entails loss of the whole GOB, without affecting decoding of other packets (GOBs). The loss rate of a pixel equals the packet erasure rate.

For fixed-length packets, packet boundaries are rarely GOB or MB boundaries. Thus, when one packet is lost, the decoder will be unable to interpret the start of the next one. We refer to this as loss of synchronization. As packet loss causes bits in the next (and perhaps subsequent) packets to be lost, the loss rate of pixels exceeds the packet erasure rate due to loss

Manuscript received March 2, 2004; revised February 7, 2005. This work was supported in part by the California Institute for Telecommunications and Information Technology, in part by Ericsson, Inc., in part by the State of California under the CoRe program, and in part by the Office of Naval Research under Grant N00014-03-1-0280. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. David S. Taubman.

The authors are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407 USA (e-mail: yushen@code.ucsd.edu; yushen@ucsd.edu; pcosman@ucsd.edu; lmilstein@ucsd.edu).

Digital Object Identifier 10.1109/TIP.2005.860598

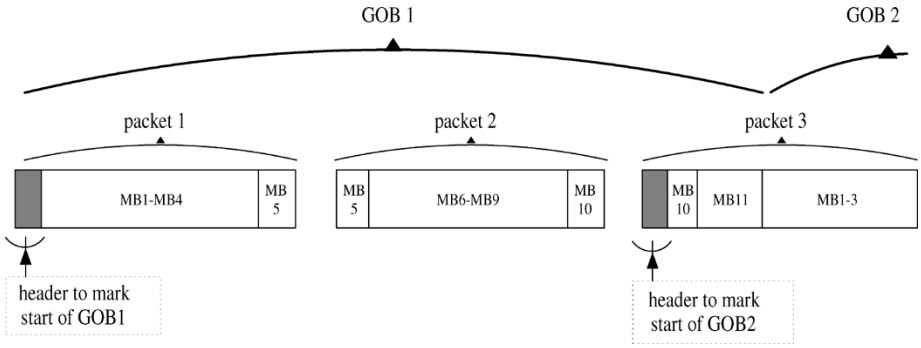


Fig. 1. Example of re-sync per GOB.

of synchronization. We propose two methods to efficiently re-synchronize. re-synchronization once per GOB, and once per packet.

In this section, we lay the groundwork for the tandem wireline/wireless channel to be presented in Section III. For ease of combining source coding with channel coding, our scheme will employ fixed-length packets. Since most previous work in this general area has been done with variable-length packets, and since, as will become obvious by the end of this section, fixed-length packets do not perform as well as do variable-length packets, we use this section to describe in detail the fixed-length packet system, and to compare its performance to that of a variable-packet scheme.

More specifically, in this section, we concentrate on the performance of a system employing fixed-length packets over an erasure channel, where the erasure rate is constant and known by the encoder. This model will be used to represent the wireline component of the tandem channel. In Section III, we will add the wireless component, and this latter component will be modeled as the concatenation of an inner RCPC coder and an outer error detection code. Thus, it, too, will function as a packet erasure channel to the source encoder.

A. Encoding With Re-Synchronization per GOB

This method inserts re-synchronizing bits at the beginning of each GOB. Video in QCIF format contains 9×11 MBs, so there are 9 GOBs per frame. With a frame rate of 30 frames per second, and bit rate of 450 kbps, each GOB occupies about 450 k/(30×9) bits on the average, or 1667 bits. Therefore, for packet sizes in the range of 400 to 800 bits, usually the bits corresponding to one GOB will be split into several packets.

We use the first bit of each packet to tell whether there is a new GOB in this packet. If there is, the next 9 or 10 bits (depending on the packet length) indicate the new GOBs starting location. The frame/GOB number follows. In this case, an MB will not be reconstructable at the decoder if either the packet containing this MB is lost, or any of the former MBs in the same GOB are lost. If any of the former MBs are lost, the decoder will lose synchronization until the next re-sync information is received; thus, the remaining MBs of the current GOB will be unreconstructable even if the decoder receives the following packets. It is possible, although unlikely, for the compressed bit stream of one MB to extend over several packets. For simplicity, we assume the decoder loses the whole MB if any one of these packets is lost.

We count the packet number from the first packet of each GOB. Assume the current MB extends to packet m of this GOB. The probability that this MB can be reconstructed at the decoder is the probability that all m packets of this GOB are received by the decoder. This equals $(1-p)^m$, where p is the packet erasure rate. If $P_{\bar{R}}$ denotes the probability that an MB cannot be reconstructed at the decoder, we have $P_{\bar{R}} = P_{\bar{R}}(m) = 1-(1-p)^m$. For example, in Fig. 1, for GOB1, $m = 1$ for MB1 to MB4, $m = 2$ for MB5 to MB9, and $m = 3$ for MB10 and MB11. For MB10, we have $P_{\bar{R}} = 1-(1-p)^3$.

When an MB is lost, the decoder uses a temporal concealment method. The three nearest MBs above the lost MB are denoted A, B, C from left to right. Their motion vectors (MVs) define the substitute motion vector (SMV), where the SMV indicates which MB in the previous frame will be used for concealment. We assume, if any of A, B, and C were intracoded, that its $\mathrm{MV} = (0,0)$. First, if MB A is lost, then so are B and C, and we set $\mathrm{SMV} = (0,0)$. If the decoder knows A, but not B and C, we set the SMV equal to the MV of A. If both A and B survive, but not C, we set the SMV equal to the MV of B. Last, if the decoder has all of A, B and C, we set the SMV equal to their median MV. When the current MB belongs to the top GOB of this frame, we set $\mathrm{SMV} = (0,0)$, and if the lost MB is on the side of the frame, we use the MV of the MB directly above.

We are ready to derive the expected decoder distortion per pixel for this case. Using the notation from [5], f_n denotes original frame n, which is compressed and reconstructed at the encoder as $\hat{f}_n$ (only quantization error is considered). The (possibly error-concealed) reconstruction at the receiver is denoted by $\tilde{f}_n$ (including quantization error, error propagation, packet loss and concealment distortion). The encoder does not know $\tilde{f}_n$, and treats it as a random variable.

Let f_n^i denote the original value of pixel i in frame n, and let $\hat{f}_n^i$ denote its encoder reconstruction. The reconstructed value at the decoder, possibly after error concealment, is denoted by $\tilde{f}_n^i$. The expected distortion for pixel i is

$$d_n^i = E\{(f_n^i - \tilde{f}_n^i)^2\} = (f_n^i)^2 - 2f_n^i E\{\tilde{f}_n^i\} + E\{(\tilde{f}_n^i)^2\}. \quad (1)$$

Calculation of d_n^i requires the first and second moments of the random variable of the estimated image sequence $\tilde{f}_n^i$. To compute these, recursion functions are developed in [5], in which it

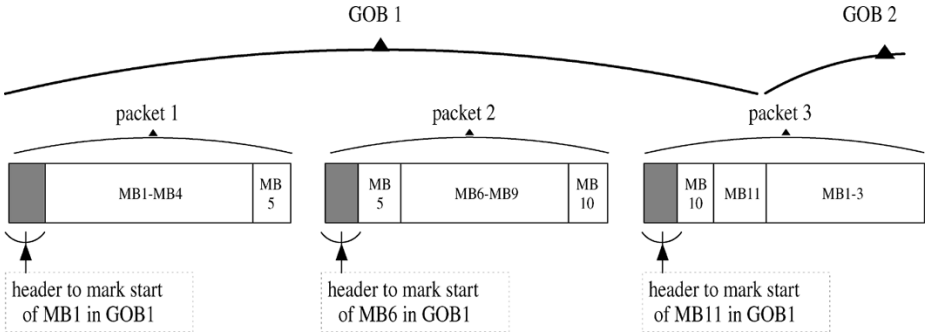


Fig. 2. Example of re-sync per packet.

is necessary to separate out the cases of intra- and intercoded MBs. Here, since we use a modified pixel loss rate and a modified concealment method for fixed-length packets, the recursion formulas must be modified.

For each MB and for each mode selection and quantization step, we determine the packet number m for the current MB and $P_{\bar{R}} = 1-(1-p)^m$. A, B, C are the three nearest MBs above this MB from left to right. We define some probabilities as follows: $P_{\bar{A}} = Pr(A \text{ lost})$, and $P_A = Pr(A \text{ received}) = 1 - P_{\bar{A}}$, where "lost" means not reconstructable at the decoder and "received" means reconstructable. We also define $P_{\bar{B}|A} = Pr(B \text{ lost} \mid A \text{ received})$, and $P_{\bar{C}|AB} = Pr(C \text{ lost} \mid A \text{ received } and\, B \text{ received})$. Similarly, $P_{A\bar{B}} = Pr(A \text{ received } and\, B \text{ lost}) = P_A P_{\bar{B}|A}$, $P_{\mathrm{ABC}} = P_A(1-P_{\bar{B}|A})(1-P_{\bar{C}|AB})$, and $P_{\mathrm{AB\bar{C}}} = P_A(1-P_{\bar{B}|A})P_{\bar{C}|AB}$. We obtain

$$P_{\bar{A}} = 1-(1-p)^{m_A} \tag{2}$$

$$P_{\bar{B}|A} = 1-(1-p)^{l_B} \tag{3}$$

$$P_{\bar{C}|AB} = P_{\bar{C}|B} = 1-(1-p)^{l_C} \tag{4}$$

where m_A is the number of packets that A spans from the beginning of its GOB, l_B is the number of packets that B spans beyond the end of the packet with A, and l_C is the number of packets that C spans beyond the end of the packet with B. Note that since we assume p is known at the encoder, the probabilities required in (2)–(4) will be computed and stored at the time the MBs are encoded.

Let $k1$, $k2$, and $k3$ correspond to the pixels in the previous frame that are used to conceal pixel i, using the MV of A, B and C, respectively, and let $k4$ correspond to the pixel for concealment using the median of the MVs of these three MBs. For an intracoded MB, $\tilde{f}_n^i = \hat{f}_n^i$ with probability $1-P_{\bar{R}}$. If the current packet is lost, and if A is also lost (with probability $P_{\bar{A}}$), so are B and C, then $\tilde{f}_n^i = \hat{f}_{n-1}^i$ because the SMV is set to (0, 0). Given A is received (with probability $1-P_{\bar{A}}$), if B is lost and so is C, then $\tilde{f}_n^i = \hat{f}_{n-1}^{k1}$; if B is received but C is lost, then $\tilde{f}_n^i = \hat{f}_{n-1}^{k2}$; last, if both B and C are received, $\tilde{f}_n^i = \hat{f}_{n-1}^{k4}$. Thus, the two moments for a pixel in an intracoded MB are given by

$$E\{\tilde{f}_n^i\} = (1-P_{\bar{R}})\hat{f}_n^i + P_{\bar{R}}\left(P_{\bar{A}}E\{\tilde{f}_{n-1}^i\} + P_{A\bar{B}}E\{\tilde{f}_{n-1}^{k1}\} + P_{\mathrm{AB\bar{C}}}E\{\tilde{f}_{n-1}^{k2}\} + P_{\mathrm{ABC}}E\{\tilde{f}_{n-1}^{k4}\}\right) \tag{5}$$

$$E\{(\tilde{f}_n^i)^2\} = (1-P_{\bar{R}})(\hat{f}_n^i)^2 + P_{\bar{R}}\left(P_{\bar{A}}E\{(\tilde{f}_{n-1}^i)^2\} + P_{A\bar{B}}E\{(\tilde{f}_{n-1}^{k1})^2\} + P_{\mathrm{AB\bar{C}}}E\{(\tilde{f}_{n-1}^{k2})^2\} + P_{\mathrm{ABC}}E\{(\tilde{f}_{n-1}^{k4})^2\}\right). \tag{6}$$

