

# Error-Resilient Unequal Error Protection of Fine Granularity Scalable Video Bitstreams

Hua Cai,<sup>1</sup> Bing Zeng,<sup>2</sup> Guobin Shen,<sup>1</sup> Zixiang Xiong,<sup>3</sup> and Shipeng Li<sup>1</sup>

<sup>1</sup>Microsoft Research Asia, Haidian District, Beijing 100080, China

<sup>2</sup>Department of Electrical and Electronic Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, HKSAR, China

<sup>3</sup>Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA

Received 12 August 2005; Revised 9 March 2006; Accepted 30 April 2006

This paper deals with the optimal packet loss protection issue for streaming the fine granularity scalable (FGS) video bitstreams over IP networks. Unlike many other existing protection schemes, we develop an error-resilient unequal error protection (ER-UEP) method that adds redundant information optimally for loss protection and, at the same time, cancels completely the dependency among bitstream after loss recovery. In our ER-UEP method, the FGS enhancement-layer bitstream is first packetized into a group of independent and scalable data packets. Parity packets, which are also scalable, are then generated. Unequal protection is finally achieved by properly shaping the data packets and the parity packets. We present an algorithm that can optimally allocate the rate budget between data packets and parity packets, together with several simplified versions that have lower complexity. Compared with conventional UEP schemes that suffer from bit contamination (caused by the bit dependency within a bitstream), our method guarantees successful decoding of all received bits, thus leading to strong error-resilience (at any fixed channel bandwidth) and high robustness (under varying and/or unclean channel conditions).

Copyright © 2006 Hindawi Publishing Corporation. All rights reserved.

## 1. INTRODUCTION

Streaming multimedia contents over the Internet is becoming more and more popular in the recent years, partially due to the extraordinary audio/video presentation capability of multimedia data and partially due to the increasing deployment of broadband networks. However, network heterogeneity and competing traffic over networks often cause fluctuation of the available bandwidth for each streaming service. In addition, the delivering process of multimedia contents is not error-free due to the best-effort nature of the current Internet.

Some scalable source coding schemes have been developed to cope with the varying bandwidth more efficiently. For example, the scalable mode can be chosen when running MPEG-2/4 [1, 2] and H.263+ [3] to mitigate the effect of network heterogeneity. However, this scalable mode alone is not sufficient in dealing with bandwidth fluctuations. Recently, the so-called fine granularity scalable (FGS) video coding scheme has proven to be able to offer much better scalability [4, 5].

For transmission over packet-switched networks such as the Internet, a long video bitstream is first partitioned into packets. Some packets will arrive promptly through the

network channel, while others may be lost or delayed. Thus, beside the bandwidth fluctuation, random packet loss also affects the streaming quality significantly. To combat with such packet loss, retransmission based on automatic repeat request (ARQ) is often adopted in the Internet. However, it is usually not acceptable for real-time streaming applications since it dramatically increases the end-to-end delay. On the other hand, various forward error correction (FEC) techniques [6] can generally correct certain errors so that the receiver can recover some losses without any further intervention from the sender.

An FGS video bitstream consists of two layers: the base layer and the enhancement layer. The base layer is usually coded by the traditional motion-compensated DCT scheme. It is typically very thin so as to fit some typical small bandwidths. The residue between the original DCT coefficients and the dequantized base-layer DCT coefficients forms the enhancement layer and is coded with the bitplane coding technology. Bitplane coding achieves the desired fine granularity scalability, thus yielding a scalable bitstream. Clearly, bits themselves in such a scalable bitstream are *unequally* important: bits on a more significant bitplane have higher contributions toward the overall quality than bits on a less significant bitplane. On the other hand, bits on the same

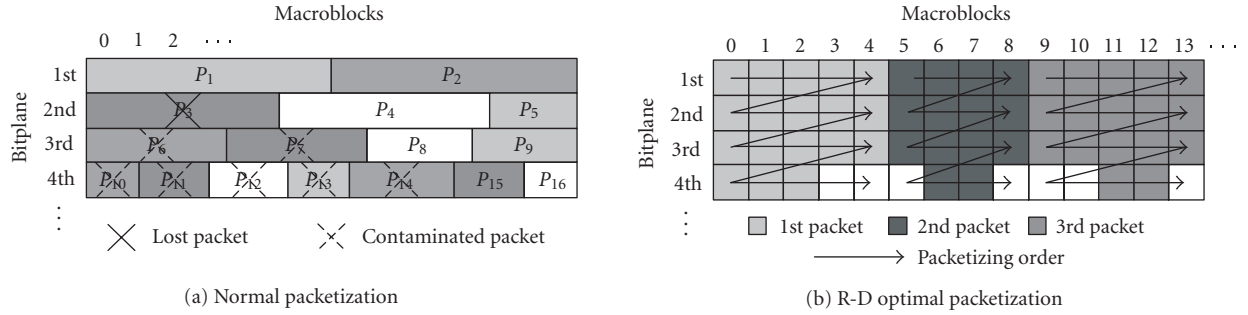


FIGURE 1: Two packetization strategies.

bitplane are causally dependent, and furthermore bits on different bitplanes are also dependent. Thus, decoding of any current bits needs the knowledge of all previous dependent bits, which adds a second interpretation, *dependency*, to the unequal importance feature of different bits.

The unequal-importance feature as discussed above naturally leads to an *unequal error protection* (UEP) policy. In fact, UEP has been widely adopted in many existing transmission schemes. In particular, a general and flexible method called priority encoding transmission (PET) [7] was proposed to cope with packet loss in which the user partitions a bitstream into segments  $m_0, m_1, \dots, m_{K-1}$  and assigns each segment with a priority value; and an FEC is then applied to encode the segments into a set of packets based on their priority values. The PET approach has been used in developing an end-to-end R-D optimized transmission scheme called FEC-based multiple description coding (MD-FEC) for scalable multimedia contents [8]. Concurrently, similar approach was proposed in [9] for the transmission of scalable coded images such that the image quality will degrade only gracefully as packet loss increases.

It seems that these UEP schemes only take into consideration the first interpretation of the unequal importance of bits in a scalable bitstream (i.e., bits themselves are unequally important). However, we believe that the second interpretation of the unequal importance (i.e., dependency—as discussed above) also has important impact. It is clear that all segments  $m_0, m_1, \dots, m_{K-1}$  generated after partitioning a scalable bitstream are dependent causally, that is, segment  $m_i$  depends on segments  $m_0, m_1, \dots, m_{i-1}$ . Thus, when an error happens in a segment, there would be many bits in those dependent segments being contaminated and becoming totally useless even if some error resilience tools are used.

