International Journal of Digital Multimedia Broadcasting

Advances in Video Coding for Broadcast Applications

Guest Editors: Susanna Spinsante, Ennio Gambi, Lorenzo Ciccarelli, Andrea Lorenzo Vitali, Jorge Sastre Martínez, and Paul Salama

Advances in Video Coding for Broadcast Applications

International Journal of Digital Multimedia Broadcasting

Advances in Video Coding for Broadcast Applications

Guest Editors: Susanna Spinsante, Ennio Gambi, Lorenzo Ciccarelli, Andrea Lorenzo Vitali, Jorge Sastre Martínez, and Paul Salama

Contents

Advances in Video Coding for Broadcast Applications, Susanna Spinsante, Ennio Gambi, Lorenzo Ciccarelli, Andrea Lorenzo Vitali, Jorge Sastre Martínez, and Paul Salama
Volume 2009, Article ID 368326, 2 pages

New Adaptive Algorithms for GOP Size Control with Return Channel Suppression in Wyner-Ziv Video Coding, Charles Yaacoub, Joumana Farah, and Béatrice Pesquet-Popescu
Volume 2009, Article ID 319021, 11 pages

Intra-Skip in Inter-Frame Coding of H.264/AVC, Hui Su
Volume 2009, Article ID 141986, 8 pages

An Adaptive Systematic Lossy Error Protection Scheme for Broadcast Applications Based on Frequency Filtering and Unequal Picture Protection, Marie Ramon, François-Xavier Coudoux, and Marc Gazalet
Volume 2009, Article ID 709813, 7 pages

Adaptive Error Resilience for Video Streaming, Lakshmi R. Siruvuri, Paul Salama, and Dongsoo S. Kim
Volume 2009, Article ID 681078, 10 pages

Statistical Time-Frequency Multiplexing of HD Video Traffic in DVB-T2, Mehdi Rezaei, Imed Bouazizi, and Moncef Gabbouj
Volume 2009, Article ID 186960, 12 pages

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2009, Article ID 368326, 2 pages
doi:10.1155/2009/368326

Editorial

Advances in Video Coding for Broadcast Applications

Susanna Spinsante,[1] Ennio Gambi,[1] Lorenzo Ciccarelli,[2] Andrea Lorenzo Vitali,[3] Jorge Sastre Martínez,[4] and Paul Salama[5]

[1] *Department of Biomedic Engineering, Electronics and Telecommunications, Università Politecnica delle Marche, Via Brecce Bianche 12, 60131 Ancona, Italy*
[2] *Video Broadcasting Coding, Basingstoke, UK*
[3] *STMicroelectronics, Research & Innovation, 20041 Milano, Italy*
[4] *Image and Video Processing Group (GPIV), Multimedia Applications and Telecommunications Institute (iTEAM), Technical University of Valencia, Camino de Vera, 46022 Valencia, Spain*
[5] *Electrical and Computer Engineering, Indiana University-Purdue University, 723 West Michigan Street, SL160, Indianapolis, IN 46202, USA*

Correspondence should be addressed to Susanna Spinsante, s.spinsante@univpm.it

Received 31 March 2009; Accepted 31 March 2009

The growing diffusion of new services, like mobile television and video communications, based on a variety of transmission platforms (3G, WiMax, DVB-S/T/H, DMB, DTMB, Internet, etc.), emphasizes the need of advanced video coding techniques able to meet the requirements of both the receiving devices and the transmission networks. In this context, scalable and layered coding techniques represent a promising solution when aimed at enlarging the set of potential devices capable of receiving video content. Video encoders' configuration must be tailored to the target devices and services, that range from high definition, for powerful high-performance home receivers, to video coding for mobile handheld devices. Encoder profiles and levels need to be tuned and properly configured to get the best tradeoff between resulting quality and data rate, in such a way as to address the specific requirements of the delivery infrastructure. As a consequence, it is possible to choose from the entire set of functionalities of the same video coding standard in order to provide the best performance for a specified service. Among the most recent video coding standards, the ITU-T H.264/AVC offers a wide set of configurations, that make it able to address several different services, ranging from video streaming, to DVB-T/H broadcasting, to videoconferencing over IP networks.

This special issue aims to present state-of-the-art research and developing activities contributing to all aspects of video coding solutions focused on broadcast applications. The Call-for-Papers for this special issue resulted in the excellent submissions from around the world in both quality and quantity. After two rounds of careful reviews from about fifty experts in related fields, five papers were selected to be included in this special issue covering major topics, from coding related issues to error resilience and protection and to statistical multiplexing, thus providing a comprehensive overview on the area.

The first paper entitled "New adaptive algorithms for GOP size control with return channel suppression in Wyner-Ziv video coding" by C. Yaacoub et al. presents a novel algorithm for adaptive GOP size control in distributed Wyner-Ziv video coding, where key frames are intra coded according to the H.264/AVC standard. In order to avoid the use of a feedback channel, which is generally missing in broadcasting applications, theoretical calculations are used to estimate the bit rate necessary for successful decoding of Wyner-Ziv frames. The proposed system is also able to automatically switch to H.264 intra coding mode in image regions where it outperforms Wyner-Ziv encoding, to improve the overall PSNR with respect to pure H.264 intra coding, and fixed-GOP Wyner-Ziv coding.

The second paper, entitled "Intra skip in inter frame coding of H.264/AVC" by H. Su, deals with a novel mathematical model to skip intra mode predictions in inter frames coding according to H.264/AVC. The model is basically built around a weighted-coefficient function used to set a proper threshold, which, in its turn, affects the skipping of intra partitions evaluation. The aim of the proposed algorithm

is to minimize the complexity of inter coding, by cutting down encoding time, while facing a very minor increase of the resulting bit rate, thus making H.264/AVC codecs more suitable to real time applications. Moreover, the proposed solution can be applied in conjunction with any proposed "fast" inter and intra methods relying on inter partition mode decision, motion search algorithms, and fast intra algorithms.

Moving from specific H.264/AVC coding issues to the problems related to broadcast video transmission, the third paper entitled "An adaptive systematic lossy error protection scheme for broadcast applications based on frequency filtering and unequal picture protection", by M. Ramon et al., discusses an adaptive systematic lossy error protection scheme in which the Wyner-Ziv stream is obtained by means of frequency filtering in the transform domain. Besides that, error resilience may vary adaptively, according to the characteristics of the compressed video. By applying the proposed solution, a graceful degradation of the reconstructed video quality may be obtained even in presence of increasing transmission errors, and the proposed scheme is shown to provide better performance than other solutions, such those based on coarser quantization.

Error resilience in the context of video streaming is the topic discussed by the fourth paper entitled "Adaptive error resilience for video streaming", by L. R. Siruvuri et al. It is well known that compressed video sequences may be strongly affected by channel errors, and suitable channel coding schemes may reduce the impact of errors on the quality of the decoded video. The paper suggests a solution to adapt the number of Reed-Solomon parity symbols used to protect a compressed video sequence against channel errors, in order to minimize the impact of the redundant bits on the available bandwidth. This paper spans the reference broadcasting scenario, by assuming the availability of a return channel through which feedbacks from the client are used to properly tune the amount of parity bits. As a matter of fact, the possibility of a return channel is already foreseen in some broadcasting systems, such as the Digital Video Broadcasting Return Channel Satellite, a possible scenario in which the proposed solution could find application.

The last paper included in this special issue, "Statistical time-frequency multiplexing of HD video traffic in DVB-T2" by M. Rezaei et al. presents a model for describing High Definition video traffic, which is used to evaluate the performance of statistical multiplexing of HD broadcast services over DVB-T2. New features introduced by DVB-T2 with respect to DVB-T, such as a two-dimensional multiplexing of broadcast services in time and frequency domains, and a time-frequency slicing transmission scheme, allow to increase the flexibility of service multiplexing, which may be of vital importance when dealing with HD video broadcasting. The model presented in the paper is able to simulate a wide range of synthetic video bit streams with practical and statistical metrics of interest; performance evaluation of DVB-T2 shows the possibility of getting improved performance in terms of bandwidth efficiency, end-to-end delay, and video quality for the broadcast system.

As we conclude this overview, we would like to express our sincere gratitude to all the reviewers for their timely and insightful comments on the submitted manuscripts, which made this special issue possible.

Susanna Spinsante
Ennio Gambi
Lorenzo Ciccarelli
Andrea Lorenzo Vitali
Jorge Sastre Martínez
Paul Salama

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2009, Article ID 319021, 11 pages
doi:10.1155/2009/319021

Research Article

New Adaptive Algorithms for GOP Size Control with Return Channel Suppression in Wyner-Ziv Video Coding

Charles Yaacoub,[1,2] Joumana Farah,[1] and Béatrice Pesquet-Popescu[2]

[1] *Engineering Department, Faculty of Sciences and Computer Engineering, Holy-Spirit University of Kaslik, P.O. Box 446, Jounieh, Keserwan, Mount Lebanon, Lebanon*
[2] *Signal and Image Processing Department, TELECOM ParisTech, 46 Rue Barrault, 75634 Paris Cedex 13, France*

Correspondence should be addressed to Charles Yaacoub, charlesyaacoub@usek.edu.lb

Received 28 June 2008; Revised 8 September 2008; Accepted 22 October 2008

Recommended by Jorge Sastre Martínez

We present novel algorithms for adaptive GOP size control in distributed Wyner-Ziv video coding, where an H.264 video codec is used for intracoding of key frames. The proposed algorithms rely on theoretical calculations to estimate the bit rate necessary for the successful decoding of Wyner-Ziv frames without the need for a feedback channel, which makes the system suitable for broadcasting applications. Additionally, in regions where H.264 intracoding outperforms Wyner-Ziv coding, the system automatically switches to intracoding mode in order to improve the overall performance. Simulations results show a significant gain in the average PSNR that can reach 3 dB compared to pure H.264 intracoding, and 0.8 dB compared to fixed-GOP Wyner-Ziv coding.

1. Introduction

Distributed source coding [1–20] has recently become a topic of great interest for the research community, especially in the world of video communications. In traditional video coding techniques, such as MPEG or H.26x, motion estimation is performed at the encoder side, which yields very complex encoders, but simple decoders. This is suitable for applications where a video sequence is encoded once and decoded several times, such as video broadcasting or video streaming on demand. A simple decoder is desired in this case to allow low-cost receivers for the end users.

On the other hand, some applications require simple encoders. Distributed Video Coding (DVC) was introduced [7, 8] to permit low-complexity encoding for small power-limited and memory-limited devices, such as camera-equipped mobile phones or wireless video sensors, by moving the computation burden from the encoder side to the decoder. Increased decoding complexity can be tolerated in this case since, in such applications, the decoder is usually located in a base station with sufficient resources.

It is known from information theory that, given two statistically dependent sources X and Y, each source can be independently compressed to its entropy limit, $H(X)$ and $H(Y)$, respectively. However, by exploiting the correlation statistics between these sources, X and Y can be jointly compressed to the joint entropy $H(X,Y)$. This results in a more efficient compression since $H(X,Y) \leq H(X) + H(Y)$. The idea behind DVC goes back to the 1970s when Slepian and Wolf [21] proved that, if the source Y is compressed to its entropy limit $H(Y)$, X can be transmitted at a rate very close to the conditional entropy $H(X|Y)$, provided that Y is available at the receiver as side information for decoding X. Since $H(X,Y) = H(Y) + H(X|Y)$, X and Y can be independently encoded and jointly decoded without any loss in the compression efficiency, compared to the case where both sources are jointly encoded and decoded. The application of this concept to lossy source coding is known as the Wyner-Ziv coding [22].

In practical DVC systems, a subset of frames, known as key frames, is usually compressed using traditional intracoding techniques. One or more frames following each key frame, known as Wyner-Ziv (WZ) frames, are then

compressed by appropriate puncturing of the parity bits at the output of a channel coder. At the receiver side, previously decoded (key or WZ) frames are interpolated to generate the necessary side information for the decoding process.

The first practical DVC systems appeared in 2002, when Puri and Ramchandran [7] proposed a block-based codec using syndromes, and Aaron et al. [8] proposed a frame-based codec using turbo codes. The frame-based approach has gained a greater interest in the research community. However, it still suffers from several weaknesses that limit its use in real-life applications.

One of the main drawbacks of current DVC systems is the use of a feedback channel (FC) [11] to allow flexible rate control and to ensure successful decoding of WZ frames. The FC is not suitable for real-time systems (e.g., broadcasting applications) due to transmission delay constraints. Additionally, in multiuser applications with rate constraints, the application of WZ coding becomes impractical because of the difficulty of implementing appropriate rate allocation algorithms. Furthermore, since several decoding runs are required to successfully recover a WZ frame, the FC imposes instantaneous decoding in the receiver. For all these reasons, the introduction of new techniques for estimating the necessary bit rate to successfully decode each WZ frame becomes crucial. In fact, the problem of the return channel in DVC has rarely been targeted in the literature. Artigas and Torres [12] and Morbée et al. [13] proposed techniques that rely on performance tables used by the encoder to predict the compression level of each particular frame. Kubasov et al. proposed in [18] an encoder rate control technique that reduces the use of the feedback channel. Transform domain WZ rate control algorithms were introduced in [19] (for a DCT-based WZ codec) and [20] (for a wavelet-based WZ codec). However, these studies do not take into account the rate constraints in limited-bandwidth applications. Besides, the influence of the channel impairments on the proposed rate allocation techniques is not considered. In [6, 14, 15], we proposed a novel technique for the removal of the feedback channel in DVC systems, using an analytical approach based on entropy calculations. Designed for a multiuser scenario, the proposed technique takes into account the amount of motion in the captured video scene as well as the transmission channel conditions for every user, in order to allocate unequal transmission rates among the different users.

Another drawback of current DVC systems is that the quality of the generated side information greatly affects the system's performance. Even though the interpolation algorithm used at the decoder strongly influences the side information quality, key frames are essential components in the interpolation process and thus, having high-quality key frames is crucial. Therefore, a very high peak signal-to-noise ratio (PSNR) is desired (for the key frames) in order to allow a successful decoding of the WZ frames at feasible WZ bit rates. This condition can result in a very high bit rate requirement, which is not possible in limited-bandwidth applications. Additionally, when the key frames are too distant apart, the quality of the side information is degraded. As a result, most research on DVC considers a group of pictures (GOP) size of 2, that is, each key frame is followed by one WZ frame. Several attempts have been made to increase the GOP size in DVC. In [16], Aaron et al. impose the use of high-quality key frames with fixed GOP sizes ranging from 2 to 5. As the GOP size increases, the system's performance decreases. However, lower rates could be reached with greater GOP sizes due to the high bit rate requirements of the key frames. In [17], Ascenso et al., present a content-adaptive GOP size selection algorithm. The number of frames in a GOP is determined dynamically depending on motion activity. However, the proposed algorithm uses four different metrics in order to determine the size of a GOP, which results in a significant increase of the encoder's complexity. Furthermore, both studies rely on a feedback channel for the decoding of WZ frames, and on H.263+ for key frame encoding. Since H.264/AVC [23] greatly outperforms H.263+, it is expected that H.264 intracoding will outperform both Wyner-Ziv systems too.

In this paper, we present novel algorithms for dynamically varying the GOP size in distributed video coding. Our simulations are performed using a pixel-domain WZ video codec. However, the same algorithms can be applied in a transform-domain codec, which improves the overall performance at the expense of a slight increase in encoding and decoding complexity. Our method relies on H.264 for the encoding of key frames, and on our previously developed WZ rate estimation technique presented in [5, 6, 14, 15], where quadri-binary turbo-codes are used for the compression of WZ frames. Only one metric is required to determine the size of a GOP, and a feedback channel is not needed for the decoding of WZ frames. Automatic mode selection allows the system to switch to H.264 intracoding mode in regions where H.264 outperforms WZ video coding. Furthermore, based on our study in [5, 6, 14, 15], our algorithms can be easily extended to take into account channel impairments and multiuser scenarios.

This paper is organized as follows: in Section 2, we present a brief description of the Wyner-Ziv video codec used in this study, along with the rate estimation technique for WZ frames. Our adaptive algorithms for GOP size control are detailed in Section 3, and the additional complexity they incur at the encoder side is analyzed in Section 4. Finally, simulation results are presented in Section 5.

2. Description of the Distributed Video Coding System

The distributed video coding system considered in this study can be represented by the block diagram in Figure 1. Key frames are compressed using H.264 intracoding. After H.264 decoding, a key frame is stored in a buffer in order to be used during the process of generating the side information necessary for the decoding of WZ frames.

Compression of the Wyner-Ziv frames starts by a uniform scalar quantization to obtain M-bit representations of the eight-bit pixels, $M \in \{1, 2, 4\}$. The turbo encoder

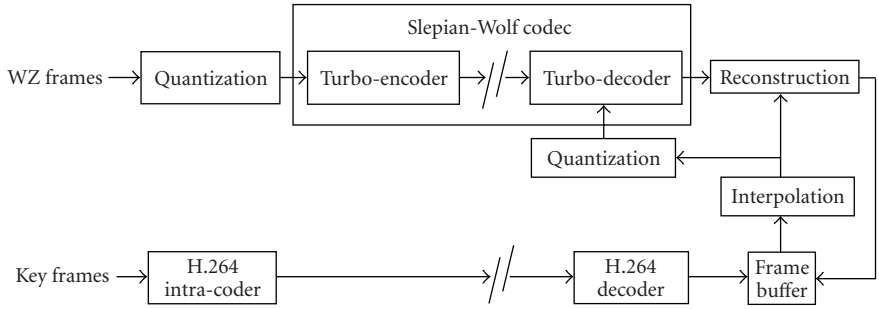

Figure 1: Block diagram of the pixel-domain distributed video coding system.

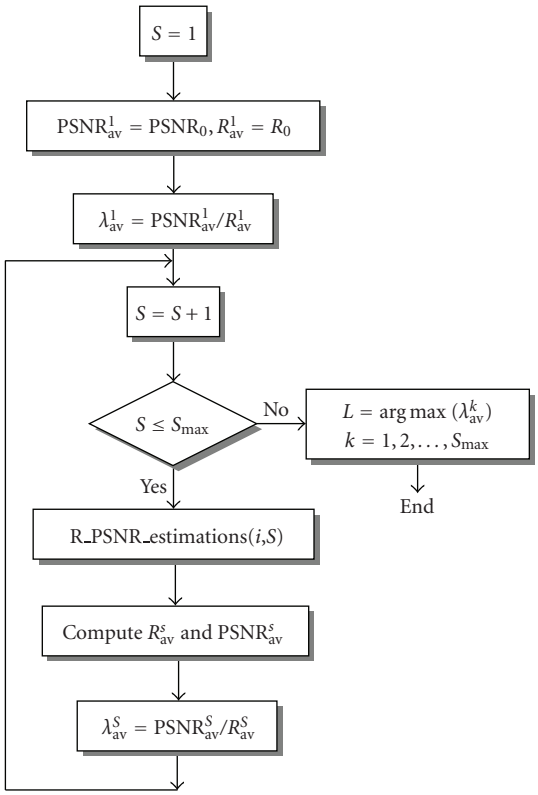

Figure 2: Flow chart diagram of the proposed GOP size control algorithm.

[24–27] consists of a parallel concatenation of two 16-state quadri-binary convolutional encoders separated by an internal interleaver and resulting in a minimum global coding rate of 2/3. The generator polynomials in octal notation are $(23, 35, 31, 37, 27)_8$ from [27]. At the encoder output, systematic information is discarded, while parity information is punctured and transmitted to the decoder. Side information (SI) of a particular WZ frame is generated at the receiver by motion-compensated interpolation of two previously reconstructed (WZ or H.264) frames. The frame interpolation technique assumes symmetric motion vectors as explained in [8, 16]. The interpolated frame is then quantized and fed to the turbo decoder as a noisy version of the missing systematic data. Turbo decoding is realized by iterative Soft Input Soft Output (SISO) decoders based on the Max-Log-MAP (maximum a posteriori) algorithm [28]. However, metric calculations are modified in order to take into account the nonbinary nature of the turbo codec, and the residual signal statistics between the WZ and SI frames. Finally, the reconstruction block is used to recover an eight-bit version of the decoded WZ frames using the available side information [8]. The final output is then stored in a buffer if needed to generate side information for another WZ frame.

In a previous work [5], we presented an analytical approach for estimating the compression limits of a pixel-domain Wyner-Ziv video coding system with a transmission over error-prone channels, without the need for a feedback channel. Simulation results showed that the theoretical bounds can be used in a broadcasting system to predict the compression level for each frame with a minor loss in the decoding PSNR, compared to the classical feedback-based coding system. In the absence of transmission errors, the theoretical compression bound of the WZ frames for the system in Figure 1 can be expressed as (see [5, 6])

$$H(X|Y) = \frac{-1}{2^M} \sum_{i=0}^{2^M-1} \sum_{j=0}^{2^M-1} g\left[c \frac{\alpha}{2} \frac{2^M e^{-\alpha |d_{i-j}|}}{L_{d_{i-j}}} \right], \qquad (1)$$

where X is a quantized WZ frame, Y is the corresponding side information, $g(a) = a \log_2(a)$, $L_{d_{i-j}} = 2^M - |i - j|$ is the number of possible couples (i, j) that yield the difference $i - j = d$, $d_{i-j} = 2^M(i - j)$ is the difference between two quantized pixel values, c is a scaling factor, and α is the parameter of the Laplacian distribution modeling the statistics of the residual error between the side information and the WZ frame [8]. Since there is always a gap between theoretical and practical bounds, we have determined in [14] that, in order to obtain a good average performance, the ratio between the average number of transmitted bits per pixel and the lower compression bound expressed in (1) must be not less than $T_4 = 2.4$ for $M = 4$, $T_2 = 4$ for $M = 2$, and $T_1 = 6$ for $M = 1$. As a result, the encoder can determine the compression rate for a given WZ frame by first determining its compression bound, then multiplying it with the corresponding coefficient T_M, depending on the value of the WZ quantization parameter M.

3. Adaptive Algorithms for GOP Size Control in Wyner-Ziv Video Coding

In a video sequence, when there is low motion, consecutive frames are highly correlated. The aim of varying the GOP size is to allow the system to better exploit this property, by reducing the number of intracoded key frames in regions where WZ frames would yield a better rate-distortion (*R-D*) performance. In regions where intracoding outperforms WZ coding (because of high motion), the GOP structure is reduced to one (H.264 intracoded) frame per GOP. This automatic mode selection allows the WZ encoder to make use of H.264 coding efficiency to better improve the system's *R-D* performance.

Let $S_{\max}$ represent the maximum allowable GOP size. For each GOP, let R_0 represent the average bit rate assigned for the first frame (intracoded key frame) in the GOP, and PSNR_0 its PSNR. $S_{\max}$ can be chosen depending on the system's delay constraints. For a GOP of size S, let F_0 denote the key frame, $F_1, F_2, \ldots, F_{S-1}$ the WZ frames, and F_S the key frame of the next GOP.

To perform GOP length decision, our proposed algorithm operates as follows:

Initially, set $S = 1$.

While $S \leq S_{\max}$ *do*:

If $S == 1$, *go to step (e), otherwise*:

(a) *Interpolate between* F_0 *and* F_S

The interpolated frame serves as an estimate of the side information available at the decoder during the decoding process of the WZ frame $F_{\lfloor S/2 \rfloor}$, located at half-distance between F_0 and F_S.

Since motion estimation is not allowed at the encoder for complexity reasons, average interpolation [8] can be used to estimate the side information that will be available at the decoder.

(b) *Estimate the average bit rate* $R_{\lfloor S/2 \rfloor}$

Given the WZ frame $F_{\lfloor S/2 \rfloor}$ and its corresponding side information (estimate), the encoder determines its lower compression bound, and consequently, its compression rate, as explained in the previous section. The computation of $R_{\lfloor S/2 \rfloor}$ becomes straightforward.

(c) *Compute* $\text{PSNR}_{\lfloor S/2 \rfloor}$

Given the WZ frame $F_{\lfloor S/2 \rfloor}$ and its corresponding side information (estimate), the encoder can determine an estimate $F'_{\lfloor S/2 \rfloor}$ of the decoded frame at the receiver by first quantizing the WZ frame, and then reconstructing an 8-bit version using the available side information. The PSNR is then computed using $F_{\lfloor S/2 \rfloor}$ and $F'_{\lfloor S/2 \rfloor}$.

(d) *Repeat steps (a) to (c) until rate and PSNR estimates are obtained for all the frames of the GOP*

However, instead of interpolating between F_0 and F_S in step (a), the frames F_0 and $F'_{\lfloor S/2 \rfloor}$ are first used to generate a side information estimate for the frame located at half distance between the two, and the same process in steps (b) to (d) is repeated. Then, a similar procedure is performed in the second half of the GOP, using the frames $F'_{\lfloor S/2 \rfloor}$ and F_S. Frame $F'_{\lfloor S/2 \rfloor}$ is used instead of $F_{\lfloor S/2 \rfloor}$ because the former better estimates the frame that will be available at the decoder side since the latter is not known by the decoder.

(e) *Estimate the average rate and PSNR obtained with a GOP of size* S, *respectively, defined as*

$$R_{\text{av}}^S = \frac{1}{S}\sum_{j=0}^{S-1} R_j, \qquad \text{PSNR}_{\text{av}}^S = \frac{1}{S}\sum_{j=0}^{S-1} \text{PSNR}_j. \tag{2}$$

(f) *Determine* $\lambda_{\text{av}}^S = \text{PSNR}_{\text{av}}^S / R_{\text{av}}^S$

This represents the average PSNR per average unit bit rate estimated for a GOP of size S.

(g) $S = S + 1$.

Intuitively, the best *R-D* performance is obtained by maximizing the average PSNR per unit bit rate. As a result, the system decides the GOP length L as

$$L = \arg\max_{k=1,2,\ldots,S_{\max}} (\lambda_{\text{av}}^k). \tag{3}$$

In other words, if the system determines that the average PSNR per unit bit rate obtained by WZ coding, for different GOP lengths ($S > 1$), is lower than the one obtained with H.264 intracoding ($S = 1$), the system switches to H.264 intracoding mode ($L = 1$) and an H.264 I-frame is then transmitted. Otherwise ($L > 1$), an H.264 intracoded key frame is transmitted, followed by $L - 1$ WZ frames. Furthermore, since motion-compensated interpolation yields better side information compared to average interpolation, in general, the decoder is expected to perform better than estimated at the encoder side. This procedure is repeated at the beginning of every GOP and thus, the GOP length is dynamically varied along the sequence, in order to optimize the overall performance.

Figure 2 presents the algorithm above as a flow chart diagram, and Pseudocode 1 shows the Pseudocode of the recursive procedure R_PSNR_estimations(i, len) used to estimate the rate and PSNR for all the frames in a GOP of size S, where i is the time index of the first frame in the GOP, and *len* is the time interval between the frame at index i and the next frame used during the interpolation process (initially, *len* = S).

In general, a constant quality is desired along the sequence, since big fluctuations in PSNR yield undesirable visual effects. For this reason, the encoder determines the rate for the H.264 intracoded frames in such a way to obtain a near-constant PSNR in the GOP. This can be done by one of several techniques:

(i) Using predefined performance tables that determine H.264 rate-distortion relationships.

(ii) Using an analytical model for H.264 rate-distortion performance. This allows the system to avoid extensive table search as in the previous technique. However, it is important in this case to have an accurate, generalized *R-D* model [29].

(iii) Trial and error: the system tries several coding rates for each H.264 intracoded frame, and determines the PSNR for each case. Then, the rate that yields a PSNR closer to the one of neighboring frames is chosen. This method is more accurate than the previous ones. However, it can result in significant delay and increased encoder complexity.

The GOP size control algorithm can be further simplified by assuming a constant PSNR for all the frames. In this case, instead of maximizing the average PSNR per unit bit rate, (3) reduces to minimizing the average bit rate per frame over all possible GOP sizes. As a result, the simplified GOP size control algorithm operates as follows:

Initially, set $S = 1$.

While $S \leq S_{\max}$ *do*:

If $S == 1$, *go to step (d), otherwise*:

(a) *Interpolate from* F_0 *and* F_S

(b) *Estimate the average bit rate* $R_{\lfloor S/2 \rfloor}$

(c) *Repeat steps (a) and (b) until a rate estimate is obtained for all the frames in the GOP (replace frames* F_0 *and* F_S *in step (a) with the corresponding frames as previously explained in step (d) of the initial algorithm).*

(d) *Determine* $R_{\text{av}}^S = (1/S)\sum_{j=0}^{S-1} R_j$

(e) $S = S + 1$

Finally, the system decides the GOP length L as

$$L = \arg\min_{k=1,2,\ldots,S_{\max}} (R_{\text{av}}^k). \tag{4}$$

This allows the system to avoid estimating the PSNR for each frame, and the average PSNR per unit bit rate for each GOP size.

4. Complexity Analysis

While the aim of DVC is mainly permitting the design of low-complexity encoders, our GOP size selection algorithms incur additional encoding complexity. In this section, we present an analysis of the proposed algorithms computational load, and we compare our dynamic algorithms with the one presented in [17].

Table 1 presents an estimation of the necessary number of additional operations (OPs) incurred by each iteration (for each frame in the GOP, for every GOP size k, $1 \leq k \leq S_{\max}$) in the initial (nonsimplified) GOP size control algorithm. $P \times Q$ represent the frame dimensions, and M the quantization parameter for WZ frames.