For an intercoded MB, assume the true MV of current pixel i is predicted from pixel j in the previous frame. Thus, the encoder prediction of this pixel is $\hat{f}_{n-1}^j$. The prediction error e_n^i is compressed and the quantized residue is $\hat{e}_n^i$. So, the encoder reconstruction is $\hat{f}_n^i = \hat{f}_{n-1}^j + \hat{e}_n^i$. The encoder transmits $\hat{e}_n^i$ and the MV. If received, the decoder knows $\hat{e}_n^i$ and the MV, but must use its own reconstruction of pixel j in the previous frame $\tilde{f}_{n-1}^j$, which may differ from the encoder value $\hat{f}_{n-1}^j$. Thus, the decoder reconstruction of pixel i is given by $\tilde{f}_n^i = \tilde{f}_{n-1}^j + \hat{e}_n^i$. The moments of $\tilde{f}_n^i$ for a pixel in an intercoded MB are given by

$$\begin{aligned} &E\{\tilde{f}_n^i\} \\ &= (1-P_{\bar{R}})\left(\hat{e}_n^i + E\{\tilde{f}_{n-1}^j\}\right) + P_{\bar{R}}\left(P_{\bar{A}}E\{\tilde{f}_{n-1}^i\} + P_{A\bar{B}}E\{\tilde{f}_{n-1}^{k1}\} + P_{\mathrm{AB\bar{C}}}E\{\tilde{f}_{n-1}^{k2}\} + P_{\mathrm{ABC}}E\{\tilde{f}_{n-1}^{k4}\}\right) \end{aligned} \tag{7}$$

$$\begin{aligned} &E\{(\tilde{f}_n^i)^2\} \\ &= (1-P_{\bar{R}})\left((\hat{e}_n^i)^2 + 2\hat{e}_n^i E\{\tilde{f}_{n-1}^j\} + E\{(\tilde{f}_{n-1}^j)^2\}\right) \\ &\quad + P_{\bar{R}}\left(P_{\bar{A}}E\{(\tilde{f}_{n-1}^i)^2\} + P_{A\bar{B}}E\{(\tilde{f}_{n-1}^{k1})^2\} + P_{\mathrm{AB\bar{C}}}E\{(\tilde{f}_{n-1}^{k2})^2\} + P_{\mathrm{ABC}}E\{(\tilde{f}_{n-1}^{k4})^2\}\right). \end{aligned} \tag{8}$$

Last, since the first frame must be intracoded, and we also assume the first frame is not lost, the initial conditions of the recursion are given as: $E\{\tilde{f}_1^i\} = \hat{f}_1^i$ and $E\{(\tilde{f}_1^i)^2\} = (\hat{f}_1^i)^2$. These recursions are performed at the encoder to calculate the expected distortion at the decoder. The encoder uses this to optimally choose the coding mode for each MB.

B. Encoding With Re-Synchronization per Packet

For re-sync per packet, we insert a header at the front of each packet, telling the location (within the packet) of the beginning of the first MB and its frame/GOB/MB number. All zero location bits are used in the very unlikely case that a packet does not contain the beginning of any MB. A typical illustration is given in Fig. 2. Now, an MB can be reconstructed at the decoder if and only if all packets that contain this MB are received. So,

TABLE I
CONCEALMENT METHOD FOR DIFFERENT SITUATIONS

Situation	Pixel	Corresponding Probability
$\bar{A}\bar{B}\bar{C}$	i	$P_{\bar{A}\bar{B}\bar{C}} = P_{\bar{A}} P_{\bar{B}\mid\bar{A}} P_{\bar{C}\mid\bar{A}\bar{B}}$
$\bar{A}\bar{B}C$	$k3$	$P_{\bar{A}\bar{B}C} = P_{\bar{A}} P_{\bar{B}\mid\bar{A}} (1 - P_{\bar{C}\mid\bar{A}\bar{B}})$
$\bar{A}BC or \bar{A}B\bar{C}$	$k2$	$P_{\bar{A}B} = P_{\bar{A}} (1 - P_{\bar{B}\mid\bar{A}})$
$AB\bar{C}$	$k2$	$P_{AB\bar{C}} = (1 - P_{\bar{A}})(1 - P_{\bar{B}\mid A}) P_{\bar{C}\mid AB}$
ABC	$k4$	$P_{ABC} = (1 - P_{\bar{A}})(1 - P_{\bar{B}\mid A})(1 - P_{\bar{C}\mid AB})$
$A\bar{B}\bar{C}$	$k1$	$P_{A\bar{B}\bar{C}} = (1 - P_{\bar{A}}) P_{\bar{B}\mid A} P_{\bar{C}\mid A\bar{B}}$
$A\bar{B}C$	$k5$	$P_{A\bar{B}C} = (1 - P_{\bar{A}}) P_{\bar{B}\mid A} (1 - P_{\bar{C}\mid A\bar{B}})$

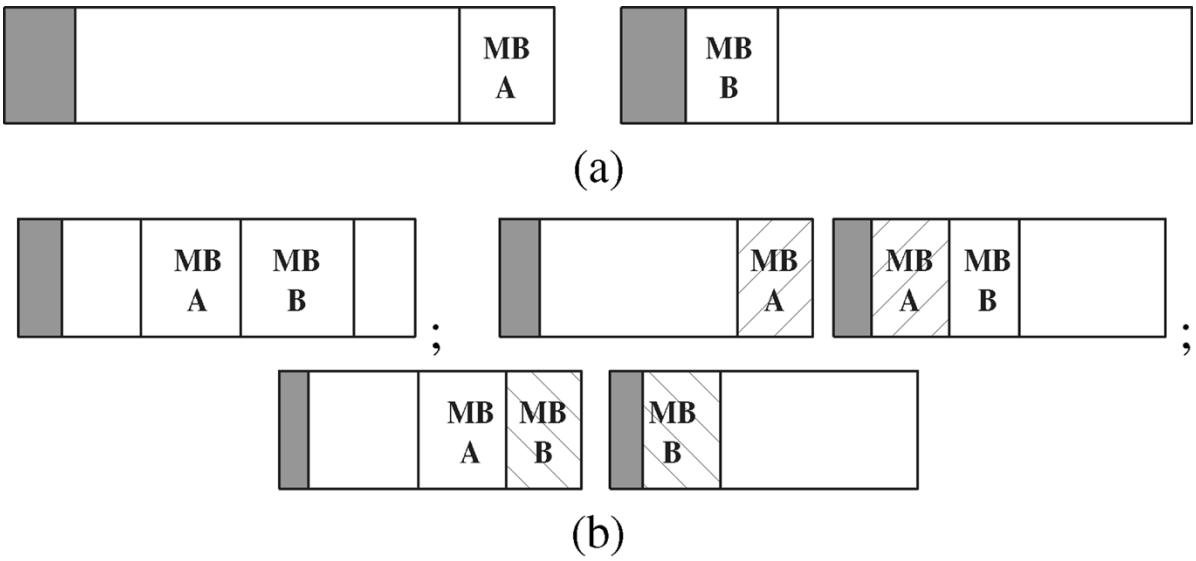


Fig. 3. Illustration of event d and the corresponding probability $P_{\bar{d}}$. (a) The end of A happens to be the boundary of a packet, so $P_{\bar{d}} = 0$. (b) Three situations in which A and B share (at least partly share) a same packet, so $P_{\bar{d}} = p$.

we count the number m of packets that include this MB. The probability that an MB cannot be reconstructed at the decoder is $P_{\bar{R}} = P_{\bar{R}}(m) = 1 - (1-p)^m$. Because usually the compressed bit stream corresponding to one MB is much smaller than the fixed packet length, m usually equals 1 or 2. For example, in Fig. 2, for GOB1, $m = 1$ for all MBs except MB5 and MB10 for which $m = 2$.

The concealment method also needs to be modified. Denote the three nearest MBs above the current decoding MB as A, B and C, from left to right. This time, loss of A does not necessarily mean loss of B or C. With re-sync per packet, it is possible that A and C are received but B is lost, although this is very unlikely because it means B occupies more than one packet. For this situation, if only one of A or C is intercoded, we set the SMV equal to the MV of the intercoded one; if both are intercoded, we use the MV with smaller value. Let $k5$ denote the pixel used for concealment under this situation. We summarize all the situations, the pixels used to conceal, and the corresponding probabilities, in Table I. For example, the first line means A, B and C are all lost, we use pixel i in the previous frame for the concealment (i.e., $\mathrm{SMV} = (0,0)$), and the probability corresponding to this situation is $P_{\bar{A}\bar{B}\bar{C}}$. Also, a modified treatment is needed for special cases when the MBs are on the boundaries of a frame.

Equations (2)–(4) are still valid to compute $P_{\bar{A}}$, $P_{\bar{B}\mid A}$ and $P_{\bar{C}\mid AB}$, respectively, for re-sync per packet, except that here m_A means the number of packets that include A. The parameters l_B and l_C have the same definitions as before, e.g., l_B is the number of packets that B spans beyond the end of the packet with A. If d is the event that the packet shared by A and B is received at the decoder, $\bar{d}$ means this packet is lost. As illustrated in Fig. 3, $P_{\bar{d}} = 0$ if and only if the end of A happens to be the boundary of a packet, and, thus, the packet shared by A and B does not exist (this situation is very unlikely), otherwise $P_{\bar{d}} = p$. Also $P_{\bar{A}\bar{d}} = P_{\bar{d}} P_{\bar{A}\mid\bar{d}} = P_{\bar{d}} \times 1 = P_{\bar{d}}$. Then, we can compute $P_{\bar{B}\mid\bar{A}}$ as follows:

$$\begin{aligned} P_{\bar{B}\mid\bar{A}} &= P_{\bar{d}\mid\bar{A}} P_{\bar{B}\mid\bar{A}\bar{d}} + P_{d\mid\bar{A}} P_{\bar{B}\mid\bar{A}d} \\ &= P_{\bar{d}\mid\bar{A}} \times 1 + (1 - P_{\bar{d}\mid\bar{A}})(1 - (1-p)^{l_B}) \\ &= \frac{P_{\bar{d}}}{P_{\bar{A}}} + \left(1 - \frac{P_{\bar{d}}}{P_{\bar{A}}}\right)(1 - (1-p)^{l_B}). \end{aligned} \tag{9}$$

Similarly, to compute $P_{\bar{C}\mid\bar{A}\bar{B}}$ and $P_{\bar{C}\mid A\bar{B}}$, we define the event e that the packet shared by B and C is received at the decoder,

and $P_{\bar{e}} = p$ except if the end of B happens to be the boundary of a packet, in which case $P_{\bar{e}} = 0$. Then

$$P_{\bar{A}\bar{B}\bar{e}} = P_{\bar{A}\bar{e}} P_{\bar{B}|\bar{A}\bar{e}} = P_{\bar{A}\bar{e}} = P_{\bar{e}} P_{\bar{A}|\bar{e}} = \begin{cases} P_{\bar{e}}, & A \text{ shares the same packet with } C \\ P_{\bar{A}} P_{\bar{e}}, & \text{no common packet for } A \text{ and } C \end{cases} \quad (10)$$

$$P_{A\bar{B}\bar{e}} = P_{A\bar{e}} = P_{\bar{e}} P_{A|\bar{e}} = \begin{cases} 0, & A \text{ shares the same packet with } C \\ P_A P_{\bar{e}}, & \text{no common packet for } A \text{ and } C \end{cases}. \quad (11)$$

At last, we have the following conditional probability in (12), shown at the bottom of the page, and we can calculate $P_{\bar{C}|A\bar{B}}$ in a similar fashion. With these, we compute the probability terms in Table I.