In this paper, we first packetize an FGS enhancement-layer bitstream into a group of independent and scalable packets: each packet is completely independent of others and can be truncated arbitrarily to represent the original video signal at a given fidelity. As a result, the dependency problem is completely solved. Parity packets are then created. Notice that these two steps are usually done offline so that the online computation during the real-time streaming service can be greatly released. Finally, unequal error protection is achieved by allocating a given rate budget (related to the current channel conditions) among all data packets and parity

packets within each time-slot, that is, we need to optimally determine how many parity symbols from all generated parity packets should be used for protecting the corresponding data symbols at different positions within each data packet.

The rest of the paper is organized as follows. Section 2 briefly reviews the optimal packetization strategy proposed in [10] that is used to create independent data packets. In Section 3, we first present a system-level description of our proposed scheme. Then, we formulate the rate budget allocation between data packets and parity packets into an optimization problem. Finally, we develop a Lagrangian-type algorithm to solve this problem. Section 4 presents three simplified versions to meet different computing requirements. Experimental results on transmitting some typical FGS video bitstreams with both the proposed scheme and the conventional UEP schemes are shown and discussed in Section 5. Finally, some conclusions are drawn in Section 6.

## 2. OPTIMAL PACKETIZATION OF FGS VIDEO BITSTREAMS

For an FGS bitstream, bits in its enhancement layer of each video frame are usually sequentially ordered. That is, bits are scanned from the most significant bitplane of all macroblocks (MBs) all the way down to the least significant bitplane of all MBs until the specified bit rate is met. A normal packetization scheme simply chops each bitstream into packets at the MB boundary subject to the maximum packet length constraint. As mentioned before, there exists a strong degree of dependency among bits in an FGS bitstream, and such dependency has significant impact on the streaming quality because a single packet loss may render many other received packets undecodable or useless (even if they are decodable). Combining some error resilience tools such as inserting resynchronization marker and MB address information periodically, the decoding dependency can be reduced. However, the usefulness dependency still exists in the normal packetization. For example, as shown in Figure 1(a), one packet loss ( $P_3$ ) will contaminate many other packets (marked as  $P_6, P_7$ , and  $P_{10} - P_{14}$ ) and render them useless even if they are received and decoded successfully.

To overcome the drawbacks of the normal packetization, an R-D optimal packetization strategy for the FGS enhancement-layer bits was developed in [10]. It first performs an

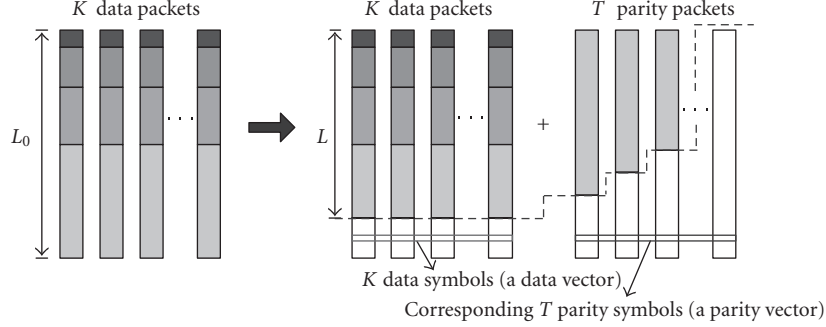


FIGURE 2: The error-resilient unequal error protection scheme.

R-D optimal bit allocation on the MB-level across all bit-planes and MBs within a time slot. Notice that collecting the R-D function of a simple FGS bitstream (e.g., generated from MPEG-4 FGS [4]) is relatively easy. However, it is more difficult for a bitstream generated from a more efficient FGS encoder such as the progressive fine granularity scalable (PFGS) encoder [5], which brings drifting errors to subsequent frames. To achieve the R-D optimal bit allocation, we need to consider the influence of the drifting errors, referring to [11] for one such method of calculating the drifting errors in the PFGS scheme.

After the bit allocation, selected bits are packetized into packets by grouping all selected bits from the same MB into one packet subject to the maximum packet length constraint. Clearly, both the decoding dependency and usefulness dependency are completely removed because each packet is now self-contained such that it can be decoded without the knowledge of other packets. Figure 1(b) shows one example of this packetization strategy. Notice that each packet is still fine scalable, as bits from the selected MBs are still scanned sequentially on the bitplane-by-bitplane basis, as depicted by the *packetizing order* in the figure. Refer to [10] for the details of the development of this optimal packetization algorithm.

### 3. ERROR-RESILIENT UNEQUAL ERROR PROTECTION

In this section, we will present our error-resilient unequal error protection (ER-UEP) scheme with emphasis on the features mentioned in Section 1.

#### 3.1. System-level description

Figure 2 shows the principle diagram of the proposed ER-UEP method. The original  $K$  data packets,  $P_1, P_2, \dots, P_K$ , are generated using the optimal packetization method in [10] with the rate budget  $R$ . In order to apply an FEC, bits in each data packet are processed on the symbol-by-symbol basis. That is, the  $k$ th data packet is interpreted as a sequence of fixed-length symbols. Let  $P_k = \{s_{k,1}, s_{k,2}, \dots, s_{k,L_0}\}$ , where  $s_{k,i}$  denotes the  $i$ th data symbol of the  $k$ th data packet and  $L_0$  is the packet length in symbols. Next,  $K$  data symbols with the same index, say  $i$ , across all  $K$  data packets are grouped to form a data vector  $\mathbf{v}_i = \{s_{1,i}, s_{2,i}, \dots, s_{K,i}\}$ . Now,  $K$  original

data packets are equivalently expressed as a list of data vectors  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{L_0}\}$ . Channel coding is then applied to generate a parity vector  $\mathbf{q}_i$ , which consists of  $T$  parity symbols for the data vector  $\mathbf{v}_i$  using the Reed-Solomon code  $RS(K+T, K)$ .<sup>1</sup> Clearly, there are totally  $L_0$  parity vectors. These generated parity vectors are then reorganized into  $T$  parity packets. Each parity packet is of length  $L_0$  with one parity symbol from each parity vector.

Notice that all data packets and parity packets are of the same length  $L_0$  so far, meaning that the protection so far is an equal protection. From the parity packet generation mechanism described above, it is evident that there is no dependency between parity symbols in a parity packet because a parity symbol only depends on its corresponding data vector. Moreover, since all data packets are independent and scalable, the resulting parity packets are also scalable and can be arbitrarily truncated. Finally, the data packets and parity packets are separate: a data packet does not contain any parity symbols and vice versa.

