Rate-Distortion Analysis and Quality Control in Scalable Internet Streaming

Min Dai, *Member, IEEE*, Dmitri Loguinov, *Member, IEEE*, and Hayder M. Radha, *Senior Member, IEEE*

***Abstract*—Rate-distortion (R-D) modeling of video coders has always been an important issue in video streaming; however, few of the traditional R-D models and their performance have been closely examined in the context of scalable (FGS-like) video. To overcome this shortcoming, the first half of the paper models rate-distortion of DCT-based fine-granular scalable coders and derives a simple operational R-D model for Internet streaming applications. Experimental results demonstrate that this R-D result, an extension of the classical R-D formula, is very accurate within the domain of scalable coding methods exemplified by MPEG-4 FGS and H.264 progressive FGS. In the second half of the paper, we examine congestion control and dynamic rate-scaling algorithms that achieve smooth visual quality during streaming using the proposed R-D model. In constant bitrate (CBR) channels, our R-D based quality-control algorithm dramatically reduces PSNR variation between adjacent frames (to less than 0.1 dB in sample sequences). Since the Internet is a changing environment shared by many sources, even R-D based quality control often cannot guarantee nonfluctuating PSNR in variable-bitrate (VBR) channels without the help from an appropriate congestion controller. Thus, we apply recent utility-based congestion control methods to our problem and show how a combination of this approach and our R-D model can benefit future streaming applications.**

***Index Terms*—MPEG-4 FGS, quality control, rate distortion, scalable streaming.**

I. Introduction

RATE-DISTORTION (R-D) curves are useful not only in source coding, but also in Internet video streaming. While it is well-known that R-D based compression approaches can adaptively select quantization steps and maximize video quality under given buffer constraints [7], [22], R-D curves can also be used during streaming rate control to optimally allocate bits in joint source-channel coding [3], [13], avoid network congestion [4], and achieve constant quality at the receiver [32], [37], [38].

Accurate modeling of R-D curves of real encoders and channel characteristics of real communication systems (e.g., the Internet) is always challenging due to the diversity of source images and the inherent complexity of Internet-like channels [28]. Typically, R-D modeling is undertaken using either the *empirical* or the *analytical* approach, each of which has its own benefits and drawbacks. The empirical approach obtains R-D curves by interpolating between (R, D) samples of a given encoder [23]. The analytical approach derives R-D models from the angle of information and/or quantization theory assuming certain (usually simplified) statistical and correlational properties of the source [8], [11], [14]. While the empirical approach usually results in better estimation of the curve, it fundamentally lacks theoretical insight into the structure of the coding system.

Thus, in order to accurately take into account complex statistical structure and source correlation of real encoders, a third, *operational*, type of R-D models is widely used in practice [4], [11], [12]. An operational R-D model obtains the basic structure of the curve from a closed-form analytical expression, but then parameterizes the equation according to several parameters sampled from the actual system (e.g., [4], [11], [12]).

Although there are numerous applications of R-D modeling in scalable Internet streaming [32], [34], [37], [38], the majority of current R-D models are built for images and/or non-scalable video coders [5], [14]. To overcome this gap in the current knowledge of scalable coding R-D systems and provide future video streaming applications with accurate R-D models, this paper derives two operational R-D models based on statistical properties of scalable sources and existing models of bitrate [12]. Our first result applies to the enhancement layer of a variety of scalable DCT coders (including FGS and PFGS) and demonstrates that distortion D is a function of both R and its logarithm $\log R$:

$$D = \sigma_x^2 - (a\log^2 R + b\log R + c)R \tag{1}$$

where σ_x^2 is the variance of the source and a, b, c are constants.

Since this formula is too complicated for delay-constrained streaming applications, our second R-D result is a polynomial approximation of (1), which can be summarized as an operational extension of the traditional model $D \sim 2^{-2R}$:

$$D = \sigma_x^2 2^{aR+b\sqrt{R}} \tag{2}$$

where $a < 0$ and b are constants [different from those in (1)] dependent on the properties of the source. During our journey to obtain these results, we also offer a new model for the distribution of DCT residue and derive an accurate Markov model for bitplane coding (more on this in the following sections).

With an accurate R-D model, we next address another important issue in video streaming, *quality control*. This is an important concern for end-users since human eyes are sensitive

Manuscript received May 11, 2004; revised February 13, 2006. This work was supported by the National Science Foundation under Grants CCR-0306246, ANI-0312461, CNS-0434940, and CNS-0519442. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Chang Wen Chen.

M. Dai is with the MediaFLO Video System Team, Qualcomm, Inc., San Diego, CA 92121 USA (e-mail: mdai@qualcomm.com).

D. Loguinov is with the Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: dmitri@cs.tamu.edu).

H. M. Radha is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: radha@egr.msu.edu).

Digital Object Identifier 10.1109/TMM.2006.884626

to quality fluctuation, which is often present in constant bitrate (CBR) coded base layers and in video streaming over variable bitrate (VBR) channels. Thus, during streaming, the server must rely on an efficient R-D model to rescale the enhancement layer to both match the available bandwidth in the network and smooth out visual quality fluctuations introduced by the base layer [32], [37], [38].

While video streaming has strict quality-of-service (QoS) requirements on bandwidth, delay, and packet loss, the current best-effort Internet does not provide any QoS guarantees to end flows. Therefore, congestion control is typically the only viable solution that allows streaming applications to avoid substantial packet loss, share the bottleneck routers fairly, and offer a pleasant video quality to end-users. Many current congestion controllers for streaming applications are built on top of TCP-friendly schemes and usually exhibit difficulty in maintaining a smooth channel due to their large rate fluctuations [1], [9] and asymptotic instability.

We take a different approach and extend the continuous-feedback congestion control methods proposed by Kelly *et al.* [19]. We study their performance in constant-quality network streaming and show that the resulting controller is stable under arbitrarily delayed feedback and offers end-flows exponential convergence to link utilization (in contrast to TCP's linear rate of convergence). This is one of the first papers to apply provably stable active queue management (AQM) congestion control [36] in constant-quality video streaming.

The rest of the paper is organized as follows. In Section II, we give a brief overview of related work. Section III provides the big picture of R-D modeling of scalable coders, presents a detailed analysis of distortion under bitplane quantization, and derives an accurate R-D model for scalable video systems. Section IV provides a simple operational R-D model that is suitable for real-time streaming. In Section V, we introduce Kelly's congestion controller and describe our quality control algorithm. Finally, Section VI concludes this paper.

II. Related Work

In this section, we briefly overview the work related to R-D modeling and quality-control during streaming.

A. R-D Modeling

We describe several closed-form R-D functions commonly used in video coding in this subsection. Recall that the most well-known R-D result stems from classical Shannon's work [5] and early developments in quantization theory [28]:

$$D = \sigma_x^2 2^{-2R} \tag{3}$$

where D denotes MSE distortion, R is the bitrate of the coded sequence, and σ_x^2 is the variance of the source. While directly applicable only to a small set of sources, this model is still widely used in video streaming [12], [32].

Shannon's classical model is the basis for many operational R-D models and is often extended to account for non-Gaussian distributions and nontrivial source correlation [11], [12]:

$$D = \gamma \varepsilon^2 \sigma_x^2 2^{-2R} \tag{4}$$

where γ is the correlation coefficient of the data and ε^2 is a source-dependent scaling parameter (1.4 for Gaussian, 1.2 for Laplacian, and 1 for uniform sources).

Distortion depends *only* on the statistical properties of the signal (i.e., its distribution); however, the rate also depends on the correlation among the input symbols [11], which explains the independent derivations of $D(\Delta)$ and $R(\Delta)$ often used in the literature. For uniform quantizers (UQ), the classical model is often decomposed into two separate models with respect to quantizer step Δ: distortion $D(\Delta)$ and rate $R(\Delta)$. Under uniform quantization, both models can be summarized as [11]

$$D(\Delta) = \frac{\Delta^2}{\beta}, \quad R(\Delta) = \frac{1}{2}\log_2\left(\frac{\varepsilon^2 \beta \sigma_x^2}{\Delta^2}\right) \tag{5}$$

where β is 12 for small Δ. To account for a wider range of Δ, parameter β typically needs to be empirically adjusted based on samples of the R-D curve or other source parameters [11].

For Laplacian sources with density $p(x) = (\lambda/2)e^{-\lambda|x|}$, the R-D function can also be written in terms of the mean absolute difference (MAD) distortion D_M [31]:

$$R = -\log(\alpha D_M) \tag{6}$$

where α is some constant. Using Taylor expansion of (6), Chiang *et al.* [4] propose an operational R-D model for Laplacian sources and apply it to the MSE distortion D:

$$R = aD^{-1} + bD^{-2} \tag{7}$$

where parameters a and b are obtained from samples of the empirical R-D curve.

Using a Cauchy density function to model DCT coefficients, Kamaci *et al.* [17] build a slightly different R-D model for H.264 video coders:

$$D = cR^{-\gamma} \tag{8}$$

where c and γ are some properly selected constants.