Consider for example the number of operations performed to compute the PSNR. The calculation of the PSNR consists of first computing the mean square error (MSE), taking its inverse, multiplying it with a constant, computing the log of the result, and finally multiplying it by 10. The square error between two pixel values requires one addition operation and one multiplication. To compute the MSE, PQ square errors are first computed (which results in PQ additions and PQ multiplications) and summed together ($PQ - 1$ additions). The final result is then divided by PQ. As a result, $2PQ - 1$ additions, $PQ + 4$ multiplications, and a log operation are performed in order to obtain the PSNR. A similar analysis was performed on the other operations involved in the GOP size control algorithm, and the results are summarized in Table 1.

The total number of operations is roughly obtained by summing the elements of the last row in the table, which results in $12PQ + (8 \times 2^{2M}) + 3$ operations. Since, in our codec, the maximum value for M is 4, and by assuming QCIF video sequences ($P \times Q = 144 \times 176$), the total number of operations becomes 306,179 OPs. In the simplified algorithm, the computation of the PSNR is not performed. Thus, a reduction of $3PQ + 3$ operations is obtained, which yields a total number of 230,144 OPs.

A similar study was performed on the algorithm presented in [17], where four different metrics were used: the difference of histograms (DH), the histogram of difference (HD), the block histogram difference (BHD) and the block variance difference (BVD). Given the parameters specified in [17], we obtain 50,784 OPs for DH, 50,736 OPs for HD, 60,191 OPs for BHD, and 161,567 OPs for BVD, which results in a total of 323,278 OPs.

Even though the computational load of our initial algorithm and the one of the algorithm in [17] are almost similar, it is important to note that our algorithm presents the additional property of estimating the necessary bit rate without the need for a feedback channel, and the possibility to take into account channel impairments based on our study in [5, 6, 14, 15]. On the other hand, a reduced complexity of approximately 25% can be obtained with our simplified algorithm, without a significant loss in *R-D* performance as will be shown in the next section.

5. Experimental Results

In our simulations, we consider three different QCIF video sequences with different levels of motion: Foreman, Grandmother, and Salesman, sampled at a rate of 30 frames per second. The first 100 frames from each sequence are first encoded using a WZ codec with fixed GOP sizes ranging from 1 to 5. For the case where the GOP size is 1, all frames are H.264 intracoded, whereas for the other cases, only the first frame (key frame) from each GOP is H.264 intracoded, while the remaining ones are WZ-coded. H.264 coding is performed using JM FREXt reference software, version 13.2, with baseline profile. The results are then compared with the case where a WZ codec with a dynamically varying GOP

```
procedure R_PSNR_estimations(i, len) {
  If len < 2
    Return                          // End of the recursive function calls
  Else
    a = i, b = i + len              // a and b are the time indices
                                    // of the frames used during the
                                    // interpolation process.
    d = ⌊(b − a)/2⌋                 // time interval from a to the
                                    // frame at mid distance
                                    // between a and b.
    SI_(a+d) = Interpolate(a, b)    // Perform average interpolation
                                    // between frames at time
                                    // indices a and b.
    Q_(a+d) = QuantizeFrame(F_(a+d))
    R_(a+d) = EstimateBitrate(Q_(a+d))   // estimate the bitrate as
                                         // explained in Section 2.
    F'_(a+d) = Reconstruct(Q_(a+d), SI_(a+d))   // reconstruct the
                                                // quantized WZ frame
                                                // given the estimated
                                                // side information.
    PSNR_(a+d) = ComputePSNR(F'_(a+d), F_(a+d))   // Compute the PSNR
                                                  // of the reconstructed
                                                  // frame.
    R_PSNR_estimations(i, d);       // Recursive function call using
                                    // the first half of the GOP.
    R_PSNR_estimations(i + d, len − d)   // Recursive function call using
                                         // the second half of the GOP.
  End If
}
```

Pseudocode 1: Pseudocode of the recursive procedure R_PSNR_estimations used to estimate the rate and PSNR for all the frames in a GOP.

Table 1: Number of additional operations per frame for each GOP size k, $1 \leq k \leq S_{\max}$, incurred by the proposed GOP size control algorithm.

Operation	Average Interpolation	Reconstruction	Quantization (WZ and SI)	Estimation of the α parameter	PSNR estimation	Rate (R) estimation
Additions	PQ	PQ		$2PQ - 1$	$2PQ - 1$	$1 \times 2^{2M} - 1$
Multiplications	PQ			$PQ + 1$	$PQ + 4$	$5 \times 2^{2M} + 1$
Log or Exp					1	2×2^{2M}
Comparisons		PQ				
Logical AND			$2PQ$			
TOTAL	$2PQ$	$2PQ$	$2PQ$	$3PQ$	$3PQ + 3$	8×2^{2M}

size is used. The GOP size is determined using our proposed algorithms as explained in Section 3, with $S_{\max}$ set to 5.

In Figure 3, we show the rate and PSNR variations along the Grandmother sequence for $M = 4$, and the quantization parameter of the H.264 intraframes QP = 25, obtained using a WZ codec with a fixed GOP size set to 3, with and without a feedback channel (FC). It can be noticed that the rate estimated without FC exceeds the rate obtained using FC most of the time. As a result, WZ frames are correctly decoded in both cases and the reconstructed output is the same. However, in rare situations (e.g., in frame 41), the encoder underestimates the rate needed for correctly decoding a WZ frame, which yields a degraded quality at the decoder output. For an average bit rate of 697 kbps

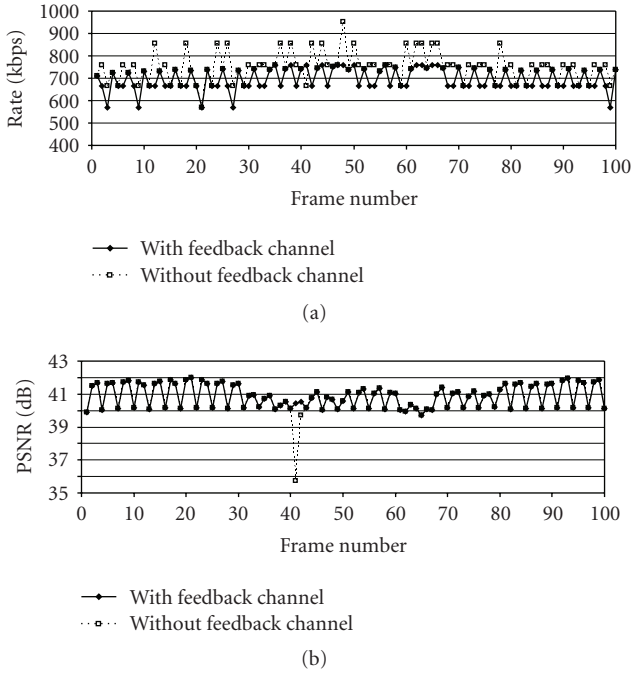

Figure 3: Rate (a) and PSNR (b) variations along the Grandmother sequence using a WZ codec with a GOP size = 3, $M = 4$, and QP = 25.

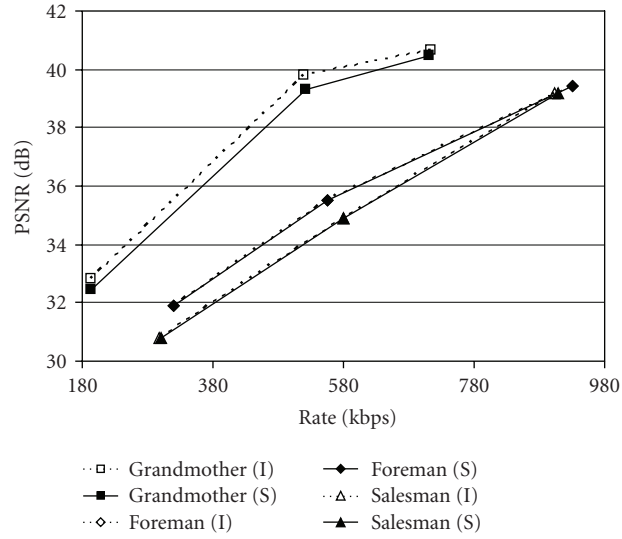

Figure 4: Average *R-D* curves obtained using the initial (I) and simplified (S) adaptive GOP size control algorithms.

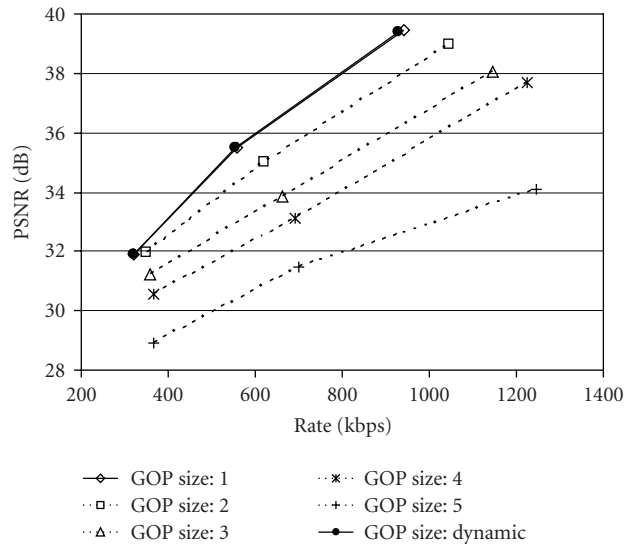

Figure 5: Average *R-D* curves for the Foreman sequence using a WZ codec with fixed and variable GOP sizes.

obtained with the feedback channel, an average rate excess of 50 kbps is observed in the case where the feedback channel is suppressed. In other words, for applications where the feedback channel is not suitable (e.g., in video broadcasting), the cost to be paid (in terms of bit rate) for the suppression of the return channel is approximately 7%.

In Figure 4, we show the average *R-D* curves obtained for the three sequences using both the initial (I) and simplified (S) algorithms. The rate and PSNR are averaged over all the sequence (key and WZ frames). Different rate points are obtained by varying the quantization parameter M for the WZ frames. As for the quantization parameter (QP) of the H.264 intraframes, it is chosen in such a way to permit a near-constant decoding quality in the output video sequence, using the approach presented in [17]. It can be clearly seen that, for the Salesman and Foreman sequences, both curves overlap (both algorithms have similar performance), whereas a negligible loss that does not exceed 0.45 dB is observed with the Grandmother by using the simplified algorithm.

Figures 5 to 7 show the average *R-D* performance for the Foreman, Grandmother, and Salesman sequences, respectively, obtained with the initial (nonsimplified) algorithm. In Figure 5 (Foreman), we notice that for the case of a fixed GOP size, the performance decreases as the GOP size increases. The best performance is thus obtained when all frames are intracoded. This is due to the high motion in this sequence, which yields less accurate side information when the key frames are further apart. A similar effect has been noticed in [16] where key frames were encoded using an H.263+ video codec. However, when the GOP size is dynamically varied along the sequence, a gain of 10 to 12 kbps is obtained compared to H.264 intracoding. For sequences with lower motion levels, different results are observed. It can be seen in Figures 6 and 7 that the best system performance is obtained with a GOP of size 3 (for the fixed-GOP case). Our proposed system outperforms both H.264 intracoding and fixed-GOP WZ coding in most cases. For example, for the Grandmother sequence at 520 kbps, a gain of 3 dB is observed with respect to the H.264 intracodec and 0.8 dB with respect to the WZ codec with a GOP size of 3. Similarly, for the Salesman sequence at 580 kbps, our proposed algorithm outperforms the H.264 intra codec and the WZ codec with a GOP size of 3 by 1 dB and 0.1 dB, respectively. However, a performance loss of 0.4 dB

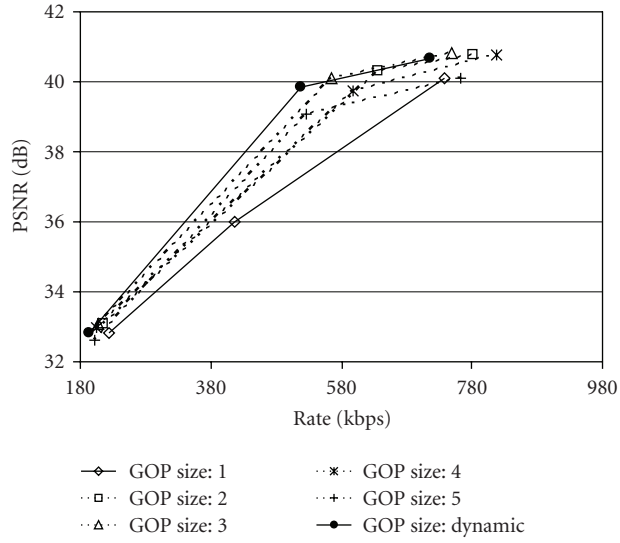

Figure 6: Average *R-D* curves for the Grandmother sequence using a WZ codec with fixed and variable GOP sizes.

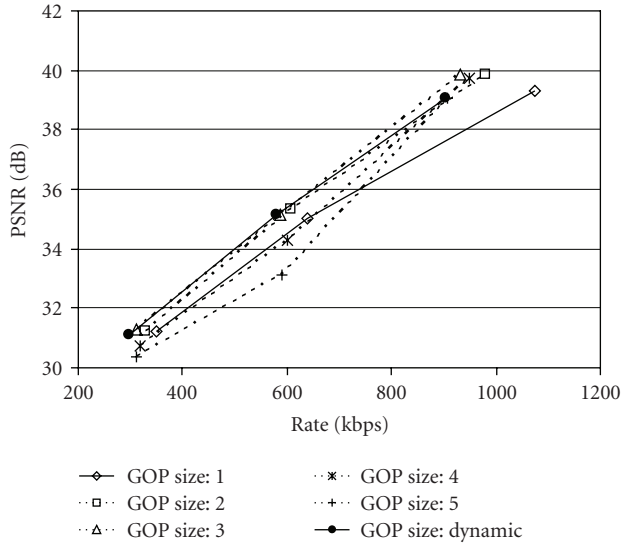

Figure 7: Average *R-D* curves for the Salesman sequence using a WZ codec with fixed and variable GOP sizes.

can be observed with the Salesman sequence at 900 kbps using a dynamic GOP size, compared to the fixed-GOP (size = 3) WZ codec. In fact, this is due to a significant mismatch between the side information available at the encoder (estimated using average interpolation) and the one available at the decoder (obtained by motion-compensated interpolation).

Figures 8 and 9 show the rate and PSNR variations along the Salesman sequence for $M = 4$ and $M = 2$, respectively, for the case where the rate estimation is done at the encoder using average-interpolated side information, and for the case where this estimation is performed at the decoder using the side information obtained by motion-compensated interpolation. The source coding rate is the one estimated by the encoder, whereas the real PSNR is the one obtained after the decoding process and thus, the corresponding curves are shown as solid lines. On the other hand, decoder-estimated bit rate and encoder-estimated PSNR (dotted curves) are shown only to analyze the system's behavior at both (encoder and decoder) sides. It can be seen that, in some regions (e.g., frames 61 to 64 and frames 71 to 74 with $M = 4$), the encoder underestimates the rate necessary for the decoding of WZ frames, which yields a very high bit error rate at the turbo decoder output. As a result, the reconstruction function of the WZ codec cannot yield a reliable output and thus, a significant performance loss is observed in these regions, which greatly affects the average system performance as shown in Figure 7. However, such estimation errors rarely occur. As it can be clearly seen in Figures 8 and 9, the encoder estimations are accurate most of the time when $M = 4$, and all the time when $M = 2$. Similar results were observed with the other sequences for different values of M.

Table 2 shows the percentage of GOP sizes obtained for each of the sequences using the proposed adaptive GOP size control algorithm (nonsimplified), for different values of the

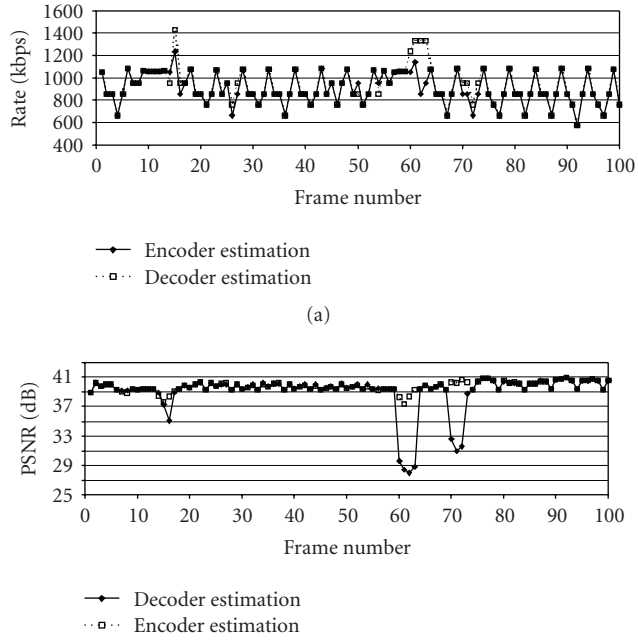

Figure 8: Rate (a) and PSNR (b) variations along the Salesman sequence using a WZ codec with the proposed GOP size control algorithm for $M = 4$.

WZ quantization parameter M. For the Foreman sequence, when $M = 1$, 100% of the GOPs are of size 1. In other words, the system switches to H.264 intracoding mode all the time, and no frame in the sequence is WZ-encoded. For $M = 2$ and 4, most of the GOPs are of size 1, while the maximum GOP size does not exceed 3. This explains the reason behind the similar performance between the H.264

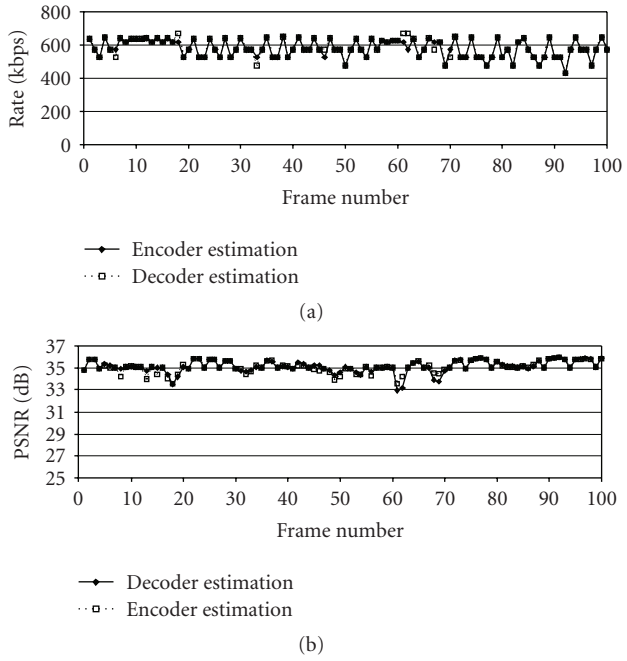

Figure 9: Rate (a) and PSNR (b) variations along the Salesman sequence using a WZ codec with the proposed GOP size control algorithm for $M = 2$.

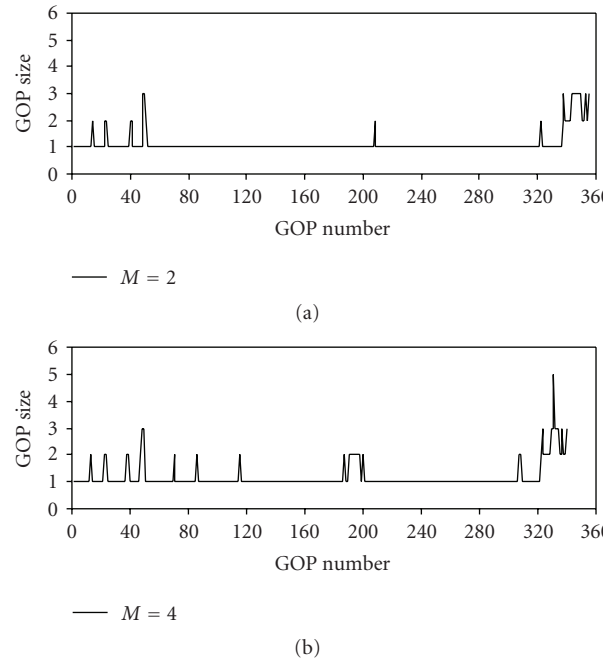

Figure 10: GOP size variations along the complete Foreman sequence (400 frames) for $M = 2$ (a) and $M = 4$ (b).

Table 2: Percentage of GOP sizes used in each sequence.

Gop size			1	2	3	4	5
Foreman	$M = 1$		100	0	0	0	0
	$M = 2$		91.1	6.7	2.2	0	0
	$M = 4$		87.4	10.3	2.3	0	0
Grandmother	$M = 1$		0	4	20	36	40
	$M = 2$	%	0	4.2	16.7	37.5	41.6
	$M = 4$		23.1	33.3	20.5	2.6	20.5
Salesman	$M = 1$		3.7	22.2	18.5	0	55.6
	$M = 2$		18.2	12.1	39.4	0	30.3
	$M = 4$		22.2	7.4	3.7	0	66.7

and our proposed WZ codec for the Foreman sequence, as shown in Figure 5. More GOP size variations can be noticed with the two other sequences. For the Grandmother sequence with $M = 1$ and $M = 2$, the system is always in WZ coding mode. In other words, not any GOP is of size 1 and only key frames are intracoded, since the WZ coding outperforms H.264 intracoding in this case, according to the encoder estimations.

The Foreman sequence is characterized by very high motion levels, especially in its second half. In such regions with high motion, H.264 intracoding usually outperforms WZ coding. For this reason, in order to analyze our system's performance with high motion video, we encoded the complete Foreman sequence (400 frames) and the results are reported in Figure 10. This figure shows the GOP size variations along the sequence for $M = 2$ and $M = 4$. Long runs of consecutive GOP sizes equal to 1 can be noticed in high motion areas, which indicate that the system has switched to H.264 intracoding mode in these regions. As a result, the *R-D* performance for the Foreman sequence is slightly better than the one obtained with a pure H.264 intracoder, as was also noticed in Figure 5.

6. Conclusion and Future Work

This paper presents simple algorithms that dynamically adapt the GOP size for a distributed Wyner-Ziv video codec, depending on the content of the video scene to be encoded. Based on H.264 intracoding for key frames, the system relies on theoretical calculations to estimate the bitrate necessary to successfully decode Wyner-Ziv frames without the need for a feedback channel, which makes it suitable for broadcasting applications. Automatic mode selection allows the system to switch between H.264 intracoding and WZ coding modes in order to optimize the overall system performance. Simulation results show an average gain that can reach 3 dB compared to an H.264 intracodec and 0.8 dB compared to a WZ codec with a fixed GOP size.

As a future work, authors will focus on taking into account the transmission channel conditions in the GOP size control algorithm, and implementing the system in a more realistic network environment with multiple users, based on their previous research in [14, 15]. Further research may consider dynamically varying the quantization parameters (QP for key frames and M for WZ frames) and using advanced interpolation techniques to improve the overall performance.

Acknowledgment

This work has been supported by a research grant from the Lebanese National Council for Scientific Research (LNCSR).

References

[1] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proceedings of Data Compression Conference (DCC '02)*, pp. 252–261, Snowbird, Utah, USA, April 2002.

[2] J. Garcia-Frias and Y. Zhao, "Near-Shannon/Slepian-Wolf performance for unknown correlated sources over AWGN channels," *IEEE Transactions on Communications*, vol. 53, no. 4, pp. 555–559, 2005.

[3] S. S. Pradhan and K. Ramchandran, "Distributed source coding: symmetric rates and applications to sensor networks," in *Proceedings of Data Compression Conference (DCC '00)*, pp. 363–372, Snowbird, Utah, USA, March 2000.

[4] J. Farah, C. Yaacoub, N. Rachkidy, and F. Marx, "Binary and non-binary turbo codes for the compression of correlated sources transmitted through error-prone channels," in *Proceedings of the 4th International Symposium on Turbo Codes and the 6th ITG Conference on Source and Channel Coding*, Munich, Germany, April 2006.

[5] J. Farah, C. Yaacoub, F. Marx, and B. Pesquet-Popescu, "Performance analysis of a distributed video coding system—application to broadcasting over an error-prone channel," in *Proceedings of the 15th European Signal Processing Conference (EUSIPCO '07)*, Poznan, Poland, September 2007.

[6] C. Yaacoub, J. Farah, and B. Pesquet-Popescu, "Joint source-channel Wyner-Ziv coding in wireless video sensor networks," in *Proceedings of IEEE International Symposium on Signal Processing and Information Technology (ISSPIT '07)*, pp. 225–228, Giza, Egypt, December 2007.

[7] R. Puri and K. Ramchandran, "PRISM: a new robust video coding architecture based on distributed compression principles," in *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, Ill, USA, October 2002.

[8] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 240–244, Pacific Groove, Calif, USA, November 2002.

[9] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform-domain Wyner-Ziv codec for video," in *Visual Communications and Image Processing*, vol. 5308 of *Proceedings of the SPIE*, pp. 520–528, San Jose, Calif, USA, January 2004.

[10] J. Ascenso, C. Brites, and F. Pereira, "Motion compensated refinement for low complexity pixel based distributed video coding," in *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '05)*, pp. 593–598, Como, Italy, September 2005.

[11] C. Brites, J. Ascenso, and F. Pereira, "Feedback channel in pixel domain Wyner-Ziv video coding: myths and realities," in *Proceedings of the 14th European Signal Processing Conference (EUSIPCO '06)*, Florence, Italy, September 2006.

[12] X. Artigas and L. Torres, "Improved signal reconstruction and return channel suppression in distributed video coding systems," in *Proceedings the 47th International Symposium on Electronics in Marine (Elmar '05)*, pp. 53–56, Zadar, Croatia, June 2005.

[13] M. Morbée, J. Prades-Nebot, A. Pižurica, and W. Philips, "Rate allocation algorithm for pixel-domain distributed video coding without feedback channel," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 1, pp. 521–524, Honolulu, Hawaii, USA, April 2007.

[14] C. Yaacoub, J. Farah, and B. Pesquet-Popescu, "A novel technique for practical implementation of pixel-domain Wyner-Ziv video coding in multi-user systems," in *Proceedings of the 16th European Signal Processing Conference (EUSIPCO '08)*, Lausanne, Switzerland, August 2008.

[15] C. Yaacoub, J. Farah, and B. Pesquet-Popescu, "Optimal rate allocation in multi-user Wyner-Ziv video coding systems with coded key frames," in *Proceedings of the 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '08)*, Cannes, France, September 2008.

[16] A. Aaron, E. Setton, and B. Girod, "Towards practical Wyner-Ziv coding of video," in *Proceedings of IEEE International Conference on Image Processing (ICIP '03)*, vol. 3, pp. 869–872, Barcelona, Spain, September 2003.

[17] J. Ascenso, C. Brites, and F. Pereira, "Content adaptive Wyner-Ziv video coding driven by motion activity," in *Proceedings of IEEE International Conference on Image Processing (ICIP '06)*, pp. 605–608, Atlanta, Ga, USA, October 2006.

[18] D. Kubasov, K. Lajnef, and C. Guillemot, "A hybrid encoder/decoder rate control for Wyner-Ziv video coding with a feedback channel," in *Proceedings of the 9th IEEE Workshop on Multimedia Signal Processing (MMSP '07)*, pp. 251–254, Crete, Greece, October 2007.

[19] C. Brites and F. Pereira, "Encoder rate control for transform domain Wyner-Ziv video coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP '07)*, vol. 2, pp. 5–8, San Antonio, Tex, USA, September-October 2007.

[20] Y. Tonomura, D. Shirai, T. Nakachi, and T. Fujii, "Optimal bit allocation for wavelet-based distributed video coding," in *Proceedings of the 8th IEEE International Symposium on Multimedia (ISM '06)*, pp. 442–448, San Diego, Calif, USA, December 2006.

[21] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[22] D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder ," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[23] ITU-T and ISO/IEC JTC1, "Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264—ISO/IEC 14496-10 AVC, 2003.

[24] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes. 1," in *Proceedings of IEEE International Conference on Communications (ICC '93)*, vol. 2, pp. 1064–1070, Geneva, Switzerland, May 1993.

[25] B. Sklar, "A primer on turbo code concepts," *IEEE Communications Magazine*, vol. 35, no. 12, pp. 94–101, 1997.

[26] C. Berrou, "Turbo codes: some simple ideas for efficient communications," in *Proceedings of the 7th International Workshop on DSP Techniques for Space Communications*, Sesimbra, Portugal, October 2001.

[27] D. Divsalar and F. Pollara, "Multiple turbo codes," in *Proceedings of IEEE Military Communications Conference (MILCOM '95)*, vol. 1, pp. 279–285, San Diego, Calif, USA, November 1995.

[28] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Transactions on Telecommunications*, vol. 8, no. 2, pp. 119–125, 1997.

[29] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2009, Article ID 141986, 8 pages
doi:10.1155/2009/141986

Research Article

Intra-Skip in Inter-Frame Coding of H.264/AVC

Hui Su

Communication System Group, Chalmers University of Technology, 412 96 Goteborg, Sweden

Correspondence should be addressed to Hui Su, luminixsu@gmail.com

Received 16 June 2008; Revised 15 January 2009; Accepted 2 February 2009

Recommended by Ennio Gambi

In inter-frame coding of H.264/AVC standard, not only seven inter-partition modes but also intra-modes are taken into account for seeking the best coding mode so as to maintain higher encoding efficiency by sacrificing the speed of H.264/AVC encoder. Aiming at intra-skip, this paper proposes a novel mathematical model for intra-skip in inter-frame coding to alleviate the complexity of the process; the model provides remarkable performance increment by cutting down encoding time while accompanying very minor bitrate increase. The critical advantage of this proposed scheme most emphasized on is that it can optimize H.264 encoder in conjunction with any proposed fast inter- and intra-methods which are focusing on inter-partition mode decision, motion search algorithms, and fast intra-algorithms.

