Combined Error-Concealment and Error-Correction in Rate-Distortion Analysis for Multiple Substream Transmissions

†Chih-Ming Fu, ‡Wen-Liang Hwang, and †Chung-Lin Huang

†Department of Electrical Engineering, National Tsing Hua University

‡Institute of Information Science, Academia Sinica, Taiwan

October 18, 2006

Abstract

We propose a new framework for multiple scalable bitstream video communications over lossy channels. The major feature of the framework is that the encoder estimates the effects of post-processing concealment and includes those effects in the rate-distortion analysis. Based on the framework, we develop a rate-distortion optimization algorithm to generate multiple scalable bitstreams. The algorithm maximizes the expected peak signal-to-noise ratio by optimally assigning forward error control codes and transmission schemes in a constrained bandwidth. The framework is a general approach motivated by previous methods that perform concealment in the decoder, as in our special case. Simulations show that the proposed approach can be implemented efficiently and that it outperforms previous methods by more than 2 dB.

1 Introduction

Forward error control (FEC) methods are promising solutions for video streaming over lossy channels [1, 2, 3]. In recent years, some wavelet-based coders [4, 5] have used FEC methods and multiple correlated bitstreams to transmit and decode each bitstream independently, which provides additional error-resilience at high loss rates [6, 7, 8, 9, 10]. The decoder, furthermore, applies a post-processing concealment procedure to the received bitstreams to conceal packets that cannot be recovered by using FEC [6] and multiple bitstreams solely. Although combining multiple bitstreams, FEC, and error concealment provides reliable transmission in a packet loss environment, to our knowledge, the encoders of existing methods do not use the post-processing method of the decoder in rate-distortion analysis. Therefore, the effects of applying the error concealment procedure on the overall performance have not been analyzed. In this paper, we demonstrate that the decoder's performance can be improved significantly if the encoder has a *priori* knowledge of the decoder's concealment method, and uses that knowledge in the rate-distortion analysis.

The contribution of the present study is twofold. First, we propose a new error-resilient framework in which the encoder uses FEC and multiple bitstreams, and incorporates the concealment method in the design of the coded bitstreams. To formulate and analyze our approach comprehensively, we use the "expected rate-distortion" metric to coordinate all transmission components, as shown in Figure 1. This enables us to obtain a unified measurement of the source, channel, and post-processing performance. In addition to the parameters for source coding and the channel statistics for transmission, two sets of parameters unique to our framework are introduced. One set measures the efficiency of the concealment method, while the other indicates whether a bitstream has been sent or not sent. The encoder uses the two sets of parameters to measure and compare the performance of sending a bitstream and not sending it, i.e. concealing it. A bitstream is usually organized as a single quality layer or as multiple quality layers. We formulate our framework for a single layer and then extend the formulation to multiple layers. Second, to efficiently adapt our method to a time-varying transmission environment, instead of using global optimization (which may involve a time-consuming solution), we modify the method in [1] so that we can use its efficient algorithm to obtain a local optimal solution of the proposed framework. Simulation results obtained from an implementation of our approach show that it is simple, fast, and robust in hostile network conditions. We compare our results with those of the method in [6], which is the motivation for our study, and show that we can improve the performance by more than 2 dB for various video sequences.

The remainder of the paper is organized as follows. Section 2 contains background informa-

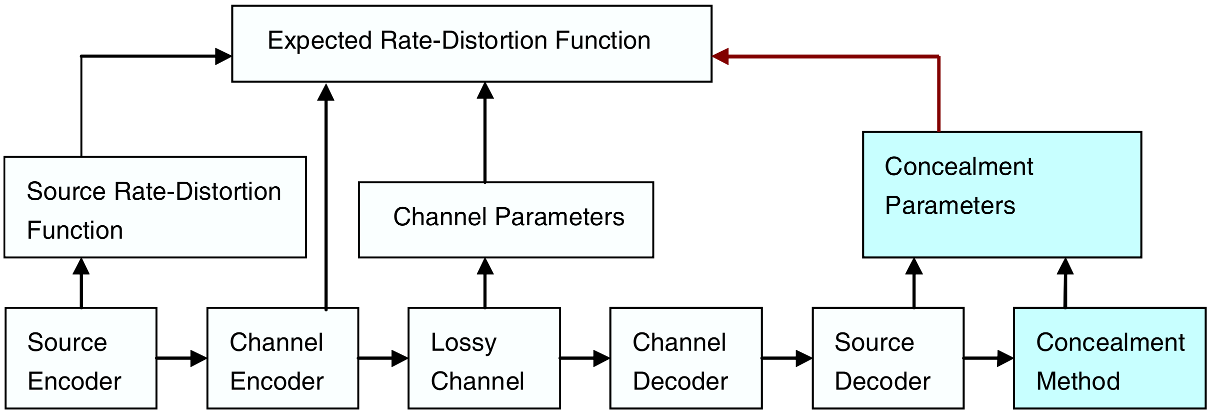


Figure 1: A pictorial diagram of our framework. Links from the concealment method to the concealment parameters and then to the expected rate-distortion function form the main conceptual path of the framework.

tion about the formulation and realization of our framework. In Section 3, we formulate the problem and propose our solution. Section 4 compares our results with those of other methods. Finally, in Section 5, we present our conclusions.

2 Overview of Existing Techniques

In this section, we review the techniques used to formulate our framework. First, we consider a 3D scalable video codec and an error concealment method; and then introduce the channel model and an error control method. The reviewed methods do not necessarily produce the best results; however, they simplify the presentation and analysis of our framework so that we can concentrate on our major conceptual and technical developments. Other methods, not reviewed, could also be applied to our framework after appropriate modifications.

A. 3D Scalable Video Codec and Error Concealment Method

We use 3D-SPIHT [4], a 3D wavelet-based scalable codec that does not adopt temporal domain motion compensation, as our source coder. The codec is simple because it does not estimate motion [11, 12, 13]. The motion vectors are important performance parameters of a video codec that must be carefully protected during transmissions; however, this is not within the

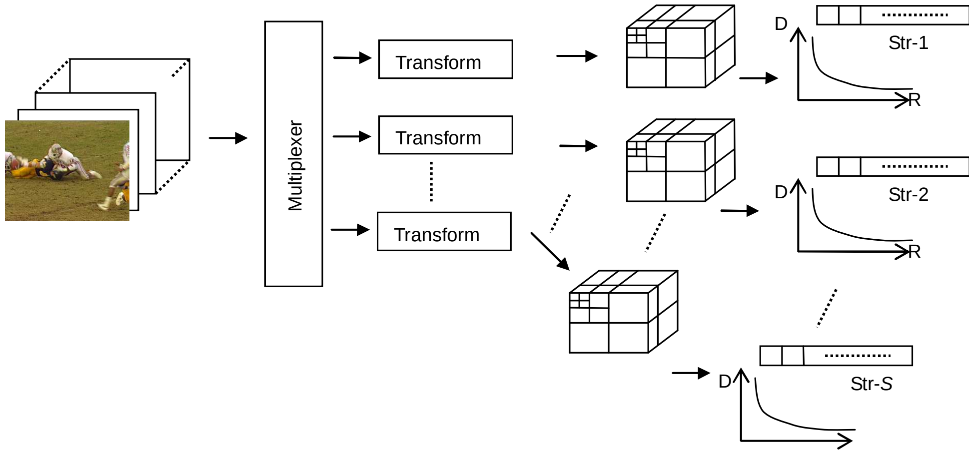


Figure 2: Generation of multiple bitstreams and rate-distortion functions

scope of the present study. To generate multiple bitstreams, we use a simple temporal-domain partitioning scheme, as shown in Figure 2. For example, to generate two bitstreams, we divide the even and odd frames in a group of pictures (GOP) to form two separate subsequences, each of which is independently encoded and quantized based on the transform coefficient bit-plane. The source bits are then divided into multiple layers so that the bits of a bit-plane correspond to a layer. Finally, the spatial and temporal bit-plane coefficients are ordered in a bitstream, as proposed in [9].

Although there are several successful concealment algorithms [6, 14, 15, 16, 17, 18], we are particularly interested in the low complexity concealment method proposed in [6] because it can be efficiently adapted to time-varying transmission environments. The method estimates the wavelet transform coefficient bit-plane of one bitstream by using its counterpart in the other bitstream. The estimated bit-plane of a coefficient, being either 0 or 1, is determined by minimizing the distortion metric, which measures the smoothness between the bitstreams in the wavelet transform domain. If bitstream-h conceals bitstream-l, then the distortion up to the d-th bit-plane is given by

$$D(d) = \sum_x c_h(d,x) - \hat{c}_l(d,x), \tag{1}$$

where x denotes a pixel of the frames in a GOP; $c_h(d,x)$ is the wavelet value of bitstream-h at x after decoding up to the d-th bit-plane; and $\hat{c}_l(d,x)$ is the estimated wavelet value of bitstream-l

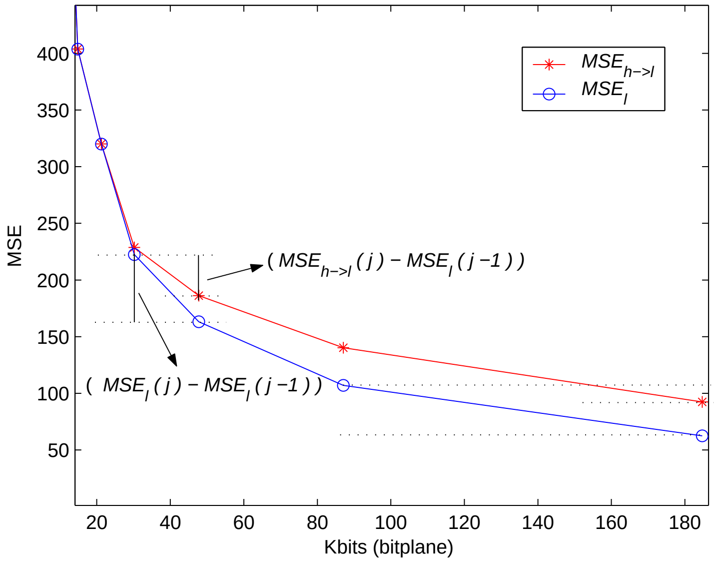


Figure 3: The top curve is an example of applying an effective concealment method to the concealed bitstream-l. The vertical distance $MSE_l(j) - MSE_l(j-1)$ is not less than $MSE_{h \to l}(j) - MSE_l(j-1)$.

at x after decoding up to the same bit-plane. The mean square error (MSE) of a bitstream after it has been decoded up to the j-th bit-plane is

$$MSE(j) = \sum_x (p(x) - p_j(x))^2 / (\text{total number of pixels in a GOP}),$$

where x is the pixel location of the video sequence and p(x) is the pixel value at x. We use the parameter $\beta^j_{h \to l}$, proposed in [19], to measure the efficiency of a concealment method as follows:

$$\beta^j_{h \to l} = \frac{MSE_{h \to l}(j) - MSE_l(j-1)}{MSE_l(j) - MSE_l(j-1)}, \tag{2}$$

where $MSE_{h \to l}(j)$ is the mean square error (MSE) of the concealed bitstream-l after it has been decoded up to the j-th bit-plane, and $MSE_l(j)$ is the MSE of the true bitstream-l decoded up to the same bit-plane. The example shown in Figure 3 demonstrates that, in an effective concealment method, the parameter has a value between [0,1] (the higher the better), which measures the proportion of the MSE reduced by concealing bit-plane-j of bitstream-l with bitstream-h.

