

Linear Rate Control and Optimum Statistical Multiplexing for H.264 Video Broadcast

Zhihai He, *Senior Member, IEEE*, and Dapeng Oliver Wu, *Senior Member, IEEE*

Abstract—The H.264 video coding standard achieves significantly improved video compression efficiency and finds important applications in digital video broadcast. To enable H.264 video encoding for digital TV broadcast and maximize its broadcast efficiency, there are two important issues that need to be adequately addressed. First, we need to understand the complex coding mechanism of an H.264 video encoder and develop a model to analyze and control its rate-distortion (R-D) behavior in an accurate and robust manner. Second, the R-D behaviors of individual channels in the broadcast system should be jointly controlled and optimized under bandwidth and buffer constraints so as to maximize the overall broadcast quality. In this paper, we develop a linear rate model and a linear rate control scheme for H.264 video coding. We develop an optimum statistical multiplexing system to allocate bits across video programs (each being encoded by an H.264 encoder) and video frames so that the overall video broadcast quality is maximized. We study the bandwidth and buffer constraints in video broadcast and formulate the optimum statistical multiplexing into a constrained mathematical optimization problem. Realizing that it is impossible to find a close-form solution for global optima, we propose a simple yet efficient algorithm to find a near-optimum solution for joint rate allocation under buffer constraints. Our extensive simulation results demonstrate that the proposed statistical multiplexing system achieves about 40–50% saving in bandwidth, provides a smooth video quality change across programs and frames, and maintains robust decoder buffer control.

Index Terms—H.264 video coding, rate control, statistical multiplexing, optimum rate allocation, TV broadcast.

I. INTRODUCTION

INTERNATIONAL standards for video coding, such as MPEG-2, H.263, and MPEG-4, have been the enabling technologies for a wide range of multimedia applications [1]–[3]. The H.264 video coding standard developed jointly by ITU-T VCEG (Video Coding Experts Group) and ISO-IEC MPEG targets at significantly improved coding efficiency as well as other important features, such as error resilience and network friendliness [4]. To enable H.264 video encoding for digital TV broadcast and maximize its broadcast efficiency, there are two important issues that need to be adequately addressed. First, we need to understand the complex coding mechanism of H.264 video encoding and develop a model to

analyze and control its rate-distortion behavior in an accurate and robust manner. This problem is called rate-distortion (R-D) analysis and control for H.264 [11], [12], [16]. Second, a digital TV broadcast system often has a number of TV programs, each being encoded by an H.264 video encoder. The H.264 encoding behaviors of individual programs in the broadcast system should be jointly controlled and optimized under bandwidth and buffer constraints so as to maximize the overall video broadcast quality. This problem is called optimum statistical multiplexing [17], [21].

A. Related Work

R-D analysis and control is an essential component in video coding and streaming over networks. The R-D behaviors of previous video coding systems, such as MPEG-2, H.263, and MPEG-4, have been well understood, and many R-D models and control algorithms have been developed in the literature [5], [6], [8], [9]. As a new international standard, the H.264 video coding has introduced many sophisticated coding features and techniques and becomes much more complex than its predecessors [4]. This makes it more difficult to develop a model to analyze and control the R-D behavior of H.264 video encoders accurately and robustly. Recently, a number of rate control algorithms have been developed for H.264 video coding [11]–[13], [15], [16]. For example, in [11], the mean of absolute difference (MAD) of each basic unit in a current frame is estimated by the MAD of the collocated basic unit in its previous frame using a linear model. Then, a quadratic rate model is used to determine the QP of the basic unit. In [12], the residual of each frame is first estimated with a reduced set of intra and inter-prediction modes and the number of reference frames is limited to one. Then, an H.263 TMN8-like R-D model is used to determine a QP. In [16], a two-staged scheme is proposed to address the inter-dependency between R-D optimization and rate control in H.264. In general, these algorithms, especially those that use previous frame statistics for rate control [11], suffer from relatively large errors in rate control or buffer regulation. In addition, some important issues, such as header bits estimation and recursive macroblock coding caused by intra prediction, have not been sufficiently addressed for accurate and robust rate control.

In this work, we propose to develop a simple yet efficient rate control algorithm, called *linear rate control*, for H.264 video coding.

Statistical multiplexing is important for many video applications, including digital TV broadcast, video surveillance, and video conferencing. In a statistical multiplexing system, as illustrated in Fig. 1, multiple video programs/streams are encoded individually, then multiplexed, and transmitted over

Manuscript received November 20, 2007; revised May 13, 2008. Current version published November 17, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaolin Wu.

Z. He is with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211 USA (e-mail: HeZhi@missouri.edu; http://www.missouri.edu/~hezhi).

D. Wu is with the University of Florida, Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: wu@ece.ufl.edu; http://www.wu.ece.ufl.edu.)

Digital Object Identifier 10.1109/TMM.2008.2004903

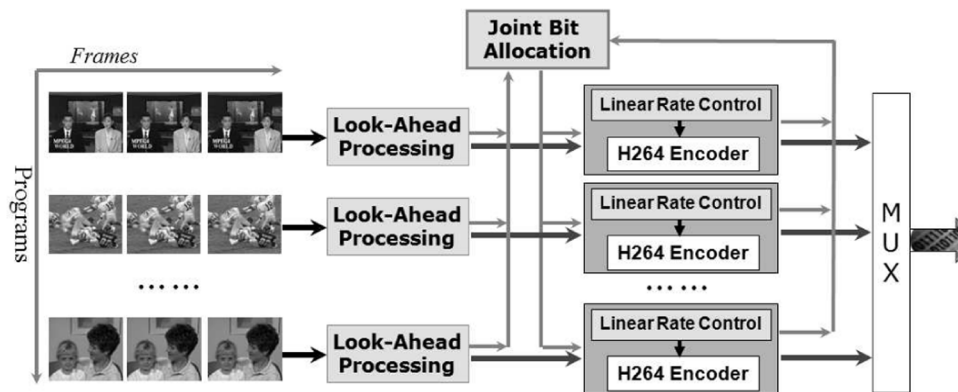


Fig. 1. Illustration of statistical multiplexing in H.264 TV broadcast.

a bandwidth-limited network to receivers for video decoding and presentation. In this case, not only the bit rate of each H.264 encoder needs to be accurately controlled, but also the total transmission bandwidth needs to be efficiently allocated between video programs such that the overall video presentation quality is maximized [17], [18], [19], [21]. Note that, in a multiplexing system, different video programs may have different scene activities; and within each program, the scene activity may change dramatically over time. To achieve a good statistical multiplexing performance, we need to analyze and estimate the R-D behavior of the H.264 encoder for each frame in each program. Based on their R-D statistics, we allocate the total network bandwidth across different video programs and frames to maximize the overall video broadcast quality. Since each video stream needs to be transmitted in real-time over the network to end users, a joint rate allocation should also consider decoder buffer constraints to avoid buffer under-flow or overflow which will cause unpleasant picture freezing or quality degradation due to data loss [19], [29].

A number of statistical multiplexing and joint rate control algorithms have been proposed for various video encoding schemes. Depending on how the R-D statistics of the video frames are obtained, these algorithms follow either of the following two approaches: *feedback* and *look-ahead* approaches. In the feedback approach [17]–[19], coded representations of previous video frames are used to derive or predict the encoding complexity of the current frame and its subsequent frames. This approach assumes that neighboring frames in the bit allocation window share similar coding characteristics, and for this reason, it often suffers from performance degradation at scene changes. In the look-ahead approach [20], [21], a pre-processing procedure is applied to future “uncoded” video frames within the look-ahead window to collect their R-D statistics; these R-D statistics are then used for joint rate allocation and rate control. Lack of an accurate R-D model for video encoders and a mathematical framework for statistical multiplexing, existing algorithms often achieve adaptivity instead of optimality in performance.

B. Overview of This Work and Major Contributions

The proposed linear rate control and optimum statistical multiplexing system for H.264 TV broadcast is illustrated in Fig. 1. Multiple video sequences captured from different sources are

encoded in parallel by a bank of H.264 video encoders. The compressed bit streams are multiplexed into a constant-bit-rate (CBR) channel and transmitted to end users. It should be noticed that different video programs have different scene activities. In addition, within each program, the scene activity may change over time (from frame to frame). Our design objective is to maximize the overall video broadcast quality by dynamically allocating the available channel bandwidth across video programs and frames. Sometimes, video programs may be pre-compressed. During the session of digital TV broadcast, these compressed video streams often need to be reprocessed using transcoding or rate shaping techniques before network transmission [22], [23]. This is because the bit rates of individual bit streams need to be adapted so as to match the time-varying network transmission condition and to maximize the overall video broadcast quality through joint rate allocation. In this case, the proposed optimum statistical multiplexing scheme also applies.