1. Introduction

The latest established video compression standard H.264/AVC [1] is recognized to be a major international standard in the next generation video compression techniques, because of higher coding efficiency and better performance in various environments, compared to previous video coding standards. In the inter-frame coding of H.264/AVC standard, intra-modes are also calculated, for seeking the best coding mode in order to obtain low bitrate and high fidelity, also for images that would be better encoded by intra-mode prediction, such as background images that frequently change in a video sequence [2]. Cheng et al. pointed out in [3] that the encoder speed would be accelerated significantly and dramatically if all computations of intra-modes were skipped in inter-frame coding, which means only seven inter-modes are taken into consideration, and all intra-modes are excluded in inter-frame coding. But the experimental results show that this choice would also result in picture fidelity deterioration and high bitrate. Lee and Jeon put forward a method [4] by which designated partition blocks are intra-skipped and also present a new mathematic model for inter-frame coding. The performance increment is dramatic while in some cases this method will miss some blocks that should have passed intra-predictions in inter-frame coding.

Afterwards, Cheng et al. [5] put forth an advanced method about P frame based on the idea that the decision for intra-skip is generated by three fixed adjacent blocks. And the speed of the encoder with this method is obviously accelerated in P frames. Pan et al. [6] introduced a new scheme mainly focusing on inter-frame. Recently, Kim et al. [7] proposed a simple method which is to adopt minimum RDcost of adjacent blocks as the threshold for intra-skip, and therefore intra-skip is reintroduced in the coding process.

The remainder of this paper is going to analyze the whole procedure of the inter-frame coding first and then to present a new mathematic model for inter-frame coding not only aiming at P frames, but also addressing B frames, in Section 2. The results of experiments and comparison to some well-known algorithms are presented in Section 3. Finally, in Section 4, the conclusion and discussion are presented.

2. Inter-Frame Coding Procedure

2.1. Prediction Modes. To achieve higher coding efficiency, H.264/AVC employs rate distortion optimization (RDO) [1, 8] to seek the best coding result in terms of maximizing image quality and minimizing resulting transmission data bits. That is to say, in order to achieve rate distortion

optimization, the encoder has to encode the video sequence by exhaustively testing all the possible mode combinations, including different intra- and inter-prediction modes, for each block that minimizes the difference between the original and its reconstruction to be encoded. As a result, due to the dramatically increased computation load of sequence coding, practical applications of an H.264 encoder are limited at large especially for real time visual communication.

The whole procedure of inter- and intra-modes in inter-frame coding is comprised of three parts. First, calculate the mincost of intra-partition modes. Second, figure out the mincost of intra-modes. And at last, compare mincost of inter-partition modes with mincost of intra-modes to decide final coding mode. If mincost of intra-mode is less than mincost of inter-mode, the final coding mode will be intra- and vice versa. In the following parts, this procedure is specified.

2.2. Intra-Modes in Inter-Frame Coding. In intra-modes, a prediction block is formed based on the previously encoded and reconstructed blocks and is subtracted from the current block prior to encoding. That means intra-modes only exploit spatial redundancies within the same frame instead of previously encoded frames as in inter-modes. The prediction mode for each block that minimizes the difference (RDcost) between original block and its prediction is selected as the best intra-mode.

In inter-frame coding, intra-modes are also taken into consideration, including intra 4 × 4, Intra 16 × 16 and intra 8 × 8 (optional since JM9.3). intra 16 × 16 has four directional predictions (Intra_16 × 16_Vertical, Intra_16 × 16_Horizontal, Intra_16 × 16_DC, Intra_16× 16_Plane) while intra 4 × 4 has nine different directional predictions (Intra_4 × 4_Vertical, Intra_4 × 4_Horizontal, Intra_4 × 4_Diagonal_Down_Left, Intra_4 × 4_Diagonal_Down_Right, Intra_4 × 4_Vertical_Right, Intra_4 × 4_Horizontal_Down, Intra_4 × 4_Vertical_Left, Intra_4 × 4_Horizontal_Up, Intra_4 × 4_DC).

2.3. Whole Procedure of Inter-Frame Coding. The entire flow inter-frame coding is shown in Figure 1. First, determine whether initial SKIP mode is adopted, which is different from SKIP mode with no coefficients later. If not, calculate seven inter-mode predictions to seek the best inter-mode and consider the corresponding cost as BEST_INTER_COST. Then, calculate intra-mode predictions to get the BEST_INTRA_COST. Finally, compare BEST_INTER_COST and BEST_INTRA_COST to obtain the best mode among all possible modes in inter-frame coding.

2.4. The Analysis and Effect of Intra-Skip in Inter-Frame Coding. The purpose of using intra-modes in inter-frame coding is to improve image fidelity and to possibly provide more precise mode prediction so as to reduce the bitrates of coded sequences by sacrificing coding speed. Thus, whether the intra-modes are frequently adopted in inter-frame coding may become an issue.

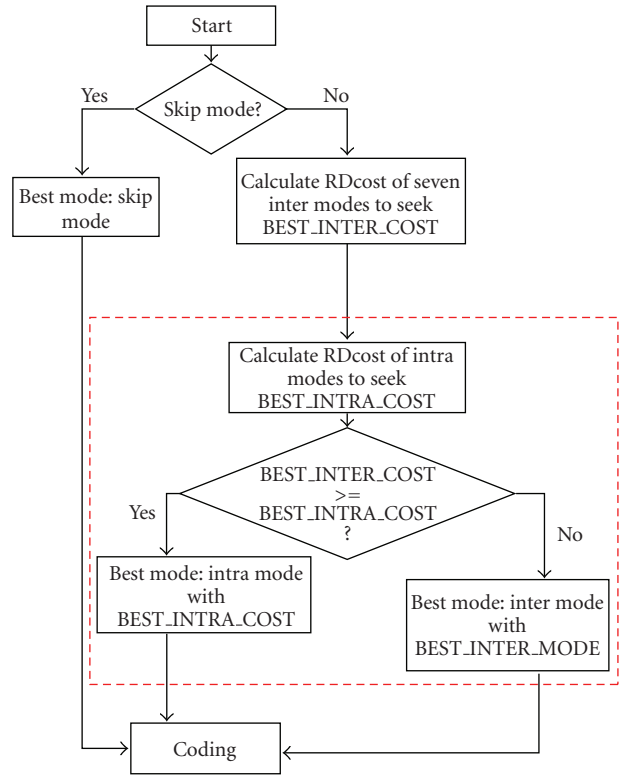

Figure 1: Flowchart of inter-frame coding.

Admittedly, a considerable amount of transmission bits is saved on the condition that RDcost of best intra-mode is lower than RDcost of best inter-mode. This means that, as the RDcost of best inter-mode is larger than the RDcost of best intra-mode, it possibly indicates that the current block is in a rapid or median motion. As a matter of fact, if the background images are changing, the complicated procedure of greatly changing match-block search in motion compensation will cost a lot of bits for motion vector predicted (MVP), and a nonoptimized prediction will require a lot of extra transmission bits for the residual [9, 10]. As a result, in this situation, intra-modes in inter-frame coding is a better choice for encoding current block. However, taking more possible modes into account in inter-frame coding sacrifices encoder's coding speed [11, 12]. Hence, the possibility that intra-modes are unnecessary to be calculated and could be skipped is largely determined by the motion of current image to be encoded.

In addition, the procedure of intra-skip is irrelevant to the procedures of motion search and seven inter-mode and nine intra-mode predictions. Hence, intra-skip can accomplish higher performance increment together with any fast motion search algorithm and any fast inter-/intra-algorithm [9–17].

Consequently, performing intra-skip or not, which largely depends on the motion range of objects and background in sequences, also plays a key role in inter-frame coding compared to the procedure of motion search and

block match [13–15]. The purpose of this paper is to focus on intra-skip and present a method to decide whether intra-modes are to be adopted or skipped in inter-frame coding by early estimation and detection.

2.5. The Proposed Mathematical Model for Intra-Skip. Since the adjacent blocks are highly correlated with the current block to be encoded, the information of encoded blocks is essential for current block. Consequently, whether encoded block was intra-skip or not is a substantial point to indicate the possibility of intra-skip for current block. In addition, according to the analysis of intra-skip in Section 2.4, it can be noted that the values of RDcost of best inter-mode largely represent the changing speed of image background and motion ranges of objects in sequences. As a result, this paper adopts the values of encoded blocks' BEST_INTER_COSTs (RDcost of best inter-mode) that are assigned to different weighted coefficients as multipliers due to the encoded blocks' various distances from current block for predetermining whether the time-exhaustive process of intra-mode predictions is necessary or not for current block. A mathematical model is proposed in order to correctly skip the blocks' intra-mode predictions in inter-frame coding. The model is provided as follows:

$$G(K) = \underbrace{\lambda_1 J_1 + \lambda_2 J_2 + \lambda_3 J_3 + \lambda_4 J_4 + \lambda_5 J_5 + \cdots + \lambda_K J_K}_{K \text{ reference}}$$
$$\max(\lambda_1, \lambda_2, \ldots, \lambda_K) \le 1,$$
$$\min(\lambda_1, \lambda_2, \ldots, \lambda_K) \ge -1, \quad (1)$$
$$\text{sum}(\lambda_1, \lambda_2, \ldots, \lambda_K) \le (1 + 0.5),$$
$$\lambda_n = f(a_0, q, n).$$

In the model, J_1 is latest encoded block's RDcost of the best inter-mode and K is the number of reference blocks. J_K denotes the RDcost of the best inter-mode in the prior Kth encoded blocks. For example, J_1 is the latest block's RDcost of inter-, which is encoded prior to current block, and J_2 is prior to J_1. In the model, λ_1 denotes the weighted coefficient of latest encoded block's RDcost of inter-. And the weighted-coefficient λ_K is the prior Kth block's weighted coefficient of the RDcost of that block, the values of which are decided by the weighted-coefficient function $f(a_0, q, n)$. This function that plays a key role in this model can be an arithmetic/geometric progression. $G(K)$, which is adopted for intra-skip decision, is the weighted average RDcost of K reference blocks. The constraint conditions in this model are weighted coefficients λ_K, which are provided based on the experimental results of more than thirty sequences (here 0.5 covers most of situations. In most cases according to experimental results, it ranges within (0.2, 0.3)). The procedure of implementation is presented in Algorithm 1, and the proposed method is illustrated in Figure 2 (gray parts) compared to the original procedure in Figure 1.

In Step (2), all the J_i taken into consideration for $G(K)$ are the RDcost values obtained with best inter-mode rather than RDcost values obtained with best final coding mode.

BEGIN
(1) Initialize K (number of reference blocks) and then select a weighted-coefficient function $f(a_0, q, n)$.
(2) Get current block's RDcost of best inter-mode, BEST_INTER_COST.
(3) Calculate $G(K)$ by the mathematic model proposed above.
(4) Compare the value of $G(K)$ and the value of BEST_INTER_COST.
If $BEST\_INTER\_COST < G(K)$, skip intra-modes (red line in Figure 2). Otherwise, do intra-modes prediction and compare the value of *BEST_INTER_COST* with BEST_INTRA_COST to decide final coding mode.
(5) Encode the block and go to Step (2) for the next block.
END

Algorithm 1

2.6. Analysis of the Proposed Model. Most of advanced inter-coding algorithms conceived for speeding up the H.264 encoder are largely concentrating on the computation of the BEST_INTER_COST (Step (2)) because partition mode decision and motion search algorithm are exhaustively calculated in this step [9–18]. However, the proposed mathematic model is carried out after this step; consequently it can optimize the H.264 encoder in cooperation with any advanced fast partition modes and search algorithms. Hence, the speed of the encoder will be accelerated immensely if we adopt this mathematical model together with fast approaches for partition modes and search algorithms.

2.7. Weighted-Coefficient Function. The core of the proposed mathematical model is the weighted-coefficient function, which in a large sense should be optimized so as to gain higher performance and mistake less blocks that should have been coded in intra-modes, since the threshold of intra-skip is generated by the weighted-coefficient function. In this paper, we propose a geometric progression for this model:

$$G(K) = \sum_{i=1}^{K} \lambda_i J_i; \qquad \lambda_n = f(a_0, q, n),$$
$$F(\lambda) = \sum_{i=1}^{n} \lambda_i = a_0 + a_0 q + a_0 q^2 + a_0 q^3 + \cdots + a_0 q^n. \quad (2)$$

According to more than thirty sequences' experimental results obtained with various weighted coefficient functions, a statistic survey indicates that the best performance increment is derived from conditions set as follows:

$$a_0 = 0.5, \qquad F(\lambda) = 1.2, \qquad K = 4. \quad (3)$$

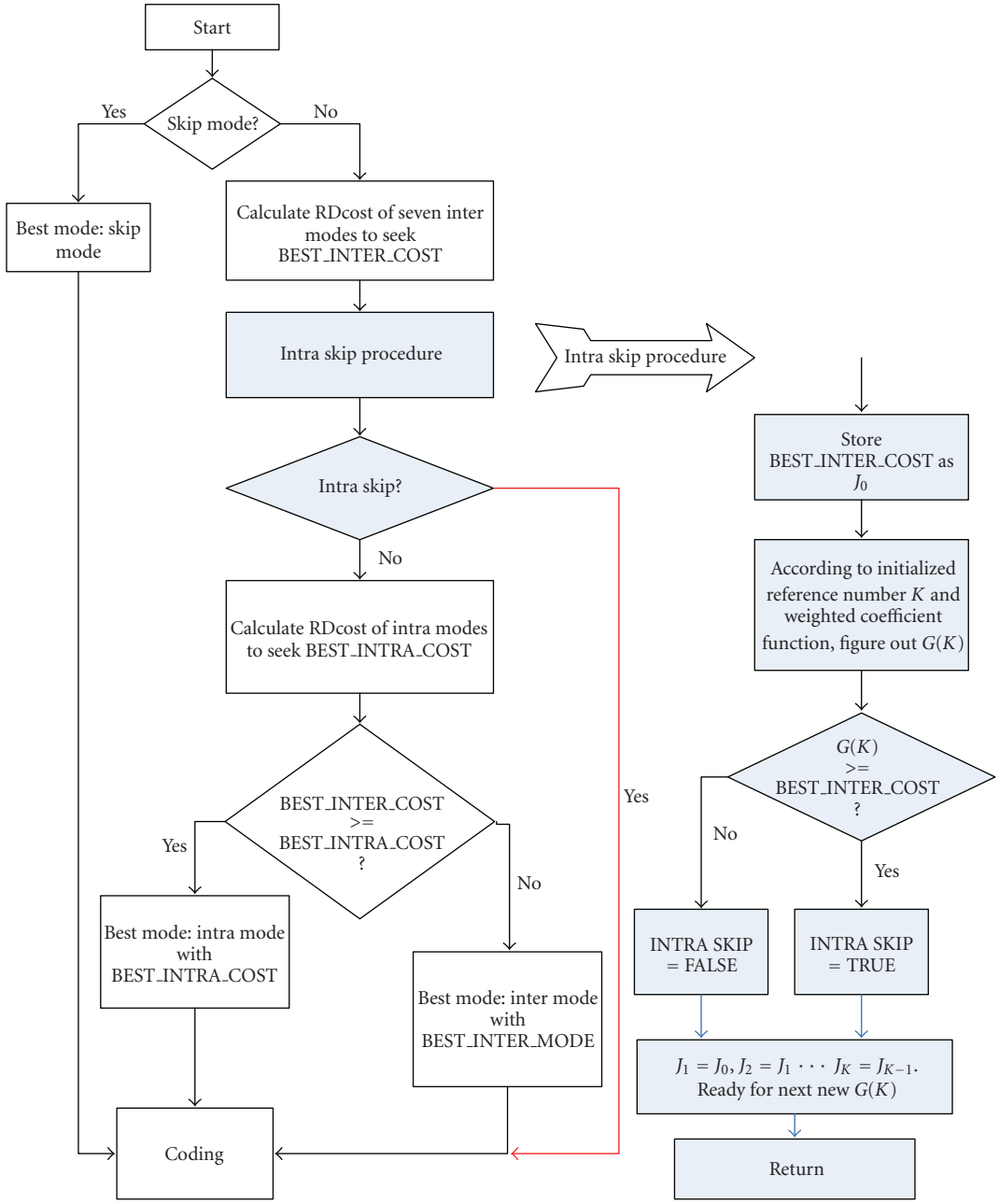

Figure 2: Flowchart of inter-frame coding with intra-skip.

They are right for a geometric progression in this model. According to Newton's Iteration Method of solving equation, the function can be expressed as follows:

$$\begin{aligned} 0.5 + 0.5 \times q + 0.5 \times q^2 + 0.5 \times q^3 &= 1.2, \\ f(q) = 0.5 \times q^3 + 0.5 \times q^2 + 0.5 \times q - 0.7 & \\ \Longrightarrow q_{K+1} = q_K - \frac{f(q_K)}{f'(q_K)} & \\ = q_K - \frac{0.5 \times {q_K}^3 + 0.5 \times {q_K}^2 + 0.5 \times q_K - 0.7}{1.5 \times {q_K}^2 + q_K + 0.5} & \\ \Longrightarrow q = 0.664643. & \end{aligned} \tag{4}$$

Therefore, $G(K) = \sum_{i=1}^{4} a_0 q^{i-1} J_i$; $G(K)$ is adopted as the self-adaptive threshold for intra-skip in the procedure of inter-frame coding.

3. Experimental Results

To verify the performance of the algorithm proposed in this paper, several common and typical QCIF (Foreman, Carphone, and Highway) and CIF (Paris, Mobile, and Bus) sequences are specified. Our experimental environment is based on JM10.1 [19], which is developed for H.264 reference, and the simulation environment of experiments is P4 1.7 G +256 M, VC+ +6.0+sp5 in Windows XP+sp2.

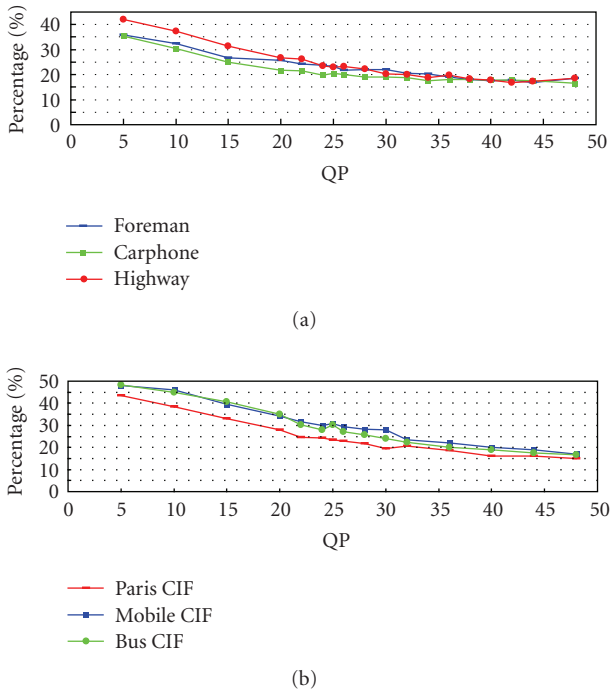

Figure 3: The speed acceleration percentage of standard H.264 encoder with mathematic model.

Table 1: Encoder parameters used in the experiments.

Use Hadamard	Yes	Frame rate	30 fps
Search range	16	Enable open GOP	Disable
NumRef frames	5	RD optimization	1
Symbol mode	CABAC	Context init method	1
Sequence type	IBPBP/IBBPBBP	Sequence format	QCIF/CIF

Experimental results are tested with the conditions indicated in Table 1, which strictly follow the simulation contexts suggested by JVT.

Figure 3, shows the speed increment by comparing the JM standard encoder combined with the mathematic model proposed in this paper to the original JM standard encoder. In this figure, the percentage of speed increment is defined as follows:

$$\text{PERCENTAGE}\,(\%) = \frac{\text{Time}_{\text{original}} - \text{Time}_{\text{new}}}{\text{Time}_{\text{original}}} \times 100\%, \quad (5)$$

where $\text{Time}_{\text{original}}$ is JM10.1 standard encoder's coding time and Time_{new} is the optimized encoder's coding time we proposed. From Figure 3, it can be seen that the percentage of whole encoder has accelerated by 35% (QCIF)/ 45% (CIF) as QP is 5 and about 25% as QP is 25. The trend is that the smaller QP is, the coding speed increment the encoder shows.

The percentage of missing blocks that should have passed intra-mode prediction is shown in Table 2. The percentage is the ratio of number of missed blocks over all blocks that need intra-calculations. From the table we can see that almost all blocks in inter-frame coding are intra-skipped correctly. In QCIF sequences, less than 0.206043% (maximum) is

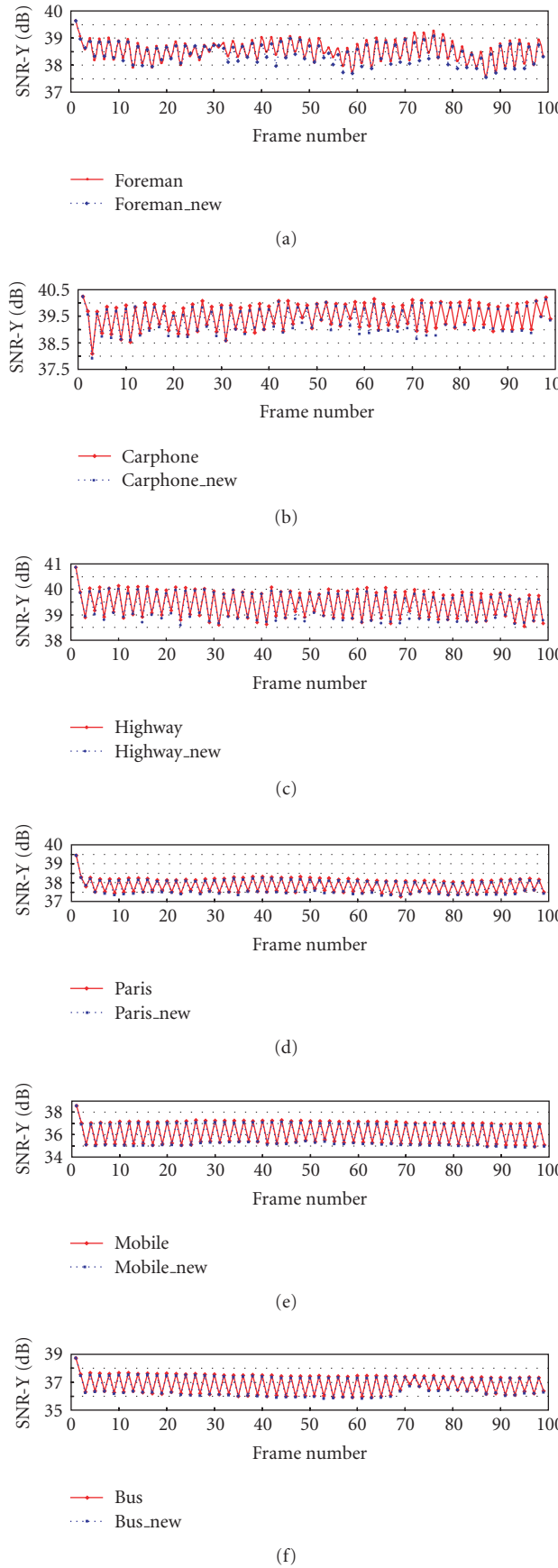

Figure 4: PSNR-Y comparisons of six test sequences.

Table 2: Percentage of missing blocks that should have passed intra-mode prediction.

QP	Foreman	Carphone	Highway	QP	Foreman	Carphone	Highway
5	0.206043%	0.816145%	1.634520%	28	0.060120%	0.145612%	0.252361%
10	0.193486%	0.481256%	1.623490%	30	0.054728%	0.165596%	0.260022%
14	0.187465%	0.380258%	1.581650%	32	0.087120%	0.182282%	0.225646%
16	0.189675%	0.147836%	1.421860%	34	0.054528%	0.115496%	0.102576%
18	0.192814%	0.253128%	0.834680%	36	0.123371%	0.048085%	0.054058%
20	0.204657%	0.303810%	0.812560%	38	0.118916%	0.000000%	0.000000%
22	0.084360%	0.360890%	0.572460%	40	0.160028%	0.017526%	0.035600%
24	0.084609%	0.219465%	0.535370%	45	0.202456%	0.045680%	0.000000%
25	0.035568%	0.226856%	0.513232%	49	0.186538%	0.032412%	0.000000%
26	0.035920%	0.233686%	0.512358%	—	—	—	—
QP	Paris	Mobile	Bus	QP	Paris	Mobile	Bus
5	0.341728%	1.113545%	1.784902%	28	0.053180%	0.160576%	0.571232%
10	0.303564%	0.928126%	1.658308%	30	0.056657%	0.125496%	0.380022%
14	0.285608%	0.612100%	1.282650%	32	0.077480%	0.182108%	0.456146%
16	0.250000%	0.448850%	1.420890%	34	0.064562%	0.153396%	0.127312%
18	0.205680%	0.546804%	1.048672%	36	0.100935%	0.082750%	0.109980%
20	0.211245%	0.563245%	0.751326%	38	0.108053%	0.024680%	0.056890%
22	0.092586%	0.623486%	0.881235%	40	0.157520%	0.026526%	0.016282%
24	0.094610%	0.480972%	0.853350%	45	0.122456%	0.085610%	0.071200%
25	0.086796%	0.356073%	0.621486%	49	0.191340%	0.061420%	0.092416%
26	0.099802%	0.15690%	0.724680%	—	—	—	—

Table 3: Comparison between the proposed and some well-known methods.

Sequences		QP = 24			QP = 28			QP = 32		
		ΔPSNR	ΔBit	ΔT	ΔPSNR	ΔBit	ΔT	ΔPSNR	ΔBit	ΔT
Carphone (QCIF)	Lee's	0.000	0	1.71	0.000	0.000	2.77	0.000	0.000	0.90
	Pan's	−0.007	0.649	13.74S	−0.001	1.245	10.31	0.000	1.816	5.88
	Kim's	−0.003	0.035	8.93	0.003	0.085	7.54	−0.003	−0.082	3.16
	Proposed	**−0.023**	**0.435**	**24.27**	**−0.033**	**−1.41**	**18.22**	**−0.013**	**−1.40**	**16.95**
Stefan (QCIF)	Lee's	−0.001	0.000	1.85	0.000	0.000	0.00	0.000	0.000	0.34
	Pan's	−0.021	0.366	15.93	−0.024	0.143	14.08	0.000	−1.142	12.45
	Kim's	−0.021	−0.578	14.95	−0.022	−0.554	14.36	−0.001	−1.706	12.07
	Proposed	−0.133	0.44	28.46	−0.223	0 .744	25.56	-0.17	0.352	23.21
Mobile (CIF)	Lee's	0.000	0.000	7.5	0.000	0.000	0.25	0.000	0.000	4.65
	Pan's	−0.132	0.885	24.69	0.013	1.429	23.44	−0.034	2.92	20.28
	Kim's	−0.006	−0.345	36.39	−0.017	−0.694	35.35	−0.020	−0.548	35.44
	Proposed	−0.043	−0.211	35.40	−0.030	−0.422	33.55	−0.043	−1.246	28.20
Bus (CIF)	Lee's	0.000	0.000	5.76	0.000	0.000	1.25	0.000	0.000	1.11
	Pan's	−0.013	0.159	16.37	−0.010	0.404	14.01	−0.001	0.447	12.17
	Kim's	−0.001	0.098	18.82	−0.004	0.44	17.79	−0.001	−0.316	14.84
	Proposed	−0.047	0.84	23.78	0.000	-0.066	20.75	−0.06	−0.164	17.35
Coastguard (CIF)	Lee's	0.000	0.000	2.00	0.000	0.000	0.81	0.000	0.000	2.00
	Pan's	−0.019	−0.238	17.10	−0.006	−0.067	14.53	−0.002	0.165	8.90
	Kim's	−0.020	−0.316	23.36	−0.013	−0.619	20.57	−0.002	−0.948	12.01
	Proposed	−0.013	−0.140	31.79	−0.007	−0.8 1	22.20	−0.043	−1.001	22.51
Paris (CIF)	Lee's	0.000	0.000	2.28	0.000	0.000	2.15	0.000	0.000	4.06
	Pan's	−0.003	0.135	9.31	0.000	1.756	7.07	−0.002	2.273	4.34
	Kim's	−0.002	−0.120	6.36	0.001	−0.023	7.06	−0.001	−0.052	3.98
	Proposed	−0.020	1.04	23.65	0.033	−0.048	22.46	0.030	−0.248	23.43

wrongly skipped for the sequence Foreman and less than 0.816145% (maximum) and 1.634520% (maximum) for Carphone and Highway, respectively. In CIF sequences, less than 0.341728% (maximum) is incorrectly skipped for the sequence of Paris, and less than 1.113545% (maximum) and 1.784902% (maximum) are for Mobile and Bus, respectively. In most cases only 0.6% or less is intra-skipped incorrectly. The statistic results of Table 2 indicate that this model with weighted coefficients we set has obvious effect on intra-skip.

Figure 4 compares PSNR-Y performance of these sequences when QP is 25. From the figure, it is clear that the PSNR-Y degradation is very minor, although sometimes there is some fluctuation, such as frame number 43 in Foreman sequence and number 70 in Bus sequence.