B. Unequal Error Protection and the Channel Model

Unequal error protection (UEP) assigns an unequal number of channel bits to protect source segments with different priorities, which facilitates video streaming in packet loss environments.

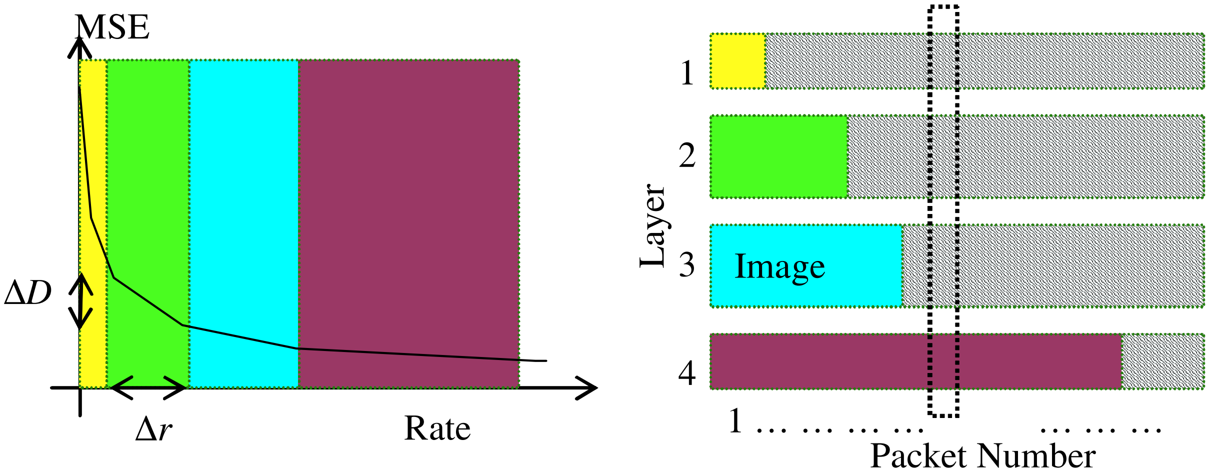


Figure 4: Multi-layer unequal error protection in PET. The gray areas in the right-hand sub-figure indicate the FEC protection bits.

The priority of a source segment is usually characterized by the ratio of the reduction in distortion to the number of bits used to encode the segment (λ = -$\Delta D/\Delta r$). Figure 4 shows the priority of different data segments in a scalable coder, where a segment with a larger ratio λ has a higher priority and should be protected by FEC with more protection bits. As shown in the figure, λ decreases as the layer number increases. Thus, the protection bits assigned to different layers should satisfy the constraint

$$c^1 > c^2 > ... > c^L, \tag{3}$$

where L is the total number of layers, and $c^j = n - k^j$ for the error correction code (n, k^j). The packet structure of priority encoding transmission (PET) satisfies the constraint. In PET, data in different layers is later interlaced into packets,which are then transmitted in the order shown in the right-hand sub-figure of Figure 4; the vertical box represents a packet. The packing structure ensures that each layer has the same number of lost packets. The sub-figure also shows that, as long as the received packets can correctly decode a layer, they can correctly decode data in any layer with a priority higher than their present layer.

The channel statistics of an Internet connection are usually obtained from reports of the Real Time Control Protocol (RTCP) [20]. To describe channel statistics, we use the two-state Markov model, which has been widely adopted in packet loss environments. The two states of the model are denoted as G (good) and B (bad). In state G, packets are received correctly, whereas in state B packets are lost. The model is fully described by the transition probabilities p_{GB} between states G and B, and p_{BG} between states B and G. The mean packet loss rate P_B and the average burst length L_B, which is the average number of consecutive symbol errors of the model are, respectively,

$$P_B = \frac{p_{GB}}{p_{GB} + p_{BG}}, \tag{4}$$

$$L_B = \frac{1}{p_{BG}} \tag{5}$$

We use the Reed-Solomon (RS) code for FEC because it is effective in recovering erased symbols when their locations are known. For the RS code operating on b-bit symbols, the maximum block length is $2^b - 1$ symbols. The RS code (n, k) can recover k source symbols correctly when the number of lost symbols is less than the minimum distance $d_{min} = n - k + 1$ of the code. For optimization, the RS code parameter is the channel coding rate $r_c = k/n$. The performance of an RS decoder can be characterized by the correct code probability

$$P_c(n, k) = \sum_{m=0}^{n-k} P(n, m), \tag{6}$$

where $P(n, m)$ is the probability of m erasures within a block of n symbols, derived analytically in [12] for the two-state Markov model.

3 Problem Formulation and Solution

To evaluate the proposed framework, we use an information theoretic approach, which formulates the framework as an expected rate-distortion optimization problem. We then propose an efficient and effective algorithm that obtains a sub-optimal solution of the problem.

3.1 Problem Formulation

Because of the complexity of formulating our problem, we give a step-by-step presentation. We begin with the simplest case, where there are two single-layer bitstreams, and then extend the

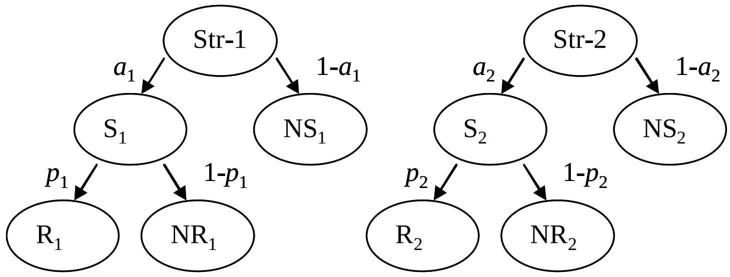


Figure 5: The state diagram of the single-layer two-bitstream case.

case to multiple layers. Interestingly, we encountered a new difficulty when we tried to formulate the general case of multiple layers with more than two bitstreams. In the case of two bitstreams, a lost bitstream can only be concealed by the other bitstream, but for more than two, we found that there are many candidates that can conceal such a bitstream. As the optimal combinatorial strategy for more than two bitstreams is still under research, we propose a simple and practical solution of the case.

3.1.1 Single-Layer: Two Bitstreams

We divide a single bitstream into two independent encoded bitstreams so that if only one is lost, the other can still maintain an acceptable decoded video quality. The encoder uses the state diagram shown in Figure 5 to analyze the case. The notation Str-s denotes a bitstream s, and the state S_s indicates that Str-s is sent to a receiver with a probability a_s. The parameter a_s is introduced because the performance improvement achieved by applying an error concealment method may be so good that it is not necessary to send one of the bitstreams. The state NS_s means that Str-s is not sent to the receiver with a probability $(1 - a_s)$. However, once the bitstream is sent, the receiver may not be able to recover it correctly. We use p_s to indicate the probability that the receiver can decode the transmitted Str-s correctly. In state R_s, the Str-s is received correctly, while in state NR_s, the bitstream is lost. We use ΔD_s to denote the reduction in distortion of Str-s. This value is always a nonnegative number. The expected distortion function can be derived from the state diagram in which we show the derivation for Str-1, but omit it for Str-2 because it can be derived similarly.

These are three causes of distortion reduction in bitstream 1. Case 1: Str-1 is sent and correctly received. The distortion reduction is ΔD_1 with a probability $a_1 p_1$. In this case, Str-2 makes no contribution to the distortion reduction of Str-1. Case 2: Str-1 is not sent, while Str-2 is sent and correctly recovered. Hence, Str-1 is concealed by Str-2. We use $\Delta D_{2\to 1}$ to denote the distortion reduction of Str-1 after it has been concealed by Str-2, and define $\beta_{2\to 1} = \Delta D_{2\to 1}/\Delta D_1$, which is the proportion of the distortion recovered by the concealment. The distortion reduction of this case is therefore $\beta_{2\to 1}\Delta D_1$, with a probability $a_2 p_2(1-a_1)$. Case 3: both bitstreams are sent; however, only Str-2 is correctly received. In this case, the probability is $a_2 p_2 a_1(1-p_1)$, Str-1 is concealed by Str-2, and the distortion reduction is $\beta_{2\to 1}\Delta D_1$. We summarize the cases where Str-1 can reduce distortion by the expected amount as follows:

$$\begin{aligned} E[\Delta D_1] &= \Delta D_1(a_1 p_1 + \beta_{2\to 1} a_2 p_2((1-a_1) + a_1(1-p_1))) \\ &= \Delta D_1(a_1 p_1 + \beta_{2\to 1} a_2 p_2(1-a_1 p_1)). \end{aligned} \tag{7}$$

Similarly, we can obtain the total expected distortion function for Str-2, which is

$$\Delta \bar{D} = \sum_s E[\Delta D_s]. \tag{8}$$

In our approach, there is a possibility that a bitstream will not be sent. As a consequence, the total transmission rate depends on the transmission parameters as well as the error correction code. The expected rate for Str-s using $(N_s\ ,\ k_s)$ as the error correction code is

$$E[\Delta r_s] \approx \Delta r_s a_s \frac{N_s}{k_s}, \tag{9}$$

where Δr_s is the source bit of Str-s. The total expected rate is therefore,

$$\Delta \bar{r} = \sum_s E[\Delta r_s] \le R, \tag{10}$$

where R is the rate bound for the video transmission. The optimization problem involves searching for the parameters a_s, the transmission scheme, and the channel bits $c_s = N_s - k_s$ that maximize the expected distortion reduction $\Delta \bar{D}$ under the constraint $\Delta \bar{r} \le R$.