In another recent development, He *et al.* [12] propose a unified *ρ-domain* R-D model, in which the bitrate is estimated by a linear function of the percentage of zero coefficients in each video frame. In this framework, distortion D for each Δ is computed directly using the DCT coefficients without any modeling.

Besides the above operational models, there are purely empirical ways to estimate R-D curves. Among the numerous studies, e.g., Lin *et al.* [23] use cubic interpolation of the empirical curve and Zhao *et al.* [37] apply similar methods to FGS-related streaming algorithms.

B. Quality Control in Streaming

The MPEG-4 standard [26] has adopted Fine Granular Scalability (FGS) into its streaming profile and motivated the development of new scalable compression paradigms such as progressive FGS (PFGS) [35]. Both FGS and PFGS consist of a single base layer and one enhancement layer that contains the residual signal coded using embedded DCT. Due to nonstationary characteristics of video sources (such as scene changes), the base layer often exhibits significant quality fluctuation that needs to be smoothed out by the server, which must properly

select (for each frame) the fraction of the FGS layer that should be transmitted over the network.

Many approaches have been proposed to achieve constant quality in video streaming, e.g., Wang *et al.* [32] use (4) to estimate the R-D curve of PFGS and propose an optimal bit allocation scheme that reduces quality fluctuation based on the estimated R-D curve, the authors of [37] apply a similar method to FGS video, while Zhao *et al.* [38] obtain the R-D curve empirically and adopt Newton's search method to achieve constant quality during transmission of video over the Internet.

III. Scalable R-D Modeling: The Big Picture

A. Preliminaries

Due to its flexibility and strong adaptability to channel conditions, scalable coding techniques are widely applied in video coding and transmission. Scalable coding can be further grouped into coarse-granular (e.g., spatial scalability) and fine-granular (e.g., FGS). While the former method provides quality improvements only when a *complete* enhancement layer has been received, the latter continuously improves video quality with every additionally received codeword of the enhancement layer bitstream [33]. In this paper, we target fine granular scalable coders due to their *progressive* quality improvement during streaming. Both FGS and PFGS use *bitplane coding*, which considers each input value as a binary number instead of a decimal integer. During video streaming, bitplanes are transmitted from the most significant to the least significant thus resulting in gradual quality improvement as streaming rates increase. The enhancement layer in FGS is used only for reconstruction purposes while that in PFGS also serves for motion-compensated prediction (MCP) of the pixels in adjacent frames.

In scalable streaming applications, R-D curves can be employed to decide the proper scaling of the enhancement layer to both match the channel capacity and achieve smooth video quality at the receiver. Although there are many R-D based streaming solutions [32], [34], [37], [38], to our knowledge, the theoretical foundation behind the shape of R-D curves of scalable coders has not been investigated thoroughly.

In what follows, we describe the structure of a DCT-based scalable coder and explain why distortion in the enhancement layer is sufficient to model the end-user visual quality. Then we derive an R-D function based on source statistical properties and a ρ-domain bitrate model [12]. We use FGS [22] and PFGS [35] as typical examples during the discussion.

In a scalable coder, we assume that the rate of the base layer is R_B and its distortion is D_B. Furthermore, we assume that the server transmits R_E bits from the enhancement layer (in addition to R_B bits from the base layer) and achieves a combined distortion D. While the traditional approach is to model the distortion D as a function of the total bitrate $R = R_B + R_E$, several simplifications of this framework are possible. First, since the server during streaming is only able to chop the enhancement layer according to user requirements or network conditions, we are concerned only with the rate of the enhancement layer R_E instead of the total bitrate R. Second, as we show below, enhancement-layer distortion created at the server by discarding several least-significant bitplanes during transmission is sufficient for estimating the actual end-user distortion D.

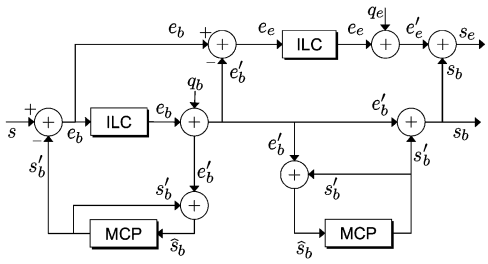


Fig. 1. Block diagram of a typical scalable coding system.

To better understand this scenario, we illustrate the encoding and decoding process of a scalable coder in Fig. 1. In the figure, symbol s represents the original frame, s'_b is the predicted frame, and $\hat{s}_b$ is the reference frame. Symbol e_b refers to the predicted error frame in the base layer, e_e is the input to the enhancement layer, and q_b and q_e are the quantization errors in the base and enhancement layers, respectively. Symbol s_b stands for the reconstructed image in the base layer and s_e is the final reconstructed image including both base and enhancement layer information. We use blocks labeled ILC in the figure to represent ideal lossless coding, which includes DCT/IDCT and entropy encoding/decoding, and blocks labeled MCP for motion estimation and compensation in the base layer.

With the above notation, we have the following lemma.

Lemma 1: The total distortion D of a scalable coder is equal to the distortion D_E caused in the enhancement layer.

Proof: Since s_e is the final reconstructed image in Fig. 1, the total distortion of a scalable coder can be written as $D = s_e - s$. Recalling that $s_e = s_b + e'_e$, $e_e = s - s_b$, and $e'_e = e_e + q_e$, where q_e is the quantization error during enhancement-layer processing, total distortion D becomes

$$D = s_e - s = (s_e - s_b) + (s_b - s) = e'_e - e_e = q_e.$$

Since quantization errors are the main reason for distortion in most lossy coding systems [14], we have $D_E = q_e$ and therefore $D = D_E$. ∎

Based on the above lemma, we model the enhancement layer distortion to capture the distortion of the whole coding system. Furthermore, it is well known that in an ideal encoder-decoder system, spatial-domain distortion D and DCT-domain distortion D_{DCT} are equal [14]. Thus, we develop the distortion model in the DCT domain throughout the paper, since the statistical properties of DCT residue are more mathematically tractable than those of the original signal. Note, however, that we verify our model using the *actual* distortion observed by the end-user, which includes the DCT/IDCT round-off errors.

B. Distortion Model

Recall that in an image/video coder, the distortion mostly comes from quantization errors, even in a lossy predictive coder [11], [33]. In a nonscalable coder or the base layer, the distortion comes from applying a uniform (usually) mid-point quantizer to each DCT coefficient (different quantizers are often applied to different frequencies) [10], [11]. On the other hand, embedded

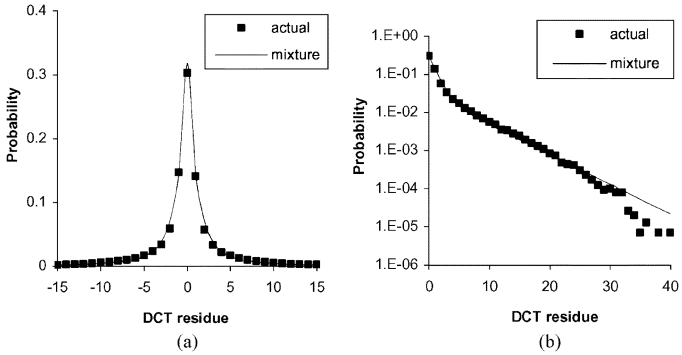


Fig. 2. Distribution of DCT residue fitted with a mixture Laplacian model in frame 0 of CIF Foreman. Sequence coded at 128 kb/s (base layer) and 10 fps using MPEG-4 FGS. (a) Full range; (b) log-scale positive tail.

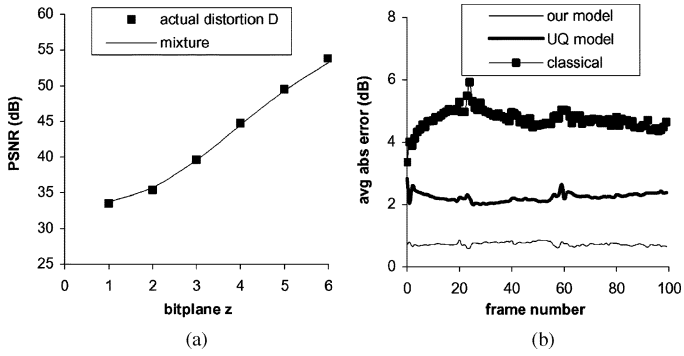


Fig. 3. (a) Spatial-domain distortion D in frame 0 of CIF Foreman and D_E estimated from model (10). (b) Average absolute PSNR error of UQ model (5) and our model (10) in CIF Coastguard. Both sequences coded at 128 kb/s (base layer) and 10 fps using MPEG-4 FGS.

coders such as FGS use *bitplane coding*, in which all coefficients are transmitted bit-by-bit from the most-significant bitplane (MSB) to the least-significant bitplane (LSB). This can be viewed as applying a quantizer with step $\Delta = 2^{n-z}$, where n is the total number of bitplanes in the frame and z is the current bitplane number.[1] For example, assuming that the maximum DCT coefficient is 40, n is 6 and Δ takes the values equal to 32, 16, 8, 4, 2, 1 for bitplanes 1 through 6, respectively.

