Scientia Iranica D (2012) 19 (3), 782–794

Sharif University of Technology

Scientia Iranica

Transactions D: Computer Science & Engineering and Electrical Engineering

www.sciencedirect.com

Multiple description coding of scalable video stream based on unequal loss protection and optimal rate allocation

M.R. Ardestani *, **A.A. Beheshti Shirazi**

Image Processing Lab, School of Electrical Engineering, Iran University of Science and Technology, Tehran, P.O. Box: 165466551, Iran

Received 3 November 2010; revised 5 May 2011; accepted 27 June 2011

KEYWORDS

Unequal loss-protected packetization;
Multiple description coding;
Joint source-channel coding;
Scalable video coding;
Rate allocation.

Abstract Forward Error Correction (FEC) codes may be used to protect a video code stream against packet erasures or errors when passing through an error-prone network. To have maximum possible fidelity at the decoder side, an Unequal Loss Protection (ULP) approach should be used to packetize the scalable video code stream, so that the different parts of the scalable video stream are protected according to their importance. Unequal loss-protected packetization leads to segmentation of the scalable code stream, such that the source can be reconstructed with the maximum possible fidelity at the decoder side. In Ardestani et al. (2009) [1] we have found an analytical relation between the optimal sizes of any two consecutive segments. This idea yields an efficient (as efficient as the local search algorithm in Stankovic et al. (2002) [2]) low-complexity progressive solution for the segmentation problem. In this paper, we use a progressive approach for ULP packetization of a scalable video stream generated from a T + 2D encoder. In addition, an optimal rate allocation is used for optimal rate budget division between successive Groups Of Pictures (GOPs) of the video sequence. The experimental results demonstrate that the optimal rate budget allocation outperforms the conventional strategy of equal rate budget distribution up to 0.65 dB.

1. Introduction

Video propagation networks (e.g. Internet and peer-to-peer networks [3,4]) are usually composed of heterogeneous nodes with different and time-varying bandwidths and various playback capabilities, such as display resolution and frame rate. In addition, the bandwidth and packet loss rate of the network links are time-varying. Therefore, it is beneficial to use Scalable Video Coding (SVC) techniques [5,6] for post-encoding adaptation of the transmitted bit stream to time-varying channel conditions and user requirements. SVC is an efficient method to produce a prioritized video stream in which different parts have different levels of importance. One of the most efficient approaches of the SVC is T + 2D scalable video encoding in which the video content in each GOP is transformed into the spatio–temporal wavelet domain.

Multiple Description Coding (MDC) is an erasure-resilient redundant coding approach for robust real-time multimedia streaming over error-prone networks, such as the widely used wired or wireless Ethernet networks. MDC produces two or more independently decodable and complementary descriptions of the same source information [7], which are usually sent to the receiver over diverse paths.

Transmitting an embedded source bit stream (e.g. video scalable bit stream) through a packet erasure network requires an appropriate packetization (MDC) scheme, so that different parts of the data stream with different levels of importance are unequally protected against packet loss. It means a ULP approach should be used to packetize the scalable video code stream. FEC-based multiple description coding (MD-FEC) proposed by Puri and Ramchandran [8] is an efficient packetization scheme with the capability of ULP.

ULP is a joint source-channel coding problem, in which an embedded source code stream with a known Rate-Distortion (R-D) characteristic is adaptively segmented according to the packet loss Probability Distribution Function (PDF) of the channel. Then, each segment is protected with systematic error correction codes (e.g. Reed-Solomon code). Afterwards, a given number, *N*, of equally important packets of the fixed length, *L*, is generated, each carrying an equal contribution

* Corresponding author.
E-mail address: m_ardestani@iust.ac.ir (M.R. Ardestani).

Peer review under responsibility of Sharif University of Technology.

Production and hosting by Elsevier

1026-3098 © 2012 Sharif University of Technology. Production and hosting by Elsevier B.V. Open access under CC BY license.
doi:10.1016/j.scient.2011.06.010

of all protected segments. The segmentation process of the embedded bit stream should be carried out in such a way that maximum reconstruction fidelity is obtained at the decoder side.

Many researchers have devised different solutions for the ULP segmentation problem. In [8], Puri and Ramchandran proposed a Lagrange multiplier-based algorithm. In [9], Dumitrescu et al. proposed an $O(N^2L^2)$ algorithm that is close to optimal in general cases and optimal if the R-D fidelity function is convex and the packet loss probability function is monotonically decreasing. Mohr et al. [10] proposed a suboptimal search algorithm. In [2] Stankovic et al. proposed an $O(NL)$ local search algorithm that starts from a solution maximizing the expected number of received source bits and iteratively improves this solution.

In [1], we have analytically derived a relation between the optimal sizes of any two consecutive segments. This idea simply enables us to progressively approximate the optimal size of each segment from the previous one. In this way, each valid value for the size of the first (the most important) segment initiates a progressive process. To keep the complexity of the search procedure reasonable, we have found a short interval around the optimal value for the size of the first segment through an optimization analysis.

In this paper, we use our progressive method for the ULP packetization of the scalable video stream generated from a T + 2D encoder. To get the minimum possible value of average distortion throughout the video sequence, we solve an optimization problem for optimal rate allocation between the successive GOPs. The simulation results demonstrate that the optimal inter-GOP rate budget allocation outperforms the conventional strategy of equal rate budget distribution up to 0.65 dB. For the purpose of decreasing the encoding delay, we define the concept of a group of GOPs (GOGOPs) containing a number of GOPs. The optimization problem and the encoding procedure are carried out for the duration of each GOGOP. The experimental results show that one may easily trade off the encoding delay with the performance gain of the optimal rate allocation, relative to the equal inter-GOP rate budget assignment.

The rest of this paper is organized as follows. In Section 2, a T + 2D scalable video encoder is described. MD-FEC problem formulation is stated in Section 3, the progressive ULP method is discussed in Section 4 and the optimal rate allocation between successive GOPs is stated in Section 5. The simulation results are presented in Section 6 to compare the optimal rate allocation with the equal rate budget assignment. Finally, Section 7 concludes the paper.

2. T + 2D scalable video coding

Scalable video coding, based on Motion Compensated Temporal Filtering (MCTF) [11,12] and 2D-Discrete Wavelet Transform (2D-DWT), is a flexible architecture for embedded video streaming. The block diagram of the T + 2D encoder is represented in Figure 1. Each GOP of the input video sequence independently generates its own specific bit stream. After buffering G frames of a GOP, MCTF is applied, such that G temporal subband frames, including one low-pass (L) frame and $G - 1$ high-pass (H) frames, are obtained. MCTF provides a non-recursive embedded video coding structure including SNR, temporal and spatial scalability [13]. The application of MCTF for temporal decomposition can be seen as an iterative filtering operation on the original input sequence along the temporal direction [13] to efficiently remove the temporal redundancy between GOP frames. All temporal subband frames are individually affected by 2D-DWT and then encoded with a kind of bit-plane coding, such that an embedded bit stream is formed for each GOP.

Since high-pass (residual) temporal subband frames can have negative-valued pixels (from −255 to 255), it is necessary to shift and scale their samples. We have shifted them up 255 gray levels and then scaled down with a factor of 2, so that a meaningful 8-bit-depth image is made from H frames. At the decoder side, the inverse of the above operations should be carried out.

Although the scalable video encoder represented in Figure 1 is of a general form, it is important to perform MCTF, 2D-DWT and bit-plane coding in an efficient standard-compatible fashion. In this paper, MCTF has been performed using the JSVM 9.17 reference software (MCTFPreProcessorStatic tool) [14] in order to benefit from all exclusive features of the H.264/AVC standard video coding, such as variable-sized block, adaptive mode decision, and deblocking filter. For 2D-DWT and bit-plane coding, the Jasper 1.701.0 reference software [15] has been used, benefiting from the well-known scalability properties and embedded entropy coding of the JPEG2000 standard coding.

2.1. Rate-distortion curve of GOP coded bit stream

After joint temporal and spatial wavelet decomposition, there are C independent Code Blocks (CBs) belonging to all temporal and spatial subbands of all YUV components within the current GOP. Each CB has an embedded bit stream. The total GOP distortion is the weighted sum of the distortions of all CBs belonging to the chrominance and luminance components. Accurate estimators have been introduced in [16] to approximate the distortion caused by the successive bit-plane coding of the transform coefficients. These accurate estimators are based on precise approximation of the coefficients' probability distribution within the quantization intervals [16]. Since JPEG2000 is employed here, we may use auxiliary information about the rate of each CB belonging to each layer, which is recorded in the JPEG2000 packet header [17,18]. In addition, the information about the distortion and R-D slope of each CB is available during the encoding process and can be transmitted, along with the bit stream, with a proprietary syntax. This auxiliary information can be used for R-D curve extraction of each CB.