3.1.2 Multiple-Layers: Two Bitstreams

We now extend our derivation from a single-layer with two bitstreams to multiple-layers with two bitstreams. Let us assume that each bitstream is divided into L quality layers. We simply

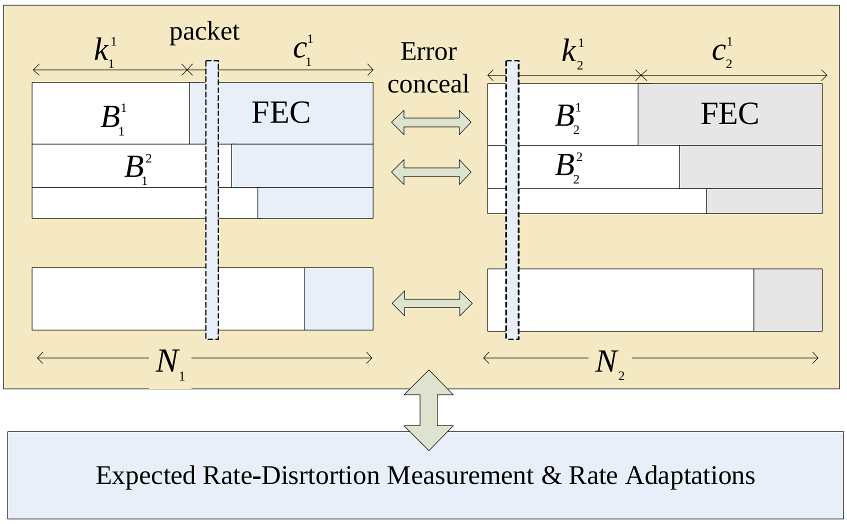


Figure 6: The PET structures for two bitstreams, each of which contains multiple layers.

divide a bitstream into quality layers according to the number of bit-planes in such a way that the bits in a bit-plane belong to a quality layer. A PET structure is used to pack the layered data of a bitstream; therefore, there are two PETs. For simplicity, we assume that, if a layer of a bitstream is lost, it can only be concealed by the same layer of the other bitstream; however, this convenient assumption is unnecessary in a more general framework.

Let b_s^j be the source data of layer-j of bit-stream s, and $\mathbf{B} = \{b_s^j\}$ represent all the source data. The layered data of a bitstream is divided and packed in an l by k array so that the protection bits of different layers satisfy $c_s^1 > c_s^2 > ... > c_s^L$. The lower quality data in a higher layer can only be recovered correctly after we decode the higher quality data in a lower layer. As shown in Figure 6, k_s^j is the source data of an (N_s, k_s^j) code. Because the source data of layer-j, b_s^j, is sometimes not divisible by k_s^j, the amount of source data protected by the (N_s, k_s^j) code is

$$B_s^j = \left\lfloor \frac{b_s^{j-1} - B_s^{j-1} + b_s^j}{k_s^j} \right\rfloor \times k_s^j. \tag{11}$$

Therefore, we have

$$l_s^j = \frac{B_s^j}{k_s^j}. \tag{12}$$

We use c_s^j to represent the channel bits of layer-j of bitstream-s, and $\mathbf{C} = \{c_s^j\}$ to represent the channel bits of all bitstreams and layers. The source data accumulated up to and including layer-j of bitstream-s is

$$B_s(j, \mathbf{C}) = \sum_{q=1}^{j} B_s^q. \tag{13}$$

We use $MSE(B_s(j, \mathbf{C}))$ to denote the mean square error distortion when a receiver decodes $B_s(j, \mathbf{C})$ data. The distortion reduction achieved when layer-j of Str-s is correctly received is therefore

$$\Delta D_s^j = MSE(B_s(j-1, \mathbf{C})) - MSE(B_s(j, \mathbf{C})). \tag{14}$$

Let a_s^j denote the probability that layer-j of bitstream-s will be sent, and let the matrix $\mathbf{A}$ represent all a_s^j. Also, let $\mathbf{E} = \{\beta_{q \to s}^j\}$, where $\beta_{q \to s}^j$ measures the efficiency of using St-q to conceal layer-j of Str-s. Next, we derive the expected distortion reduction when a receiver decodes layer-j. Because the concealment is performed by the same layer in different bitstreams, we can calculate the expected distortion reduction of layer-j in the same way that we derive the function for the single-layer, two-bitstream case. The expected distortion of layer-j in this case is

$$\begin{aligned} \Delta \bar{D}^j(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{E}) &= \Delta D_1^j (a_1^j p_1(c_1^j) + \beta_{2 \to 1}^j a_2^j p_2(c_2^j)(1 - a_1^j p_1(c_1^j))) \\ &+ \Delta D_2^j (a_2^j p_2(c_2^j) + \beta_{1 \to 2}^j a_1^j p_1(c_1^j)(1 - a_2^j p_2(c_2^j))), \end{aligned} \tag{15}$$

where the terms beginning with ΔD_1^j and ΔD_2^j on the right are, respectively, the expected distortion of Str-1 and Str-2; and $p_s(c_s^j)$ is the recovery probability of layer-j of Str-s derived by using c_s^j channel bits for the layer. The overall expected distortion reduction for all the L layers is

$$\Delta \hat{D}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{E}) = \sum_{j=1}^{L} \Delta \bar{D}^{(j)}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{E}). \tag{16}$$

Similar to the single-layer, two-bitstream case, the rate constraint is given by

$$\Delta \hat{r}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{E}) = \sum_{j=1}^{L} \Delta \bar{r}^j(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{E}) \leq R, \tag{17}$$

where R is the total rate, and the expected rate allocated to layer-j is

$$\Delta \bar{r}^j(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{E}) = \sum_{s=1}^{2} a_s^j B_s^j \frac{N_s}{N_s - c_s^j}. \tag{18}$$

The parameters in **B** and **E** are source information that do not relate to the channel statistics. Thus, we only search for the optimal parameters in the transmission scheme **A** and the channel bit assignment **C**. Now, we can formulate our problem as a rate-distortion optimization problem as follows:

$$\max_{\mathbf{A},\mathbf{C}} \Delta\hat{D}(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{E}) \text{ subject to } \Delta\hat{r}(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{E}) \leq R. \tag{19}$$

Although this equation is derived according to a two-bitstream case, it can be extended to more than two bitstreams after appropriately modifying the expected distortion function; however, the modification is not trivial. In the case of two bitstreams, the lost bitstream is always concealed by the other bitstream. However, if there are more than two bitstreams, we encounter a new difficulty in that any correctly recovered bitstream may be used to conceal a lost bitstream. In the following, we present a simple way to extend our method to more than two bitstreams.

3.1.3 Multiple-Layers: More Than Two Bitstreams

When the number of bitstreams, S, is greater than two, a lost bitstream may be concealed by any combination of correctly recovered bitstreams. Because finding the optimal subset that can conceal a lost bitstream is computationally infeasible, we propose the following practical solution. Note that we assume the lost data in a layer of a bitstream can only be concealed by the correctly received data of the same layer in another bitstream.

Our concealment strategy uses the other bitstreams one at a time to conceal a lost bitstream. This corresponds to modelling the method by a bipartite graph with S nodes in each column. An arc between two nodes in the graph indicates that one node can conceal the other node. Figure 7 shows an example of our graph for three bitstreams. As shown in the figure, Str-1 can be concealed by Str-2 or Str-3. The priority of the bitstream used to conceal Str-1 is given in the polling table of Str-1. Note that the order of concealing Str-1 is $\beta_{3\to 1}$ above $\beta_{2\to 1}$, indicating that if Str-1 is lost, a decoder will poll Str-3 first. If Str-3 is correctly recovered, then it is used to conceal Str-1. Otherwise, the decoder polls Str-2 and uses the correctly received Str-2 to conceal Str-1. The ordering is arranged according to the concealment performance, as measured by the concealment parameter $\{\beta^j_{p\to s} | j = 1, \cdots, L; s, q = 1, \cdots, S; s \neq q\}$. The higher the value of a bitstream used to conceal the target bitstream, the higher that bitstream will be in the polling table of the target bitstream. The encoder computes all the concealment coefficients, ranks them to form the tables, and sends the tables to the receivers as side information. This strategy is computationally practical because it reduces the number of all possible concealment subsets from $S2^{S-1}$ to $S(S-1)$. However, this ordering needs extra bits to send the side information of the tables. An alternative approach, without side information, is to enforce the concealment according to a pre-given order, such as an incremental order. That is, the order for concealing Str-s is $(s+1)$ mod S, followed by $(s+2)$ mod S, etc. We can extend our strategy to more than three bitstreams in a similar way, but it increases the notational complexity; thus, we do not describe it here. For the case in Figure 7, the expected rate-distortion function for layer-j of Str-1 is

$$E[\Delta D_1^j] \;=\; \Delta D_1^j(a_1^j p_1^j + (1 - a_1^j p_1^j)(\beta_{3\to1}^j a_3^j p_3^j + \beta_{2\to1}^j a_2^j p_2^j(1 - a_3^j p_3^j))), \tag{20}$$

where the term $(1 - a_1^j p_1^j)a_3^j p_3^j$ is the probability that layer-j of Str-3 will be used to conceal layer-j of Str-1, while $(1 - a_1^j p_1^j)a_2^j p_2^j(1 - a_3^j p_3^j)$ is the probability that layer-j of Str-2 will be used to conceal layer-j of Str-1. The overall expected distortion of layer-j is the sum of all the bitstreams, given as:

$$\begin{aligned}
\Delta \bar{D}^{(j)}(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{E}) \;=\; & \Delta D_1^j(a_1^j p_1^j + (1 - a_1^j p_1^j)(\beta_{3\to1}^j a_3^j p_3^j + \beta_{2\to1}^j a_2^j p_2^j(1 - a_3^j p_3^j))) \\
+\; & \Delta D_2^j(a_2^j p_2^j + (1 - a_2^j p_2^j)(\beta_{1\to2}^j a_1^j p_1^j + \beta_{3\to2}^j a_3^j p_3^j(1 - a_1^j p_1^j))) \\
+\; & \Delta D_3^j(a_3^j p_3^j + (1 - a_3^j p_3^j)(\beta_{2\to3}^j a_2^j p_2^j + \beta_{1\to3}^j a_1^j p_1^j(1 - a_2^j p_2^j))).
\end{aligned} \tag{21}$$

Using similar derivations to those in Equations 16, 17, and 18, the rate-distortion optimization problem, shown in Equation 19, can be formulated for this case. We omit the detailed derivation because it is a simple extension of our previous derivations.

