

# Fine-Grained Rate Shaping for Video Streaming over Wireless Networks

**Trista Pei-chun Chen**

NVIDIA Corporation, Santa Clara, CA 95050, USA  
Email: tchen@nvidia.com

**Tsuhan Chen**

Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA  
Email: tsuhan@cmu.edu

Received 30 November 2002; Revised 14 October 2003

Video streaming over wireless networks faces challenges of time-varying packet loss rate and fluctuating bandwidth. In this paper, we focus on streaming precoded video that is both source and channel coded. Dynamic rate shaping has been proposed to “shape” the precompressed video to adapt to the fluctuating bandwidth. In our earlier work, rate shaping was extended to shape the channel coded precompressed video, and to take into account the time-varying packet loss rate as well as the fluctuating bandwidth of the wireless networks. However, prior work on rate shaping can only adjust the rate coarsely. In this paper, we propose “fine-grained rate shaping (FGRS)” to allow for bandwidth adaptation over a wide range of bandwidth and packet loss rate in fine granularities. The video is precoded with fine granularity scalability (FGS) followed by channel coding. Utilizing the fine granularity property of FGS and channel coding, FGRS selectively drops part of the precoded video and still yields decodable bitstream at the decoder. Moreover, FGRS optimizes video streaming rather than achieves heuristic objectives as conventional methods. A two-stage rate-distortion (RD) optimization algorithm is proposed for FGRS. Promising results of FGRS are shown.

**Keywords and phrases:** fine-grained rate shaping, rate shaping, fine granularity scalability, rate-distortion optimization, video streaming.

## 1. INTRODUCTION

Due to the rapid growth of wireless communication, video over wireless network has gained a lot of attention [1, 2, 3]. However, wireless network is hostile for video streaming because of its time-varying error rate and fluctuating bandwidth. Wireless communication often suffers from multipath fading, intersymbol interference, and additive white Gaussian noise, and so forth; thus, the error rate varies over time. In addition, the bandwidth of the wireless network is also time varying. Therefore, it is important for a video streaming system to address these issues.

Joint source-channel coding (JSCC) techniques [4, 5] are often applied to achieve error-resilient video transport with *online coding*. Given the bandwidth requirement, the joint source-channel coder seeks the best allocation of bits for the source and channel coders by varying the coding parameters. However, JSCC techniques are not suitable for streaming *pre-coded* video. The precoded video is both source and channel coded prior to transmission. The network conditions are not known at the time of coding. “Rate shaping,” which was

called dynamic rate shaping (DRS) in [6, 7, 8], was proposed to solve the bandwidth adaptation problem. DRS “shapes,” that is, reduces the bit rate of the single-layered pre source coded (pre-compressed) video to meet the real-time bandwidth requirement. DRS adapts the bandwidth by dropping either high-frequency coefficients of each block or by dropping several blocks in a frame.

To protect the video from transmission errors, source coded video bitstream is often protected by forward error correction (FEC) codes [9]. Redundant information, known as parity bits, is added to the original source coded bits, assuming that systematic codes are adopted. Conventional DRS did not consider shaping for the parity bits in addition to the source coding bits. In our earlier work, we extended rate shaping for streaming the precoded video that is both pre-source-and-channel coded [10]. Such a scheme was called “baseline rate shaping (BRS).” BRS can be applied to precoded video that is source coded with H.263 [11], MPEG-2 [12], or MPEG-4 [13] scalable coding and channel coded with Reed-Solomon codes [9] or rate-compatible punctured convolutional (RCPC) codes [14]. By means of

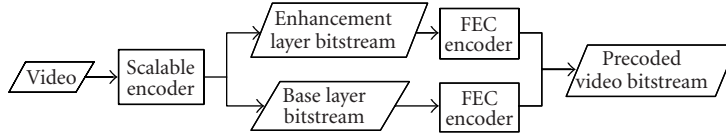


FIGURE 1: System diagram of the precoding process: scalable encoding followed by FEC encoding.

discrete rate-distortion (RD) combination, BRS chooses the best state, which corresponds to a certain way to drop part of the precoded video, to satisfy the bandwidth constraint.

The state chosen by BRS, however, only allows for coarse bandwidth adaptation capability. In this paper, we adopt MPEG-4 fine granularity scalability (FGS) [15] for source coding, and erasure codes [9, 16] for FEC coding. Unlike conventional scalability modes such as signal-to-noise ratio (SNR) scalability, MPEG-4 FGS generates a bitstream that is partially decodable over a wide range of bit rates. The more bits the FGS decoder receives, the better the decoded video quality is. On the other hand, it has been known that erasure codes are still decodable if the number of erasures is within the error/loss protection capability of the codes. Therefore, the proposed “fine-grained rate shaping (FGRS),” which is based on the fine granularity property of FGS and erasure codes, allows for fine rate shaping. Moreover, the proposed FGRS optimizes video streaming rather than achieves heuristic objectives such as unequal packet loss protection (UPP). A two-stage (RD) optimization algorithm is proposed. Note that FGRS focuses on the transport aspect as opposed to the coding aspect of video streaming.

The two-stage RD optimization is designed to find the solution fast and optimally. In Stage 1, a model-based hypersurface is trained with a small set of rate and distortion pairs to approximate the relationship between all rate and distortion pairs. The solution of Stage 1 can be found in the intersection in which the hypersurface meets the bandwidth constraint. In Stage 2, the near-optimal solution from Stage 1 is refined with the hill-climbing-based approach. We can see that Stage 1 aims to find the optimal solution globally with the model-based hypersurface and Stage 2 refines the solution locally.

This paper is organized as follows. In Section 2, we introduce BRS for bandwidth adaptation of the precoded video, which is both scalable and FEC coded. Discrete RD combination algorithm is applied to deliver the best video quality. In Section 3, FGRS is proposed for streaming the FEC coded FGS bitstream. We first formulate the RD optimization problem then provide a two-stage RD optimization algorithm to solve the problem. In Section 4, experiments are carried out to show the superior performance of the proposed FGRS. Concluding remarks are given in Section 5.

## 2. BASELINE RATE SHAPING

We propose to use BRS to reduce the bit rate of the precoded video, which is both source and channel coded, given the

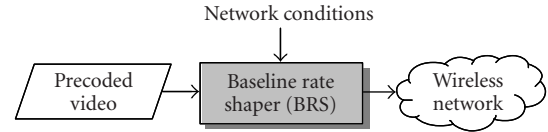


FIGURE 2: Streaming of the precoded video with BRS.

time-varying error rate and bandwidth. Unlike JSCC techniques that allocate the bits for the source and channel coders by varying the coding parameters, BRS performs bandwidth adaptation for the *precoded* video at the time of delivery. BRS decision, as to select which part of the precoded video to drop, varies from time to time. There is no need to re-encode as JSCC with different source and channel coder parameters at later time with a different channel condition. Only a different BRS decision needs to be made for the same bitstream. In addition, rate shaping can be applied to adapt to the network condition of each link along the path of transmission from the sender to the receiver. This is in particular suitable for wireless video streaming since wireless networks are heterogeneous in nature. One single joint source-channel coded bitstream cannot meet the needs of all the links along the path of transmission. Rate shaping can optimize video streaming for each link.

We start by giving the system description of BRS then provide the algorithm for RD optimization.

### 2.1. System description of video streaming with baseline rate shaping

Video streaming consists of three stages from the sender to the receiver: (i) precoding, (ii) streaming with rate shaping, and (iii) decoding, as shown in the following from Figure 1 to Figure 3.

The precoding process (Figure 1) refers to source coding using scalable video coding [11, 12, 13] and FEC coding. Scalable video coding yields prioritized video bitstream. The concept of rate shaping works for any prioritized video bitstream in general.<sup>1</sup> Without loss of generality, we consider SNR scalability. Reed-Solomon codes [9] are used as the FEC codes in this paper.

<sup>1</sup>For example, in DRS [6], bits that carry the information of the low-frequency DCT coefficients are ranked with high priorities in the video bitstream, as opposed to the ones that carry the information of the high-frequency DCT coefficients. By means of data partitioning, the single-layered non-scalable coded bitstream can have different priorities among different segments of the video bitstream.

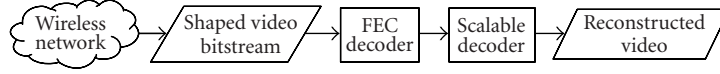


FIGURE 3: System diagram of the decoding process: FEC decoding followed by scalable decoding.

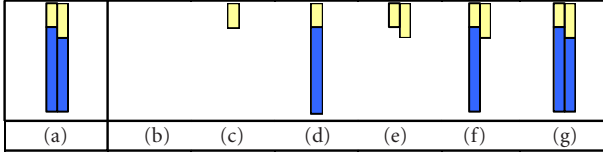


FIGURE 4: (a) All four segments of the precoded video and (b)–(g) valid states of BRS: (b) state (0, 0), (c) state (1, 0), (d) state (1, 1), (e) state (2, 0), (f) state (2, 1), and (g) state (2, 2).

In Figure 2, the pre-source-and-channel coded bitstream is then passed through BRS to adjust its bit rate before being sent to the wireless network. BRS will perform bandwidth adaptation considering the given packet loss rate in an RD optimized manner. The distortion here is described by the mean square error (MSE) of the decoded video. Packet loss rate, instead of bit error rate (BER), is considered since the shaped precoded video will be transmitted in packets.