In statistical multiplexing system for digital TV broadcast, we need to allocate the channel bandwidth across programs and frames to maximize the overall video quality. The proposed multiplexing system has two modules: *look-ahead processing* and *joint rate allocation*. In look-ahead processing, we propose an efficient scheme to handle the complicated inter-frame dependency problem so that those rate-distortion (R-D) statistics needed by joint rate allocation can be obtained at low computational complexity. We will study bandwidth and buffer constraints in video broadcast and formulate the optimum statistical multiplexing into a constrained nonlinear optimization problem. Realizing that it is impossible to find a close-form solution for global optima, we propose a simple yet highly efficient algorithm to find a near-optimum solution for joint rate allocation and quality smoothing under buffer constraints.

Once the channel bandwidth has been allocated to each video program and the target bit rate for each video frame is determined, we need to develop an accurate and robust rate control algorithm for the H.264 video encoder to achieve the target bit rate. In this work, we find that, in H.264 video coding, the encoding bit rates of transform coefficients and overhead information are both linear functions of the fraction of quantized zeros. Based on this finding, we establish a linear rate model and then develop a linear rate control for H.264 video coding. Our extensive simulation results demonstrate that the proposed linear rate control and statistical multiplexing system achieves about

40–50% saving in bandwidth, provides smooth video quality change across programs and frames, and maintains robust decoder buffer control.

The rest of the paper is organized as follows. In Section II, we introduce the linear rate model and develop a linear rate control algorithm for H.264 video encoding. Section III presents the look-ahead processing scheme. Our joint rate allocation and its near-optimum solution are described in Section IV. Experimental results are presented in Section V. Section VI concludes the paper.

II. R-D ANALYSIS AND CONTROL FOR H.264 VIDEO CODING

In this section, we will develop a linear rate model and a linear rate control algorithm for H.264 video coding.

A. Rate Control for H.264

H.264 employs a number of sophisticated encoding schemes which are substantially different from previous standards, such as multiframe motion prediction, variable block sizes, multi-directional intra prediction, 4×4 integer transform, nonlinear and nonuniform quantization, and context adaptive binary arithmetic coding [4], [14]. This makes accurate R-D modeling and control more challenging. In addition, an effective rate control algorithm for H.264 video coding needs to address the following two issues. 1) In H.264 video coding, the fraction of overhead information bits, including bits for motion vectors, macroblock coding modes, and quantization parameters, is significant and often as high as 50% of the total bit rate [16]. These bits are not related to transform coefficients and not controlled by quantization parameters. More importantly, the fraction of these bits changes from frame to frame. Therefore, overhead information bits estimation has become an important issue in H.264 rate control. 2) H.264 employs an intra-prediction scheme where the encoding of an MB requires the reconstruction of previous MBs in the encoding order. This type of MB-level recursive encoding structure prevents us from collecting frame-level statistics for accurate rate control. In this work, we find that, in H.264 video coding, there is a linear relationship between the overall encoding bit rate and the fraction of zeros in the quantized transform coefficients. We then propose a scheme to cope with the recursive encoding structure and collect frame-level statistics. Based on the simple yet efficient linear rate model and frame-level statistics, we will develop a linear rate control algorithm for H.264 video coding.

B. Linear Rate Model for H.264 Encoding

In our previous work [6], we have demonstrated that, in typical image/video coding systems, including JPEG, JPEG2000, MPEG-2, H.263, and MPEG-4, there is a linear relationship between the coding bit rate R and the fraction of zeros among quantized transform coefficients. In other words, we have the following linear rate model,

$$R(\rho) = \theta \cdot (1 - \rho) + C_h \quad (1)$$

where ρ is the fraction of zeros and θ is a frame-dependent constant. Here, C_h is the coding bit rate of header information, including motion vectors, coding modes, picture headers, and

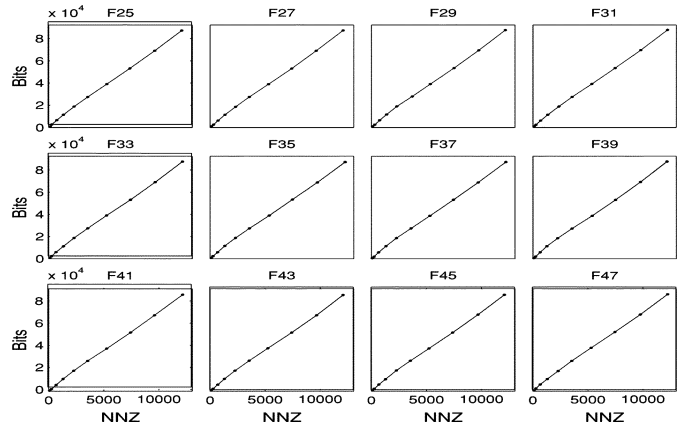


Fig. 2. The linear relationship between encoding bits (Y-axis) and the number of nonzero coefficients (X-axis) in H.264 coding for “Foreman” CIF video. All the subplots share the same coordinates.

other overhead information. In our previous work on rate control [6], the number of motion vector bits was estimated using table lookup after motion estimation. The rest of the header bits is relatively small and remains nearly constant between neighboring frames of the same picture type. Therefore, it is efficient to estimate it from previous frames. A detailed treatment of this linear rate model is available in [6].

As mentioned previously, in H.264 video coding, the amount of header bits, including bits for motion vectors, coding modes, and picture headers, is significant and sometimes changes significantly between neighboring frames. Accurate estimation of the overhead information bit rate has become one of the major issues in rate control for H.264 video coding. Kwon *et al.* [16] found that there is a linear relationship between header information bit rate and the number of nonzero motion vectors. This finding is very useful for overhead information bit rate estimation. However, it involves a relatively high computational complexity since it requires the information about motion vectors.

In this work, we find that, in H.264, not only the bit rate of transform coefficients but also the bit rate C_h of header information is a linear function of ρ . This interesting finding helps us to solve the problem of estimating header information bits and allows us to establish an accurate rate model for H.264 video coding. In the following, we present one example (out of many) to demonstrate this linear relationship. We execute the H.264 video encoder (version JM 9.5) [24] on “Foreman” QCIF (176×144) video sequence coded at 15 frames per second (fps) with CABAC (context adaptive binary arithmetic coding). We encode each frame with multiple quantization parameters, record the corresponding number of encoding bits and the number of nonzero coefficients (NNZs). Fig. 2 shows the total number of encoding bits as a function of NNZs. It can be seen that there is a strong linear relationship between them. Since $(1 - \rho)$ represents the fraction of nonzeros coefficients, we have

$$R = \theta \cdot (1 - \rho) \quad (2)$$

where θ is constant. This linear rate model also holds for Intra frames, as well as B-frames with bidirectional motion prediction. It also holds when CAVLC (context adaptive variable length coding) entropy coding is used [4]. The linear rate

model in (2) applies not only to a video frame, but also to a slice or slice group [4]. It should be noted that the linear rate model may lose its accuracy at extremely low or high bit rates. However, within a normal operation range of video encoding, for example, between 25 dB and 45 dB in picture quality, this model is fairly accurate. It is also worthwhile to mention that, when the fraction of zeros approaches 1 or the fraction of nonzeros approaches 0, the number of encoding bits will be very small (about 300–400 bits), but won't go to exactly 0. These bits are called overhead bits. We find that the amount of these overhead bits is very small and does not change significantly from frame to frame. Therefore, we can simply estimate it using the statistics from previous frames.

C. Linear Rate Control for H.264 Video Encoders

Based on the linear rate model in (2), we are going to develop a linear rate control (LRC) algorithm for H.264 video encoding. To this end, there are two major issues that need to be addressed. First, how to estimate the value of θ in (2)? Second, how to determine the quantization parameter for each MB during H.264 video encoding?

1) *Estimating the Value of θ* : To estimate the value of θ , our basic approach is to use the coding statistics of previous encoded MBs in the current video frame. Let N_m be the number of coded MBs in the current frame and R_m the number of bits already used to encode these N_m MBs. We denote the number of zeros produced by these MBs by ρ_m . Note that each 16×16 MB in a YUV 4:2:0 format has 384 coefficients. Therefore, the average coding bit rate of these coded MBs is given by $R_m/(384 \cdot N_m)$ and the fraction of zeros is $\rho_m/(384 \cdot N_m)$. According to (2), θ can be estimated as follows:

$$\theta = \frac{R_m}{384 \cdot N_m - \rho_m}. \quad (3)$$

As more and more MBs are encoded, the estimated value of θ should approach its true value of the current frame.