Since a uniform quantizer is widely applied in video coders for its approximate optimality in the high bitrate case [7], model (5) is popular due to its simplicity and accuracy under these assumptions. However, when the output rate is not high enough, (5) requires complex adjustments to β to achieve satisfying results [11], [30]. Instead of coping with empirical parameter adjustments, we will derive an accurate distortion model based on source statistical properties.

As discussed earlier, the input to the enhancement layer is the DCT residue between the original signal and the reconstructed image of the base layer [26]. Our previous work [6] shows that the PMF (probability mass function) of DCT residue follows a *mixture* Laplacian distribution with density:

$$p(x) = q\frac{\lambda_0}{2}e^{-\lambda_0|x|} + (1-q)\frac{\lambda_1}{2}e^{-\lambda_1|x|} \tag{9}$$

where random variable x represents the DCT residue, q is the probability to obtain a sample from one of the two Laplacian components (e.g., the low-variance Laplacian distribution), and λ_0 and λ_1 are the shape parameters of the corresponding Laplacian distributions. The parameters of (9) can be optimally estimated using a variety of methods, such as the Expectation-Maximization (EM) algorithm [2]. As an example, Fig. 2 shows that the mixture Laplacian distribution (9) accurately models both the peak and the tail of the actual PMF (numerous additional examples omitted for brevity).

With the help of (9), we can now focus on understanding the properties of distortion $D(\Delta)$ caused by bitplane coding. The proof of the following lemma can be found in [6].

[1]While traditional quantizers implement mid-point reconstruction, bitplane coding can be viewed as a floor function applied to the result. MPEG-4 FGS has an option for "quarter-point" reconstruction, in which the decoder adds $\Delta/4$ to the result. For brevity, we omit $\Delta/4$ in all derivations; however, it can be shown that our final result holds for quarter-point quantizers as well.

Lemma 2: For Laplacian sources with PMF $p(m) = ae^{b|m|}$, $a > 0$ and $b < 0$, the MSE distortion after uniform quantization with step Δ is

$$D(\Delta) \approx \frac{2a\xi}{1-e^{b\Delta}} \tag{10}$$

where ξ is given by

$$\xi = e^{b(\Delta-1)}\left(\frac{(\Delta-1)^2}{b} - \frac{2(\Delta-1)}{b^2} + \frac{2}{b^3}\right) - \frac{2}{b^3}. \tag{11}$$

Notice that when $\Delta = 1$, (10) produces $D = 0$ and when $\Delta = \infty$, the distortion increases to $D = 2/\lambda^2 = \sigma_x^2$, where σ_x^2 is the variance of a Laplacian distribution. A distortion model for a mixture-Laplacian distribution is easily constructed by linearly combining (10) with the corresponding probability q and $1-q$ as shown in (9). The result of applying model (10) to frame 0 in CIF Foreman is plotted in Fig. 3(a), which shows a very good match and additionally demonstrates that D_E is almost the same as spatial-domain distortion D.

We extensively analyzed the performance of model (10) in other sequences and found that it was very accurate. Fig. 3(b) compares the performance of (10) to that of the classical model (4) and UQ model (5) in FGS-coded CIF Coastguard. The error in the figure is computed for each frame in the PSNR domain and then averaged over all bitplanes. As the figure shows, (10) maintains the average error below 0.8 dB, while the errors in the other two methods average between 2 and 6 dB.

Note, however, that this form of averaging can be misleading since large errors in the last bitplane (where they do not matter due to high signal PSNR) may skew the result obtained from the other bitplanes. Thus, in Table I, we examine the average errors *for each bitplane* over the entire CIF Foreman sequence (similar results hold for Coastguard and Carphone, both of which are omitted for brevity). As the table shows, the PSNR error is quite small for all bitplanes except the last one where approximation (10) is the weakest and results in the largest discrepancy between the model and the data. It is also worthwhile to note that a 1-dB error in a signal reconstructed at 56 dB is not noticeable, as well as that 0.15-dB errors in 30+ dB signals are relatively minor. Finally note that (10) applies to any Laplacian source

TABLE I
ESTIMATION ACCURACY OF (10) IN CIF FOREMAN

Δ	Average D	Average abs. error	Error in dB
64	81.5 (29.9 dB)	2.987	0.15
32	51.6 (31.2 dB)	1.768	0.15
16	23.1 (34.6 dB)	0.558	0.10
8	7.92 (39.2 dB)	0.239	0.13
4	2.16 (44.6 dB)	0.128	0.24
2	0.62 (49.8 dB)	0.039	0.25
1	0.08 (56.6 dB)	0.043	1.15

regardless of reconstruction points and whether the source contains FGS residue or base-layer DCT coefficients.

C. R-D Model

Notice that the traditional R-D model $D \sim 2^{-2R}$ is typically obtained under the assumptions of an infinite block length and high-resolution (i.e., small Δ) quantization that allows the PMF of the signal in each Δ-bin to be approximated by a constant [14], [28]. Neither of these two assumptions generally holds in practice, especially in cases of sharply decaying PMF of DCT residue (which is not constant even in small bins) and low-bitrate streaming (which inherently relies on high Δ).

To better understand this situation, we evaluate the accuracy of models (4) and (7), which are extensions and/or improvements of the traditional R-D model. Fig. 4 shows the R-D curves produced by (4) (labeled as "classical" in the figure) and (7) (labeled as "Chiang *et al.*"). Note the log-scale of the x-axis, which is used here to demonstrate that (7) exhibits bending and produces *negative* values of R for sufficiently large D (this cannot be shown in the PSNR figure, so the curve simply stops).

Due to the unsatisfied accuracy of current R-D models, we derive an alternative R-D model based on source statistical properties and a ρ-domain bitrate model [12].

Lemma 3: For Laplacian sources, distortion $D(R)$ can be expressed in closed form as:

$$D = \sigma_x^2 - (c_1 \log^2 R + c_2 \log R + c_3)R \tag{12}$$

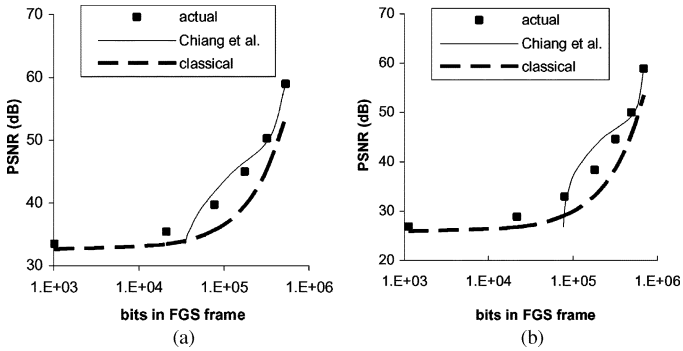
where constants $c_1 - c_4$ only depend on the shape of the distribution λ and ρ-domain constant γ [12].

Proof: First recall that He *et al.* [12] demonstrated in numerous simulations that in a variety of image and video coding methods, rate $R(z)$ was proportional to the percentage of nonzero coefficients $1-\rho$ in the source data:

$$R(z) = \gamma(1-\rho) \tag{13}$$

where γ is a constant. While we do not offer a deeper analytical treatment of (13) at this time, we utilize this empirical fact in our subsequent derivations, especially since this model holds very well for scalable coders (not shown here for brevity, but verified in simulations). Next notice that for Laplacian distributed sources, the percentage of nonzero coefficients is

$$1 - \rho = 1 - 2\int_0^{\Delta} \frac{\lambda}{2} e^{-\lambda x} dx = e^{-\lambda \Delta}, \tag{14}$$



(a) (b)

Fig. 4. R-D models (4), (7), and the actual R-D curve in CIF Foreman. Sequence coded at 128 kb/s (base layer) and 10 fps using MPEG-4 FGS. (a) Frame 0; (b) frame 84.

where λ is the shape parameter of the Laplacian distribution and Δ is the quantization step. Inserting (13) into (14), we express Δ in terms of rate R:

$$\Delta = -\frac{1}{\lambda} \log \frac{R}{\gamma},\ 0 < \frac{R}{\gamma} < e^{-\lambda}. \tag{15}$$

Combining this result (15) with our earlier distortion model (10), we have

$$D(\Delta) \approx \frac{\lambda \xi}{1 - R/\gamma}, \tag{16}$$

where ξ is:

$$\xi = \frac{2}{\lambda^3} - \frac{e^{\lambda} R}{\lambda^3 \gamma} \left[(\log R - \log \gamma + \lambda - 1)^2 + 1 \right]. \tag{17}$$

Expanding (17) and combining it with (16), we notice that

$$D = \begin{cases} \frac{2}{\lambda^2} = \sigma_x^2, & R = 0 \\ 0, & R \geq e^{-\lambda}\gamma \end{cases} \tag{18}$$

where σ_x^2 is the variance of the source. This observation makes perfect sense since distortion D should not be larger than σ_x^2 [5] and should equal zero when $R = e^{-\lambda}\gamma$ (i.e., the quantization step $\Delta = 1$ and there is no loss of information). After absorbing the various constants and neglecting small terms, we have the desired result in (12). ■

Estimation of γ for FGS sources is very simple. Once the FGS layer is coded, the number of bits $R(z)$ in each bitplane can be easily obtained by scanning the FGS layer for bitplane start codes (whose location can also be saved during encoding). Computing the percentage of zeros ρ_z in each bitplane directly from the DCT residue, the encoder can build the curve $(1-\rho_z, R(z))$ and estimate its linear slope γ. Simulation results in Fig. 5 show that model (12) outperforms traditional R-D models and maintains high accuracy in a variety of FGS-coded video frames.