According to the UEP principle, different numbers of parity symbols are desired for different data vectors. This can be easily achieved by pruning away some less important parity symbols. Doing this ensures that more important symbols (e.g., bits from more significant bitplanes) obtain more protection. Nevertheless, in order to meet the overall rate constraint,  $R$ , we also need to prune away some data vectors of less significance. Thanks to the scalability of both data packets and parity packets, the pruning is feasible. In practice, such pruning is much faster than repacketization because there is almost no memory shuffling. This feature enables us to generate all data packets and parity packets offline and perform necessary online pruning during the streaming services. This is in sharp contrast against conventional UEP schemes which inevitably require repacketization because the data symbols and parity symbols in those schemes are interleaved together. In the following, we will first formulate the

<sup>1</sup> A Reed-Solomon code is specified as  $RS(n, k)$  with  $m$ -bit symbols [12]. The encoder takes  $k$  data symbols of  $m$  bits each and adds  $n - k$  parity symbols to make an  $n$  symbol codeword. The decoder can correct up to  $n - k$  symbols that are lost in a codeword. The total number of  $m$ -bit symbols in the encoded block is  $n = 2^m - 1$ . Thus, a Reed-Solomon code operating on 8-bit symbols has 255 symbols per block.

optimal budget allocation between data packets and parity packets into an optimization problem, and then develop a Lagrangian-type algorithm to solve this problem.

### 3.2. Problem statement

Streaming quality can be quantitatively measured by the expected distortion at the receiver side. In this paper, we assume that the base layer of an FGS video bitstream is always received correctly<sup>2</sup> and focus on the error protection for the enhancement layer. All notations such as *bitstream*, *packet*, and *rate* hereafter refer to those for the enhancement-layer bitstream.

For the  $i$ th data symbol of the  $k$ th data packet,  $s_{k,i}$ , its expected contribution (i.e., distortion reduction) is

$$\begin{aligned} \mathcal{E}\{\Delta D(s_{k,i})\} \\ = \Delta D(s_{k,i}) \times (1 - p_e(s_{k,i})) \times p(\mathcal{A}(s_{k,i}) | s_{k,i}), \end{aligned} \quad (1)$$

where  $\Delta D(s_{k,i})$  is the actual distortion reduction contributed by successfully receiving and decoding symbol  $s_{k,i}$ ;  $p_e(s_{k,i})$  is the loss probability after the FEC recovery for  $s_{k,i}$ ;  $\mathcal{A}(s_{k,i})$  represents the dependent symbol set of  $s_{k,i}$ ; and the conditional probability  $p(\mathcal{A}(s_{k,i}) | s_{k,i})$  expresses the impact of bitstream dependency. Thanks for the optimal packetization used in our ER-UEP scheme,  $\mathcal{A}(s_{k,i}) = \{s_{k,1}, s_{k,2}, \dots, s_{k,i-1}\}$ . Hence, the decoding of symbol  $s_{k,i}$  is independent of  $\mathcal{A}(s_{k,i})$ . In other words, the conditional probability  $p(\mathcal{A}(s_{k,i}) | s_{k,i})$  always equals 1. Therefore, (1) can be simplified as

$$\mathcal{E}\{\Delta D(s_{k,i})\} = \Delta D(s_{k,i}) \times (1 - p_e(s_{k,i})). \quad (2)$$

Let  $\Delta D(\mathbf{v}_i)$  be the distortion reduction of data vector  $\mathbf{v}_i$ . It is easy to see that the distortion reduction is additive, and thus  $\Delta D(\mathbf{v}_i)$  can be computed by accumulating the distortion reduction of its component data symbols:

$$\Delta D(\mathbf{v}_i) = \sum_{k=1}^K \Delta D(s_{k,i}). \quad (3)$$

Clearly, the importance of data symbols decreases from more significant bitplanes to less significant bitplanes, and  $\Delta D(\mathbf{v}_i)$  is ensured to be convex [10]. Thus,  $\Delta D(\mathbf{v}_i) \leq \Delta D(\mathbf{v}_j)$  for all  $i > j$ . Let the packet loss rate after loss recovery be  $P_e(k, t)$  when  $k$  data symbols are protected by  $t$  parity symbols. This function quantifies the loss recovery performance and can be either obtained in the transmission system or calculated through some mathematical approaches [13]. Now, the overall expected distortion (with UEP) at the receiver side can be calculated as follows:

$$\mathcal{E}\{D\} = D_{BL} - \sum_{i=1}^L (1 - P_e(K, T_i)) \times \Delta D(\mathbf{v}_i), \quad (4)$$

where  $D_{BL}$  denotes the distortion when only the base layer is received,  $L$  (with  $L \leq L_0$ ) is the number of selected data vectors, and  $T_i$  is the number of parity symbols for the  $i$ th data vector. Note that UEP is achieved by varying the parity symbol number  $T_i$  for different data vectors, with constraint  $T_i \leq T_j$ , for all  $i > j$ , which is derived from the fact that  $\Delta D(\mathbf{v}_i)$  is monotonously decreasing.

Finally, as the data packet rate  $R_S$  and the parity packet rate  $R_C$  are constrained by the total budget rate  $R$ , the rate constraint can be expressed as

$$R_S + R_C = \sum_{i=1}^L (K + T_i) \times m \leq R, \quad (5)$$

where  $m$  is the symbol length in bits.

Now, the optimization problem can be formulated as follows: given the number of data packets  $K$  (each data packet has  $L_0$  symbols), the R-D function  $(R(\mathbf{v}_i), \Delta D(\mathbf{v}_i))$  (which degenerates to  $\Delta D(\mathbf{v}_i)$  as the rate for each data vector is equal) and the loss-recovery performance function  $P_e(k, t)$  find the most important data vectors and determine the protection strength for each data vector such that  $\mathcal{E}\{D\}$  is minimized subject to the rate constraint. In other words, we need to find the number of selected data vectors  $L$  and the number of parity symbols  $T_i$  for each data vector  $\mathbf{v}_i$  ( $i = 1, 2, \dots, L$ ).

### 3.3. Solution

Since the ultimate protection strength  $T_i$  satisfies  $T_i \leq T_j$  for all  $i > j$ , when a certain data vector  $\mathbf{v}_i$  is received or recovered, all its dependent vectors  $\mathbf{v}_j$  ( $j = 1, 2, \dots, i-1$ ) are ensured to be received or recovered. Therefore, in the ER-UEP scheme, the R-D function after loss recovery for each data vector can be computed independently without requiring other data vectors. As a result, the Lagrangian optimization can be applied here to solve the optimization problem formed above [14].