2) *Determining the Quantization Parameter*: The proposed rate control operates at the MB level, which determines the quantization parameter of each MB to be encoded. Let the bit budget of the current video frame be R_t . Since R_m bits have been already used for encoded previous N_m , we have $R_t - R_m$ bits left for remaining MBs. Let M be total number of MBs in the current video frame. Therefore, the average coding bit rate of remaining MBs is given by

$$\bar{R} = \frac{R_t - R_m}{384(M - N_m)}. \quad (4)$$

According to (2), the fraction of zeros that need to be produced by quantizing the remaining MBs is given by

$$\bar{\rho} = 1 - \bar{R}/\theta. \quad (5)$$

We observe that, in typical transform coding systems, including H.264, the fraction of zeros increases with the quantization parameter. This implies that, in general, there is a one-to-one mapping between them [6]. Based on this one-to-one $\rho - q$ mapping, we can determine the quantization parameter \bar{q} for a given

$\bar{\rho}$, which should achieve the target coding bit rate \bar{R} . In the following section, we will discuss how to collect the distribution of transform coefficients of the whole video frame and estimate the $\rho - q$ mapping.

D. Collection of Frame-Level Statistics

In H.264 video coding, spatial intra prediction, which explores the spatial correlation among neighboring MBs, is used to improve the compression efficiency. The prediction of the current MB relies on the reconstruction of its previous MBs in the encoding order [4]. This type of recursive encoding structure requires that the encoding process be performed only on an MB-by-MB basis and potentially prevents the rate controller from accessing future MBs to collect frame-level statistics, such as the distribution of transform coefficients needed by our linear rate control scheme.

To collect frame-level statistics, we propose a simple method to break the recursive structure of H.264 encoding and implement a two-loop encoding pipeline. To elaborate, in the first loop, the motion compensation of all possible modes, as well as spatial intra prediction, are performed for all MBs in the current frame. It should be noted that in this loop we use *original* pixel values for intra prediction instead of reconstructed ones, because they are simply not available at this moment. After motion compensation and spatial prediction, block transform is applied to obtain the distribution of transform coefficients. In the second loop of actual encoding and reconstruction, the intra prediction mode is checked again because reconstructed neighboring pixels are available now. If the MB coding type is changed after mode decision, we need to perform the block transform again on this MB. It can be seen that, compared to the one-loop encoding scheme, the only extra computation introduced by this two-loop pipeline is intra-prediction and block transform (if the MB type is changed), which involves very low computational complexity. However, with such type of two-loop encoding, the rate controller has access to frame-level statistics, which is critical for accurate and robust rate control.

After spatial intra prediction and temporal motion prediction, we apply the 4×4 transform [4] to obtain transform coefficients. Let $P(x)$ be the distribution of the un-quantized transform coefficients. We have

$$\rho(q) = \frac{1}{S} \sum_{x \in \Omega(q)} P(x) \quad (6)$$

where S is the total number of transform coefficients in the frame, and $\Omega(q)$ represents the dead zone of the quantizer. It should be noted that 1) H.264 adopts a nonlinear scalar quantization scheme with perceptual weighting; 2) Intra and Inter MBs are using different dead zone thresholds during quantization. Therefore, when computing the $\rho - q$ mapping, we need to maintain separate distributions for transform coefficients in Intra and Inter MBs. It should be noted that some other H.264 encoding features, such as skipping of blocks [4], which might also affect the computation of $\rho - q$ mapping, have not been considered here. However, for the purpose of rate control, the distribution of transform coefficients is able to efficiently capture content variation within a frame and allows us to determine

the quantization parameter at low complexity while achieving accurate and robust rate control, as we can see from the experimental results in Section V.

E. Rate Control Algorithm

The proposed linear rate control (LRC) algorithm can be summarized and implemented as follows. Let the target bit rate (in bits) per frame be R_T . The value of R_T will be determined by joint rate allocation explained in Section IV. Let the encoder buffer size be B_T and the number of bits in the buffer be B_0 . The available bits for coding the current frame is set to be

$$R = R_T - B_0 + \alpha \cdot B_T \quad (7)$$

where the target buffer fullness α is by default set to be 0.2 [9]. The following steps are performed to achieve the target coding bit rate R .

- Step 1) **Collecting frame statistics:** As explained in Section II-D, perform motion compensation and/or spatial prediction and block transform for all MBs in the current frame. Find the distribution of transform coefficients.
- Step 2) **Initialization of LRC:** Before encoding the first MB, set $N_m = R_m = \rho_m = 0$. Also set θ to be its value in the previous frame of the same picture type.
- Step 3) **Determine q for the current MB:** Determine the target fraction of zeros for the remaining MBs using (5). Based on the one-to-one mapping between ρ and q given in (6), determine the quantization step size q , which will be used to quantize the current MB.
- Step 4) **Update:** Let ρ_0 and R_0 be the number of zeros and number of bits produced by the current MB. Set $\rho_m = \rho_m + \rho_0$, $R_m = R_m + R_0$, and $N_m = N_m + 1$. If $N_m \geq 0.1N$, update the value of θ according to (3). Update the distribution $P(x)$ by removing the frequency statistics of transform coefficients in the current MB from the distribution, so that $P(x)$ always represents the distribution of transform coefficients in these remaining uncoded MBs.
- Step 5) **Loop:** Repeat steps 3 and 4 until all MBs in the frame are encoded.

It can be seen that the LRC algorithm always divides a video frame into two parts: coded and uncoded MBs, uses the distribution of transform coefficients as their coding complexity measure to perform bit allocation, and uses the linear rate model to determine the quantization parameter. The major computational complexity of the proposed algorithm is in collecting the distribution of transform coefficients. The actual rate control only involves a small number of addition and multiplication operations. Therefore, the overall complexity is very low. The performance of the LRC algorithm will be evaluated in Section V. The LRC algorithm will be used to control the output bit rate of the H.264 encoders in our statistical multiplexing system.

III. LOOK-AHEAD PROCESSING

In statistical multiplexing, the transmission bandwidth needs to be allocated across different video programs and frames to

maximize the overall video broadcast quality. To do this, we need to collect statistics of future frames. Without the knowledge of future frames, it will be very difficult to take advantage of the encoder buffer resource and perform joint rate allocation to optimize the overall video quality. It is also very important for maintaining robust buffer control, especially during dramatic scene changes. Predicting the statistics of future frames is challenging because of an interdependent quantization problem [8]. More specifically, just like the previous video coding standards, H.264 uses temporal motion prediction, in which previous encoded frames are used for motion prediction of the current frame. This implies that the R-D behavior of the current frame depends on the quality levels of its previous frames, which leads to an interdependent effect between their R-D behaviors. This type of inter-frame R-D dependency make it difficult to estimate the R-D functions of future frames during look-ahead processing.

In this work, we develop a simple yet efficient look-ahead processing scheme to handle this inter-frame R-D dependency so as to predict the R-D characteristics of future frames before the actual H.264 encoding. According to the classical R-D formula [7], the R-D function is given by

$$R_{nk}(D) = \frac{1}{2} \gamma \log_2 \frac{\sigma^2(n, k)}{D} \quad (8)$$

where $\sigma^2(n, k)$ is the variance of DCT coefficients in frame k and program n . γ is an encoder-related parameter. This R-D model suggests that the major task in look-ahead processing is to estimate $\sigma^2(n, k)$. As discussed in the above, $\sigma^2(n, k)$, the variance of the difference picture, depends on the history of the encoding process, especially the quality levels of previous encoded frames. Denote the k th frame in program n by $F(n, k)$. Let $\hat{F}(n, k-1)$ be the encoder reconstruction of frame $F(n, k-1)$. By definition,

$$\sigma^2(n, k) = \text{MSE}[F(n, k) \stackrel{\text{mc}}{\leftarrow} \hat{F}(n, k-1)], \quad (9)$$

where $\stackrel{\text{mc}}{\leftarrow}$ represents the motion prediction difference operation [3]. Let

$$D = \text{MSE}[F(n, k-1) - \hat{F}(n, k-1)] \quad (10)$$

which is the MSE picture distortion of frame $k-1$. Note that in statistical multiplexing, the H.264 encoder is trying to allocate bits among neighboring frames, making full use of the transmission buffer to maintain a steady and smooth video streaming quality. Therefore, we can assume that D is constant within the look-ahead window. Let

$$F_d(n, k) = \text{MSE}[F(n, k) \stackrel{\text{mc}}{\leftarrow} F(n, k-1)], \quad (11)$$

which is the variance of the motion prediction difference picture with the original frame as reference. Note that the picture distortion D is caused by quantization errors, and $F_d(n, k)$ is caused by scene motion. Therefore, we can assume that these two types of errors (picture difference) are uncorrelated. Based on this assumption, as illustrated in Fig. 3, we have

$$\sigma^2(n, k) = F_d(n, k) + D. \quad (12)$$

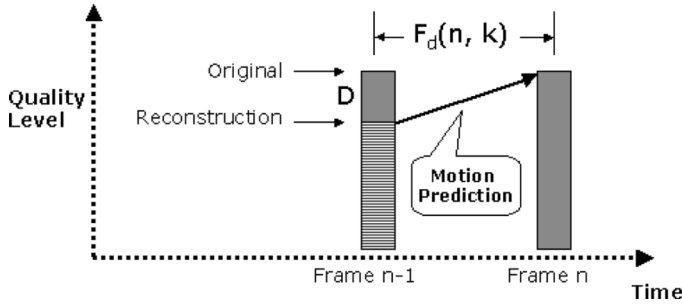


Fig. 3. Estimation of the frame variance during the look-ahead processing.

