Rate Control in DCT Video Coding for Low-Delay Communications

Jordi Ribas-Corbera, *Member, IEEE,* and Shawmin Lei, *Senior Member, IEEE*

***Abstract*—An important motivation for the development of the emerging H.263+ and MPEG4 coding standards is to enhance the quality of highly compressed video for two-way, real-time communications. In these applications, the delay produced by bits accumulated in the encoder buffer must be very small, typically below 100 ms, and the rate control strategy is responsible for encoding the video with high quality and maintaining a low buffer delay. In this work, we present a simple rate control technique that achieves these two objectives by smartly selecting the values of the quantization parameters in typical discrete cosine transform video coders. To do this, we derive models for bit rate and distortion in this type of coders, in terms of the quantization parameters. Using Lagrange optimization, we minimize distortion subject to the target bit constraint, and obtain formulas that indicate how to choose the quantization parameters. We implement our technique in H.263 and MPEG4 coders, and compare its performance to TMN7 and VM7 rate control when the encoder buffer is small, for a variety of video sequences and bit rates. This new method has recently been adopted as a rate control tool in the test model TMN8 of H.263+ and (with some modifications) in the verification model VM8 of MPEG4.**

***Index Terms*—Bit allocation, H.263, MPEG, quantization, rate control, rate-distortion optimization, video coding.**

I. Introduction

In real-time video communications, the end-to-end delay for transmitting video data needs to be very small, particularly in two-way interactive applications such as videophone or videoconferencing. If the encoded video is transmitted through a fixed-rate channel, the coded bits are placed into a small buffer and a finite number of bits B can be sent from the buffer during each frame interval (where B is the channel rate divided by the frame rate). If one or several frames occupy more than B bits, the additional bits accumulate in the encoder buffer[1] and increase the *buffer delay*, which is the time needed (at the beginning of a frame's interval) to send the buffer bits remaining from previous frames. When the number of bits in the buffer is too large, the encoder usually skips encoding frames to reduce the buffer delay and avoid buffer overflow. The frame skipping produces undesirable motion discontinuity in the encoded video sequence. Conversely, if several frames occupy fewer than B bits, there may be periods of time in which no bit is transmitted through the channel, and hence some channel bandwidth is wasted.

Discrete cosine transform (DCT) video coders based on the H.263+ and MPEG4 coding standards are expected to have better compression efficiency than those compliant with previous standards, particularly at low bit rates. But since standards only specify the decoder's syntax, the encoder must select some rate control strategy that allocates the available bits, and not surprisingly, the quality of the encoded video depends significantly on the efficiency of the rate control technique. In this type of coder, the rate control performs the bit allocation by selecting the encoder's quantization parameters, which consist of a quantizer step size or scale for each block of 16×16 pixels.

Several rate control methods in the literature encode each image block several times with different quantization parameters, and then intelligently select the best parameters [1]–[3], but these approaches are generally not suitable for real-time encoding because of their high computational complexity. Other approaches select the quantizers according to a formula derived from some model of the encoder, where the formula uses several simple measurements such as the number of bits available, the fullness of the encoder buffer, the variance of the pixels in the image blocks, etc. [5], [7], [14], [15], [22], [23]. However, these simple model-based approaches generally do not achieve the target number of bits B accurately, and while they are adequate in applications that allow large buffer delays (where many bits can accumulate in the encoder buffer), they can suffer from frequent frame skipping and wasting of channel bandwidth in low-delay applications.

In this work, we present a simple, yet fairly effective model-based rate control scheme for operating DCT-based video coders in low-delay environments. Our method focuses on rate control for the motion-compensated or intercoded frames, which is the type of frames commonly used in low-delay scenarios, and selects the encoder's quantization parameters within each frame to accurately meet the target number of bits B and encode that frame with high image quality. To do this, we find models for the encoding bit rate and distortion in a given frame in terms of the quantizer parameters. Using Lagrange optimization, we minimize the frame distortion subject to the target B and obtain formulas for the best quantization parameters. These formulas indicate that the optimized quantizers are a function of B, the pixel variances of the motion-compensated blocks, and some parameters of our rate model. We implement our new rate control strategy in an

Manuscript received October 23, 1997; revised May 14, 1998. This work was presented in part at the Picture Coding Symposium '97, Berlin, Germany, September 1997. This paper was recommended by Associate Editor M.-T. Sun.

The authors are with the Department of Digital Video, Sharp Labs of America, Camas, WA 98607 USA.

Publisher Item Identifier S 1051-8215(99)02041-8.

[1] In this work, we will focus on the regulation of the encoder buffer, whose fullness is related to that of the decoder buffer [26].

H.263 and an MPEG4 coder, and compare its performance to that of the well-known rate control techniques in TMN7 [5] and VM7 [14], [15].

The experimental results indicate that our technique performs an effective bit allocation while maintaining a low buffer delay. In comparison to TMN7 and VM7 rate control, our method significantly improves the motion continuity by dropping fewer frames and increases the video quality up to 1 dB in average PSNR. This new rate control scheme has been adopted in TMN8, a recent test model of the H.263+ standard [6] and, with some modifications, as one of the rate control tools in the video verification model VM8 [28] of MPEG4 (version 1).

In Section II, we derive models for bit rate and distortion in DCT video coding and discuss previous work on this topic. In Section III, we find the optimal quantization parameters that minimize distortion subject to the target bit budget. Our new rate control technique, which uses the optimized quantizers, is described in Section IV. The experimental results and a brief description of TMN7 and VM7 rate control are presented in Section V. Finally, our summary and conclusions are presented in Section VI.

II. Modeling Rate and Distortion in DCT Video Coding

In typical block-based video coders, such as those based on H.263 [5], [6] and MPEG [4], [15], [28] standards, the current video frame to be encoded is decomposed into macroblocks of 16×16 pixels per block, and the pixel values for each of the four 8×8 blocks in a macroblock are transformed into a set of coefficients using the DCT. These coefficients are then quantized and encoded with some type of variable-length coding. The number of bits and distortion for a given macroblock depend on the macroblock's quantization parameter, which determines the step size used for quantizing the transformed coefficients. For example, in H.263, the quantization parameter is denoted as QP, and its value corresponds to half the quantization step size. In this section, our goal is to model the effect of the macroblocks' quantization step sizes on encoding bit rate and distortion. We focus on modeling rate and distortion for motion-compensated or intercoded frames. Similar models could be obtained for intracoded frames, but we will not derive them here because these frames usually occupy many bits and require several frame intervals to be transmitted and, as a result, they are not commonly used for low-delay communications.

A. Rate Model

We initially focus on gray-scale video sequences. As in previous work [16], we assume that the DCT coefficients of the motion-compensated difference frame are approximately uncorrelated and Laplacian distributed with variance σ^2. Since the DCT is an orthogonal transformation, σ^2 is also the variance of the difference frame pixels. The DCT coefficients are quantized with a uniform scalar quantizer of step size Q, and the difference frame rate (in bits per pixel) is $R(Q) \approx H(Q)$, where $H(Q)$ is the empirical entropy of the Q-quantized coefficients. We use the approximation in [9] and [18] for $H(Q)$

$$H(Q) = \begin{cases} \dfrac{1}{2}\log_2\left(2e^2\dfrac{\sigma^2}{Q^2}\right), & \dfrac{\sigma^2}{Q^2} > \dfrac{1}{2e} \\ \dfrac{e}{\ln 2}\dfrac{\sigma^2}{Q^2}, & \dfrac{\sigma^2}{Q^2} \le \dfrac{1}{2e}. \end{cases} \quad (1)$$

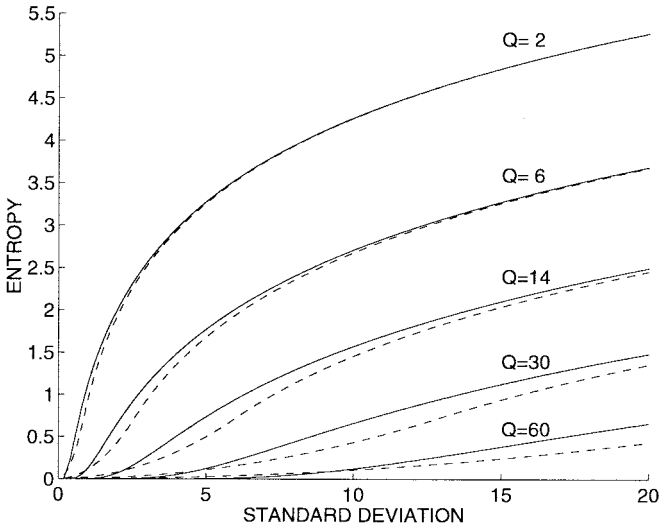


Fig. 1. From top to bottom, the solid lines are the entropy of a Q-quantized Laplacian distribution with respect to standard deviation σ when $Q = 2$, 6, 14, 30, 60. The dashed lines are the respective approximations obtained using (1).