3.2 Fast Algorithm

We propose an efficient algorithm that finds $\mathbf{A}$ and $\mathbf{C}$ to optimize the rate-distortion function. First, we discuss the procedure for finding $\mathbf{C}$, followed by that for $\mathbf{A}$. To find the optimal $\mathbf{C}$, i.e., to solve the bit allocation problem, we encounter two difficulties: how to assign bits to each

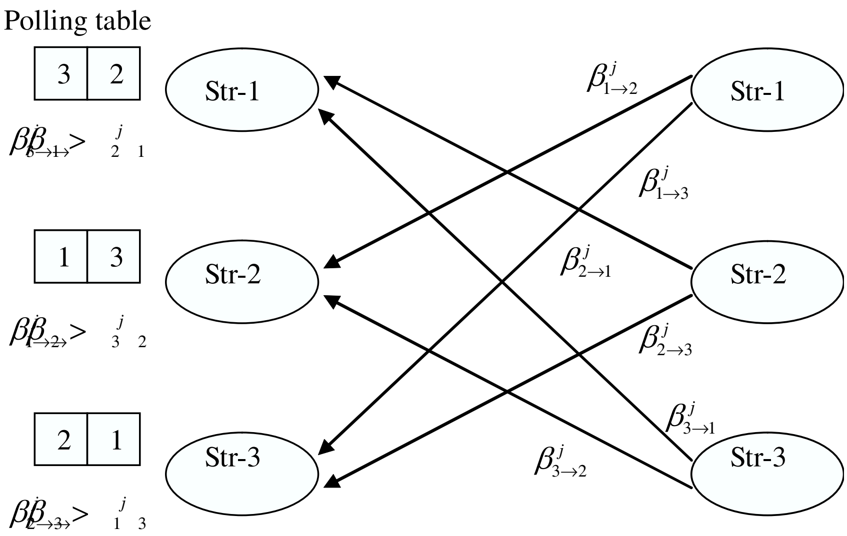


Figure 7: The graph represents a simplified implementation of transmitting three bitstreams. The graph for more than three bitstreams can be generated easily. The polling table of each layer of a bitstream records the order in which the other bitstream conceals the layer.

bitstream and how to assign protection bits to protect each source layer in a bitstream. We use a heuristic approach to solve the first difficulty. Because we use temporal partitioning to generate bitstreams, it is intuitively correct to assume that each bitstream takes the same number of bits. Therefore, we only deal with the second difficulty. In Kim et al. [6], the framework does not include error-concealment in the analysis, and the channel bit assignment to a bitstream is independent of the assignment of bits to the other bitstreams. However, our analysis shows that error concealment induces a dependency between the channel bits of different bitstreams. Therefore, the method in [6] can not be applied straightforwardly to find a solution for our case. In the following, we present an efficient algorithm that solves the problem in a general case.

The optimal solution for **C** can be found by using an exhaustive search method. However, such methods are unrealistic for real-time video transmissions because of the excessive computation time required. We therefore developed a local hill-climbing algorithm that makes limited assumptions about the data, but is computationally tractable. Our method is inspired by the algorithm proposed in [1], which is designed to assign a sub-optimal **C** to protect multiple layers in a single PET. We extend the algorithm to assign a sub-optimal **C** to protect multiple layers

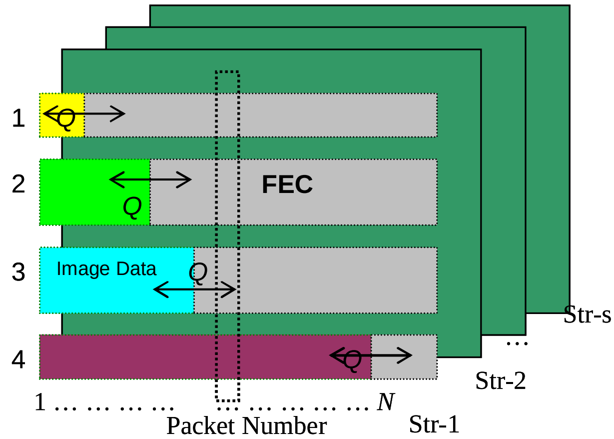


Figure 8: The search range of each iteration is $2Q$.

of more than one PET for each configuration of $\mathbf{A}$.

Initially, each layer only contains the source; $k_s^j = N_s$ and $c_s^j = 0$ for all j and s, respectively. In each iteration, for each PET, our algorithm examines a number of possible assignments that could be equal to $2QL$, where Q is the search distance, and $2QL$ corresponds to the maximum number of FEC symbols that can be added to or subtracted from a bitstream of L layers in one iteration. We determine $\Delta\hat{D}$ after adding or subtracting between 1 and Q FEC symbols in each layer of PET, while satisfying the constraint $c_s^j \geq c_s^{j+1}$ of PET. We choose the $\mathbf{C}$ corresponding to the highest $\Delta\hat{D}$, update the allocation of FEC symbols to all affected layers, and repeat the search until none of the cases examined improves the expected distortion reduction. The pseudo code of our algorithm is given in Figure 9.

This hill-climbing algorithm finds a local maximum that is reasonably close to the global maximum and, in some cases, may be identical to it. The search distance Q is a pre-defined parameter of the algorithm. There is clearly a tradeoff: the larger the value of Q, the higher the probability that the algorithm will find the global optimum, because it will require more time to run. Note that, for every symbol of FEC data added to a layer, a source symbol needs to be moved to the next row. We start at the first row affected by the new allocation, move its last data symbol to the next row, move the last data symbol of that row to the following row, and so on. As a result, a cascade of data symbols moves down the rows until the rate constraint R is satisfied. This part of the algorithm is based on the assumption that the compressed sequence

```
best[*]:=(0,0,...,0)
Until best[*]:=last[*] Do:
      last[*]:=best[*]
      For each substream s from 1 to S:
            For each layer j from 1 to L:
                  For each search_value from -Q to +Q
                        temp[*]:=last[*]
                        temp[j]:=temp[j] + search_value
                        If temp[j]<0 or temp[j]>N then continue to next search_value
                              If search_value > 0 then for all i>j
                                    Do temp[i]:= min(temp[j],temp[i])
                              Else for all i <j
                                    Do temp[i]:= max(temp[j],temp[i])
                              End if
                             Calculate expected MSE reduction for temp[*] using Eq 19
                             If dMSE(temp[*])>dMSE(best[*]) then
                                    best[*]:=temp[*]
                             End if
                  End For each
            End For each
      End For each
End until
```

Figure 9: The pseudo code of our fast algorithm

is progressive, because the data that we discard is the least important information embedded in the bitstream. The algorithm derives a set of error correction codes of different strength in different bitstreams.

We assume that each element of $\mathbf{A}$ is either 0 or 1, where 0 means the data is not sent; and 1 means the data is sent. With the binary assignment of each element in $\mathbf{A}$, we can enumerate all possible values of $\mathbf{A}$. For each value, we use the algorithm to search for $\mathbf{C}$ that gives a sub-optimal solution, and take the pair of $\mathbf{A}^*$ and $\mathbf{C}^*$ that gives the maximum distortion reduction as our solution. Note that $\mathbf{C}^*$ and the polling table of each bitstream need to be sent to the decoder as side information. How this is implemented depends on the system used. We can save more bits by not sending the polling tables if we impose an order to conceal all the bitstreams. This is known as a *priori* ordering of encoders and decoders and does not need to be sent as side information.

4 Implementation and Experiment Results

We now describe an implementation of our approach and then compare our results with those of other methods. Our test sequences are the Akyio, Foreman, and Hall sequences in CIF format, and the Football sequence in SIF format. We use a 2-state Markov model to describe the lossy channel and use the RS code as the FEC code. Our subsequences are obtained by using temporal partitioning to divide a video sequence. All subsequences are assigned an equal number of bits, and independently encoded to obtain a progressive bitstream by using the 3D-SPIHT algorithm. We perform three levels of spatial and temporal decomposition using the 9-7 and Harr filters, respectively. We only conceal the wavelet coefficients corresponding to low frequencies of spatial and temporal components, indicated by the gray area in Figure 10, because the performance gain of recovering those coefficients is usually higher than that of recovering the rest. Recall that we described the concealment method in Section 2. Figure 11 shows the average values of the concealment parameters $\beta^j_{2\to1}$ and $\beta^j_{1\to2}$, calculated according to Equation 2, in different layers of various sequences for two bitstreams. As shown in the figure, the slow motion sequences and the lower layers achieve better concealment, because the bitstreams in both cases are similar; therefore, the average concealment performance is higher.