To compare our proposed scheme with recently proposed well-known methods, IBBPBBP sequence format is also selected in our experiments. Experimental results are presented in Table 3. When the value of ΔPSNR is negative and ΔBit is positive (a negative value of ΔPSNR means decreased PSNR and negative value of ΔBit means decreased bitrate), it corresponds to performance degradation and vice versa. ΔT denotes the percentage of saved encoding time. In the table, Lee's method is in paper [4] and Pan's method is in paper [6] and Kim's method is in paper [7]. They are evaluated against our proposed method. The experimental environment and configuration of JM are the same as shown in Table 1.

From the table, it is shown that in most sequences, the speed acceleration obtained by the proposed scheme is the best among four methods and provides very minor PSNR deterioration. The coding speed in QCIF sequences is almost three times faster than the other three methods, although there is some minor PSNR and bitrate degradation that affect image fidelity. Admittedly, in Mobile sequence, the proposed scheme just keeps the same level compared to Kim's. However, in the Coastguard and Paris sequences, the performance increase is considerable compared to the other three ones.

4. Conclusion

In this paper, we first give a brief introduction about H.264 and then exhaustively specify the whole procedure of inter-frame coding, especially concerning the conjunction of inter-partition modes and intramodes, to demonstrate that intra-skip is a very effective method to increase the speed of the encoder if adopted in the inter-frame coding. After that, we discuss a mathematical model for intra-skip and the critical advantage of this model that can optimize H.264 encoder together with any proposed fast inter-partition mode decision, search algorithms, and fast intra-algorithms. At last, experimental results are provided and illustrated to substantiate the practical value of this model in the inter-frame coding.

The coefficients of the mathematical model proposed in this paper might not bring perfect performance, as there is also some bitrate increase and PSNR degradation on certain circumstances, which means that few blocks are intra-skipped incorrectly. These few blocks therefore lead to incorrect mode prediction that brings out more extra residual to be encoded, which is also confirmed by our experiments. For example, when the proposed method is compared to Kim's method in the CIF sequence Mobile, it does not show great performance improvement like, other sequences because some few blocks are wrongly intra-skipped and lead to inexact prediction. The drawback could be tentatively solved by neural networks applied to PID control field [20, 21] and Fuzzy Control [22, 23] or similar areas for coefficients tracing so as to seek better performance.

References

[1] "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 — ISO/IEC 14496-10 AVC)," March 2003.

[2] C. Ning and S. Hui, "Research on a novel model for intra-skip in inter coding," in *Proceedings of the 9th International Symposium on Signal Processing and Its Applications (ISSPA '07)*, pp. 1–4, Sharjah, UAE, February 2007.

[3] Y. Cheng, Z. Y. Wang, K. Dai, and J. J. Guo, "Analysis of inter-frame coding without intra modes in H.264/AVC," in *Proceedings of the 7th Eurographics Symposium on Multimedia*, pp. 77–86, Nanjing, China, October 2004.

[4] J. Lee and B. Jeon, "Fast mode decision for H.264," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04)*, vol. 2, pp. 1131–1134, Taipei, Taiwan, June 2004.

[5] Y. Cheng, Z. Wang, J. Guo, and K. Dai, "Research on intra modes for inter-frame coding in H.264," in *Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design*, vol. 2, pp. 740–744, Coventry, UK, May 2005.

[6] F. Pan, X. Lin, S. Rahardja, et al., "Fast mode decision algorithm for intraprediction in H.264/AVC video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 813–822, 2005.

[7] B.-G. Kim, J.-H. Kim, and C.-S. Cho, "A fast intra skip detection algorithm for H.264/AVC video encoding," *ETRI Journal*, vol. 28, no. 6, pp. 721–731, 2006.

[8] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*, John Wiley & Sons, England, UK, 2003.

[9] T.-Y. Kuo and C.-H. Chan, "Fast variable block size motion estimation for H.264 using likelihood and correlation of motion field," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 10, pp. 1185–1195, 2006.

[10] Z. Zhou, J. Xin, and M.-T. Sun, "Fast motion estimation and inter-mode decision for H.264/MPEG-4 AVC encoding," *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 243–263, 2006.

[11] S.-E. Kim, J.-K. Han, and J.-G. Kim, "An efficient scheme for motion estimation using multireference frames in H.264/AVC," *IEEE Transactions on Multimedia*, vol. 8, no. 3, pp. 457–466, 2006.

[12] C. Grecos and M. Y. Yang, "Fast inter mode prediction for P slices in the H264 video coding standard," *IEEE Transactions on Broadcasting*, vol. 51, no. 2, pp. 256–263, 2005.

[13] S. Zhu and K.-K. Ma, "Correction to "a new diamond search algorithm for fast block-matching motion estimation"," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 287–290, 2000.

[14] Z. Chen, P. Zhou, and Y. He, "Fast integer pel and fractional pel motion estimation for JVT," in *6th Meeting of the Joint Video Team of ISO/IEC MPEG & ITU-T VCEG*, Awaji Island, Japan, December 2002, JVT-F017.

[15] Y. Nie and K.-K. Ma, "Adaptive irregular pattern search with matching prejudgment for fast block-matching motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 6, pp. 789–794, 2005.

[16] J. Bu, S. Lou, C. Chen, and Z. Yang, "A novel fast approach for H.264 inter mode decision," in *Proceedings of the 4th IASTED International Conference on Communications, Internet, and Information Technology (CIIT '05)*, pp. 220–224, Cambridge, Mass, USA, October-November 2005.

[17] T.-Y. Kuo and C.-H. Chan, "Fast macroblock partition prediction for H.264/AVC," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04)*, vol. 1, pp. 675–678, Taipei, Taiwan, June 2004.

[18] D. Wu, F. Pan, K. P. Lim, et al., "Fast intermode decision in H.264/AVC video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 953–958, 2005.

[19] Joint Video Team (JVT) Test Model JM10.1, January 2006, http://iphome.hhi.de/suehring/tml/download.

[20] L. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, Prentice-Hall, Upper Saddle River, NJ, USA, 1994.

[21] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.

[22] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control*, Springer, New York, NY, USA, 1996.

[23] H. Ying, *Fuzzy Control and Modeling: Analytical Foundations and Applications*, Wiley-IEEE Press, New York, NY, USA, 2000.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2009, Article ID 709813, 7 pages
doi:10.1155/2009/709813

Research Article

An Adaptive Systematic Lossy Error Protection Scheme for Broadcast Applications Based on Frequency Filtering and Unequal Picture Protection

Marie Ramon, François-Xavier Coudoux, and Marc Gazalet

Département d'Opto-Acousto-Electronique, Institut d'Electronique de Microélectronique et de Nanotechnologie, UMR 8520, Université de Valenciennes, Le Mont Houy, 59313 Valenciennes, Cedex 9, France

Correspondence should be addressed to François-Xavier Coudoux, coudoux@univ-valenciennes.fr

Received 30 May 2008; Accepted 22 September 2008

Recommended by Susanna Spinsante

Systematic lossy error protection (SLEP) is a robust error resilient mechanism based on principles of Wyner-Ziv (WZ) coding for video transmission over error-prone networks. In an SLEP scheme, the video bitstream is separated into two parts: a systematic part consisting of a video sequence transmitted without channel coding, and additional information consisting of a WZ supplementary stream. This paper presents an adaptive SLEP scheme in which the WZ stream is obtained by frequency filtering in the transform domain. Additionally, error resilience varies adaptively depending on the characteristics of compressed video. We show that the proposed SLEP architecture achieves graceful degradation of reconstructed video quality in the presence of increasing transmission errors. Moreover, it provides good performances in terms of error protection as well as reconstructed video quality if compared to solutions based on coarser quantization, while offering an interesting embedded scheme to apply digital video format conversion.

1. Introduction

Over the last few years, the Wyner-Ziv coding theorem [1] has found several applications in digital video coding and transmission [2–5]. Among these applications, error resilience properties of Wyner-Ziv (WZ) coding are here considered to strengthen the robustness of transmitted video bitstreams against channel distortions. Many application scenarios are concerned, including broadcast TV or video transmission over mobile networks. In [4], Rane et al. proposed a systematic lossy error protection (SLEP) scheme, in which a supplementary bitstream is generated using WZ coding and transmitted jointly with the unprotected MPEG-2 compressed bitstream. The so-called WZ stream is obtained by coarsely quantization and entropy coding of the main MPEG video stream. After transmission over error prone packet networks, the WZ stream is used to replace the lost data from the main stream, leading to a graceful degradation of reconstructed video quality with worsening error conditions.

In this paper, we present a new SLEP scheme which is based on the solution proposed in [4]. MPEG-2 video compression is considered in this work because of its widespread use in digital video broadcasting [5]. Regarding the scheme described in [4], however, many techniques have been added or improved within the present work to enhance the performances of the SLEP architecture. First, the supplementary WZ bitstream is generated using frequency filtering [6], instead of coarser quantization. This modification gives good performances in terms of error protection as well as reconstructed video quality compared to coarse quantization. Moreover, frequency filtering can be combined conveniently with decimation to perform video format conversion easily, which constitutes a great advantage. Finally, the proposed SLEP scheme is adaptive, so that error resilience varies according to picture encoding mode as well as motion properties of the video scene.

The remainder of the paper is organized as follows: first, we remind the systematic lossy source channel coding framework proposed in [4] for error resilient MPEG-2 broadcasting. Then, we detail our modified SLEP scheme

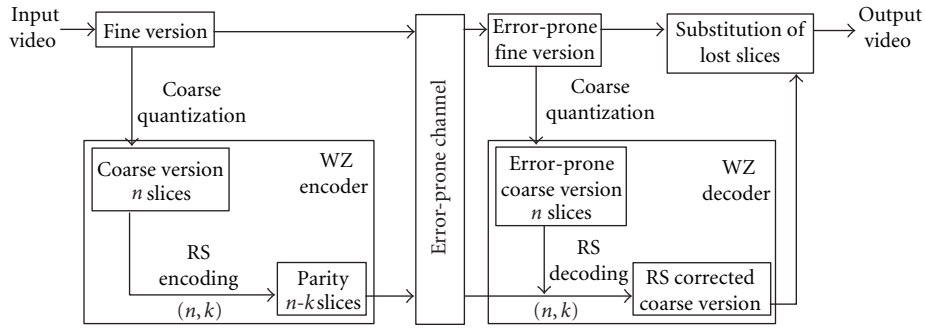

Figure 1: Block diagram of the SLEP scheme proposed in [4].

based on combined frequency filtering and unequal picture protection, and demonstrate its advantages over the one based on coarser quantization. In Section 3, we give experimental results that illustrate the performances of the proposed algorithm, and then propose a hybrid SLEP scheme, which switches adaptively between spatial error concealment and WZ decoding, based on motion detection. Finally, concluding remarks are given in Section 4.

2. The Proposed Slep Scheme

In this section, we describe the proposed SLEP scheme which is based on frequency filtering and unequal picture protection. The block diagram of the systematic lossy source-channel coding framework proposed in [4] for error resilient MPEG-2 broadcasting is shown in Figure 1.

The input video signal is first encoded by means of MPEG-2, and the resulting bit stream is transmitted over the error-prone packet network without error protection. In addition, a supplementary bit stream is generated using WZ encoding. First, a coarsely quantized version is generated from the main MPEG bit stream and entropy coded. As the entropy-encoded slices are of variable length, shorter slices are filled with zero bytes in order to adjust the k slices to the same size. Then, systematic Reed-Solomon (RS) (n, k) codes are applied across the k slices of the resulting data stream, after zero filling, as illustrated in Figure 2. Only the $(n\text{-}k)$ generated parity slices which constitute the so-called WZ stream are transmitted to the decoder. If packet losses occur, the WZ decoder uses both parity packets and the error-prone decoded MPEG video sequence as side information in order to obtain the error-free WZ description. Since the location of the lost slices is known, the RS decoder can perform erasure decoding across the error-prone slices. Therefore, the erroneous slices can be substituted with the corresponding correct but coarser versions, leading to a reconstructed video sequence of better visual quality. If RS error correction capacity is overcame, spatial error concealment is applied using the previously decoded frame.

This transmission scheme is fully compatible with actual digital video coding standards. It is more resilient to channel losses [4, 5] while adding negligible complexity increase with respect to conventional FEC systems. Indeed, it only requires

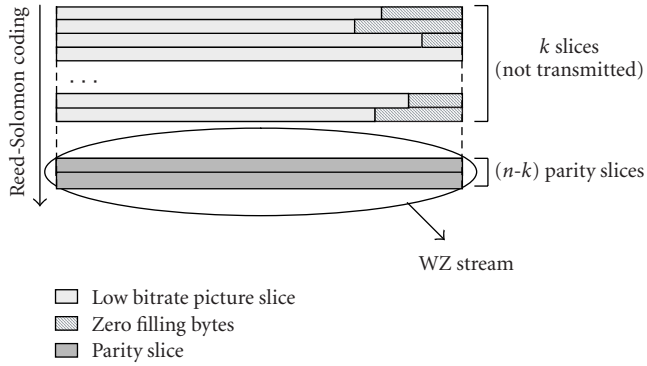

Figure 2: Reed-Solomon encoding across slices.

requantization of DCT coefficients, the coding parameters (motion vectors, mode decisions) from the main compressed version being reused. Such requantization process is simple to implement in the SLEP scheme. We note that FEC constitutes a special case of the SLEP scheme, for which the quantization step is the same for both MPEG and Wyner-Ziv video encoders.

We proposed in a previous work [6] to replace coarse quantization in the SLEP scheme, described in Figure 1, with frequency filtering (also called hereafter frequency scalability). Doing so, only those transform coefficients within a specified zone of a block are processed further, with the remaining set to zero. This process is also simple to implement, and corresponds to low-pass filtering if only low-frequency transform coefficients are selected. By computing directly the 2D IDCT of $N \times N$ blocks, we reconstruct a WZ description which consists in an image version of original picture size, but reduced details. The corresponding low-pass filtering process, which retains $N_2 \times N_2$ low-frequency DCT coefficients from the original $N_1 \times N_1$ block (in our case, $N_1 = 8$), is denoted hereafter N_1:N_2 (i.e., 8:4 means halving the number of coefficients in both directions: only the 4×4 lowest DCT coefficients of the original block are retained).

Moreover, low-pass filtering can also be combined with decimation to provide format conversion capabilities. Indeed, it is well known that downsampling and upsampling can also be performed in the transform domain [7–10]. To

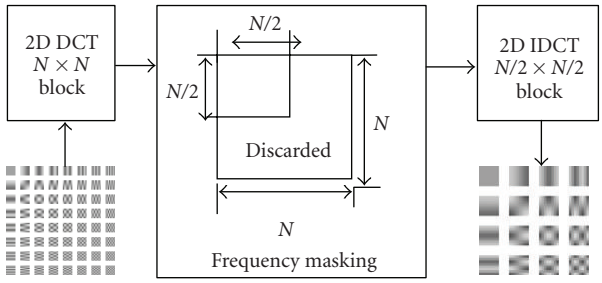

Figure 3: Spatial resolution downsizing by frequency filtering.

convert a high-resolution video signal into a low-resolution version of half size in both directions, frequency filtering is first applied to the 2D DCT of $N \times N$ blocks of the original signal to retain only the $N/2 \times N/2$ low-frequency coefficients (Figure 3). Then, a low-resolution video signal is obtained from the 2D IDCT of $N/2 \times N/2$ blocks after zonal filtering [7]. Such downscaling method in the transform domain is of low-computational complexity in comparison with those performed in the spatial domain: it avoids full decoding, spatial filtering and decimation, then full reencoding of the video signal. The authors in [10] show that more than 40% savings can be obtained as compared to spatial methods. This characteristic of our scheme should be of great interest with the recent deployment of technologies such as HDTV or mobile video, associated to a wide variety of receiving devices. Professional video manipulation (including video browsing, compositing, editing, or previewing) is also concerned, for which a low-spatial resolution version of the video content is generally used.

In the following subsections, we compare the proposed SLEP scheme with the one based on coarse quantization, in terms of error protection, as well as reconstructed video quality. The performance of our system has been evaluated over a wide range of symbol error rates and test video contents, as well as different bit rates for the main MPEG-2 bit stream. For comparison, we consider in what follows that the main MPEG-2 bit stream is encoded at the same bit rate as reported in [5], that is to say, 2 Mbps. For clarity, the SLEP scheme based on coarser quantization is called the SNR scalability-based SLEP scheme; the one using frequency filtering is called frequency scalability-based SLEP scheme.

2.1. Error Protection. In order to ensure a fair comparison between the two SLEP schemes, we consider, hereafter, a fixed bit rate for error protection. It means that the numbers of parity slices as well as their length are set identical. Indeed, the length of a parity slice from the WZ bit stream corresponds to the maximal size of a picture slice generated from the low-bit rate (coarsely quantized or low-pass filtered) version. Hence, in the present case, we consider the maximal image slice length after entropy coding $L_{\max}$ as a significant parameter. The calculation of the WZ stream size denoted R_{WZ}, as a function of $L_{\max}$ is given by

$$R_{\mathrm{WZ}} = (n\text{-}k) \times L_{\max}, \tag{1}$$

where n and k are the RS encoding parameters, and $L_{\max}$ is the maximum length of an entropy-encoded slice (expressed in bits). Frequency scalability is applied with 8:4 ratio, that is, only the 4×4 lowest frequency DCT coefficients are retained before applying IDCT. It provides efficient robustness to transmission errors, with a resulting WZ image of good visual quality. Obviously, a higher downscaling ratio could be applied in order to increase error robustness; but we verified that the reconstructed video quality rapidly decreases with increasing downscaling ratio. The corresponding coarsely quantized version has been determined in order to reach as much as possible the same bit rate of error protection. As mentioned above, only the parity slices are sent to the decoder, once shorter slices have been filled with zeros up to the maximal slice length. Hence, the more variable the image slice size is, the more the protection is unnecessarily applied to filling zeros, so the less useful the protection is. We define the ratio $L_{\max}/L_{\mathrm{mean}}$ as a parameter representative of the slice length variability, where L_{mean} corresponds to the mean slices length expressed in bits. The higher this ratio is, the more variable the encoded slices length.

Table 1 gives the slices length characteristics of WZ images for different 352×288 CIF test sequences. Only the results obtained for intracoded frames have been reported because it is well known that the distortion associated to intracoded pictures mostly impacts the overall video quality due to error propagation. We can see that, for a given maximal slice length $L_{\max}$, the variability of slices length is higher in the SNR scalability case than in the frequency scalability one. Indeed, by limiting each block to the same number of low-frequency DCT coefficients, frequency scalability clearly makes the slices length more homogeneous. In the same way, experiments have shown that the variability in the SNR scalability case is more dependent on video content (motion amount, detailed areas corresponding to high-frequency content). As zero-filling is applied before RS encoding across slices, the increased variability causes more zeros to be added to useful data and unnecessarily protected in the SNR scalability case. Hence, for the same SLEP parity overhead, the protected image data rate is superior in the case of a frequency-filtered WZ video bit stream. Consequently, the proposed scheme provides better efficiency in terms of error resilience in comparison with the one based on coarse quantization.

2.2. Reconstructed Video Quality. The reconstructed WZ images exhibit different kinds of artefacts depending on the use of SNR or frequency scalability mode. In the case of SNR scalability, both high-frequency and low-frequency DCT coefficients are strongly quantized. Inner distortions of block content (ringing) as well as well-known blocking effect appear consequently in the reconstructed image [11]. In the case of frequency scalability, the higher frequency coefficients are discarded, leading to a smoothing effect in areas of high-spatial activity. But as the low-frequency DCT coefficients are left untouched, areas of low and moderate activity are not affected, preserving the overall image quality. Thus, the frequency filtering drawback is generally less salient

Table 1: Comparison of Wyner-ZIV image characteristics for SNR and frequency scalability for different test images (L_{max} = maximum slices length, in bits; L_{mean} = mean slices length, in bits).

Video test sequence	L_{max} SLEP strategy		L_{max}/L_{mean} SLEP strategy	
	Freq.	SNR	Freq.	SNR
Foreman	2768	3008	1,49	1,67
Conference	2448	2264	1,51	1,57
Mob. Cal.	3608	5088	1,21	1,27
Old boat	3592	4544	1,59	1,71
Football	2496	2704	1,13	1,16
Map	2840	3488	1,16	1,29
Speaker	1792	1800	1,25	1,36
Flower	3816	5000	1,72	1,85
Mean	2920	3487	1,38	1,48

than the coding artefacts due to coarse quantization, which are reinforced with respect to the ones due to the original MPEG2 compression process.

We use the PSNR metric in order to evaluate the quality of the reconstructed video for the two SLEP schemes using the same protection rate. The PSNR is increased of 1.7 dB on average when frequency scalability is used (Table 2). This corresponds to a significant improvement in image quality. We also notice that the variations of PSNR values are strongly related to the spatial activity of the processed video sequences. But even for highly detailed videos (the worst case being *Old boat*) results are in favor of the frequency case. We can conclude that the proposed SLEP scheme offers better performances in terms of reconstructed video quality of intracoded pictures, at a parity of protection bit rate.

2.3. Unequal Protection Based on Picture Type. We propose now an adaptation of our SLEP scheme to account for I, P, or B picture coding mode during the WZ encoding step. It relies on changing the resolution protected by the WZ stream rather than the RS capacity, as proposed in [12]. Typically, an MPEG-2 compressed video sequence is made of a series of groups of pictures (GOPs), each GOP being composed of one intracoded (I) picture and the subsequent intercoded predicted (P) and/or bidirectional (B) pictures. The transmitted data in the latter include motion information (i.e., motion vectors) and intercoded residual error data.

The study described below was conducted for different MPEG-2 bit rates on a set of well-known CIF video sequences edited by the video quality expert group [13]. Table 3 gives as an example the results for the targeted bit rate of 2 Mbps with 30 frames/sec; the GOP characteristics are given by N = 12, M = 3, where N defines the distance between I frames, and M is the distance between consecutive I or P frames. We use as a distortion measure the normalized mean squared error (MSE) with respect to overall picture variance.

It is clear from the results in Table 3 that frequency filtering mostly affects intracoded pictures. In addition, since (I) pictures serve as the reference for (P)/(B) picture reconstruction, the corresponding high distortion will propagate to the subsequent pictures inside the entire GOP. In the case of intercoded pictures, motion should be recovered with no loss. Hence, the loss of intercoded residual data results in a hardly noticeable blurring around edges, with little or no propagation. In addition, the effectiveness of lowering the resolution becomes questionable for picture areas with little or no motion, while masking acts in the case of moving scenes.

For these reasons, we propose to adapt the WZ protection depending on the picture coding type. This protection consists in varying the 8:N_2 low-pass filtering strength over the entire GOP structure, so that the spatial resolution associated to (I) pictures is higher than the one of (P) and (B) pictures. This prevents blur from affecting visual quality of static scenes containing details. On the other hand, we propose also to adapt the decoding strategy depending on the motion characteristics of the video scene. Such adaptations are described in the following section.

3. Performance Evaluation

In this section, we analyze the performances of the proposed frequency scalability SLEP scheme by considering different configurations. Unequal picture protection is applied using the following WZ protection streams:

I(8:4), P(8:2), B(8:1) noted hereafter I4-P2-B1,

I(8:2), P(8:1), B(8:1) noted hereafter I2-P1-B1.

We also consider equal picture protection using 8:4, 8:2, and 8:1 (DC only) WZ protection. Finally, the conventional FEC case is presented as a point of comparison. Error concealment is achieved by copying the colocated slice in previous reference frame. We choose previous frame copy error concealment because this simple method is widely used in actual video decoders, although more sophisticated concealment strategies are available in the literature [14]. It is well known that this method can cope well with data losses when there is slow motion. The main MPEG-2 bit

Table 2: Comparison of PSNR results for different intracoded images and same protection rate.

Video sequence	SNR scalability—PSNR (dB)	8:4 frequency scalability—PSNR (dB)	PSNR gain (dB)
Foreman	30.0	32.2	2.2
Conference	32.4	35.9	3.5
Mobile Calendar	21.8	22.5	0.7
Old boat	25.2	25.3	0.1
Football	28.0	29.8	1.8
Map	26.4	27.8	1.4
Speaker	32.7	35.8	3.1
Flower garden	24.4	25.2	0.8
Mean	27.6	29.3	1.7
Max	—	—	3.5
Min	—	—	0.1

Table 3: Average distortion as a function of picture type.

Normalized MSE	I-picture	P-picture	B-picture
WZ 8:4	0,058	0,017	0,004
WZ 8:2	0,146	0,028	0,007
WZ 8:1	0,245	0,037	0,010

stream is encoded at 2 Mbps, with the IBBP frame structure, and 10% worth of RS parity information is added. These values are consistent with experimental conditions described in [4]. The Reed-Solomon (n, k) encoding parameters were determined experimentally based on the analytical model proposed in [12]. The WZ stream size depends on both the size of the parity slice (maximum size of a picture slice) and the RS parameters, which determine the number of parity slices. For the same final bit rate overhead, the lighter the reduced stream, the stronger the applied RS protection.

In the experiments, we simulate video streams transmission over a heterogeneous wired/wireless packet network with unpredictable error bursts. The slice loss process for this scenario can be modelled using a two-state Gilbert-Elliott model [15]. Figures 4 and 5 present experimental results in terms of error resilience as well as reconstructed video quality, respectively, with average burst length of 1,2 slices. The 8:1 (DC only) case clearly represents a border case, since it provides the highest error resilience also gives an error-free reconstructed sequence with low-visual quality.

Figure 4 gives the displayed erroneous slices rate after WZ correction. According to these results, the lower the resolution used for the WZ stream, the stronger the error resilience properties of the SLEP scheme. SLEP protection allows the RS error correction capability to practically double each time the spatial resolution of the WZ stream is lowered. Unequal picture protection permits rate savings on intercoded pictures protection, thus improving the error resilience properties for (I) pictures.

Figure 5 gives the distortion after transmission over error-prone networks as a function of the slice error rate. The overall distortion expressed in terms of MSE results from both uncorrected (concealed) and corrected slices. At

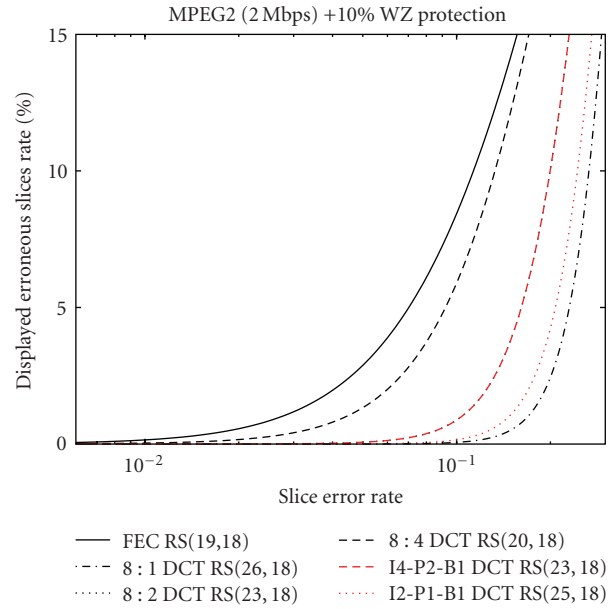

Figure 4: Evolution of the displayed erroneous slices rate after RS correction as a function of the incoming slice error rate.

the decoding stage, the corrected slices are identical to the transmitted slices, whereas in other cases, the corrupted slices are replaced by their frequency-filtered version. When WZ streams are used, correction induces a distortion due to lowering the resolution, which could cause the loss of texture information, especially for the 8:1 description. However, motion information is preserved, so that error concealment and most severe artefacts can be avoided.

According to the results, unequal error protection significantly improves the performances of the proposed SLEP scheme. For example, the I4-P2-B1 scheme has an error correction capacity and associated error concealment distortion equivalent to the fixed 8:2 scheme (5 lost slices per picture). An important issue is the distortion associated with slice substitution. For (P) pictures, the protected resolution is 8:2 in both schemes; the distortion due to downscaling remains unchanged. The reconstructed resolution of

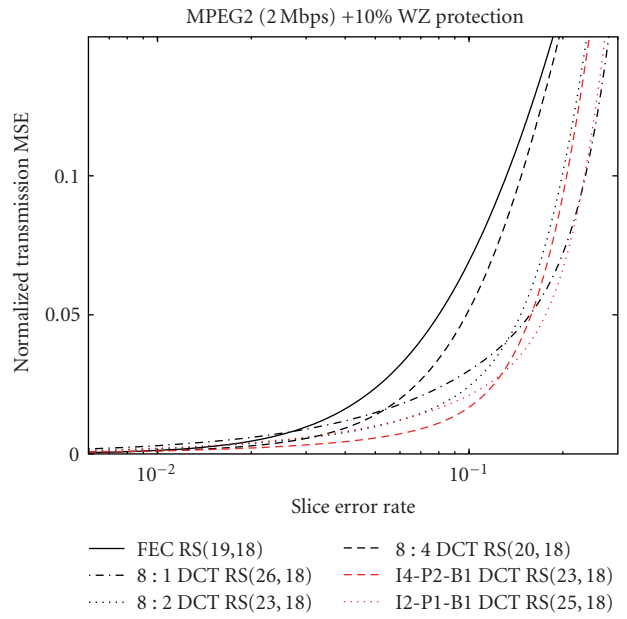

Figure 5: Evolution of normalized MSE transmission correction as a function of the incoming slice error rate.