The expected distortion for pixel i is given by (1). For each MB and for each mode selection and quantization step, we first calculate the loss probability $P_{\bar{R}} = 1-(1-p)^m$, where m is the number of packets that contain this MB. Then, for an intracoded MB, $\tilde{f}_n^i = \hat{f}_n^i$ with probability $1 - P_{\bar{R}}$, corresponding to correct receipt of the MB. The recommended concealment method is used if the current MB is lost. The two moments for a pixel in an intracoded MB are given by

$$\begin{aligned} &E\{\tilde{f}_n^i\} \\ &= (1-P_{\bar{R}})\hat{f}_n^i + P_{\bar{R}} \Big(P_{\bar{A}\bar{B}\bar{C}} E\{\tilde{f}_{n-1}^i\} + P_{\bar{A}\bar{B}C} E\{\tilde{f}_{n-1}^{k3}\} \\ &\quad + (P_{\bar{A}B} + P_{\mathrm{AB}\bar{\mathrm{C}}}) E\{\tilde{f}_{n-1}^{k2}\} + P_{\mathrm{ABC}} E\{\tilde{f}_{n-1}^{k4}\} \\ &\quad + P_{A\bar{B}\bar{C}} E\{\tilde{f}_{n-1}^{k1}\} + P_{A\bar{B}C} E\{\tilde{f}_{n-1}^{k5}\} \Big) \end{aligned} \quad (13)$$

$$\begin{aligned} &E\{(\tilde{f}_n^i)^2\} \\ &= (1-P_{\bar{R}})(\hat{f}_n^i)^2 + P_{\bar{R}} \Big(P_{\bar{A}\bar{B}\bar{C}} E\{(\tilde{f}_{n-1}^i)^2\} \\ &\quad + P_{\bar{A}\bar{B}C} E\{(\tilde{f}_{n-1}^{k3})^2\} + (P_{\bar{A}B} + P_{\mathrm{AB}\bar{\mathrm{C}}}) E\{(\tilde{f}_{n-1}^{k2})^2\} \\ &\quad + P_{\mathrm{ABC}} E\{(\tilde{f}_{n-1}^{k4})^2\} + P_{A\bar{B}\bar{C}} E\{(\tilde{f}_{n-1}^{k1})^2\} \\ &\quad + P_{A\bar{B}C} E\{(\tilde{f}_{n-1}^{k5})^2\} \Big). \end{aligned} \quad (14)$$

Similarly, for an intercoded MB, assume the true MV of current pixel i is predicted from pixel j in the previous frame. The first and second moments of $\tilde{f}_n^i$ for a pixel in an intercoded MB are given by

$$\begin{aligned} E\{\tilde{f}_n^i\} = &(1-P_{\bar{R}}) \left(\hat{e}_n^i + E\{\tilde{f}_{n-1}^j\} \right) + P_{\bar{R}} \Big(P_{\bar{A}\bar{B}\bar{C}} E\{\tilde{f}_{n-1}^i\} \\ &+ P_{\bar{A}\bar{B}C} E\{\tilde{f}_{n-1}^{k3}\} + (P_{\bar{A}B} + P_{\mathrm{AB}\bar{\mathrm{C}}}) E\{\tilde{f}_{n-1}^{k2}\} \\ &+ P_{\mathrm{ABC}} E\{\tilde{f}_{n-1}^{k4}\} + P_{A\bar{B}\bar{C}} E\{\tilde{f}_{n-1}^{k1}\} \\ &+ P_{A\bar{B}C} E\{\tilde{f}_{n-1}^{k5}\} \Big) \end{aligned} \quad (15)$$

$$\begin{aligned} E\{(\tilde{f}_n^i)^2\} = &(1-P_{\bar{R}}) \Big((\hat{e}_n^i)^2 + 2\hat{e}_n^i E\{\tilde{f}_{n-1}^j\} \\ &+ E\{(\tilde{f}_{n-1}^j)^2\} \Big) + P_{\bar{R}} \Big(P_{\bar{A}\bar{B}\bar{C}} E\{(\tilde{f}_{n-1}^i)^2\} \\ &+ P_{\bar{A}\bar{B}C} E\{(\tilde{f}_{n-1}^{k3})^2\} + (P_{\bar{A}B} + P_{\mathrm{AB}\bar{\mathrm{C}}}) E\{(\tilde{f}_{n-1}^{k2})^2\} \\ &+ P_{\mathrm{ABC}} E\{(\tilde{f}_{n-1}^{k4})^2\} + P_{A\bar{B}\bar{C}} E\{(\tilde{f}_{n-1}^{k1})^2\} \\ &+ P_{A\bar{B}C} E\{(\tilde{f}_{n-1}^{k5})^2\} \Big). \end{aligned} \quad (16)$$

C. Rate-Distortion Framework

We take into account the expected distortion due to both compression and transmission errors for optimal mode switching. The distortion is computed recursively by the formulas given above for the two possible re-synchronization schemes separately. We incorporate this overall expected distortion within the rate-distortion framework at the encoder, to optimally switch between intra and intercoding on a MB basis. The goal is to minimize the total distortion D subject to a bit rate constraint R.

This problem is an unconstrained Lagrangian minimization, where the algorithm minimizes the total cost $J = D + \lambda R$. Individual MB contributions to this cost are additive, so it can be minimized on a MB basis [6]. Therefore, the encoding mode and the quantization parameter (QP) for each MB are chosen by minimizing

$$\min_{(\mathrm{mode,QP})} J_{\mathrm{MB}} = \min_{(\mathrm{mode,QP})} (D_{\mathrm{MB}} + \lambda R_{\mathrm{MB}}) \quad (17)$$

where the distortion D_{MB} is the sum of the distortion contributions of the individual pixels (d_n^is), and d_n^i is calculated by (1), where the first and second moments of $\tilde{f}_n^i$ are given by (5) to (8) for re-sync per GOB, and by (13) to (16) for re-sync per packet.

Rate control is achieved by modifying λ. As in ROPE [5], we update λ per frame via

$$\lambda_{n+1} = \lambda_n \left(1 + \alpha \left(\sum_{i=1}^{n} R_i - n R_S^* \right) \right) \quad (18)$$

where R_S^* is the target encoding bit rate, $\alpha = 1/5R_S^*$, and λ_0 is set to be 70.

The coding mode and QP are chosen to minimize the Lagrangian cost. For each choice of mode and QP, the encoder computes the number of bits needed for the current MB, the reconstruction failure probability $P_{\bar{R}}$, the individual pixel distortions, and D_{MB}. The algorithm chooses the mode/step size such that D_{MB} and R_{MB} minimize J. Since QP ranges from 1 to 31, and the mode has two choices (intra or inter), this algorithm optimizes over 62 potential combinations.

As to the complexity of this approach, a computational burden is incurred in computing the probabilities corresponding to the different concealment scenarios and the two moments of

$$\begin{aligned} P_{\bar{C}|\bar{A}\bar{B}} &= P_{\bar{e}|\bar{A}\bar{B}} P_{\bar{C}|\bar{A}\bar{B}\bar{e}} + P_{e|\bar{A}\bar{B}} P_{\bar{C}|\bar{A}\bar{B}e} = \frac{P_{\bar{A}\bar{B}\bar{e}}}{P_{\bar{A}\bar{B}}} \times 1 + \left(1 - \frac{P_{\bar{A}\bar{B}\bar{e}}}{P_{\bar{A}\bar{B}}} \right) (1-(1-p)^{l_C}) \\ &= \begin{cases} \frac{P_{\bar{e}}}{P_{\bar{A}} P_{\bar{B}|\bar{A}}} + \left(1 - \frac{P_{\bar{e}}}{P_{\bar{A}} P_{\bar{B}|\bar{A}}} \right) (1-(1-p)^{l_C}), & A \text{ shares the same packet with } C \\ \frac{P_{\bar{e}}}{P_{\bar{B}|\bar{A}}} + \left(1 - \frac{P_{\bar{e}}}{P_{\bar{B}|\bar{A}}} \right) (1-(1-p)^{l_C}), & \text{no common packet for } A \text{ and } C \end{cases} \end{aligned} \quad (12)$$

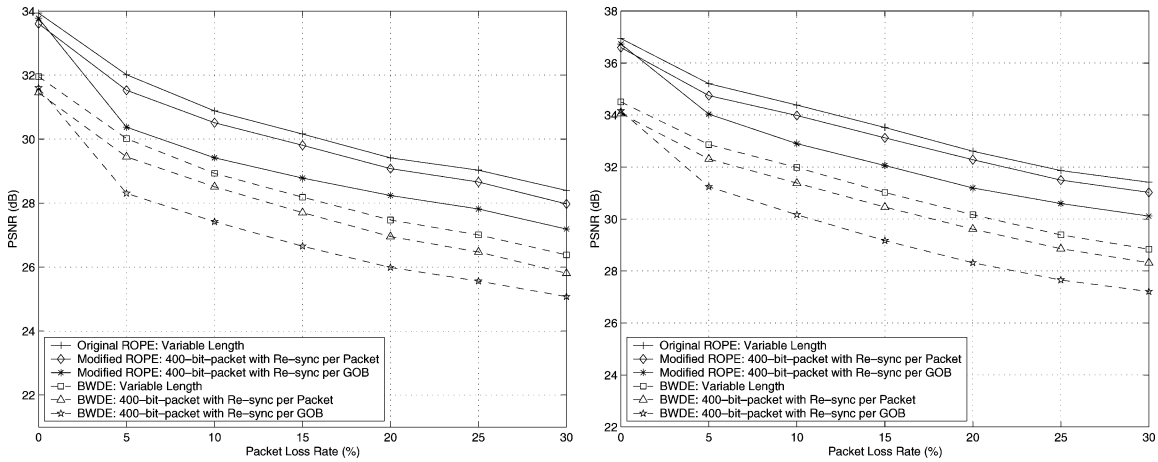


Fig. 4. PSNR performance versus packet loss rate. (a) Carphone QCIF at 200 kbps and 30 fps. (b) Container QCIF at 100 kbps and 15 fps.

$\tilde{f}$ for each mode choice for each pixel. For re-sync per GOB, for each pixel, the algorithm typically needs about 8 addition/multiplication operations to calculate $P_{\bar{A}}$, $P_{\bar{B}|A}$ and $P_{\bar{C}|AB}$, and about 32 addition/multiplication operations to calculate the two moments in (5) to (8) (note that the identical concealment for both intra- and intercoding reduces the complexity). For re-sync per packet, for each pixel, the algorithm typically requires about 36 addition/multiplication operations to create Table I, and about 42 addition/multiplication operations to calculate the two moments in (13) to (16). This complexity is comparable to that of the original ROPE algorithm, which needs about 27 operations to calculate the two moments for each pixel [5]. Also, note that all the complexity mentioned above is incurred only at the encoder.

D. Performance Analysis and Simulation Results

We anticipate that fixed-length packets will perform worse than variable GOB-length packets. Three kinds of penalties explain this performance downgrade. *Rate penalty* comes from sending re-sync information. Re-synchronization per packet involves more re-sync bits than re-sync per GOB. For a shorter fixed packet length, re-sync bits are sent more often. *Division penalty* arises because usually bits of one GOB extend over several fixed-length packets. For example, suppose GOB1 is encoded into packets a and b, and suppose packet and MB boundaries coincide. Similarly GOB2 is encoded into packets c and d. Under the same packet erasure rate, losing one variable-length packet which contains an entire GOB, is equivalent to losing two fixed-length packets. However, losing two fixed-length packets means losing more than one GOB on the average because of sync loss. For example, if packets a and c are lost and we re-sync once per GOB, both GOBs will be entirely lost. A smaller fixed packet length entails a more severe division penalty. If we re-sync once per packet, this penalty will still exist, but will be smaller. *Boundary penalty* occurs whenever the boundary of a lost fixed-length packet is not exactly the boundary of an MB (or GOB). Suppose packet b contains a few bits of GOB2; losing packet b causes the loss of half of GOB1 and the entire GOB2 if we re-sync per GOB. It causes the loss of half of GOB1 and the first MB of GOB2 if we re-sync per packet. Losing two such packets at different points in the stream causes the loss of two GOB halves plus two additional MBs.