Assuming the spatial and temporal filters are orthogonal, the GOP distortion in the words of the Sum of the Squared Differences (SSD) between all pixels of the original and reconstructed GOP frames can be expressed as [18,19]:

$$D = \sum_{i=1}^{C} \omega_t^i \omega_s^i \omega_{sc}^i \omega_{YUV}^i d_i, \tag{1}$$

where d_i is the distortion of the ith CB, when truncated at rate r_i. Distortion weights, ω_t^i and ω_s^i, denote temporal and spatial weights, defined as the magnitude of the spatial and temporal wavelet synthesis filter coefficients [17–19]. This means that the reconstruction distortions of different CBs belonging to different spatio–temporal subbands contribute the total distortion with different weights. Since JPEG2000 is applied to each temporal subband frame, the spatial weights are the same as those used in JPEG2000. Therefore, the spatial weights are automatically applied by means of Jasper

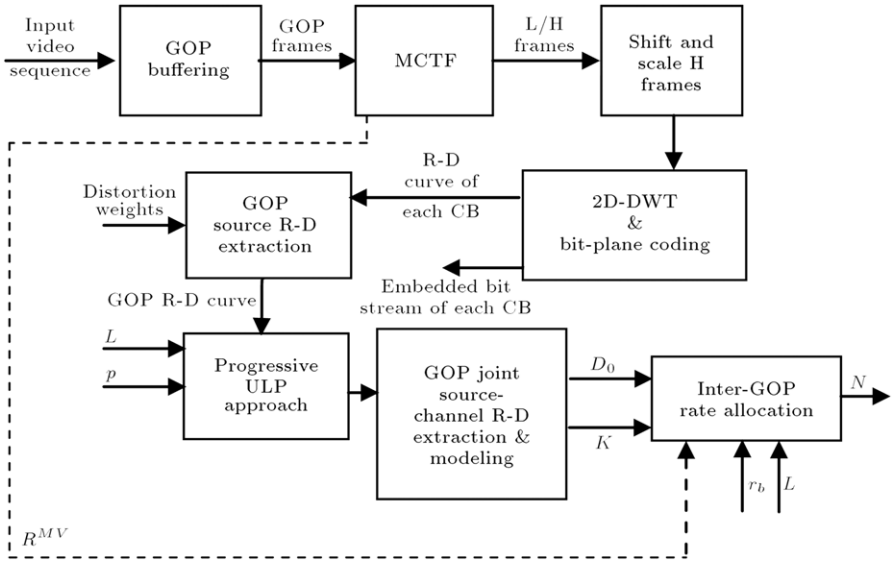

Figure 1: The block diagram of T + 2D scalable video encoder together with the inter-GOP rate allocation.

1.701.0 reference software. To obtain the temporal weights, we have performed experiments using MCTFPreProcessorStatic and PSNRStatic tools of JSVM 9.17 reference software [14]. To this aim, we separately applied some definite distortions to each individual temporal subband frame and then measured the total distortion (by means of reconstructing the video using the MCTFPreProcessorStatic tool and comparing it with the undistorted video using the PSNRStatic tool), in order to calculate the distortion weight corresponding to that temporal subband frame. Note that, each time, only one of the temporal subband frames is distorted and the others are unaffected. This procedure is separately carried out for all temporal subband frames to calculate their corresponding temporal weights. We have performed these experiments for all GOPs of several test sequences and then averaged over the results. ω_{sc}^{i} is the weight induced from scaling the samples of high-pass temporal subband frames before JPEG2000 is applied. ω_{sc}^{i} equals 4 or 1, if the ith CB belongs to a high-pass or low-pass frame, respectively. This is because we have shifted up samples of the high-pass temporal subband frames 255 gray levels and then scaled down with a factor of 2, as discussed earlier. ω_{YUV}^{i} equals 1 (θ_U) (θ_V), if the ith CB belongs to the Y (U) (V) video component. These component-related distortion weights allow us to arbitrarily differentiate between the distortions of the different components, depending on their importance. We have chosen $\theta_U = \theta_V = 1$, so that the distortions of all components have the same importance in the total distortion.

Since each CB is coded using bit-plane coding, any CB has an embedded bit stream that can be truncated at several points [18]. Let us consider the R-D curve of each CB. We have implicitly assumed that each CB has a convex hull, which is true about Jasper 1.701.0 reference software (rate allocation procedure of Jasper software automatically extracts all truncation points of each CB and then preserves only those located on a convex hull). Figure 2 schematically depicts the R-D curve of each CB contained in a GOP and the resultant R-D curve of the GOP. Every truncation point of the i'th CB embedded bit stream corresponds to a triplet, consisting of reconstruction distortion, d_i^j, truncated rate, r_i^j and R-D slope, λ_j. As shown in Figure 2, the R-D curve of each GOP is formed from a number of discrete truncation points, each corresponding to an R-D slope,

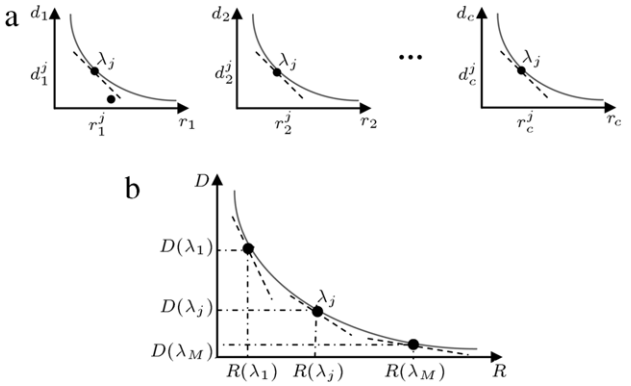

Figure 2: The R–D curve of (a) each CB and (b) the total GOP.

λ_j. In fact, there is a one-to-one correspondence between total rates and the R-D slopes of each GOP R-D curve. The total rate of each GOP at R-D slope, λ, is equal to the sum of the rates of its CBs, each truncated at R-D slope, λ [18, Proposition 1]. This means that, for each total GOP rate $R(\lambda)$, it is enough to find the minimum possible absolute value of the R-D slope, λ, such that the sum of the rates of all the CBs (each truncated at R-D slope λ) is not greater than R. Consequently, a sufficient number of R-D slope values, λ, should be engaged to find all possible truncation points of each GOP. For each λ value, it is enough to search for the truncation point of every CB with the highest rate, whose absolute value of R-D slope is not smaller than λ. Afterwards, the sum of truncated rates (distortions) of all CBs is equal to the rate (distortion) of a GOP possible truncation point corresponding to R-D slope λ. All possible truncation points of a GOP form its R-D curve.

3. MD-FEC problem statement

In this section, we state the composite problem of segmentation, protection and packetization of a scalable video code stream using the ULP strategy. Each GOP of a video sequence is independently packetized. The scalable code stream of each GOP can be generated using a T + 2D scalable video encoder, as mentioned in Section 2. In what follows, we will use the word

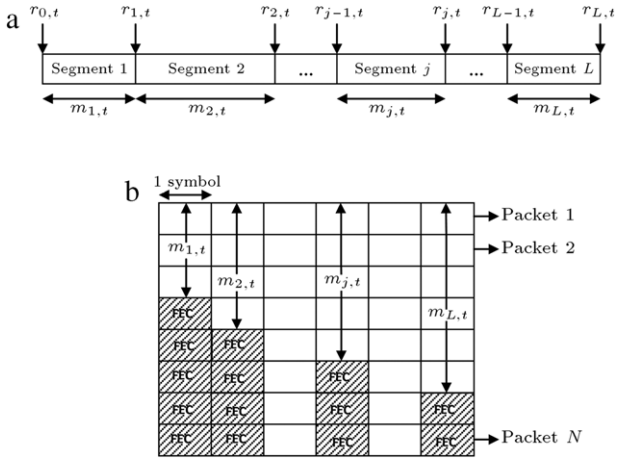

Figure 3: Segmentation of each GOP t embedded bit stream for ULP packetization. (a) Segmentation; and (b) FEC protection and packetization.

"packet" as synonymous with the word "description" in terms of the terminology of MDC. Our notations are essentially the same as [1,2,9].

We want to obtain N_t equally important packets (descriptions) of L symbols (e.g. bytes) from the embedded code stream of the tth GOP. First, we partition the embedded code stream of the tth GOP into L segments with non-decreasing sizes of $m_1 \leq m_2 \leq \cdots \leq m_L \leq N_t$ symbols, as shown in Figure 3a. Then, each segment, j, is protected with $f_j = N_t - m_j$ parity symbols, using the (N_t, m_j) systematic Reed-Solomon FEC code, as depicted in Figure 3b. Therefore, each segment is extended to contain N_t symbols, such that the reception of each m_j symbol (including data or FEC symbols) of the jth segment guarantees perfect reconstruction of that segment (i.e., the jth segment can be perfectly reconstructed if, and only if, less than f_j symbols out of its N_t symbols are lost at the decoder side). Afterwards, the ith symbols of all protected segments are grouped to form the ith description (packet), $i = 1, .2, \ldots, N_t$. According to the capabilities of FEC codes, if n descriptions are lost at the decoder side, such that $f_j \geq n > f_{j+1}$, then, the first j segments of the source code stream can be reconstructed.

Let p_n stand for the probability of losing exactly n packets (out of N_t) and $c(k) = \sum_{n=0}^{k} p_n$, $k = 0, \ldots, N_t$. Then, $c(f_j)$ is the probability that the decoder correctly reconstructs the source code stream up to the j'th segment. Let $d_t(r)$ denote the R-D function of the scalable code stream of the tth GOP and let X stand for a random variable, whose value is the number of lost packets. Therefore, $\{p_n\}_{n=0}^{N_t}$ and $\{c_k\}_{k=0}^{N_t}$ are the Probability Mass Function (PMF) and Cumulative Distribution Function (CDF) of the random variable, X, respectively. For example, if we assume a binomial distribution for the packet loss of the channel, then;

$$p_n = \binom{N_t}{n} p^n (1-p)^{N_t - n},$$

$$c(k) = \sum_{n=0}^{k} \binom{N_t}{n} p^n (1-p)^{N_t - n}, \tag{2}$$

where p is the packet loss probability of the propagation channel.