The result in (12) shows that the R-D curve of scalable coders (mixture Laplacian sources) is both complex and highly nonlinear in both the MSE and PSNR domains. Nevertheless, this model provides valuable insight into the coding process and suggests the shape of the resulting R-D curve. For practical purposes, this model is still rather complex and hard to use for R-D analysis due to the numerous nonlinear terms in (17). Thus, we

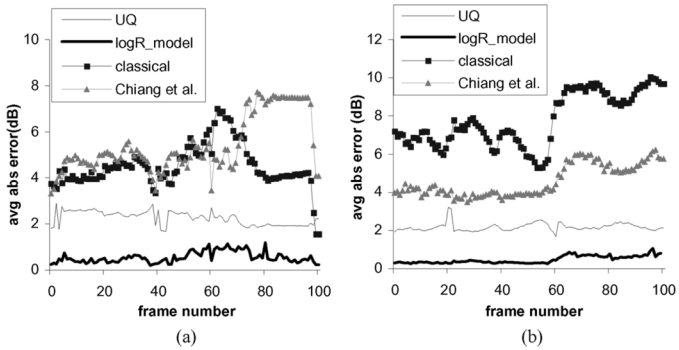


Fig. 5. Comparison between several popular R-D models and logarithmic model (12). Both sequences coded at 128 kb/s (base layer) and 10 fps. (a) CIF foreman (MPEG-4 FGS); (b) CIF carphone (MPEG-4 FGS).

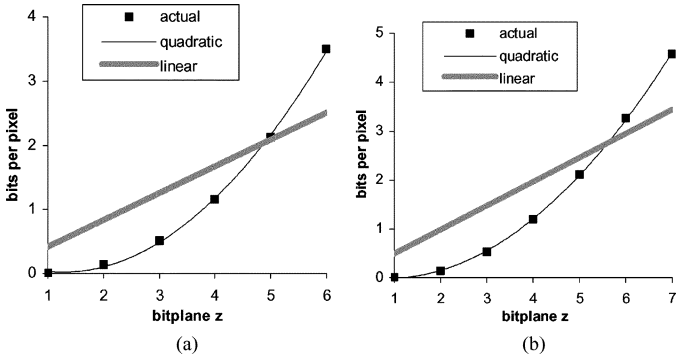


Fig. 6. Quadratic model (22) and the traditional linear model in CIF Foreman. Sequence coded at 128 kb/s (base layer) and 10 fps using MPEG-4 FGS. (a) Frame 0; (b) frame 84.

examine an even simpler operational model in the next section and use it for quality control in the subsequent parts of the paper.

IV. Square-Root R-D Model

Notice that the previously derived distortion model is too complicated for further analytical manipulation. Thus, we use the theory of coconvex/comonotone approximation [21] to simplify the equations. As shown in [21], if function $f \in C[-1,1]$ change its convexity finitely many times in a given interval, we can estimate f by polynomials that are coconvex with it, i.e., polynomials that change their convexity exactly at the points where f does.

In the next two subsections, we show how to use coconvex approximation theory to represent the distortion and bitrate as different polynomial functions of bitplane z.

A. Simple Bitrate Model

We start with the following supplementary result.

Lemma 4: Function $R(z)/\gamma$ for $z \in [1,n]$ is monotonically increasing, changes convexity no more than once, and remains in [0, 1) for all bitplanes z.

Proof: Combining (13) with (14) and keeping in mind that $\Delta = 2^{n-z}$, we have

$$\psi(z) = \frac{R(z)}{\gamma} = e^{-\lambda 2^{n-z}} < 1. \tag{19}$$

Taking the first two derivatives of (19), we have

$$\psi'(z) = \lambda 2^{n-z} \log(2) \psi(z) > 0, \tag{20}$$

$$\psi''(z) = \lambda \log(2) 2^{n-z} \left[-\psi(z) + \psi'(z)\right]. \tag{21}$$

Analysis of (21) shows three important points: (a) for $\lambda \geq 1$, the function ψ remains strictly convex in the entire interval, (b) for $\lambda \leq 2^{1-n}$, the function remains strictly concave, and (c) for the remaining values of λ, there is exactly one point $z = n + \log_2 \lambda$, in which the function changes convexity. ■

Using the theory of coconvex/comonotone approximation [21], an accurate polynomial approximation of $R(z)$ would require a cubic curve to match the possible change in convexity of the curve (the rest of the error is small since (19) exhibits a good degree of smoothness). However, for Laplacian source, we have $\lambda = \sqrt{2}/\sigma$, where the source standard deviation σ is always larger than $\sqrt{2}$ in video sequences. Thus, $0 < \lambda < 1$ and a quadratic approximation is accurate enough to estimate $R(z)$ in this range. We therefore approximate (19) with

$$R(z) = a_1 z^2 + a_2 z + a_3 \tag{22}$$

where constants a_1, a_2, a_3 can be estimated from empirical data. To better understand this operational model, we conducted numerous experiments and found that while cubic polynomials were a very good match to $R(z)$, quadratic functions also performed extremely well. Fig. 6 shows one such example for two frames in CIF Foreman together with a linear fit derived from model (5).

B. Simple Quality (PSNR) Model

Since PSNR is a popular quality measure in real video applications, we convert D into the PSNR domain here and reduce it to a simpler formula through a series of approximations. Recall that PSNR is a logarithmic function of distortion D:

$$PSNR = 10 \log_{10} \frac{255^2}{D} = 20 \log_{10} 255 - 10 \log_{10} D. \tag{23}$$

Substituting distortion function (10) into (23), we have the following lemma.

Lemma 5: For Laplacian distributed source with PDF $f(x) = \lambda e^{-\lambda x}/2, \lambda \in [0, 1]$, function $\varphi(z) = PSNR(z)$ is monotonically increasing for $z \in [1,n]$ and changes convexity no more than once for integer $z \geq 1$.

Proof: Taking the derivative of (23), we have

$$\varphi'(z) = -\frac{10 \cdot 2^{n-z} \theta (2^{n-z}-1)^2 \ln 2}{(\theta\kappa + 2/\lambda^3) \ln 10} + \frac{10 \cdot 2^{n-z} \lambda^2 e^{-\lambda 2^{n-z}} \ln 2}{(1 - e^{-\lambda 2^{n-z}}) \ln 10} \tag{24}$$

where $\theta = e^{-\lambda(2^{n-z}-1)}$ and $\kappa = -(2^{n-z}-1+1/\lambda)^2/\lambda - 1/\lambda^3$.

It is obvious that $0 < \theta < 1$, $\kappa < 0$, and the last item of (24) is positive. Whether the first item of (24) is positive or not depends on if $\theta\kappa + 2/\lambda^3$ in the first item is negative or not. Note that when $z = n$, $\theta_{max} = 1$. Thus,

$$\theta\kappa + 2/\lambda^3 \leq -\frac{(2^{n-z}-1)^2}{\lambda} - \frac{2(2^{n-z}-1)}{\lambda^2} \leq 0 \tag{25}$$

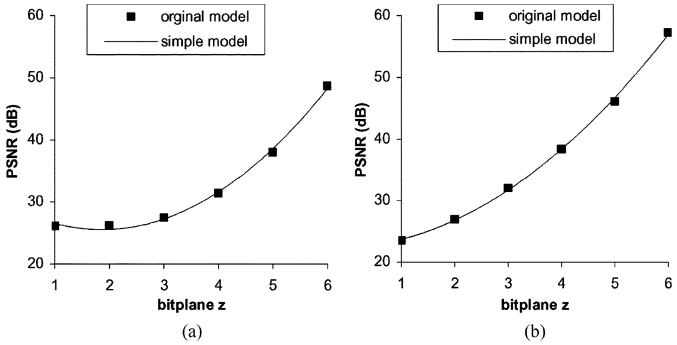


Fig. 7. Comparison between the original Laplacian model (10) and simple model (27). (a) $\lambda = 0.5$; (b) $\lambda = 0.12$.