Thus, the performance with fixed-length packets should be worse than that with variable-length packets. Re-sync per packet has higher rate penalty but much smaller division and boundary penalties, so it should yield a better performance than re-sync per GOB. Note that we assume Internet congestion causes an equal loss probability for packets of any size.

We will also compare our scheme with the "block-weighted distortion estimate" (BWDE) [5], with the same two fixed-length packetization approaches. BWDE assumes that the current block is correctly received, while the MBs of the previous frame may be lost and concealed; thus, the current block may have concealment distortion because it may be intercoded using the previous frame. The estimate of decoder distortion is $\hat{D} = D_{q1}$ for intramode and $\hat{D} = pD_c + (1-p)D_{q2}$ for intermode, where D_{q1} is the quantization distortion of the current intra-coded pixel, D_c is the weighted average of the concealment distortion of the previous frame blocks that are mapped to the current MB, and D_{q2} is the quantization distortion of the residual for the current intercoded pixel. The Lagrangian $J = \hat{D} + \alpha R$ is minimized among coding modes and QPs for each MB. Because this algorithm unrealistically assumes that the current block is always received, and because the distortion is not additive in its concealment and quantization components, performance with BWDE is expected to be worse than with modified ROPE.

In our simulation results, the system was evaluated using an H.263+ codec with standard QCIF (176×144) video sequences at frame rates of 10, 15, or 30 frames per second (fps). Various target transmission bit rates were tested ranging from 50 to 450 kbps. A random packet loss generator was used to drop packets with variable erasure rates p. Different fixed packet lengths from 100 to 1000 bits were also tested.

Fig. 4 shows the PSNR performance versus packet erasure rate. Fig. 4(a) is for the "Carphone" QCIF sequence at 200 kbps and 30 fps with packet length 400 bits. For a given distortion estimation method (ROPE or BWDE), variable-length packets outperform fixed-length packets, and re-sync per packet outperforms re-sync per GOB. For the ROPE algorithm, from

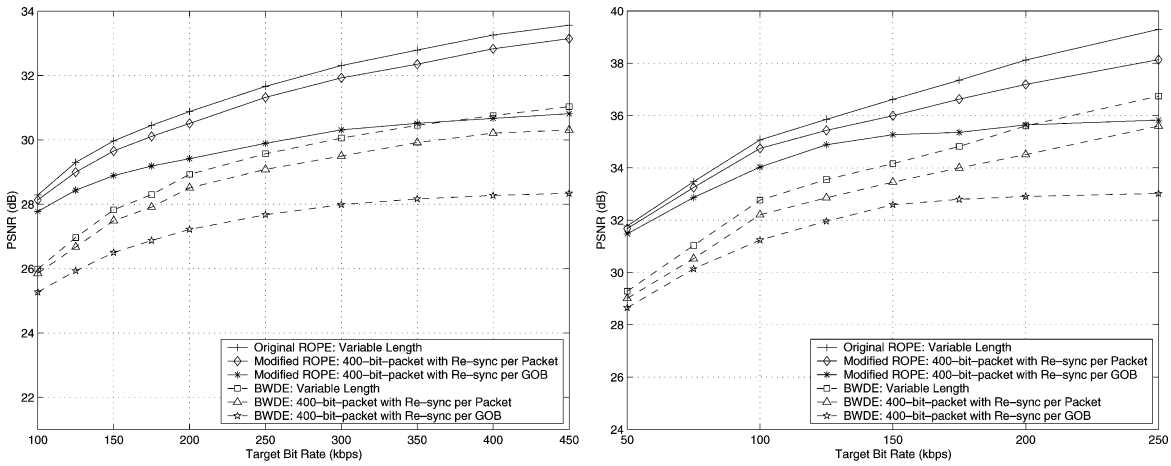


Fig. 5. PSNR performance versus target bit rate. (a) Carphone QCIF at 30 fps, with packet erasure rate $\mathrm{p} = 10\%$. (b) Container QCIF at 15 fps, with packet erasure rate $\mathrm{p} = 5\%$.

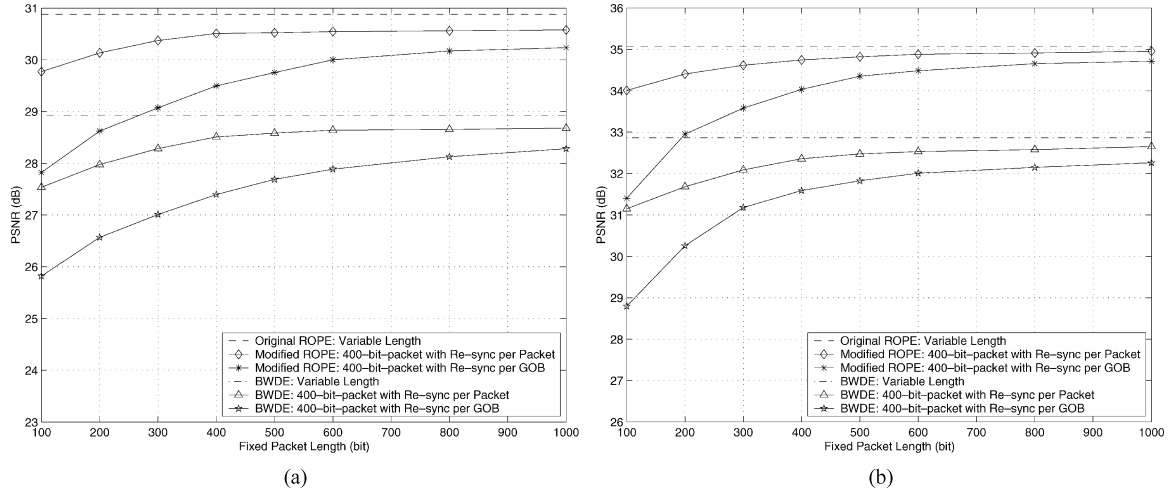


Fig. 6. PSNR performance versus fixed packet length. (a) Carphone QCIF at 200 kbps and 30 fps, with packet erasure rate $\mathrm{p} = 10\%$. (b) Container QCIF at 100 kbps and 15 fps, with packet erasure rate $\mathrm{p} = 5\%$.

$p = 5\%$ to $p = 30\%$, re-sync per fixed-length packet is about 0.2–0.4 dB lower than variable-length packets, and about 1.0 dB higher than re-sync per GOB. At $p = 0\%$, re-sync per packet performs slightly worse than re-sync per GOB because only rate penalty applies. For the same packing method (variable length, fixed length with re-sync per packet or per GOB), ROPE outperforms BWDE by about 2.0 dB. Similar trends appear in Fig. 4(b), which contains results for the "Container" QCIF image sequence at 100 kbps and 15 fps with 400-bit fixed-length packets.

Fig. 5 shows PSNR versus transmission rate. Fig. 5(a) is for "Carphone" at 30 fps with packet length 400 bits and error rate $p = 10\%$. For the same distortion estimation method, as the transmission rate grows, the gap between variable-length packets and fixed-length with re-sync per packet is nearly constant. For ROPE, this constant is about 0.35 dB. However, the gap between variable-length packets and fixed-length with re-sync per GOB increases dramatically, mostly due to the more serious division penalty as rate increases. For ROPE, it goes from 1.0 dB at 100 kbps up to 2.7 dB at 450 kbps. For the same packing method, ROPE beats BWDE by about 2.0–2.5 dB, and the gap increases with rate. In Fig. 5(b), which is for "Container" at 15 fps with 400-bit fixed-length packets and $p = 5\%$, we observe similar trends.

Fig. 6 shows PSNR versus packet length ranging from 100 bits to 1000 bits. Fig. 6(a) is for "Carphone" at 200 kbps and 30 fps with packet loss rate $p = 10\%$, and Fig. 6(b) is for "Container" at 100 kbps and 15 fps with $p = 5\%$. For the same distortion estimation algorithm, a larger fixed packet size leads to a smaller gap between variable-length and fixed-length packet results. Again, the ROPE algorithm yields consistent and significant gains over BWDE.

In summary, to integrate this source encoder with forward error correction (FEC) to operate over a wireless/Internet channel, we change the variable-length packetization to fixed-length packetization, and modify the distortion estimation approach accordingly. In doing this, one pays three kinds of penalties. Experimental results demonstrated this PSNR downgrade of about 0.2–0.5 dB. Simulation results also showed re-sync per packet outperformed re-sync per GOB.

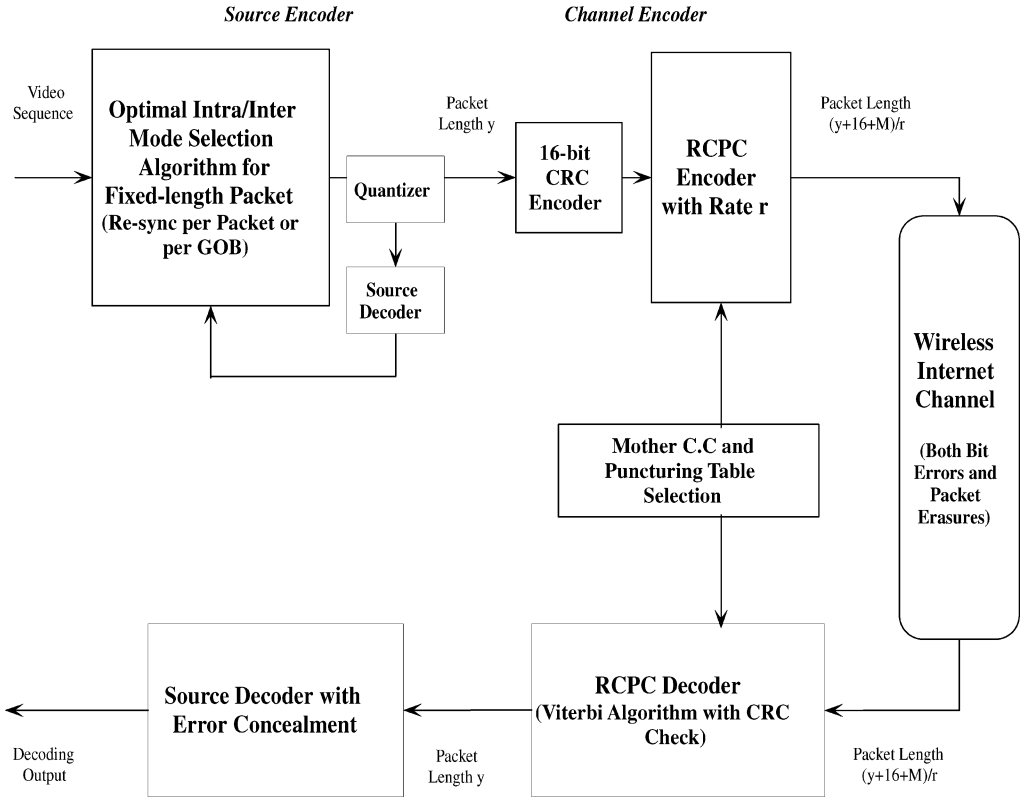


Fig. 7. System overview.