At the decoder side, the scalable code stream of the tth GOP may be reconstructed up to the first, second, or Lth segment, according to the number of received packets. Therefore, the reconstructed bit rate may be one of the values of r_j, for $j = 0, 1, \ldots, L$, in which:

$$r_0 = 0,$$

$$r_j = \sum_{k=1}^{j} m_k = jN - \sum_{k=1}^{j} f_k, \quad \text{for } = 1, \ldots, L. \tag{3}$$

Accordingly, the reconstruction distortion of the code stream may be one of the values of $d_t(r_j)$, for $j = 0, 1, \ldots, L$. The probability of the code stream to be reconstructed at the bit rate $r_j, j = 0, 1, \ldots, L$, is as follows:

$$P_0(\boldsymbol{f}) = P(X > f_1) = 1 - c(f_1),$$
$$P_j(\boldsymbol{f}) = P(f_{j+1} < X \leq f_j) = c(f_j) - c(f_{j+1}),$$
$$\text{for } j = 1, \ldots, L-1,$$
$$P_L(\boldsymbol{f}) = P(X \leq f_L) = c(f_L), \tag{4}$$

where $\boldsymbol{f} = (f_1, \ldots, f_L)$ is the parity vector. The optimal MD-FEC problem consists of determining the optimal sizes of the segments. In the other words, it is desired to find a parity vector, $\boldsymbol{f} = (f_1, \ldots, f_L)$, that minimizes the expected distortion:

$$D_t^{avg}(\boldsymbol{f}) = \sum_{j=0}^{L} P_j(\boldsymbol{f})\, d_t(r_j). \tag{5}$$

Although $f_1, \ldots, f_L, r_1, \ldots, r_L, c(\cdot)$, and $d_t(\cdot)$ are discrete-valued quantities, we will virtually treat them as continuous-valued quantities, so that a meaningful derivation can be normally applied. In this manner, $c'(\cdot)$ and $d_t'(\cdot)$ denote the derivatives of $c(\cdot)$ and $d_t(\cdot)$, respectively, with respect to their input arguments. $c'(f)$ can be roughly estimated by p_f.

4. Progressive approach for ULP packetization

In this section, we discuss the progressive solution [1] for the segmentation problem stated in Section 3. To solve this problem in an optimal manner, it is necessary to search among all candidate parity vectors, $\boldsymbol{f} = (f_1, \ldots, f_L)$, with successively non-increasing elements ($N_t \geq f_1 \geq f_2 \geq \cdots \geq f_L$) and then select the one that minimizes the expected distortion in (5). This full search method is not feasible for real-time applications, especially for large values of N_t and L. In this section, an optimality condition is derived which provides a rather simple relation between any two consecutive elements of the optimal $\boldsymbol{f}$ vector. This optimality condition yields a low complexity progressive approach.

Let $D_i(x), i = 1, \ldots, L-1$, denote the expected distortion of the received video bit stream when two consecutive elements, f_i and f_{i+1}, of the optimal solution, $\boldsymbol{f}$, are changed by a minute positive amount of x. That is:

$$D_i(x) = D_t^{avg}(\{\ldots, f_i + x, f_{i+1} - x, \ldots\}). \tag{6}$$

$D_i(x)$ is defined, such that the overall parity budget remains unchanged and only a limited number of parameters of $\boldsymbol{f} = (f_1, \ldots, f_L)$ and $\boldsymbol{r} = (r_0, r_1, \ldots, r_L)$ are affected. In fact, f_i and m_{i+1} are increased by x, and f_{i+1}, m_i and r_i are decreased by the same amount of x. The other parameters remain unchanged. The optimality condition of $\boldsymbol{f}$ imposes:

$$I_i(\boldsymbol{f}) \triangleq \left.\frac{\partial D_i(x)}{\partial x}\right|_{x=0} = 0, \tag{7}$$

for $i = 1, \ldots, L-1$. Considering the changing parameters in (7) (i.e. f_i, f_{i+1} and r_i), we have: Eq. (8) in Box I.

$$I_i(\boldsymbol{f}) = \left.\frac{\partial\left[P_{i-1}(\boldsymbol{f})\cdot d_t(r_{i-1}) + P_i(\boldsymbol{f})\cdot d_t(r_i) + P_{i+1}(\boldsymbol{f})\cdot d_t(r_{i+1})\right]}{\partial x}\right|_{x=0}$$
$$= \left.\frac{\partial\left[-c(f_i)\cdot d_t(r_{i-1}) + (c(f_i) - c(f_{i+1}))\cdot d_t(r_i) + c(f_{i+1})\cdot d_t(r_{i+1})\right]}{\partial x}\right|_{x=0}. \quad (8)$$

Box I:

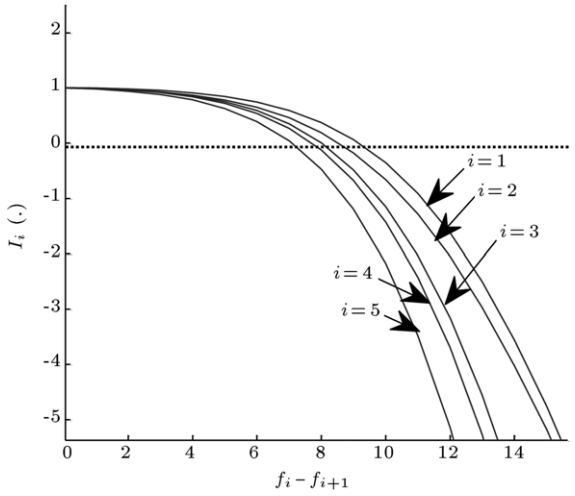

Figure 4: The typical profile of $I_i(f)$ for some successive values of i, the horizontal axis is the difference between f_{i+1} and f_i. In this figure, $N = 200$, $L = 47$ and $p = 0.2$. The similar profiles will be obtained for other values of these parameters.

As mentioned above, f_i (f_{i+1} and r_i) is increased (are decreased) by x. Hence, $\frac{\partial f_i}{\partial x} = 1$ and $\frac{\partial f_{i+1}}{\partial x} = \frac{\partial r_i}{\partial x} = -1$. Finally, the optimality condition is derived from Eq. (8) as follows:

$$I_i(\boldsymbol{f}) = -c'(f_i).d_t(r_{i-1}) + \left[c'(f_i) + c'(f_{i+1})\right].d_t(r_i) - \left[c(f_i) - c(f_{i+1})\right]\cdot d_t{}'(r_i) - c'(f_{i+1})\cdot d_t(r_{i+1}) = 0, \quad (9)$$

for $i = 1, \ldots, L-1$. Eq. (9) is dependent on only a limited number of parameters including $f_i, f_{i+1}, r_{i-1}, r_i$ and r_{i+1}. However, the parameters r_k are calculated based on f_j values, $j = 1, \ldots, k$. It is noteworthy that the value of f_{i+1} can be calculated by satisfying Eq. (9), when all previous values of $f_j, j = 1, \ldots, i$, are available. Hence, we can calculate f_i values starting from $i = 1$ toward $i = L$, step-by-step and "progressively". In fact, Eq. (9) provides a relation between any two consecutive elements of the "optimal" parity vector. It means, if we have f_1, then we can find f_2. When f_2 is found, f_3 will be subsequently determined and so on. Therefore, if we have the first element of the optimal solution, we can progressively find the others.

Although the behavior of $I_i(\boldsymbol{f})$ functions is not analytically studied in this research, experimental results presented in Figure 4 show that $I_i(\boldsymbol{f})$ is a well-behaved function, with respect to the f_{i+1} value when f_i and r_i values are known. It is noteworthy that the values of f_i and r_i are previously known at each step, and we aim to find the value of f_{i+1}, which makes $I_i(\cdot)$ zero. Therefore, finding the root of $I_i(\cdot)$ corresponds to finding the f_{i+1} value, given the known values of f_i and r_i. In Figure 4, showing $I_i(\boldsymbol{f})$ for some different values of i, the horizontal axis is the difference between f_{i+1} and f_i and the vertical axis is the value of I_i. It is seen that $I_i(\boldsymbol{f})$ starts from a positive value and goes to negative values after crossing $I_i = 0$. We have observed

```
Compute f*_1 using (12)
r_0 = 0;
for f_1 = f*_1 - h/2 to f*_1 + h/2{
  r_1 = N_t - f_1;
  for i = 1 to L - 1{
    f_{i+1} = f_i;
    r_{i+1} = r_i + N_t - f_{i+1};
    Compute I_i using (9)
    while (I_i ≠ 0 and f_{i+1} >= 0){
      f_{i+1} = f_{i+1} - 1;
      r_{i+1} = r_{i+1} + 1;
      Compute I_i using (9)
    }
  }
Compute D_avg using (5)
Store D_avg along with f = (f_1, .., f_L)
}
Select f with the least D_avg
```

Figure 5: The pseudo code of the proposed method to find the optimum $\boldsymbol{f}$. The parameter h represents the length of search interval of f_1 around the f_1^*, usually $h = 10$ is appropriate.

that zero crossing of $I_i(\boldsymbol{f})$ functions usually occurs for a little difference between f_{i+1} and f_i, i.e. starting from $f_i = f_{i+1}$ and going on, we will soon reach the value of f_{i+1}, which is the root of $I_i(\boldsymbol{f})$ and is located at a short distance from the starting point (i.e. f_i). Hence, to find the root of each $I_i(\boldsymbol{f})$ in Eq. (9), we use a simple search method, in which f_{i+1} starts from f_i and, after a small number of increments, the root will be met. Another property of $I_i(\boldsymbol{f})$ functions is that the difference between f_i and f_{i+1} decreases with i. This can help to limit the cost of root finding.