The following experiment shows that, even though some layers of a bitstream are not sent, we can still achieve a better performance than by sending all the layers of the bitstream. Figure 12 shows that the results of our method with different configurations of **A**, labeled Framework-A, Framework-B, and Framework-C, achieve a significant PSNR gain over not performing concealment at all. At a bit rate below 20 Kbps for all the sequences, Framework-B and Framework-C, which do not send some layers, perform better than Framework-A, which sends all layers. This indicates that, at low bit rates, the performance degradation by not sending some layers may be completely compensated for by using the concealment from the same layers of the other bitstream. As the bit rate increases, Framework-A achieves the best performance, because enough bits are used for FEC to protect all the bitstreams; even so, the curves of Framework-B and Framework-C are very close to the curve of Framework-A. The loss in performance is due to errors in the concealment of the lost layers. Such errors occur even when the data in the higher layers is correctly recovered. Figure 13 compares the performance of various methods with different mean packet loss rates. The performance decrease of our approach is graceful as the mean packet loss rate increases.

We compare the performance results of our method with those of Kim et al.'s method [6], which differs from our approach in the encoder's design and in the channel bit allocation algorithm. In [6], the authors do not incorporate the concealment in the rate-distortion analysis, which corresponds to our special case when all layers of all bitstreams are sent and all concealment parameters are set to 0. In the channel bit allocation algorithm, the optimal channel bits are allocated to all layers of each bitstream independently using dynamic programming. However, in our algorithm, because of the concealment, the same layers in different bitstreams are correlated; therefore, the channel bit allocated to a layer of a bitstream depends on the bit allocated to the same layer of the other bitstream. Since the correlation increases the complexity of using dynamic programming, we propose a fast algorithm to solve the allocation problem. Figure 14 shows the ratio of source bits assigned to different layers by different methods. The channel bits allocated to the same layers of different bitstreams of the method in [6] may be different to those allocated by our method because the bits can be compromised by concealment.

Figure 15 compares the results of our method to those of the other methods for two bitstreams. For all bit rates, our method outperforms the other approaches. In addition, our PSNR performance is higher than that of [6] by an average of more than 2 dB. The performance gain is mainly due to our inclusion of the concealment in the rate-distortion analysis. In our framework, a layer is protected by FEC as well as concealment, thus fewer FEC bits need to be assigned to protect the layer. Consequently, our method has extra bits to encode the source data. As is shown in Figure 15, our method's performance improves as the bit rate increases. In Figure 16, we compare the performance of the methods versus their mean packet loss rate. When the latter is large, most bits are used for FEC; thus, the performance gain of our method over that of [6] declines. Figure 17 compares some snapshots of the different methods under various conditions. As the examples show, our method produces images with better contrast and perceptual quality than those of [6].

Figures 18 compares the performance of our method for different numbers of bitstreams using various sequences. Sub-figures 18(a) and (b) show the performance of source coding. The performance of three bitstreams is worse than that of two bitstreams for all bit rates of sequences. However, as shown in 18(c) and (d), with our approach, the performance of three bitstreams is better than that of two bitstreams for all sequences. In our approach, a data layer in the three-bitstream case has more protection from concealment than the same data in the two-bitstream case. This is because either layer of the other two bitstreams can conceal the data; hence, the case of three bitstreams yields a better result. To summarize our observations of (a),(b), (c), and (d), the source coding performance deteriorates as the number of bitstreams increases, whereas our method improves the performance when the number of bitstream increases. Figure 19 compares the performance of the proposed method in an unbalanced channel and a balanced channel environment, each of which contains two bitstreams. In the former, the channel's statistics are different for each bitstream. For example, in our experiments, the mean packet loss rate is 0.1 for one bitstream and 0.3 for the other bitstream. The curves of the unbalanced channel in all cases are above those of the balanced channel, which has a mean packet loss rate of 0.2 for each bitstream. Transmitting in unbalanced channels allows more flexibility to compromise between concealment and FEC, and hence improves the performance. Finally, Figure 20 shows the computation time of the proposed algorithm. The time was measured on a Pentium4 1.6 GHz PC with a 512 RAM in the Matlab environment. We applied our algorithm to 32 frames for two bitstreams of the Akiyo sequence with different bit rates. The experiment was performed ten times for each bit rate to obtain the average time, which was less than 1 second for the 32 frames.

5 Conclusion

We have proposed a new framework in which the encoder incorporates the concealment in the rate-distortion analysis. The concealment induces a correction between different bitstreams. We formulate the framework as a rate-distortion optimization problem, and propose a fast algorithm to solve it. Our approach has the advantage that data in a layer can be protected from concealment and FEC. As a consequence, fewer FEC bits need to be used; therefore, more source data can be transmitted. Compared to the algorithm in [6], which does not include concealment in the encoder's design, our algorithm achieves an improvement of more than 2dB in the PSNR of various video sequences. Note that we did not use a state-of-art 3D wavelet codec to perform our simulations. Also, for simplicity, we did not incorporate important video compression features, such as prediction and data partitioning techniques, into our framework. In our future work, we will extend the framework to include the features so that it can be applied to advanced video codecs.

References

[1] A. Mohr, E. Riskin, and R. Ladner, "Unequal loss protection: Graceful degradation of image quality over packet erasure channel through forward error correction," *IEEE J. Sel. Areas Commun.*, vol. 18, pp. 818–828, Oct. 2000.

[2] R. Puri, K. W. Lee, K. Ramchandran, and V. Bharghavan, "An integrated source and congestion control framework video streaming in the internet," *IEEE Trans. Multimedia*, vol. 1, pp. 18–12, Mar. 2001.

[3] J. Kim, R. M. Mersereau, and Y. Altunbasak, "Error-resilient image and video transmission over the internet using unequal error protection," *IEEE Trans. Image Process.*, vol. 12, pp. 121–131, Feb. 2003.

[4] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-d set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1374–1387.

[5] C. Creusere, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," *IEEE Trans. Image Process.*, vol. 6, pp. 1436–1442, Oct. 1997.

[6] J. Kim, R. M. Mersereau, and Y. Altunbasak, "A multiple-substream unequal error-protection and error-concealment algorithm for spiht-coded video bitstreams," *IEEE Trans. Image Process.*, vol. 13, pp. 1547–1553, Dec. 2004.

[7] S. Cho and W. A. Pearlman, "Error resilient compression and transmission of scalable video," *Proc. SPIE*, vol. 4115, pp. 396–405.

[8] A. A. Alatan, M. Zhao, and A. N. Akansu, "Unequal error protection of SPIHT encoded image bit streams," *IEEE J. Sel. Areas Commun.*, vol. 18, pp. 814–818, Jun. 2000.

[9] S. Cho and W. A. Pearlman, "A full-featured, error-resilient, scalable wavelet video codec based on the set partitioning in hierarchical trees algorithm," *IEEE Trans. CSVT*, vol. 12, pp. 157–171, Mar. 2002.

[10] J. Kim, R. M. Mersereau, and Y. Altunbasak, "Distributed video streaming using multiple descryiption coding and unequal error protectiton," *IEEE Signal Processing*, vol. 14, pp. 849–861, July 2005.

[11] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (limat) framework for highly scalable video compression," *IEEE Trans. Image Process.*, vol. 12, pp. 1530V1542, Dec. 2003.

[12] S.-J. Choi and J.W. Woods, "Motion-compensated 3-d subband coding of video," *IEEE Transactions on Image Processing*, vol. 8, pp. 155–167, Feb. 1999.

[13] J.-R. Ohm, "Three-dimensional subband coding with motion compensatio," *IEEE Transactions on Image Processing*, vol. 3, pp. 559–571, Sep. 1994.

[14] I.V. Bajic and J.W. Woods, "Domain-based multiple description coding of images and video," *IEEE Trans. Image Process.*, vol. 12.

[15] L. Atzori, G. Ginesu, and A. Raccis, "JPEG2000-coded image error concealment exploiting convex sets projections," *IEEE Trans. Image Process.*, vol. 14, pp. 487 – 498, Apr. 2005.

[16] S. Belfiore, M. Grangetto, E. Magli, and G. Olmo, "Concealment of whole-frame losses for wireless low bit-rate video based on multiframe optical flow estimation," *IEEE Transactions on Multimedia*, vol. 7, pp. 316–329, Apr. 2005.

[17] S. Belfiore, M. Grangetto, E. Magli, and G. Olmo, "Frame concealment for H.264/AVC decoders," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 227–233, Feb. 2005.

[18] P. C Chung, C. W. Wu, and Y. L. Huang, "A jpeg 2000 error resilience method using uneven block-sized information included markers," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 420–424, March 2005.

[19] C. M. Fu, W. L. Hwang, and C. L. Huang, "Error concealment protection for loss resilient bitplane-coded video communications," *IEEE Proc. on ISCAS'06.*

[20] X. Yang, C. Zhu, Z. G. Li, X. Lin, and N. Ling, "An unequal packet loss resilience schemefor video over the internet," *IEEE Transactions on Multimedia*, vol. 7, pp. 753–765, August 2005.

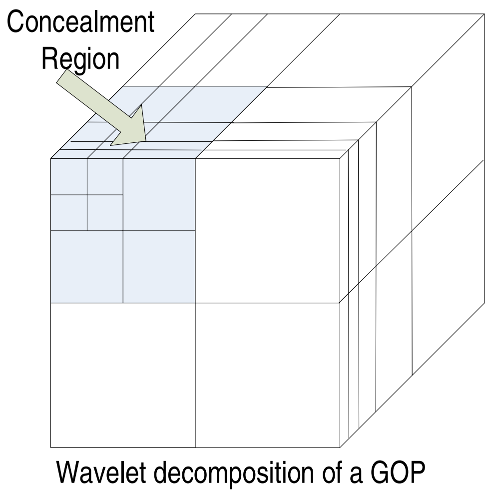


Figure 10: The shaded area indicates the wavelet coefficients that are refined using an error-concealment algorithm.