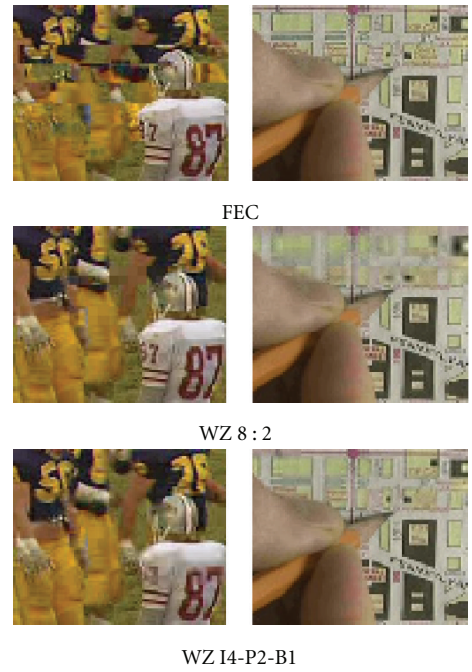

Figure 6: Displayed frame from FEC, WZ 8:2, and WZ I4-P2-B1 for *Football* (left) and *Map* (right) sequences.

Table 4: Average distortions associated with each frequency-filtered version and error concealment for the *Football* and *Map* sequences.

	Football	*Map*
WZ 8:4	0,004	0,012
WZ 8:2	0,009	0,022
WZ 8:1	0,019	0,032
Error concealment	0,021	0,015

the B-picture-coded difference is lowered, causing a small unpropagated distortion, whose overall effect is negligible. However, the distortion due to the lowering of the I-picture resolution is comparable to the 8:4 scheme. Hence, ensuring a higher resolution for reconstructed I-picture frames greatly improves the quality of the displayed sequence by limiting the distortion due to WZ correction.

In order to clearly analyze the properties of the proposed scheme, several tests have been conducted with different video sequences. Simulation results are given here for two specific extreme cases.

Case one: the *Football* sequence, which contains high motion and moderate texturing, and is representative of most standard TV programs.

Case two: the *Map* sequence, which contains very little motion and is highly textured, and more atypical.

Intuitively, the distortion due to replacing slices with their reduced spatial resolution versions is generally much lower than the distortion caused by error concealment. However, for the *Map* sequence, error concealment gives more satisfying results than slice substitution given the absence of motion and highly detailed content (Table 4). The only convenient SLEP protection schemes for this kind of sequence are the ones based on 8:4 downscaling of I-pictures (i.e., WZ 8:4 or WZ I4-P2-B1).

Figure 6 shows one displayed I-picture frame from each sequence. The first one corresponds to a standard FEC scheme, the second to our SLEP 8:2 scheme, and the third to the new I4-P2-B1 scheme.

As seen in the *Football* sequence, the FEC lost slices are concealed, but the motion information cannot be recovered and the resulting misalignment is visually very annoying. In the WZ case, correction is performed by replacing the slices with their lower resolution versions, with exact motion compensation. The recovery of the lost slices is very efficient due to the stronger error resilience of the proposed SLEP method: indeed, the improved WZ scheme avoids the large distortions due to unrecovered motion information. Subjective comparison with standard methods shows the clear benefit of our SLEP scheme when dealing with motion.

For the *Map* sequence, no motion information is needed since the video content is quasistatic, and thus error concealment is able to estimate the lost content. The failure of the FEC correction induces only a slight distortion. On the other hand, we notice that the distortion due to WZ 8:2 reduced resolution becomes visually annoying. Applying unequal picture protection improves the displayed resolution without sacrificing error resilience. The visual distortion introduced by replacing corrupted slices with their low-esolution versions remains slight, thus the visual quality of displayed video is higher (see bottom-right image, Figure 6).

Finally, in order to account for spatiotemporal properties of video scenes, we investigate a hybrid WZ decoder, which switches adaptively between previous frame copy error concealment and WZ substitution, based on slice-based motion

detection algorithm. Different motion detection algorithms have been previously described in the literature [16]. We suggest that a motion activity parameter v is computed for each slice from the coding parameters available in the WZ bit stream (motion vector coordinates, mode decisions), and compared to a predefined motion threshold T. As a first approach, the threshold value should be fixed experimentally based on statistics of the compressed video database. In the case of video scenes with little or no motion like the *Map* sequence ($v < T$), error concealment (see top-right image, Figure 6) is preferred to WZ substitution. Otherwise, WZ substitution is applied. In this case, the overall visual quality should be significantly improved for most video sequences, whatever motion amount. Such extension of our SLEP scheme based on motion adaptation is currently under consideration.

4. Conclusion

We have proposed a new systematic lossy error protection scheme based on frequency filtering, which ensures unequal picture protection depending on picture coding type. This scheme gives better performances compared to the classic FEC method as well as previous SLEP implementations, while adding limited complexity increase. It offers also interesting properties for digital video format conversion. Because the SLEP scheme is independent of the broadcast network, other protection tools linked to the network layer could be associated independently, improving performance (e.g., if a feedback channel was available, the SLEP protection could be adaptive). The scheme is also independent of the main stream codec and could be applied to other compression techniques than MPEG-2, including the new H.264/AVC coding standard.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments.

References

[1] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[2] S. S. Pradhan and K. Ramchandran, "Enhancing analog image transmission systems using digital side information: a new wavelet-based image coding paradigm," in *Proceedings of the Data Compression Conference (DCC '01)*, pp. 63–67, Snowbird, Utah, USA, March 2001.

[3] A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-Ziv coding of video: an error-resilient compression framework," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 249–258, 2004.

[4] S. D. Rane, A. Aaron, and B. Girod, "Systematic lossy forward error protection for error-resilient digital video broadcasting," in *Visual Communications and Image Processing 2004*, vol. 5308 of *Proceedings of SPIE*, pp. 588–595, San Jose, Calif, USA, January 2004.

[5] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.

[6] M. Ramon, F.-X. Coudoux, M. Gazalet, M. Gharbi, and P. Corlay, "Systematic lossy error protection of video based on reduced spatial resolution," in *Proceedings of the 13th IEEE International Conference on Electronics, Circuits, and Systems (ICECS '06)*, pp. 850–853, Nice, France, December 2006.

[7] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, Boston, Mass, USA, 1990.

[8] Y.-R. Lee, C.-W. Lin, S.-H. Yeh, and Y.-C. Chen, "Low-complexity DCT-domain video transcoders for arbitrary-size downscaling," in *Proceedings of the 6th IEEE Workshop on Multimedia Signal Processing (MMSP '04)*, pp. 31–34, Siena, Italy, September-October 2004.

[9] A. N. Skodras and C. A. Christopoulos, "On the down-scaling of compressed pictures," in *Proceedings of the IEEE International Workshop on Intelligent Signal Processing and Communication Systems (ISPACS '98)*, pp. 363–367, Melbourne, Australia, November 1998.

[10] A. N. Skodras and C. A. Christopoulos, "Down-sampling of compressed images in the DCT domain," in *Proceedings of the 9th European Signal Processing Conference (EUSIPCO '98)*, Island of Rhodes, Greece, September 1998.

[11] H. R. Wu and K. R. Rao, *Digital Video Image Quality and Perceptual Coding*, chapter 3, CRC Press, Taylor and Francis, Boca Raton, Fla, USA, 2006.

[12] S. Rane and B. Girod, "Systematic lossy error protection of video based on H.264/AVC redundant slices," in *Visual Communications and Image Processing 2006*, vol. 6077 of *Proceedings of SPIE*, San Jose, Calif, USA, January 2006.

[13] VQEG, "Multimedia Group TEST PLAN Draft Version 1.15," September 2006.

[14] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*, chapter 14, Prentice Hall, Upper Saddle River, NJ, USA, 2002.

[15] E. N. Gilbert, "Capacity of a burst-noise channel," *The Bell System Technical Journal*, vol. 39, pp. 1253–1265, 1960.

[16] M. A. Tekalp, *Digital Video Processing*, Prentice Hall, Upper Saddle River, NJ, USA, 1995.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2009, Article ID 681078, 10 pages
doi:10.1155/2009/681078

Research Article

Adaptive Error Resilience for Video Streaming

Lakshmi R. Siruvuri, Paul Salama, and Dongsoo S. Kim

Department of Electrical and Computer Engineering, Purdue School of Engineering and Technology, Indiana University-Purdue University at Indianapolis, 723 West Michigan Street, SL160, Indianapolis, IN 46202, USA

Correspondence should be addressed to Paul Salama, psalama@iupui.edu

Received 1 July 2008; Revised 29 January 2009; Accepted 24 March 2009

Recommended by Lorenzo Ciccarelli

Compressed video sequences are vulnerable to channel errors, to the extent that minor errors and/or small losses can result in substantial degradation. Thus, protecting compressed data against channel errors is imperative. The use of channel coding schemes can be effective in reducing the impact of channel errors, although this requires that extra parity bits to be transmitted, thus utilizing more bandwidth. However, this can be ameliorated if the transmitter can tailor the parity data rate based on its knowledge regarding current channel conditions. This can be achieved via feedback from the receiver to the transmitter. This paper describes a channel emulation system comprised of a server/proxy/client combination that utilizes feedback from the client to adapt the number of Reed-Solomon parity symbols used to protect compressed video sequences against channel errors.

1. Introduction

With the advances in technology, applications such as video telephony, video conferencing, video-on-demand, video broadcasting, and video email have become a reality. In fact, during the last thirty years, there has been a tremendous advance in the visual communication field [1]. One of the drawbacks of digital video signals is that they occupy too much bandwidth, and thus compressing video prior to storage and/or transmission is necessary. This has led to the development of many successful video compression standards [1–9].

Compressed video is very vulnerable to channel errors, and different means have to be used to protect the compressed data [10]. These can be classified into three categories: error resilience, error control coding, and error concealment. Error resilience refers to schemes that introduce error resilient elements at the video compression stage, which reduce the interdependencies of the data-stream, in order to mitigate error propagation. Inevitably, the use of error resilience degrades coding efficiency.

Error control coding refers to error protection by using channel coding schemes in the application layer. Contrary to source coding, channel coding inserts redundant information into the data-stream, which helps detect and correct errors due to the channel impairments. When video signals are transmitted over packet networks or wireless channels, serious corruption might occur to the data-stream due to (1) burst packet losses that stem from network congestion and/or (2) burst bit errors that are a consequence of channel fading. Hence, it is wise to introduce channel coding in the application layer to obtain better error protection. One particular channel coding scheme, namely, Reed-Solomon (RS) coding, is effective since RS codes are maximum distance separable codes that are well suited for error protection against burst errors.

It is possible to jointly use error resilience and error control schemes, commonly referred to as joint source and channel coding. This requires two types of optimized data rate allocations: (1) the optimized data rate allocation between source coding and channel coding, and (2) the optimized allocation of the source coding data rate among the various source coding elements to introduce an appropriate amount of error resilience into the data-stream. Unlike the above schemes that actively place error protection at the encoder side, error concealment is a passive scheme that

attempts to perform error recovery at the decoder by using actual video content, where that content is correctly decoded pixel values and motion vectors.

In general, the performance of source coding or joint source and channel coding schemes can be further improved through the use of feedback. In [11] a Rate-Distortion (R-D) framework for rate-constrained motion estimation and mode decision based on the use of several reference frames is proposed. In addition, a feedback channel is used by the decoder to inform the encoder about successful or unsuccessful transmission events by sending positive (ACK) or negative (NACK) acknowledgments. This information is utilized for updating the error estimates at the encoder, which the encoder utilizes in the compression of subsequent frames. The disadvantages of this method are: (1) it is not standard compliant, (2) feedback is assumed to be received in a timely manner, and (3) the increase in complexity resulting from having to search for optimal modes and motion vectors over multiple reference frames. In addition, it assumed that two subsequent error events are unlikely to occur.

Alternatively, [12] presents feedback and error-protection strategies for wireless transmission of progressively coded video using an unequal error protection scheme that employs high-memory rate-compatible punctured convolutional codes with a sequential decoding algorithm. This method assumes that the transmitter knows perfectly the long term channel statistics as well as the existence of an error-free low bit rate channel in both directions. The feedback channel is used for sending feedback via several ARQ strategies, but the paper does not describe what happens when feedback is delayed long enough to be useful.

In both [11, 12], the R-D framework does not take into consideration distortion due to data corruption and/or loss. Recent methods that take into consideration end-to-end distortion have been proposed in [13, 14]. In [13], a stochastic framework for an R-D-based encoder design for video compression and transmission over error-prone networks was described. The distortion measure used was the mean square error evaluated over an ensemble of channels of known packet loss probabilities. Furthermore, the channels are assumed to have uniform loss probabilities, and the packet loss probabilities were used in the stochastic R-D optimization framework for selecting the optimal macroblock mode and motion vectors. Similarly, [14] describes an end-to-end method for R-D-optimized selection of the coding modes for H.264 video coding. The proposed scheme assumes that the channel error rates as well as the error concealment method used by the decoder are both known at the encoder. A new Lagrange multiplier was derived, but the derivation of the Lagrange multiplier assumes high-resolution quantization, which does not necessarily apply well for video compressed at low or very low data rates.

The schemes proposed in [11–14] all tackle the problem of optimally coding video (in the R-D sense) for the purpose of transmission, based on feedback from the receiver to the transmitter, but do not address the problem of reliably transmitting previously coded video streams over data channels while utilizing feedback from the receiver to the transmitter. Furthermore, they assume that the channel conditions are perfectly known or that feedback arrives in a timely fashion, which are not always possible. Streaming preencoded video sequences over both constant and variable bit rate channels, such that each video unit is decoded before a deadline was addressed in [15]; however it was assumed that (1) the channel was error free although it had limited transmission rate, (2) the packet losses and delays in the access network can be neglected, and (3) immediate retransmission is available.

In this paper we address the problem of transmitting previously coded H.264 [9] video streams when channel loss probabilities are not exactly known by the transmitter, and when feedback from the receiver to the transmitter does not always arrive in a timely fashion. We first describe a system that emulates a real-time network, and then we propose an adaptive Reed-Solomon error control coding scheme that utilizes feedback from a client to adjust the protection level at the server. The server uses a three-state channel model and the feedback data from the client to estimate the future channel state and consequently the protection level needed.

2. Emulation System

Any communication system consists of a transmitter, a receiver, and a channel through which data is transmitted. In general any communication channel, especially wireless networks, suffers from data loss and/or bit corruption, which affect the quality of data being transmitted. In particular when compressed video sequences are being sent, the various channel impairments will lead to visual distortion when the signal is reconstructed at the receiver. Furthermore, due to the predictive nature of current video compression standards, this distortion will propagate through the video sequence. In fact, based on many factors such as source coding, the transport protocol, and the amount and type of information loss, the distortion induced can range from degradation that may encompass a few frames to a completely unusable video sequence. Therefore it is necessary for the transmitter to attempt to protect the data and for the receiver to perform error concealment at the receiver to minimize any observed distortion.

In order for the transmitter to adequately protect compressed video sequences, it needs to use some form of channel coding. However, in order to avoid wasting bandwidth unnecessarily, the transmitter needs to tailor the protection to the current channel conditions. This can be achieved via feedback messages sent by the receiver that indicate the latest channel loss rates. The advantage of this approach is that the transmitter can attempt to track the channel changes and accordingly adapt the protection it affords to the compressed data stream instead of assuming certain loss probabilities.

In our work we emulate the communication channel by a proxy server that intentionally generates packet delays and losses. In fact, all the transmissions between the server and the client are done via the proxy. Since the receiver needs to convey loss information back to the transmitter

as it receives the incoming video, two independent channels are established. One channel is utilized to reliably transmit control information from the client/receiver to the server/transmitter, while the other is used for the actual transfer of compressed video. The UDP/IP protocol suite is used for the transmission of data packets and the TCP/IP protocol suite for the transmission of control/feedback packets. All the control signals are sent via the TCP ports, and the data signals are sent through the UDP ports. Request signals and feedback signals are considered as control signals.

The process is as follows. The client requests a particular video stream at a particular frame and data rate from the server via the proxy. The proxy, which is waiting for a signal from the client, receives the client's requests, parses the data to be forwarded to the server, makes a connection with the server, and forwards the request. The server receives the request and transmits the required data. The proxy then receives the data sent from the server and forwards it to the client. In the process, the proxy delays both control and data packets and induces loss to the data carrying packets only. The entire process is illustrated in Figure 1.

The server channel encodes the requested stream by applying the appropriate RS coding level and interleaves the channel-coded data. The size of each interleaving block is $N \times S$, where N is length of each RS code word, and the span size S is the number of RS code words that are to be interleaved together. The interleaved blocks are numbered and passed as header information to be used at the client for error detection. The data of size $(K \times S)$, where K is the number of message symbols in each code word, is passed through the RS encoder. The RS encoder adds the necessary parity symbols based on the values of N and K. The number of parity symbols added is $N - K$. In our work, the value of N is always fixed at 255 as each symbol is equal to 8 bits. The value of K is varied according to the packet loss rate estimates obtained based on the feedback from the client.

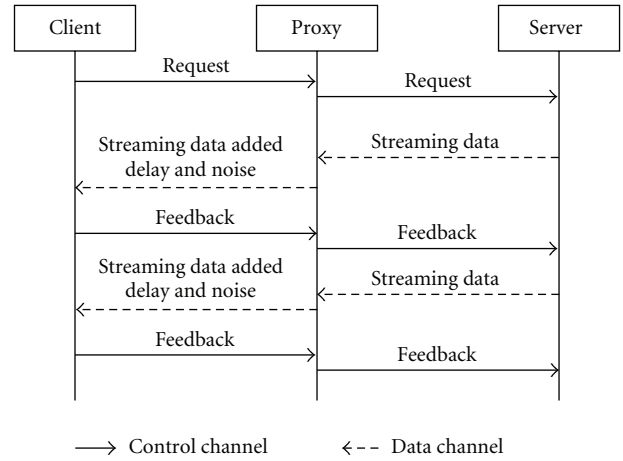

Figure 1: Sequence diagram of the server-proxy-client model.

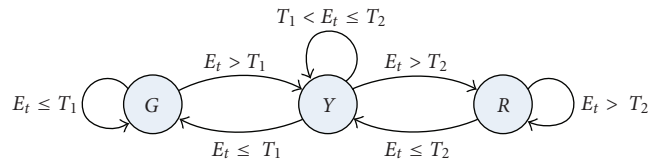

Figure 2: State transition diagram of the adaptive channel coding.

3. Adaptive Reed-Solomon Channel Coding

In order to efficiently utilize the channel and to adapt the transmitted data to varying channel conditions, the server applies varying protection levels, based on feedback from the client, which contains information regarding the number of packets lost in each interleaved block. The server tries to predict the number of losses in the future based on the feedback information received using a weighted moving average and deviation as given in (1):

$$\begin{aligned} \tau_{t+1} &= \alpha x_t + (1-\alpha)\tau_t, \\ \delta_{t+1} &= \beta |x_t - \tau_{t+1}| + (1-\beta)\delta_t, \\ E_t &= \tau_t + c\delta_t. \end{aligned} \tag{1}$$

Here α and β are smoothing factors whose values are 0.4, c is tolerance factor whose value is either 1 or 2, x_t is the number of lost packets sent by the client, E_t the predicted future losses, τ_t the average loss, and δ_t the standard deviation of the number of lost packets. Based on the predicted value E_t, the server changes the message length K, which amounts to changing the error protection.

The process through which the server modifies the protection it applies to data packets is described via a three-state state diagram as shown in Figure 2. Again E_t is the predicted loss at the server based on the feedback information received, and T_1 and T_2 are two preselected thresholds known as the lower threshold level and the upper threshold level, respectively.

When E_t is less than T_1, the server views the current error level as being under control and the system is considered to be in the good state G. At this stage, the message length K is maintained at N-TP/S, where P is the packet size, and T a variable threshold that is set equal to T_1. When the predicted error goes beyond T_1, while the system is in state G, the system then enters state Y, the threshold T incremented step by step towards T_2, and the message length adjusted accordingly. The system remains in state Y as long as the predicted error level is less than or equal to T_2. When E_t exceeds T_2, the system enters state R, where the error level is considered too high. At this point the message length is set to and kept at N-T_2P/S. When the predicted error falls below T_2 while the system is in state R, the threshold T is decremented step by step toward T_1 and the message length is set to N-TP/S accordingly.

4. Results

To test the efficacy of the proposed approach, simulations were run for different video sequences by setting different proxy parameters. The main parameters of interest at the

server are the coded bit rate of the video sequence, packet size, and interleaving span size. The different sequences used were *foreman*, *Akiyo*, *Carphone*, and *Mother-Daughter* in QCIF format. All of the above mentioned sequences have 300 frames except for the *Mother-Daughter* sequence which has 950 frames. A long sequence was made by combining all the sequences, *Akiyo*, *Carphone*, *Foreman*, *Salesman*, *Container*, *Mother-Daughter*, *Coastguard*, and *Claire*. This long sequence named "*all*" has 3315 frames with duration of 110.5 seconds at 30 frames/sec. The sequences were coded at bit rates of 32 kbps, 64 kbps, and 128 kbps. The packet sizes used were 128 bits, 256 bits, 512 bits, and 720 bits. The span sizes used were 8, 16, and 20. Furthermore, each QCIF frame was divided into 9 slices, and an *IPPP*... GOP structure was used, whereby an I frame is inserted every 15 frames.

At the receiver, the decoder attempts to decompress the video sequence after the Reed-Solomon decoder has fixed any channel errors. If the Reed-Solomon decoder cannot fix some errors, the decoder will then partially decode the video stream and will discard whatever it cannot decode until it finds a valid start code. This process is repeated until the entire video stream has been decompressed. As a consequence, some frames within the video sequence will incur damage. The damaged regions in any frame are replaced by the colocated regions in its previous neighbor.

The parameters of interest at the proxy are the delay used and the buffer size at the proxy. The different delays used were constant delay in the range of 0.22–0.35, constant interleaving delay in the range of 0.026–0.03, and random delay with mean in the range of 0.01–0.035. Different buffer sizes used were in the range of 8–1000. In the current work buffer size refers to the size of the priority queue used at the proxy. The experiments were conducted with a given set of parameters using feedback, and the same experiment was conducted without feedback. The values of T_1 and T_2 used in the experiments with feedback are approximately 5% and 20% of the number of packets in an interleaving block assuming that the packet losses in the network range are from 0 to 20%. The maximum number of packets the adaptive algorithm can recover is T_2. The experiments without feedback have message length K set at T_1 (which is usually 5% of the number of packets in an interleaving block), for the entire experiment. All comparisons are based on PSNR values and the total symbol errors that occurred and were corrected in each case. The PSNR values were computed as PSNR = $10\log(255^2/\text{MSE})$, where MSE denotes the mean square error between the decoded uncorrupted frames and those that have been reconstructed using both the adaptive and nonadaptive schemes. A PSNR value of 100 dB or more was used to indicate perfect reconstruction. In other words a reconstructed frame matches exactly with its corresponding decoded uncorrupted counterpart.

Table 1 provides a summary of the different rates, delays, buffer sizes, packet sizes, and span sizes used throughout. Figures 3, 4, and 5 illustrate the performance of the overall system when *foreman* was coded at a data rate of 64 kbps, and the data packets experienced random delay with a mean of 0.057 seconds. The receiver had a buffer size of 40 packets where each packet size was 720 bits. An interleaving span size

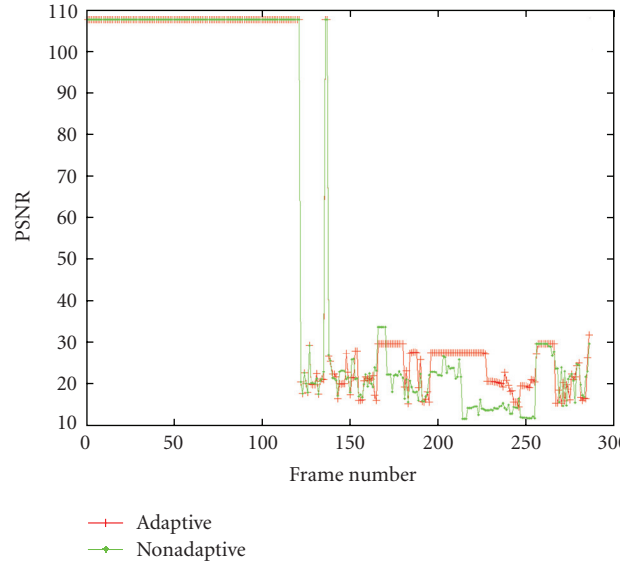

Figure 3: PSNR for "*foreman*" sequence at 64 kbps, random delay 0.057, buffer size 40, packet size 720, span 20, PL rate 30.9%.

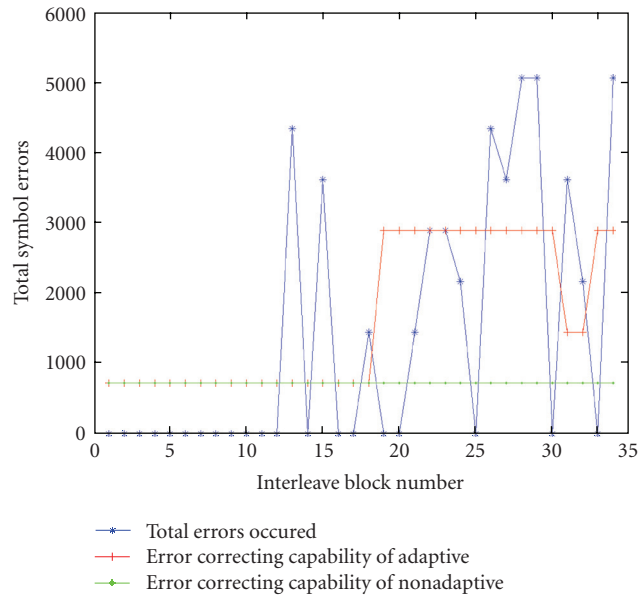

Figure 4: Performance of feedback for "*foreman*" sequence at 64 kbps, random delay 0.057, buffer Size 40, packet size 720, span 20, PL rate 30.9%.

of 20 was used for a packet loss (PL) rate of 30.9%, and the feedback packets were subjected to a constant delay of 0.057 seconds. The PSNR values were better in the adaptive case when compared to the nonadaptive case by about 40.39%, as the adaptive algorithm had the capacity to correct the losses 28.57% of the time. The values of T_1 and T_2 used in this experiment are 1 and 4 which are approximately 5% and 20% of the number of packets in the interleaving block. The nonadaptive algorithm had the capacity to correct the losses 0% of the time. The mean error in predicting the number of packets and hence symbols lost is 1.3529 with a variance

Table 1: Summary of the data rates, delays, buffer sizes, packet sizes, and span sizes for the experiments.

Seq. name-rate-buffer (fig #)	Data rate	Delay	Buffer size	Packet size	Span size
foreman-mid-small (3–5)	64 kbps	0.057 s	40 packets	720 bits	20
all-low-large (6–8)	32 kbps	0.026 s	500 packets	256 bits	16
all-low-small (9–11)	32 kbps	0.028 s	175 packets	256 bits	8
all-mid-mid (12–14)	64 kbps	0.020 s	250 packets	256 bits	16
all-high-large (15–17)	128 kbps	0.012 s	500 packets	256 bits	8

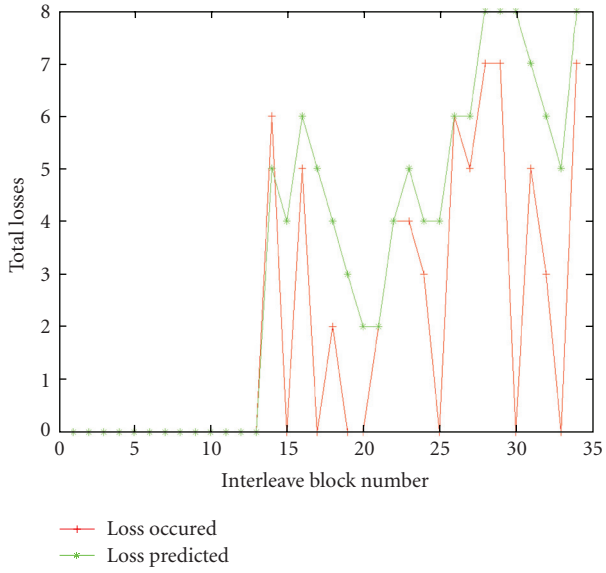

Figure 5: Performance of prediction for "*foreman*" sequence at 64 kbps, random delay 0.057, buffer size 40, packet size 720, span 20, PL rate 30.9%.

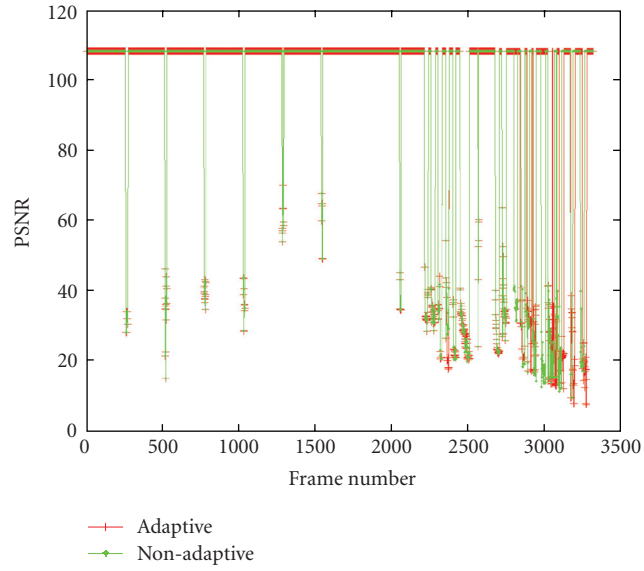

Figure 6: PSNR comparisons for "*all*" sequence at 32 kbps, random delay 0.026, buffer size 500, packet size 256, span 16, PL rate 4.9%.

of 3.6898. Since the maximum number of packets that can be recovered using the current work is T_2, it is not possible to recover losses greater than T_2 even if the predicted loss is always greater than the actual loss. The value of c (prediction parameter) used in this experiment is 2.