Note that σ^2/Q^2 will be larger than $1/(2e)$ for small values of Q (low distortion or high-rate case) and smaller for large values of Q (high distortion or low-rate case). The approximation in (1) for the high rate is obtained by relating the entropy $H(Q)$ to the differential entropy of a Laplacian [19, p. 228] and for the low rate by choosing the linear function of σ^2 to meet the log function at a point of equal slope, which turns out to be at $\sigma^2 = Q^2/(2e)$. In Fig. 1, for several values of Q, we plot $H(Q)$ as a function of σ using the formula in [17] for the exact entropy of a Q-quantized Laplacian, and we also plot the approximation in (1).

In practice, the empirical rate of typical DCT-based coders is lower than that predicted by (1). This is because our assumptions above are only approximations. In particular, there is usually some correlation left among the DCT coefficients, and these coders use some additional techniques such as run-length coding of the quantized coefficients or a wider quantization step size around zero, which reduce the rate in comparison to uniform scalar quantization with entropy coding. However, rate–distortion theory suggests that the rate curves of these more advanced coders should be lower than that in (1) by approximately a constant, and hence they are expected to have a similar form or shape. To verify this, we performed some empirical experiments that are described in Fig. 2. These experiments confirm that at a high bit rate, the rate curves have a convex-cap shape similar to that of a logarithm, and at a low bit rate, the curves follow a convex-cup form that can be approximated by a quadratic formula.

From now on, we focus on the more interesting low rate case in (1), which typically corresponds to PSNR's below 40 dB. We first select a set of macroblocks with similar statistics, and

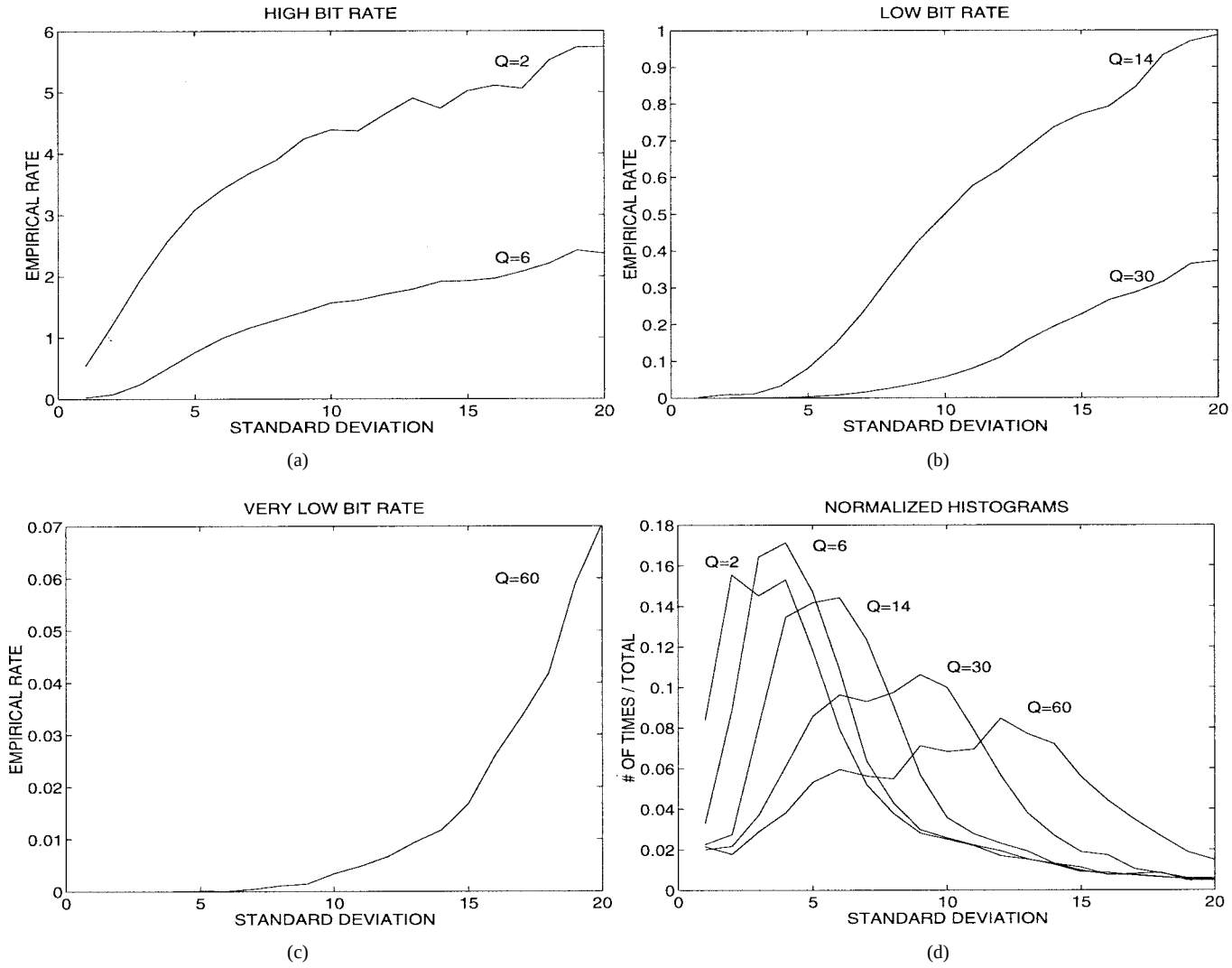


Fig. 2. (a)–(d). Verification of the functional form of the encoding rate (in bits/pixel) as a function of standard deviation σ. These results were obtained by encoding 10 s of the video sequence "foreman" (QCIF) at 10 frames/s, using Telenor's H.263 encoder [12] with a fixed quantization parameter QP ($Q = 2QP$). At each macroblock, we measured σ (rounded to the nearest integer) and the number of luminance bits. The curves plot the average rate (in bits/pixel) for each σ. Specifically, (a) shows the rate curves for $Q = 2$ and 6 (or $QP = 1$ and 3, respectively), (b) for $Q = 14$ and 30, and (c) for $Q = 60$. Finally, (d) plots the normalized histograms of the values taken by σ in each case; for higher values of Q, the increased quantization noise makes the histogram peaks move to the right. The rate curves in (a)–(c) are rougher for larger values of σ because there are fewer samples to average.

find a model for the average number of DCT bits produced by such a set. Specifically, let S_V be the set of macroblocks in a motion-compensated difference frame whose standard deviation σ is in the interval $(V-\delta, V+\delta]$, where δ is a small value. For example, V could be an integer and $\delta = 0.5$. We fix a common quantization step size Q_V for all of the macroblocks in S_V and, given (1) and our empirical results, select the following model for the average encoding rate (in bits per pixel) R_V for the DCT coefficients in S_V:

$$R_V = \frac{1}{AN_V}\sum_{n=1}^{N_V} B_{V,n} \approx K\frac{V^2}{Q_V^2} \tag{2}$$

where N_V is the number of macroblocks in S_V, $B_{V,n}$ is the number of DCT bits produced by the nth macroblock in S_V, and A is the number of pixels in a macroblock (i.e., $A = 16^2$ pixels). The value of K could be set to $e/\ln 2$ if the DCT coefficients were Laplacian distributed and independent, as in (1). But since these assumptions are only approximate, we will adapt K to the statistics of the specific frame during encoding. Next, we model the average or expected number of bits produced by the ith macroblock in a frame as follows:

$$B_i = A\left(K\frac{\sigma_{L,i}^2}{Q_i^2} + C\right) \approx A\left(K\frac{V_i^2}{Q_{V_i}^2} + C\right) \tag{3}$$

where $\sigma_{L,i}$ is the macroblock's empirical standard deviation and is a value in $(V_i-\delta, V_i+\delta]$. $Q_i = Q_{V_i}$ is the quantization step used for that macroblock. The constant C in (3) models the average rate (in bits/pixel) to encode the motion vectors and the coder's header and syntax for the frame. We refer to C as the *overhead rate*, and its value is also estimated during encoding. The estimation of C and K is discussed in Section IV.