The value of f_1 cannot be found using Eq. (9). So, in general cases, for different values of f_1, the corresponding $\boldsymbol{f}$ should be found by the step-by-step progressive process, and D_{avg} should be calculated using Eq. (5). Then, D_{avg} and the corresponding $\boldsymbol{f}$, suggesting minimum D_{avg}, are selected as the best solution. However, an approximation method is used to reduce the number of iterations for finding the optimal value of f_1, which is discussed in the next subsection. The proposed optimization process can be summarized as in Figure 5.

4.1. Approximation of the optimal f_1 value

So far, we have derived a progressive relation between the sizes of any two consecutive segments. It means that, if we have f_1, then we can obtain f_2, then f_3 and etc.. In this way, each possible value of f_1 (from 0 to $N_t - 1$) can initiate a progressive procedure, which results in a candidate solution of $\boldsymbol{f}$. Among all possible resultant solutions, we should select the one that minimizes the expected distortion in Eq. (5). To keep the computational complexity of the proposed progressive method low, it is necessary to strictly keep the search interval of f_1 as short as possible. In this subsection, we find a short search interval around the optimal value of f_1.

Making zero the derivation of the expected distortion in (5), with respect to f_1, we obtain:

$$c'(f_1)\cdot[d_t(N_t-f_1)-d_t(0)]=\sum_{j=1}^{L}P_j(\boldsymbol{f})\,d_t'(r_j). \tag{10}$$

Using Eq. (10), we can find the optimal value for f_1, provided that all other f_j values, $j=2,\ldots,L$, are known. To obtain a good approximation of the optimal f_1 value from Eq. (10), we set $f_2=f_3=\cdots=f_L=f_r$, where f_r is the rate-optimal solution [2], which is the solution maximizing the expected number of received source bits and which is simply expressed as [2]:

$$f_r=\arg\max_{i=0,\ldots,N_t-1}(N_t-i)\cdot\sum_{n=0}^{i}p_n. \tag{11}$$

In this way, $P_2=P_3=\cdots=P_{L-1}=0$. Usually, we can consider $d_t'(r_L)\cong 0$ because r_L is sufficiently large. Therefore, Eq. (10) is reduced to:

$$c'(f_1)\cdot[d_t(N_t-f_1)-d_t(0)]\cong[c(f_1)-c(f_r)]\cdot d_t'(N_t-f_1). \tag{12}$$

Let the solution of Eq. (12) for f_1 be f_1^*, which can be found with a simple search. Afterwards, we can say the optimal f_1 value is located at a short interval around f_1^*. To validate this claim, several simulations have been carried out for various selections of parameter set including N_t, L, $c(\cdot)$ and $d_t(\cdot)$. Simulation results show that $\left[f_1^*-h/2, f_1^*+h/2\right]$ is an appropriate interval for nearly all natural cases in which h is in the order of 10 (albeit, h is dependent on N_t and L values to some extent, but not severely).

If the search interval for f_1 consists of h points and the root finding search of each $I_i(\boldsymbol{f})$ runs for an average of b times (successive increments of starting f_i) until the solution of Eq. (9) is found, then we can roughly say that the complexity of the progressive method is $O(hbL)$. This means that the complexity of the proposed progressive method does not essentially depend on N_t and this is a great advantage, especially for large values of N_t. As mentioned earlier, the local search algorithm has a complexity of $O(N_tL)$ [2], which is linearly dependent on N_t.

5. Optimal inter-GOP rate allocation

In Section 4, we described the progressive approach for the segmentation and packetization of the embedded bit stream of each GOP. For each GOP, the progressive approach gets some input parameters and gives the parity vector (the number of parity symbols for protection of each segment) as output. For each GOP $t=1,2,\ldots$, the progressive approach requires the following parameters as input:

- The total rate budget (in symbols) allocated to GOP t: R_t,
- The required rate (in symbols) for encoding the motion information of GOP t: R_t^{MV},
- The packet size (in symbols) : L,
- The number of packets allocated for joint source-channel coding of the texture of GOP t: $N_t=\frac{R_t-R_t^{MV}}{L}$,
- The source R-D characteristic of GOP t: $d_t(r)$,
- The channel profile of packet loss: $\{p_n\}_{n=0}^{N_t}$ and $\{c(k)\}_{k=0}^{N_t}$.

When all the above parameters of each GOP t are known, then the progressive approach calculates the suboptimal parity vector, $\boldsymbol{f}=(f_1,\ldots,f_L)$. Subsequently, the average reconstruction distortion for the scalable code stream of GOP t, D_t^{avg}, can be easily calculated from Eq. (5). One may change the total rate budget (required for source, channel parity, and motion information) of each GOP t, R_t, and then calculate the corresponding value of D_t^{avg}. Since the required rate, R_t^{MV}, for motion information coding and the packet size L are fixed, changing the value of R_t translates to changing the number, N_t, of packets allocated for joint source-channel coding of the texture of GOP t. Therefore, one can consider two R-D curves for each GOP t:

- The pure source R-D curve of GOP t: $d_t(r)$,
- The joint source-channel R-D curve of GOP: $D_t^{avg}(N_t)$.

The first R-D curve depends only upon the video source content in the duration of that GOP. However, the second R-D characteristic depends not only on source content, but also on the channel packet loss profile and the performance of the progressive approach in the segmentation and packetization of the embedded code stream. In this section, we derive a closed-form relation for the optimal value of N_t for each GOP $t=1,2,\ldots$, such that the overall average distortion over the whole video sequence is minimized and a predetermined average bit rate is obtained. In fact, we want to divide a predetermined value of the bit rate budget between the different GOPs of a video sequence. When the share of each GOP from the total bit rate budget is determined through an inter-GOP rate allocation procedure (i.e. N_t is known for GOP $t=1,2,\ldots$), then the progressive segmentation approach determines the portion of the source and channel parity for each segment of each GOP through an intra-GOP joint source-channel rate partition procedure.

5.1. Problem statement

It is desired to find the optimal number of allocated packets for each GOP t, such that the overall average distortion over the whole sequence D_{avg} is minimized, and a fixed average bit rate, r_b, (bits per second) is obtained. It is clear that the total symbol budget, S, and overall average distortion, D_{avg}, over the whole sequence can be written as follows:

$$S=\frac{r_b}{q}\cdot\frac{M\cdot T}{r_f},$$

$$D_{avg}=\frac{1}{M}\sum_{t=1}^{M}D_t^{avg}(N_t), \tag{13}$$

where:

M: the total number of GOPs contained in the video sequence,

T: the GOP size (i.e. the number of frames contained in each GOP),

q: the number of bits per symbol,

r_f: the frame rate (frames per second).

The inter-GOP rate allocation problem can be formulated as follows:

$$\left\{N_t^*\right\}_{t=1}^{M}=\arg\min_{N_t}D_{avg},$$

$$\text{s.t.}\sum_{t=1}^{M}R_t=\sum_{t=1}^{M}\left(N_t\cdot L+R_t^{MV}\right)=S.$$

With a little simplification, we have:

$$\left\{N_t^*\right\}_{t=1}^{M} = \arg\min_{N_t} \sum_{t=1}^{M} D_t^{avg}\left(N_t\right),$$

$$\text{s.t.} \sum_{t=1}^{M} N_t = \frac{\left(S - \sum_{t=1}^{M} R_t^{MV}\right)}{L} \triangleq P. \tag{14}$$

The inter-GOP rate allocation problem in Eq. (14) can be solved using the Lagrange method. The Lagrangian cost function may be written as:

$$J\left(N_1, \ldots, N_M, \lambda\right) = \sum_{t=1}^{M} D_t^{avg}\left(N_t\right) + \lambda\left(\sum_{t=1}^{M} N_t - P\right), \tag{15}$$

in which λ is the Lagrangian multiplier. To obtain the optimal solution, we should apply $\frac{\partial J}{\partial N_t} = 0$, for $t = 1, \ldots, M$. Therefore, we have the following optimality condition:

$$\frac{\partial D_t^{avg}}{\partial N_t} = -\lambda, \quad \text{for } = 1, \ldots, M. \tag{16}$$

To obtain a closed-form solution for $\left\{N_t^*\right\}_{t=1}^{M}$, we may fit an appropriate model to $D_t^{avg}\left(N_t\right)$ for $t = 1, \ldots, M$. In the next subsection, we will consider a hyperbolic model for this purpose.

5.2. Problem solving

In [20], we have proposed a hyperbolic R-D model for modeling R-D characteristics of each GOP scalable bit stream. This model is empirical and not derived analytically. The experiments in [20] show that the hyperbolic model has an acceptable performance for nearly all practical working ranges of the bit rate. In this subsection, we use the same hyperbolic model for the joint source-channel R-D curve of each GOP t, $D_t^{avg}\left(N_t\right)$, i.e.:

$$D_t^{avg}\left(N_t\right) = D_{0,t}\left(N_t L\right)^{-K_t}. \tag{17}$$

This simple and efficient model has only two parameters, i.e. $D_{0,t}$ and K_t. To obtain these parameters, we should first measure a sufficient number of points on the joint source-channel R-D curve, $D_t^{avg}\left(N_t\right)$, for each GOP $t = 1, \ldots, M$. In fact, we should run the progressive approach for a large number of N_t values (with a fixed value of L) and then measure the corresponding values of D_t^{avg}.