The decoding process (Figure 3) consists of FEC decoding followed by scalable decoding. The task of rate shaping is performed in the sender and/or midway gateways/routers.

## 2.2. Discrete rate-distortion optimization algorithm

BRS reduces the bit rate of each *decision unit* of the precoded video before it sends the precoded video to the wireless network. A decision unit can be a frame, a macroblock, and so forth, depending on the granularity of the decision. We use a frame as the decision unit herein. BRS performs two kinds of RD optimizations with (i) mode decision and (ii) discrete RD combination, depending on how much delay the rate shaping decisions can allow. We will discuss both mode decision and discrete RD combination in the following.

### (a) BRS by mode decision

We consider the case in which the video is scalable coded into two layers: one base layer and one enhancement layer. These two layers are FEC coded with UPP. That is, the base layer is FEC coded with stronger packet loss protection. Therefore, there are four *segments* in the precoded video. The first segment consists of the bits of the base layer video bitstream (upper-left segment of Figure 4a). The second segment consists of the bits of the enhancement layer video bitstream (upper-right segment of Figure 4a). The third segment consists of the parity bits for the base layer video bitstream (lower-left segment of Figure 4a). The fourth segment consists of the parity bits for the enhancement layer video bitstream (lower-right segment of Figure 4a). BRS decides a subset of the four segments to send. Note that some

constraints need to be imposed for a valid subset. For example, if the segment that consists of the parity bits for the base layer video bitstream is selected, the segment that consists of the bits of the base layer video bitstream must be selected as well. In the case of two layers of video bitstream, six valid combinations are shown in Figures 4b, 4c, 4d, 4e, 4f, and 4g. We call each valid combination a *state*. Each state is represented by a pair of integers  $(x, y)$ , where  $x$  is the number of segments selected counting from the segment consisting of the bits of the base layer, and  $y$  is the number of segments selected counting from the segment consisting of the parity bits for the base layer. Note that  $x$  counts from the base layer because the enhancement layer cannot be decoded without the base layer;  $y$  counts from the base layer because the base layer needs to be protected by parity bits more than the enhancement layer. The two integers  $x$  and  $y$  satisfy the relationship of  $x \geq y$ .

Each state has its RD performance represented by a dot in the RD map, such as the ones shown in Figures 5a and 5b. The state constellations are different for different frames because of variations in video content and packet loss rate for different frames. If the bandwidth requirement is “B” for each frame, BRS performs mode decision by selecting the state that has the least distortion. For example in Figure 5, state (1, 1) of Frame 1 and state (2, 0) of Frame 2 are chosen.

### (b) BRS by discrete RD combination

By allowing some delay in making the rate shaping decision, BRS can optimize video streaming with a better overall quality. By allowing some delay, we mean to accumulate the total bandwidth for a group of pictures (GOP) and to allocate the bandwidth intelligently among frames in a GOP. Video is typically coded with variable bit rate in order to maintain a constant video quality. We want to allocate different numbers of bits for different frames in a GOP to utilize the total bandwidth more efficiently.

Assume that there are  $F$  frames in a GOP and the total bandwidth budget for these  $F$  frames is  $C$ . Let  $x(i)$  be the state (represented by a pair of integers mentioned in (a)) chosen for frame  $i$ , and let  $D_{i,x(i)}$  and  $R_{i,x(i)}$  be the resulting distortion and rate allocated at frame  $i$ , respectively. The goal of the rate shaper is to minimize

$$\sum_{i=1}^F D_{i,x(i)} \quad (1)$$

subject to

$$\sum_{i=1}^F R_{i,x(i)} \leq C. \quad (2)$$

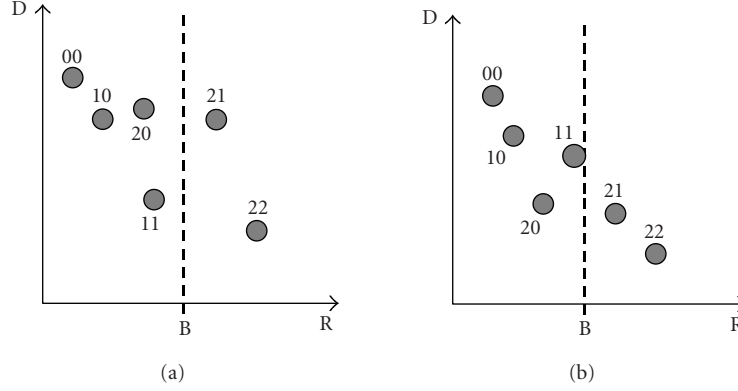
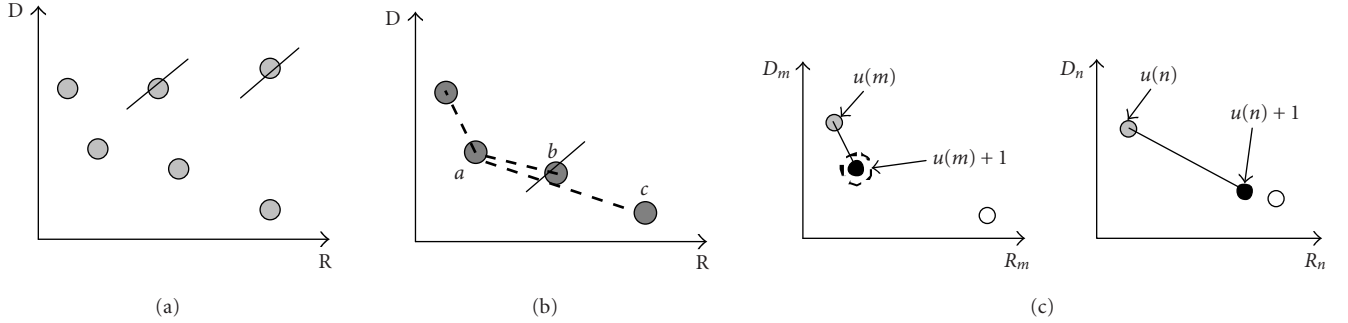


FIGURE 5: RD maps of (a) Frame 1, (b) Frame 2.

FIGURE 6: Discrete RD combination algorithm: (a) and (b) elimination of states inside the convex hull of each frame, and (c) allocation of rate to the frame  $m$  that utilizes the rate more efficiently.

The discrete RD combination algorithm [10, 17] finds the solution by first eliminating the states that are inside the convex hull (Figures 6a and 6b) for each frame. The algorithm then allocates the rate step by step to the frame that utilizes the rate more efficiently. That is, among frame  $m$  and frame  $n$ , if frame  $m$  gives a better ratio than frame  $n$  regarding distortion decrease over rate increase by moving from the current state  $u(m)$  to the next state  $u(m) + 1$ , then the rate is allocated to frame  $m$  (the next state  $u(m) + 1$  of frame  $m$  is circled in Figure 6c) from the available total bandwidth budget. The allocation process continues until the total bandwidth budget has been consumed completely.

### 3. FINE-GRAINED RATE SHAPING (FGRS)

As mentioned, BRS performs the bandwidth adaptation for the precoded video by selecting the best state of each frame at any given packet loss rate. Since the packet loss rate and the bandwidth at any given time could lie in any value over a wide range of values, we want to extend the notion of rate shaping to allow for finer grained decisions. There then prompts the need for source and channel coding techniques that offer fine granularities in terms of video quality and packet loss protection, respectively.

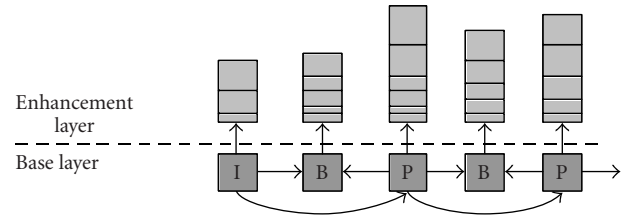


FIGURE 7: Dependency graph of the base layer and FGS enhancement layer. Base layer has temporal prediction with P and B frames. Enhancement layer is encoded with reference to the base layer only.

FGRS has been proposed to provide bitstreams that are still decodable when truncated at any byte interval. That is, FGS enhancement layer bitstream is decodable at any rate provided with an intact base layer bitstream. With such a property, FGS was adopted by MPEG-4 for streaming applications [15]. Figure 7 illustrates two layers of video bitstream: the base layer and the FGS enhancement layer. The base layer is predictive coded while the FGS enhancement layer only uses the corresponding base layer as the reference.

On the other hand, it has been known that the erasure codes provide “fine-grained” packet loss protection with

more and more symbols<sup>2</sup> received at the FEC decoder [9, 16]. The “shaped” erasure code is still decodable if the number of erasures/losses from the transmission is no more than  $d_{\min} - 1$  (number of unsent symbols), where  $d_{\min}$  is the minimum distance of the code. An erasure code can successfully decode the message with the number of erasures up to  $d_{\min} - 1$ , considering both the unsent symbols and the losses taken place in the transmission. Therefore, the more symbols are sent, the better the sent bitstream can cope with the losses. In this paper, we use Reed-Solomon codes as the erasure codes as mentioned in Section 2. In Reed-Solomon codes,  $d_{\min} - 1$  equals  $n - k$ , where  $k$  is the message size in symbols and  $n$  is the code size in symbols. Thus, the partial code with size  $r \leq n$  is still decodable if the number of losses from the transmission is no more than  $r - k$ .

### 3.1. System description of video streaming with fine-grained rate shaping

Similar to BRS, there are three stages for transmitting the video from the sender to the receiver: (i) precoding, (ii) streaming with rate shaping, and (iii) decoding, as shown in Figures 8, 9, and 10.