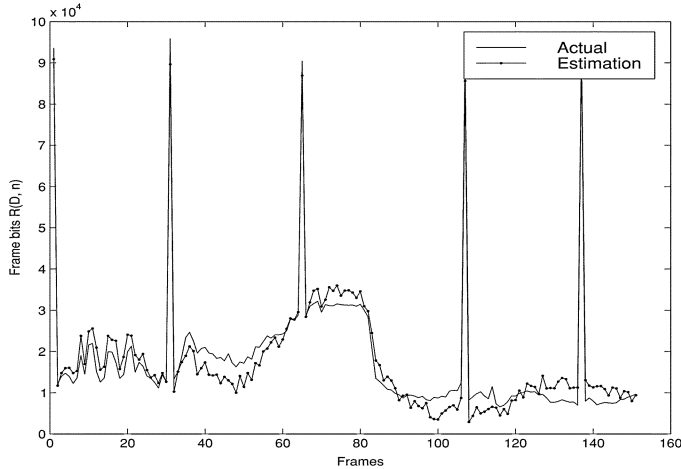


Fig. 4. Estimation of the R-D function $\{R(D, n)\}$ for "Foreman" CIF coded at 30 fps with an average quality of 33.5 dB.

TABLE I
PERFORMANCE EVALUATION OF LOOK-AHEAD PROCESSING

Video	Settings	Quality	Average REE
Stephan	CIF, 30 fps	33.5 dB	7.1%
Carphone	CIF, 30 fps	32.0 dB	6.2%
Football	CIF, 15 fps	33.5 dB	9.4%
News	QCIF, 30 fps	34.0 dB	4.3%

Therefore,

$$R_{nk}(D) = \frac{1}{2}\gamma \log_2 \frac{F_d(n, k) + D}{D} = \frac{1}{2}\gamma \log_2 \left[1 + \frac{F_d(n, k)}{D} \right]. \quad (13)$$

This is the simple and efficient model that we are going to use to predict the R-D behaviors of video frames in the look-ahead window. It should be noted that here we assume that motion vectors in the actual H.264 video encoding are the same as those obtained from the look-ahead process. In addition, we have not considered the spatial intra prediction in H.264 encoding [4], [14]. However, these mismatches will be compensated by adjusting the encoder parameter γ . In this work, we simply use the R-D statistics of previous encoded frames to estimate the value of γ based on the R-D formula in (8).

To verify the efficiency of the look-ahead model in (13), we encode the "Foreman" CIF sequence (150 frames) coded at 30 fps with a target picture quality of 34.5 dB using H.264. Fig. 4

shows the actual frame bits statistics and the estimation given by (13). Table I shows the relative estimation error (REE) of bit rate for four additional sequences (the first 150 frames). We can see that the estimated bit rates are close to actual measurements. Note that the purpose of look-ahead processing is to probe the scene complexity of future video frames and collect their R-D statistics for joint rate allocation. Therefore, there is no need for the model in (13) to be very accurate only if it is able to adequately differentiate the scene complexity between different frames and programs for the joint rate allocation purpose.

From (13), we can see that, to determine the R-D functions of video frames in the look-ahead window, we only need to find $F_d(n, k)$. To this end, we perform motion estimation and compensation on the original frames. It should be noted that the reference frame for motion prediction is the original video frame instead of the reconstructed one. To reduce the overall encoding complexity, the motion estimation results obtained here in look-ahead processing can be utilized by motion estimation in actual video encoding.

IV. JOINT RATE ALLOCATION

In this section, we formulate the optimum statistical multiplexing into a constrained optimization problem and study buffer constraints in real-time video streaming. We realize that it is very difficult to obtain a close-form solution for this nonlinear constrained optimization problem. By assuming a constant quality within the look-ahead window, this problem can be greatly simplified and converted into a linear optimization problem, which enables us to find a near-optimum solution.

A. Measuring Overall Video Broadcast Quality

The most widely used measure for picture quality is the Mean Squared Error (MSE) between the decoded picture and the original one. The MSE is very popular because of its simple mathematical formulation. In bit allocation and quality optimization, analytic solutions can be often obtained if we use the MSE as the image/video quality metric in the objective function. Certainly, perceptual video quality metrics [25], or their simplified versions, such as perceptually weighted MSE, can be also considered. In this work, we just use the MSE-based video quality metric as an example to demonstrate how the overall video quality of a video broadcast system can be optimized with optimum statistical multiplexing and accurate rate control. It should be noted that the MSE measures the picture quality of a single video frame. A typical approach to extend this picture quality metric to a video sequence is to consider the video sequence as a series of still images, evaluate the picture quality for each individual frame, and then pool these quality measurements together across different video frames, or even different video programs using some mathematical formulation, such as Minkowski summation [26], [27]. More specifically, let D_{nk} be the MSE distortion of the k th frame in the n th video program. Within a certain period of time, from frame k to frame $k + L$, the overall video quality of program n is measured by the following Minkowski summation [26]:

$$D_n = \left[\frac{1}{L} \sum_{i=k+1}^{k+L} D_{ni}^p \right]^{\frac{1}{p}} \quad (14)$$

where p is a positive constant, typically ranging from 1 to 9 [26]. Likewise, the overall video quality for all video programs is measured by

$$D = \left\{ \frac{1}{N} \sum_{n=1}^N \left[\frac{1}{L} \sum_{i=k+1}^{k+L} D_{ni}^p \right]^{\frac{q}{p}} \right\}^{\frac{1}{q}} \quad (15)$$

where q is a positive constant, also typically ranging from 1 to 9. Here, p and q specify the relative importance of those frames with peak visual distortion values. A possible further extension of this video quality metric is to consider different perceptual weights [26] or priority levels of different videos programs. Let w_n be the weight for program n . The overall video quality is given by

$$D = \left\{ \frac{1}{N} \sum_{n=1}^N \left[\frac{1}{L} \sum_{i=k+1}^{k+L} w_n D_{ni}^p \right]^{\frac{q}{p}} \right\}^{\frac{1}{q}}. \quad (16)$$

One of the major advantages of Minkowski summation is that it is able to assign different levels of penalty on low-quality pictures by choosing different values of p and q . For example, if we set p and q to be 1, the Minkowski summation is just an arithmetic averaging operation. If we set them to be a very large value, the overall video distortion will then approach the peak value of $\{D_{ni} | 1 \leq n \leq N, k \leq i \leq k+L\}$.

B. Buffer Constraints and Mathematical Formulation

In video streaming, a decoder buffer is used to receive the bit stream from transmission channels, and the decoder drains the compressed picture data from the buffer, decodes it, and presents the picture to end users. Therefore, in statistical multiplexing, when we determine the encoding and transmission rates for each video program, we need to guarantee that the output bit stream satisfies the decoder buffer constraint [28]. Let $C(n, k)$ be the transmission rate of program n at frame time k . Since all N video programs are multiplexed onto a single channel with a bandwidth W , we should have

$$\sum_{n=1}^N C(n, k) \leq W. \quad (17)$$

Let $R(n, k)$ be the encoding bit rate allocated by the statistical multiplexing algorithm to frame k in program n . Let T_0 be the start-up delay (in frames) of the decoder. The decoder buffer level, denoted by $B_d(n, k)$, is given by the difference between output and input bit rates of the buffer, i.e.,

$$B_d(n, k) = \begin{cases} \sum_{i=1}^k C(n, i) - \sum_{i=1}^{k-T_0} R(n, i), & \text{when } k \geq T_0; \\ \sum_{i=1}^k C(n, i), & \text{when } k < T_0. \end{cases} \quad (18)$$

Therefore, the objective of joint rate allocation in statistical multiplexing is to determine the coding bit rate and transmission rate for every program, $\{R(n, k), C(n, k)\}$, such that the overall visual distortion metric in (16) is minimized under buffer constraints. Mathematically, this can be formulated into the following constrained optimization problem:

$$\begin{aligned} & \text{minimize} && D = \left\{ \frac{1}{N} \sum_{n=1}^N \left[\frac{1}{L} \sum_{i=k+1}^{k+L} D_{ni}^p \right]^{\frac{q}{p}} \right\}^{\frac{1}{q}}, \\ & \text{s.t.} && \sum_{n=1}^N C(n, k) \leq W, \\ & && 0 \leq B_d(n, k) \leq B \end{aligned} \quad (19)$$

where B is the buffer size, and the second constraint implies there are no buffer overflow and under-flow. Here,

$$D_{nk} = D_{nk}(R(n, i)) = \frac{F_d(n, k)}{2^{\frac{2}{\gamma} R(n, k)} - 1} \quad (20)$$

which is derived from (13).