In color video sequences, the bit rate produced for encoding the DCT chrominance values is usually small, and could be considered in C as part of the overhead rate. But the chrominance bits can also be modeled equivalently as the

luminance bits in (3)

$$\begin{aligned} B_i &= AK\frac{\sigma_{L,i}^2}{Q_i^2} + A'K\frac{\sigma_{C,i}^2}{Q_i^2} + AC \\ &= AK\frac{\sigma_{L,i}^2 + \sigma_{C,i}^2(A'/A)}{Q_i^2} + AC \\ &= A\left(K\frac{\sigma_i^2}{Q_i^2} + C\right) \end{aligned} \tag{4}$$

where A' is the number of chrominance values in the ith macroblock, $\sigma_{C,i}^2$ is the variance of these values, and $\sigma_i^2 = \sigma_{L,i}^2 + \sigma_{C,i}^2(A'/A)$. We refer to σ_i^2 as the variance of the ith macroblock. For simplicity, we assume that the mean of the luminance equals that of the chrominance (since, in motion-compensated macroblocks, both means are typically very small anyway) and compute σ_i^2 as follows:

$$\begin{aligned} \sigma_i^2 &= \frac{1}{A}\sum_{j=1}^{A}\left(P_i(j) - \overline{P}_i\right)^2 + \frac{A'}{A}\frac{1}{A'}\sum_{j=A+1}^{A+A'}\left(P_i(j) - \overline{P}_i\right)^2 \\ &= \frac{1}{A}\sum_{j=1}^{A+A'}\left(P_i(j) - \overline{P}_i\right)^2 \end{aligned} \tag{5}$$

where $P_i(j)$ is the jth value of the set of luminance and chrominance samples (in which the first A samples are of luminance and the next A' are of chrominance) and $\overline{P}_i$ is the sample average.

Other models for the bit-quantizer performance of DCT coders have been proposed in the literature [2], [3], [7], [8], [14], [20], [22], [23], [27]. Some of them are based on classical rate–distortion models that lead to logarithmic expressions similar [7], [20], [22] or equal [8] to our log formula in (1). Our analysis using the Laplacian assumption and our empirical experiments in Fig. 2 indicate that these logarithmic models are more appropriate for a high bit rate. Other types of models have been proposed in the literature, such as power [2], spline [3], or polynomial based [14], [23]. The models in [2], [3], [14], and [23] are somewhat similar to our low-rate model in (4), but ours is simpler and explicitly uses the variance σ^2 and the overhead rate C in the model. Recently, a model that combines a logarithmic and a quadratic formula has also been proposed [27]. The more complex formulas in [2], [3], [14], [23], and [27] could have been used in this work, but we will see in Section III that the range of our optimized quantizers is usually small and, as shown in previous work [2], [23], a simple formula similar to ours is sufficient for localized modeling. Additionally, more complex formulas would likely make the optimization in Section III intractable or computationally expensive.

B. Distortion Model

The distortion in the ith macroblock is introduced by uniformly quantizing its DCT coefficients with a quantizer of step size Q_i. We consider the following typical distortion measure for the encoded macroblocks [8]:

$$D = \frac{1}{N}\sum_{i=1}^{N}\alpha_i^2\frac{Q_i^2}{12} \tag{6}$$

where N is the number of macroblocks in a frame and α_i is the *distortion weight* of the ith macroblock. The latter can be chosen to incorporate the importance or weight of that macroblock's distortion. For example, we could choose larger α_i's for macroblocks that belong to more important objects in the scene or in which artifacts are more visible to the human eye. However, in this paper, we do not explore the use of distortion weights for object-based or perceptual coding (see [29] for work on this topic). Instead, as we will see later in Section III-A, these weights will be used for reducing the quantization overhead (of encoding the Q_i's) at low bit rates. On the other hand, at medium and high bit rates (above 0.5 bits/pixel), the weights will be set to $\alpha_1 = \cdots = \alpha_N = 1$, and then our distortion measure is the well-known approximation to the mean-squared error (MSE) between the original and encoded macroblocks [10, p. 152]:

$$D_{\mathrm{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\frac{Q_i^2}{12} \tag{7}$$

in which $Q_i^2/12$ is the approximate MSE distortion for the ith macroblock.

At very low bit rates, there are often macroblocks whose Q_i's are three or four times larger than their respective σ_i's, and then the typical $Q_i^2/12$ approximation to the MSE in these macroblocks is not as effective. Usually, these Q_i's are so large (in relation to their σ_i's) that these macroblocks do not produce any bits, and in fact, some encoders [13] use a flag in such cases to indicate that the DCT coding is skipped. But we can still use a formula like (7) to model the MSE of the encoded (nonskipped) macroblocks, and hence the derivations and the general nature of the conclusions found in this work remain relevant to MSE even at very low bit rates.

III. Quantizer Optimization in DCT Video Coding

In this section, we derive formulas for the quantization step sizes that minimize distortion when typical DCT video coders operate at low bit rate. Our rate control technique will be based on such formulas. We also briefly describe earlier related work (at high bit rate) and discuss some conclusions learned from this analysis.

A. Optimization at Low Bit Rate

We want to find an expression for the quantization parameters $Q_1^*, \cdots, Q_N^*$ that minimizes the distortion in (6) subject to the constraint that the total number of bits, i.e., the sum of the macroblock B_i's (4), must be equal to B:

$$Q_1^*, \cdots, Q_N^* = \underset{\substack{Q_1,\cdots,Q_N \\ \sum_{i=1}^{N} B_i = B}}{\arg\min} \frac{1}{N}\sum_{i=1}^{N}\alpha_i^2\frac{Q_i^2}{12}. \tag{8}$$

Observe that, since we are minimizing a convex, differentiable function on a convex set, there is a unique solution that can be obtained using Lagrange theory [21]. To do this, we define λ and λ^* as, respectively, the Lagrange multiplier and its optimal value, and write the optimization problem in (8) in

its equivalent (unconstrained) form:

$$\begin{aligned} &Q_1^*, \cdots, Q_N^*, \lambda^* \\ &\quad = \underset{Q_1,\cdots,Q_N,\lambda}{\arg\min} \ \frac{1}{N} \sum_{i=1}^{N} \alpha_i^2 \frac{Q_i^2}{12} + \lambda \left[\sum_{i=1}^{N} B_i - B \right] \\ &\quad = \underset{Q_1,\cdots,Q_N,\lambda}{\arg\min} \ \frac{1}{N} \sum_{i=1}^{N} \alpha_i^2 \frac{Q_i^2}{12} \\ &\qquad + \lambda \left[\sum_{i=1}^{N} A\left(K \frac{\sigma_i^2}{Q_i^2} + C \right) - B \right] \end{aligned} \tag{9}$$

where we replaced B_i with (4) in the last step. By setting partial derivatives to zero in (9), we obtain the following expression for the optimized quantization step size:

$$Q_i^* = \sqrt{\frac{AK}{(B - ANC)} \frac{\sigma_i}{\alpha_i} \sum_{k=1}^{N} \alpha_i \sigma_i}, \qquad i = 1, \cdots, N. \tag{10}$$

Equation (10) is the key for our new rate control technique. It indicates that the optimized quantization step at the ith macroblock Q_i^* increases with the square root of the macroblock's standard deviation σ_i. Also, the Q_i^*'s increase with the sum of the weighted σ_i's, and decrease when there are more bits available for the frame (i.e., larger B).