According to the Lagrangian optimization principle, the optimal solution can be found by applying the *equal slope* (or, constant slope) optimization [14], where the term slope means the expected distortion reduction efficiency of a data vector after being protected by one more parity symbol. To apply the equal slope optimization, we should compute the slopes of each data vector when it is protected by different numbers of parity symbols. Specifically, for a data vector  $\mathbf{v}_i$ , two vectors  $\mathbf{S}_i$  and  $\mathbf{R}_i$ , which represent the protection efficiency (slope) and the corresponding rate, can be obtained as follows:

$$\mathbf{S}_i = [s(i, t)]_{t=0,1,\dots,T}, \quad \mathbf{R}_i = [r(i, t)]_{t=0,1,\dots,T}, \quad (6)$$

where

$$\begin{aligned} r(i, t) &= (K + t) \times m, \\ s(i, t) &= \Delta D(\mathbf{v}_i) \cdot \frac{P_e(K, t-1) - P_e(K, t)}{r(i, t) - r(i, t-1)}. \end{aligned} \quad (7)$$

Here, we define  $P_e(K, -1) = 1$  and  $r(i, -1) = 0$  for completeness. Moreover,  $\mathbf{S}_i$  can be interpreted as a projection of

<sup>2</sup> This assumption is reasonable since the base layer of an FGS bitstream is very small and yet very important, heavy error protection (even ARQ) can usually be applied to ensure error-free transmission in practice.

distortion reduction function over a common vector  $\mathbf{W}$  of length  $T + 1$ , that is,

$$\mathbf{S}_i = \Delta D(\mathbf{v}_i) \cdot [w_0 \ w_1 \ \cdots \ w_T], \quad (8)$$

where

$$w_t = \frac{P_e(K, t-1) - P_e(K, t)}{r(i, t) - r(i, t-1)}. \quad (9)$$

Note that applying the equal slope optimization requires that elements of the slope vector should be monotonously decreasing. However, because of the introduction of the loss recovery function, even though the R-D function of data vectors is convex, elements of the slope vectors  $\mathbf{S}_i$  (or equivalently, elements of the common vector  $\mathbf{W}$ ) may not be strictly monotonously decreasing in general. Consequently, a postprocessing stage is required for merging those non-decreasing elements in  $\mathbf{W}$ . The postprocessing includes two iterative steps: (1) divide the elements in  $\mathbf{W}$  into rising, flat, and falling sections; and (2) if there are any rising or flat sections, merge all elements in the rising or the flat sections as one single element and then return to step (1), otherwise, the postprocessing is completed. A similar postprocessing method and a relevant example can also be found in [8].

After the postprocessing, we can obtain a strictly monotonously decreasing vector  $\mathbf{W}'$  of length  $T' + 1$ :

$$\mathbf{W}' = [w'_0 \ w'_1 \ \cdots \ w'_{T'}], \quad (10)$$

where

$$w'_j = \frac{P_e(K, t_{j-1}) - P_e(K, t_j)}{r(i, t_j) - r(i, t_{j-1})} \quad (11)$$

and  $t_j$  is the corresponding protection strength of the  $j$ th element in  $\mathbf{W}'$ . Next, the strictly monotonously decreasing slope matrix  $\mathbf{S}'$  and the corresponding rate matrix  $\mathbf{R}'$  can be easily obtained from  $\mathbf{W}'$ , each of size  $L_0 \times (T' + 1)$ :

$$\begin{aligned} \mathbf{S}' &= \begin{bmatrix} \mathbf{S}'_1 \\ \vdots \\ \mathbf{S}'_{L_0} \end{bmatrix} = \begin{bmatrix} s'(1, t_0) & \cdots & s'(1, t_{T'}) \\ \vdots & \ddots & \vdots \\ s'(L_0, t_0) & \cdots & s'(L_0, t_{T'}) \end{bmatrix}, \\ \mathbf{R}' &= \begin{bmatrix} \mathbf{R}'_1 \\ \vdots \\ \mathbf{R}'_{L_0} \end{bmatrix} = \begin{bmatrix} r'(1, t_0) & \cdots & r'(1, t_{T'}) \\ \vdots & \ddots & \vdots \\ r'(L_0, t_0) & \cdots & r'(L_0, t_{T'}) \end{bmatrix}, \end{aligned} \quad (12)$$

where

$$\begin{aligned} r'(i, t_j) &= r(i, t_j) = (K + t_j) \times m, \\ s'(i, t_j) &= \Delta D(\mathbf{v}_i) \times w'_j. \end{aligned} \quad (13)$$

Now it is ready to apply the equal slope optimization. The optimal solution that minimizes (4) can be found through looking for the best protection strength  $T_i = t_j$  for the  $i$ th data vector that satisfies  $s'(i, t_{j+1}) < \lambda \leq s'(i, t_j)$ , with the

Initially, let  $\lambda_L = 0$ ,  $\lambda_H =$  a large number,  $R_{\text{cost}} = 0$ , and let  $\delta$  be a given parameter for exiting condition.  
While ( $|R_{\text{cost}} - R| > \delta$ )  
{  
   $\lambda = (\lambda_L + \lambda_H)/2$ ;  
  Find  $T_i = t_j$  for the  $i$ th data vector that satisfies  $s'(i, t_{j+1}) < \lambda \leq s'(i, t_j)$ ;  
  Find  $L$ —the maximum  $i$  satisfying  $\lambda \leq s'(i, T_i)$ ;  
  If  $R_{\text{cost}} = \sum_{i=0}^L r'(i, T_i) \leq R$ , then  $\lambda_H = \lambda$ ; else,  
   $\lambda_L = \lambda$ .  
}

ALGORITHM 1

constraint of the total rate budget  $R$  for the time-slot under optimization:

$$\sum_{i=1}^L r'(i, T_i) \leq R, \quad (14)$$

where  $\lambda$  is the Lagrangian multiplier and  $L$  is the maximum  $i$  that satisfies  $s'(i, T_i) \geq \lambda$ . Some efficient iterative algorithms such as the bisection searching can be applied here (see Algorithm 1).

Finally, rate shaping can be efficiently performed since both the data packets and parity packets are scalable. Specifically, for each data packet, the first  $L$  data symbols are kept whereas the data symbols from position  $L + 1$  to  $L_0$  are discarded. Similarly, the parity packets are selected and truncated according to the determined optimal protection strength  $T_i$ .

#### 4. FAST PROTECTION SCHEMES

The complete ER-UEP framework consists of four steps, namely data packets generation, parity packets generation, data and parity rate calculation, and rate shaping. Since generating data packets and parity packets can be performed offline in ER-UEP and the rate shaping is also very simple, the complexity only comes from the process of data and parity rate calculation, that is, selecting data vectors and their corresponding parity symbols. The optimal algorithm is detailed in Section 3.3, with a moderate/high computing cost that is acceptable perhaps only when supporting a limited number of users. In this section, we present three simplified schemes for supporting a large number of users simultaneously at cost of marginal quality degradation.