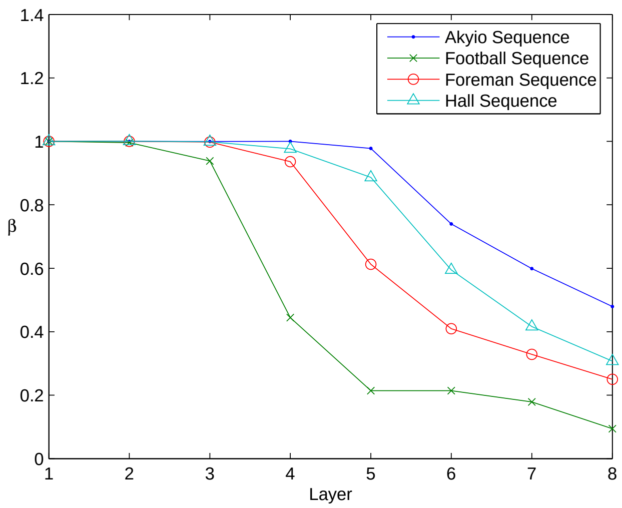


Figure 11: The vertical axis measures the average concealment efficiency of a GOP using the first 32 frames of different sequences on different layers. The efficiency of the concealment is higher at lower layers, corresponding to the coarse information of the subsequence.

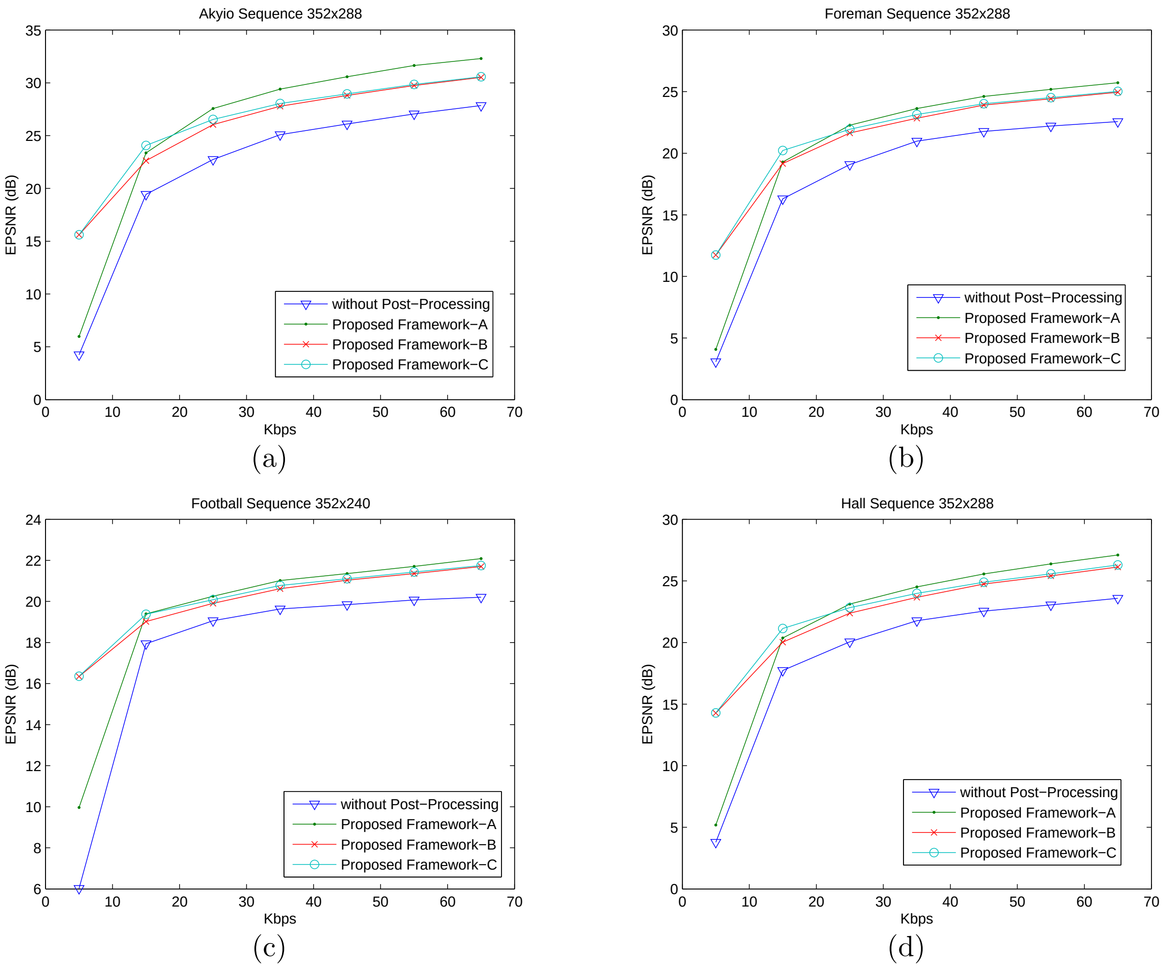


Figure 12: Comparison of the performance of different methods with various bit rates for two bitstreams. The curves labeled "without post-processing" are assigned optimal FEC for each layer of a bitstream. Note that neither the encoder nor the decoder perform concealment. The transmission parameters of Framework-A are $a_s^j = 1$ for all j, and $s = 1, 2$; those of Framework-B are $a_2^1 = 0$, and all the other entries are set to one; and those of Framework-C are $a_2^j = 0$ for $j = 1, 2$ and all the other entries are set to one. The mean packet loss rate is 5% and the average burst length is 5. Note that if a_s^j is set to 1, then layer-j of bitstream s is sent; otherwise, it is not sent.

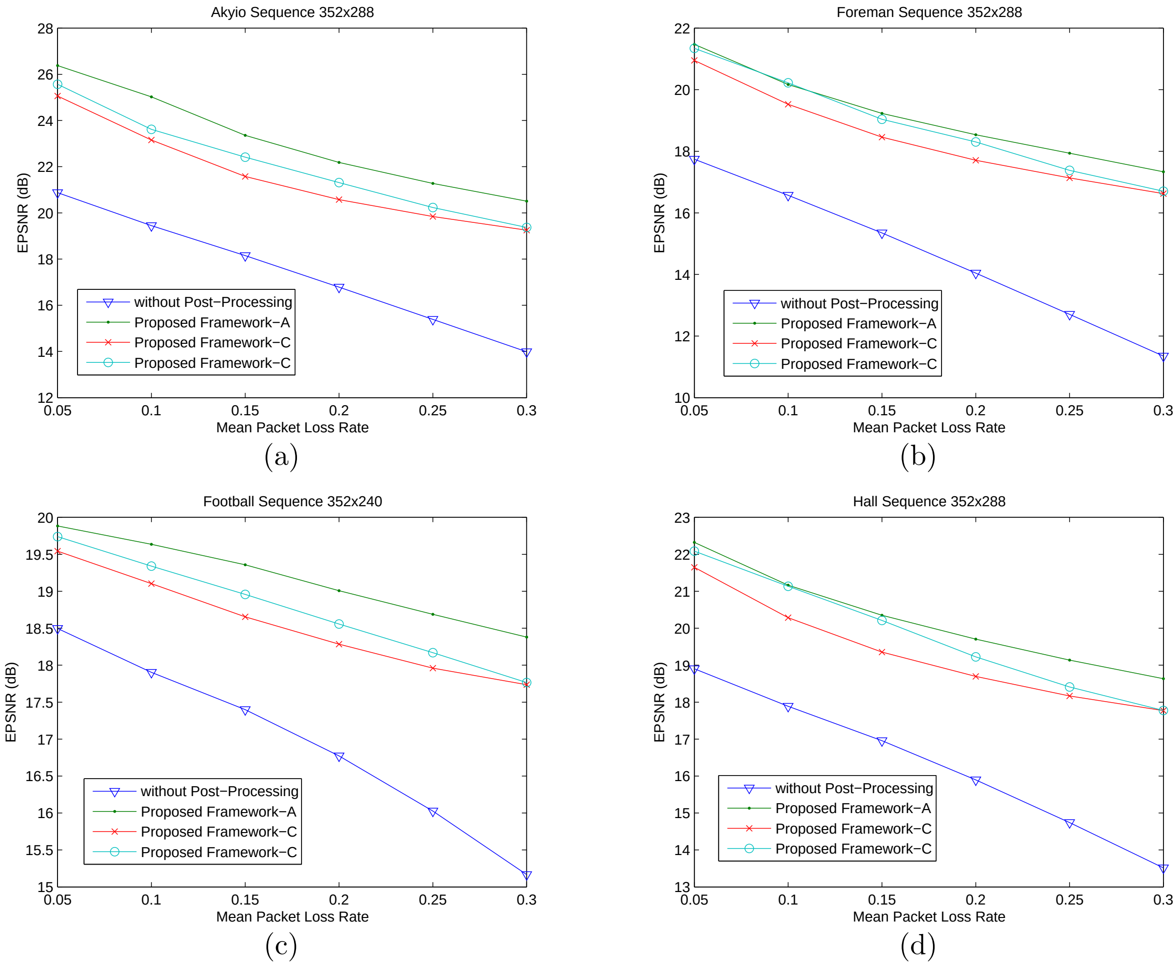


Figure 13: Comparison of the performance of different methods with various mean packet loss rates at 20 Kbps. The other parameters of each method are the same as those in Figure 12.

	Layer-1	Layer-2	Layer-3	Layer-4	Layer-5	Layer-6
Proposed in [6]	0.87	0.58	0.42			
	0.90	0.58	0.42			
Proposed Method-A	0.87	0.58	0.48			
	0.55	0.52	0.48	0.35	0.32	
Proposed Method-B	0.90	0.58	0.39			
		0.52	0.48	0.45	0.32	

Figure 14: The ratio of source bits, $\frac{k}{N}$, allocated to each layer of two bitstreams of the Akiyo sequence at 65 Kbps. Our method assigns different bits to a layer in different bitstreams, while the method in [6] assigns the same bit to a layer in different bitstreams.