Through MPEG-4 encoding, two layers of bitstream are generated: one base layer and one FGS enhancement layer (Figure 7). We will consider hereafter the bandwidth adaptation and packet loss resilience for the FGS enhancement layer bitstream only, assuming that the base layer bitstream is reliably transmitted as shown in Figure 9b or is considered by approaches outside the scope of this paper. The general rule is to perform enhancement layer bandwidth adaptation after the base layer is reliably transmitted. The enhancement layer bitstream will not enhance the quality of the video if its reference base layer is corrupted. Otherwise, a drift prevention remedy is needed.

Recalling that we use a frame as the decision unit, we look at the FGS enhancement layer bitstream of a frame. FGS enhancement layer bitstream consists of bits of all the bit planes of this frame. The most significant bit plane (MSB plane) is coded before the less significant bit planes until the least significant bit plane (LSB plane). In addition, since the data in each bit plane is variable-length coded (VLC), if some part of a bit plane is corrupted (due to packet losses), the remaining part of the bit plane becomes undecodable. Bits at the beginning of the enhancement layer bitstream of a frame is more important than the following bits.

Before appending the parity symbols to the FGS enhancement layer bitstream, we first divide all the symbols (in this paper, each symbol consists of 14 bits) for this frame into several *sublayers* (Figure 11a). The way to divide the symbols into sublayers is arbitrary except that the later sublayers are longer in length than the previous ones, that is  $k_1 \geq k_2 \geq \dots \geq k_h$ , since we want to achieve UPP. A natural way to construct the sublayers is to let Sublayer 1 consist of

symbols of the MSB plane, Sublayer 2 consist of symbols of the MSB-1 plane, ..., and Sublayer  $h$  consist of symbols of the LSB plane. Each sublayer is then FEC encoded with erasure codes to the same length  $n$  (Figure 11b). The lower portions of the stripes in Figure 11b consist of the parity symbols. The precoded video is stored and can be used later at the time of delivery.

At the transport stage, FEC coded FGS bitstream is passed through FGRS for bandwidth adaptation, given the current packet loss rate. Note that FGRS is different from JSCC-like approaches, which perform FEC encoding for the FGS bitstream at the time of delivery with a bit allocation scheme that achieves certain objectives, as proposed by Radha and van der Schaar [18, 19, 20] and Yang et al. [21]. That is, FGRS focuses on the transport aspect as opposed to the coding aspect. Moreover, FGRS optimizes video streaming rather than achieves certain objectives. We will elaborate on the optimization algorithm taken later.

### 3.2. Fine-grained rate shaping

With the precoded video, bandwidth adaptation can be implemented naively by dropping the symbols in the order shown in Figure 12a. Given a certain bandwidth requirement for this frame, Sublayer 1 has more parity symbols kept than Sublayer 2 and so on. Shaped bitstream with such a bandwidth adaptation scheme has UPP to the sublayers. We will refer to this method as “UPPRS” herein. However, such UPPRS scheme might not be optimal. We propose FGRS (Figure 12b) for bandwidth adaptation given the current packet loss rate. The darken bars in Figure 12b are selected to be sent by FGRS.

We start from the problem formulation. A FGS enhancement layer bitstream provides better and better video quality as more and more sublayers are correctly decoded. In other words, the total distortion is decreased as more sublayers are correctly decoded. With Sublayer 1 correctly decoded, we reduce the total distortion by  $G_1$  (accumulated *gain* is  $G_1$ ); with Sublayer 2 correctly decoded, we reduce the total distortion further by  $G_2$  (accumulated gain is  $G_1 + G_2$ ), and so on. If Sublayer  $i$  is corrupted, the following Sublayers  $i + 1$ ,  $i + 2$ , and so forth, become undecodable. Note that gain  $G_i$  of Sublayer  $i$  can either (i) be calculated, given the FGS bitstream, after performing partial decoding; or (ii) be embedded in the bitstream as the “metadata.” Gain  $G_i$  of Sublayer  $i$  is different for every frame.

Since the precoded video is transmitted over error prone wireless networks, sublayers are subject to loss and have certain recovery rates given a particular rate shaping decision. The *expected* accumulated gain is then

$$G = \sum_{i=1}^h \left( G_i \prod_{j=1}^i v_j \right), \quad (3)$$

where  $h$  is the number of sublayers of this frame and  $v_j$  is the recovery rate of Sublayer  $j$ , which is a function of  $r_j$  as will be shown later. Sublayer  $j$  is recoverable (or successfully decodable) if the number of erasures resulting from the lossy

<sup>2</sup>“Symbols” are used instead of “bits” since the FEC codes use a symbol as the encoding/decoding unit. In this paper, we use 14 bits for one symbol. The selection of the symbol size in bits depends on the user.



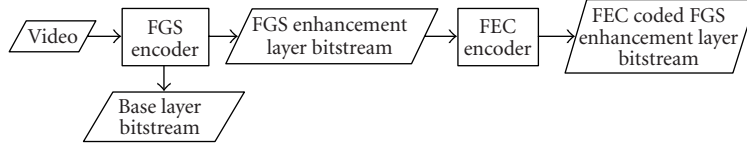


FIGURE 8: System diagram of the precoding process: FGS encoding followed by FEC encoding.



FIGURE 9: Transport of the precoded bitstreams: (a) transport of the FEC coded FGS enhancement layer bitstream with rate shaper via the wireless network and (b) transport of the base layer bitstream via the reliable channel.

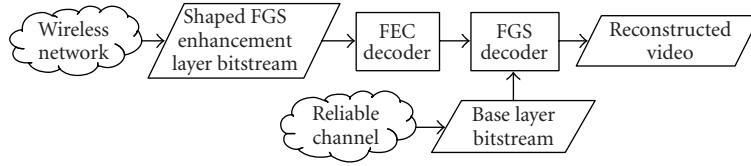


FIGURE 10: System diagram of the decoding process: FEC decoding followed by FGS decoding.

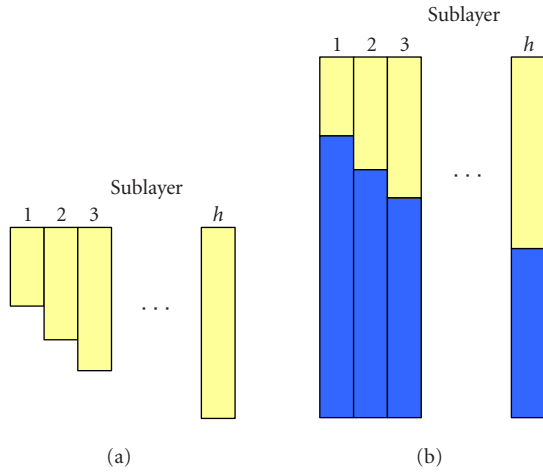


FIGURE 11: Precoded video: (a) FGS enhancement layer bitstream in sublayers and (b) FEC coded FGS enhancement layer bitstream.

transmission is no more than  $r_j - k_j$ ;  $k_j$  is the message (the symbols from the FGS bitstream) size of Sublayer  $j$ , and  $r_j$  is the number of symbols selected to be sent for Sublayer  $j$ . The recovery rate  $v_j$  is the summation of the probabilities that no

loss occur, one erasure occurs, and so on until  $r_j - k_j$  erasures occur:

$$v_j = \sum_{l=0}^{r_j - k_j} p\{l\}, \quad j = 1 \sim h, \quad (4)$$

where  $l$  is the number of erasures that occur. If each erasure occurs as a Bernoulli trial with probability  $e_m$ , the probability of having  $l$  erasures out of  $r_j$  symbols is

$$p\{l\} = \binom{r_j}{l} (e_m)^l (1 - e_m)^{r_j - l}. \quad (5)$$

The symbol loss rate can be derived from the packet loss rate as  $e_m = 1 - (1 - e_p)^{m/s}$ , where  $s$  is the packet size and  $m$  is the symbol size in bits. Depending on the error model (Bernoulli trial, two-state Markov model, etc.), (5) can be replaced with different probability functions.

By choosing different combinations of the number of symbols for each sublayer, the expected accumulated gain will be different. The rate-shaping problem can then be formulated as follows: maximize

$$G = \sum_{i=1}^h \left( G_i \prod_{j=1}^i v_j \right) \quad (6)$$

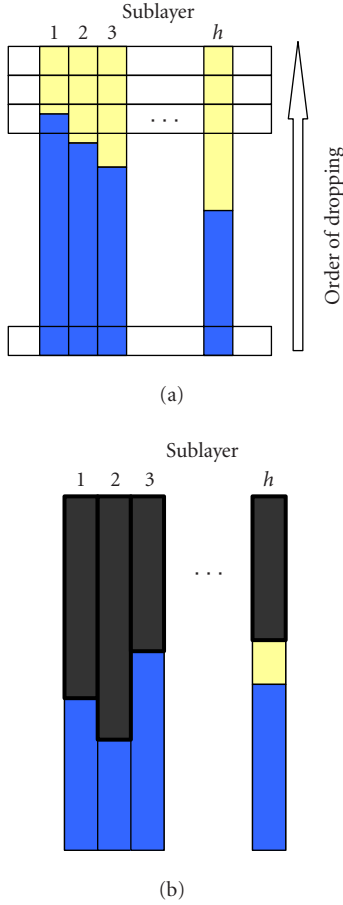


FIGURE 12: Bandwidth adaptation with (a) UPPRS and (b) FGRS. The part represented by darken bars are selected to be sent by FGRS.