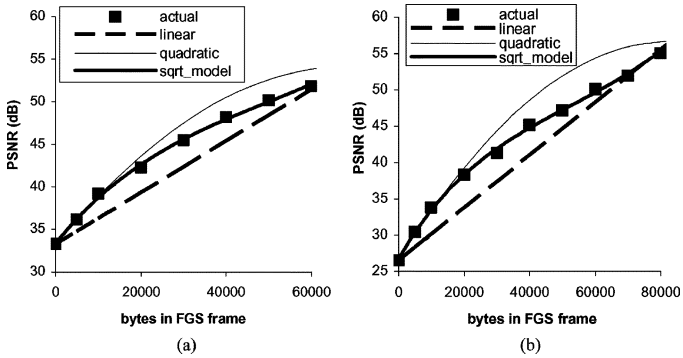


Fig. 8. CIF Foreman fitted with linear, quadratic, and SQRT model (31). Sequence coded at 128 kb/s (base layer) and 10 fps using MPEG-4 FGS. (a) Frame 39; (b) frame 73.

and we have $\varphi'(z) > 0$ for $z \in [1, n]$.

Further taking the second derivative of $\varphi(z)$, we get

$$\varphi''(z) = -d\ln 2 \times \left(\frac{e^{-\lambda(2d-1)}\left[3\lambda^3\mu + 4\lambda(d-1)+4\right] + 2e^{-\lambda d}}{(1-e^{-\lambda d})^2} + \frac{e^{-\lambda(d-1)}\left[\lambda^3\mu + 4(d-1)\lambda + 2\right]}{1-e^{-\lambda d}} + \frac{\left(2e^{-\lambda(d-1)}\lambda^3\mu + 4\right)e^{\lambda^2 d^2}}{(1-e^{-\lambda d})^3} \right) \tag{26}$$

where $d = 2^{n-z}$ and $\mu = -(d-1)^2/\lambda - 2(d-1)/\lambda^2 - 2/\lambda^3$.

Analysis of (26) shows that: (a) $\varphi''(z) < 0$ for $z < n$, (b) $\varphi''(z) = 0$ when $z = n$, and (c) $\varphi''(z) > 0$ for $z > n$. In other words, function $\varphi(z)$ changes convexity only once for integer $z \geq 1$. ■

Since the working range of video coders is $1 \leq z \leq n$, we can simply use a quadratic polynomial of the bitplane number z to approximate (23) [21]:

$$PSNR(z) \approx g_1 z^2 + g_2 z + g_3 \tag{27}$$

for some constants g_1, g_2, g_3.

We also calculate the approximation error of the quadratic function. Assume that $f \in \Delta^2(Y_s)$ represents the fact that function f changes convexity s times in its range and $p_n \in \Delta^2(Y_s)$ means that polynomials p_n are coconvex with f. Then the approximation error $E_n^{(2)}(f, Y_1)$ is [21]:

$$E_n^{(2)}(f, Y_1) \leq c\omega_2^{\varphi}\left(f, \frac{1}{n}\right), \quad n \geq 1 \tag{28}$$

where

$$E_n^{(2)}(f, Y_s) = \inf_{p_n \in \Pi_n \cap \Delta^2(Y_s)} \|f - p_n\| \tag{29}$$

Π_n is the set of polynomials of degree not exceeding n, and

$$\omega_2^{\varphi}\left(f, \frac{1}{n}\right) := \sup_{0 \leq h \leq 1/n} \sup_x |f(x-h) + f(x) + f(x+h)| \tag{30}$$

with $n = 2$ in (22) and (27).

To verify this approximation, we conducted a series of tests by fitting the simplified model (27) to the PSNR calculated from the original model (10) and found them to be an almost perfect match. The quality of the fit is illustrated on two different Laplacian distributions in Fig. 7. The left side of the figure shows a low-variance (large λ) case and the right side of the figure shows a high-variance (small λ) case; both matched the quadratic model (27) with very high accuracy.

C. SQRT Model

We next combine our proposed bitrate result in (22) with the earlier distortion model in (27) to obtain a final usable R-D model. After inverting the polynomial in (22), inserting $z(R)$ into (27), and dropping insignificant terms, we obtain the model that we call *Square Root* (SQRT):

$$PSNR(R) = AR + B\sqrt{R} + C \tag{31}$$

where constants A and B are estimated from at least two (R,D) samples, and $C = 10\log_{10}(255^2/\sigma_x^2)$ for uncorrelated (or weakly correlated) sources such as those in FGS coders. Parameter A and B are strongly negative-correlated (e.g., the 0-lag cross-correlation coefficient between these two parameters is -0.99 in the CIF Foreman sequence).

Recall that the traditional R-D framework (3) converted to PSNR quality becomes a linear function of rate R. However, as shown in Fig. 8 for two different frames of CIF Foreman, the actual R-D curve often cannot be modeled by a straight line over the entire range of R. In fact, even a heuristically selected quadratic curve in the figure (used here only for illustration purposes) is incapable of modeling the entire range of the bitrate. Both linear and quadratic models exhibit significant discrepancy that reaches as high as 5 dB. However, the figure demonstrates that the SQRT model (31) has a much better fit than was possible before.

To better understand the estimation accuracy of the different models discussed so far, we compare the SQRT model (31), Chiang's model (7), the UQ model (5), the model based on Cauchy distribution (8), and the classical model (4) in various video sequences. Fig. 9 shows the average absolute error between the actual R-D curve in the PSNR domain and each of the models in several FGS-coded sequences. For example, in the Foreman sequence, the error in SQRT averages 0.25 dB, while it stays as high as 2–8 dB in the other four models. We find that

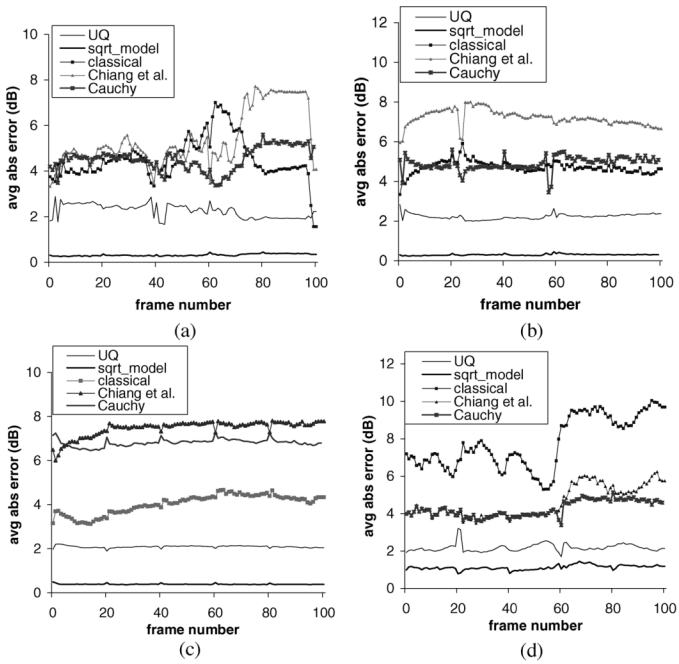


Fig. 9. Average absolute PSNR error in MPEG-4 FGS. Test sequences coded at 128 kb/s (base layer) and 10 fps. (a) CIF foreman; (b) CIF coastguard; (c) CIF mobile; (d) CIF carphone.

TABLE II
AVERAGE ABSOLUTE PSNR ERROR (dB)

Sequence	Format	SQRT	UQ	Classical	Chiang
Mobile	SIF	0.425	2.162	5.802	9.041
Silent	CIF	0.307	1.878	2.741	7.412
Akiyo	CIF	0.344	1.770	2.454	8.140
Container	CIF	0.251	1.669	2.353	7.226
Stefan	QCIF	0.459	1.835	4.589	7.920
Table Tennis	QCIF	0.338	1.878	3.598	10.466

our model significantly outperforms traditional models, which often require estimation of the same number of parameters.

We next test (31) using six additional sequences with different contents and compression rates. In Table II, the base layer of SIF Mobile is coded at 256 kb/s and that of the other five streams at 128 kb/s (all examples use 10 fps). Experimental results in the table demonstrate that the SQRT model is very accurate in a variety of FGS-coded video sequences.

We finally examine the accuracy of SQRT in H.264 PFGS. Recall that PFGS uses prediction in the enhancement layer to achieve better compression in sequences with high degrees of temporal correlation. Assuming that all predicted bits are transmitted to the client, our derivations and models are applicable to PFGS. Fig. 10 shows that model (31) also significantly outperforms the traditional R-D models in H.264 PFGS. The figure demonstrates that UQ, Cauchy, and Chiang's model exhibit large error variations in these sequences, which happens because PFGS not only uses the enhancement layer for prediction but also for reconstruction, which is beyond the range of these models.