We now set the distortion weights in (10) to one ($\alpha_1 = \cdots = \alpha_N = 1$), so that the optimal quantizers minimize the approximate MSE distortion (7). In that interesting case, we find when our optimization is correct by checking how small the bit rate needs to be so that the optimized quantizers in (10) belong to the low rate region in (1). Specifically, we want to find when the following inequality holds:

$$Q_i^{*2} = \frac{AK}{(B - ANC)} \sigma_i \sum_{k=1}^{N} \sigma_k \geq 2e\sigma_i^2, \qquad i = 1, \cdots, N. \tag{11}$$

After some straightforward manipulations, we find that (11) is equivalent to

$$\sigma_i \leq \frac{K}{2e \dfrac{B - ANC}{AN}} \frac{1}{N} \sum_{j=1}^{N} \sigma_i = \frac{K}{2er} \hat{\sigma} = \frac{0.72}{r} \hat{\sigma} \tag{12}$$

where R is the rate in bits/pixel for the DCT coefficients, and $\hat{\sigma}$ is the average of the macroblock standard deviations (we used $K = e/\ln 2$). In the typical case that $r = 0.2$, (12) becomes $\sigma_i \leq 3.6\hat{\sigma}$, and hence our optimization holds if all macroblocks have a standard deviation smaller than 3.6 times the average $\hat{\sigma}$, which is usually the case for most macroblocks in a frame [e.g., observe the σ_i distributions in Fig. 2(d)].

The values of $Q_1^*, \cdots, Q_N^*$ in (10) need to be encoded and sent to the decoder. For example, in H.263 and MPEG4 [5], [15], they are differentially encoded in a raster-scan order, and there is, on average, a 5-bit penalty for changing the quantizer value. Specifically, 2 bits are used in the syntax element $DQUANT$ to indicate the change of value for the quantization parameter QP ($DQUANT$ and QP are restricted to values in $\{-2, 1, 1, 2\}$ and $\{1, 2, \cdots, 31\}$, respectively), and three additional bits are needed on average in the syntax element $MCBPC$ to indicate that the value of QP in the current block is different from that in the previous macroblock. At a high bit rate, the amount of overhead rate due to changing the quantizer is negligible, but at a lower bit rate, this quantization overhead becomes increasingly significant, and the range of the Q_i's must be reduced for a better bit allocation (in an MSE sense) [30]. Unfortunately, an optimization procedure that would take the overhead into account is mathematically intractable or computationally expensive (cf. [1], [30]). We propose a heuristic method for reducing the quantization overhead at lower bit rates which consists of simply setting the values of the α_i weights as follows:

$$\alpha_i = \begin{cases} 2 \dfrac{B}{AN} (1 - \sigma_i) + \sigma_i, & \dfrac{B}{AN} < 0.5 \\ 1, & \text{otherwise} \end{cases} \tag{13}$$

where we note that $B/(AN)$ is the rate in bits/pixel for the current frame. If the rate is above 0.5, the α_i's are all equal to 1, and hence they have no effect. At lower bit rates, the α_i's will linearly approach the respective σ_i's and progressively reduce the range of the Q_i^*'s in (10). In fact, if the bit rate $B/(AN)$ is small, then $\alpha_i \approx \sigma_i$, and all of the quantization steps are approximately equal:

$$Q_1^* \approx \cdots \approx Q_N^* \approx \sqrt{\frac{AK}{(B - ANC)} \sum_{k=1}^{N} \sigma_k^2}. \tag{14}$$

B. Optimization at High Bit Rate

For some additional insight, we consider the optimization for the high-rate case where the logarithmic formula in (1) is used for modeling the average rate of the DCT bits. Then, the quantization overhead is negligible and the optimized step sizes are those that minimize the following expression:

$$\begin{aligned} Q_1^*, \cdots, Q_N^*, \lambda^* = &\underset{Q_1,\cdots,Q_N,\lambda}{\arg\min} \ \frac{1}{N} \sum_{i=1}^{N} \alpha_i^2 \frac{Q_i^2}{12} + \lambda \\ &\cdot \left[\sum_{i=1}^{N} A\left(\frac{1}{2} \log_2 \left(2e^2 \frac{\sigma_i^2}{Q_i^2} \right) + C \right) - B \right] \end{aligned} \tag{15}$$

which is equivalent to (9), but with the log formula for the DCT bit rate. The minimization in (15) has been done in previous work (cf. [8]), and when the distortion weights are equal to 1, i.e., the distortion is MSE, the optimal quantizer step sizes happen to be equal to some fixed $\tilde{Q}$ for all macroblocks, where the value of $\tilde{Q}$ is a function of the number of bits available B (and other parameters). Hence, at a high bit rate, an optimal rate control strategy should choose the same quantization step size for all of the macroblocks in a frame such that the total number of bits produced is B. Although our method focuses on low bit rate and hence uses the Q_i^*'s from (10), it is interesting to note that at a high bit rate (large B), for a given range of σ_i's, the range of values taken by our Q_i^*'s will be smaller because B is in the denominator of (10).

C. Conclusions on Optimal Quantization

A more extensive study could evaluate the optimization at some medium bit rates where macroblocks may operate in the logarithmic and quadratic ranges of (1) simultaneously. But from our previous low- and high-rate optimizations, we can conclude that, in order to minimize the MSE of a motion-compensated frame, the range of optimized quantizers should be small at high bit rates, increase with the range of macroblock variances from middle to low bit rates, and at some point, decrease again for very low bit rates to compensate for the quantization overhead. This is captured and quantified in (10) and (13), which are the basis of our rate control. Observe that with this strategy, the range of optimized quantizers is generally small, which indicates that our quadratic formula (4) will often model only a local region of the bit-quantizer curves. Finally, note that these conclusions and results apply when the objective is to minimize the typical approximation of MSE, but not when other distortion criteria are considered (cf. [25], [29]).

IV. Rate Control

In this section, we use the expressions in (10) and (13) to design a rate control scheme to operate H.263-compliant encoders with a low-delay (small) buffer. The scheme could be easily applied to other similar DCT-based coders. For example, in Section V-B, we use this technique with an MPEG4 codec. In our experiments, the first frame is intracoded (I frame) with a fixed quantizer. By default, we use $QP = 15$, which corresponds to a step size of 30. The next frames are of type P, i.e., they are predicted by the respective previous encoded frames using motion compensation. Encoding only P frames (after a first I frame) is a common strategy for keeping the end-to-end delay low in video communications, and is also used by TMN7 and VM7 rate control [5], [15].

A. Frame Skipping

Before the current frame is (inter) coded, we update the number of bits in the encoder buffer[2] W:

$$W = \max(W_{\text{prev}} + B' - R/F, 0) \tag{16}$$

where B' is the actual number of bits used for encoding the previous frame, W_{prev} is the previous number of bits in the buffer (initially, $W_{\text{prev}} = 0$ and B' is the number of bits for the I frame), and R and F are the channel and frame rate, respectively. The updated number of bits in the buffer W equals the previous buffer fullness W_{prev}, plus the bits used for the previous frame B' (these are the bits placed into the buffer for transmission), and minus the number of bits taken by the channel per frame interval R/F.