C. Constant Quality Assumption and Model Simplification

We can see that the optimization problem in (19) is highly nonlinear with a number of constraints. It is often very difficult, or even impossible to derive a close-form solution for this minimization problem. In this work, realizing this difficulty, we attempt to find a robust and near-optimum solution. This is accomplished by making the following assumption: with optimum statistical multiplexing, the quality change across video programs and frames should be smooth, or in an even more ideal case, the video quality of each frame within the look-ahead window is constant. In other words, $D_{n,i} = D$, for $1 \leq n \leq N$ and $k+1 \leq i \leq K+L$, where D is the operational video distortion level of the H.264 encoder. Note that this constant can also change as the window moves to the next frame, as we can see from the experimental results in Section V. This assumption is reasonable. First, in its theoretical analysis [10] an optimum solution of rate allocation will achieve equal distortion in each input source. Second, within a small look-ahead window (typically 10–50 frames), video frames share similar image features and R-D characteristics. Therefore, practically, it is reasonable to assume that they have the same coding distortion level during joint rate allocation. This constant visual distortion assumption is also desired in many applications. For example, in digital TV broadcast, when viewers flip through TV channels, they don't expect a sudden change in the video quality of TV programs. In this work, this assumption significantly simplifies the computation and enables us to find a near-optimum solution to the optimization problem in (19).

Since the operational visual distortion level of the H.264 encoder is D . According to (13), the corresponding bit rate of frame (n, i) , denoted by $R(n, i)$ is given by

$$R(n, i) = \frac{1}{2} \gamma \log_2 \left[1 + \frac{F_d(n, i)}{D} \right]. \quad (21)$$

Using the constant visual distortion assumption, the optimization problem in (19) is simplified to

$$\begin{aligned} \min_{\{R(n,k), C(n,k)\}} \quad & D \\ \text{s.t.} \quad & \sum_{n=1}^N C(n,k) \leq W, \\ & 0 \leq \sum_{i=1}^k C(n,i) \\ & - \sum_{i=1}^{k-T_0} \frac{1}{2} \gamma \log_2 \left[1 + \frac{F_d(n,i)}{D} \right] \leq B \end{aligned} \quad (22)$$

which can be solved using the numerical algorithm presented in the following section.

D. Algorithm for Statistical Multiplexing

The simplification from (19) to (22) enables us to find a highly efficient and robust solution to the optimum statistical multiplexing problem. Note that the objective of (22) is to find the minimum value of D while satisfying the buffer constraints. Our basic approach is to start from a very large initial value of D and then decrease its value step by step until the buffer constraints are no longer satisfied. Note that, in statistical multiplexing, transmission rates $\{C(n,k)\}$ and encoding bit rates $\{R(n,k)\}$ are inter-dependent and coupled under the buffer constraints. We observe that it is computationally intensive to determine their values simultaneously. Instead, we propose to decouple them and update their values in an iterative manner. The detailed statistical multiplexing algorithm is summarized as follows.

Step 1) Initialization: At the beginning of the video session, i.e., $k = 1$, set the encoder parameter γ to be its empirical value obtained from previous coding experiments.

Step 2) Look-ahead processing: Suppose the current frame number is k . As described in Section III, apply the look-ahead processing procedure to video frames within the look-ahead window, i.e., frames $k + 1$ to $k + L$, to obtain the original frame difference $\{F_d(n,i) | 1 \leq n \leq N, k + 1 \leq i \leq K + L\}$. It should be noted that, except the first time, we only need to perform look-ahead processing on the new video frame as the look-ahead window shifts to the position of this frame.

Step 3) Determine the transmission rates $\{C(n,k)\}$: Let D_0 be the average video distortion of all H.264 video programs at frame $k - 1$. When $k = 1$, we can simply set D_0 to be a video quality level desired by the application. It should be noted that this D_0 is used for look-ahead processing only. According to (21), the expected bit rate generated by the H.264 encoder, denoted by $R_0(n,k)$, will be

$$R_0(n,k) = \frac{1}{2} \gamma \log_2 \left[1 + \frac{F_d(n,k)}{D_0} \right]. \quad (23)$$

Certainly, those video programs with larger $R_0(n,k)$ should get higher transmission rates. Based on this observation, we use the following formula to assign the transmission rates:

$$C(n,k) = W \cdot \frac{\bar{R}_0(n,k)}{\sum_{j=1}^N \bar{R}_0(j,k)} \quad (24)$$

where W is the total network transmission bandwidth, and $\bar{R}_0(n,k)$ is the average bit rate of program n given by

$$\bar{R}_0(n,k) = \frac{1}{L} \sum_{i=k+1}^{k+L} R_0(n,i). \quad (25)$$

Step 4) Determine the encoding bit rates $\{R(n,k)\}$: Once the transmission rates $\{C(n,k)\}$ are assigned, the encoding bit rate $\{R(n,k)\}$ can be obtained using the following method: we start from a very large value of D and reduce its value by a small step size until the buffer overflow constraints in (22) are satisfied. It should be noted that when D is very large, $\{R(n,k)\}$ will be very small and the buffer constraints will be satisfied. This will guarantee that the minimum solution will be found if the step size is sufficiently small. To prevent potential buffer overflow or underflow, if needed, we modify the value of encoding bit rate $R(n,k)$ such that the buffer level of each individual video is above 10% and below 90% of the buffer size.

Step 5) Linear rate control: $R(n,k)$ obtained from Step 4 is the target encoding bit rate for the current frame (n,k) . The linear rate control algorithm presented in Section II-E is used to adjust the encoding quantization parameter to achieve this target bit rate.

Step 6) Loop: Increase the frame number k by 1 and repeat steps 2 to 5 until all frames are finished.

One unique feature of the proposed statistical multiplexing algorithm is the constant video distortion constraint within the look-ahead window. It has two major advantages. First, it enables us to find a fast and efficient solution to the highly non-linear constrained optimization problem in (19). Second, when the look-ahead window moves to its next position, the constant video distortion constraint acts like a low-pass filter, which efficiently smoothes out the frame-to-frame and program-to-program quality variation. Experimental results presented in Section V will demonstrate this unique property.

E. Complexity Analysis

The additional computational complexity introduced by the proposed statistical multiplexing system is low. More specifically, the system has three components: look-ahead processing, joint rate allocation, and linear rate control. The look-ahead processing performs motion compensation and/or spatial prediction using original frames as reference. When the look-ahead processing shifts to the next video frame, results for existing video frames in the window can be reused. We only need to perform look-ahead processing for the new-coming frame. The motion vectors obtained from this look-ahead processing are reused

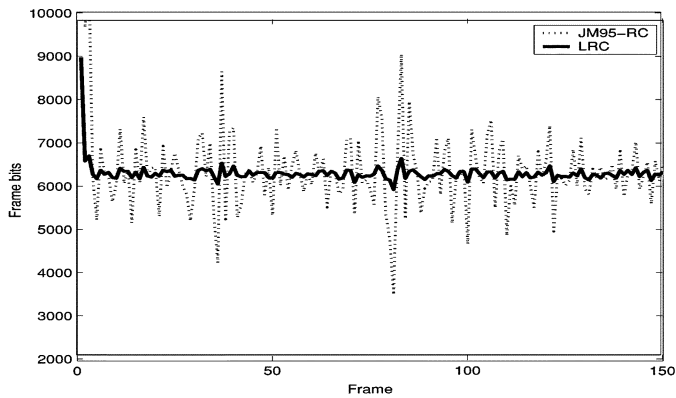


Fig. 5. Actual frame bits of the “Foreman” QCIF video coded at 96 kbps and 15 fps when the LRC algorithm and the H.264 rate control (H.264-RC) are applied.

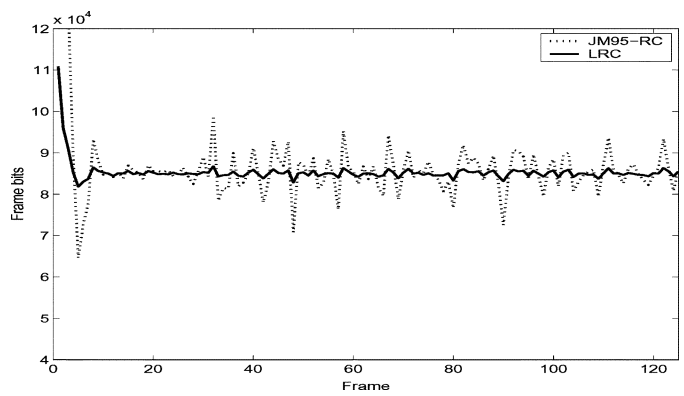


Fig. 6. Actual frame bits of the “NBA” CIF video coded at 1280 kbps and 15 fps when the LRC algorithm and the H.264 rate control (H.264-RC) are applied.

by the actual encoding process. Therefore, the additional complexity introduced by look-ahead processing is relatively small. According to our experience, it is about 5–7% of the overall encoding complexity.