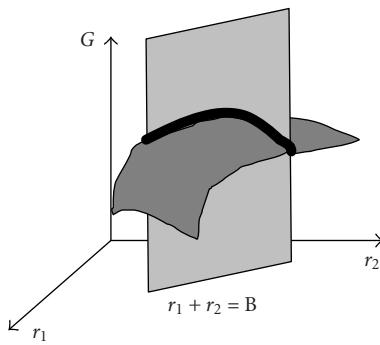


FIGURE 13: Intersection of the model-based hypersurface (dark surface) and the bandwidth constraint (gray plane), illustrated with  $h = 2$ .

subject to

$$\sum_{i=1}^h r_i \leq B. \quad (7)$$

To solve the problem, an exhausted search on all possible combinations of  $\mathbf{r} = [r_1 \ r_2 \ \dots \ r_h]$  or hill-climbing-based approaches as described in [22, 23, 24], where RD optimization is made for automatic repeat request (ARQ) decisions, can be performed. We propose in this paper a two-stage RD optimization algorithm. The two-stage RD optimization algorithm first finds the *near-optimal* solution fast. The near-optimal solution is then *refined* by the hill climbing approach. The proposed two-stage RD optimization is different from [22, 23, 24] in three folds. First, the model-based Stage 1 allows us to examine fewer samples from all operational RD states. Only a small set of samples are needed to train the model needed for RD optimization. Second, the proposed distortion measure (or “expected accumulated gain” in the terminology of the paper) accounts for the effects of packet loss as well as the channel codes by means of recovery rates. Finally, the proposed two-stage RD optimization algorithm can avoid the problem that the solution could be trapped in the local maximum or reach the local maximum too slow. Due to the complexity consideration, Stage 2 can be skipped. Stage 1 does not just serve as a simple initialization stage. It can already find a near-optimal solution.

Packetization is performed after rate shaping. That is, symbols are grouped into packets after the decision of  $\mathbf{r} = [r_1 \ r_2 \ \dots \ r_h]$  has been made. Similar packetization method can be found in [20], while [25] applied bit errors on the bitstream directly. The packets can be sent with “user datagram protocol (UDP)” [26]. It is assumed that any error in the packet will result in a packet loss. More considerations on packetization can be found in UDP-Lite [27]. This paper focuses on rate shaping, assuming that the network condition is provided regardless of which specific packetization method is used.

#### (1) Two-stage RD optimization: Stage 1

We can see from (3) and (4) that the expected accumulated gain  $G$  is related to  $\mathbf{r} = [r_1 \ r_2 \ \dots \ r_h]$  implicitly through the recovery rates  $\mathbf{v} = [v_1 \ v_2 \ \dots \ v_h]$ . We can instead find a model-based hypersurface that explicitly relates  $\mathbf{r}$  and  $G$ . The model parameters can be trained from a set of training data  $(\mathbf{r}, G)$ , where  $\mathbf{r}$  values are chosen by the user and  $G$  values can be computed from (3) and (4). The optimal solution is in the intersection (Figure 13) in which the model-based hypersurface meets the bandwidth constraint. A complex model, with a lot of parameters, can be used to describe as close as possible the true distribution of the RD states. The solution obtained with this model will be as close to optimal as possible. However, the number of  $(\mathbf{r}, G)$  pairs needed to train the model-based hypersurface increases with the number of parameters.

In this paper, we use a quadratic equation to describe the relation between  $\mathbf{r}$  and  $G$  as follows:

$$\hat{G} = \sum_{i=1}^h a_i r_i^2 + \sum_{i,j=1, i \neq j}^h b_{ij} r_i r_j + \sum_{i=1}^h c_i r_i + d. \quad (8)$$

To distinguish the hypersurface modeled  $\hat{G}$  from the real expected gain  $G$ , we denote the former with a “head” sign. The model parameters  $a_i$ ,  $b_{ij}$ ,  $c_i$ , and  $d$  are trained differently for each frame. They can be solved by surface fitting with a set of training data  $(\mathbf{r}, G)$  obtained by (3) and (4). For example, the parameters can be computed by

$$\begin{pmatrix} a_i's \\ b_{ij's} \\ c_i's \\ d \end{pmatrix} = (R^T R)^{-1} R^T \begin{pmatrix} {}^1G \\ {}^2G \\ \vdots \\ \Xi G \end{pmatrix}, \quad (9)$$

where the left super index of  $G$  is the index of the training data and  $R$  is a matrix consisting  $\Xi$  rows of  $(r_i^2's, r_i r_j's, r_i's, 1)$ .

The complexity of computing  $a_i's$ ,  $b_{ij's}$ ,  $c_i's$ , and  $d$  relates to the number of parameters  $h^2 + h + 1$  and the number of training data  $\Xi$ , using (9). Note that the number of training data  $\Xi$  is in general much greater than the number of parameters  $h^2 + h + 1$ . Thus, a more complex model, such as a third-order model with  $h^3 + h^2 + h + 1$  parameters, is not suitable since it requires much more training data than a quadratic model. In addition, second-order Taylor expansion can nicely approximate most functions. Equation (8) can be seen as a second-order approximation to (3). To reduce the computation complexity in reality, we can also choose a smaller  $h$  if the precoding process is also under our control (which is outside the scope of the rate shaper).

With (8), the near-optimal solution can be obtained by the use of Lagrange multiplier as follows:

$$J = \left( \sum_{i=1}^h a_i r_i^2 + \sum_{i,j=1, i \neq j}^h b_{ij} r_i r_j + \sum_{i=1}^h c_i r_i + d \right) + \lambda \left( \sum_{i=1}^h r_i - B \right). \quad (10)$$

By  $\partial J / \partial r_i = 0$ , we get

$$r_i = \frac{-1}{2a_i} \left( \sum_{j=1, j \neq i}^h b_{ij} r_j + c_i + \lambda \right), \quad (11)$$

where

$$\lambda = \frac{2B + \sum_{i=1}^h (1/a_i) \left( \sum_{j=1, j \neq i}^h b_{ij} r_j + c_i \right)}{-\sum_{i=1}^h (1/a_i)}. \quad (12)$$

The near-optimal solution can be solved recursively using (11) and (12), starting from the initial condition that all sublayers are allocated with equal number of symbols,  $r_1 = r_2 = \dots = r_h = B/h$ .

## (2) Two-stage RD optimization: Stage 2

Stage 1 of the two-stage RD optimization gives a near-optimal solution. The solution can be refined by a hill-climbing-based approach (Algorithm 1). The solution from Stage 1 is perturbed in Stage 2 in order to yield a larger ex-

```

While (stop == false)
   $z^i = r^i$  for all  $i = 1 \sim h$ 
  For ( $j = 1; j \leq h; j++$ )
    For ( $k = 1; k \leq h; k++$ )
       $z^k = z^k + \text{delta}$  for  $k == j$  //Increase sublayer  $j$ 
       $z^k = z^k - \text{delta} / (h - 1)$  for  $k! = j$  //Decrease others
    End
    Evaluate  $G_j$ 
  End
  Find the  $j^*$  with the largest  $G_{j^*}$ .
  For ( $i = 1; i \leq h; i++$ )
     $r^i = r^i + \text{delta}$  for  $i == j^*$ 
     $r^i = r^i - \text{delta} / (h - 1)$  for  $i! = j^*$ 
  End
  Calculate the stop criterion.
End

```

ALGORITHM 1: Pseudocodes of hill-climbing algorithm.

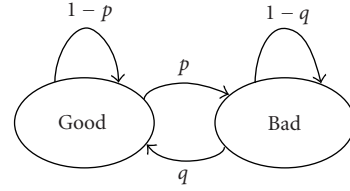


FIGURE 14: Two-state Markov chain for bit error simulation.

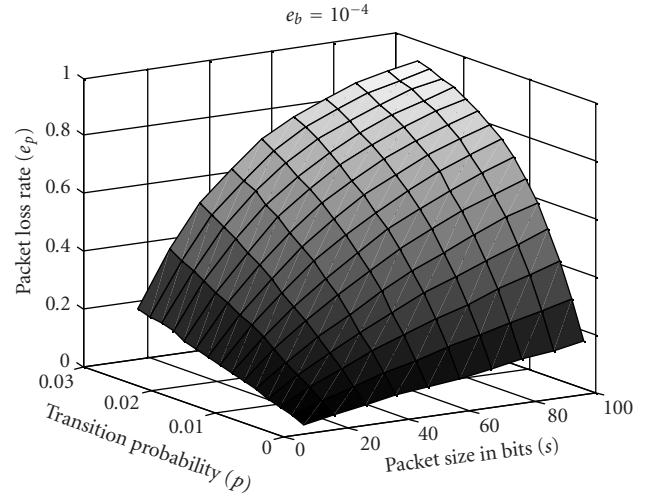


FIGURE 15: Packet loss rate as a function of the transition probability and the packet size.

pected accumulated gain. The process can be iterated until the solution reaches a stopping criterion such as the convergence.