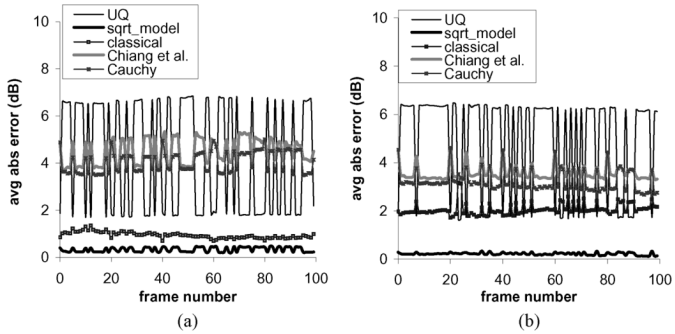


Fig. 10. Average absolute PSNR error in H.264 PFGS. Test sequences coded at 128 kb/s (base layer) and 10 fps. (a) CIF mobile; (b) CIF coastguard.

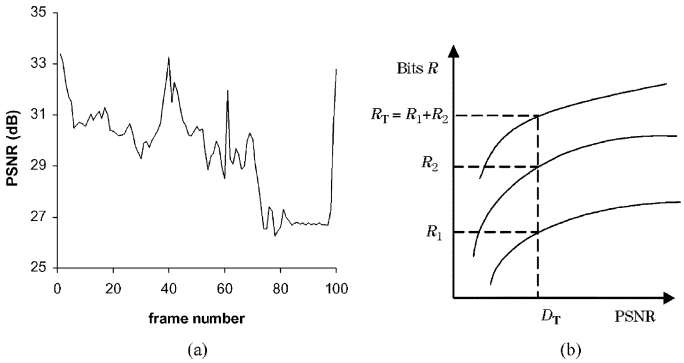


Fig. 11. (a) Base layer quality of MPEG-4 FGS-coded CIF Foreman. (b) Combined R-D curves in a two-frame sequence given target rate R_T and constant quality/distortion D_T.

We conclude this section by noting that (31) takes the following simple shape in the distortion domain:

$$D = c2^{aR+b\sqrt{R}} \tag{32}$$

where $a < 0$ and c is proportional to the source variance. Note that the parameters of (32) can be estimated from several empirical (R, D) samples and that this model is a generalization of the traditional R-D function $D = c2^{-2R}$ in which $b = 0$.

V. R-D BASED QUALITY CONTROL IN INTERNET STREAMING

In streaming applications, fluctuating visual quality is often unpleasant to the humans, who are normally used to relatively constant quality found in broadcast TV, VCR, and DVD programming [37], [38]. However, due to the inherent nature of current video coding schemes, the base layer usually suffers from substantial quality fluctuation as shown in Fig. 11(a) for Foreman CIF (note a 6-dB drop in quality within just a 10-s fragment).

Therefore, one of the goals of streaming servers is often to select such quantities of the enhancement layer that provide constant PSNR quality at the receiver. Furthermore, by rescaling the enhancement layer according to its R-D curve, the server can not only provide low fluctuation of video quality, but also match the available bandwidth in the network. The latter goal is achieved by coupling rate-scaling decisions with congestion control. In what follows in this section, we first describe very simple R-D based constant-quality streaming and then examine asymptotically stable congestion control methods that provide

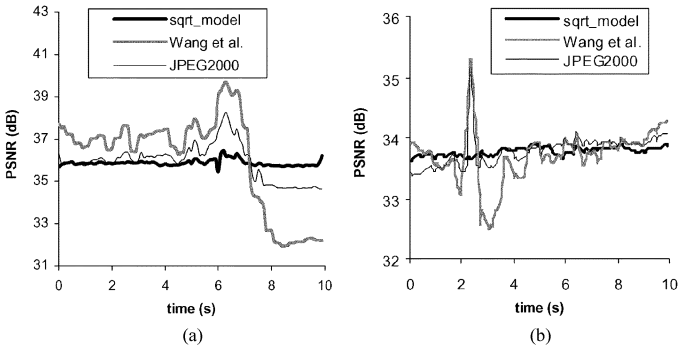


Fig. 12. Comparison in CBR streaming between our R-D model, the method from [32], and rate control in JPEG2000 [16]. (a) CIF foreman; (b) CIF coast-guard.

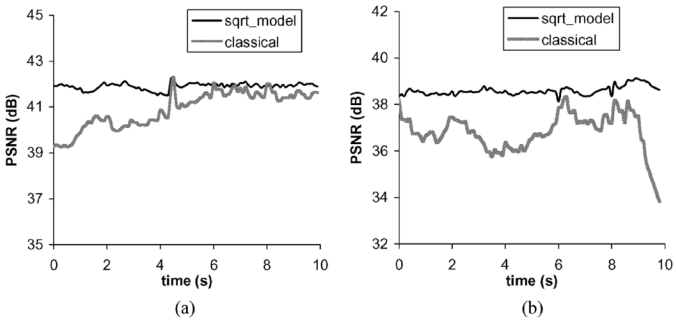


Fig. 13. Performance of our quality control in CBR channels with SQRT and classical R-D models. (a) QCIF table tennis; (b) QCIF stefan.

a foundation for oscillation-free transport of video over the Internet.

A. Quality Control Algorithm

As we mentioned earlier, rate control is one popular application of R-D models. The main question here is how to scale the FGS layer to both match the available bandwidth R_T (total bits allowed for the entire sequence) and achieve certain *constant* quality D after decoding. We illustrate the solution to this problem using Fig. 11(b) and a simple sequence consisting of two frames. First, the server inverts the result in (31) or (32) and obtains two $R(D)$ curves (one for each frame). Second, it generates the combined rate curve $R_1(D) + R_2(D)$, which shows the amount of *total* bits required to achieve constant D in both frames. Knowing R_T, the combined curve needs to be inverted one more time to obtain the value of D_T that corresponds to the total required bitrate R_T. The size of individual frames is then given by $R_1(D_T)$ and $R_2(D_T)$ as the final step.

For longer sequences, the server adds the R-D curves of all frames and obtains a combined function $F(D)$, which is constrained by R_T

$$F(D_T) = \sum_{i=t}^{N} R_i(D_T) = R_T, \tag{33}$$

where $R_i(D)$ is the R-D function of frame i, N is the number of frames in the sequence, and t the frame at which the server decides to change its rate R_T in response to congestion signals. Partial summation in (33) is important since congestion control often changes its rate in the middle of actual streaming and (33) needs to be recomputed every time such a change is encountered. Finding the root of (33) involves inverting $F(D)$ and evaluating

$$D_T = F^{-1}(R_T). \tag{34}$$

Once D_T is known, each enhancement layer frame i is scaled to $R_i(D_T)$ and then transmitted to the receiver. Even though the new R-D framework does not lead to a closed-form solution for F^{-1}, each of the individual curves can be generated with high accuracy using only a 3-point interpolation and the resulting function $F(D)$ can be computed (and then inverted) very efficiently.

In Fig. 12, we illustrate this simple rate control algorithm applied to the SQRT R-D model assuming that the channel capacity is fixed (variable channel rates are studied in the next section). The figure shows simulation results using Foreman CIF with 768 kb/s available in the network for the enhancement layer in comparison with two other rate-control methods—those proposed in the JPEG2000 [16] image coding standard and in Wang *et al.* [32]. Experimental results show that the new R-D framework can be successfully used to both dramatically reduce undesirable quality fluctuation during streaming and to relieve the server from expensive interpolation. The variance in PSNR between adjacent frames in the SQRT curve is only 0.04 dB in Fig. 12(a) and 0.004 dB in Fig. 12(b).

To demonstrate the importance of an accurate R-D model in quality control, we also compare simulation results of our quality control scheme (34) coupled with the proposed SQRT model (31) and the classical R-D model (4) in two sequences. In Fig. 13, we show that our SQRT model generates much smaller quality variation than the classical model in both QCIF Table Tennis and QCIF Stefan.

Many constant quality control approaches in related work stop after solving the problem for CBR channels [32], [37], [38]. We, on the other hand, find that neither the exact method of scaling the enhancement layer (this section), nor the underlying R-D model (the previous section) is very important if the application relies on any of the wide variety of AIMD-style congestion control methods. Hence, we feel that with goals of constant-quality streaming, it becomes more important to continue the research into the area of smooth congestion control, which is a pre-requisite to actual implementation of any of these methods. Unfortunately, the current Internet does not provide an environment where smooth (asymptotically stable) sending rates can be easily achieved; however, there are promising classes of congestion controllers for the future Internet that may fulfill these requirements. One such class is studied next.

B. Congestion Control Overview

There are many challenges facing Internet streaming applications, all of which stem from the lack of quality-of-service (QoS) guarantees in the transport layer. One of the primary impediments to high-quality delivery of real-time video to the end-user is the *variable* channel bandwidth. Notice that even though end-to-end paths often experience relatively stationary conditions (in terms of the number of competing flows, average

long-term packet loss, etc.), current congestion control methods built on top of a variety of TCP-friendly schemes cannot asymptotically converge (from a control-theoretic point of view) to a single stationary rate or provide a smooth "virtual" channel to the video application.