III. Source and Channel Coding Over Wireless/Internet

The delivery of packet video over tandem Internet and wireless channels is discussed in this section. We assume the wireless channel introduces uniform random bit errors with rate P_b, and the Internet loses packets with erasure rate p. We assume P_b and p are constant and known at the encoder. In practice, this information may come from a test data sequence and tracking of channel conditions. The major resource shared between the source and channel encoders is the given target transmission rate. If the channel condition is poor (say, $P_b \geq 0.01$), more bits are needed for channel error detection and correction; thus, a smaller bit rate is used for source encoding. The system diagram is shown in Fig. 7. In particular, the wireless component is modeled as the concatenation of an inner RCPC code and an outer error detection code; thus, the tandem channel can be dynamically converted into a erasure channel for the source encoder, and the algorithm proposed in Section II can be easily re-used. We now discuss each component in detail.

A. Source Encoder

The video source is encoded using the optimal inter/intramode selection algorithm with fixed-length packets (re-sync per GOB and per packet are analyzed and compared). The mode selection algorithm was designed for a given output bit rate of the source coder and a given packet erasure rate. Here we are given instead the target transmission rate (that is, the output bit rate of the channel coder), and the wireless bit errors may increase the packet loss rate if the corrupted packets cannot be corrected and are, thus, discarded.

Given the bit-error rate P_b, the channel coder (as discussed below) chooses a rate-compatible punctured convolutional (RCPC) code with channel code rate r from a family of RCPC codes so as to keep the probability of packet drop due to uncorrectable bit errors (p'_e) at about 1% for most of the transmission rates of interest. The packet erasure rate due to Internet congestion is p; thus, the total packet loss rate is $\hat{p} = p + p'_e - p \times p'_e \approx p + 0.01 - p \times 0.01 = 0.99p + 0.01$.

Knowing r, the transmission target rate R^* and frame rate f, as well as the fixed packet length, the source encoder determines the corresponding target source coding output bit rate R^*_S. With the target output bit rate of the source coder and the total packet loss rate $\hat{p}$, we may use the intra/intermode selection algorithm directly as derived in Section II.

B. Channel Encoder

We use a concatenated code consisting of a CRC outer coder and RCPC inner coder. That is, the grouped fixed-length y source information bits are appended with a 16-bit CRC and M zero ending bits to flush the memory and terminate the trellis decoding in the zero state. Then the $(y + 16 + M)$ bits are convolutionally encoded using a rate r RCPC coder [7].

CRCs provide error detection with low complexity and flexible block length. The optimal 16-bit CRCs for different packet lengths are proposed in [8], [9]. In particular, C_1, C_3, and C_4 [8] are typically used for packet lengths less than 151, between 151 and 257, and greater than 257 (and less than 28 658 bits), respectively. All of these yield a very low probability of undetected error, typically less than 10^{-5}.

RCPC codes are a powerful extension of punctured convolutional codes [10], [11]. Here, the RCPC code is chosen adaptively to make the probability of packet drop due to uncorrectable bit error about 1%, under the given channel bit-error rate P_b ($P_b \leq 0.15$) for most of the transmission rates of interest. As a practical matter, the 1% cannot be exactly achieved, and we used a rate 2/7 RCPC code when $P_b > 0.05$, a rate 2/3 RCPC code when $0.005 < P_b \leq 0.05$, a rate 8/9 RCPC code when $10^{-5} < P_b \leq 0.005$, and no channel coder is used if $P_b \leq 10^{-5}$. All of these RCPC codes have a memory $M = 6$ and a puncturing period length 8. The details of their construction are given in Table III.

To avoid an unacceptable corresponding packet loss rate, the FEC selection needs to guarantee that the bit-error probability after correction is very small. Fig. 8(a) shows the relationship between the bit-error rate and the corresponding packet error rate without error correction. When bit-error rate is very small ($\leq 5 \times 10^{-5}$), the packet error rate is roughly the product of the bit-error rate and the fixed packet length. If the bit-error rate is larger ($\geq 5 \times 10^{-4}$), the corresponding packet error rate goes up dramatically and reaches nearly 100% as the bit-error rate goes to 0.02. Thus, a powerful RCPC code is needed to avoid bad system degradation.

Simulations also show that it is reasonable to choose the packet drop rate due to uncorrectable bit error to be roughly 1%. Fig. 8(b) shows the PSNR gap for different target packet drop rates, where PSNR gap (on the y-axis) refers to the average gap between the PSNR with zero packet drop rate and the PSNR under the given drop rate over different wireless bit-error rates. When the drop rate is high, the gap is large, but when the drop rate goes down to roughly 1%, the PSNR gap is very small. Returns diminish when the drop rate due to uncorrectable bit errors is pushed below 1%.

For the efficient detection of uncorrected errors, the serial list-Viterbi algorithm at the channel decoder was used with a list of 100 paths [11], [12]. The optimal path in the Viterbi decoding is chosen among those paths that satisfy the checksum equations. If at a given depth of trellis decoding, none satisfied the checksum equations, then an uncorrected error is declared and this packet is discarded. The corresponding MBs are then reconstructed from the previously received MBs using the concealment methods. Here we check 100 paths; increasing the number of paths does not necessarily improve the performance of the system, because we may reach a point where the probability of undetected errors becomes too high, and it is shown that dropping the uncorrected packet and using a proper concealment method may give a better result than using an uncorrected packet [12].

C. Performance Analysis and Simulation Results

This system was evaluated for the transmission of video over a tandem channel. The packet erasure rates tested were $p = 5\%$ and 10%, and bit-error probabilities ranged from $P_b = 0$ to $P_b = 0.15$. The same error patterns were used for all algorithm versions. Again, we compare modified ROPE and BWDE distortion estimation.

TABLE II
PARAMETERS OF FIG. 9 FOR MODIFIED ROPE AND RE-SYNC PER PACKET

RCPC Code Rate	Mother Convolutional Code			Puncturing Table
	Rate	Memory	Generation Matrix	
8/9	1/3	6	1 0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1	1 1 1 1 0 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
2/3	1/3	6	1 0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1	1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0
2/7	1/4	6	1 1 0 1 1 0 1 1 0 1 0 0 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0

Fig. 9 shows PSNR versus bit-error rate from $P_b = 0$ to $P_b = 0.15$. Fig. 9(a) is for "Carphone" at 400 kbps and 30 fps with packet length 400 bits, and $p = 10\%$. Fig. 9(b) is for "Container" at 150 kbps and 15 fps with packet length 400 bits, and $p = 5\%$. Results are consistent with our predictions. With the same distortion estimation method (ROPE or BWDE), re-sync per packet yields better performance than does re-sync per GOB; with the same fixed packetization method, modified ROPE outperforms BWDE. Table II shows parameters for the simulation for modified ROPE with re-sync per packet. Note that, as the bit-error rate increases, a lower rate channel coder is used, and so the bit rate for source coding decreases. The estimates of the total packet loss rate at the encoder are close to the actual packet loss rate found at the decoder, consistent with our goal that the packet loss due to uncorrectable bit error is about 1%.

Fig. 10(a) shows PSNR versus target transmission rate, and Fig. 10(b) shows PSNR versus time (frame number) at 300 kbps. The image sequence is "Salesman" at 10 fps with packet length 800 bits, $p = 10\%$ and $P_b = 0.01$. Again re-sync per packet yields a much better performance than re-sync per GOB, and modified ROPE outperforms BWDE.

We also compare our system with a recent system [2] which uses a H.263+ source coder, and a concatenated FEC scheme employing interlaced Reed-Solomon (RS) codes and RCPC codes to protect the video data from packet loss and bit errors, respectively. We compare the performance of our system with the results given in [2, Fig. 6], where the comparison system is operated over a wired IP and a wireless Rician fading channel with parameter K. Because sufficient interleaving is assumed to randomize the burst errors in [2], the SNR of the fade can be translated to a bit-error rate as follows:

$$P_e = \int_r P(e|r) f(r) dr = \int_0^\infty \varphi\left(-x\sqrt{SNR}\right)\left(xe^{-K+x^2/2} I_0(x\sqrt{2K})\right) dx \quad (19)$$

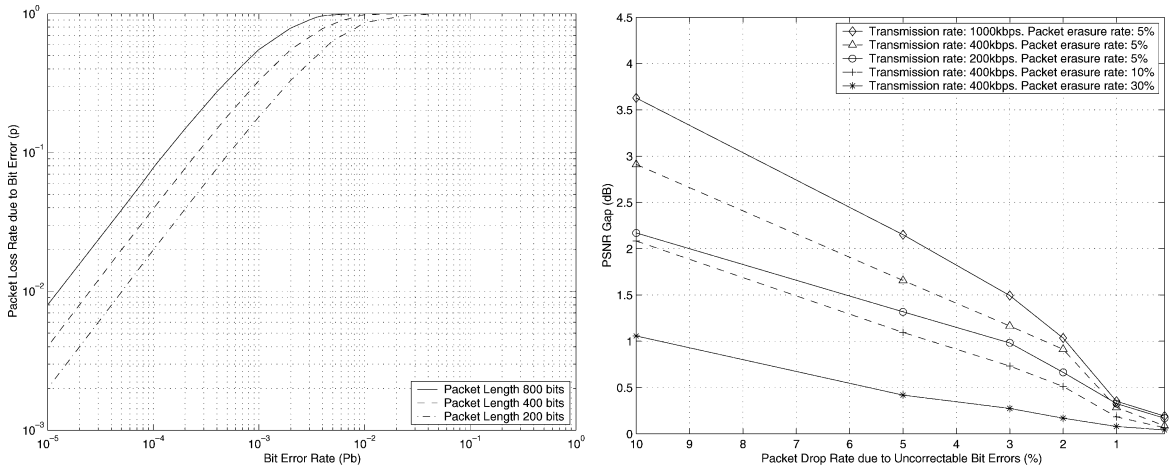
where $\varphi(x) = (1/\sqrt{2\pi}) \int_{-\infty}^{x} e^{-t^2/2} dt$ and

$$I_0(x) = \sum_{n=0}^{\infty} \frac{(-1)^n \left(\frac{x}{2}\right)^{2n}}{(n!)^2} = \frac{1}{\pi} \int_0^\pi \cos(x \sin\theta) d\theta$$

TABLE III
RCPC CODES USED IN THE SYSTEM

Carphone QCIF Re-synchronization per Packet						
Bit Error Rate	10^{-6}	10^{-5}	10^{-4}	10^{-3}	0.01	0.10
Source Bit Rate (kbps)	390	390	328	328	246	106
Assumed Packet Loss Rate (%)	10.9					
Total Packet Loss Rate (%) (Found at the Decoder)	10.04	10.35	10.06	10.58	10.89	11.52
PSNR (dB)	32.73	32.61	32.28	32.20	31.02	28.75

Container QCIF Re-synchronization per Packet						
Bit Error Rate	10^{-6}	10^{-5}	10^{-4}	10^{-3}	0.01	0.10
Source Bit Rate (kbps)	132	132	117	117	88	39
Assumed Packet Loss Rate (%)	5.95					
Total Packet Loss Rate (%) (Found at the Decoder)	5.04	5.38	5.06	5.61	5.93	6.61
PSNR (dB)	35.77	35.76	35.60	35.57	34.15	32.72



(a) (b)

Fig. 8. Illustration of why we chose the packet drop rate due to uncorrectable bit errors to be 1%. (a) Bit-error rate versus corresponding packet loss rate without error correction; (b) PSNR gap for different target packet drop rates, "Carphone" QCIF sequence at 10 fps, and fixed-packet length 400.

which are the cumulative Gaussian distribution function and the modified Bessel function of order zero, respectively. The simulation results are for "Susie" at 128 kbps and 7.5 fps. The comparison system generates 9 packets per frame, with the fixed packet length $128\,\mathrm{k}/(7.5\times 9) = 1896$ bits, and our system is operated with an 800-bit packet. The results are shown in Fig. 11. Over most bit-error rates, our system outperforms the comparison system by about 0.4 dB. The comparison system outperforms ours in a small interval, perhaps because it selects among a larger set of RS and RCPC codes.