##### 4.1. Segment-level ER-UEP scheme

Algorithm 1, described in Section 3.3, tries to allocate the rate budget between data packets and parity packets at the symbol level. The complexity is therefore determined by the size of the rate-contribution matrices,  $L_0 \times (T' + 1)$ . Obviously, one way to reduce the complexity is to design the protection at a coarser level. For instance, we can group  $M$

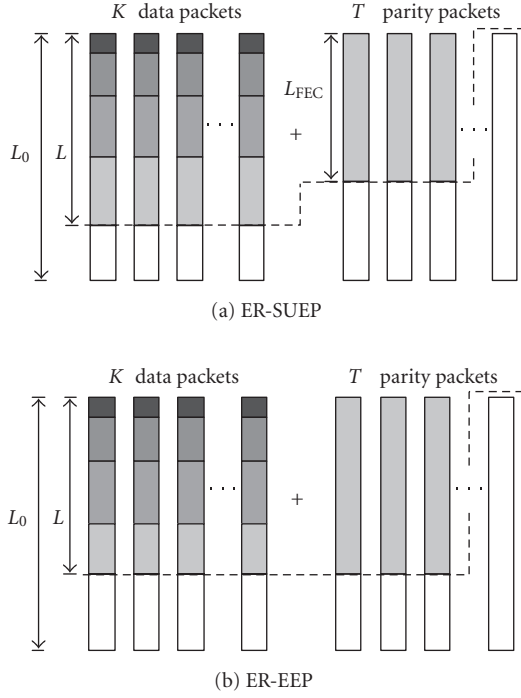


FIGURE 3: Two fast implementations of the ER-UEP scheme.

symbols within each data packet into one segment and provide equal protection to all symbols in the same segment. As a result, the size of the rate-contribution matrices is reduced to  $(L_0/M) \times (T' + 1)$ , and the computing cost is only  $1/M$  of the original one. Moreover, the value of  $M$  may be altered to achieve different speedups.

#### 4.2. Error-resilient simple unequal error protection

As depicted in Figure 3(a), in this error-resilient simple unequal error protection (ER-SUEP) scheme, each data packet is divided into two parts. The upper part with  $L_{FEC}$  symbols is of high importance and will be protected by sending  $\tilde{T}$  parity packets, while the lower part with  $L - L_{FEC}$  symbols is of low importance and will not be protected. The expected distortion is now simplified as

$$\begin{aligned} \mathcal{E}\{D_1\} = & D_{BL} - (1 - P_e(K, \tilde{T})) \cdot \sum_{i=1}^{L_{FEC}} \Delta D(\mathbf{v}_i) \\ & - (1 - P_e(K, 0)) \cdot \sum_{i=L_{FEC}+1}^L \Delta D(\mathbf{v}_i), \end{aligned} \quad (15)$$

while the optimization problem is simplified as follows: given the available rate  $R$  for a time slot and the loss-recovery performance function  $P_e(k, t)$ , choose the number of parity packets  $\tilde{T}$  and parity packet length  $L_{FEC}$  such that the expected distortion  $\mathcal{E}\{D_1\}$  is minimized with the rate constraint:  $(L \times K + L_{FEC} \times \tilde{T}) \times m \leq R$ .

#### 4.3. Error-resilient equal error protection

The maximum number of searching points equals to  $L_0 \times (T' + 1)$  in the ER-SUEP scheme. To further reduce it, an error-resilient equal error protection (ER-EEP) scheme is proposed in the following. In this scheme, all selected data symbols are equally protected with strength  $\tilde{T}$ , as illustrated in Figure 3(b). The simplified optimization problem can be stated as follows: given the available rate  $R$  for a time-slot and the loss-recovery performance function  $P_e(k, t)$ , choose the best protection strength  $\tilde{T}$  such that the expected distortion is minimized:

$$\mathcal{E}\{D_2\} = D_{BL} - (1 - P_e(K, \tilde{T})) \cdot \sum_{i=1}^L \Delta D(\mathbf{v}_i), \quad (16)$$

where

$$L = \left\lceil \frac{R}{(K + \tilde{T}) \times m} \right\rceil. \quad (17)$$

Notice that the complexities of the above-presented three simplified schemes are decreasing, and one later scheme can be viewed as a special case of an earlier scheme, as can be seen from (15) and (16).

### 5. EXPERIMENTAL RESULTS

The proposed ER-UEP scheme and all its simplified versions are extensively tested against various packet loss cases to simulate streaming FGS video bitstreams over the Internet. Some standard test sequences *Foreman*, *Coastguard*, *News*, and *Silence* in CIF format and 10 Hz are used in our experiments. As the PFGS scheme [5] gives the highest coding efficiency among all the available FGS schemes, it is used for generating the FGS bitstream in our experiment. Only the first frame is encoded as *I* frame and all others as *P* frames. The bit rate for the base layer is chosen as 96 kbps and that for the enhancement layer is allowed to be up to 5 000 kbps. Assume that the base-layer bitstream is transmitted without errors.

To simulate the bandwidth fluctuation in the Internet, the total available enhancement-layer rate is assumed to be uniformly distributed within the range of (512, 1024) kbps for each time slot of one second. Meanwhile, to simulate the burst loss in the Internet, a two-state Gilbert model, characterized by the global packet loss rate (PLR) and the average burst length (ABL), is used in our experiments. Furthermore, in order to evaluate the performance and robustness of our ER-UEP scheme under degraded channel conditions, the enhancement-layer bitstreams are first protected at three Gilbert models with different (PLR, ABL): (0.01, 1.5), (0.05, 2.0), and (0.10, 2.5), and then transmitted over channels with varying PLR (over a wide range) but fixed ABL (as given in the three models selected above). Finally, to randomize the burst packet loss, packets from two adjacent FEC blocks,  $\text{BLOCK}_A = \{P_1^A, P_2^A, P_3^A, \dots\}$  and  $\text{BLOCK}_B = \{P_1^B, P_2^B, P_3^B, \dots\}$ , are interleaved before the transmission. That is, the packet transmission order is