Let us use $\left\{\left(N_t^i, D_t^{avg,i}\right)\right\}_{i=1}^{W}$ to denote W measured points on the joint source-channel R-D curve of the GOP t. We define the optimal values of the model parameters as those minimizing the Mean Square Error (MSE) between the logarithm of the measured and modeled average distortion values. According to the results obtained in [20], the optimal values of $D_{0,t}$ and K_t for the hyperbolic model are derived as follows:

$$K_t = \frac{\sum_{i=1}^{W} n_t^i \cdot \sum_{i=1}^{W} s_t^i - W \sum_{i=1}^{W}\left(s_t^i \cdot n_t^i\right)}{\left(\sum_{i=1}^{W} n_t^i\right)^2 - W \sum_{i=1}^{W}\left(n_t^i\right)^2}, \tag{18}$$

$$\beta_t = \frac{1}{W}\sum_{i=1}^{W}\left(s_t^i - K_t \cdot n_t^i\right), \tag{19}$$

$$D_{0,t} = e^{-\beta_t}, \tag{20}$$

Table 1: The hyperbolic model parameters and modeling error for joint-source channel R–D curve of various GOPs of *Foreman* sequence (QCIF, 15 fps) with 16 frames per GOP, $p = 0.05$, $L = 47$ Bytes.

GOP no.	Destination D_0	Destination K	Model error, dB
1	1,293,050,244	0.509	0.25
2	857,470,736	0.472	0.31
3	1,383,019,709	0.518	0.25
4	2,406,661,738	0.579	0.23
5	2,172,929,156	0.544	0.25
6	1,090,048,970	0.497	0.32
7	5,135,430,246	0.642	0.24
8	3,985,402,152	0.617	0.18
9	1,923,111,533	0.550	0.23
10	2,896,067,775	0.581	0.26
11	2,372,731,053	0.565	0.24
12	7,026,446,431	0.646	0.25
13	3,294,201,795	0.605	0.23
14	961,052,392	0.508	0.34
15	6,810,166,876	0.654	0.23
16	21,360,706,706	0.749	0.30
17	7,247,911,640	0.656	0.19
18	21,876,707,076	0.750	0.21
19	22,574,308,663	0.787	0.11
20	2,400,189,501	0.564	0.27
21	1,131,316,259	0.453	0.28
22	1,440,267,105	0.483	0.31
23	1,313,696,939	0.486	0.33
24	1,616,642,033	0.519	0.40

Table 2: The hyperbolic model parameters and modeling error for joint-source channel R–D curve of various GOPs of *Akiyo* sequence (QCIF, 15 fps) with 16 frames per GOP, $p = 0.05$, $L = 47$ Bytes.

GOP no.	Destination D_0	Destination K	Model error, dB
1	2,190,710,697	0.750	0.16
2	5,006,832,790	0.799	0.30
3	4,269,025,146	0.792	0.23
4	3,738,694,334	0.793	0.16
5	4,926,953,231	0.796	0.34
6	3,510,816,266	0.775	0.22
7	3,139,910,504	0.774	0.13
8	3,277,941,064	0.785	0.17
9	4,811,189,246	0.805	0.26
10	4,095,703,566	0.794	0.27
11	3,879,842,420	0.802	0.15
12	3,272,009,695	0.786	0.15
13	2,848,615,049	0.772	0.15
14	4,701,795,601	0.792	0.32
15	4,151,167,486	0.792	0.22
16	4,712,397,221	0.802	0.24
17	4,165,792,200	0.801	0.20
18	4,255,727,287	0.789	0.26

where $n_t^i = \ln N_t^i$ and $s_t^i = -\ln D_t^{avg,i}$. Tables 1–4 show the hyperbolic model parameters and modeling errors for various GOPs of four different video sequences; *Foreman*, *Akiyo*, *Stefan* and *Mobile*. In these tables, a binomial distribution with $p = 0.05$ is assumed for the packet loss event of the channel. The size of each packet is considered $L = 47$ Bytes. These tables demonstrate that the hyperbolic model has a low error model and fits well to the joint source-channel R-D characteristics of the above mentioned video sequences.

Using the hyperbolic model in Eq. (17), one can rewrite the optimality conditions in Eq. (16) as follows:

$$\frac{D_{0,t} \cdot K_t}{N_t^{K_t+1} \cdot L^{K_t}} = \lambda. \tag{21}$$

In the other words, the optimal values of the number of allocated packets for each GOP t are derived as:

Table 3: The hyperbolic model parameters and modeling error for joint-source channel R–D curve of various GOPs of *Stefan* sequence (CIF, 30 fps) with 16 frames per GOP, $p = 0.05$, $L = 47$ Bytes.

GOP no.	Destination D_0	Destination K	Model error, dB
1	22,216,914,269	0.445	0.36
2	35,999,792,814	0.505	0.50
3	21,562,980,218	0.460	0.42
4	15,531,211,381	0.463	0.45
5	12,971,508,875	0.415	0.39
6	18,300,099,461	0.435	0.36
7	41,110,747,128	0.498	0.47
8	73,056,346,027	0.598	0.55
9	12,972,237,313	0.421	0.38
10	10,559,801,841	0.399	0.39
11	24,440,096,552	0.462	0.43
12	93,633,400,700	0.580	0.41
13	36,244,121,490	0.526	0.47
14	39,124,187,846	0.532	0.46
15	87,275,607,351	0.569	0.44
16	174,175,725,509	0.599	0.39
17	77,499,397,266	0.523	0.28
18	34,335,804,370	0.420	0.33

Table 4: The hyperbolic model parameters and modeling error for joint-source channel R–D curve of various GOPs of *Mobile* sequence (CIF, 30 fps) with 16 frames per GOP, $p = 0.05$, $L = 47$ Bytes.

GOP no.	Destination D_0	Destination K	Model error, dB
1	8,175,364,705	0.358	0.25
2	8,317,037,117	0.362	0.26
3	9,066,159,604	0.372	0.25
4	9,341,413,718	0.373	0.28
5	8,088,031,611	0.348	0.23
6	7,990,566,829	0.343	0.24
7	8,089,925,563	0.343	0.25
8	8,411,037,879	0.350	0.25
9	8,760,825,805	0.354	0.27
10	8,632,472,714	0.355	0.29
11	31,657,748,492	0.499	0.50
12	22,142,073,551	0.485	0.48
13	15,040,455,725	0.436	0.44
14	15,843,245,584	0.421	0.40
15	29,881,899,631	0.492	0.47
16	24,557,759,608	0.495	0.50
17	26,411,053,777	0.494	0.50
18	29,469,016,846	0.518	0.54

$$N_t^* = \left[\frac{D_{0,t} \cdot K_t}{\lambda . L^{K_t}}\right]^{\frac{1}{K_t+1}}. \tag{22}$$

To obtain the Lagrangian multiplier, λ, we should satisfy the rate budget constraint in Eq. (14), i.e.:

$$\sum_{t=1}^{M} N_t^* = \sum_{t=1}^{M} \left[\frac{D_{0,t} \cdot K_t}{L^{K_t}}\right]^{\frac{1}{K_t+1}} \cdot \lambda^{-\frac{1}{K_t+1}} = P. \tag{23}$$

Therefore, the optimal solution of the inter-GOP rate allocation problem in Eq. (14) may be expressed in the words of closed-form relations, as in Eqs. (22) and (23). However, there is a problem in obtaining the optimal solution from Eq. (22) to (23). According to Eq. (22), N_t^* for each GOP t depends upon R-D model parameters ($D_{0,t}$ and K_t) and the Lagrangian multiplier, λ. To obtain the Lagrangian multiplier, λ, from Eq. (23), the following parameters are required:

- The joint source-channel R-D model parameters, $D_{0,t}$ and K_t, for all the GOPs of the video sequence, $t = 1, \ldots, M$,
- The required rate for encoding motion information, R_t^{MV}, for all the GOPs of the video sequence, $t = 1, \ldots, M$.

Hence, a "pre-coding stage" is needed to compute the λ value, i.e. it is required to calculate the values of $D_{0,t}$, K_t and R_t^{MV} for "all" the GOPs of the video sequence, $t = 1, \ldots, M$, before calculating the optimal values of the allocated number of packets for each GOP. This fact causes a very long delay before encoding a desired video sequence. In addition, we cannot use this procedure for "real-time" application of the video coding because there is no access to the parameters ($D_{0,t}$, K_t and R_t^{MV}) of future GOPs while encoding any GOP. Therefore, the optimal inter-GOP rate allocation can be used for only "offline" video coding with a very long pre-coding delay. To solve this problem, we propose a suboptimal inter-GOP rate allocation in the next subsection.

5.3. Compromising between encoding delay and quality improvement

As mentioned in the previous subsection, a very long pre-coding delay may be needed to obtain the value of the Lagrangian multiplier, λ. In this subsection, we propose a suboptimal solution for this problem by means of defining the concept of a group of GOPs (GOGOPs). However, we leave the problem of finding the optimal value of λ as an open problem for the readers.