The sensitivity to mismatched channel status is examined in Fig. 12, where the channel status used at the transmitter for the optimization mismatches the actual channel status in the network. The figures are for "Carphone" with transmission rate 400 kbps and packet length 400 bits, with re-sync per packet. Fig. 12(a) is for performance of mismatched bit-error rate under a correct packet erasure rate estimate. The horizontal axis is the actual channel bit-error rate; each curve represents the performance of the system that persists in using a particular rate RCPC code (so it is mismatched out of the correct bit-error range). Performance drops dramatically when the actual bit-error rate is higher than the estimate. The upper bound curve is the performance of a properly matched system. Fig. 12(b) illustrates the mismatched packet erasure rate under a matched 0.001 wireless bit-error rate. Again, each curve represents the performance where a particular packet erasure rate is assumed. At each actual channel status, the matched estimate yields the best performance, and poorer performance goes along with increasing

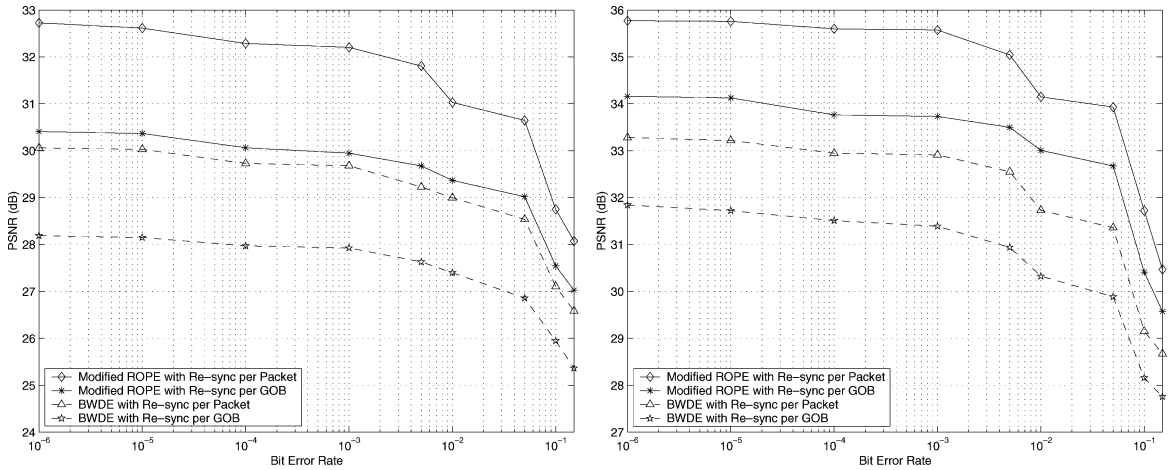


Fig. 9. PSNR performance versus bit-error rate. (a) Carphone QCIF at 400 kbps and 30 fps, p $= 10\%$, and packet length 400 bits. (b) Container QCIF at 150 kbps and 15 fps, p $= 5\%$. and packet length 400 bits.

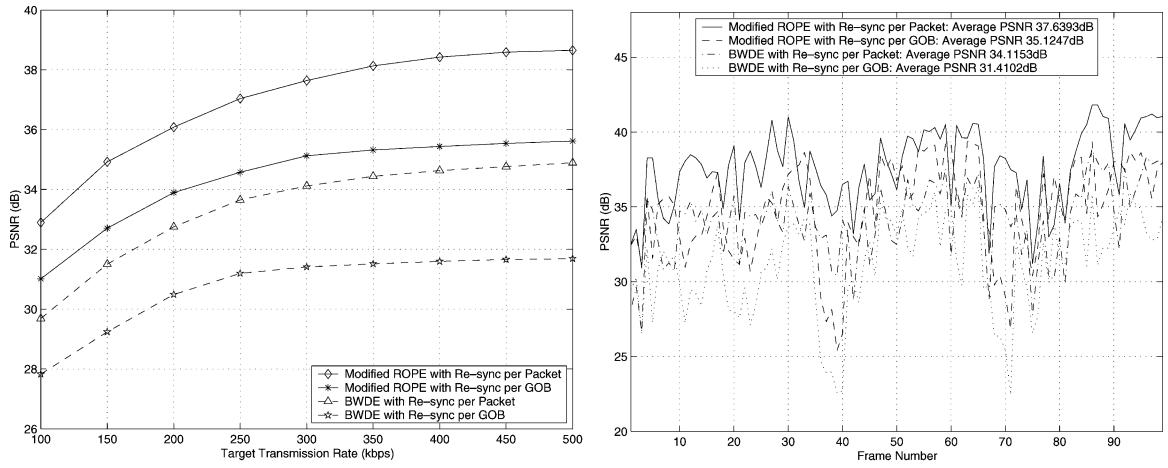


Fig. 10. PSNR performance versus transmission rate and versus frame number. (a) Salesman QCIF at 10 fps, p $= 10\%$, and $P_b = 0.01$, packet length 800 bits. (b) Salesman QCIF at 300 kbps and 10 fps, p $= 10\%$, and $P_b = 0.01$, packet length 800 bits.

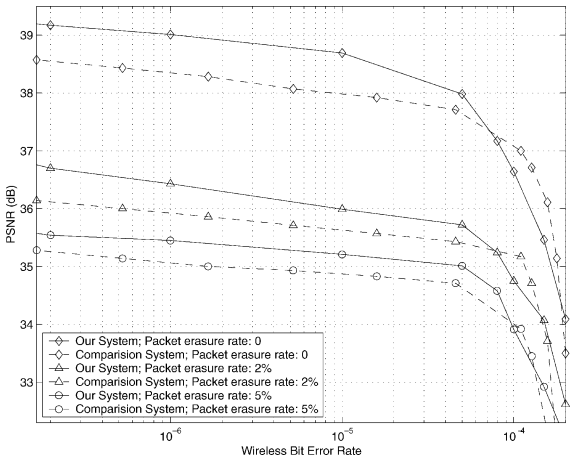


Fig. 11. PSNR performance versus wireless bit-error rate, Susie QCIF with 128 kbps and 7.5 fps, 800-bit fixed packet length for our system, nine packets per frame for the comparison system.

mismatch. The upper bound curve shows performance of the matched system.

In many applications, both bit errors and packet erasures occur in bursts, and the Gilbert–Elliot model is good for capturing bursty loss patterns. A two-state Gilbert–Elliot model with the states named *Good* and *Bad* is illustrated in Fig. 13. Note that the state transition characteristics are completely determined by the values $P_{\rm GG}$ and $P_{\rm BB}$, where, for example, $P_{\rm GG}$ is the probability that the next state is *Good*, given the current state is *Good*. Then the mean time durations (measured in number of steps) that the channel is in the *Good* and *Bad* states are $T_G = 1/(1 - P_{\rm GG})$ and $T_B = 1/(1 - P_{\rm BB})$, respectively. In Fig. 14, we compare the performance of our system when used over a constant random channel to that when the channel is bursty. The top curve is the system performance for a constant channel with $p = 10\%$ and $P_b = 0.01$, which is the same as the top curve in Fig. 10(a). The lower curve is the system performance for a channel with a constant $P_b = 0.01$, while the packet erasures are determined from a Gilbert–Elliot model utilizing the limiting per_state error probabilities of one and zero for each packet. We chose $P_{\rm GG} = 0.9$ and $P_{\rm BB} = 0.1$, thus $T_G = 10$, $T_B = 10/9$. The overall erasure rate over a long

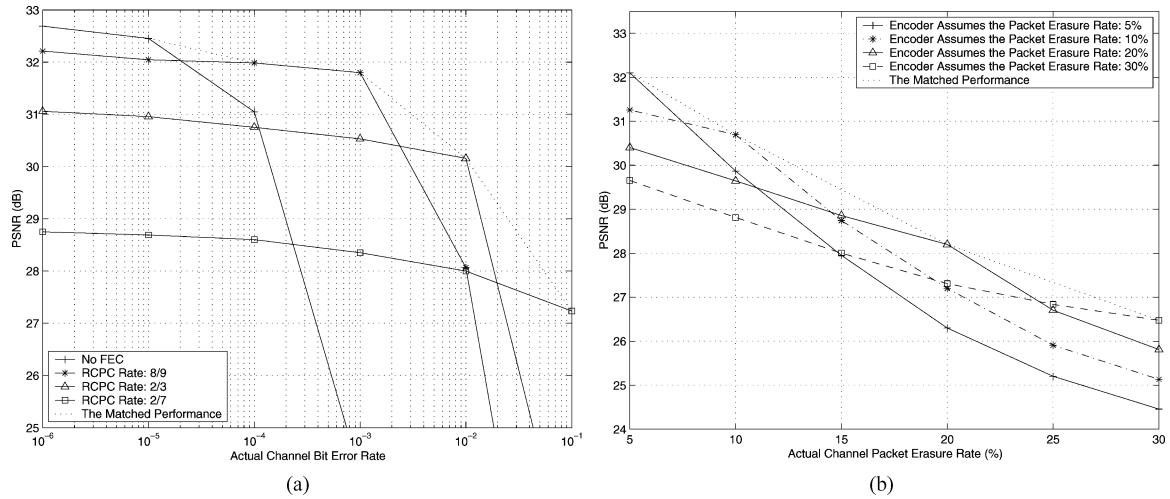


Fig. 12. PSNR performance for mismatched system, Carphone QCIF at 400 kbps and 15 fps, with packet length 400 bits. (a) $P_b = 0.001$ and $\mathrm{p} = 0\%$. (b) $P_b = 0$ and $\mathrm{p} = 5\%$.

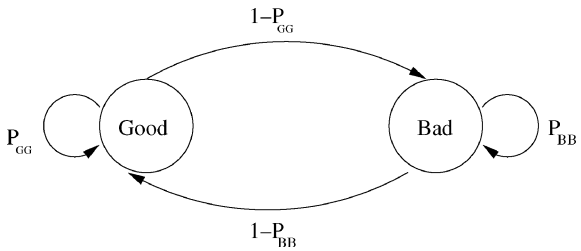


Fig. 13. Two-state Gilbert–Elliot model.

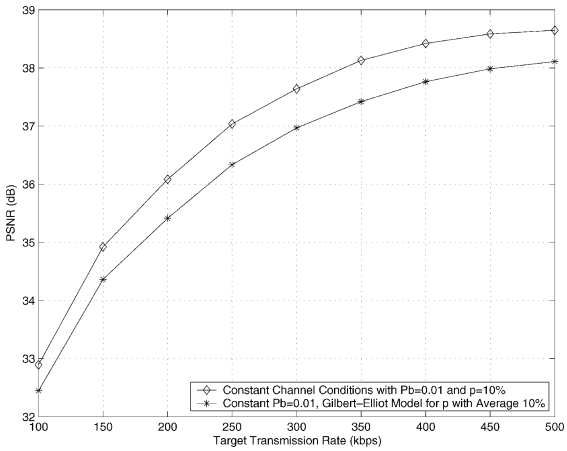


Fig. 14. PSNR performance versus transmission rate, Salesman QCIF at 10 fps, re-sync per packet, with packet length 800 bits.

period of time, which is equal to the percentage of time that the channel is in the *Bad* state, is also 1%. Note that the performance degrades when the channel follows the Gilbert–Elliot model, because of the mismatch of the channel status, that is, the transmitter assumes the packet erasure is a constant 10%, while actually there are two states of erasure rate 0% and 100% with a certain coherent time.