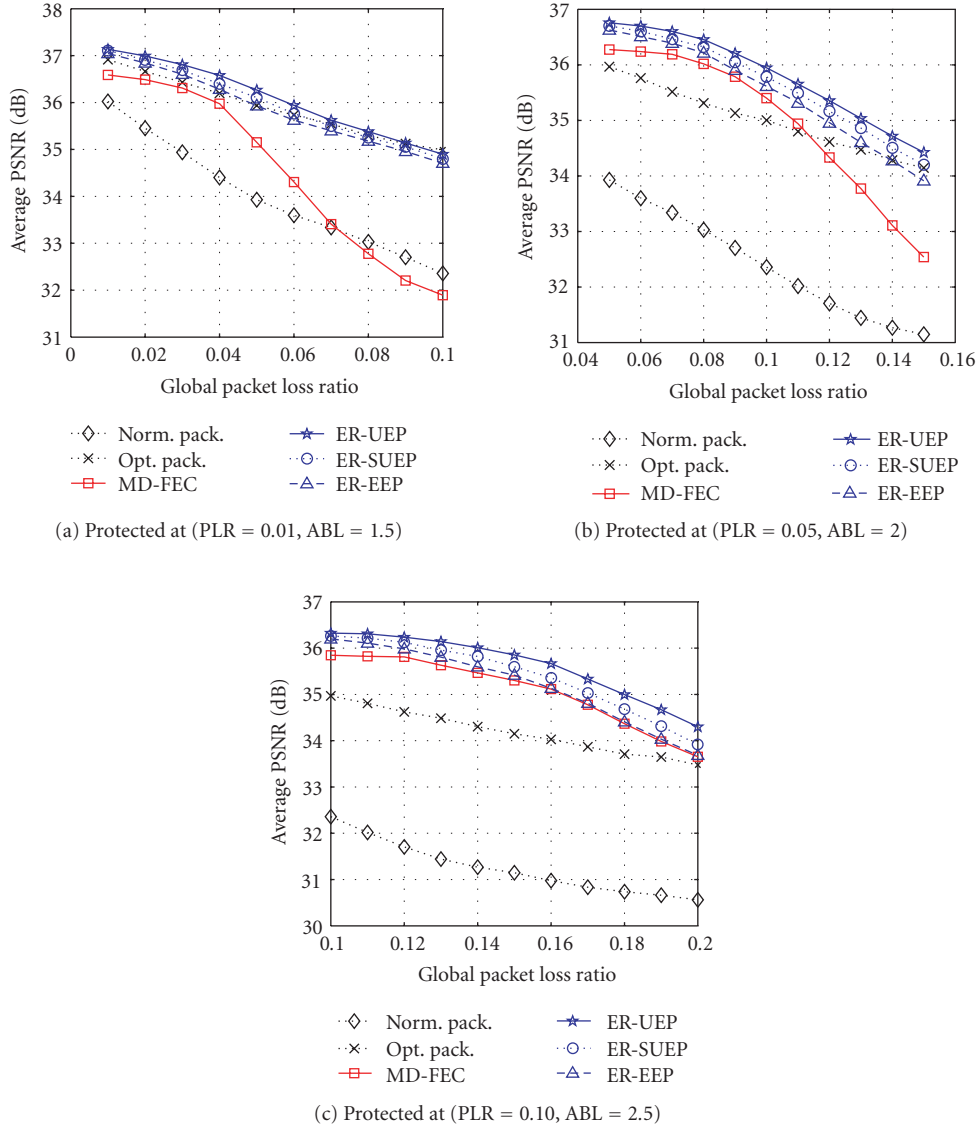


FIGURE 4: Comparative evaluation of the proposed scheme at different packet loss rates (for *Foreman* sequence).

$P_1^A, P_1^B, P_2^A, P_2^B, \dots$ , where  $P_j^i$  denotes the  $j$ th packet of the  $i$ th FEC block.

The MD-FEC method [8] mentioned before is chosen as the benchmark for comparison. In our implementation of the MD-FEC scheme, the enhancement-layer bitstream of each frame is first ordered as that in the normal packetization: bits of all MBs are ordered MB by MB and bitplane by bitplane, from the most significant bitplane of all MBs to the least significant bitplane of all MBs. As a result, the importance of the bitstream from the first to the last bit is in a decreasing way. The bitstream is then partitioned into decreasing prioritized segments  $m_0, m_1, \dots$ . Usually, bits from the same bitplane can be considered as one segment. For the given channel bandwidth and the loss-recovery performance function  $P_e(k, t)$ , the optimal protection parameters  $(K_i, T_i)$  of segment  $m_i$  can be calculated by locating the points on the

R-D curve of the enhancement-layer bitstream. After that, the Reed-Solomon code  $RS(K_i + T_i, K_i)$  is used to generate parity symbols for segment  $m_i$  based on the found protection parameters  $(K_i, T_i)$ . In the end, the protected segments along with their parity symbols are packetized into 800-byte long packets using the packetization scheme used by MD-FEC [8]. Refer to reference [8] for more details. Notice that to improve error resilience for both the MD-FEC scheme and the normal packetization scheme without error protection, we insert a 23-bits resynchronization marker followed by 9-bits MB address information at the MB boundary for any bits interval greater than 1000 bits.

In our ER-UEP scheme, all enhancement-layer bits in the current transmission time slot are selected based on the R-D criterion under the constraint of total available rate of that time-slot. Data packets are then created using the

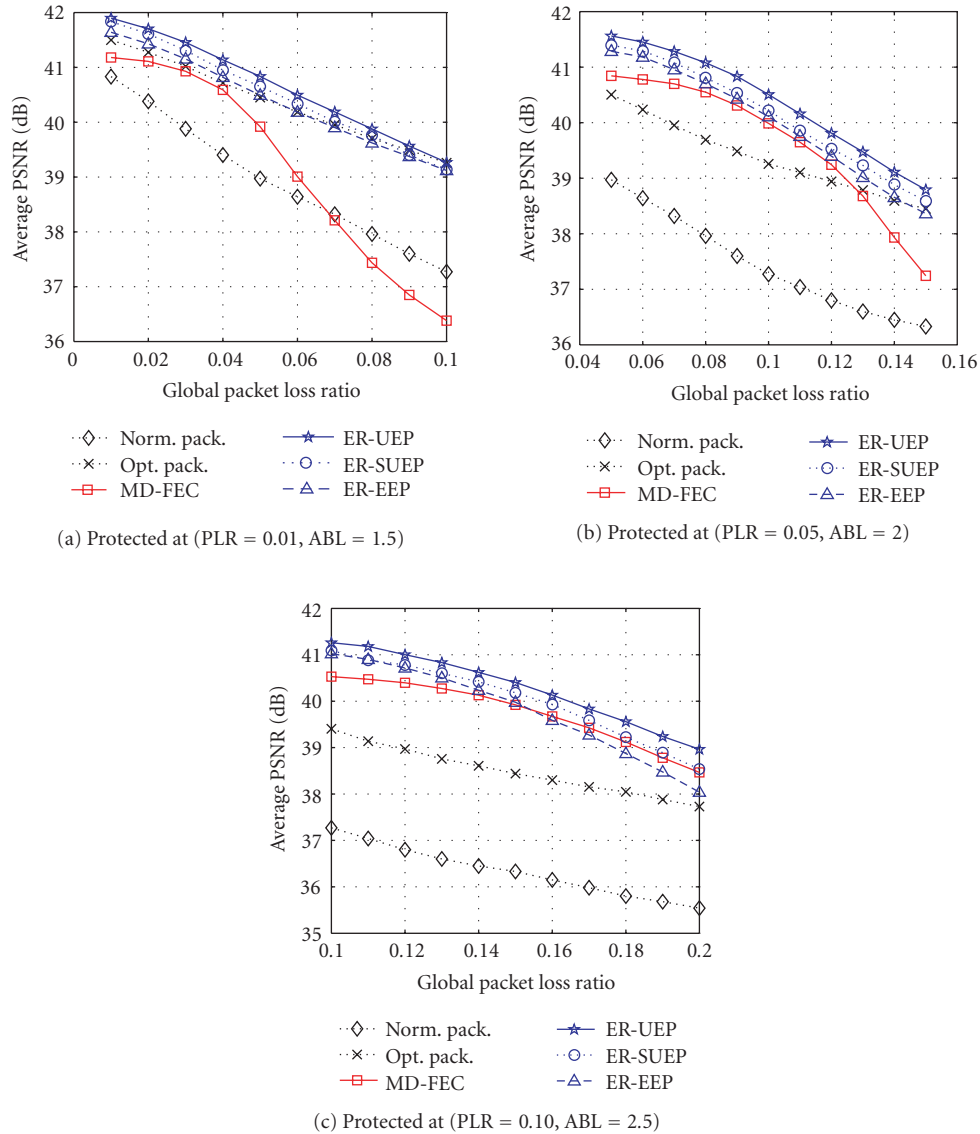


FIGURE 5: Comparative evaluation of the proposed scheme at different packet loss rates (for *News* sequence).