We define the GOGOP as a group of successive GOPs of a video sequence. We assume no overlapping between the successive GOGOPs. The number of GOPs contained in a GOGOP is denoted by M', i.e. there are M' GOPs in each GOGOP. In the other words, there are $M' \cdot T$ frames per GOGOP. The possible values for M' are $1, \ldots, M$.

To compromise the encoding delay with PSNR quality improvement, we propose to perform the inter-GOP rate allocation throughout the GOPs of each GOGOP, rather than the GOPs throughout the whole video sequence. In fact, we can buffer the frames of each GOGOP before encoding and then calculate the corresponding values of $D_{0,t}$, K_t and R_t^{MV} for all GOPs contained in the buffered GOGOP. Similar to Eq. (23), we can calculate the suboptimal value of the Lagrangian multiplier, λ', for the current GOGOP (rather than calculating the λ value over the whole sequence) from the following relation:

$$\sum_{t=1}^{M'} \left[\frac{D_{0,t} \cdot K_t}{L^{K_t}}\right]^{\frac{1}{K_t+1}} \cdot \lambda'^{-\frac{1}{K_t+1}} = P', \tag{24}$$

where:

$$P' = \frac{\left(S' - \sum_{t=1}^{M} R_t^{MV}\right)}{L}, \tag{25}$$

and:

$$S' = \frac{r_b}{q} \cdot \frac{M'.T}{r_f}. \tag{26}$$

The proposed solution allows us to limit the encoding delay for offline video coding, i.e. the encoding delay is equal to the duration of a GOGOP rather than the whole video duration. By changing the GOGOP size, M', one can control the trade-off between encoding delay and quality improvement (i.e., the more GOGOP size, the more encoding delay and the more quality improvement relative to the equal inter-GOP rate allocation). By means of the concept of GOGOP, we can apply the suboptimal inter-GOP rate allocation procedure (and, subsequently, the progressive ULP packetization approach) for real-time video coding. In the case of real-time video coding, an

adjustable buffering delay is introduced through the encoding procedure.

There is an alternative approach for compromising the encoding delay with PSNR efficiency. One can use a causal sliding window approach to estimate a suboptimal value for λ. In fact, it is possible to compute the optimal number of allocated packets to the tth GOP, N_t^*, from Eq. (22), whose λ value is calculated from Eq. (23) over a Z-sized sliding window, including the GOPs $t-Z+1$,$t-Z+2$,. . .,t (i.e., the summation in Eq. (23) is over the GOPs $t-Z+1$,$t-Z+2$,. . .,t). The difference between the sliding window approach and the GOGOP scheme refers to the window consideration (non-overlapping windows in the case of GOGOPs vs. overlapping sliding windows). We have tested the overlapping sliding windows in our experimentations (although not reported in the experimental results) and compared them with GOGOPs. Both windowing schemes have nearly identical efficiencies in the sense of average PSNR. The overlapping sliding windowing scheme, however, has less encoding delay compared to the non-overlapping GOGOPs, because for each incoming GOP t, it is enough to measure the corresponding model parameters ($D_{0,t}$ and K_t values) and then calculate the required λ value from the model parameters of the current GOP and its $Z-1$ previous GOPs. In the case of non-overlapping GOGOPs, it is necessary to buffer the GOPs of each GOGOP and then measure their model parameters, so that one can calculate the λ value from the model parameters of the GOPs contained in the GOGOP. Therefore, the pre-encoding delay of the GOGOP is more than the sliding window scheme. The advantage of the GOGOP scheme compared to the sliding window is its lower computational complexity, as the GOGOP scheme calculates the λ value only for each GOGOP, whereas the overlapping sliding window calculates the λ value for each individual GOP. It is noteworthy that calculating the λ value from (23) necessitates solving a non-linear equation using a numerical method (e.g. Newton–Raphson method).

6. Experimental results

In [1], the performance of the progressive ULP packetization approach is evaluated and compared with that of the local search algorithm [2]. In this section, we compare the PSNR performance of the progressive approach with two inter-GOP rate allocation strategies: "equal" rate allocation and "optimal" rate distribution. In the equal inter-GOP rate allocation strategy, the same total rate budget is allocated to each GOP, i.e.:

$$R_1 = R_2 = \cdots = R_M = \frac{r_b}{q}\frac{T}{r_f}. \tag{27}$$

Therefore, in the case of equal inter-GOP rate allocation strategy, the number of allocated packets for each GOP t is as follows:

$$N_t^{equal} = \frac{\left(\frac{r_b T}{q r_f} - R_t^{MV}\right)}{L}. \tag{28}$$

In the case of "optimal" inter-GOP rate allocation, the number of packets allocated to each GOPt, i.e. N_t^*, is calculated from Eqs. (22) to (23).

Experimental results were preformed on four different 4:2:0 YUV test sequences, which are summarized in Table 5. The sequences are selected because of their different characteristics in terms of motion and spatial detail, and also various spatial/temporal resolutions. Each GOP includes 16 frames. The size of wavelet-decomposed CBs in T+2D scalable video coding is 16×16. We have assumed a binomial distribution, with a packet loss rate of p for the packet loss of the channel. It means that each of the packets may be independently lost with the probability of p. We have not aimed the precise modeling of the channel packet erasure, which may be bursty. However, one may refer to [21] for more information about Markov modeling of burst packet losses. Packet losses are simulated by 120 Monte-Carlo simulations and the results are averaged over them. Because ATM packets have a payload length of 48 bytes and one byte is required for the sequence number, we have chosen $L = 47$ bytes to mimic a practical application. The average PSNR of the Y component is used as a meaningful measure of performance.

Table 5: The source signals used in simulations.

Sequence name	Frame size	#of frames	Frame rate (fps)
Foreman	QCIF	384	15
Akiyo	QCIF	288	15
Mobile	CIF	288	30
Stefan	CIF	288	30

Table 6: The average PSNR improvement (*Imp*) of the optimal rate allocation relative to the equal inter-GOP rate distribution for the four test sequences and two different channel conditions (p and r_b).

Sequence name	Good channel conditions	Bad channel conditions
Foreman	$p = 0.05$, $r_b = 65$ Kbps, $Imp = 0.27$ dB	$p = 0.10$, $r_b = 45$ Kbps, $Imp = 0.65$ dB
Akiyo	$p = 0.05$, $r_b = 21$ Kbps, $Imp = 0.03$ dB	$p = 0.10$, $r_b = 7$ Kbps, $Imp = 0.11$ dB
Mobile	$p = 0.05$, $r_b = 1024$ Kbps, $Imp = 0.01$ dB	$p = 0.10$, $r_b = 512$ Kbps, $Imp = 0.17$ dB
Stefan	$p = 0.05$, $r_b = 1024$ Kbps, $Imp = 0.05$ dB	$p = 0.10$, $r_b = 512$ Kbps, $Imp = 0.08$ dB

Figure 6 evaluates the performance of the progressive ULP packetization approach with optimal and equal inter-GOP rate allocation strategies. Figure 6 depicts the profile of the average Y-component PSNR for each GOP of the four test sequences. For each test sequence, two different channel conditions are considered: one with $p = 0.05$ and a high average bit rate and another with $p = 0.10$ and a lower average bit rate. It can be seen from Figure 6 that the optimal inter-GOP rate allocation strategy yields a relatively flatter PSNR profile throughout the video sequence. In fact, there is not a deep valley in the PSNR profile of the optimal rate distribution, because the optimal strategy distributes the total rate budget between the GOPs, such that GOPs with more (less) motion get more (less) rate budget relative to the equal strategy. Figure 7 depicts the number of packets allocated to each GOP using optimal and equal rate distribution. Table 6 summarizes the average PSNR improvement of the optimal rate allocation relative to the equal inter-GOP rate distribution for the four test sequences. Table 6 reveals that average PSNR improvement (due to the optimal rate allocation) is more significant for lower bit rate budgets and higher packet loss rates. In fact, it is more beneficial to use optimal rate allocation when the channel has less bit rate budget and more packet loss rate. In addition, Table 6 shows that the optimal rate allocation has more PSNR gain for video