The joint rate allocation algorithm outlined in Section IV-D also have low complexity since it only involve a relatively small number of arithmetic operations. The process in Step 4 converges very fast, especially if we start from a reasonable initial value of D . For example, we can use two times the average distortion of video frames in the previous multiplexing window as the initial value. The third component, linear rate control, also has low computational complexity, as discussed in Section II-E. According to our experience, the overall computational complexity of the proposed statistical multiplexing system is about 10–12% of the overall encoding complexity.

V. EXPERIMENTAL RESULTS

We implement the linear rate control (LRC) algorithm and the statistical multiplexing algorithm in an H.264 video encoder (version JM 9.5) [24]. To simulate the statistical multiplexing process, we execute the H.264 encoding on multiple video sequences on a single PC and use a shared computer file with a proper access control to synchronize the H.264 video encoders and exchange information for joint rate allocation. The R-D statistics obtained from the look-ahead module are written into this file to be used by the joint rate allocation module. After joint rate allocation, the control information, including frame bit rate targets and transmission rates, is sent back to each H.264 encoder. The LRC algorithm operates within each video frame to achieve the target frame coding bit rate allocated by the joint rate allocation module. In all experiments, the H.264 encoder uses a motion search range of ± 16 pixels with up to five reference frames and CABAC for entropy coding. No B frames are used.

Our experimental evaluation consists of two parts. In the first part, we test the performance of the LRC algorithm and compare it with the rate control provided by the reference H.264 JM9.5 video encoder [24]. Except the first Intra frame, the rest frames are encoded as P frames. The following test video sequences are used: “Foreman” QCIF (176×144) coded at 96 kbps and 15 fps, and “NBA” CIF (352×288) coded at 1280 kbps and 15 fps. In Figs. 5 and 6, we plot the actual encoding bits per

TABLE II
PERFORMANCE EVALUATION OF LINEAR RATE CONTROL

Video	Settings	Target	Avg RCE	Max RCE	PSNR Gain
Foreman	CIF, 15 fps	256 kbps	1.4%	5.1%	0.42 dB
Foreman	QCIF, 10 fps	96 kbps	1.6%	6.3%	0.49 dB
Carphone	CIF, 30 fps	320 kbps	0.8%	4.2%	0.39 dB
Football	CIF, 15 fps	640 kbps	2.9%	6.1%	0.55 dB
News	QCIF, 30 fps	96 kbps	0.5%	3.3%	0.35 dB

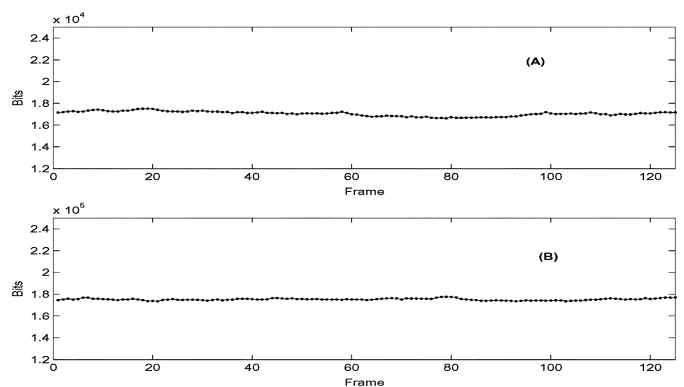


Fig. 7. Linear rate control for all Intra frames: (A) actual frame bits of the “Foreman” QCIF video coded at 256 kbps and 15 fps; (B) actual frame bits of the “NBA” CIF video coded with all Intra frames at 2560 kbps and 15 fps.

frame when the LRC (solid line) and the JM9.5 rate control (JM95-RC) (dotted line) are applied. It can be seen that the proposed LRC algorithm achieves significantly more accurate and robust rate control. Table II shows the rate control performance of the proposed LRC algorithm for other test video sequences at various encoding settings. It can be seen that the average relative control error (RCE) is mostly less than 3% and the maximum RCE is less than 7%. About 0.3–0.6 dB quality improvement has been observed during our simulations when the LRC algorithm is used. The proposed linear rate control algorithm also applies to Intra frames. To demonstrate its performance, we encode both sequences using all Intra frames: “Foreman” QCIF video at 256 kbps and 15 fps and “NBA” CIF video at 2560 kbps and 15 fps. Fig. 7 shows the actual bits of each frame. It can be seen that it is very close the frame bit rate target. The average relative control errors are 1.7% for “Foreman” and 3.3% for “NBA”.

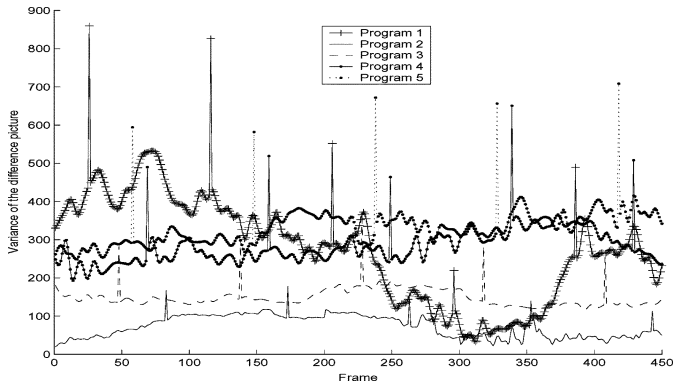


Fig. 8. Variance of the original picture difference $F_d(n, k)$ of the test video clips.

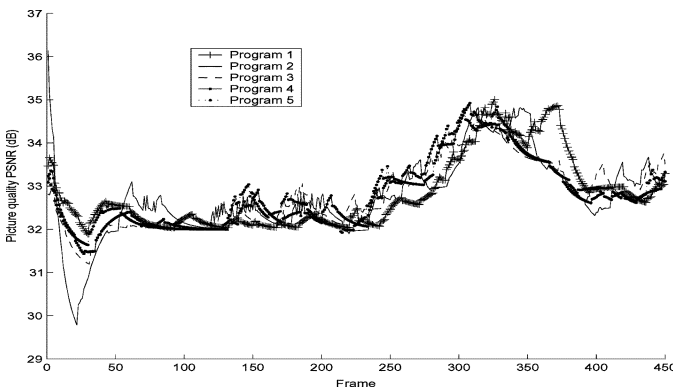


Fig. 9. Output video quality (PSNR) of each program in the H.264 statistical multiplexing system.

In the second part, to test the performance of the statistical multiplex algorithm presented in Section IV-D, we use five TV video clips which have a wide range of scene activities, including sports, news, and movie scenes. The input videos are encoded at CIF size and 30 fps. The period of Intra frames is 90 (or 3 seconds). Fig. 8 shows the original frame difference $F_d(n, k)$, as defined in Section III, of the five video clips. It should be noted in this experiment the Intra frames of different videos are not synchronized. This is because in practice a video stream can join the statistical multiplexing system at any time. We can see that the scene activity variation across video programs and among video frames is fairly large, which imposes challenges to the bit allocation, quality smoothing, and buffer regulation modules in the statistical multiplexing system. The total network bandwidth is set to be 7.5 Mbps. The start-up delay is 1 second, and the look-ahead window size is 15 frames. Here, a start-up delay is needed so that the buffer can build up to prevent potential underflow.

First, we consider that all video programs have the same perceptual weight w_n . Fig. 9 shows the PSNR of each encoded video frame of the H.264 statistical multiplexing system. It can be seen that the temporal quality change from frame to frame is smooth; and the quality variation across programs is very small, mostly less than 1.0 dB. Fig. 10 shows the actual coding bits $R(n, k)$ of each frame. (Those peaks are coding bit rates of Intra frames.) The transmission rate $C(n, k)$ is shown in Fig. 11. Fig. 12 plots the decoder buffer level (in bits) of each video

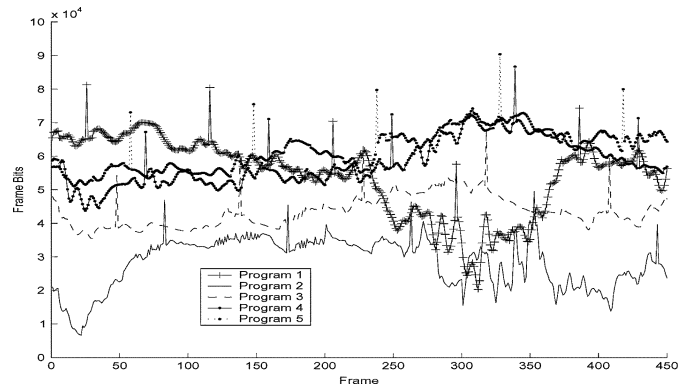


Fig. 10. Coding bits of each frame in the H.264 statistical multiplexing system.