Figures 6, 7, and 8 illustrate the performance of the overall system when the "*all*" sequence was coded at 32 kbps, the data packets experienced random delay with a mean of 0.026 seconds, and the packet loss rate was 4.9%. The experiment also used a random delay with a mean of 0.026 seconds for the feedback packets, c was set to 1, and the values of T_1 and T_2 used were 1 and 4 which are approximately 5% and 20% of the number of packets in the interleaving block. The PSNR performance was better in the adaptive case when compared to the nonadaptive case by about 12.54%. The adaptive algorithm had the capacity to correct the losses 17.39% of the time, whereas the nonadaptive algorithm had the capacity to correct the losses 8.7% of the time. The average error in predicting the number of packets and hence symbols lost is 0.99 with a variance of 3.35. From Figure 8, it can be seen that at low packet losses, the adaptive case underestimated the losses most of the time. As a consequence, its error correcting capability was weak. In this case it may have been better to use $c = 2$.

Figures 9, 10, and 11 depict the performance of the overall system when "*all*" sequence was coded 32 kbps, the data packets experienced random delay with a mean of 0.028 seconds, and the packet loss rate was 10.07%. The experiment also used a random delay with a mean of 0.028 seconds for the feedback packets, c was set to 1, and the values of T_1 and T_2 used were 1 and 3 which are approximately 5% and 20% of the number of packets in the interleaving block. The average gain, in PSNR, afforded by the adaptive scheme when compared to the nonadaptive case was about 62.49%. The adaptive algorithm had the capacity to correct the losses 16.88% of the time, and the nonadaptive algorithm had the capacity to correct the losses 7.79% of the time. From Figure 11, it can be seen that the estimation was better than in the previous experiment. It can be deduced that at higher packet losses the prediction is more efficient than at lower packet losses, which is reflected by the fact that the mean error in predicting the number of packets and hence symbols lost is 0.91 with a variance of 1.83.

Figures 12, 13, and 14 exhibits the PSNR comparisons, performance of feedback, and prediction for the "*all*" sequence when coded 64 kbps. The experiment used random delay of 0.02 seconds for the feedback packets, c was set to 1, the packet loss rate was 18.63%, and the values of T_1

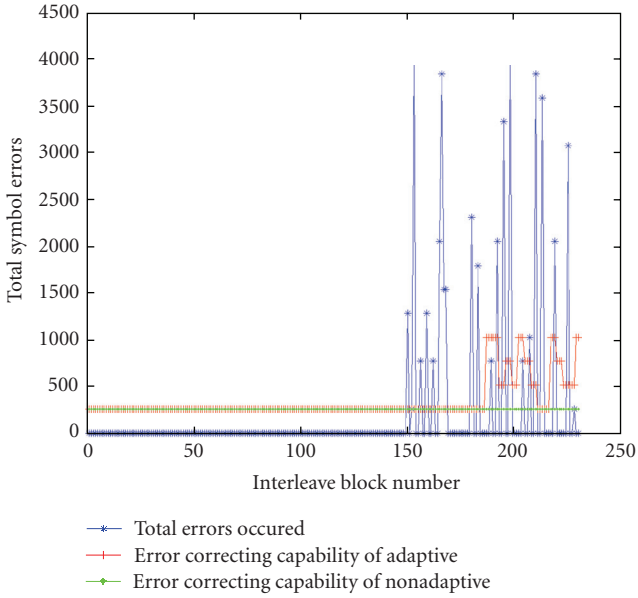

Figure 7: Performance of feedback for "*all*" sequence at 32 kbps, random delay 0.026, buffer size 500, packet size 256, span 16, PL rate 4.9%.

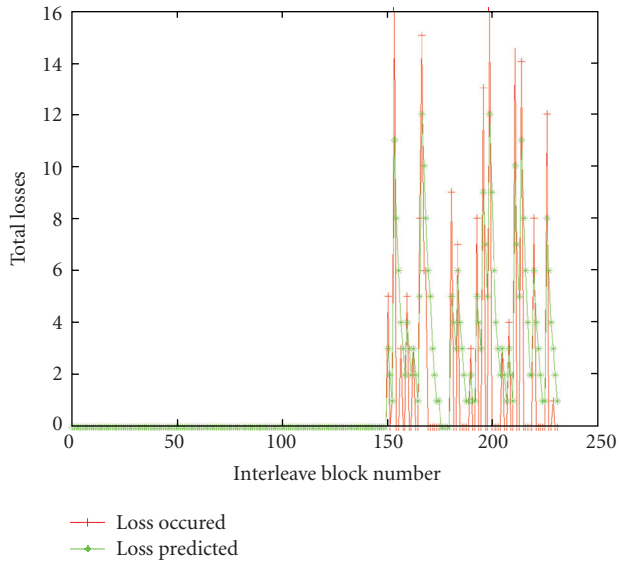

Figure 8: Performance of prediction for "*all*" sequence at 32 kbps, random delay 0.026, buffer size 500, packet size 256, span 16, PL rate 4.9%.

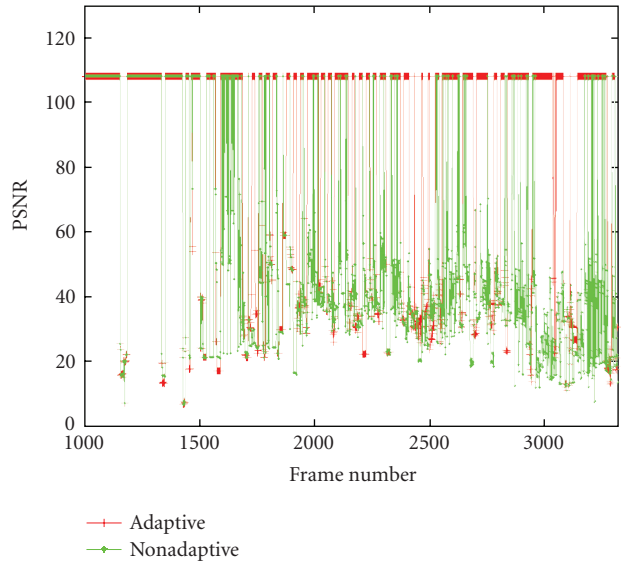

Figure 9: PSNR comparisons for "*all*" sequence at 32 kbps, random delay 0.028, buffer size 175, packet size 256, span 8, PL rate 10.07%.

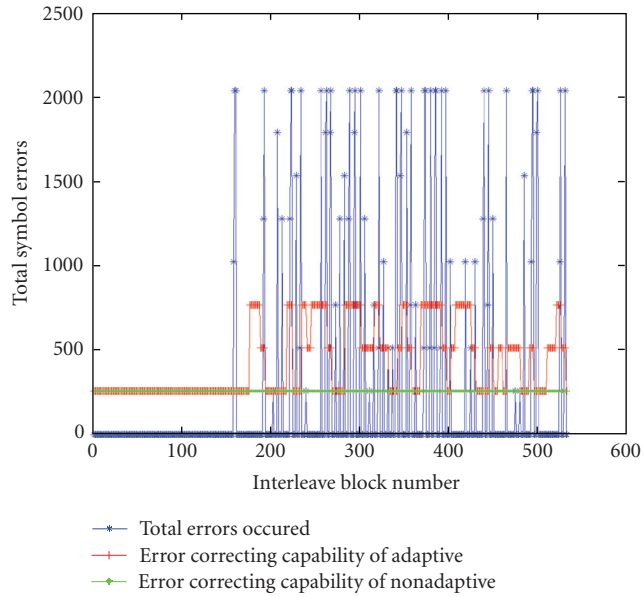

Figure 10: Performance of feedback for "*all*" sequence at 32 kbps, random delay 0.028, buffer size 175, packet size 256, span 8, PL rate 10.07%.

and T_2 used were 1 and 4 which are approximately 5% and 20% of the number of packets in the interleaving block. The PSNR performance was better in the adaptive case when compared to the nonadaptive case by about 31.1%. The adaptive algorithm had the capacity to correct the losses 18.80% of the time, whereas the nonadaptive algorithm had the capacity to correct the losses 6.84% of the time. The mean error in predicting the number of packets and hence symbols lost is 3.41 with a variance of 9.32.

Figures 15, 16, and 17 illustrate the PSNR comparisons, performance of feedback for "*all*" sequence at 128 kbps. The experiment used a packet loss rate of 15.66%, random delay of 0.012 seconds for the feedback packets, c was set to 1, and the values of T_1 and T_2 used were 1 and 3 which are approximately 5% and 20% of the number of packets in the interleaving block. The PSNR performance was better in the adaptive case when compared to the nonadaptive case by about 26.56%. The adaptive algorithm had the capacity

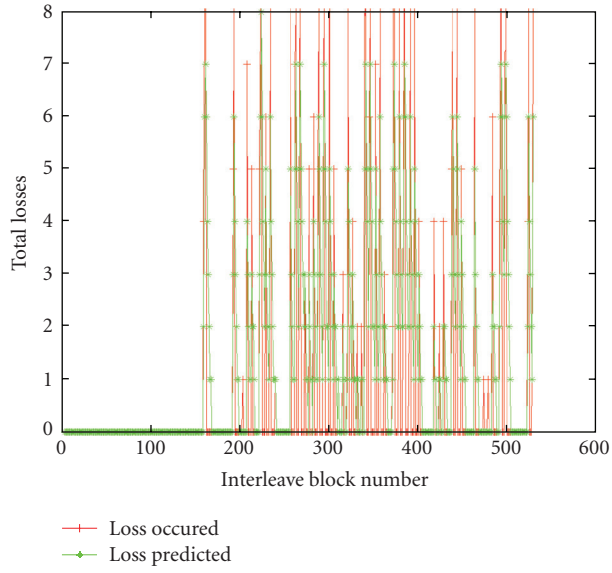

Figure 11: Performance of prediction for "*all*" sequence at 32 kbps, random delay 0.028, buffer size 175, packet size 256, span 8, PL rate 10.07%.

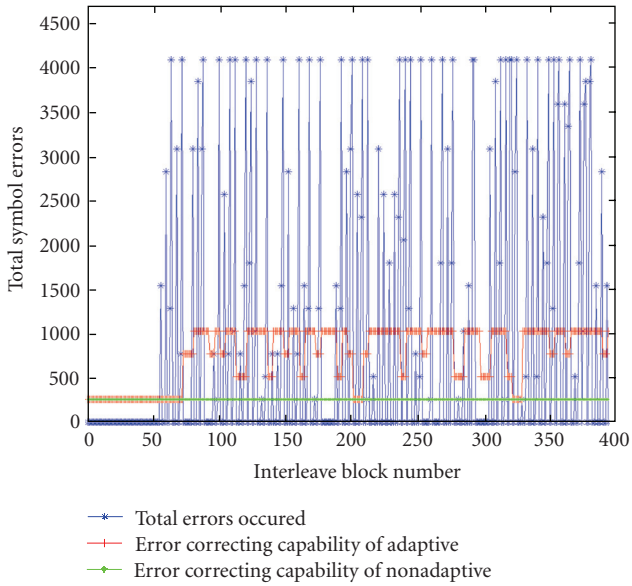

Figure 13: Performance of feedback for "*all*" sequence at 64 kbps, random delay 0.02, buffer size 250, packet size 256, span 16, PL rate 18.63%.

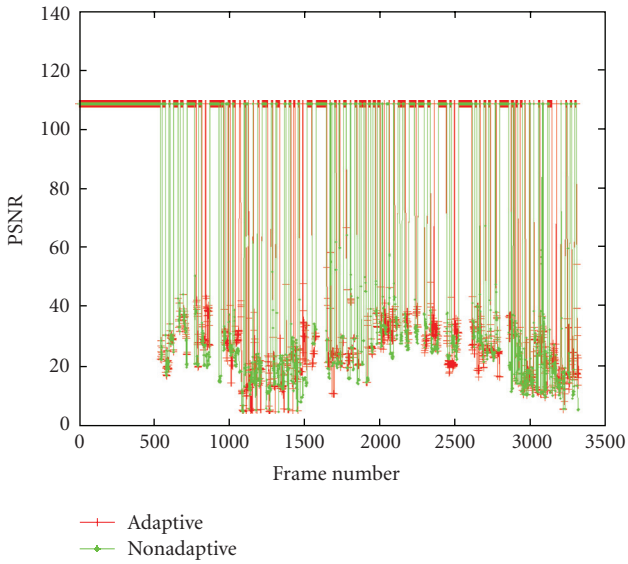

Figure 12: PSNR comparisons for "*all*" sequence at 64 kbps, random delay 0.02, buffer size 250, packet size 256, span 16, PL rate 18.63%.

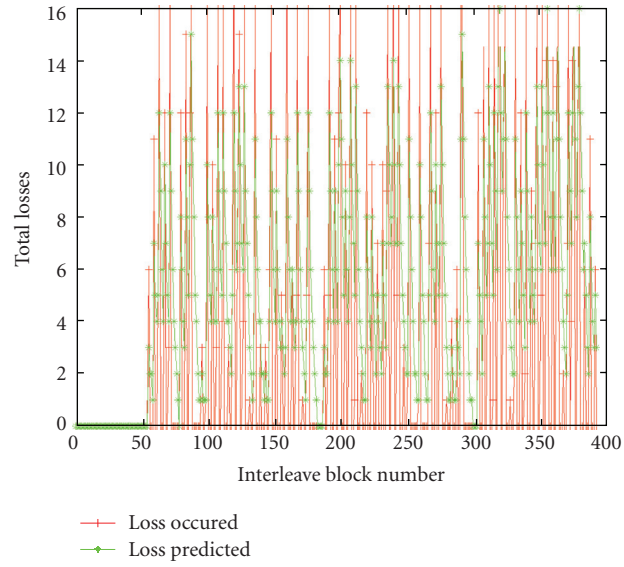

Figure 14: Performance of prediction for "*all*" sequence at 64 kbps, random delay 0.02, buffer size 250, packet size 256, span 16, PL rate 18.63%.

to correct the losses 10.53% of the time, in contrast to the nonadaptive algorithm which had the capacity to correct the losses 4.14% of the time. From Figure 17 it can be seen that the estimation was effective and resulted in an error in predicting the number of packets and hence symbols lost whose mean is 0.94 with a variance of 2.24. The results are summarized in Table 2.

Table 3 presents the utilization of the *all* sequences at different data rates and proxy parameters. BW (A-O) stands for the percentage increase in data rate between the original (O) and adaptive (A) cases, BW (A-NA) denotes the percentage increase in data rate between the adaptive (A) and nonadaptive (NA) data, and BW (NA-O) indicates the percentage increase in data rate between the nonadaptive (NA) and original (O) cases. As can be seen, the additional bandwidth used decreases as the span sizes increases. Based on the estimated packet loss, the server tries to change the message length K according to the rule N-TP/S. Since the code words are written column by column into the interleaving block and read out row by row, each packet

Table 2: Comparison of the performance for all at data rates 32, 64, and 128 kbps at packet losses of 4.9%, 10.07%, 18.63%, and 15.66%, respectively.

Sequence, rate (kbps), packet loss rate (%)	PSNR gain of adaptive over nonadaptive	Loss recovery of adaptive	Loss recovery of nonadaptive
all, 32, 4.9	12.54%	17.39%	8.70%
all, 32, 10.07	62.49%	16.88%	7.79%
all, 64, 18.63	31.10%	18.80%	6.84%
all, 128, 15.66	26.56%	10.53%	4.14%

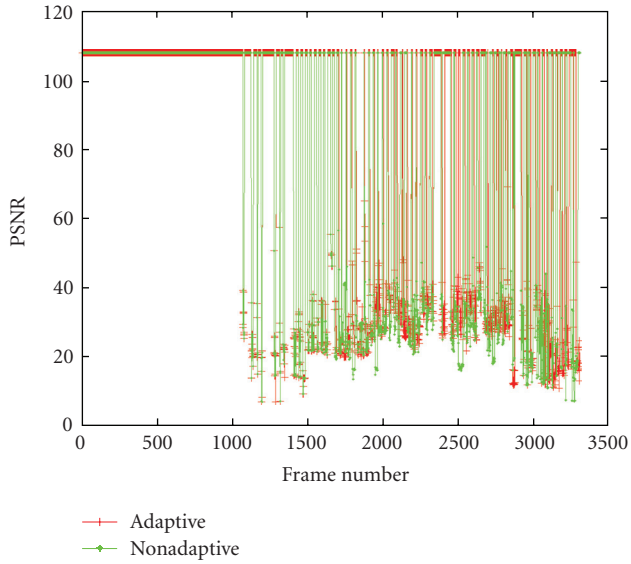

Figure 15: PSNR comparisons for "*all*" sequence at 128 kbps, random delay 0.012, buffer size 500, packet size 256, span 8, PL rate 15.66%.

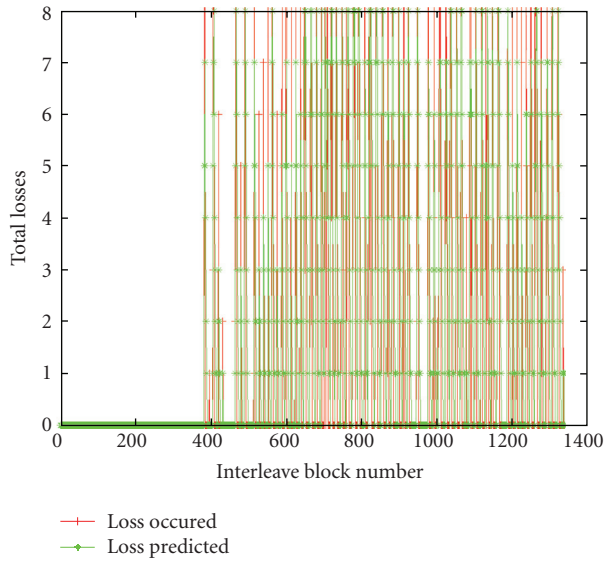

Figure 17: Performance of prediction for "*all*" sequence at 128 kbps, random delay 0.012, buffer size 500, packet size 256, span 8, PL rate 15.66%.

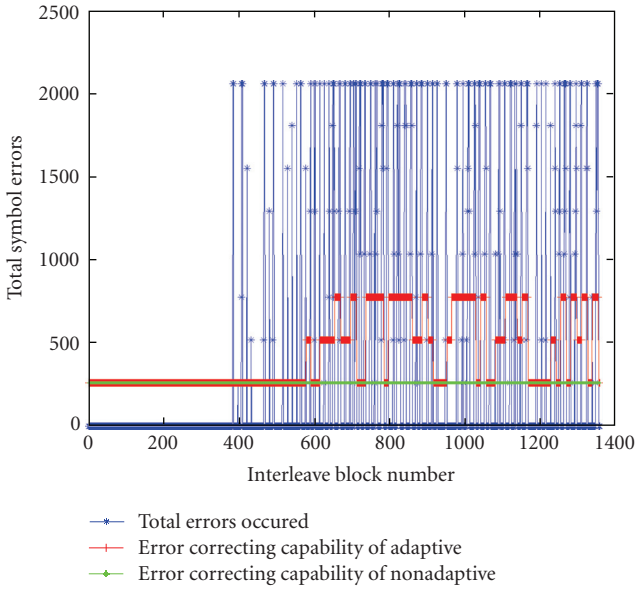

Figure 16: Performance of feedback for "*all*" sequence at 128 kbps, random delay 0.012, buffer size 500, packet size 256, span 8, PL rate 15.66%.

corresponds to P/S rows in the interleaving block. In order to correct a single packet, there should be P/S parity symbols, however, the value P/S decreases with an increase in span size. As the span size increases, the number of rows a packet occupies in the interleaved block decreases, and consequently the number of parity symbols used to correct a packet decreases. This would decrease the value of K indicating that for the same packet size less protection is required to recover a packet when the span size increases, which in turn results in less bandwidth utilization when the span sizes are increased.

Table 4 summarizes the performance of the adaptive error resilience strategy versus the nonadaptive error strategy for different data rates, delays, packet sizes, buffer sizes, span, and packet loss rates. The last column in Table 4 indicates the improvement in PSNR when using the adaptive versus the nonadaptive method. It can be observed that the adaptive algorithm works better at packet losses higher than 6%. In many cases the number of errors fixed by the adaptive strategy was twice as much as those fixed by the nonadaptive one. At lower packet losses, estimating future packet losses was not very effective, and the performance of both the adaptive and nonadaptive was close. It was also observed that the server was not able to effectively estimate

Table 3: Bandwidth utilization of QCIF sequences at different proxy parameters.

Sequence (bit rate in kbps)	Delay	Packet size	Buffer size	Span	Packet loss (%)	BW (A-O) (%)	BW (A-NA) (%)	BW (NA-O) (%)
all (32)	0.0260	256	500	16	4.90	13.11	2.22	10.65
all (32)	0.0280	256	175	8	10.07	26.35	10.42	14.44
all (32)	0.0350	128	175	16	16.57	14.95	3.88	10.65
all (32)	0.0155	128	500	8	17.54	29.17	12.60	14.44
all (64)	0.0209	256	600	8	6.11	21.97	3.01	18.40
all (64)	0.0200	128	600	16	10.21	19.05	7.62	10.61
all (64)	0.0200	256	250	16	18.63	27.98	15.24	10.61
all (64)	0.0200	256	250	8	14.86	32.00	11.50	18.40
all (128)	0.0200	256	500	8	10.45	27.22	7.40	18.44
all (128)	0.0135	128	1000	16	17.86	16.92	5.61	10.64
all (128)	0.0120	256	500	8	15.66	30.80	10.51	18.44
all (128)	0.0120	256	1000	16	5.25	13.32	2.44	10.62

Table 4: Performance comparisons for *all* at different rates and proxy settings.

Sequence (Bit rate in kbps)	Delay	Packet size	Buffer size (in packets)	Span	Packet loss (%)	Gain in PSNR (%)
all (32)	0.0260	256	500	16	4.90	12.54
all (32)	0.0280	256	175	8	10.07	62.49
all (32)	0.0350	128	175	16	16.57	37.34
all (32)	0.0155	128	500	8	17.54	37.92
all (64)	0.0209	256	600	8	6.11	16.48
all (64)	0.0200	128	600	16	10.21	30.84
all (64)	0.0200	256	250	16	18.63	31.1
all (64)	0.0200	256	250	8	14.86	20.32
all (128)	0.0200	256	500	8	10.45	23.6
all (128)	0.0135	128	1000	16	17.86	26.83
all (128)	0.0120	256	500	8	15.66	25.56
all (128)	0.0120	256	1000	16	5.25	6.34

sudden high losses that were preceded and followed by continuous low losses. Furthermore, it can be seen from Table 4 that as the packet sizes increased, the performance of both strategies decreased. This can be attributed to the following: as the packet sizes decreased, the threshold values T_1 and T_2 increased. Larger T_1 and T_2 values lead to better error recovery, as the server can change the value of K over a wide range depending on the range of T_2–T_1. Finally, it was observed that using $c = 1$ resulted in under prediction and using $c = 2$ resulted in over protection. This is because $c = 2$ results in larger estimate of error, and hence the server increases its protection levels in anticipation of many errors.

5. Conclusion

This paper described an adaptive system for error resilient transmission of compressed video over data networks. The system utilizes feedback information sent by the client to the server to enable the server to predict future channel conditions and consequently update the level of protection given to the data. The objective was to efficiently utilize bandwidth. It was observed that adapting the protection level performed much better than using nonadaptive forward error protection. The paper also described an emulation system used to emulate a data network. The system used three computers, one dedicated to be a server, another to serve as the client, and the third (the proxy) to play the role of a data network. The proxy was used to emulate randomly delayed data packets as well as to drop them in the event of buffer overflow.

References

[1] G. J. Sullivan and T. Wiegand, "Video compression-from concepts to the H.264/AVC standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, 2005.

[2] J. L. Black, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*, Chapman & Hall, New York, NY, USA, 1996.

[3] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*, Chapman & Hall, New York, NY, USA, 1997.

[4] R. Koenen, F. Pereira, and L. Chiariglione, "MPEG-4: context and objectives," *Signal Processing: Image Communication*, vol. 9, no. 4, pp. 295–304, 1997.
[5] ISO/IEC 144962-2, "Overview of the MPEG-4 Standard," ISO, 1997.
[6] K. R. Rao and J. J. Hwang, *Techniques and Standards for Image, Video and Audio Coding*, Prentice-Hall, Upper Saddle River, NJ, USA, 1996.
[7] ITU-T, "Draft ITU-T Recommendation H.263 Version 2: Video Coding for Low Bitrate Communication," International Telecommunication Union, 1997.
[8] G. Côté, B. Erol, M. Gallant, and F. Kossentini, "H.263+: video coding at low bit rates," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 849–866, 1998.
[9] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.
[10] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, 1998.
[11] T. Wiegand, N. Färber, K. Stuhlmüller, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1050–1062, 2000.
[12] T. Stockhammer, H. Jenkač, and C. Weiss, "Feedback and error protection strategies for wireless progressive video transmission," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 465–482, 2002.
[13] O. Harmanci and A. M. Tekalp, "A stochastic framework for rate-distortion optimized video coding over error-prone networks," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 684–697, 2007.
[14] Y. Zhang, W. Gao, Y. Lu, Q. Huang, and D. Zhao, "Joint source-channel rate-distortion optimization for H.264 video coding over error-prone networks," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 445–454, 2007.
[15] T. Stockhammer, H. Jenkač, and G. Kuhn, "Streaming video over variable bit-rate wireless channels," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 268–277, 2004.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2009, Article ID 186960, 12 pages
doi:10.1155/2009/186960

Research Article

Statistical Time-Frequency Multiplexing of HD Video Traffic in DVB-T2

Mehdi Rezaei,[1] Imed Bouazizi,[2] and Moncef Gabbouj[1]

[1] *Department of Signal Processing, Tampere University of Technology, 33720 Tampere, Finland*
[2] *Media Laboratory, Nokia Research Center, 33720 Tampere, Finland*

Correspondence should be addressed to Mehdi Rezaei, mehdi.rezaei@ieee.org

Received 27 May 2008; Revised 25 August 2008; Accepted 21 October 2008

Recommended by Susanna Spinsante

Digital video broadcast-terrestrial 2 (DVB-T2) is the successor of DVB-T standard that allows a two-dimensional multiplexing of broadcast services in time and frequency domains. It introduces an optional time-frequency slicing (TFS) transmission scheme to increase the flexibility of service multiplexing. Utilizing statistical multiplexing (StatMux) in conjunction with TFS is expected to provide a high performance for the broadcast system in terms of resource utilization and quality of service. In this paper, a model for high-definition video (HDV) traffic is proposed. Then, utilizing the proposed model, the performance of StatMux of HDV broadcast services over DVB-T2 is evaluated. Results of the study show that implementation of StatMux in conjunction with the newly available features in DVB-T2 provides a high performance for the broadcast system.

1. Introduction

Digital video broadcast-terrestrial 2 (DVB-T2) is going to be a new European Telecommunications Standards Institute (ETSI) standard specification for digital terrestrial television. DVB-T2 is an upgrade of the DVB-T system designed to provide new high quality services. It utilizes advanced techniques that provide more flexibility for the broadcast system. Figure 1 shows an overview of the DVB-T2 system with its main components. The *generic stream encapsulation (GSE)* module encapsulates protocol data units in a protocol-independent manner into GSE packets, which are arranged into the so-called *baseband (BB)* frames by the *input stream processor* module. *Forward error correction (FEC)* encoding is performed at the *bit-interleaved coding* module using a *low-density parity code (LDPC)* concatenated to a BCH code. Subsequent interleaving and mapping to *physical layer (PL)* frames as well as OFDM symbol mapping is performed at the *frame mapper* module. The resulting PL frames are then passed to the *modulator* modules for modulation and transmission. The newly defined modulation modes 64 and 256 QAM and OFDM carrier modes significantly enhance the spectral efficiency; achieving bandwidths of up to 40 Mbps (not accounting for signaling overhead) thus, enabling the broadcast of HDTV services over terrestrial networks.

As yet, another new feature, DVB-T2, utilizes an optional time-frequency slicing (TFS) scheme for data transmission that provides a great flexibility for system design so that a different range of services can be deployed in the system. In this approach, multiple radio frequency (RF) channels are combined into a coherent high-capacity channel to utilize advantages of statistical multiplexing (StatMux) across several high-definition (HD) services. It allows implementing a two-dimensional StatMux over the services to improve the performance of the broadcast system.

In digital communications, video signals are compressed in order to use transmission bandwidth efficiently. In video compression, a video sequence can be encoded at a constant bit rate (CBR) or a variable bit rate (VBR) bit stream. With a similar average bit rate, VBR bit streams consume more resources in terms of transmission bandwidth and delay than CBR bit streams. When encoding a CBR bit stream, a rate controller strictly controls the bit rate mainly by adjusting the quantization parameter (QP). Generally, a CBR can be achieved by large variations in QP and also in video quality. A VBR video bit stream can be produced by encoding a video sequence with or without a rate controller.