The idea of allocating bandwidth optimally for sublayers can be extended to a higher level to allocate bandwidth efficiently among frames in a GOP. The problem formulation is



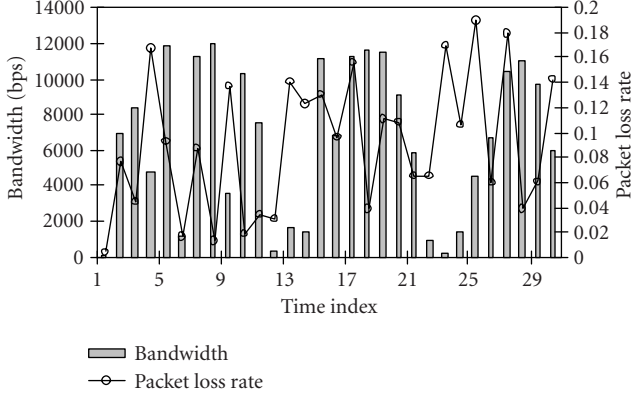


FIGURE 16: Network conditions: bandwidth and packet loss rate fluctuations.

slightly different from the original (6) as follows: maximize

$$G = \sum_{m=1}^F \left[ \sum_{i=1}^h \left( G_{mi} \prod_{j=1}^i v_{mj} \right) \right] \quad (13)$$

subject to

$$\sum_{m=1}^F \sum_{i=1}^h r_{mi} \leq C, \quad (14)$$

where  $F$  is the number of frames in a GOP. FGRS will incur delay with duration of  $F$  frames if it allows for optimization among frames in a GOP.

To summarize, the proposed FGRS achieves the best streaming performance for FEC coded FGS bitstream with the two-stage RD optimization. The two-stage RD optimization obtains the optimal solution by first finding the near-optimal solution, then refining the solution with a hill-climbing-based approach.

## 4. EXPERIMENT

We start by describing the wireless network simulation for the experiment. We then compare the proposed FGRS with the naive UPPRS described in Figure 12a.

### 4.1. Experiment setup

Wireless networks are generally associated with time-varying packet loss rate and fluctuating bandwidth. The packet loss rate and bandwidth vary at each time interval. We simulate random bandwidth fluctuation according to an autoregressive (AR) process [28] and use a two-state Markov model [29, 30] to simulate the bursty bit errors. The two-state Markov model is also adopted by [31, 32]. “Good” and “Bad” in Figure 14 correspond to error free and erroneous states of a bit, respectively. The BER  $e_b$  is related to the transition probabilities  $p$  and  $q$  by  $e_b = p/(p + q)$ .

Since the coded bitstream is transmitted in packets, let us look at how the packet loss rate  $e_p$  relates to the transition

TABLE 1: PSNR gains in Y, U, and V components with sequences Akiyo, Foreman, and Stefan.

PSNR gain (dB)	Y component	U component	V component
Akiyo	1.38	1.28	0.87
Foreman	0.86	0.44	0.52
Stefan	0.76	0.34	0.38

probability  $p$  and the BER  $e_b$ . With BER  $e_b$ , transition probability  $p$ , and packet size  $s$ , the packet loss rate of the  $s$ -bit packet is

$$e_p = 1 - (1 - e_b)(1 - p)^{s-1}. \quad (15)$$

We observe two properties from (15) given the same BER  $e_b$ : (i) the smaller the transition probability  $p$ , the smaller the packet loss rate  $e_p$ , and (ii) the smaller the packet size  $s$ , the smaller the packet loss rate  $e_p$ . These two properties are shown in Figure 15 with  $e_b = 10^{-4}$ .

Besides the two properties we have just seen, it is also known that to detect the loss of packets, some information such as the packet number has to be added to each packet. The smaller the packet is, the heavier the overhead is. Therefore, it is a trade-off between the selection of the packet size and the resulting packet loss rate. We use  $s = 280$  (bits) in this paper. Users can select the packet size  $s$  according to real system consideration using (15).

The time-varying bandwidth is simulated pseudorandomly according to an AR process. The bandwidth available at current time  $t$  is fed to FGRS optimization of time  $t + 1$  in order to simulate the delay nature of the network feedback. Such delay in feedback will not affect too much the performance since the bandwidth requirements of the two consecutive frames are closely related, given the AR assumption. Example traces of simulated packet loss rate and bandwidth observed at the rate shaper are shown in Figure 16. The packet loss rate is plotted using the line and the bandwidth is illustrated using the vertical bars. Each interval in the axis of time index represents 0.33 seconds.

The test video sequences are “Akiyo,” “Foreman,” and “Stefan” in common intermediate format (CIF) (Figures 17a, 17b, and 17c). The frame rate is three frames/s.

### 4.2. Experiment result

Results for sequence Akiyo are shown in Figures 18 and 19. Results for sequence Foreman is shown in Figures 20 and 21. Results for sequence Stefan is shown in Figures 22 and 23. The overall PSNR performance for all the three test sequences are listed in Figure 24 and Table 1. Results for different wireless channel conditions are shown in Figure 25.

Figures 18, 20, and 22 show how bit allocation with UPPRS and FGRS is done in bytes (converted from number of symbols) for each sublayer. After bit allocation, the number of symbols to send is constrained to be at least  $k_i$  for each sublayer (i.e., to satisfy  $r_i \geq k_i$ ) by moving the number of symbols allocated for the higher sublayers to the lower layers that does not satisfy  $r_i \geq k_i$  as shown in Algorithm 2.



FIGURE 17: Test video sequences in CIF: (a) Akiyo, (b) Foreman, and (c) Stefan.

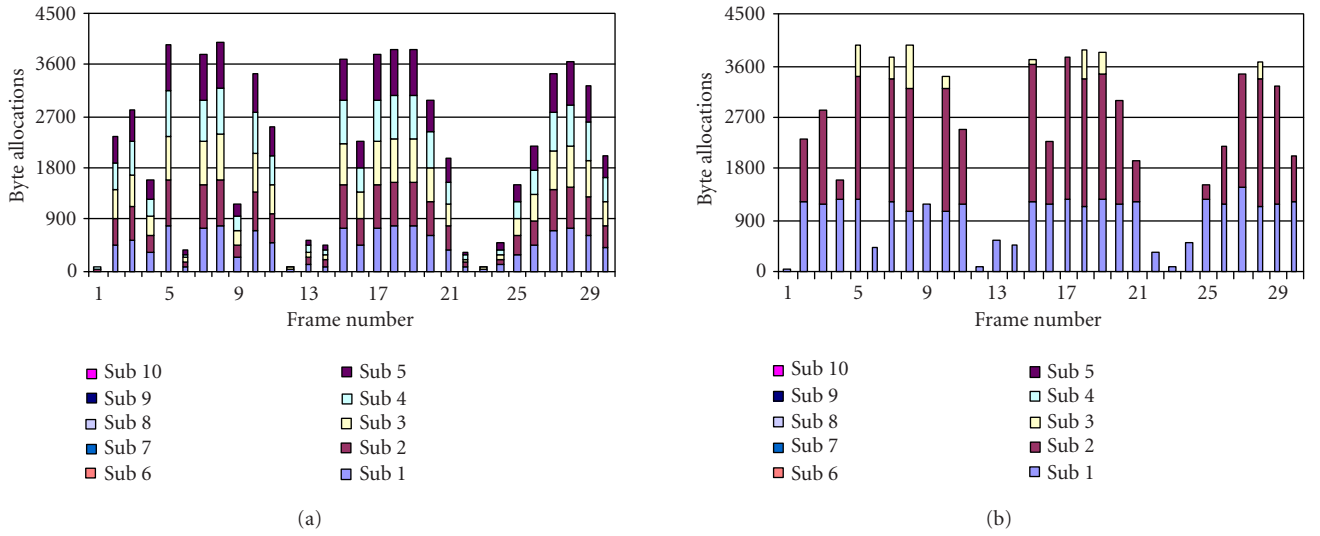


FIGURE 18: Sublayer byte allocations with sequence Akiyo by (a) UPPRS and (b) FGRS.

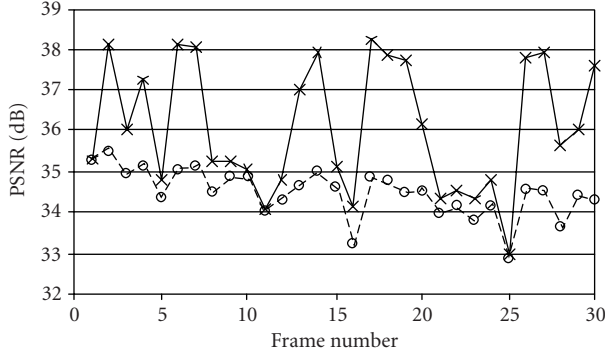
With limited bandwidth, FGRS allocates enough bytes to Sublayer 1 (indicated as sub 1 in the figures) first, than to Sublayer 2, and so on. Allocating enough bytes to a sublayer means providing enough packet loss protection, but not allocating too many bytes as to include too much redundancy. The bit allocation process happens automatically by the proposed two-stage RD optimization, considering the current packet loss rate and the bandwidth requirement.