IV. Performance Over Time Varying Channels With Feedback

In the previous sections, we assumed the channel conditions (packet erasure rate and bit-error rate) are known in advance by the transmitter, and stay constant. We also assumed that there is no feedback information from the receiver. However, real channels are usually time varying, and a backward channel from the receiver to the transmitter is available in many applications. Through this feedback channel, the receiver can signal to the transmitter its estimate of the current channel conditions and the actual packet loss rate found at the decoder, and the transmitter can adapt its encoding choices accordingly. What is more, as indicated in [5], the backward channel can also specify lost packets via acknowledgment (ACK) or negative-acknowledgment (NACK), to obtain additional gain in the performance. We will extend our system to time-varying channels and feedback.

For convenience, we assume the wireless bit-error rate P_b and the packet erasure rate p are constant for the packets of the same frame, and they vary from frame to frame. We assume the transmitter knows the channel status correctly for the first frame. After that, it needs feedback to track channel variation. We also assume that the feedback link is error free.

A. Feedback of Channel Conditions

Here, we will not include channel estimation; we assume the decoder can estimate channel conditions correctly and instantaneously, and the transmitter will use this error free information, possibly with some delay, to choose intra/intermodes or adjust channel code rates.

If the feedback information arrives at the transmitter with negligible delay, the bit-error rate and packet erasure rate used at the transmitter match actual channel conditions, so it should yield the upper limit of the performance of our system for the given channel model. In practice, there usually exists some feedback delay due to propagation time or buffering time. We assume a fixed feedback delay d. The transmitter knows the exact channel conditions of the $(n-d)$th frame as it encodes the nth frame. At that time, all frames before the nth are already transmitted.

Due to the memory in the channel, a natural guess is that the erasure rate and bit-error rate seen by the packets of the nth frame are the same as those seen by the $(n-d)$th frame, as that is the newest feedback information obtained by the transmitter. With this information, the transmitter first selects the proper RCPC code according to the bit-error rate, and then the modified ROPE algorithm does distortion estimation and selects the mode and QP that minimize the Lagrange (17).

When the ROPE algorithm estimates distortion for the nth frame, it has the first and second moments of the expected distortions for each pixel in frame $(n-1)$. These are used in the recursive formulas to compute the estimates for frame n. With feedback, the transmitter knows the channel conditions for the $(n-d)$th frame and its packet loss rate experienced at the decoder. Although the transmitter cannot use this information to re-encode frames $(n-d)$ through $(n-1)$, because they are already sent out, it can use the feedback information to refine the distortion estimate for these frames and, therefore, for the nth frame as well.

The estimation refinement starts with the $(n-d)$th frame, because now the transmitter has the exact channel conditions for this frame. For purposes of the recursive computations, it also temporarily assumes that the channel conditions stay constant at the conditions of the $(n-d)$th frame up to the nth frame. From frame $(n-d)$ up to and including frame $(n-1)$, the source transmitter recursively recomputes the first and second moments for each pixel according to this newest known packet loss rate. For frame n, the transmitter estimates distortion based on these refined estimates $E\{\tilde{f}_{n-1}^i\}$ and $E\{(\tilde{f}_{n-1}^i)^2\}$, and selects a mode. The refined computation prevents the accumulation of estimation error.

This refined estimation algorithm should yield better performance than the simple estimation method. The refined estimation method adjusts the estimates at each time interval, so only the moments of $\tilde{f}$ in the last $(d-1)$ frames may be incorrect because the transmitter does not yet have feedback information for these frames.

The computational complexity is higher than for the simple estimation case, because we need to re-compute the moments of the previous d frames. For d in the range of 0–20 (equivalently, 0–600 ms for 30 fps, and 0–200 ms for 10 fps), this complexity is modest. Also, the refined estimation algorithm needs more storage to store the moments of the $(n-1)$th frame $E\{\tilde{f}_{n-1}^i\}$ and $E\{(\tilde{f}_{n-1}^i)^2\}$. As in the simple estimation method, it needs to store the moments of the $(n-d-1)$th frame $E\{\tilde{f}_{n-d-1}^i\}$ and $E\{(\tilde{f}_{n-d-1}^i)^2\}$ and all the intra/intermode selections and quantization step choices of each MB from the $(n-d)$th frame through the current frame.

B. Feedback of ACK/NACK

Another kind of feedback information is to specify lost packets via ACK or NACK. This type of feedback information was used in [5], where the refined distortion estimation was proposed and shown to outperform simple estimation. For the packet erasure channel, the packet erasure rate of the channel can be inferred from the ACK/NACK feedback; while for the wireless or the tandem channels, the channel conditions cannot be inferred from the packet drop rate after the channel decoder, since different FEC is used for different wireless channel conditions.

For a fixed feedback delay d, the transmitter can now exactly calculate the decoder reconstruction up to frame $(n-d)$, but the packet loss history from frame $(n-d+1)$ to frame n is still unknown. To use the feedback information, as shown in [5, Section V], the transmitter will recompute exactly the $(n-d)$th frame of decoder reconstruction by employing error concealment whenever the packets were lost; then the reconstructed frame is used to initialize the recursion formulas to estimate the distortion from frame $(n-d+1)$ up to frame n; at last, the refined estimates $E\{\tilde{f}_n^i\}$ and $E\{(\tilde{f}_n^i)^2\}$ are incorporated into the R-D optimization mode selection.

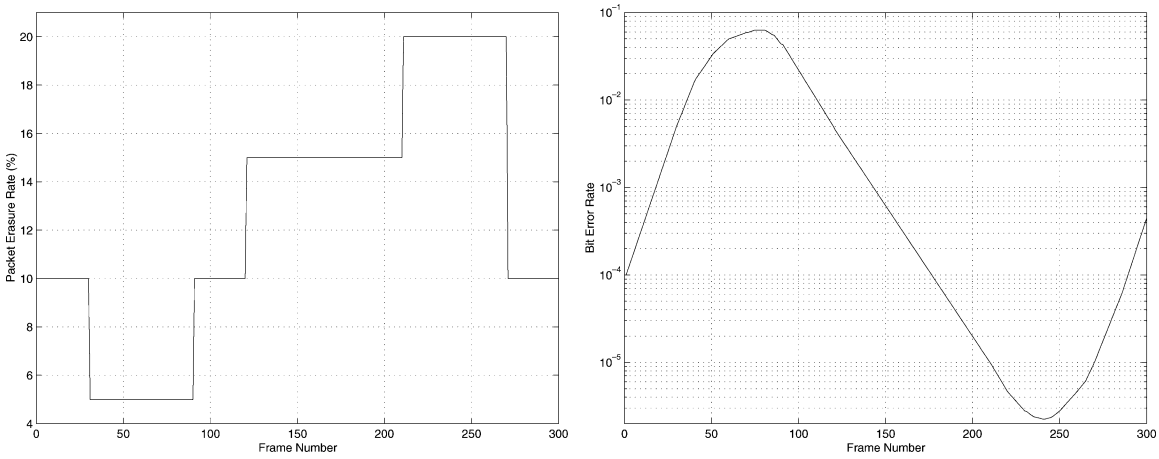
For the tandem varying channel, sending back *both* the channel conditions and the ACK/NACK information can result in further improvement of the performance, by decreasing the mismatch loss from tracking the channel variation, and employing the exact error concealment from the ACK/NACK information together.

Again, the computational complexity involved in updating all the intermediate frames may be a problem, and the performance degrades as the delay increases. When the delay is large, we can ignore the feedback information to reduce complexity with a relatively small penalty in performance.

C. Performance Analysis and Simulation Results

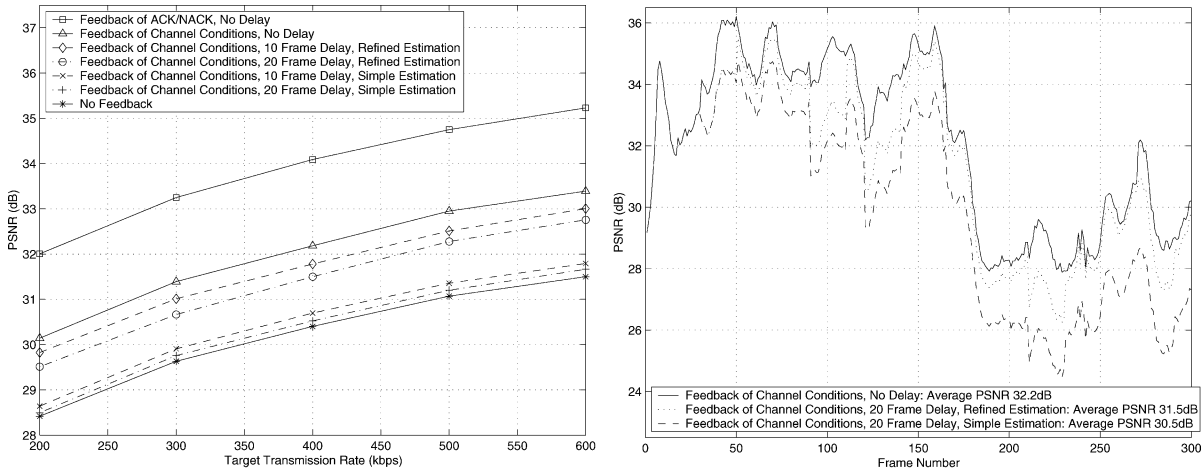
As before, the source encoder is implemented by modifying the H.263+ coder. The system is operated over a time-varying tandem channel. The source is 300 frames from "Carphone" at 30 fps with packet length 400 bits. The feedback performance is compared with delays of zero, 10 and 20 frames.

In Fig. 16, a channel with $P_b = 0$ and varying packet loss rate is considered. The variation of p over time (frame) is shown in Fig. 15(a) in the range from 5% to 20%. Fig. 16(a) shows the system performance with re-sync per packet over different target transmission rates. The top curve is for the instantaneous feedback of ACK/NACK; note that, for a pure packet erasure channel, the packet erasure rate can be inferred from the ACK/NACK information, so this curve actually corresponds to the use of both instantaneous ACK/NACK feedback information and channel condition feedback. The bottom curve is for the system without feedback; the encoder assumes a packet erasure rate equal to the average 12.5%. The other curves on the figure corresponds to feedback of only channel conditions with delay of 0, 10, and 20 frames; for the delayed feedback, simple and refined estimation are also compared. It is shown that the refined estimation method outperforms the simple estimation by more than 1 dB, and the feedback of the additional information of ACK/NACK can yield a further gain in performane. In Fig. 16(b), we show the PSNR of each frame for the system with re-sync per packet at the target transmission rate of 400 kbps, for the feedback of channel conditions with both instantaneous and 20 frame delay. The PSNR with refined feedback almost achieves the upper limit for instantaneous feedback, because the error model of this channel is piecewise constant with period longer than the fixed feedback time. The PSNR with simple estimation results in a larger gap.