In uncontrolled VBR, a constant QP is used for encoding to provide a quasiconstant and better visual quality for compressed video. In controlled VBR, the QP is controlled by a soft rate controller to smooth the variations in the bit rate and also in the quality. Generally, in comparison with CBR, controlled VBR can provide a better visual quality at the expense of more variations in the bit rate. On the other hand, in comparison with uncontrolled VBR, controlled VBR can provide less variation in bit rate at the expense of more variations in the quality.

In video broadcasting, the video sources are encoded to VBR bit streams to provide a better average quality for broadcasted services. However, VBR services need more resources in terms of transmission bandwidth and delay than CBR services. When several VBR video services are broadcasted simultaneously, utilizing StatMux can improve the bandwidth efficiency and end-to-end delay of the broadcast system.

In StatMux, a fixed bandwidth communication channel is shared for transmitting several bit streams. The channel is virtually divided into several variable bandwidth channels that are adapted to the variations in the bit rate of the bit streams. The attempt is to distribute the channel capacity among the bit streams dynamically according to the required bandwidth by the bit streams such that a virtual variable bandwidth channel is allocated to each bit stream.

The performance of StatMux depends on the statistical properties of the multiplexed bit streams as well as the number of bit streams. The statistical properties of video bit streams depend on the encoding parameters such as bit rate, frame rate, and picture size as well as video content and the rate control method. On the other hand, the number of services depend on the service bit rates and the channel capacity. Consequently, the performance of StatMux is application dependent and it should be evaluated specifically for each application. The TFS, introduced in DVB-T2, that allows implementation of StatMux in two dimensions, makes this application more specific. The main goal of this research is to evaluate the performance of StatMux specifically in DVB-T2 by computer simulations. To obtain accurate evaluation results, multiplexing simulations should be repeated many times with different video bit streams. A huge amount of traffic is needed that can be provided synthetically by a video traffic model. The accuracy of the simulation results depends on the accuracy of the model. Therefore, the first attempt is to provide an accurate model for video traffic in this application. Studying statistical properties of HDV traffic, a model for VBR video traffics is proposed in this paper. Then, the proposed traffic model is used to generate synthetic traffic for evaluating the performance of StatMux in DVB-T2.

The rest of this paper is organized as follows. Background information about the VBR video traffic modeling is presented in Section 2. The proposed model for VBR video traffic is presented in Section 3. In Section 4, the performance of StatMux in DVB-T2 is evaluated. Some simulation results are presented in Section 5. The paper is closed with conclusions in Section 6.

2. VBR Video Traffic Modeling

Accurate modeling of VBR video traffic is important in this research. A good model predicts or provides a desired metric or a set of desired metrics for the modeled data similar to the original data. For example, if the packet loss probability is the desired metric, then a good model produces traffic that precisely provides this metric of interest in simulations.

Generally, the performance of a communication network in terms of delay, data drop rate, and bandwidth usage depends on the statistical properties of the traffic in the network. For example, the autocorrelation function (ACF) of the service traffic has a major impact on the performance of communication networks. VBR video traffic was found to exhibit self-similar characteristics [1]. In mathematics, a self-similar object is exactly or approximately similar to a part of itself, for example, the whole has the same shape as one or more of the parts. Self-similarity is a typical property of fractals. A fractal is a rough or fragmented geometric shape that can be subdivided in parts, each of which is (at least approx.) a reduced-size copy of the whole.

The main feature of self-similar processes is that they exhibit long range dependence (LRD), that is, their autocorrelation function $r(k)$ decays less than exponentially fast, and is nonsummable, that is, $r(k) \sim k^{-\beta}$, as $k \to \infty$, for $0 < \beta \leq 1$. The quantity $H = 1 - \beta/2$ is called *Hurst parameter* or *Hurst exponent*. The Hurst exponent was originally developed in hydrology [2]. It shows whether the data is a purely *random walk* or has underlying trends. The Hurst exponent is related to the *fractal dimension* and it is a measure of the smoothness of fractal time series based on the asymptotic behavior of the *rescaled range* of the process. The Hurst exponent is defined as

$$H = \frac{E[\log(R/S)]}{\log(T)}, \tag{1}$$

where T is the duration of the data sample and R/S is the corresponding value of the rescaled range, where S denotes the standard deviation of the sample data and R stands for the difference between the max and min of accumulated deviation from the mean value during the time period T. If $H = 0.5$, the behavior of the time series is similar to a random walk process and samples are uncorrelated. When $H > 0.5$, the time series covers more distance than a random walk. In this case, the process is, namely, *persistent* and samples are positively correlated. This means that if the time series is increasing, it is more probable that it will continue to increase. When $H < 0.5$, the time series covers less distance than a random walk, in which case the process is, namely, *antipersistent* and samples are negatively correlated. This means that if the time series is increasing, it is more probable that it will then decrease, and vice versa.

In communication networks, the Hurst exponent of traffic is relevant to the buffering requirements for traffic transmission. Considering the definition of R in (1), in fact it is equal to the minimum buffering space for perfect transmission of the data during the time period T by a channel with a bandwidth equal to the average of the bit rate. Therefore, the performance of communication

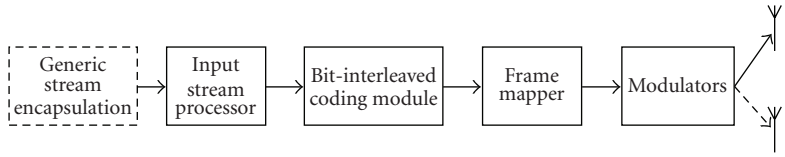

Figure 1: Overview of DVB-T2 system.

networks depends on the statistical properties of traffic such as self-similarity and smoothness. Many video traffic models attempt to capture these relevant statistics.

Several stochastic models for video traffic have been proposed in the past [3]. Maglaris et al. [4] used two models for a video source: a continuous-state autoregressive (AR) Markov model and a discrete-state continuous-time Markov process. Heyman et al. [5] and Lucantoni et al. [6] also used a Markov chain process to develop models for video traffic at the frame level. Grunenfelder et al. [7] used an autoregressive moving-average (ARMA) process to model video conference traffic at ATM cell level. Ramamurthy and Sengupta [8] proposed a hierarchical composite model which uses three processes: two AR processes and one Markov chain. The first AR process attempts to match ACF at short lags while the second attempts to match ACF at long lags. The Markov process captures the effects of scene changes. A combination of the three processes yields the final model. Another hierarchical model was proposed by Heyman and Lakshman in [9] that consists of three different stochastic processes for video scene length, size of the first frame in the scene, and the size of other frames in the scene, respectively. The scene change process was found to be uncorrelated and it was enough to match the distribution of scene length. It was found that the scene length distribution fits Weibull, Gamma, and Pareto distributions. It was also found that the number of ATM cells in a frame of a scene change fits Weibull and Gamma distributions. A Markov chain was used for the frame size within a video scene. Melamed et al. [10] developed a model for video traffic based on Transform-Expand-Sample (TES) process for the number of bits in one group of pictures (GOPs). TES processes are designed to fit simultaneously both the distribution and ACF of the empirical data. Lazar et al. [11] and Reininger et al. [12] used a TES process for modeling of frame and or slice sizes. A process was used for each type of I, P, and B frames (or slices). The final model is composed according to a deterministic structure of the GOP. Garrett and Willinger [13] used a fractional autoregressive integrated moving average (F-ARIMA) process to provide a model for video traffic at the frame level. They used a hybrid distribution which consisted of a concatenation of a Gamma and a Pareto distributions for the frame size distribution. A background sequence is generated by an F-ARIMA process based on a desired value for Hurst exponent and the final sequence is generated by a transformation on the background sequence based on the parameters of the desired distribution. In a similar approach, Huang et al. [14] used an F-ARIMA process to generate background sequences for different frame types based on the value of Hurst exponent. The background sequences were transformed by a weighted sum of exponentials to match the distribution. Kruns and Tripathi [15] proposed a model in which the video scene length is generated by a geometric distribution. The size of I frames is modeled by the sum of two random components: a scene-related component and an AR-2 component that accounts for the fluctuation within the scene. The sizes of P and B frames are modeled by two processes of i.i.d. random variables with Lognormal distributions. The final model is obtained by combining the three submodels according to a given GOP pattern. Liu et al. [16] proposed a video traffic model in which a hybrid Gamma-Pareto distribution is used for all three types of frames and the autocorrelation structure is modeled using two second-order nested AR processes. One AR process is used to generate the mean frame size of the scenes to model the long-range dependence and the other is used to generate the fluctuations within the scene to model the short-range dependence. Sarkar et al. [17] proposed another model for VBR video traffic in which a video sequence is segmented using a classification based on size of three types of video frames. In each class, the frame sizes are produced by a shifted Gamma distribution. Markov renewal processes model video segment transitions. Dai et al. [18] presented a hybrid wavelet framework for modeling VBR video traffic. They modeled the size of I frames in the wave domain and the size of P and B frames based on the intra-GOP correlation. The reviewed models above are samples of different approaches. The review is not exhaustive and some related approaches are not reviewed in this paper.

The proposed models for VBR video traffic in earlier works attempt to fit some statistical properties such as frame size distribution, ACF, and Hurst exponent for sample video traffic data that are encoded for a special application (e.g., video conference) by a particular encoder (e.g., H.263, MPEG-4). Then, the proposed models have been validated based on some practical measures such as data drop rate and delay in buffering simulations.

There are some concerns about the previous proposed traffic models. The first concern is that most of the models have been built based on a limited number of sample real bit streams; therefore, the accuracy of these models is limited to special applications in terms of video content, encoding method, and encoding parameters. The second concern is that, although these models capture some statistical properties of the traffic that may be correlated with practical metrics of interest, the correlation may not be always accurate. For example, it is possible to find bit streams with different Hurst exponents and similar practical performance in terms of data drop rate and delay and also it is possible to find bit streams with similar statistical properties that have different

performances in terms of data drop rate and delay. Some examples are shown in Section 5. The other concern is that all practically possible traffics may not be covered by a model that captures only some statistical parameters. On the other hand, some synthetic bit streams may be generated by the models which is difficult to find a match for them among the real bit streams because some practical constraints that exist on real bit streams are not considered in the models. These concerns affect the accuracy of the simulation results, where synthetic traffics are used.

Considering the concerns above, in a new approach, a model for VBR video traffic is proposed in this paper. In the new approach, the first attempt is to capture the practical metrics of interest, such as buffering parameters, while some statistics are used. The new model is not limited to any special distribution, ACF, or range of Hurst exponent. The new modeling approach simulates the interaction between the video encoder and the video source to generate a synthetic video traffic. The interaction of the encoder with the video source is controlled by a rate controller. Unlike previous modeling approaches in which first statistical properties such ACF are captured to achieve practical properties such as buffering parameters, in the new approach, practical properties are captured directly. The model is tuned similar to a video encoder with a rate controller to generate traffics with desired buffering properties. The practical and statistical properties of a video traffic depend on the video content properties, encoding method, and rate control algorithm. Accordingly, the proposed model can generate various traffics according to the content, for example, sport, movie, news, and so on. Also, it can produce video traffics according to the encoding parameters such as bit rate, frame rate, picture size, and so on. Moreover, it can produce video traffics according to rate control parameters such as buffering delay. These features are beneficial in simulation tasks in which the effects of content properties and encoding parameters on the results of simulation are studied.

From a modeling point of view, the self-similarity properties of VBR video traffic depend on the degree of control that is imposed on the bit rate. While uncontrolled VBR bit streams usually have persistent behaviors with a large Hurst exponent, the controlled VBR bit streams, depending on the degree of control, tend toward the antipersistent case.

We proposed a model for antipersistent video traffic in [19]. A multi-Gamma model was proposed for video frame sizes in which a Gamma distribution is considered for each picture type (e.g., I, P, and B) in each video scene. The proposed model has many parameters to be determined. Considering the functionality of the video rate controller and assuming uniform distributions over some parameters of the model, the final model parameters are reduced to few parameters. Later on, statistics collected from a large video database showed that the assumed uniform distributions should be modified to Gamma distributions. Accordingly, a modified version of the model is presented in [20]. The modified parts of the model are used for the case in which synthetic bit streams are generated without any prototype bit streams. However, the results presented in [19, 20] do not show the effect of these modifications because the models have been validated for the case in which they have been parameterized based on extracted parameters from a prototype bit stream not based on the provided statistics in the modified parts.

The proposed traffic model in this paper is a modified and a generalized form of our previous models. The previous models can generate only antipersistent traffics in which $H < 0.5$, while the new model proposed in this paper can be used for both persistent and antipersistent traffics, that is, the Hurst exponent can assume any value in the range of $0 < H < 1$. The previous models were targeted for controlled VBR with small variations in the bit rate while the new model can be used for a wider range of VBR video including controlled and uncontrolled bit streams with any level of variations in the bit rate. In the previous models, a constant average bit rate is assigned to all video scenes while in the new model video scenes may have different average bit rates that are defined according to a Gamma distribution and also according to the buffering constraint imposed on the bit stream. The self-similarity and LRD properties of video traffics are captured indirectly when the buffering constraint is imposed on the bit stream. Generating synthetic traffic by the proposed model is straightforward with a low degree of complexity.

3. Proposed Model for VBR Video Traffics

A video sequence includes several scenes and each scene includes a number of video frames from different types such as I, P, and B frames According to the proposed model, a Gamma distribution is used for each frame type in each video scene. Note that at the sequence level, each frame type can have a PDF which may be very different from the scene level because the PDFs of video scenes are combined together at the sequence level. In the proposed model, the PDF of each frame type can have any distribution at the sequence level. Although other distributions such as Lognormal may be used at the scene level, the Gamma distribution has been used because it fits well enough the practical results and it simplifies the modeling approach. According to the model, a Gamma PDF for the size of frame (x) of type i in scene s is considered as

$$\text{Gamma}\left(x, k_{is}, \theta_{is}\right) = x^{k_{is}-1} \frac{e^{-x/\theta_{is}}}{\theta_{is}^{k_{is}} \Gamma\left(k_{is}\right)}, \quad x > 0, \tag{2}$$

where $k_{is} > 0$ is the shape parameter and $\theta_{is} > 0$ is the scale parameter of Gamma distribution. i stands for I, P, or B frame type. $s = 1, \ldots, S$ denotes the scene index.

To generate a synthetic video traffic by the proposed model, several parameters should be determined. The main parameters include the total number of frames in the video sequence (N), structure of GOP, that is, the number of P pictures (N_{P}) and B picture (N_{B}) in GOP, the length of video scenes as well as their parameters (k_{is}, θ_{is}), average bit rate (B), frame rate (F), and smoothing buffer size (S_B). To produce synthetic traffics, the main parameters such as $N, N_{\mathrm{P}}, N_{\mathrm{B}}, B, F$, and S_B are set directly by the user whereas the remaining parameters are determined as explained in the sequel.

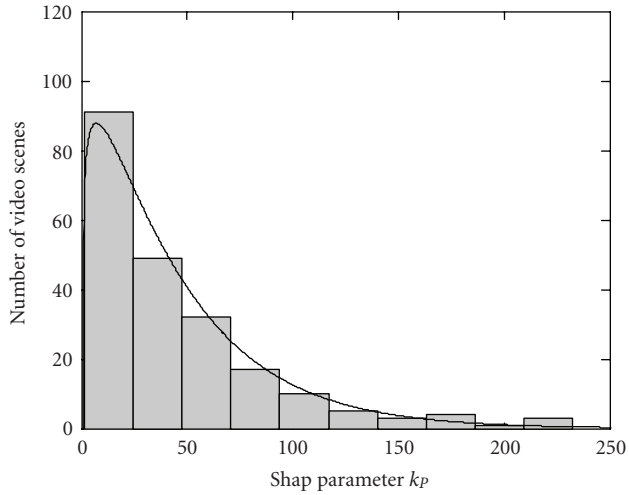

Figure 2: Histogram of shape parameter for Gamma distributions of P frames (k_P) over video scenes that is fitted to a Gamma PDF.

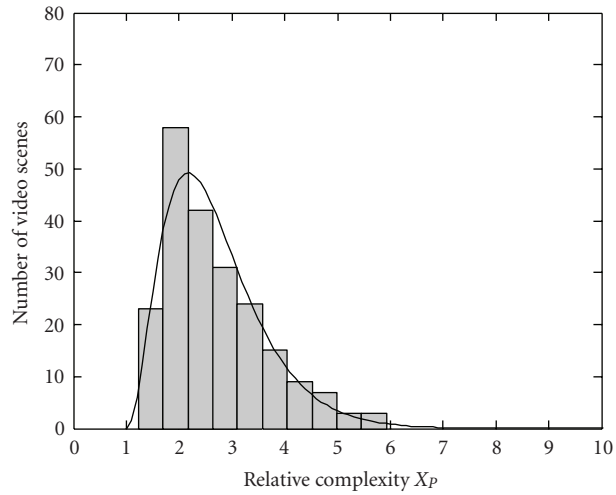

Figure 3: Histogram of the relative complexity X_P over video scenes that is fitted to a Gamma PDF.

Statistics collected from a large video database show that a Gamma PDF can be considered over the length of video scenes as

$$P_{L_s} = \text{Gamma}\,(L_s, k_{L_s}, \theta_{L_s}), \tag{3}$$

where L_s denotes the length of a video scene s. This distribution is used to generate the length of video scenes. The shape and scale parameters (k_{Ls}, θ_{Ls}) are content dependent. Moreover, the statistics show that a Gamma PDF can be assumed also for the shape parameters k_{Is}, k_{Ps}, and k_{Bs} used in (2) over the scenes as

$$\begin{aligned} P_{k_I} &= \text{Gamma}\,(k_I, k_{k_I}, \theta_{k_I}), \\ P_{k_P} &= \text{Gamma}\,(k_P, k_{k_P}, \theta_{k_P}), \\ P_{k_B} &= \text{Gamma}\,(k_B, k_{k_B}, \theta_{k_B}). \end{aligned} \tag{4}$$

These distributions are used to generate the shape parameters of the distributions used in (2). A sample histogram of k_P and its related Gamma PDF are depicted in Figure 2. More details about the collected statistics are presented in Section 5.

As a new measure, *relative coding complexity* is defined. This measure reflects the video content properties as well as the encoding parameters. The relative coding complexity is defined between two picture types. The relative complexity of I to P and I to B pictures in a scene s is defined as $X_{Ps} = \overline{I}_s/\overline{P}_s$ and $X_{Bs} = \overline{I}_s/\overline{B}_s$, respectively, where $\overline{I}_s, \overline{P}_s$, and $\overline{B}_s$ denote the average size of I, P, and B pictures, respectively, in the scene s. The relative coding complexity is a known concept that is used in some control algorithms, for example, in [21, 22]. Experimental results show that the values of relative complexities are not only dependent on the properties of video content such as motion activities but they are also dependent on the encoding parameters, such as bit rate, frame rate, GOP structure, and picture dimensions. Moreover, they are affected by the rate control algorithm and the smoothing buffer size. Statistics from different video sequences which are encoded with similar encoding parameters show that the values of $X_{Ps} - 1$ and $X_{Bs} - 1$ have distributions close to Gamma PDF over the scenes as

$$\begin{aligned} P_{X_P - 1} &= \text{Gamma}\,(X_P - 1, k_{X_P}, \theta_{X_P}), \\ P_{X_B - 1} &= \text{Gamma}\,(X_B - 1, k_{X_B}, \theta_{X_B}). \end{aligned} \tag{5}$$

These distributions are used to generate values for the relative complexities. A sample histogram of X_P collected from real traffics and related Gamma PDF are shown in Figure 3.

To find the remaining parameters, a long-term average bit rate is defined for the video sequence, while it may exhibit a large variation over the scenes. In our previous models presented in [19, 20] for antipersistent traffic, it was assumed that all video scenes in the sequence have a similar average bit rate. To generalize the model, it is assumed that video scenes can have different average bit rates B_s. However, some constraints over the average bit rates of a video scene are imposed to provide a buffering constraint over the final bit stream. Generally, a Gamma PDF can be assumed for the average bit rate of video scenes as follows:

$$P_{B_s} = \text{Gamma}\,(B_s, k_{B_s}, \theta_{B_s}), \tag{6}$$

where the distribution parameters depend on the control strength and smoothing buffer size. Using this distribution, preliminary values for the average bit rates are generated. The preliminary values are modified if the final bit stream should be constrained to a buffering constraint. Consider that the desired bit stream has an average bit rate B over all scenes and it preserves a buffering constraint with buffer size S_B. To achieve the buffering constraint, similar to a rate controller, the following constraints are imposed on the preliminary values of the scene bit rates:

$$0 < \left(\sum_{s=1}^{n} B_s \cdot L_s/F - B/F \cdot \sum_{s=1}^{n} L_s \right) < S_B, \quad n = 1, \ldots, S, \tag{7}$$

where F denotes the video frame rate. The first term in the parenthesis corresponds to the expected value of the overall input to the buffer and the second term corresponds to the overall output from the buffer. Therefore, this condition can guarantee a kind of buffering constraint based on the expected values of scene bit rate. This condition is examined for all n from 1 to S. If it is not met for some values of n, then the value of B_n is corrected by a minimum change such that the condition is met. The resulting bit stream is constrained to an expected buffer size. However, the buffer constraint is not strict because it is imposed based on expected values of scene bit rates. To ensure a strict buffering constraint for the bit stream, margins are considered for the critical buffer conditions and formula (7) is rewritten as

$$M_L S_B F < \left(\sum_{s=1}^{n} B_s L_s - B\sum_{s=1}^{n} L_s\right) < M_H S_B F, \quad n = 1,\ldots,S, \tag{8}$$

where M_L and M_H are two margins (e.g., 0.2 and 0.8) for low and high buffer fullness states, respectively.

For a GOP in a video scene, the average frame size can be estimated as

$$\bar{x} = \frac{\mu_{Is} + N_P\mu_{Ps} + N_B\mu_{Bs}}{1 + N_P + N_B} = \frac{B_s}{F}. \tag{9}$$

From the definition of relative complexity, it is concluded that

$$\mu_{Is} = \mu_{Ps}X_{Ps} = \mu_{Bs}X_{Bs}, \tag{10}$$

where μ_{is} denotes the mean frame size of type i in a video scene s. Combining (9) and (10), the values of μ_{Is}, μ_{Ps}, and μ_{Bs} are obtained for each video scene. For a Gamma distribution, Mean $= k\theta$; and therefore, the scale parameters are obtained as

$$\theta_{Is} = \frac{\mu_{Is}}{k_{Is}}, \qquad \theta_{Ps} = \frac{\mu_{Ps}}{k_{Ps}}, \qquad \theta_{Bs} = \frac{\mu_{Bs}}{k_{Bs}}. \tag{11}$$

The shape parameters have been already generated by (4). Now, all the required parameters for generating the video scenes and the desired bit stream are available.

There are only few parameters that are defined by the user for the model and still they can be reduced. Experimental results show that the model is not very sensitive to the shape parameters used in Gamma distributions (3), (4), and (5) for a relative wide range of bit streams. Therefore, it is enough to consider constant values for $k_{SL}, k_{k_I}, k_{k_P}, k_{k_B}, k_{X_P}$, and k_{X_B} in the model. The user only defines the mean values for L_s, k_I, k_P, k_B, X_P, and X_B then the scale parameters are calculated according to the shape parameters and the mean values by Mean $= k\theta$. Typical values for $k_{SL}, k_{k_I}, k_{k_P}, k_{k_B}, k_{X_P}$, and k_{X_B} are 1.5, 5, 3, 3, 2.5, and 2.5, respectively. The algorithm of generating synthetic video traffics is summarized as follows.

(1) Define the desired encoding parameters including the number of frames (L), the average bit rate (B), the frame rate (F), and the GOP structure (N_P, N_B).

(2) Define the mean values for L_s, k_I, k_P, k_B, X_P, and X_B according to the content and encoding parameters.

(3) Using the mean values of L_s, k_I, k_P, k_B, X_P, and X_B, calculate the scale parameters $\theta_{L_s}, \theta_{k_I}, \theta_{k_P}, \theta_{k_B}, \theta_{X_P}$, and θ_{X_B} according to Mean $= k\theta$.

(4) Using (3), generate S scene length L_s, such that

$$\sum_{s=1}^{S} L_s \geq L. \tag{12}$$

(5) Using (6) and (8), generate the scene bit rates.

(6) Using (5), generate the relative complexities X_{Ps} and X_{Bs}.

(7) Combine (9) and (10), calculate μ_{Is}, μ_{Ps}, and μ_{Bs} for each video scene.

(8) Using (4), generate k_{Is}, k_{Ps}, k_{Bs}.

(9) Using (11), calculate θ_{Is}, θ_{Ps}, and θ_{Bs} for the scenes.

(10) Using (2), generate the frame sizes for each video scene.

4. Performance of StatMux in DVB-T2

In this section, the performance of StatMux in DVB-T2 is evaluated by simulations. In the TFS transmission scheme as defined by DVB-T2, the service data is transmitted as time-frequency slices, that is, time-slice frames that are transmitted by parallel radio channels. The time slices have durations of about a few hundred milliseconds (typically 180 milliseconds) and a number of maximum 6 RF channels can be used for transmission of time-sliced data. Figure 4 shows an example of a TFS frame for 4 RF channels and 15 services. There is a time shift between the services in different RF channels to enable frequency hopping at the receiver. At the beginning of each frame, two synchronizing symbols are inserted (shown as P1 and P2 in the figure). The synchronization symbols allow a receiver to rapidly detect the presence of DVB-T2 signal, as well as to synchronize to the frame. Data related to a number of different services can be statistically multiplexed over the two dimensions of time and frequency. Performance of StatMux in DVB-T2 depends on the bandwidth of the coherent transmission channel, the number of multiplexed services, and the statistical properties of service traffics. A set of comprehensive simulations were performed to evaluate the performance of StatMux of HDV services over DVB-T2.

4.1. Simulations. To evaluate the performance of StatMux in DVB-T2, StatMux is compared with deterministic multiplexing (DetMux) in which a fixed bandwidth is allocated to each service. To provide accurate results, the multiplexing simulations were performed as close as possible to a real system. Service traffics were generated with parameters similar to typical real traffics and typical values were selected for simulation parameters. According to the simulation, for each service, video frames are packetized into protocol data unit (PDU) and then GSE packets [23]. BB frames are formed from the GSE packets and FEC parity check data with a code rate of 1/4 were added [24]. BB frames are

RF 1	RF 2	RF 3	RF 4
15	11	7	Service 3
14	10	6	Service 2
13	9	5	Service 2
12	8	4	Service 1
11	7	Service 3	Service 1
10	6	Service 2	Service 1
9	5	Service 2	15
8	4	Service 1	14
7	Service 3	Service 1	13
6	Service 2	Service 1	12
5	Service 2	15	11
4	Service 1	14	10
Service 3	Service 1	13	9
Service 2	Service 1	12	8
Service 2	15	11	7
Service 1	14	10	6
Service 1	13	9	5
Service 1	12	8	4
P2	P2	P2	P2
P1	P1	P1	P1

Figure 4: Example of a TFS frame for 4 RF channels and 15 services.

buffered in the service buffers. In a real system, convolutional interleaving is performed on BB frames, that is not essential for the multiplexing performance and, hence, it is not implemented in the simulations. Multiplexing simulations are performed over the BB frames stored in the service buffers. Detailed simulation parameters are presented in Section 5. Multiplexing algorithms are explained in the sequel.

4.2. Multiplexing Algorithms. In an ideal case of StatMux, the available bandwidth is distributed between the services proportional to their temporal required bandwidth. A multiplexing algorithm was used in the simulation that performs close to the ideal case. According to the method used, the TFS frames are formed such that the number of allocated BB frames to each service is proportional to the amount of stored BB frames in the service buffer. As a simple case, consider the case of N services being multiplexed that have similar average bit rates and each TFS frame carries B_{TFS} number of BB frames. When forming a TFS frame, if the service buffers contain $B_1, B_2, \ldots,$ and B_N number of BB frames, $b_1, b_2, \ldots,$ and b_N number of BB frames from services $1, 2, \ldots,$ and N, respectively, are used for forming the TFS frame such that

$$b_i = \frac{B_{\text{TFS}} B_i}{\sum_{j=1}^{N} B_j}, \quad i = 1, 2, \ldots, N. \tag{13}$$

In a general case in which the multiplexed services have different average bit rates, the buffer occupancies are normalized to the average service bit rates as

$$b_i = \frac{B_{\text{TFS}} B_i / R_i}{\sum_{j=1}^{N} B_j / R_j}, \quad i = 1, 2, \ldots, N, \tag{14}$$

where R_i denotes the average bit rate of the ith service.

In the simulation of DetMux, TFS frames are formed such that a fixed number of BB frames is allocated to each service in all TFS frames. Details of simulation parameters are presented in Section 5.

5. Simulation Results

Some simulation results are presented in this section that can be divided into two parts. The first part is related to the proposed video traffic modeling approach and the validation of the model. The second part of the results presents the performance of StatMux in DVB-T2.