If W is larger than or equal to some maximum value M, the encoder skips encoding frames until the buffer fullness is below M. For each skipped frame, the buffer fullness is reduced by an additional R/F bits. Therefore, if the encoder needs to skip L frames, the updated buffer fullness is $W = \max(W_{\text{prev}} + B' - (L+1)R/F, 0)$. This simple frame-skipping strategy is used in TMN7 and VM7 rate control, and hence we use it here in order to perform a fair comparison, but other strategies could be used as well with our method. As in TMN7 rate control [5], we assume that the process of encoding is temporarily stopped when the physical buffer (whose size should be at least as large as M) is nearly full. This means that buffer overflow will not occur, at the cost of frame skipping.

The W bits in the encoder buffer belong to previously encoded frames and must be sent before those of the current frame, thus introducing a delay in the system denoted as buffer delay. Other sources of delay such as the encoding time can also be important, although the latter is implementation dependent and hence hard to quantify. Nevertheless, we focus on buffer delay since that is generally a major delay component when transmitting real-time video. With our frame skipping, the number of bits in the buffer W will not be larger than M (at the beginning of the current frame's interval), and hence the buffer delay will be less than M/R seconds. By default, we set $M = R/F$ (as in [5]), and hence the maximum buffer delay is only $M/R = 1/F$ s, i.e., the time of a frame interval. In practice, the threshold M could be increased or decreased if we wanted a larger or smaller (maximum) buffer delay, respectively.

B. Frame-Layer Rate Control

The frame-layer selects a target number of bits for the current frame to be encoded. We use the following simple formula for the frame target:

$$B = \frac{R}{F} - \Delta \tag{17}$$

where Δ is defined as follows:

$$\Delta = \begin{cases} \dfrac{W}{F}, & W > Z \cdot M \\ W - Z \cdot M, & \text{otherwise} \end{cases} \tag{18}$$

in which by default we set $Z = 0.1$. Δ is a small value that provides feedback from the buffer fullness W. If W is larger than 10% of the maximum M, the frame target B is slightly decreased. Otherwise, B is slightly increased. Since $B' \approx B$, the buffer fullness (16) is $W \approx \max(W_{\text{prev}} - \Delta, 0)$ and, as a result, the Δ correction will help maintain a small number of bits in the buffer (and hence a low buffer delay) without underflowing the buffer. In any case, the value of Δ is usually small, and hence our method selects a frame target B that is nearly constant throughout the video sequence. Since the complexity of the video sequence may change along time, this means that the quality of the encoded video will also vary. Nevertheless, having a near-constant target B is quite common for low-delay rate control [5], [15] because, since M is small, larger fluctuations of B would produce large fluctuations of buffer fullness W that could easily result in undesirable frame skipping (when W is larger than M) or buffer underflow (when $W_{\text{prev}} + B' - R/F$ is less than zero).

[2] In the buffer simulations implemented in Telenor's H.263 [12] and MoMuSys' VM7 MPEG4 [13], the buffer fullness is updated according to $W = W_{\text{prev}} + B' - R/F$. Occasionally, this results in a negative number of bits in the buffer, which is impossible in a real case. In our tests, we modified these encoders so that the updating is done using (16).

C. Macroblock-Layer Rate Control

This layer selects the values of the quantization step sizes for all of the macroblocks in a frame, so that the sum of the macroblock bits is close to the frame target B (17). The following is a step-by-step description of this method.

Step 1: Initialization

Step 1.1: Computation of parameter S_1—Compute the variances of the macroblock prediction errors $\sigma_1^2, \cdots, \sigma_N^2$ as indicated by (5). Macroblocks in P frames are occasionally intracoded, and in that case, we use (5) with the macroblock pixel values and divide the result by 3. This is a simple heuristic that compensates for the fact that intracoded macroblocks with a given pixel variance produce fewer bits than intercoded blocks with the same prediction error variance (since the former pixel values are more correlated). Then, compute $S_1 = \Sigma_{k=1}^{N} \alpha_k \sigma_k$, where α_k is given by (13).

Step 1.2: Initialize frame counters and model parameters K and C—Let $i = 1$ and $j = 0$. Let $\beta_1 = B$ (total number of bits for the frame), $N_1 = N$ (number of macroblocks in the frame). If this is the first P frame, set $K = K_1 = 0.5$, $C = C_1 = 0$, $QP_{\text{prev}} = 15$. Otherwise, set K_1 and C_1 to the values of K and C obtained after encoding the previous frame, respectively.

Step 2: Compute Q^* for ith Macroblock—Compute $L = \beta_i - 16^2 N_i C$. If $L > 0$, compute the optimized quantizer using (10) with $A = 16^2$:

$$Q_i^* = \sqrt{\frac{16^2 K}{L} \frac{\sigma_i}{\alpha_i} S_i}.$$

Otherwise, we are running out of bits, so simply set $Q_i^* = 2(QP_{\text{prev}} + 2)$.

Step 3: Find QP and Encode Macroblock:—QP is set to $(Q_i^*/2)$ rounded to nearest integer in $\{1, \cdots, 31\}$. Let DQUANT $= QP - QP_{\text{prev}}$. If DQUANT > 2, set DQUANT $=2$. If DQUANT < -2, set DQUANT $= -2$.

Set $QP = QP_{\text{prev}} +$ DQUANT. DCT encode macroblock with quantization parameter QP, and set $QP_{\text{prev}} = QP$.

Step 4: Update Counters:—Let B_i' be the number of bits used to encode the ith macroblock. Then compute:

The number of bits left for encoding the frame: $\beta_{i+1} = \beta_i - B_i'$.

The number of macroblocks that remain to be encoded: $N_{i+1} = N_i - 1$.

The sum of the $\sigma_k \sigma_k$'s for the remaining macroblocks: $S_{i+1} = S_i - \alpha_i \sigma_i$.

Step 5: Update Model Parameters K and C

Step 5.1: Compute the model parameters for the ith macroblock:—Compute model parameters using (3) with $A = 16^2$:

$$\hat{K} = \frac{B_{LC,i}'(2QP)^2}{16^2 \sigma_i^2} \quad \text{and} \quad \hat{C} = \frac{B_i' - B_{LC,i}'}{16^2}$$

where $B_{LC,i}'$ is the number of bits spent for the luminance and chrominance of the macroblock.

Step 5.2: Find the average of the $\hat{K}$'s and $\hat{C}$'s measured so far in the frame:—Compute the average of the $\hat{C}$'s using recursive form:

$$\tilde{C}_i = \tilde{C}_{i-1}(i-1)/i + \hat{C}/i.$$

TABLE I
DESCRIPTION OF THE EXPERIMENTS IN THE H.263 CODEC: NAMES, VIDEO SEQUENCES, FRAME RATE F, AND TARGET BIT RATE R

Test Name	Video Sequence	F fps	Target R Kbps	TMN7 rc	TMN8 rc
fmn48	"foreman"	10	48	47.8	48.0
fmn64	"foreman"	10	64	64.1	64.1
fmn112	"foreman"	10	112	111.5	112.1
mad10	"m & d"	7.5	10	10.0	10.0
mad24	"m & d"	10	24	24.1	24.0
mad48	"m & d"	10	48	48.3	48.0
news48	"news"	10	48	44.7	48.0
sil24	"silent"	10	24	23.8	24.0
sil48	"silent"	10	48	47.7	48.0
sil112	"silent"	10	112	109.9	112.1

All sequences lasted 10 s and were QCIF ("m & d" is an abbreviation for "mother and daughter"). The two rightmost columns compare the bit rates achieved by TMN7 rate control (TMN7 rc) and the new method (TMN8 rc).

If $(\hat{K} > 0$ and $\hat{K} \leq 4.5)$, do the following:

set $j = j + 1$,

compute average: $\tilde{K}_j = \tilde{K}_{j-1}(j-1)/j + \hat{K}/j$.

Otherwise, do not use that value of $\hat{K}$ in the estimation since K cannot be zero and should not be much larger than the Laplace, uncorrelated case $K = e/\ln 2 = 3.92$ (1)—the value 4.5 was empirically determined.