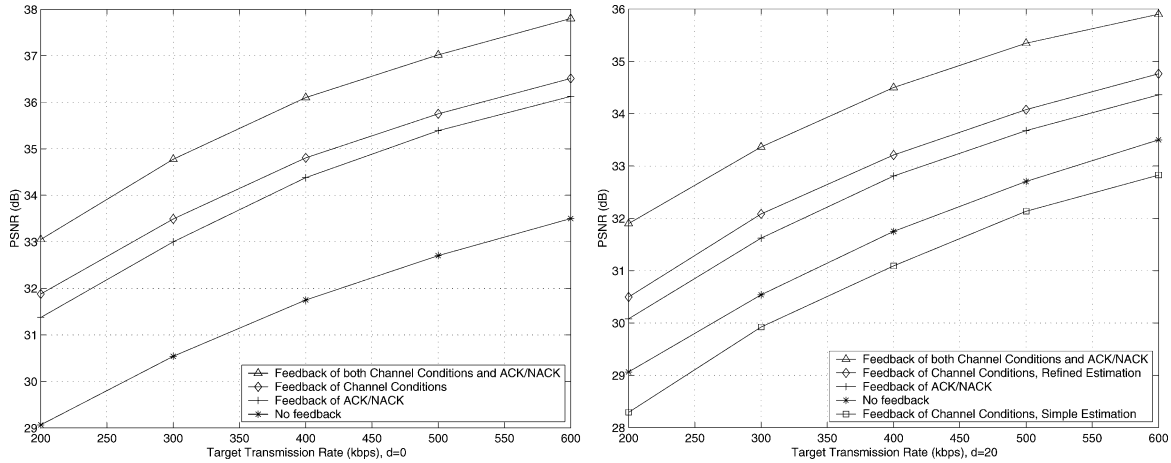
(a) (b)

Fig. 15. Channel variation model over time. (a) Time-varying channel packet erasure rate over time. (b) Time-varying channel bit-error rate over time.



(a) (b)

Fig. 16. PSNR performance over the time-varying pure packet erasure channel given in Fig. 15(a), system with re-sync per packet, Carphone QCIF 30 fps, and packet length 400 bits. (a) PSNR performance versus transmission rate. (b) PSNR performance versus frame number at 400 kbps.



(a) (b)

Fig. 17. PSNR performance over the time-varying pure wireless bit-error channel given in Fig. 15(b), system with re-sync per packet, Carphone QCIF 30 fps, and packet length 400 bits. (a) PSNR performance versus transmission rate, with instantaneous feedback. (b) PSNR performance versus transmission rate, with 20 frames delayed feedback.

Fig. 17 shows the performance over a channel with varying bit errors, and packet erasure rate $p = 0$. The variation of P_b is shown in Fig. 15(b). We chose a smoothly varying curve so that it plausibly could represent a realization of a channel with

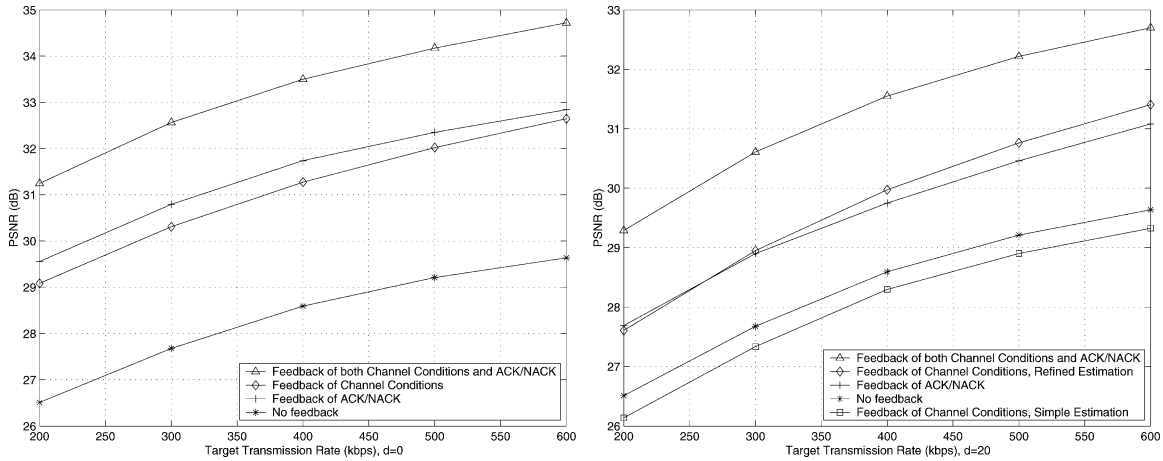


Fig. 18. PSNR performance over a tandem channel with both time-varying packet erasure rate and bit-error rate, which is the combination of Fig. 15(a) and (b), system with re-sync per packet, Carphone QCIF 30 fps, and packet length 400 bits. (a) PSNR performance versus transmission rate, with instantaneous feedback. (b) PSNR performance versus transmission rate, with 20 frames delayed feedback.

memory. The performance versus transmission bit rate for various combinations of instantaneous feedback of channel conditions and of ACK/NACK are shown in Fig. 17(a), for the system with re-sync per packet. For the case of no feedback, the transmitter assumes the channel bit-error rate is always 0.01; thus, it keeps using the RCPC code with rate 2/3. It is shown that combined feedback yields better performance than the use of only one type of feedback. In Fig. 17(b) we show the PSNR versus transmission bit rate for 20-frame delayed feedback of both types of information. For feedback of channel conditions, the refined and simple estimation methods are compared. Again, combined feedback results in best performance, and refined estimation outperforms simple estimation. Note that the performance of the simple estimation scheme with feedback is worse than that of choosing an appropriate "average" channel condition in the absence of feedback.

Fig. 18 shows the performance over a tandem channel model with time-varying bit-error rate and time-varying packet erasure rate, which accounts for the conditions illustrated in Fig. 15(a) and (b). Fig. 18(a) and (b) show the PSNR performance versus transmission bit rate of various combinations of feedback information, in conjunction with either instantaneous feedback or 20-frame delayed feedback, respectively. We observe similar trends here; once again the advantage of combined feedback information and refined estimation is evident.

In summary, simulation results showed that combined feedback of both channel conditions and ACK/NACK information improve system performance compared to the feedback of just one type of information. For feedback of channel conditions, the refined estimation method substantially outperforms the simple estimation method.

V. Conclusion

In this paper, we present a transmission scheme for fixed-length packet video. The transmission channel is a tandem channel which models both packet erasures and bit errors. We solve this tandem channel R-D optimization problem in two steps. First, we propose a video encoder using optimal inter/intramode selection, operating over the wireline erasure-only channel. Then we added the wireless component. For this we used a concatenation of an inner RCPC coder and an outer CRC coder. Packets that fail the CRC check are dropped, so the tandem channel could be treated as a packet erasure channel. Detailed simulations were done to evaluate the performance over both constant and varying hybrid channel conditions. For the varying channel with delayed feedback information, it was shown that the refined estimation could dramatically improve the performance.

References

[1] H. Yousefi'zadeh, H. Jafarkhani, and F. Etemadi, "Distortion-optimal transmission of progressive images over channels with random bit errors and packet erasures," in *Proc. IEEE Data Compression Conf.*, Mar. 2004, pp. 132–141.

[2] Y. Pei and J. W. Modestino, "Use of concatenated FEC coding for real-time packet video over heterogeneous wired-to-wireless IP networks," in *Proc. Int. Symp. Circuits and Systems*, vol. 2, Mar. 2003, pp. 25–28.

[3] R. Anand, C. Podilchuk, and H. Lou, "Progressive video transmission over a wired-to-wireless network," in *Proc. Vehicular Technology Conf.*, vol. 3, May 2000, pp. 2424–2428.

[4] Y. Shen, P. C. Cosman, and L. B. Milstein, "Video communications with optimal intra/inter-mode switching over wireless Internet," in *Proc. 37th Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, CA, Nov. 2003, pp. 1548–1552.

[5] R. Zhang, S. L. 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.

[6] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 11, pp. 74–90, Nov. 1998.

[7] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

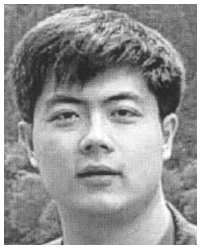
[8] G. Castagnoli, J. Ganz, and P. Graber, "Optimum cyclic redundancy-check codes with 16-bit redundancy," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 111–114, Jan. 2000.

[9] T. V. Ramabadran and S. S. Gaitonde, "A tutorial on CRC computations," *IEEE Micro*, vol. 8, no. 8, pp. 62–75, Aug. 1998.

[10] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.

[11] P. G. Sherwood and K. Zeger, "Progressive image coding on noisy channels," *IEEE Signal Process. Lett.*, vol. 4, pp. 189–191, Jul. 1997.

[12] M. Roder and R. Hamzaoui, "Fast list Viterbi decoding and application for source-channel coding of images," *Konstanz Univ. Rev.*, no. 182, Dec. 2002.



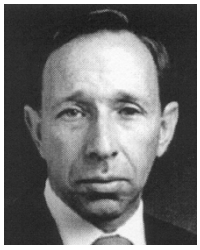
Yushi Shen received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2001, and the M.S. degree in electrical and computer engineering from the University of California at San Diego (UCSD), La Jolla, in 2003, where he is currently pursuing the Ph.D. degree.

He is currently a Graduate Student Researcher at UCSD. His research interests are in the area of video and multimedia communications, communication and information theory, source coding, channel coding, and spread-spectrum.

Pamela C. Cosman (S'88–M'93–SM'00) received the B.S. degree (with honors) in electrical engineering from the California Institute of Technology, Pasadena, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1989 and 1993, respectively.

She was an NSF Postdoctoral Fellow at Stanford University and a Visiting Professor at the University of Minnesota, Minneapolis, from 1993 to 1995. Since July 1995, she has been with the faculty of the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, where she is currently an Associate Professor. Her research interests are in the areas of image and video compression and processing.

Dr. Cosman is a member of Tau Beta Pi and Sigma Xi. She is the recipient of the ECE Departmental Graduate Teaching Award (1996), a Career Award from the National Science Foundation (1996 to 1999), and a Powell Faculty Fellowship (1997 to 1998). She was an Associate Editor of the IEEE COMMUNICATIONS LETTERS from 1998 to 2001, a Guest Editor of the June 2000 special issue of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS on "error-resilient image and video coding," and the Technical Program Chair of the 1998 Information Theory Workshop, San Diego. She is currently an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS and a Senior Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS.



Laurence B. Milstein (S'66–M'68–SM'75–F'85) received the B.E.E. degree from the City College of New York, New York, in 1964, and the M.S. and Ph.D. degrees in electrical engineering from the Polytechnic Institute of Brooklyn, Brooklyn, NY, in 1966 and 1968, respectively.

From 1968 to 1974, he was with the Space and Communication Group of Hughes Aircraft Company, and from 1974 to 1976, he was a Member of the Department of Electrical and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY. Since 1976, he has been with the Department of Electrical and Computer Engineering, University of California at San Diego (UCSD), La Jolla, where he is a Professor and former Department Chairman, working in the area of digital communication theory, with special emphasis on spread-spectrum communication systems. He has also been a Consultant to both government and industry in the areas of radar and communications.

Dr. Milstein was an Associate Editor for communications Theory for the IEEE TRANSACTIONS ON COMMUNICATIONS, an Associate Editor for Book Reviews for the IEEE TRANSACTIONS ON INFORMATION THEORY, an Associate Technical Editor for the IEEE COMMUNICATIONS MAGAZINE, and Editor-in-Chief of the IEEE JOURNAL ON SELECTED AREA IN COMMUNICATIONS. He was the vice President for Technical Affairs in 1990 and 1991 of the IEEE Communications Society and the IEEE Information Theory Society. He has been a member of the IEEE Fellows Selection Committee since 1996, and he currently is the Chair of that committee. He is also the Chair of ComSoc's Strategic Planning Committee. He is a recipient of the 1998 Military Communications Conference Long-Term Technical Achievement Award, Academic Senate 1999 UCSD Distinguished Teaching Award, an IEEE Third Millenium Medal, 2000, and the 2000 IEEE Communication Society Armstrong Technical Achievement Award.