To collect some statistics form real video bit streams, a comprehensive study on a large set (40 sequences) of long (about 2500 to 5000 frames per sequence) HDV sequences was performed. After a preliminary study, a number of 25 HDV sequences with a resolution of 1280×720 (720 p) were selected from [25–27]. The selected video sequences, which were encoded with a bit rate higher than 6 MB/s, were decoded and used as source signals when they are again encoded at a bit rate of 6 MB/s in our simulations. The video sequences were encoded several times by the FFMPEG H.264/AVC encoder with different buffering constraints [28]. A VBR rate controller is implemented in FFMPEG encoder that was used in the simulations. Smoothing buffers with sizes corresponding to 0.5, 1, 2, 3, 4, and 10 seconds buffering delay were used for the rate control. Moreover, the sequences were encoded with constant QPs and without any buffering constraint. Various statistics related to the proposed traffics model were collected. These include video scene length, scene bit rate, relative complexity of picture types, shape and scale parameters of the Gamma PDFs, Hurst exponent, minimum buffering delay, variance, and mean of different picture types. The collected statistics formed a rich database that was used for building and parameterizing the proposed traffic model. Few hundred video scenes were used in the simulations. Due to space limitation, the results presented in this section constitute only a small part of collected results.

As sample results, Figure 5 to Figure 10 compare the results of encoding "*The Living Sea*" video sequence in two cases: uncontrolled VBR and controlled VBR cases. The other encoding parameters such as average bit rate, frame rate, and GOP structure are similar for both cases. The fullness of the decoder buffer (with zero buffering period) and the size of the video frames for the two cases are shown in Figures 5 and 8. Histograms of I and P frames are depicted in Figures 6 and 9 for the two cases. Figures 7 and 10 show the ACF of video frames size for the two cases. The figures show that the size of the video frames, the distribution of the video frame size, and the ACF are very different in the two cases. These sample results prove that the statistical properties of VBR video traffics depend on the encoding process. Therefore, the encoding process is considered in the proposed modeling approach.

To show the relation between the statistical properties and the practical parameters of real bit streams, the Hurst exponent and the minimum buffering delay for a number of encoded bit streams were measured and are depicted in Figure 11. This figure shows that some bit streams

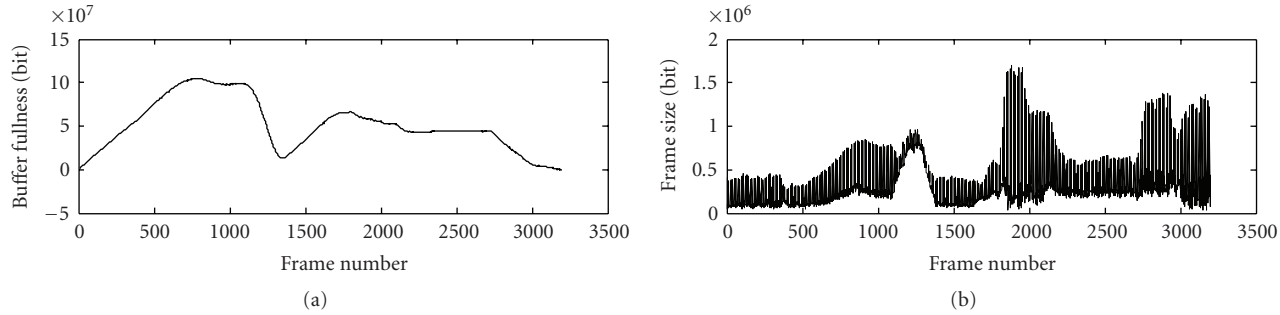

Figure 5: Buffer occupancy and frame size of "The Living Sea" video sequence encoded with a constant QP.

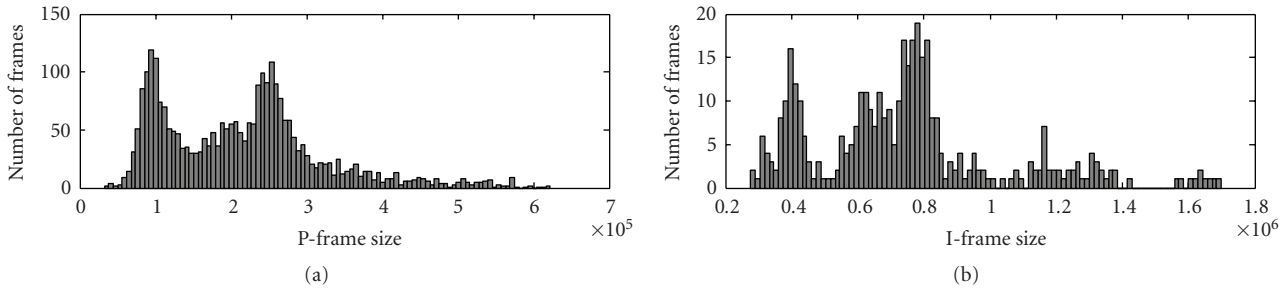

Figure 6: Histograms of P and I frame size in "The Living Sea" video sequence encoded with a constant QP.

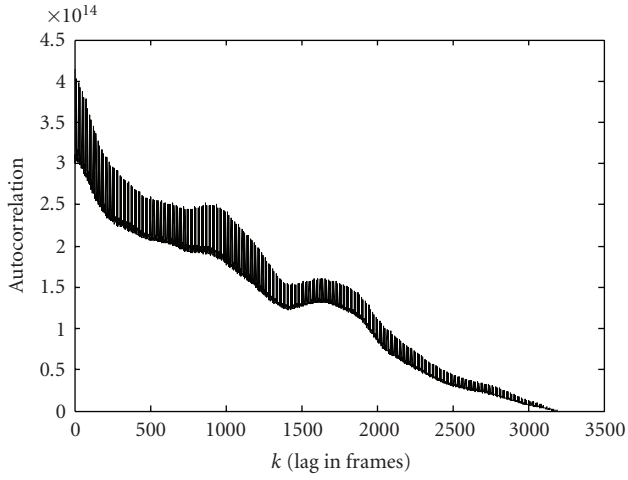

Figure 7: ACF of picture size in "The Living Sea" video sequence encoded with a constant QP.

with similar Hurst exponents have very different buffering requirements and also some bit streams with similar buffering requirements have very different Hurst exponents. Note that there is a tradeoff between buffering requirement and bandwidth in a communication network. Consequently, another important result is that statistical properties may not always reflect the practical parameters and thus previous models that rely only on capturing such statistical properties may not be accurate for estimating practical metrics of interest. The proposed model solves this problem by taking the practical parameters such as encoding parameters and buffering constraints into consideration in the modeling approach.

The proposed traffic model is a modified version of our previous models that were validated successfully in [19, 20]. To validate the multi-Gamma video traffic model proposed in [20], we selected a set of known video sequences including *Foreman, Carphone, Silent, New York,* and *Football* sequences. We repeated and concatenated each of these sequences to provide longer sequences (900 frames) and then the resulting sequences were concatenated again to make a longer video sequence. The fact that the resulting video sequence has several different scenes was suitable for evaluating the model. The video sequence was encoded with a bit rate of 300 kb/s, a frame rate of 30 f/s, and a buffering delay of 0.4 second to produce a prototype video bit stream. The model parameters were extracted based on the prototype bit stream and a synthetic sequence was generated by the proposed model. The prototype and the synthetic traffics were compared by several measures including histogram, ACF, Hurst exponent, and buffering requirements. The simulation results presented in [20] show that the multi-Gamma model can generate synthetic bit streams close to the prototype real bit streams when they are parameterized according to the prototypes. The modifications of the model are related to the case in which the synthetic traffics are generated without the use of any prototype. Therefore, the validation results presented in [20] are not repeated in this paper and only the modified part of the model is validated. Generating the multi-Gamma model parameters is part of these modifications. The collected statistics from the real bit streams show that the Gamma distributions can be fitted to the shape and the scale parameters of the multi-Gamma

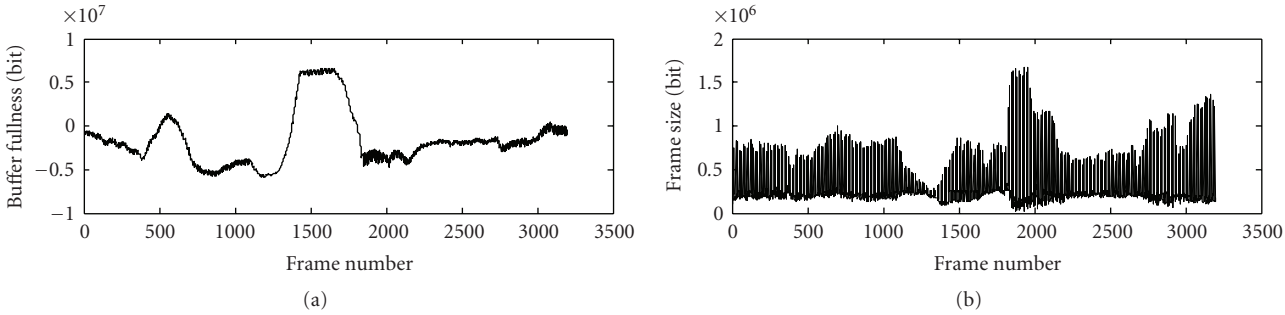

Figure 8: Buffer occupancy and frame size of "The Living Sea" video sequence encoded with FFMPEG VBR video rate controller.

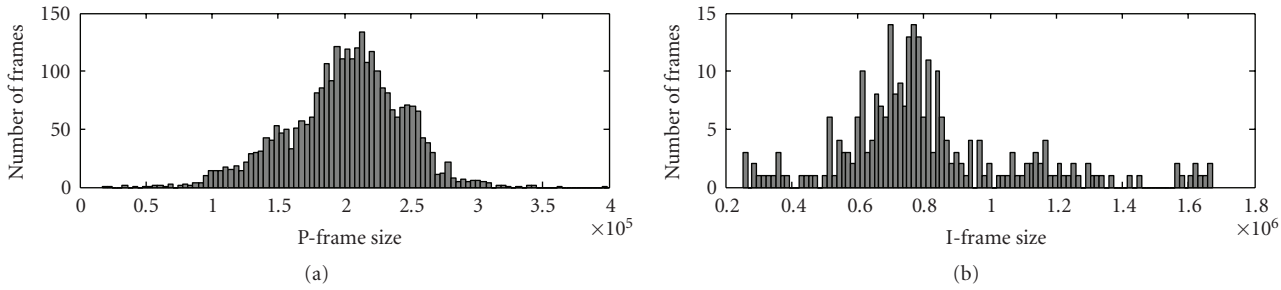

Figure 9: Histograms of P and I frame size in "The Living Sea" video sequence encoded by FFMPEG VBR video rate controller.

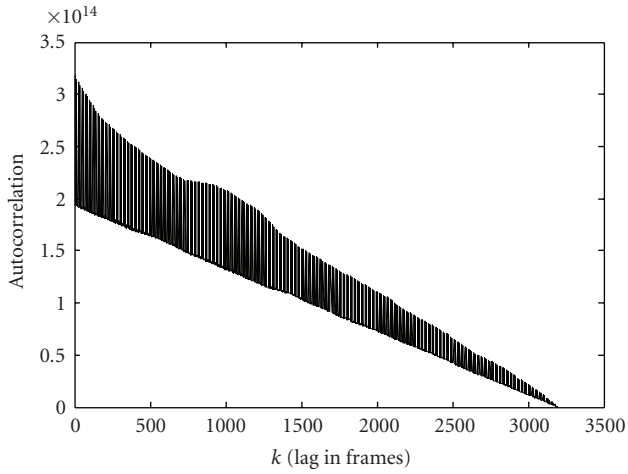

Figure 10: ACF of picture size in "The Living Sea" video sequence encoded with FFMPEG VBR video rate controller.

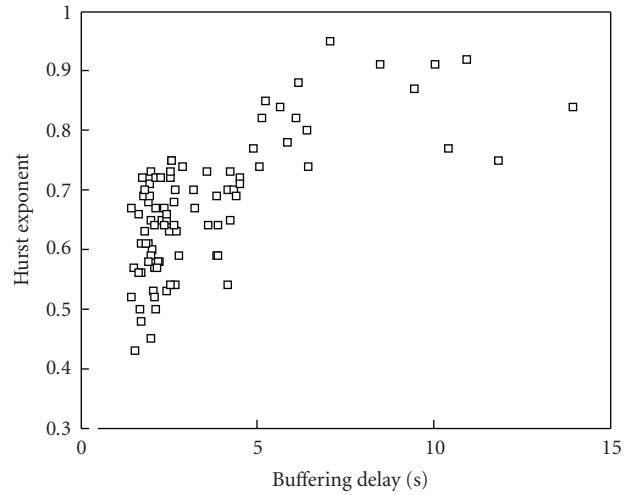

Figure 11: Hurst exponent versus buffering delay in real traffics.

model over the video scenes. Figure 2 shows the histogram of the shape parameter of the Gamma distributions of the P frames over the scenes (k_P) as a sample. As shown, a Gamma PDF is fitted to the histogram. Moreover, the statistics show that other Gamma distributions can be fitted to the relative complexities X_P and X_B over the video scenes. Figure 3 depicts the histogram of the relative complexity X_P over the scenes and also a Gamma PDF that is fitted to the histogram. The number of parameters that need to be determined for the multi-Gamma model is proportional to the number of scenes in a video sequence. The above results are used to generate the parameters of the multi-Gamma model by only few other Gamma distributions each defined by only two parameters. In fact, we model the parameters of the multi-Gamma model to decrease the number of parameters that is required for generating synthetic traffic. Another part of the modification of the model is related to the range of operation that is validated below.

The model has been modified to generate bit streams with a wide range of statistics and practical metrics of

interest. To validate the proposed model in a wide range of operation, the model was parameterized to generate synthetic video bit streams with different buffering constraints. Buffer sizes corresponding to target maximum buffering delays of 1 to 15 seconds were used in the model. For each buffer size or target delay, a number of 20 bit streams, each including 3000 frames, with a bit rate of 6 MB/s were generated. Values of $100, 180, 46, 46, 3$, and 4.3 were used for mean of L_s, k_I, k_P, k_B, X_P, and X_B, respectively, as user-defined parameters. Buffering simulations were performed on the bit streams and the minimum (over the frames) buffering delay for zero data drop rate was measured for each bit stream. The measured values have been compared with the target maximum buffering delay in Figure 12. As shown, the maximum (over 20 samples) delay obtained is close to the target maximum delay in different operating points or target delays. Moreover, delays obtained for 20 samples in each target delay have been distributed below and are close to the maximum values. This is very similar to real conditions in which the encoded bit streams by a rate controller may not use the whole available range of the buffer space. When generating the bit streams above, only the sizes of buffer S_B and k_{B_s} were changed for different operating points and all other parameters were kept fixed. Simulation results show how well synthetic bit streams are in conformance with the desired practical constraints. Previous traffics models are usually validated by comparing the performance of real and modeled traffics in term of data drop rate in a buffering delay. In the simulation above, we consider the performance of modeled traffics in term of minimum delay for the zero data drop rate case which is a fixed practical reference point. This is beneficial from two points of view. First, when the model is used for simulation of StatMux in DVB-T2, we are interested in the zero drop rate case. Second, when in practice a video sequence is encoded, it is encoded with a buffering constraint for a zero data drop rate not for a target nonzero drop rate. However, the proposed model can be easily tuned by k_{B_s} for a target nonzero data drop rate and a given delay. Therefore, the proposed model can be tuned similar to a video encoder and a video rate controller. This is a great advantage of the proposed model.

The above results show that the proposed model can provide practical metrics of interest for the synthetic traffics with a wide range of statistics. To assess the range of statistics of the metrics, for the generated bit streams explained above, the Hurst exponents were computed and are depicted in Figure 13. An approximate exponential function between the buffering delay and Hurst exponent can be considered over the results. This is in conformance with collected statistics from real bit streams depicted in Figure 11. Using the approximate exponential function, the model can be tuned to generate bit streams with a target Hurst exponent in the whole range. Note that our previous models are valid only for $H < 0.5$ while the new model is valid for $0 < H < 1$.

To evaluate the performance of StatMux over video broadcast services in DVB-T2, the proposed traffic model was parameterized to generate synthetic traffics corresponding to HDV contents with 3000 frames, 6 MB/s, and with a GOP structure as "I B P B P B P B P B P B".

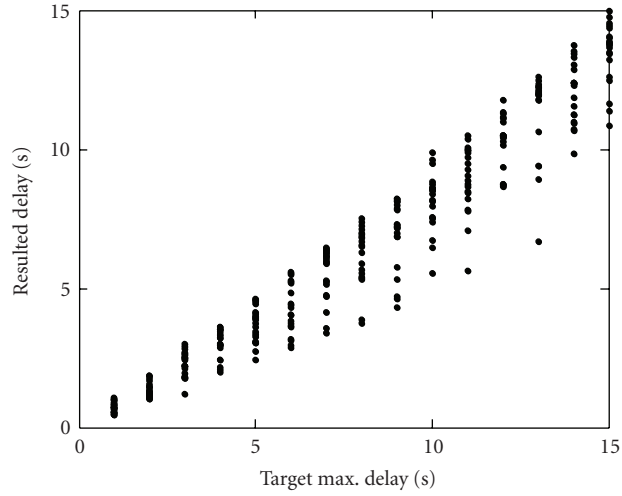

Figure 12: Resulted delay for modeled traffics versus target max delay.

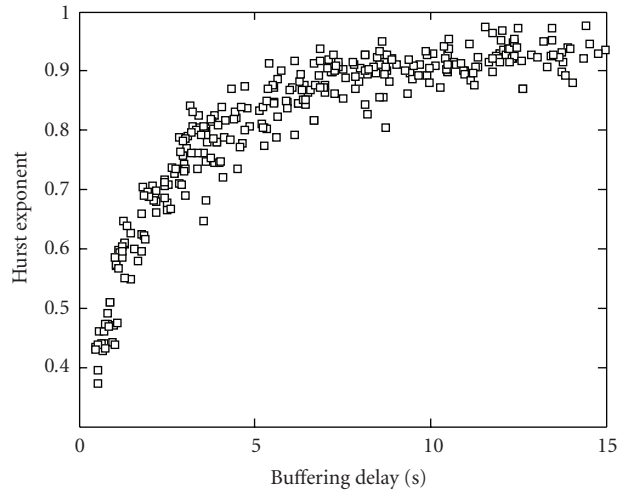

Figure 13: Hurst exponent versus buffering delay in modeled traffics.

Table 1: Gain of StatMux in different buffering delays when the bit streams are constrained to a buffering delay of 3 Seconds.

Delay "ms"	Bandwidth utilization		Gain in BW %
	DetMux	*StatMux*	
26	50.6	80.0	58.10
42	54.1	85.0	57.11
61	57.2	90.0	57.34
105	62.6	95.0	52.73
200	67.2	95.6	42.26

The model was tuned to generate traffics with different buffering constraints including 4, 5, and 7 seconds target buffering delays. Multiplexing simulations were performed over the synthetic video bit streams as explained in Section 4. 6 RF channels were considered in the simulations. The

Table 2: Gain of StatMux in different buffering delays when the bit streams are constrained to a buffering delay of 5 Seconds.

Delay "ms"	Bandwidth utilization		Gain in BW %
	DetMux	*StatMux*	
33	47.1	80.0	69.85
57	51.2	85.0	66.02
86	54.7	90.0	64.53
105	58.6	92.2	57.34
232	61.8	95.0	53.72

Table 3: Gain of StatMux in different buffering delays when the bit streams are constrained to a buffering delay of 7 Seconds.

Delay "ms"	Bandwidth utilization		Gain in BW %
	DetMux	*StatMux*	
44	43.1	80.0	85.61
86	46.7	85.0	82.01
170	50.6	90.0	77.87
250	52.9	91.2	72.40
501	58.0	95.0	63.79

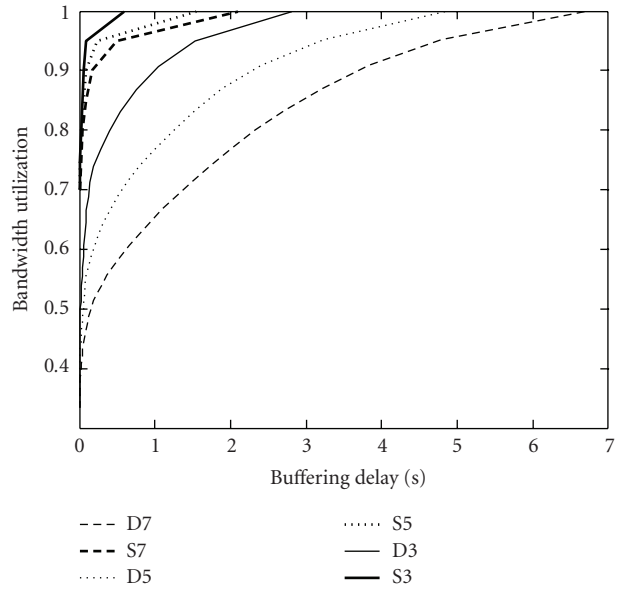

Figure 14: Bandwidth utilization versus buffering delay for StatMux and DetMux.

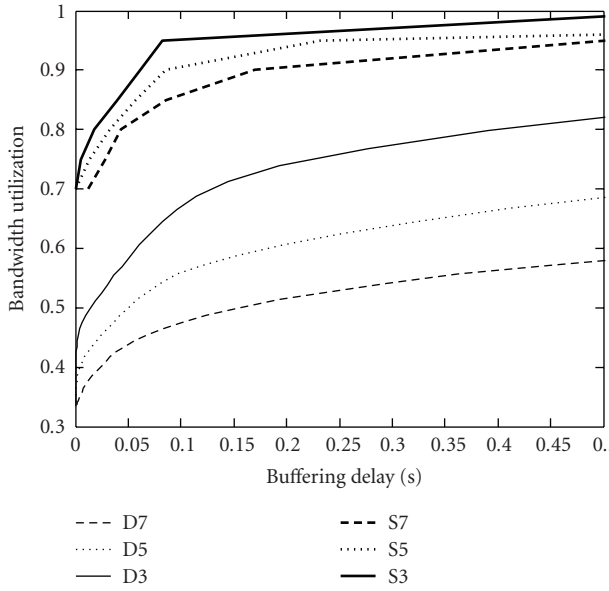

Figure 15: Bandwidth utilization versus buffering delay for StatMux and DetMux in low-delay operating area.

performance of StatMux was compared with the performance of DetMux at several operating points in a two-dimensional space of bandwidth utilization and delay. For each operating point, two simulations corresponding to StatMux and DetMux were performed on a number of 14 to 20 bit streams. To provide different operating points, the number of multiplexed services has been changed from 14 to 20 while the transmission bandwidth was kept constant. To get statistically acceptable results, for each operating point, the simulations were repeated 5 times. The whole procedure above was repeated 3 times to get 3 performance curves corresponding to the bit streams with 3 different buffering constraints. The performance curves are depicted in Figures 14 and 15. The bandwidth utilization is depicted as a function of buffering delay in StatMux and DetMux for three different groups of video bit streams. The groups have different buffering constraints corresponding to $3, 5, 7$ seconds. D3, D5, and D7 in the figures correspond to DetMux while S3, S5, and S7 correspond to StatMux. Figure 15 is a zoomed version of Figure 14 in a low-delay practical operating area. The high delay end points on the curves of DetMux in Figure 14 are very close to the target delays ($3, 5, 7$ seconds) used for generating traffics in the model. This closeness shows that the model performs accurately in different operating points. Moreover, it proves the accuracy of the multiplexing simulations. Sample results from the curves shown in Figure 15 are presented in Tables 1–3. Moreover, the gain of StatMux is presented for 5 operating points in the tables. The gain of StatMux was computed in term of percentage of bandwidth increase with respect to DetMux. According to Table 1, when the bit streams are constrained to a buffering delay of 3 seconds, for a buffering delay between 26 to 200 milliseconds, a gain of 42–58% increase in bandwidth is expected. Table 2 shows that when the bit streams are constrained to a buffering delay of 5 seconds, for a buffering delay between 33 to 232 milliseconds, a gain of 54–70% increase in bandwidth is expected. According to Table 3, when the bit streams are constrained to a buffering delay of 7 seconds, for a buffering delay between 44 to 501 milliseconds, a gain of 64–86% increase in bandwidth is expected. Simulation results show that using StatMux in DVB-T2 can considerably improve the bandwidth efficiency and end-to-end delay of a broadcast system.

From a video quality point of view, a typical buffering constraint about 5 seconds is large enough to allow a

quasiconstant quality for an encoded video. According to the results presented in Table 2 for such high-quality bit streams, a bandwidth efficiency of 95% can be achieved with only a buffering delay of 0.23 second by StatMux.

6. Conclusions

A model for variable bit rate video traffics was proposed that can generate a wide range of synthetic video bit streams with practical and statistical metrics of interest. The proposed model was validated successfully and was used to study the performance of statistical multiplexing of HDV services in a DVB-T2 broadcast system by computer simulations. Simulation results showed that the TFS introduced in DVB-T2 in conjunction with StatMux can provide a high performance in terms of bandwidth efficiency, end-to-end delay, and video quality for the broadcast system.

Acknowledgment

This work was partially supported by Nokia and the Academy of Finland, Project no. 213462 (Finnish Centre of Excellence program 2006–2011).

References

[1] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 1566–1579, 1995.

[2] H. E. Hurst, R. P. Black, and Y. M. Simaika, *Long-Term Storage: An Experimental Study*, Constable, London, UK, 1965.

[3] M. R. Izquierdo and D. S. Reeves, "Survey of statistical source models for variable-bit-rate compressed video," *Multimedia Systems*, vol. 7, no. 3, pp. 199–213, 1999.

[4] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance models of statistical multiplexing in packet video communications," *IEEE Transactions on Communications*, vol. 36, no. 7, pp. 834–844, 1988.

[5] D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical analysis and simulation study of video teleconference traffic in ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 1, pp. 49–59, 1992.

[6] D. M. Lucantoni, M. F. Neuts, and A. R. Reibman, "Methods for performance evaluation of VBR video traffic models," *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 176–180, 1994.

[7] R. Grunenfelder, J. P. Cosmas, S. Manthorpe, and A. Odinma-Okafor, "Characterization of video codecs as autoregressive moving average processes and related queueing system performance," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 3, pp. 284–293, 1991.

[8] G. Ramamurthy and B. Sengupta, "Modeling and analysis of a variable bit rate video multiplexer," in *Proceedings of the 11th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '92)*, vol. 2, pp. 817–827, Florence, Italy, May 1992.

[9] D. P. Heyman and T. V. Lakshman, "Source models for VBR broadcast-video traffic," *IEEE/ACM Transactions on Networking*, vol. 4, no. 1, pp. 40–48, 1996.

[10] B. Melamed, D. Raychaudhuri, B. Sengupta, and J. Zdepski, "TES-based video source modeling for performance evaluation of integrated networks," *IEEE Transactions on Communications*, vol. 42, no. 10, pp. 2773–2777, 1994.

[11] A. A. Lazar, G. Pacifici, and D. E. Pendarakis, "Modeling video sources for real-time scheduling," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '93)*, vol. 2, pp. 835–839, Houston, Tex, USA, November 1993.

[12] D. Reininger, B. Melamed, and D. Raychaudhuri, "Variable bit-rate MPEG video: characteristics, modeling and multiplexing," in *Proceedings of the 14th International Teletraffic Congress (ITC)*, pp. 295–306, Antibes Juan-les-Pins, France, June 1994.

[13] M. Garrett and W. Willinger, "Analysis, modaling and generation of self-similar VBR video traffic," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 269–280, 1994.

[14] C. Huang, M. Devetsikiotis, I. Lambadaris, and R. Kaye, "Modeling and simulation of self-similar variable bit-rate compressed video: a unified approach," *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 4, pp. 114–125, 1995.

[15] M. Krunz and S. K. Tripathi, "On the characterization of VBR MPEG streams," in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, vol. 25, pp. 192–202, Seattle, Wash, USA, June 1997.

[16] D. Liu, E. I. Sára, and W. Sun, "Nested auto-regressive processes for MPEG-encoded video traffic modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 2, pp. 169–183, 2001.

[17] U. K. Sarkar, S. Ramakrishnan, and D. Sarkar, "Modeling full-length video using Markov-modulated gamma-based framework," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 638–649, 2003.

[18] M. Dai, D. Loguinov, and H. Radha, "A hybrid wavelet framework for modeling VBR video traffic," in *Proceedings of the International Conference on Image Processing (ICIP '04)*, vol. 5, pp. 3125–3128, Singapore, October 2004.

[19] M. Rezaei, I. Bouazizi, and M. Gabbouj, "A model for controlled VBR video traffic," in *Proceedings of the IEEE International Conference on Signal Processing and Communications (ICSPC '07)*, Dubai, UAE, November 2007.

[20] M. Rezaei, I. Bouazizi, and M. Gabbouj, "Generating antipersistent VBR video traffic," in *Proceedings of the Picture Coding Symposium (PCS '07)*, p. 6, Lisbon, Portugal, November 2007.

[21] ISO/IEC JTC/SC29/WG11/N0400 MPEG93/457, Test Model 5, TM5, April 1993.

[22] M. Rezaei, M. M. Hannuksela, and M. Gabbouj, "Semi-fuzzy rate controller for variable bit rate video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 633–644, 2008.

[23] ETSI, "Generic Stream Encapsulation (GSE) Protocol," *ETSI standard*, DVD Document A116, May 2007.

[24] ETSI, "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)," *ETSI standard*, EN 302 307, V1.1.2, Jun 2006.

[25] Microsoft, http://www.microsoft.com/windows/windowsmedia/musicandvideo/hdvideo/contentshowcase.aspx.

[26] HD-Channel, http://www.hd-channel.com.

[27] http://www.highdefforum.com/showthread.php?t=6537.

[28] http://ffmpeg.mplayerhq.hu.