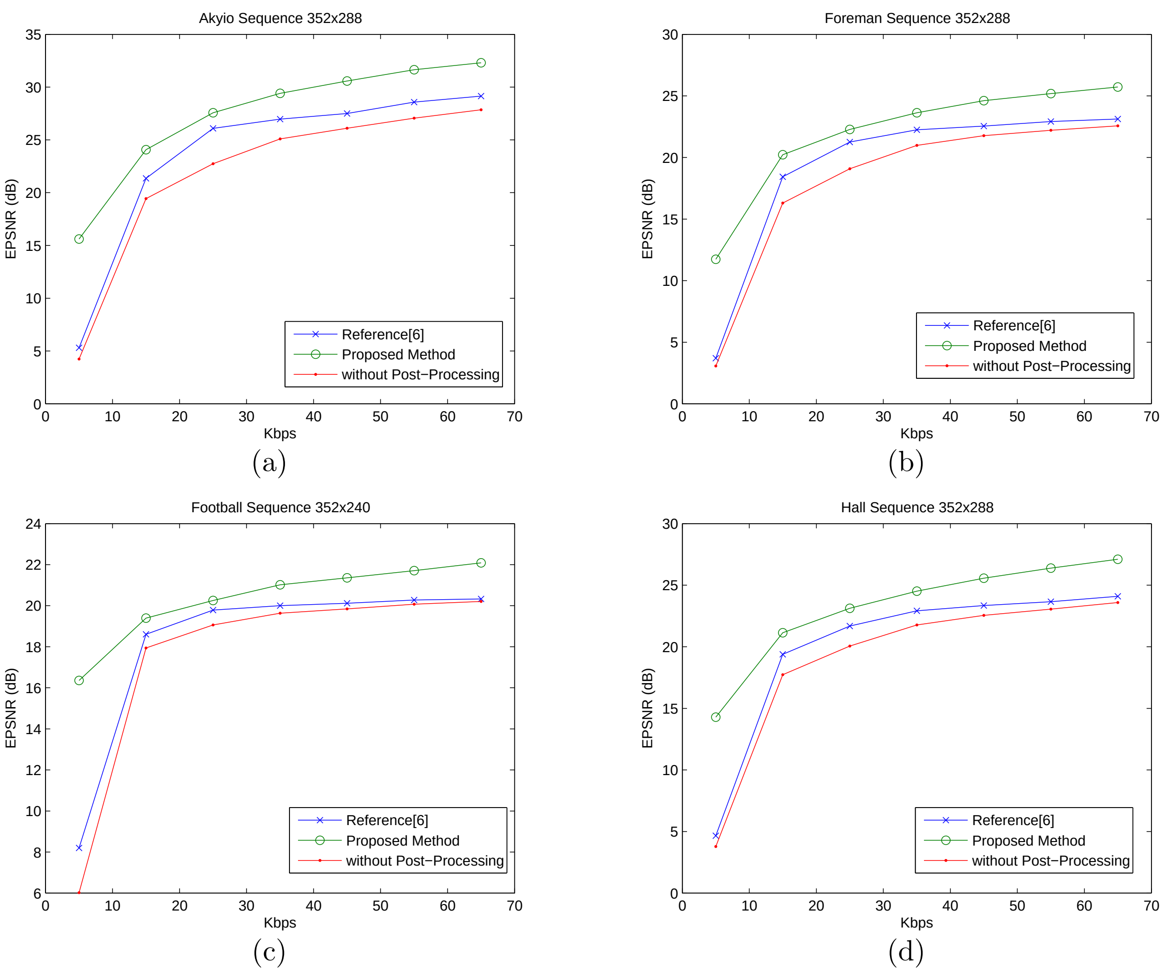
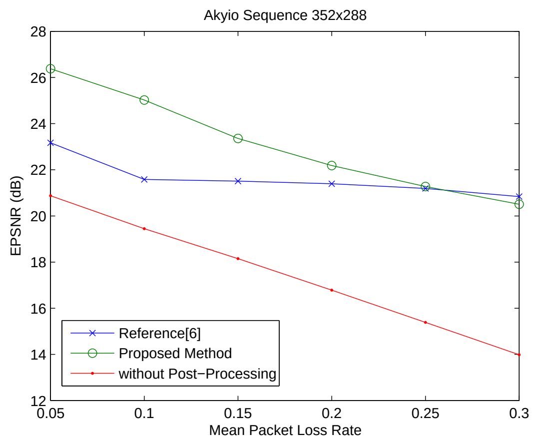
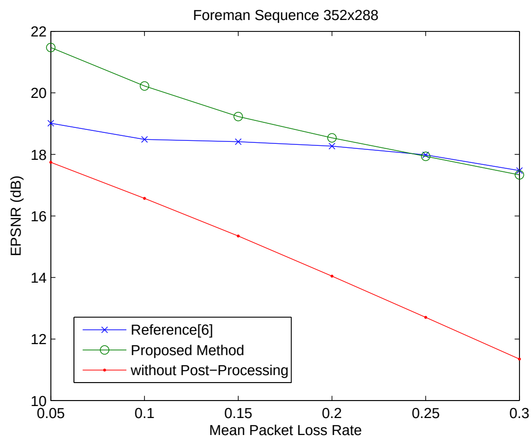


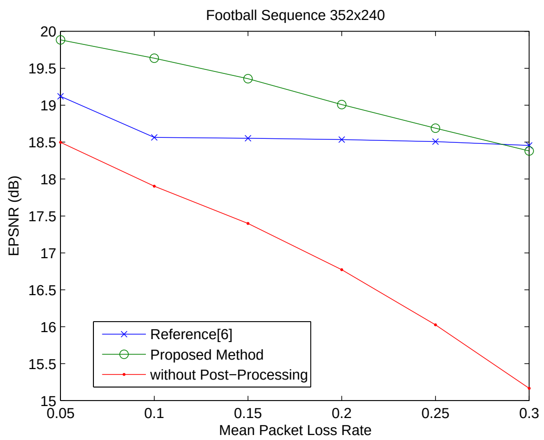
Figure 15: Performance comparison of various methods at bit rates below 70 Kbps. Our method's improvement over Reference [6] increases as the bit rate increases. The mean packet loss rate is 5% and the average burst length is 5.



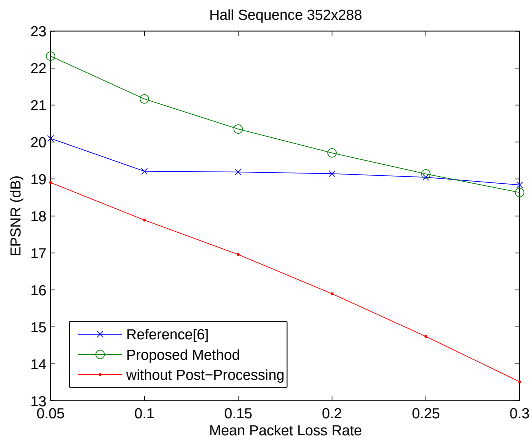
(a)



(b)

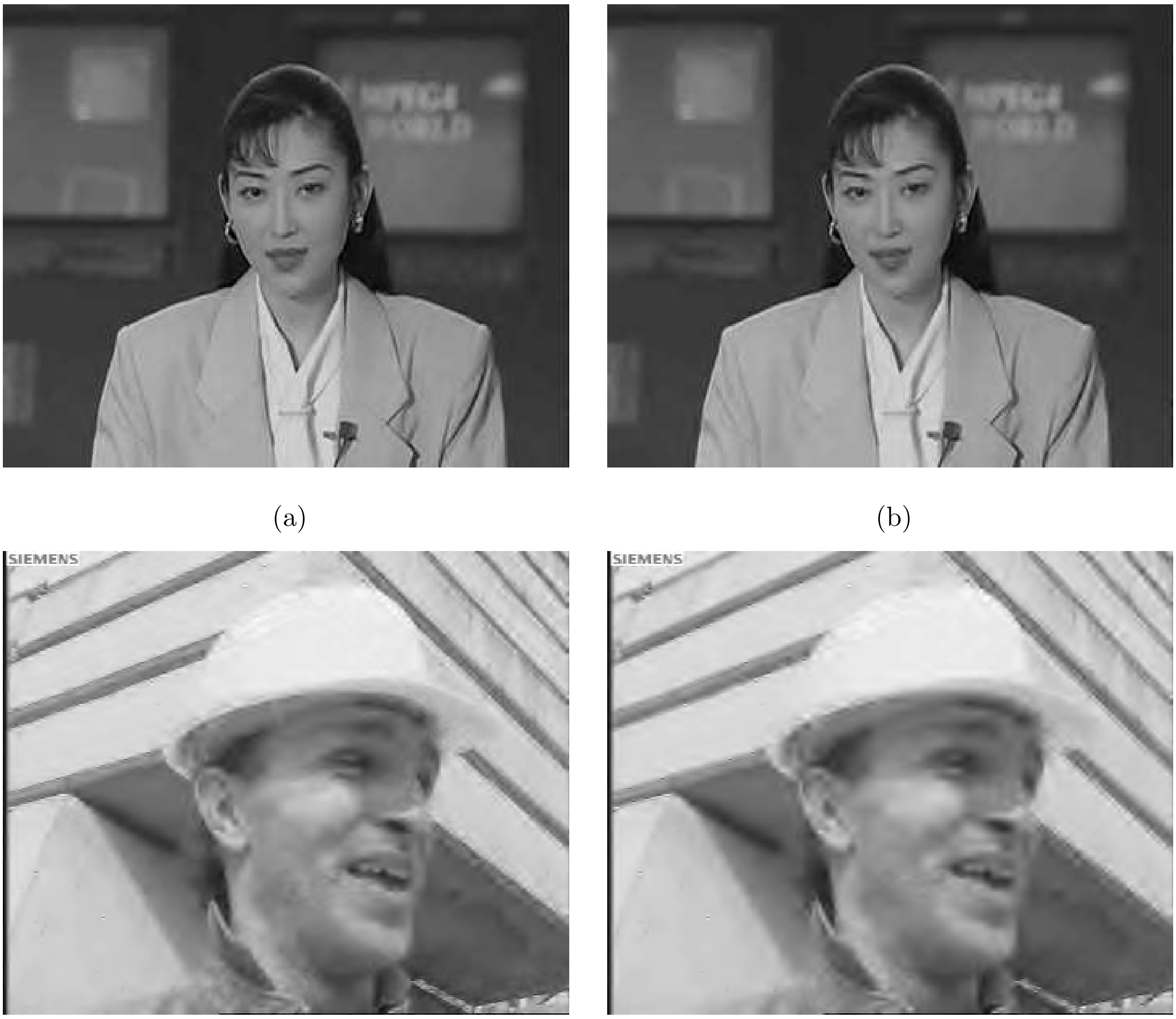


(c)



(d)

Figure 16: Performance comparison of various methods with different mean packet loss rates at 20 Kbps. Most of the bits are used for FEC when the packet loss rate is high; therefore, the performance of the top two curves is similar.