Step 5.3: Find the weighted average of the initial estimates K_1 and C_1 with $\tilde{K}_j$ and $\tilde{C}_i$:—When only a few macroblocks in the current frame have been encoded (i.e., "i" is small), $\tilde{K}_j$ and $\tilde{C}_i$ are the average of a few values, and hence are not robust estimates of K and C for the current frame. Since K_1 and C_1 are reliable estimates from the previous frame, a reasonable approach to estimate K and C is to use estimates close to K_1 and C_1 when "i" is small and estimates close to $\tilde{K}_j$ and $\tilde{C}_i$ when "i" approaches N. This can be done using the following linear or weighted average:

$$K = \tilde{K}_j(i/N) + K_1(N-i)/N$$
$$C = \tilde{C}_i(i/N) + C_1(N-i)/N.$$

In practice, the values of our estimates for K and C remain fairly constant throughout a video sequence.

Step 6: Loop Condition—If $i < N$, let $i = i + 1$, and go to Step 2. Otherwise, stop (all macroblocks are coded). If there are more frames to code, set $W_{\text{prev}} = W$, $B' = \Sigma_{i=1}^{N} B_i' = B - \beta_{N+1}$, and go back to Section IV-A.

D. Reducing Computational Complexity

The computational complexity of our method is reasonable for real-time encoding because only a few operations per macroblock are needed and each macroblock is coded only once. Nevertheless, our technique can be easily modified for reducing computation. For example, if the motion estimation algorithm uses either the mean-squared error (MSE) or the mean absolute difference (MAD) criteria, we can use the macroblocks' MSE's or the squared MAD's, respectively, as the values of the macroblock variances $\{\sigma_i^2\}$ for S_1 in Step 1.1. We tested these two simpler versions of our method, and found

TABLE II
COMPARISON OF THE NUMBER OF FRAMES SKIPPED AND AVERAGE PSNR FOR TMN7 RATE CONTROL (TMN7 rc) AND THE NEW METHOD (TMN8 rc) IN THE H.263 CODEC

Test Name	Total Frames	TMN7 rc # Frames Skipped	TMN8 rc # Frames Skipped	TMN7 rc PSNR dB	TMN8 rc PSNR dB	Gain in PSNR dB
fmn48	97	5	0	30.79	31.13	+ 0.34
fmn64	98	5	0	32.11	32.48	+ 0.37
fmn112	99	5	0	34.57	35.10	+ 0.53
mad10	67	6	0	32.24	32.28	+ 0.04
mad24	96	8	0	34.83	35.02	+ 0.19
mad48	98	8	0	37.47	37.83	+ 0.36
news48	96	0	0	33.12	34.14	+ 1.02
sil24	94	5	0	30.84	30.89	+ 0.05
sil48	97	4	0	33.88	34.10	+ 0.22
sil112	99	5	0	38.38	39.30	+ 0.92

The total number of P frames available for each case is shown in the second leftmost column. When frames were skipped, the respective previous encoded frames were used in the PSNR computation. The rightmost column indicates the gain achieved by the new method in luminance PSNR.

that the overall performance did not change much with respect to the original (which uses (5) for the σ_i^2's). On the other hand, if motion estimation can only be done in advance for some of the macroblocks (e.g., in one-pass hardware codecs), S_1 could be estimated using its value in the previous frame and the σ_i^2's available.

E. The Rate Control in TMN8 and VM8

The test model TMN and verification model VM are documents that describe examples of encoders compliant with the H.263 and MPEG4 syntax, respectively, and are usually updated at each of the respective standard meetings. Our rate control method was adopted in the test model TMN8 of H.263+ [6], and is implemented in the two publicly available H.263+ encoders from Telenor/UBC and Deutsche Telekom.[3] Hence, from now on, we refer to our method as TMN8 rate control. A method similar to TMN8 rate control was also adopted as "mode 1" of the macroblock rate control in the verification model VM8 of MPEG4 [28]. But that method is not discussed here to avoid confusion and because it lacks some recent improvements such as the adaptation of σ_i with (13).

V. EXPERIMENTAL RESULTS

A. TMN8 Rate Control in an H.263 Encoder

We used Telenor's H.263 codec (version 2.0) [12] with the advanced prediction and unrestricted motion vector modes, and tested two rate control schemes: 1) our new TMN8 rate control, and 2) the well-known rate control in TMN7[4] [5]. In both cases, the first frame was intracoded (I frame) with $QP = 15$, several frames were skipped after the first to decrease the number of bits in the buffer below $M = R/F$, and the remaining frames were all intercoded (P frames).

In TMN7 rate control, the target number of bits per frame is the bit rate divided by the frame rate, i.e., R/F, which is like ours (17), but with $\Delta = 0$. In a given frame, the quantization parameter QP increases with: 1) the difference between the bits occupied by the previous frame and the frame target R/F, and 2) the difference between the number of bits spent so far in the frame and $(k/N)R/F$, where k is the current macroblock and N is the total number of macroblocks. This method is attractive for one-pass hardware implementations because it does not require a preanalysis of the frame.

In Table I, we describe the video sequences, target bit rates, frame rates, and names assigned to each of the tests. The sequences are all well known, and we encoded 10 s of each (in $4:1:1$, YUV format). Table I also shows the actual bit rates achieved by the two rate control strategies. Observe that our TMN8 rate control achieves a bit rate closer to the target than TMN7's. In Table II, we show the total number of P frames and the number skipped by the two rate control methods. TMN7 rate control skipped frames in most of the tests (up to 9% of the total), while the new method did not skip any. As a result, TMN8 rate control encodes the video sequences with better motion continuity, which is particularly important to avoid lip-sync problems and for specific applications such as sign-language communications.

Fig. 3(a)–(f) shows plots of the number of bits in the buffer W at each frame when using TMN8 rate control (solid line) and TMN7's (dashed line). The test names in theses and other figures are indicated on the top of the respective plots. The straight, dotted line indicates the value of M, the threshold used for frame skipping. Since we set $M = R/F$, if W is close to the dotted line, the buffer delay W/R is close to $1/F$ seconds, i.e., one frame interval. If W is close to 0, the buffer delay is near 0 s. Also, recall that if there are more than M bits in the buffer, both rate control schemes skip frames until the buffer fullness is below M. For example, in Fig. 3(a), the dashed line reaches M (dotted line) five times, which indicates that five frames are skipped in the encoder when TMN7 rate

[3]UBC's H.263+ codec is available by anonymous ftp to monet.ee.ubc.ca/pub/tmn or from http://www.ece.ubc.camg. Deutsche Telekom's codec is available by ftp to ftp.berkom.ded/h263plus (login: fz141, password: video).

[4]TMN7 rate control was also in the test models TMN5 and TMN6 of H.263, and was kept in TMN8 as an example of an effective, lower complexity rate control method.