From the frame-by-frame PSNR performance in Figures 19, 21, and 23, we see that the proposed FGRS provides superior results to UPPRS. Comparing performance with different sequences, the PSNR improvement of FGRS over UPPRS is the most significant in sequence Akiyo, followed by sequence Foreman and Stefan. Sequence Stefan is the most challenging one with the most complex scene and the highest motion. The source coding rates of the FGS enhancement layer bitstream of Akiyo, Foreman, and Stefan are 354.69 kbps, 747.74 kbps, and 975.70 kbps. Hence, given the

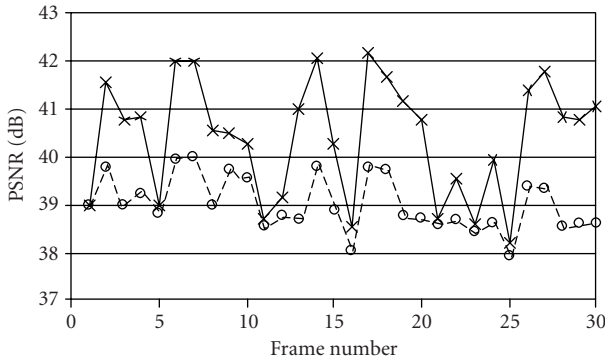
same amount of bits allocated by FGRS, the PSNR of sequence Stefan is the smallest among the three. Considering the gain in the Y component, FGRS yields 0.76 dB to 1.38 dB improvement compared to UPPRS as shown in Table 1.

To validate the performance of the proposed algorithm, the performance in terms of the overall PSNR of the Y components at various wireless channel conditions is shown in Figure 25, where we consider a two-state Markov model at various speeds and SNRs [29]. Figure 25a shows the 3D plots of the overall PSNR. At all wireless channel conditions, FGRS outperforms UPPRS.

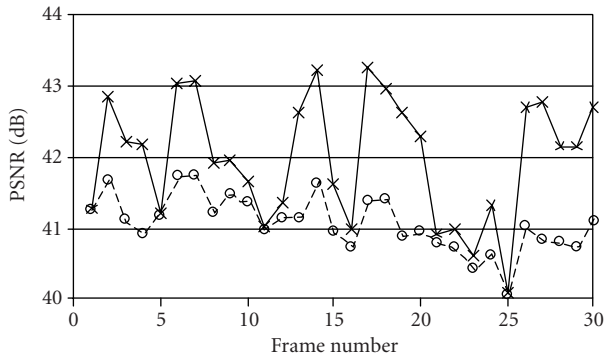
Figure 25b shows the overall PSNR at various speeds at SNR = 10 dB. Fixed SNR value gives the same BER of the wireless channel. The higher the speed is, the more bursty the bit error of the wireless channel is. In other words, the larger the transition probability is. From the results, we see that the PSNR drops as the speed increases. The higher the transition probability is, the higher the packet loss rate is, given



(a)



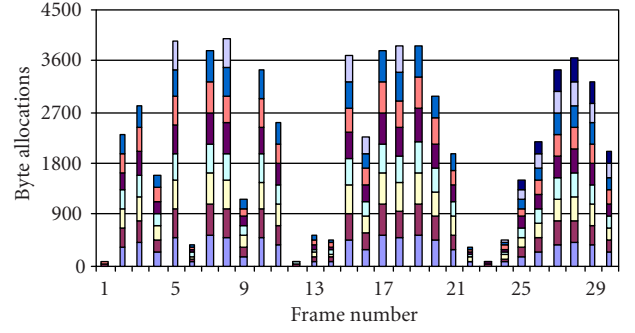
(b)



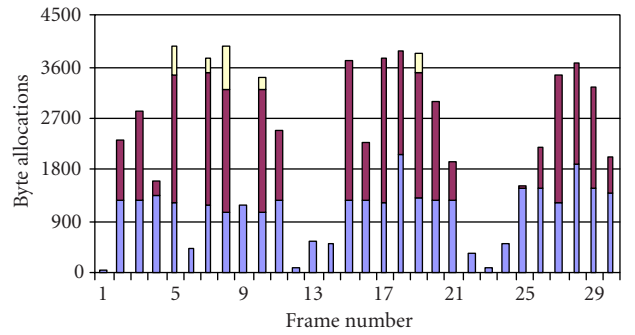
(c)

FIGURE 19: Frame-by-frame PSNR of UPPRS and FGRS with sequence Akiyo: (a) PSNR of the Y component, (b) PSNR of the U component, and (c) PSNR of the V component.

the same BER. Higher packet loss rate has the effect of requiring more parity bits in the shaped bitstream, and higher



(a)



(b)

FIGURE 20: Sublayer byte allocations with sequence Foreman by (a) UPPRS and (b) FGRS.

probability of corrupting the packets that carries the shaped bitstream, thus, the PSNR value is lower.

Figure 25c shows the overall PSNR at various SNRs at speed = 10 km/h. Fixed speed gives the same burstiness of the bit errors of the wireless channel. The larger the SNR is, the smaller the BER is. We see from the results that the PSNR value increases with SNR. Smaller packet loss rate then leads to a higher PSNR.

Optimization for video streaming needs to be real time. As mentioned, in the training process for the model-based hypersurface, only a few number of operational RD states need to be examined, which saves the time. Thus, the two-stage RD optimization is preferred over the hill-climbing-based approach. In addition, as mentioned in Section 3.2, Step 2 can be skipped without too much performance degradation.

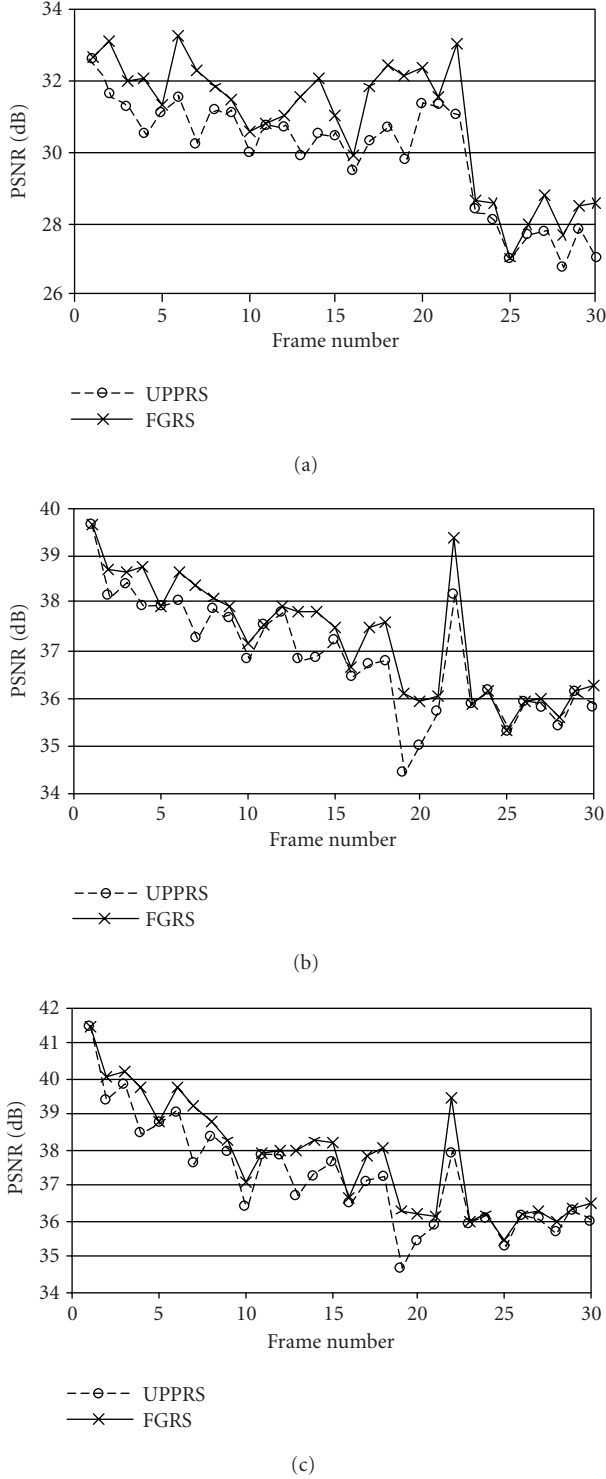


FIGURE 21: Frame-by-frame PSNR of UPPRS and FGRS with sequence Foreman: (a) PSNR of the Y component, (b) PSNR of the U component, and (c) PSNR of the V component.

## 5. CONCLUSION

We proposed in this paper a novel FGRS approach to perform bandwidth adaptation for the precoded video, which