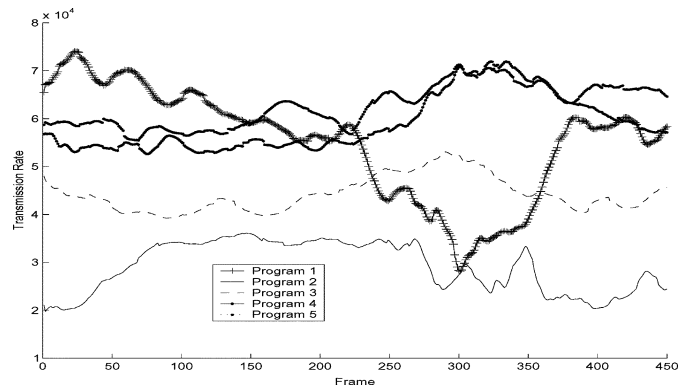


Fig. 11. Transmission rate of each program in the H.264 statistical multiplexing system.

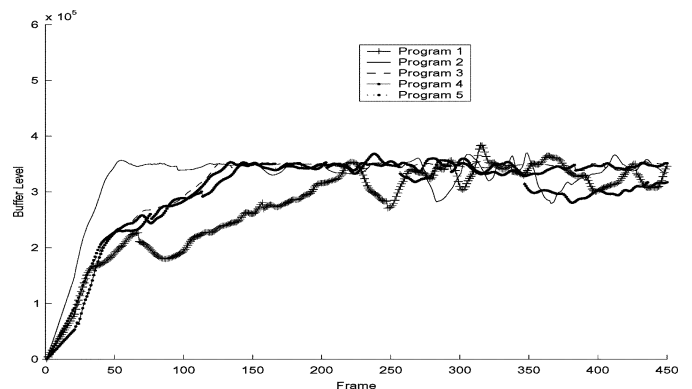


Fig. 12. Decoder buffer level of each program in the H.264 statistical multiplexing system.

program. It can be seen that the by selecting appropriate transmission rates and performing efficient bit allocation, the statistical multiplexing system achieves robust regulation of decoder buffers.

Next, we consider that different video programs have different perceptual weights or priority levels. The perceptual weight can be obtained from one of those perceptual visual quality models developed in the literature [25]–[27]. The priority levels can be assigned by users or determined by application requirements. To begin with, we set the perceptual weights for those 5 video programs to be (2.5, 2.0, 1.0, 0.5, 0.7). Figs. 13–16 show the PSNR quality, frame encoding bits,

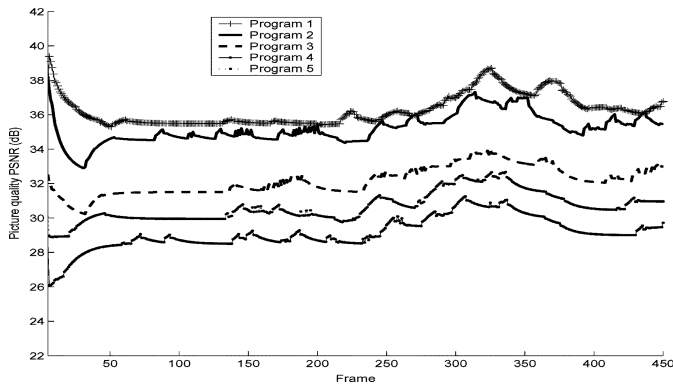


Fig. 13. Output video quality (PSNR) of the H.264 statistical multiplexing system when different programs have different perceptual weights.

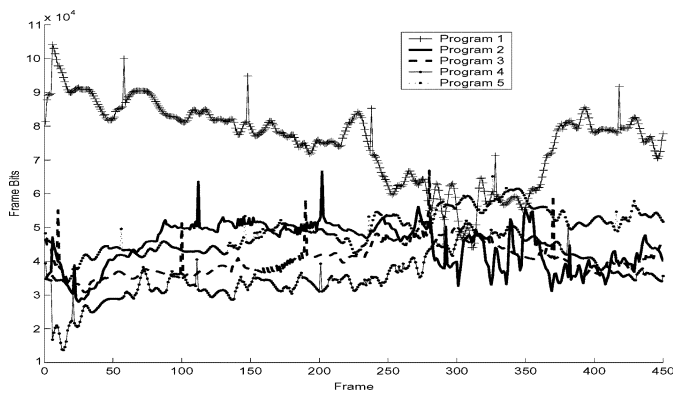


Fig. 14. Coding bits of each frame in the H.264 statistical multiplexing system when different programs have different perceptual weights.

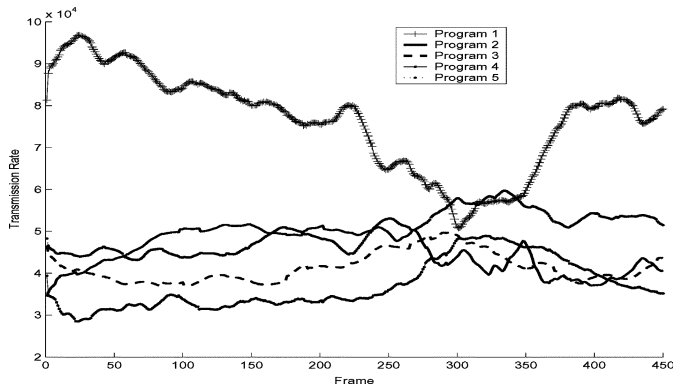


Fig. 15. Transmission rate of each program in the H.264 statistical multiplexing system when different programs have different perceptual weights.

transmission rate, and decoder buffer level of each program. Note that in Fig. 13 program 1 has a much higher PSNR quality than others because its perceptual weight is the largest. It can be seen that the proposed statistical multiplexing system is still able to maintain a smooth video presentation quality and robust buffer control when different video programs have different perceptual weights or encoding priority levels.

Fig. 17 shows the broadcast quality, defined by (16), at different bandwidth with and without optimum statistical multiplexing. In the “without” case, an equal partition of the total

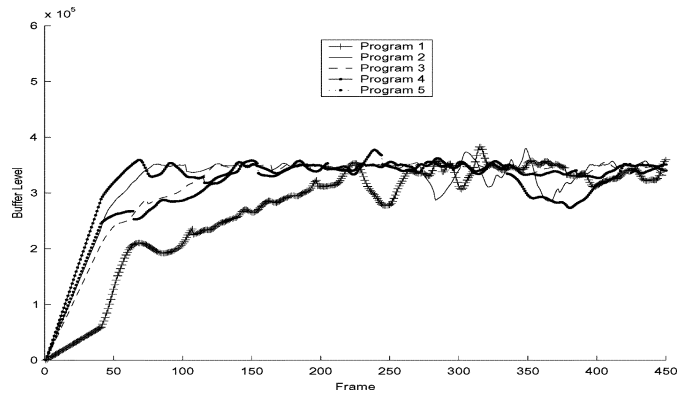


Fig. 16. Decoder buffer level of each program in the H.264 statistical multiplexing system when different programs have different perceptual weights.

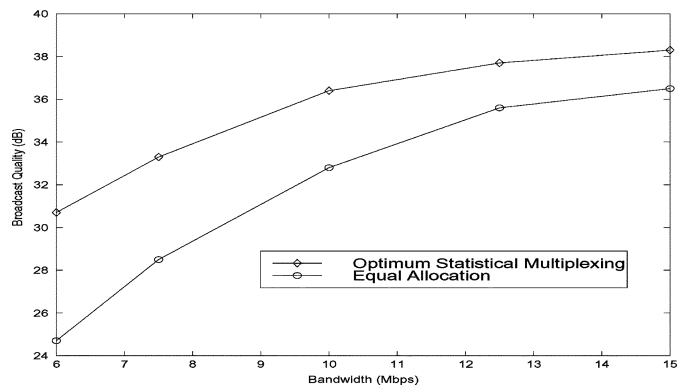


Fig. 17. Broadcast quality with and without optimum statistical multiplexing.

TABLE III
QUALITY IMPROVEMENT WITH THE PROPOSED
STATISTICAL MULTIPLEXING SYSTEM

Perceptual Weights of Videos	Average PSNR Improvement
(1.0, 1.0, 1.0, 1.0, 1.0)	1.9 dB
(2.5, 2.0, 1.0, 0.5, 0.7)	2.9 dB
(2.0, 1.0, 0.5, 0.7, 2.5)	2.6 dB
(1.0, 0.5, 0.7, 2.5, 2.0)	2.0 dB
(0.5, 0.7, 2.5, 2.0, 1.0)	2.1 dB
(0.7, 2.5, 2.0, 1.0, 0.5)	2.2 dB

bandwidth is used. For comparison, we also include a low-complexity version of the proposed algorithm which performs look-ahead processing only for the first 30 frames and assumes the subsequent video frames share similar statistics. Table III shows some examples of different perceptual weight assignments and the corresponding quality improvement in terms of PSNR. It can be seen that the optimum statistical multiplexing achieves about 2–3 dB improvement (on average) in the broadcast quality. If translated into bits saving, this is about 30–50% saving in bandwidth. In other words, with the proposed optimum statistical multiplexing, the digital broadcast system is able to transmit nearly twice the TV programs.