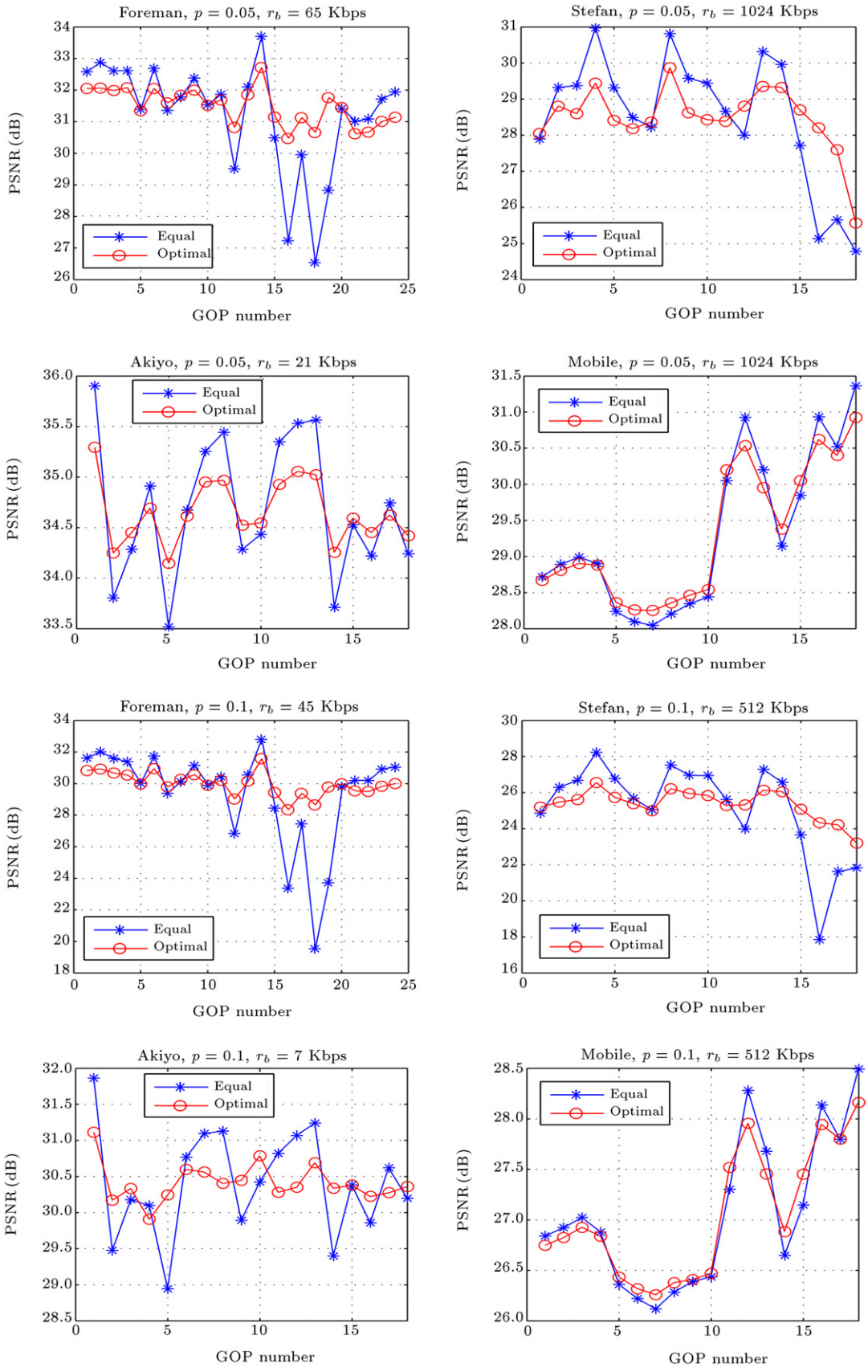

Figure 6: The average PSNR of the Y-component for each GOP of the four test sequences. The progressive ULP packetization approach is applied with optimal and equal inter-GOP rate allocation strategies. For each test sequence, two different channel conditions (p and r_b) are considered.

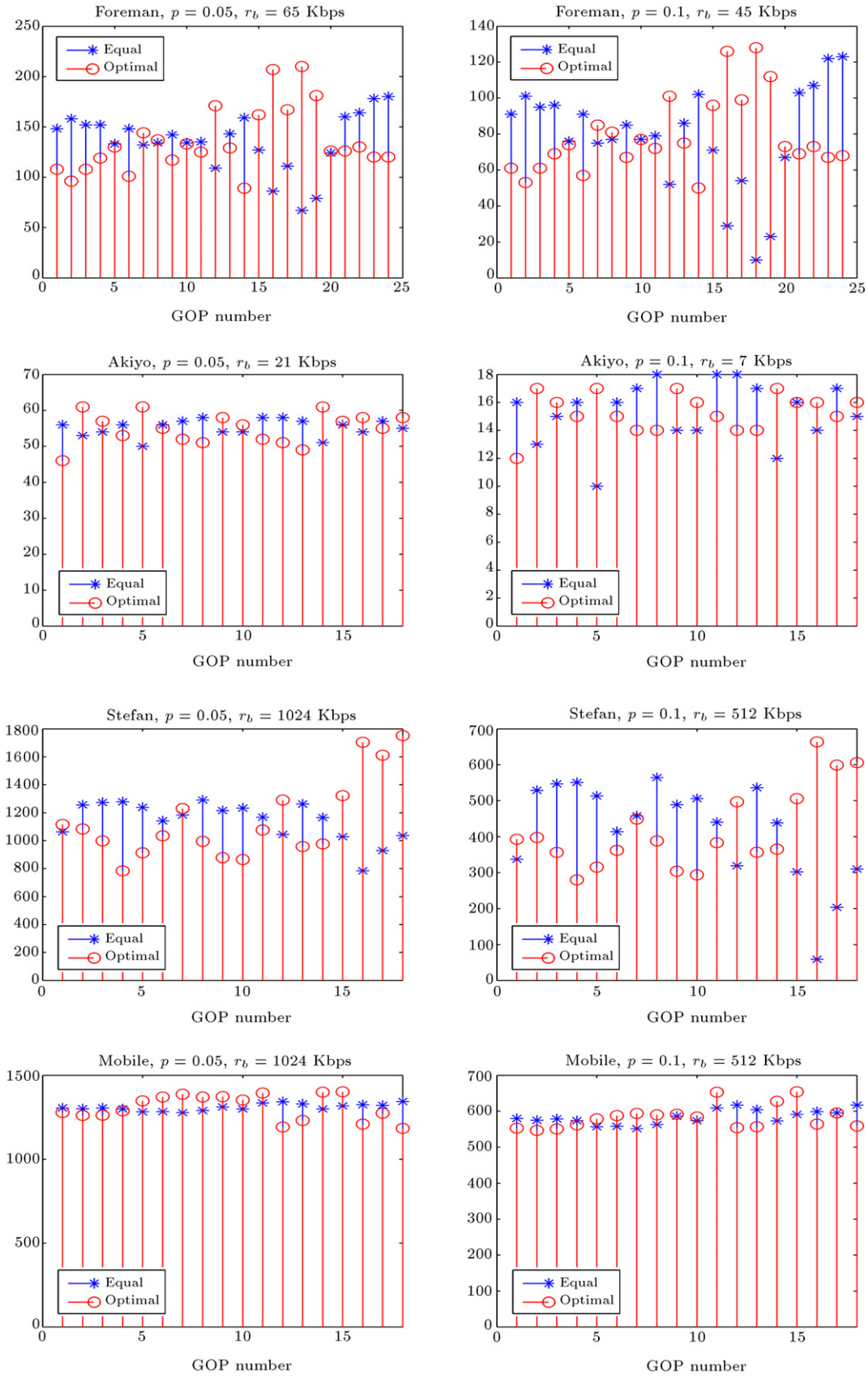

Figure 7: The Number of allocated packets for each GOP of the four test sequences. The progressive ULP packetization approach is applied with optimal and equal inter-GOP rate allocation strategies. For each test sequence, two different channel conditions (p and r_b) are considered.

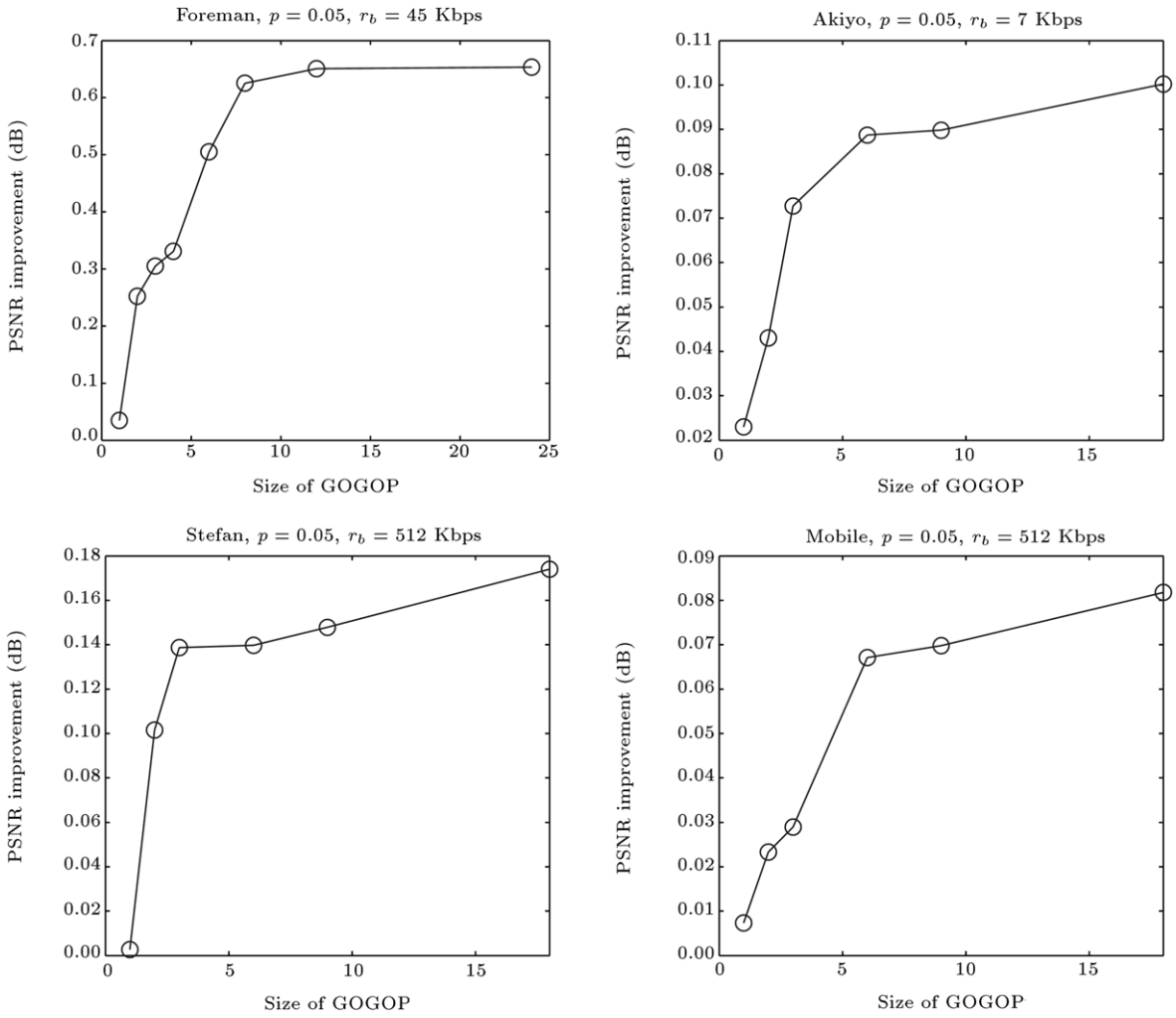

Figure 8: The average PSNR improvement of the suboptimal inter-GOP rate allocation relative to the equal inter-GOP rate distribution versus the GOGOP size for the four test sequences.