Recently, a major effort has been dedicated to developing smoother congestion control methods for multimedia streaming (e.g., TFRC [9] and binomial algorithms [1]). Nevertheless, these methods are not asymptotically stable, nor do they have any stationary points in the feasible operating range of a typical application.

In this section, we study continuous-feedback congestion controllers proposed by Kelly *et al.* [19] and investigate whether their performance provides the necessary foundation for achieving the goals of this paper.

C. Kelly Controls

Recall that TCP and classical binary-feedback methods (such as AIMD and binomial algorithms) rely on packet loss in order to increase or decrease their rates. Since the decision about changing the current rate is binary, we can summarize their control functions as follows:

$$\frac{dr}{dt} = [1 - sgn(p)]\, F(r) - sgn(p) G(r) \tag{35}$$

where $r(t)$ is the rate at time t, p is the current packet loss rate, $F(r)$ is the increase function, and $G(r)$ is the decrease function. Notice that with a reasonable choice of functions F and G, the right side of (35) does not have roots, which means that the equation does not have stationary points. Since (35) cannot be stabilized, it must oscillate or diverge. It is easy to show that under certain mild conditions on $F(r)$ and $G(r)$, (35) oscillates around the equilibrium (equal-share) rate. The amount of oscillations depends on the choice of $F(r)$ and $G(r)$ and typically leads to a tradeoff between the size of oscillations and the speed of response to congestion signals.

What is interesting about binary-feedback methods is that they typically do not possess any methods that can force the oscillations to asymptotically decay to zero, even under *stationary* cross-traffic conditions. Therefore, we seek alternative methods that provide this functionality and are provably stable under both immediate and *delayed* feedback. One such alternative is given by Kelly's congestion control framework called *proportional fairness* [19]:

$$\frac{dr}{dt} = r\left(\alpha U'(r) - \beta \sum_{l \in P} p_l\right), \tag{36}$$

where $U(r) = \log r$ is the utility function of the end-user, $\alpha > 0$ and $\beta > 0$ are constants, and p_l is the price that the flow pays for using resource (router) l along the end-to-end path P. Kelly's controls have received significant attention in the theoretical networking community [15], [19], [20], [25]; however, their application in real networks or streaming applications has been limited.

Several modifications to the original framework (36) are necessary to make this controller practical. First, it is common to use packet loss as the continuous feedback (instead of the price) simply because the current Internet is still best-effort and prices are a meaningless metric for individual routers. Second, instead of summing up the packet loss experienced by *all* routers of an end-to-end path, it sometimes makes more sense to use the *maximum* packet loss among these routers in order to match the rate of the application to the bandwidth of the *slowest* link in the path:

$$p(t) = \max_{l \in P} p_l(t). \tag{37}$$

Expanding (36) using a *single* feedback $p(t)$ of the most-congested resource and converting the system into the discrete domain, we have a more application-friendly version of the controller:

$$r_i(t) = r_i(t - D_i) + \alpha - \beta r(t - D_i) p\left(t - D_{li}^{\leftarrow}\right) \tag{38}$$

where i is the flow number, D_i is its round-trip delay, and $D_{li}^{\leftarrow}$ is the backward feedback delay from router l to user i. Note that this version of Kelly controls includes max–min changes to the feedback and an extra delay applied to the additive term $r_i(t - D_i)$ in (38).

Full analysis of this framework is beyond the scope of this paper, but the following important result is available in [36].

Lemma 6: Discrete controller (37), (38) is asymptotically stable and fair regardless of round-trip delays D_i, the exact shape of packet loss $p(t)$, or feedback delays $D_{li}^{\leftarrow}$ as long as $0 < \beta < 2$.

Our final issue to address is the shape of packet loss $p(t)$. While (37) and (38) can operate in the end-to-end context where $p(t)$ is estimated by the receiver, we find that involvement of explicit feedback significantly improves the performance of this controller. To accomplish such a functionality, each router performs a very simple operation of counting the total arriving traffic into each queue, dividing the result by the fixed duration of the control interval, and inserting feedback $p_l(t)$ into packets passing through the queue:

$$p_l(t) = \frac{\sum_{i \in S_l} r_i(t) - C_l}{\sum_{i \in S_l} r_i(t)} \tag{39}$$

where S_l is the set of flows passing through resource l and C_l is the speed of the resource (i.e., its outgoing bandwidth). Notice that the router does not need to count the number of flows or estimate their individual rates r_i. This means that the feedback is based on the *aggregate* flow rate $\sum_{i \in S_l} r_i(t)$ rather than individual flow statistics. For additional implementation discussion, see [18].

It is also possible to demonstrate that the convergence rate of Kelly controls is at least exponential, which makes this framework appealing for future high-speed networks. We illustrate this result in Fig. 14, in which $\beta = 0.5$ and $\alpha = 10$ kb/s. The figure shows that it takes 8 steps for a single-flow to fill a 1.5-mb/s T1 bottleneck and it takes only 16 steps for the same flow to fill a 10 gb/s link. Note that both flows reach within 5% of link capacity in just 6 RTT steps.

D. SQRT Quality Control in VBR Networks

We finish this section by examining the PSNR quality curves when the target rate R_T is not know *a-priori*, but is instead supplied by real-time congestion control (38), (39). We obtained

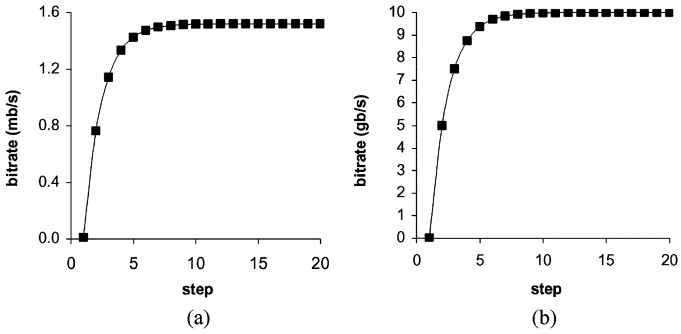


Fig. 14. Exponential convergence of (38). (a) $C = 1.5\ \mathrm{mb/s}$; (b) $C = 10\ \mathrm{gb/s}$.

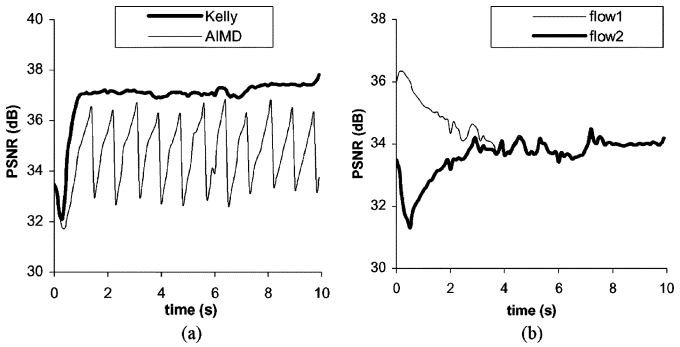


Fig. 15. (a) Comparison of AIMD and Kelly controls over a 1 mb/s bottleneck link. (b) Kelly controls with two flows starting in unfair states.

the traces of $r(t)$ from ns2 simulations and then applied them to the video scaling algorithm offline. We should point out that one limitation of this approach is that we did not take into account the effect of lost packets during the simulation on the quality of the stream. This is reasonable in streaming scenarios where the application protects its packets by FEC or some form of retransmission. Since in Kelly controls, the amount of packet loss p^* in the steady state is fixed and known to the end flow once it reaches the equilibrium [36], it becomes easy to send enough FEC to cover the exact amount of lost data.

To set a baseline example, Fig. 15(a) compares TCP-like AIMD control with modified framework (38), (39) using PSNR quality curves. In this simulation, a single flow is run over a bottleneck resource of capacity $C = 1\ \mathrm{mb/s}$ and round-trip delay 100 ms. As the figure shows, both controls at first follow the PSNR of the base layer since there is not enough discovered bandwidth to send any FGS data. Once this stage is passed, both controls achieve high PSNR; however, the difference is that AIMD backs off by half upon every packet loss, while Kelly controls eventually stabilize at a fixed rate. Rate fluctuation in AIMD results in periodic jumps (sometimes as high as 4 dB) throughout the entire sequence.

Fig. 15(b) shows another scenario where two Kelly flows are sharing the same bottleneck link C under identical 100-ms round-trip delays. Flow_1 in the figure is started with $r_1(0) = C$ and flow_2 is started with its base-layer bandwidth. As seen in the figure, the two flows converge to a fair allocation at approximately $t = 3$ s and then follow the same flat quality curve that is perfectly fair.