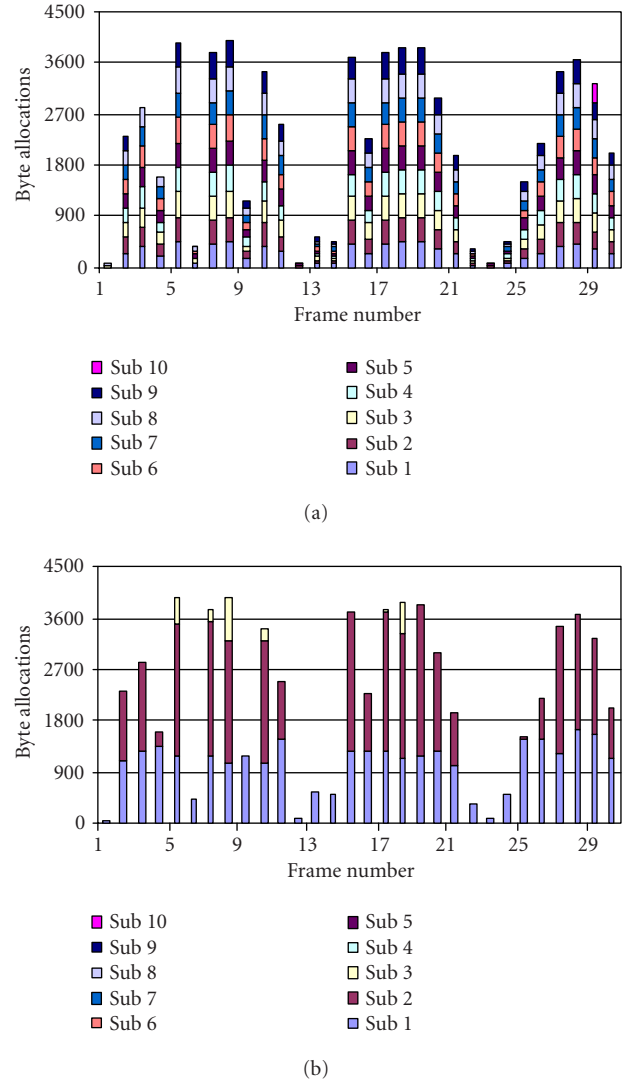
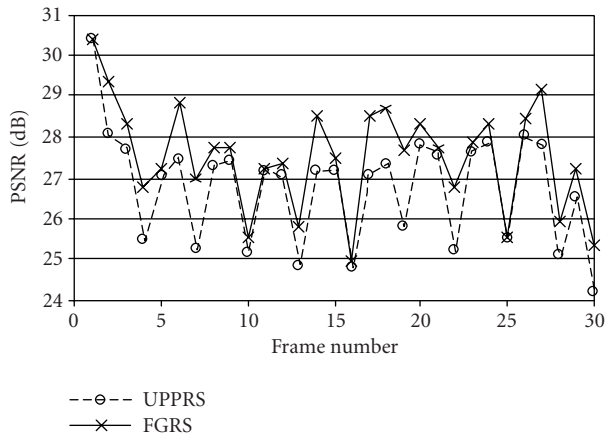


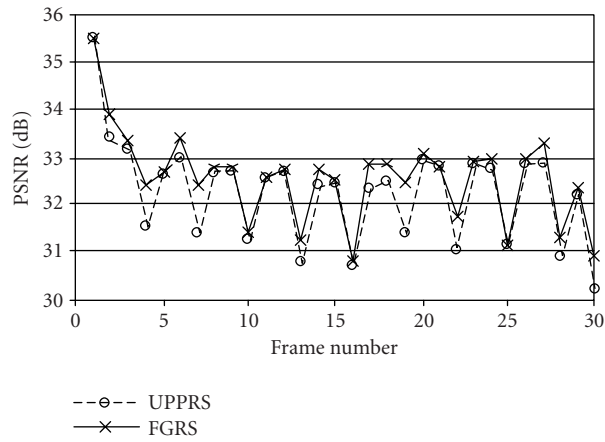
FIGURE 22: Sublayer byte allocations with sequence Stefan by (a) UPPRS and (b) FGRS.

is both FGS coded and FEC coded. FGRS utilizes the fine granularity property of FGS and FEC. Moreover, FGRS optimizes video streaming rather than achieves heuristic objectives. A two-stage rate-distortion (RD) optimization algorithm is used. The two-stage RD optimization algorithm finds the solution efficiently. The proposed FGRS outperforms UPPRS.

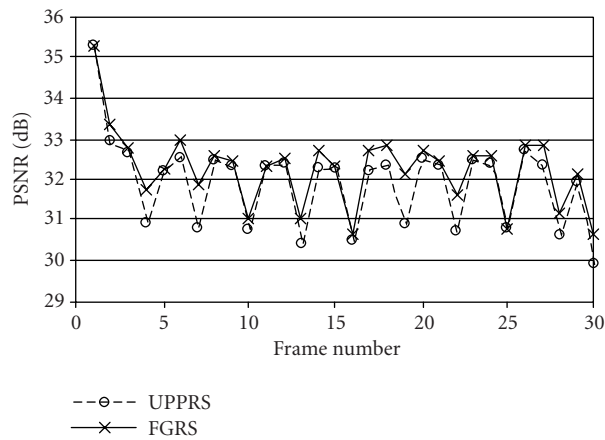
The novelty of the paper lies in three aspects. Although FGS has been proposed to provide fine granularity for pre-compressed video, none of the prior works has shown how to adapt the rate of the FGS bitstream that is protected by the FEC codes. Note that related work performs FEC encoding for the FGS bitstream at the time of delivery. Secondly, we formulate the FGRS problem as an RD optimization problem, while the work by van der Schaar and Radha [20] is not optimized but to achieve a certain target recovery rate. In addition, the distortion measure, which is called



(a)

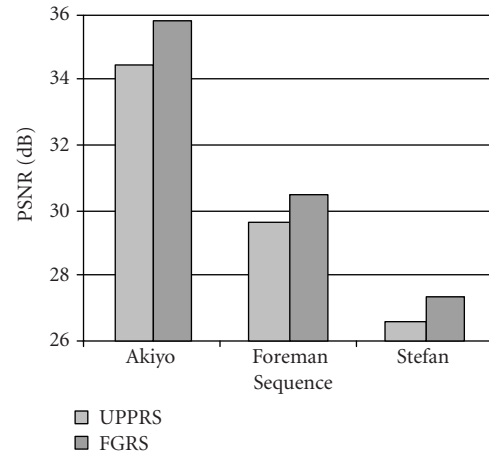


(b)

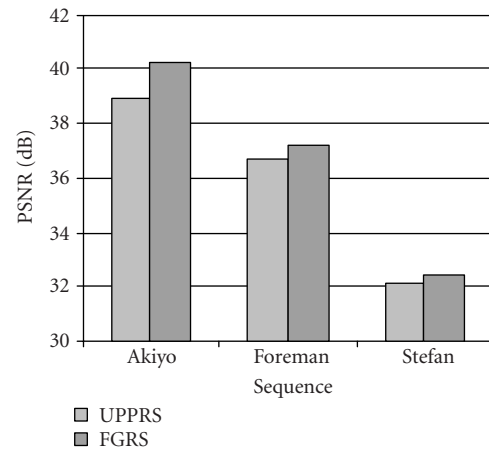


(c)

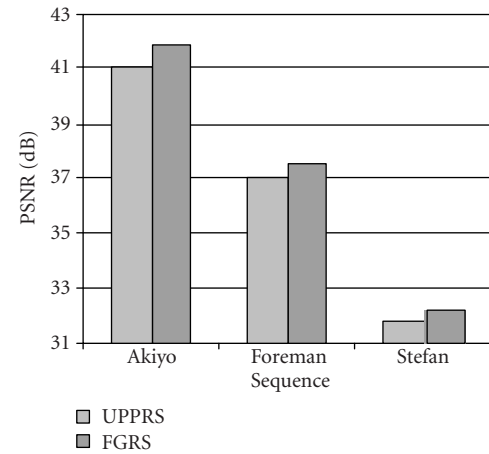
FIGURE 23: Frame-by-frame PSNR of UPPRS and FGRS with sequence Stefan: (a) PSNR of the Y component, (b) PSNR of the U component, and (c) PSNR of the V component.



(a)



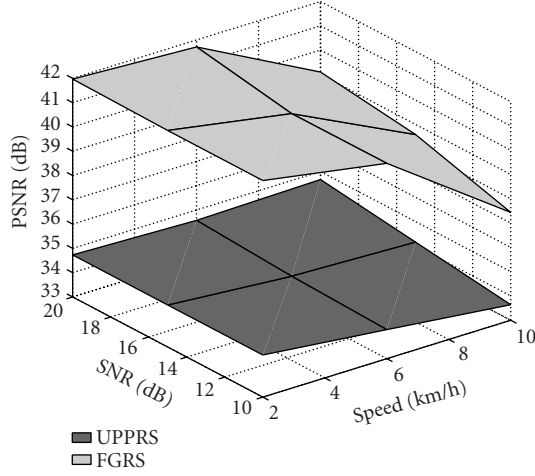
(b)



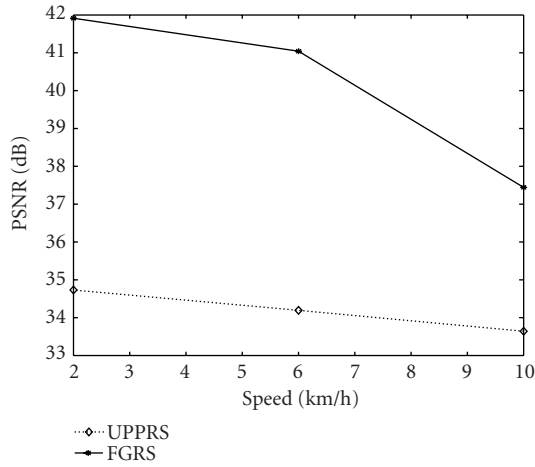
(c)

FIGURE 24: Overall PSNR of UPPRS and FGRS with sequences Akiyo, Foreman, and Stefan: (a) PSNR of the Y component, (b) PSNR of the U component, and (c) PSNR of the V component.