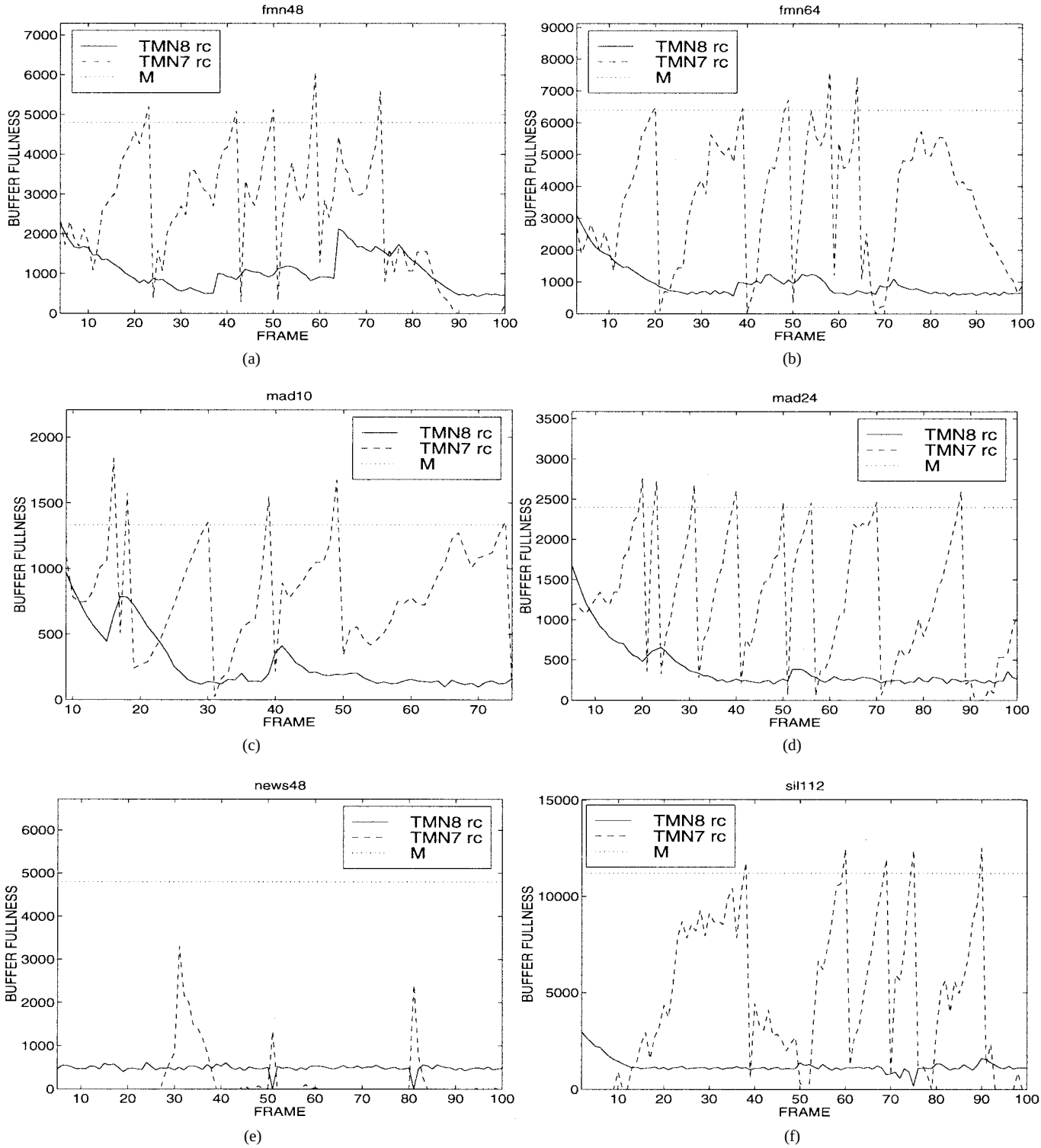
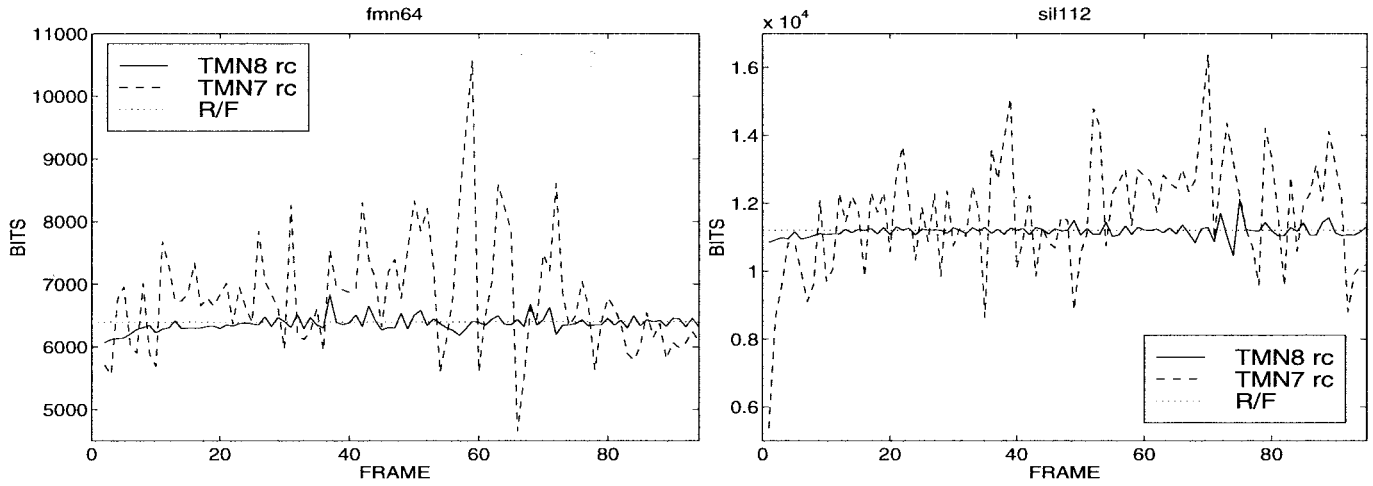


Fig. 3. (a)–(f) Comparison of the number of bits in the encoder buffer when our TMN8 rate control (solid line) and TMN7's (dashed line) are used in the H.263 codec. The straight, dotted line indicates the value of M, the threshold used for frame skipping. If there are more than M bits in the buffer, both rate control schemes skip frames until the buffer fullness is below M. For example, in Fig. 3(a), TMN7 rate control skips five frames.

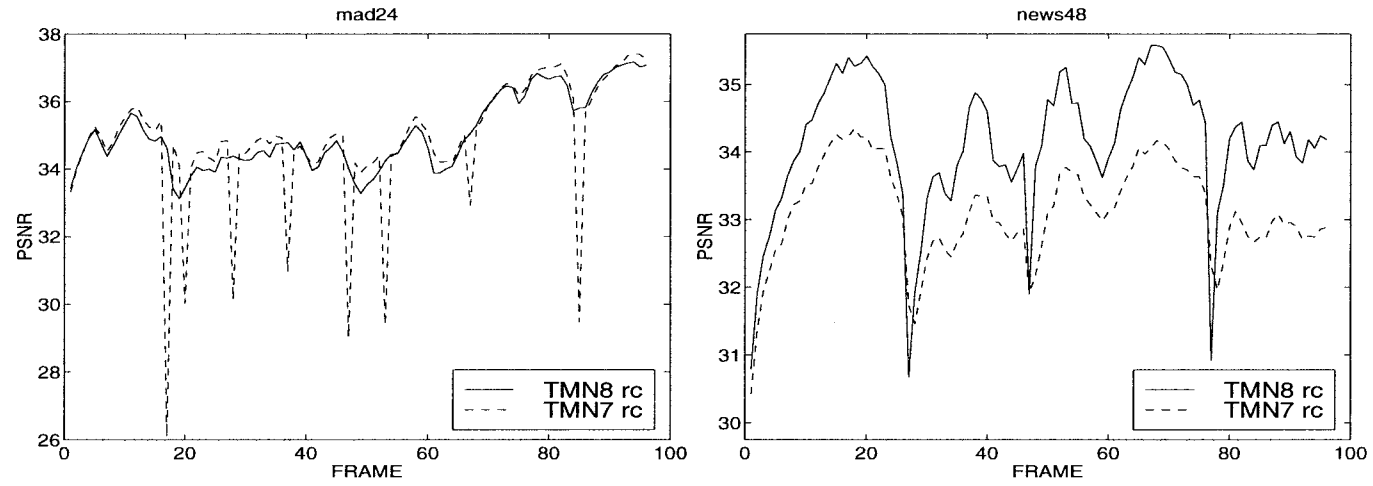
control is used. The buffer delay produced by our rate control method is always very low since there are typically few bits in the buffer.