sequences with more variations in their motion characteristics. For example, Table 6 shows that the *Foreman* sequence has the most PSNR improvement because the *Foreman* sequence has dissimilar motion trends from the start to the end (i.e., medium, high, and low motion at its beginning, middle, and end parts, respectively). However, the other test sequences have a relatively homogeneous motion trend all over their duration.

As mentioned in Section 5, obtaining the optimal numbers of allocated packets for each GOP requires the knowledge of $D_{0,t}$, K_t and R_t^{MV} for all GOPs contained in the video sequence. Therefore, the optimal inter-GOP rate allocation is impossible for real-time applications. In addition, the optimal inter-GOP rate allocation needs a pre-coding stage for buffering all GOPs of the video sequence and calculating their corresponding values of $D_{0,t}$, K_t and R_t^{MV}. This pre-coding stage may introduce a very long pre-coding delay. To compromise the pre-coding delay with the average PSNR quality, we proposed a suboptimal procedure to perform the inter-GOP rate allocation for the GOPs of each GOGOP. Figure 8 depicts the average PSNR improvement of the suboptimal inter-GOP rate allocation relative to the equal inter-GOP rate distribution versus the GOGOP size (i.e. the number of GOPs contained in each GOGOP) for the four test sequences. Figure 8 shows a saturating behavior for this curve. Therefore, one may choose a relatively small GOGOP size (much smaller than the total number of GOPs contained in the video sequence, e.g. GOGOP size of 8 GOPs) and yet, the PSNR quality of the suboptimal inter-GOP rate allocation is near to the optimal inter-GOP rate distribution.

7. Concluding remarks

In this paper, the ULP packetization problem, including the segmentation of a generally-defined embedded source code stream, has been explained and a low complexity progressive method to determine the suboptimal size of each data segment is expressed. The progressive method is based on an analytically-derived relation between the sizes of any two successive segments.

It is reasonable to consider a joint source-channel R-D curve describing average reconstruction distortion, in terms of the number of allocated packets (i.e. the rate budget). In this paper, we have proposed a hyperbolic model for this curve, which helps us derive some simple closed-form relations for the number of allocated packets (i.e. the optimal rate budget) for each GOP of the video sequence. The experimental results show that the optimal inter-GOP rate allocation strategy outperforms the conventional strategy of equal inter-GOP rate distribution up to 0.65 dB. Experimental results show that average PSNR

improvement (due to the optimal rate allocation) is more significant for lower bit rate budgets and higher packet loss rates. In addition, the optimal rate allocation has more PSNR gain for the video sequences with more variations in their motion characteristics.

The optimal inter-GOP rate allocation has an essential shortage, due to a long pre-coding delay. To calculate the optimal number of allocated packets for each GOP, it is needed to know the motion information rate and hyperbolic model parameters of the joint source-channel R-D curve of the whole GOPs of the video sequence. This limits the usage of the optimal inter-GOP rate allocation to offline applications with a rather short duration of the video. To compromise the pre-coding delay with the average PSNR quality, we proposed a suboptimal procedure to perform the inter-GOP rate allocation for the GOPs of each GOGOP. The simulation results show that one may choose a rather small GOGOP size (e.g. GOGOP size of 8 GOPs) and yet, the PSNR quality of the suboptimal inter-GOP rate allocation is near to the optimal inter-GOP rate distribution. This fact yields an important conclusion; "To obtain the optimal number of allocated packets for each GOP, it is sufficient to take a few number of near GOPs into account, however, the impact of the far GOPs is nearly negligible".

References

[1] Ardestani, M.R., et al. "Unequal loss-protected multiple description coding of scalable source streams using a progressive approach", *16th IEEE Int. Conf. Image Process. (ICIP)*, pp. 901–904 (2009).

[2] Stankovic, V., et al. "Packet loss protection of embedded data with fast local search", *Proc. IEEE ICIP-2002*, Rochester, NY (Sept. 2002).

[3] Xu, X., et al. "A peer-to peer video-on-demand system using multiple description coding and server diversity", *Proc. IEEE Int. Conf. Image Processing* (2004).

[4] Shen, Y., et al. "Streaming layered encoded video using peers", *Proc. IEEE Int. Conf. on Multimedia and Expo* (2005).

[5] Ohm, J.-R. "Advances in scalable video coding", *Proc. IEEE*, 93(1), pp. 42–56 (2005).

[6] Adami, N., et al. "State-of-the art and trends in scalable video compression with wavelet-based approaches", *IEEE Trans. Circuits Syst. Video Techno.*, 17(9), pp. 1238–1255 (2007).

[7] Goyal, V.K. "Multiple description coding: compression meets the network", *IEEE Signal Process. Mag.*, 18, pp. 74–93 (2001).

[8] Puri, R. and Ramchandran, K. "Multiple description coding using forward error correction codes", *Proc. 33rd Asilomar Conf. on Signals and Systems*, Pacific Grove, CA Oct. (1999).

[9] Dumitrescu, S., et al. "Globally optimal uneven error-protected packetization of scalable code streams", *Proc. DCC'02*, Snowbird, Utah (Apr. 2002).

[10] Mohr, A.E., et al. "Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction", *IEEE J. Sel. Areas Commun.*, 18(1), pp. 819–828 (2000).

[11] Flierl, M. and Girod, B. "Video coding with motion-compensated lifted wavelet transforms", *Signal Process., Image Commun.*, 19(7), pp. 561–575 (2004).

[12] Turaga, D., et al. "Unconstrained motion compensated temporal filtering (UMCTF) for efficient and flexible interframe wavelet video coding", *Signal Process., Image Commun.*, 20(1), pp. 1–19 (2005).

[13] Joint Scalable Video Model JSVM-6, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Doc. JVT-S202, 31 Mar.-7 (Apr. 2006).

[14] Joint Scalable Video Model, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Doc. JVT-X202 (Jul. 2007).

[15] Available at http://www.ece.uvic.ca/mdadams/jasper.

[16] Aulí-Llinàs, F. and Marcellin, M.W. "Distortion estimators for bitplane image coding", *IEEE Trans. Image Processing*, 18(8), pp. 1772–1781 (2009).

[17] Taubman, D.S. and Marcellin, M.W., *JPEG 2000: Image Compression Fundamentals, Standards, and Practice Ser*, Kluwer Engineering and Computer Science (2002).

[18] Akyol, E., et al. "A flexible multiple description coding framework for adaptive peer-to-peer video streaming", *IEEE J. Sel. Top. Sign. Process.*, 1(2), pp. 231–245 (2007).

[19] Cheng, Ch.-Ch., et al. "Distortion estimation and bit allocation for MCTF based3-D wavelet video coding", *IEEE Int. Conf. Image Processing*, pp. 2844–2847 (2008).

[20] Ardestani, M.R., et al. "Rate-distortion modeling for scalable video coding", *17th IEEE Int. Conf. Telecommun. (ICT)*, pp. 923–928 (2010).

[21] Li, Z., et al. "Modeling and analysis of distortion caused by Markov-model burst packet losses in video transmission", *IEEE Trans. Circuits Syst. Video Techno.*, 19(7), pp. 917–931 (2009).

Majid Roohollah Ardestani received a B.S. degree in Electrical Engineering in 2003 from Tehran University (UT), and a M.S. degree in Communication Systems Engineering in 2005 from Iran University of Science and Technology (IUST), Tehran. He is currently pursuing a Ph.D. degree at IUST. He worked as a research assistant at the Iran Telecommunication Research Center (ITRC) in 2002 for eight months. His research interests include scalable video compression, network distribution of the compressed multimedia content, mobile communications, OFDM multicarrier modulation over DVB and mobile systems, and spread spectrum wireless communications. He is also co-author of many publications.

Ali Asghar Beheshti Shirazi received B.S. and M.S. degrees in Communication Engineering from Iran University of Science and Technology (IUST) in 1984 and 1987, respectively, and a Ph.D. from Okayama University, Japan in 1995. In 1995, he joined the Department of Electrical Engineering, IUST, where he is currently Assistant Professor. His research interests include digital image processing and compression, data communication networking and secure communication.