optimal packetization strategy presented in Section 2. Each data packet is also 800 bytes long. After generating parity packets, the length of data packets and the number of parity packets are computed for the given channel conditions. Finally, all the packets are shaped accordingly by pruning away the least significant symbols.

To differentiate the actual gain of the proposed ER-UEP scheme, we also performed experiments where only the optimal packetization is applied (without any error protection). Figures 4 and 5 show the performances of the ER-UEP scheme, its simplified versions, and the benchmarks for the *Foreman* and *News* sequences. As for the other two sequences, we did not include their figures since they are quite similar to Figures 4 and 5.

A few observations can be made from Figures 4 and 5. (1) The performance of all UEP schemes indeed degrades gracefully when the actual PLR deviates from the assumed

one when performing error protection. However, conventional UEP schemes achieve graceful degradation only in a small range while the proposed ER-UEP schemes (including the simplified versions) are more robust over a much wider range. Clearly, our proposed UEP framework is more error resilient. (2) Under the best conditions (i.e., packet loss rate prediction is accurate), the proposed ER-UEP schemes outperform the MD-FEC scheme. The gain comes from two sources: optimal packetization and UEP. (3) The optimal packetization provides significant gain and the UEP further improves the performance significantly as well. (4) The performance degradation for the simplified ER-UEP schemes (ER-SUEP and ER-EEP) is marginal.

Another interesting observation is that all the UEP schemes work best when the actual packet loss rate is exactly as those assumed when performing error protection. This can be clearly seen from the subplots at the same packet



TABLE 1: Channel rate percentage under different (PLR, ABL).

Sequence	(PLR, ABL)	ER-UEP	MD-FEC
Foreman	(0.01, 1.5)	2.3%	4.3%
	(0.05, 2)	7.4%	11.9%
	(0.1, 2.5)	12.1%	19.7%
Coastguard	(0.01, 1.5)	3.3%	4.6%
	(0.05, 2)	9.5%	12.8%
	(0.1, 2.5)	15.5%	20.8%
News	(0.01, 1.5)	2.2%	4.1%
	(0.05, 2)	7.9%	13.3%
	(0.1, 2.5)	14.5%	21.7%
Silence	(0.01, 1.5)	1.2%	4.2%
	(0.05, 2)	4.8%	12.8%
	(0.1, 2.5)	8.4%	18.7%

TABLE 2: Comparison of average PSNR (dB) under varying PLR (PLR denotes the predicted PLR).

Sequence	$\overline{\text{PLR}}$	NO-EP	MD-FEC	ER-UEP
Foreman	0.01	36.23	36.8	37.21
	0.05	35.34	36.33	36.92
	0.1	33.01	35.75	36.54
Coastguard	0.01	33.62	34.05	34.49
	0.05	31.7	33.54	33.98
	0.1	30.23	33.07	33.57
News	0.01	40.84	41.32	41.89
	0.05	38.75	40.98	41.57
	0.1	37.5	40.57	41.22
Silence	0.01	38.18	38.16	38.79
	0.05	36.88	37.51	38.39
	0.1	35.77	36.94	38.03

loss rate. For example, we can find that the UEP schemes aiming at  $\text{PLR} = 0.1$  (the bottom sub-plot) yield the best performance among all three experiments when the actual PLR is exactly 0.1. This observation confirms with our conclusion that a good packet loss prediction is still critical to UEP schemes.

As mentioned in Section 1, the proposed ER-UEP scheme achieves higher bandwidth utilization because of the error resilient property. The reason is that in our ER-UEP framework any received data bits can be decoded, whereas this cannot be guaranteed in conventional schemes. Furthermore, because our scheme is less sensitive to transmission errors, more bits can be allocated for data packets. In Table 1, we present the percentage of parity bits for different UEP

schemes under three experimental scenarios when the total enhancement-layer rate equals 768 kbps. Clearly, our scheme needs lighter protection. Notice that even though less protection is applied, the resulting PSNR is higher in our scheme thanks for its strong error resilient capability.

At last, we evaluate the performances on channels with prediction errors when the total enhancement-layer rate equals 768 kbps. This kind of channel is simulated by adding a Gaussian noise on the PLR of the Gilbert loss process. That is, for the predicted PLR on which the loss protection is based, the actual packet loss rate equals  $\text{PLR} + w$ , where  $w$  is an additive Gaussian noise (updated every time slot) with zero mean and  $\sigma^2(\text{PLR})^2$  variation ( $\sigma = 0.2$  in our experiments). Hence, the channel condition for each time slot can be either better or worse than the predicted one. It can be seen from Table 2 that the MD-FEC scheme improves the quality of the normal packetization scheme a lot, and our ER-UEP scheme provides the best quality.

## 6. CONCLUSIONS AND FUTURE WORKS

We presented an error resilient unequal error protection scheme for streaming FGS video bitstreams over the Internet. Based on the optimal packetization method, our proposed scheme overcomes the common constraints that other conventional UEP schemes suffer from. As a result, the proposed scheme not only provides better quality at the target packet loss rate, but also is more robust over a wide range of packet loss rates. Several fast implementations were also presented. Extensive simulation results demonstrated the effectiveness of our proposed scheme.

Besides the FGS video bitstreams, the proposed method can also work for other scalable image/video bitstreams such as the SPIHT [15] encoded image bitstream and the SVC [16] encoded enhancement-layer video bitstream, as long as they can be packetized into independent and scalable data packets. Moreover, we believe that the unequal error protection and error-resilience concept could give remarkable quality improvements for wireless videos, which is getting more and more interests recently. This is one focus of our future works.

## ACKNOWLEDGMENT

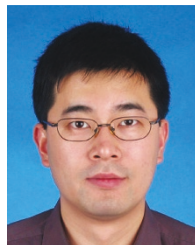
The authors would like to thank Dr. Feng Wu from Microsoft Research Asia for many fruitful discussions on the implementation of the proposed protection scheme for FGS video bitstreams.