We can also see that the low-complexity version does result in some improvement over equal rate allocation. However, sometimes, this low-complexity scheme might be even worse than the equal rate allocation scheme when the scene content changes dramatically over time because it does not track the dynamic change of scene content and allocate bit rate accordingly. This

is why continuous look-ahead processing is needed in the proposed scheme. We have not conducted performance comparisons between the proposed system and the existing schemes in the literature, because we found that it is difficult to implement those algorithms, and hard to achieve the same performance as described by those papers [17]–[19]. We believe that the major quality improvement comes from robust joint rate allocation and accurate rate control. The joint rate allocation, coupled with look-ahead processing, is able to exploit the content variations between video programs and frames. The linear rate control is able to accurately determine the quantization parameters to achieve the target bit rates assigned by joint rate allocation.

VI. CONCLUDING REMARKS

To enable H.264 digital TV broadcast, this paper studied two important problems, namely, R-D analysis and statistical multiplexing. The major contributions of this work are in the following. First, we demonstrated that in H.264 video coding there is a linear relationship between the overall encoding bit rate and the fraction of zeros among quantized transform coefficients. Based on this linear rate model, we developed a linear rate control (LRC) scheme for H.264 video coding. Second, we developed a simple and efficient look-ahead processing scheme to handle the inter-frame R-D dependency in H.264 video encoding and collect the R-D statistics of future frames for joint rate allocation. Third, we developed a fast and robust algorithm to find a near-optimum solution to the joint rate allocation under the decoder buffer constraints. The proposed statistical multiplexing system represents one of the first systematic studies in this important area. Our extensive simulation results demonstrated that the proposed statistical multiplexing system is able to maintain a smooth video presentation quality across programs and among frames, as well as a robust decoder buffer regulation.

REFERENCES

- [1] "Video coding for low bit rate communications," in ITU-T Recommendation H.263 Jan. 1998, ITU-T, version 1.
- [2] D. LeGall, "MPEG: A video compression standard for multimedia application," *Commun. ACM*, vol. 34, pp. 46–58, Apr. 1991.
- [3] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 19–31, Feb. 1997.
- [4] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, p. 560576, July 2003.
- [5] W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 12–20, Feb. 1996.
- [6] Z. He and S. K. Mitra, "A linear source model and a unified rate control algorithm for DCT video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 970–982, Nov. 2002.
- [7] T. Berger, *Rate Distortion Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [8] L.-J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 38, pp. 82–93, Jan. 1990.
- [9] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 172–185, Feb. 1999.
- [10] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 9, pp. 1445–1453, Sep. 1988.
- [11] Z. G. Li, F. Pan, K. P. Lim, and S. Rahardja, "Adaptive rate control for H.264," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2004, p. 745748.
- [12] P. Yin and B. J. J., "A new rate control scheme for H.264 video coding," in *Proc. Int. Conf. Image Processing*, Oct. 24–27, 2004, vol. 1, pp. 449–452.
- [13] H. Yu, Z. Lin, and F. Pan, "An improved rate control algorithm for H.264," in *Proc. IEEE Int. Symp. Circuits and Systems, ISCAS 2005*, May 23–26, 2005, vol. 1, pp. 312–315.
- [14] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [15] S. Kim and Y. S. Ho, "Rate control algorithm for H.264/AVC video coding standard based on rate-quantization model," in *Proc. IEEE Int. Conf. Multimedia and Expo.*, Jun. 2004, p. 165168.
- [16] D.-K. Kwon, M.-Y. Shen, and C.-C. J. Kuo, "A novel two-stage rate control scheme for H.264," in *IEEE Int. Conf. Multimedia and Expo (ICME)*, Toronto, ON, Canada, Jul. 9–12, 2006.
- [17] L. Brzczyk, A. Y. Ngai, and E. F. Westermann, "Statistical multiplexing using MPEG-2 video encoders," *IBM J. Res. Develop.*, vol. 43, no. 4, pp. 511–520, Jul. 1999.
- [18] M. Perkins and D. Arnstein, "Statistical multiplexing of multiple MPEG-2 video programs in a single channel," *SMPTE J.*, vol. 104, no. 9, pp. 596–599, 1995.
- [19] L. Wang and A. Vincent, "Joint rate control for multi-program video coding," *IEEE Trans. Consumer Electron.*, vol. 42, no. 3, pp. 300–305, Aug. 1996.
- [20] M. Balakrishnan and R. Cohen, "Global optimization of multiplexed video encoders," in *Proc. ICIP'97*, Santa Barbara, CA, Oct. 26–29, 1997, vol. 1, pp. 377–380.
- [21] J. Yang, X. Fang, and H. Xiong, "A joint rate control scheme for H.264 encoding of multiple video sequences," *IEEE Trans. Consumer Electron.*, vol. 51, no. 2, pp. 617–623, May 2005.
- [22] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," *Proc. IEEE*, vol. 93, no. 1, pp. 84–97, Jan. 2005.
- [23] A. Eleftheriadis and P. Batra, "Dynamic rate shaping of compressed digital video," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 297–314, Apr. 2006.
- [24] *JVTH.264 JM Software*, [Online]. Available: <http://bs.hhi.de/suehring/tml/>
- [25] J. Lubin and D. Fibush, Sarnoff JND Vision Model ANSI T1 Standards Committee, T1A1.5 Working Group Document No. 97-612, 1997.
- [26] M. Masry and S. S. Hemami, "A metric for the continuous evaluation of compressed video with severe distortions," *Signal Process.: Image Commun., Special Issue on Video Quality*, vol. 19, no. 2, Feb. 2004.
- [27] H. de Ridder, "Minkowski-metrics as a combination rule for digital-image-coding impairments," in *Proc. SPIE-Human Vision, Visual Processing and Digital Display III*, 1992, p. 1626.
- [28] *Information Technology Generic Coding of Moving Pictures and Associated Audio Information: Video (MPEG-2/H.262)*, , 2000, MPEG-2, Annex C, video buffering verifier, ISO/IEC 138 180-2.
- [29] C.-Y. Hsu, A. Ortega, and A. R. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints," *IEEE J. Sel. Areas Commun., Special Issue on Real-Time Video Services in Multimedia Networks*, vol. 15, no. 6, pp. 1016–1028, Aug. 1997.



Zhihai He (S'98-M'01-SM'06) received the B.S. degree from Beijing Normal University, Beijing, China, and the M.S. degree from Institute of Computational Mathematics, Chinese Academy of Sciences, Beijing, in 1994 and 1997, respectively, both in mathematics, and the Ph.D. degree in electrical engineering from the University of California, Santa Barbara, in 2001.

In 2001, he joined Sarnoff Corporation, Princeton, NJ, as a Member of Technical Staff. In 2003, he joined the Department of Electrical and Computer Engineering, University of Missouri, Columbia, as an assistant professor. His current research interests include image/video processing and compression, network transmission, wireless communication, computer vision analysis, sensor networks, and embedded system design.

Dr. He received the 2002 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT) Best Paper Award and the SPIE VCIP Young Investigator Award in 2004. Currently, he serves as an Associate Editor for IEEE TCSVT and for the *Journal of Visual Communication and Image Representation*. He is also a guest-editor for the IEEE TCSVT Special Issue on Video Surveillance. He is a member of the Visual Signal Processing and Communication Technical Committee of the IEEE Circuits and Systems Society, and serves as Technical Program Committee member or session chair of a number of international conferences.



Dapeng Oliver Wu (S'98-M'04-SM'06) received the B.E. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1990, the M.E. degree in electrical engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1997, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2003.

Since August 2003, he has been with Electrical and Computer Engineering Department at the University of Florida, Gainesville, as an Assistant Professor. His research interests are in the areas of networking, communications, multimedia, signal processing, and information and network security.

Dr. Wu received the ONR YIP award in 2008, NSF CAREER award in 2007, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT) Best Paper Award for Year 2001, and the Best Paper Award at the

International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine) 2006. Currently, he serves as the Editor-in-Chief of the *Journal of Advances in Multimedia*, and an Associate Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TCSVT, and the *International Journal of Ad Hoc and Ubiquitous Computing*. He was an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY between 2004 and 2007. He is also a guest-editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) Special Issue on Cross-layer Optimized Wireless Multimedia Communications. He has served as Program Chair for IEEE International Conference on Communications (ICC 2008), Signal Processing for Communications Symposium, and as a member of executive committee and/or technical program committee of over 50 conferences. He is Vice Chair of Mobile and wireless multimedia Interest Group (MobIG), Technical Committee on Multimedia Communications, IEEE Communications Society. He is a member of the Best Paper Award Committee, Technical Committee on Multimedia Communications, IEEE Communications Society.