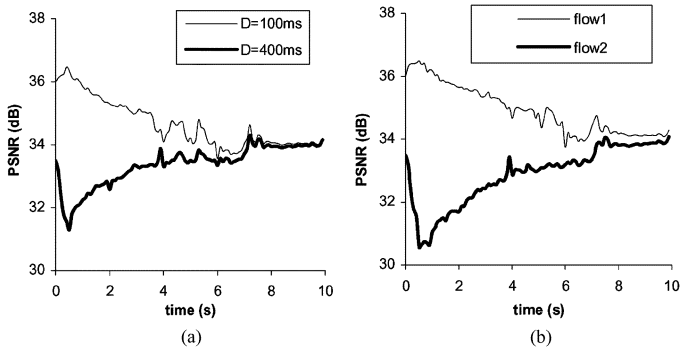


Fig. 16. PSNR comparison of (a) two flows with different (but fixed) round-trip delays D and (b) two flows with random round-trip delays.

The next issue to examine is whether different round-trip delays D have any effect on fairness. Fig. 16(a) shows a scenario in which two flows with different RTTs start in the same unfair states as before. The corresponding delays are 400 and 100 ms; however, this has little effect on the resulting fairness as both flows stabilize at 34.5 dB around $t = 7$ s .

We finally examine the effect of *random time-varying* feedback delays on our quality-control framework, in which the round-trip delay is now uniformly random between 100 and 400 ms and the initial states are as before. Fig. 16(b) shows that although the convergence is somewhat slower than in the previous examples, both flows manage to achieve stable quality after convergence. This confirms our earlier result regarding stability of (38), and (39) under arbitrary delays.

In summary, Kelly controls converge to equilibrium without oscillation and then stay there as long as the number of flows at the bottleneck remains fixed. When new flows join or leave, the transition between fair (equilibrium) points is monotonic in most situations. This provides a nice foundation for video-on-demand and other entertainment-oriented video services where each flow is long-lived and can take full advantage of this smooth congestion control framework.

VI. Conclusion

This paper presented a detailed analysis of distortion in fine-granular scalable coders and provided an accurate R-D model for FGS-like sequences. After obtaining an efficient operational R-D model, we applied it to Internet streaming for quality control purposes and demonstrated in simulations that our algorithm worked well at achieving constant quality not only in CBR networks, but also in VBR channels coupled with congestion control. To overcome the limitations of TCP-friendly methods, we used modified Kelly controls and showed that they could achieve stable sending rates in practical network environments and provided an appealing framework for future high-speed networks.

References

[1] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 631–640.

[2] J. A. Bilmes, A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, Univ. California, Berkeley, Tech. Rep. ICSI-TR-97-021, Apr. 1998.

[3] J.-J. Chen and D. W. Lin, "Optimal bit allocation for coding of video signals over ATM Networks," *IEEE. J. Sel. Areas Commun.*, vol. 15, no. 6, pp. 1002–1015, Aug. 1997.
[4] T. Chiang and Y. Q. Zhang, "A new rate control scheme using quadratic distortion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 246–250, Feb. 1997.
[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[6] M. Dai, D. Loguinov, and H. Radha, "Statistical analysis and distortion modeling of MPEG-4 FGS," in *Proc. IEEE ICIP*, Sept. 2003, pp. 301–304.
[7] R. A. Devore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 719–746, Mar. 1992.
[8] 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, no. 1, pp. 12–20, Feb. 1996.
[9] S. Floyd, M. Handley, and J. Padhye, "Equation-based congestion control for unicast applications," in *ACM SIGCOMM*, Aug. 2000, pp. 45–58.
[10] R. M. Gray, *Source Coding Theory*. Norwell, MA: Kluwer, 1990.
[11] H.-M. Hang and J.-J. Chen, "Source model for transform video coder and its application—part I: fundamental theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 197–211, Apr. 1997.
[12] Z. He and S. K. Mitra, "A unified rate-distortion analysis framework for transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1221–1236, Dec. 2001.
[13] C.-Y. Hsu, A. Ortega, and A. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints," *IEEE. J. Sel. Areas Commun.*, vol. 15, no. 6, pp. 1016–1027, Aug. 1997.
[14] N. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
[15] R. Johari and D. Tan, "End-to-end congestion control for the internet: delays and stability," *IEEE/ACM Trans. Networking*, vol. 9, no. 6, pp. 818–832, Dec. 2001.
[16] *JEG2000 Part I Final Committee Draft Ver. 1.0*, ISO/IEC JTCI/SC29 WGI, JPEG, Mar. 2000.
[17] N. Kamaci, Y. Altunbasak, and R. M. Mersereau, "Frame bit allocation for the H.264/AVC video coder via cauchy-density-based rate and distortion models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp. 994–1006, Aug. 2005.
[18] S.-R. Kang, Y. Zhang, M. Dai, and D. Loguinov, "Multi-layer active queue management and congestion control for scalable video streaming," in *Proc. IEEE ICDCS*, Mar. 2004, pp. 768–777.
[19] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, Nov. 1998.
[20] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: utility functions, random losses and ECN marks," in *IEEE INFOCOM*, Mar. 2000, pp. 1323–1332.
[21] D. Leviatan and I. A. Shevchuk, "Coconvex approximation," *J. Approx. Theory*, vol. 118, no. 1, pp. 20–65, Sept. 2002.
[22] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301–317, Mar. 2001.
[23] J. Lin and A. Ortega, "Bit-rate Control using piecewise approximation rate-distortion characteristics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 446–459, Aug. 1998.
[24] F. Ling, W. Li, and H. Sun, "Bitplane coding of DCT coefficients for image and video compression," in *SPIE Conf. Visual Communications and Image Processing*, Jan. 1999, pp. 500–508.
[25] L. Massoulié, "Stability of distributed congestion control with heterogeneous feedback delays," *IEEE Trans. Automat. Contr.*, vol. 47, no. 6, pp. 895–902, June 2002.
[26] *Coding of Moving Pictures and Audio*, ISO/IEC JTC1/SC29/WG11 N3908, MPEG, Jan. 2001.
[27] A. N. Netravali and B. G. Haskell, *Digital Pictures Presentation, Compression, and Standards*. New York: Plenum, 1988.
[28] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.
[29] H. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 53–68, Mar. 2001.
[30] N. M. Rajpoot, "Simulation of the rate-distortion behavior of a memoryless laplacian source," in *Middle Eastern Symp. Simulation and Modelling*, Sep. 2002.
[31] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.
[32] Q. Wang, Z. Xiong, F. Wu, and S. Li, "Optimal rate allocation for progressive fine granularity scalable video coding," *IEEE Signal Process. Lett.*, vol. 9, no. 2, pp. 33–39, Feb. 2002.
[33] Q. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Upper Saddle River, NJ: Prentice-Hall, 2001.
[34] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Transporting real-time video over the internet: challenges and approaches," *Proc. IEEE*, vol. 88, no. 12, pp. 1855–1875, Dec. 2000.
[35] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 332–344, Mar. 2001.
[36] Y. Zhang, S.-R. Kang, and D. Loguinov, "Delayed stability and performance of distributed congestion control," in *ACM SIGCOMM*, Aug. 2004, pp. 307–318.
[37] L. Zhao, J. W. Kim, and C.-C. Kuo, "MPEG-4 FGS video streaming with constant-quality rate control and differentiated forwarding," in *SPIE VCIP*, Jan. 2002, pp. 230–241.
[38] X. Zhao, Y. He, S. Yang, and Y. Zhong, "Rate allocation of equal image quality for MPEG-4 FGS video streaming," in *Packet Video Workshop*, Apr. 2002.

Min Dai (S'04–M'05) received the B.S. and M.S. degrees in electrical engineering from Shanghai Jiao Tong University, China, in 1996 and 1998, respectively, and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, in 2004.

She currently works in the MediaFLO Video System Team at Qualcomm, Inc., San Diego, CA. Her research interests include scalable video streaming, R-D modeling, video traffic analysis, image denoising and classification.

Dmitri Loguinov (S'99–M'03) received the B.S. degree (with honors) in computer science from Moscow State University, Russia, in 1995 and the Ph.D. degree in computer science from the City University of New York in 2002.

Since 2002, he has been an Assistant Professor of computer science with Texas A&M University, College Station. His research interests include peer-to-peer networks, congestion control, Internet video streaming, topology modeling, traffic measurement, overlay networks, and video coding.

Hayder M. Radha (M'92–SM'01) received the B.S. degree (with honors) from Michigan State University (MSU), East Lansing, in 1984, the M.S. degree from Purdue University, West Lafayette, IN, in 1986, and the Ph.M. and Ph.D. degrees from Columbia University, New York, in 1991 and 1993, all in electrical engineering.

He joined MSU in 2000 as Associate Professor in the Department of Electrical and Computer Engineering. Between 1996 and 2000, he worked at Philips Research USA, where he initiated the Internet Video project and led a team of researchers working on scalable video coding and streaming algorithms. Prior to working at Philips, he was a Distinguished Member of Technical Staff at Bell Labs, where he worked between 1986 and 1996 in the areas of digital communications, signal/image processing, and broadband multimedia. His research interests include image and video coding, wireless technology, multimedia communications and networking. He has more than 20 patents in these areas.

Dr. Radha served as Co-Chair and Editor of the ATM and LAN Video Coding Experts Group of the ITU-T in 1994–1996. He is a Philips Research Fellow.