## REFERENCES

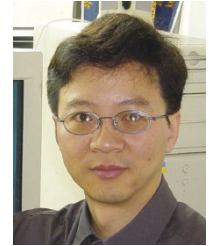
- [1] ISO/IEC 13818-2, “Generic coding of moving pictures and associated audio, part-2 video,” November 1994.
- [2] ISO/IEC 14496-2, “Coding of audio-visual objects, part-2 visual,” December 1998.
- [3] ITU-T Recommendation H263, “Video coding for low bit-rate communication,” 1998.

- [4] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301–317, 2001.
- [5] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 332–344, 2001.
- [6] R. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, Mass, USA, 1993.
- [7] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Transactions on Information Theory*, vol. 42, no. 6 pt 1, pp. 1737–1744, 1996.
- [8] R. Puri, K.-W. Lee, K. Ramchandran, and V. Bharghavan, "An integrated source transcoding and congestion control paradigm for video streaming in the internet," *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 18–32, 2001.
- [9] 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.
- [10] H. Cai, G. Shen, Z. Xiong, S. Li, and B. Zeng, "An optimal packetization scheme for fine granularity scalable bitstream," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 5, pp. 641–644, Scottsdale, Ariz, USA, May 2002.
- [11] H. Cai, G. Shen, S. Li, and B. Zeng, "Optimal rate allocation for macroblock-based progressive fine granularity scalable video coding," in *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 745–748, Rochester, NY, USA, September 2002.
- [12] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1983.
- [13] P. Frossard, "FEC performance in multimedia streaming," *IEEE Communications Letters*, vol. 5, no. 3, pp. 122–124, 2001.
- [14] 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.
- [15] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, 1996.
- [16] J.-R. Ohm, "Advances in scalable video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 42–56, 2005.

**Hua Cai** received the B.S. degree from the Shanghai Jiaotong University, Shanghai, China, in 1999, and the Ph.D. degree from the Hong Kong University of Science and Technology (HKUST) in 2003, all in electrical and electronic engineering. He is a Member of the IEEE and ACM. He joined Microsoft Research Asia, Beijing, China, in December 2003 and is currently an Associate Researcher in the Media Communication Group. His research interests include digital image/video signal processing, image/video coding and transmission, multiview video system, multiview video coding and transmission, and mobile media computing.



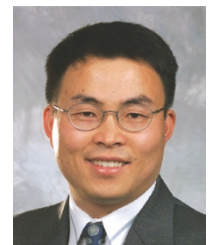
**Bing Zeng** joined the Hong Kong University of Science and Technology in 1993 and is currently an Associate Professor at the Department of Electrical and Electronic Engineering. His general research interests include digital signal and image processing, linear and nonlinear filter design, and image/video coding and transmission. His most recent research focus is on some fundamental issues in image/video coding such as directional transform, truly optimal rate allocation, and smart motion estimation/compensation, as well as various solutions for real-time video streaming applications over the Internet and wireless. His research efforts in these areas have produced over 150 journal and conference publications. He received the B.Eng. and M.Eng. degrees from the University of Electronic Science and Technology of China in 1983 and 1986, respectively, and the Ph.D. degree from Tampere University of Technology, Finland, in 1991, all in electrical engineering. He worked as a postdoctoral fellow at the University of Toronto and Concordia University during 1991–1993 and was a Visiting Researcher at Microsoft Research Asia, Beijing, China, in 2000. He was an Associate Editor for the *IEEE Transactions on Circuits and Systems for Video Technology* during 1995 to 1999 and served in various capacities in a number of international conferences. He is currently a Member of the Visual Signal Processing & Communications Technical Committee of the IEEE CAS Society.



**Guobin Shen** received the B.S. degree from Harbin University of Engineering, Harbin, China, in 1994, the M.S. degree from Southeast University, Nanjing, China, in 1997, and the Ph.D. degree from Hong Kong University of Science and Technology (HKUST) in 2001, all in electrical and electronic engineering. He is a Member of the IEEE and ACM. He was a Research Assistant at HKUST from 1997 to 2001. Since then, he has been with Microsoft Research Asia where he is now a Researcher and Project Leader in the Wireless and Networking Group. His research interests include digital image and video signal processing, video coding and streaming, distributed/parallel computing and peer-to-peer networking, general computing on GPU, wireless networking and mobile computing, and media management. He has published about a dozen journal papers and more than thirty conference papers. He has been granted two US patents and filed more than a dozen patent applications. He is now serving as a TPC Member for several international conferences and as a Reviewer for several journals and many conferences.



**Zixiang Xiong** received the Ph.D. degree in electrical engineering in 1996 from the University of Illinois at Urbana-Champaign. From 1997 to 1999, he was with the University of Hawaii. Since 1999, he has been with the Department of Electrical and Computer Engineering at Texas A&M University, where he is an Associate Professor. He spent the summers of 1998 and 1999 at Microsoft Research, Redmond, Wash, and the summers of 2000 and 2001 at Microsoft Research in Beijing. His current research interests are network information theory and code designs, genomic signal processing, and networked multimedia. He received an NSF Career Award in 1999, an ARO Young Investigator



Award in 2000, and an ONR Young Investigator Award in 2001. He also received Faculty Fellow awards in 2001, 2002, and 2003 from Texas A&M University. He served as an Associate Editor for the IEEE Trans. on Circuits and Systems for Video Technology (1999–2005) and the IEEE Trans. on Image Processing (2002–2005). He is currently an Associate Editor for the IEEE Trans. on Signal Processing and the IEEE Trans. on Systems, Man, and Cybernetics (part B).

**Shipeng Li** received the B.S. and M.S. degrees from the University of Science and Technology of China (USTC), Hefei, in 1988 and 1991, respectively, and the Ph.D. degree from Lehigh University, Bethlehem, Pa, in 1996, all in electrical engineering. He was with the Electrical Engineering Department, USTC, during 1991 and 1992. He was a Member of the Technical Staff at Sarnoff Corporation, Princeton, NJ, during 1996–1999. He has been a Researcher with Microsoft Research Asia, Beijing, China, since May 1999, and is now a Research Manager of the Internet Media Group. His research interests include image/video compression and communications, digital television, wireless and mobile communication, and digital rights management and security. He has contributed several technologies to the MPEG-4 and H.264 international standards. He is a Member of Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society and a Member of Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society. He serves in the Editorial Boards of the IEEE Transactions on Circuits and Systems for Video Technology and the Journal of Visual Communications and Image Representation. He was a Special Session Chair of the IEEE PCM 2000 and a Local Chair of the IEEE PCM 2001, the Technical Program Cochair for VCIP 2005, General Cochair of PV 2006, and he is now a Track Cochair of the IEEE ICME 2006. He holds Guest Professorships in Sichuan University, Shandong University, Huazhong University of Science and Technology, Shanghai Jiaotong University, and the University of Science and Technology of China.