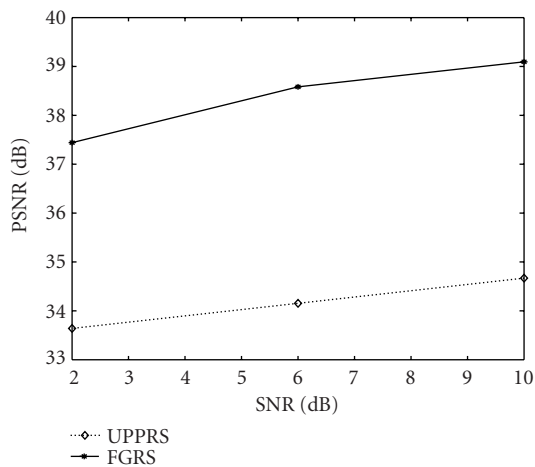




(a)



(b)



(c)

FIGURE 25: Performance (PSNR of the Y component) of all methods at various wireless channel conditions for sequence Foreman: (a) 3D view of PSNR at various speeds and SNRs; (b) PSNR at various speeds; (c) PSNR at various SNRs.

```

For ( $i = 1; i \leq h; i++$ )
  If  $r_i < k_i$ 
     $a = k_i - r_i$ 
    //the difference needed to satisfy  $r_i > k_i$ 
     $b = c = 0$ 
    For ( $j = h; j \geq 1 \ \& \ c < a; j--$ )
       $b = r_j > a ? a : r_j$ 
      //the symbols got from Sublayer  $j$ 
       $c += b$ 
       $r_j -= b$ 
    End
     $r_i = k_i$ 
  End
End
End

```

ALGORITHM 2: Pseudocodes satisfying  $r_i \geq k_i$  after bit allocation.

“gain” in the paper, is derived from the current packet loss rate in addition to the video characteristics. The gain is defined as the expected gain given the current packet loss rate. Prior work of DRS defines the distortion measure solely from the video characteristics. Thirdly, the RD optimization problem is solved by the proposed two-stage RD optimization algorithm, which can achieve the optimal solution fast. It is crucial that optimization for video streaming is done in real time.

Future work includes considering the smoothness criterion in FGRS optimization such as [33] to smooth the fluctuating PSNR resulted from the time-varying network conditions. Such fluctuation is not inherent from the FGRS algorithm. We can also investigate more the effect of outdated network information on FGRS, in addition to the simulation done in this paper by delaying the network bandwidth feedback. Moreover, deploying FGRS in a large network system, such as the “end system multicast (ESM)” [34] system, can be an exciting future research direction.

## ACKNOWLEDGMENTS

This work was supported in part by Industrial Technology Research Institute. The authors would like to acknowledge the suggestions of Professor Mihaela van der Schaar, University of California at Davis, Professor Jose Moura and Professor Rohit Negi, Carnegie Mellon University, Professor Alex Eleftheriadis and Professor Shih-Fu Chang, Columbia University, Professor Antonio Ortega, University of Southern California, and the reviewers of the paper.

## REFERENCES

- [1] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, “Error resilient video coding techniques,” *IEEE Signal Processing Magazine*, vol. 17, no. 4, pp. 61–82, 2000.
- [2] J. Cabrera, A. Ortega, and J. I. Ronda, “Stochastic rate-control of video coders for wireless channels,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 496–510, 2002.

- [3] Z. He, J. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 511–523, 2002.
- [4] G. Cheung and A. Zakhor, "Bit allocation for joint source/channel coding of scalable video," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 340–356, 2000.
- [5] L. P. Kondi, F. Ishtiaq, and A. K. Katsaggelos, "Joint source-channel coding for motion-compensated DCT-based SNR scalable video," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 1043–1052, 2002.
- [6] A. Eleftheriadis and D. Anastassiou, "Meeting arbitrary QoS constraints using dynamic rate shaping of coded digital video," in *Proc. 5th International Workshop on Networking and Operating System Support for Digital Audio and Video*, pp. 95–106, Durham, NH, USA, April 1995.
- [7] W. Zeng and B. Liu, "Rate shaping by block dropping for transmission of MPEG-precoded video over channels of dynamic bandwidth," in *Proc. 4th ACM International Conference on Multimedia*, pp. 385–393, Boston, Mass, USA, November 1996.
- [8] S. Jacobs and A. Eleftheriadis, "Streaming video using dynamic rate shaping and TCP congestion control," *Journal of Visual Communication and Image Representation*, vol. 9, no. 3, pp. 211–222, 1998.
- [9] S. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice Hall, Englewood Cliffs, NJ, USA, 1995.
- [10] T. P.-C. Chen and T. Chen, "Adaptive joint source-channel coding using rate shaping," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Orlando, Fla, USA, May 2002.
- [11] D. S. Turaga and T. Chen, "Fundamentals of video compression: H.263 as an example," in *Compressed Video over Networks*, M.-T. Sun and A. R. Reibman, Eds., pp. 3–34, Marcel Dekker, NY, USA, 2001.
- [12] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*, Chapman & Hall, NY, USA, 1997.
- [13] Motion Pictures Experts Group, "Overview of the MPEG-4 standard," ISO/IEC JTC 1/SC 29/WG 11 N 2459, 1998.
- [14] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Communications*, vol. 36, no. 4, pp. 389–400, 1988.
- [15] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301–317, 2001.
- [16] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, 1997.
- [17] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.
- [18] H. M. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 53–68, 2001.
- [19] M. van der Schaar and H. M. Radha, "A hybrid temporal-SNR fine-granular scalability for internet video," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 318–331, 2001.
- [20] M. van der Schaar and H. M. Radha, "Unequal packet loss resilience for fine-granular-scalability video," *IEEE Trans. Multimedia*, vol. 3, no. 4, pp. 381–394, 2001.
- [21] X. K. Yang, C. Zhu, Z. G. Li, G. N. Feng, S. Wu, and N. Ling, "A degressive error protection algorithm for MPEG-4 FGS video streaming," in *Proc. 2002 IEEE International Conference on Image Processing*, pp. 737–740, Rochester, NY, USA, September 2002.
- [22] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 819–828, 2000.
- [23] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," submitted to *IEEE Trans. Multimedia*.
- [24] J. Chakareski, P. A. Chou, and B. Aazhang, "Computing rate-distortion optimized policies for streaming media to wireless clients," in *Proc. Data Compression Conference*, Snowbird, Utah, USA, April 2002.
- [25] G. Cote, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 952–965, 2000.
- [26] J. Postel, "User datagram protocol (RFC 768)," Internet Engineering Task Force, Internet draft, RFC-768, August, 1980, <http://www.ietf.org/rfc/rfc768.txt>.
- [27] L.-A. Larzon, M. Degermark, and S. Pink, "Efficient use of wireless bandwidth for multimedia applications," in *Proc. 6th IEEE International Workshop on Mobile Multimedia Communications*, pp. 187–193, San Diego, Calif, USA, November 1999.
- [28] A. J. Ganesh, "Estimating effective bandwidths from traffic data," in *Proc. IEEE Conference on Global Communications*, pp. 654–658, London, UK, November 1996.
- [29] J.-P. Ebert and A. Willig, "A Gilbert-Elliott bit error model and the efficient use in packet level simulation," Tech. Rep. TKN-99-002, Telecommunication Networks Group, Technical University of Berlin, Berlin, 1999.
- [30] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modeling of the temporal dependence in packet loss," in *18th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 345–352, NY, USA, March 1999.
- [31] S. A. Khayam, S. S. Karande, M. Krappel, and H. M. Radha, "Cross-layer protocol design for real-time multimedia applications over 802.11b networks," in *IEEE 2003 International Conference on Multimedia and Expo*, pp. 425–428, Baltimore, Md, USA, July 2003.
- [32] F. Yang, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "An end-to-end TCP-friendly streaming protocol for multimedia over wireless internet," in *IEEE 2003 International Conference on Multimedia and Expo*, pp. 429–432, Baltimore, Md, USA, July 2003.
- [33] X. M. Zhang, A. Vetro, Y. Q. Shi, and H. Sun, "Constant quality constrained rate allocation for FGS coded video," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 2, pp. 121–130, 2003.
- [34] Y.-H. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1456–1471, 2002.

**Trista Pei-chun Chen** received her B.S. and M.S. degrees from National Tsing Hua University, Hsinchu, Taiwan, in 1997 and 1999, respectively, and her Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, Pennsylvania, in 2003. Trista is currently a video architect at NVIDIA Corporation, Santa Clara, California, performing design and testing of video hardware. From July 1998 to June 1999, she was a software engineer developing fingerprint identification algorithms at Startek Engineering Incorporated, Hsinchu, Taiwan. During the summer of 2000, she was with



HP Cambridge Research Laboratory, Cambridge, Massachusetts, conducting a research in image retrieval for massive databases. During the summer of 2001, she was with Pittsburgh Sony Design Center, Pittsburgh, Pennsylvania, designing circuits for video watermarking (VWM). Her research interests include multimedia hardware, networked video, watermark/data hiding, image processing, and biometric signal processing. She is a Member of the IEEE.

**Tsuhan Chen** has been with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania since October 1997, where he is now a Professor. He directs the Advanced Multimedia Processing Laboratory. His research interests include multimedia signal processing and communication, audio-visual interaction, biometrics, processing of 2D/3D graphics, bioinformatics, and building collaborative virtual environments. From August 1993 to October 1997, he worked in the Visual Communications Research Department, AT&T Bell Laboratories, Holmdel, New Jersey, and later at AT&T Labs-Research, Red Bank, New Jersey. Tsuhan helped create the Technical Committee on Multimedia Signal Processing, as the Founding Chair, and the Multimedia Signal Processing Workshop, both in the IEEE Signal Processing Society. He has recently been appointed as the Editor-in-Chief for IEEE Transactions on Multimedia for 2002–2004. He has coedited a book, *Advances in Multimedia: Systems, Standards, and Networks*. Tsuhan received the B.S. degree in electrical engineering from the National Taiwan University in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, California, in 1990 and 1993, respectively. He is a recipient of the National Science Foundation CAREER Award.