(a) (b)

(c) (d)

Figure 17: The format of all images is CIF. The channel parameters: packet loss rate is 20% and average burst length is 5. Top: Frame 6 of Akiyo sequence at rate 150k bps. (a) is the result of our method, while (b) is that of the method of [6]. Bottom: Frame 5 of Foremen sequence at rate 500k bps. (c) is our result, while (d) is that of [6].

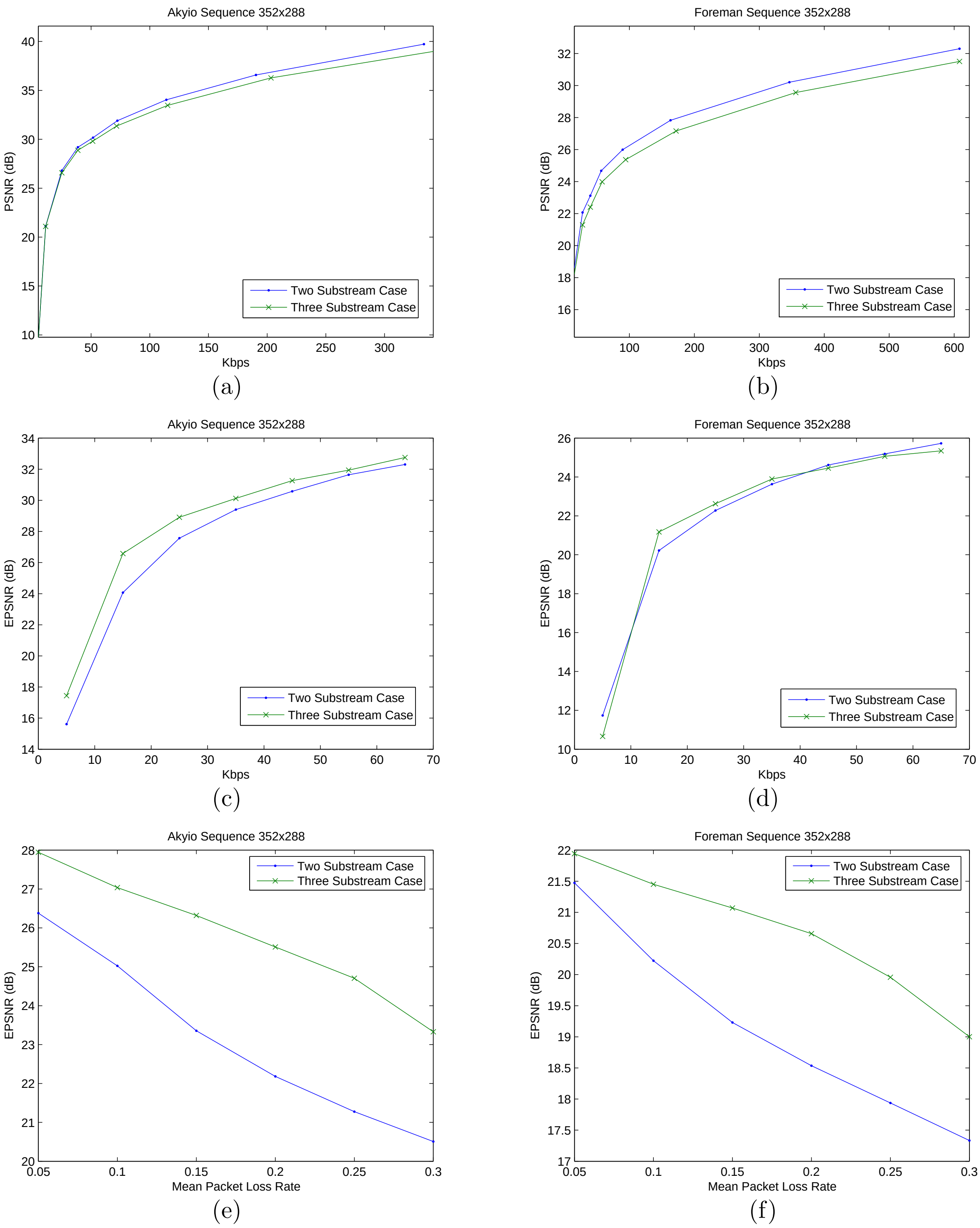


Figure 18: Comparison of the performance of different numbers of bitstreams for source coding and for our method. (a) and (b) show the source coding performance. The two-bitstream case is better. (c) and (d) are the performance curves of our method. The three-bitstream case is better. The channel parameters are 5% mean packet loss rate and the average burst length is 5. (e) and (f) are the performance curves versus the mean packet loss rate at 20 Kbps.

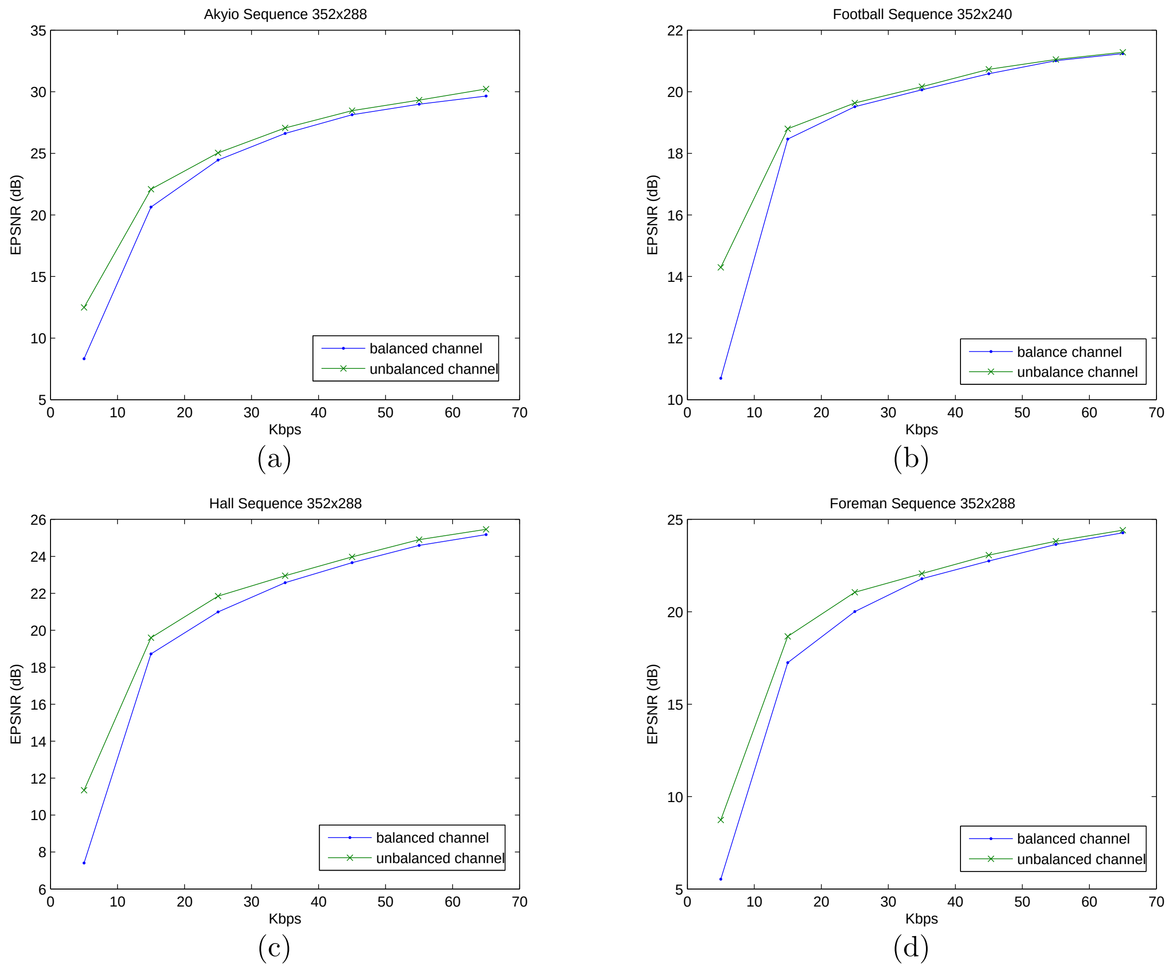


Figure 19: Performance comparison of our method for an unbalanced channel and a balanced channel. Two bitstreams are transmitted, and the average burst length is 5. For the balanced channel, the mean packet loss rate is 0.2 for each bitstream, while for the unbalanced channel, the mean packet loss rate for one bitstream is 0.1 and the other is 0.3.

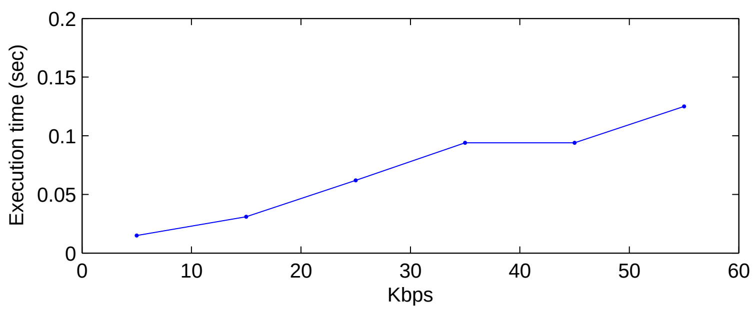


Figure 20: Computation time of the proposed method versus the number of bit rates.