Occasionally, there are fewer bits in the buffer than those accepted by the channel, and the buffer is emptied. This phenomenon is known as buffer underflow, and occurs when the encoder produces bits at a lower rate than the channel. For example, if a channel accepts 1000 bits per frame interval and the encoder only produces 800 bits per frame, the buffer will soon be empty and 200 bits will be lost (or used as bit stuffing) at each frame interval. In Fig. 3(e), observe that the buffer is empty for many frames with TMN7 rate control. Not surprisingly, TMN7 rate control used only 44.7 kbits/s (recall Table I), which is about 10% below the channel rate.



(a) (b)

Fig. 4. (a), (b) Comparison of the number of bits/frame used by TMN7 rate control (dashed line) and our TMN8 rate control (solid line) for two of the tests in the H.263 codec. The dotted line indicates the target average bits/frame R/F (where R is the channel or target bit rate and F is the frame rate).



(a) (b)

Fig. 5. (a), (b) Comparison of the PSNR/frame for TMN7 rate control (dashed line) and our TMN8 rate control (solid line) for two of the tests in the H.263 codec. In (a), TMN7's PSNR drops sharply eight times due to frame skipping since, when a frame is skipped, the PSNR is computed with respect to the previous encoded frame. In (b), no frames are skipped by either of the two methods (the PSNR drops are caused by sudden background changes in the sequence "news").

Fig. 4(a) and (b) shows the number of bits per frame B' used by both rate control methods for two of the tests. These plots are related to the buffer occupancy plots of Fig. 3(b) and (f), respectively, as given by (16). Observe that our new method often achieves a nearly constant number of bits/frame, while TMN7's experiences fairly high bit rate fluctuations throughout the video sequence. The latter are what causes frame skipping and buffer underflow under the tight requirements of a low-delay buffer.

A fair comparison of PSNR between rate control methods that skip different frames is not trivial. For example, a rate control technique that skips many frames would typically use more bits per coded frame, and could easily have a very high average PSNR per coded frame. But using that PSNR value as a measure of compression quality would be unfair because the distortion of the nonencoded video data needs to be taken into account somehow. In the rate control tests at MPEG4, it was decided that when a frame was skipped, the previous encoded frame should be used in the PSNR computation because the decoder displays the previous encoded frame in place of the skipped one. We computed the average PSNR using this approach, and show the results in Table II. However, these PSNR values should be interpreted as a measure of compression quality with the procedure above in mind. In Fig. 5(a) and (b), we show the PSNR per frame for TMN7 and TMN8 rate control for two of the tests. In Fig. 5(a), TMN7's PSNR drops sharply eight times due to frame skipping. TMN8 rate control does not skip any frame and still maintains a PSNR similar to TMN7's. In Fig. 5(b), there is no frame skipping by either method, and TMN8 rate control has typically higher PSNR than TMN7. The latter is in part because our method performs a more careful bit allocation, and also because TMN7 rate control wastes some channel bits due to buffer underflow (as discussed above). Overall, observe that the PSNR's in the table and plots are either similar or better with TMN8 rate control. We also confirmed that the subjective image quality was either similar or better.

Fig. 6. (a1), (b1), (c1) show frames 19, 20, 21, respectively, of test "mad48" coded with TMN7 rate control. (a2), (b2), (c2) are the same frames, but coded with TMN8 rate control. Frames (a1) and (b1) are identical because TMN7 rate control skipped one frame after encoding (a1) since TMN7's buffer occupancy W was above M, and hence (a1) was simply repeated at the decoder side. TMN8 rate control did not skip that frame, and hence provides better motion continuity.

In general, the averaged chrominance PSNR[5] was close in most cases. At higher bit rates, the chrominance PSNR of TMN8 rate control was in general a little higher than TMN7's, and at lower bit rates, it was a little lower. No significant subjective chrominance differences were observed here or in the tests of Section V-B.

Finally, we illustrate the effect of frame skipping by showing several successive coded frames in Fig. 6. These frames are of "mad48," and are encoded with TMN7 (top) and TMN8 (bottom) rate control. Fig. 6(a1) and (b1) are identical because TMN7 rate control skipped one frame after encoding (a1) and, since no new frame was available, the frame (a1) is simply repeated at the decoder side. TMN8 rate control did not suffer from frame skipping and, as a result, the frame in (b2) conveys new information and improves the motion continuity.

B. TMN8 Rate Control in an MPEG4 Codec

We used the MoMuSys MPEG4 codec (version 7.0) [13] with H.263-type quantization and tested: 1) the new TMN8 rate control, and 2) the rate control in VM7[6] [14], [15]. We used $QP = 16$ for the first I frame, intercoded the rest as P frames using rate control, and set the frame skipping for both methods to $M = 0.1R$ (i.e., the maximum buffer delay was 100 ms). In TMN8 rate control, we set $Z = 0.2$ (18) since, experimentally, we found that in MPEG4 this value worked a little better than the default.

[5]Detailed comparisons of chrominance PSNR using TMN8 rate control, TMN7's, and VM7's are reported in [11] and [24].

[6]This rate control was also in the verification models VM5 and VM6 of MPEG4 video, and remained in VM8 and later models as the basic rate control for applications with larger buffers (and hence longer delays) than the ones considered here.

TABLE III
DESCRIPTION OF THE EXPERIMENTS IN THE MPEG4 CODEC: NAMES, VIDEO DATA SOURCES, FRAME RATE F, AND TARGET BIT RATE R

Test Name	Video Sequence	F fps	Target R Kbps	VM7 rc	TMN8 rc
HALL10	"hall"	7.5	10	9.4	10.0
MAD10	"m & d"	7.5	10	9.2	10.0
MAD24	"m & d"	10	24	22.9	24.0
NEWS112	"news"	10	112	107.5	112.1
NEWS48_C	"news", CIF	7.5	48	44.8	48.0
SIL24	"silent"	10	24	23.4	24.0

All the video sequences lasted 10 s and were of QCIF Resolution, except for NEWS48_C, which was CIF. and comparison of the bit rates achieved by VM7'3 rate control (VM7 rc) and the new method (TNN8 rc).

VM7 rate control [15] is based on Chiang and Zhang's recent paper [14]. The target number of bits/frame is initially set to a weighted average of the number of bits used in the previous frame and R/F, and then the target is slightly modified to prevent buffer underflow and overflow. Next, the average MAD of the frame's motion-compensated prediction error is computed, and using a model that relates MAD, QP^2, QP, and bit rate, the method estimates the appropriate value of QP (same for all macroblocks). Frame skipping occurs when the number of bits in the simulated buffer is above 80% of a desired buffer size. We selected a size of $0.125R$, and hence frame skipping also occurred at $M = 0.1R$.

In VM7 rate control, the number of bits for the first I frame is not input in the encoder buffer, and the buffer fullness is set to 50%. This eliminates the initial frame skipping (to lower the buffer fullness below M), but it is not realistic. In any case, it does not affect the comparison of performance between the two rate control techniques, and hence we initialized TMN8 rate control using the same procedure so that the two methods

TABLE IV
COMPARISON OF THE NUMBER OF FRAMES SKIPPED AND AVERAGE PSNR FOR VM7 AND TMN8 RATE CONTROL IN THE MPEG4 CODEC

Test Name	Total Frames	VM7 rc # Frames Skipped	TMN8 rc # Frames Skipped	VM7 rc PSNR dB	TMN8 rc PSNR dB	Gain in PSNR dB
HALL10	74	4	1	29.24	29.49	+0.25
MAD10	74	17	1	31.77	31.55	-0.22
MAD24	99	2	0	34.69	34.72	+0.03
NEWS112	99	1	0	38.77	39.71	+0.94
NEWS48_C	74	1	1	30.35	30.72	+0.37
SIL24	99	4	0	30.48	30.41	-0.07

The total number of P frames available for each case is shown in the second leftmost column. When frames were skipped, the respective previous encoded frames were used in the PSNR computation. The rightmost column indicates the gain achieved by the new method in luminance PSNR.

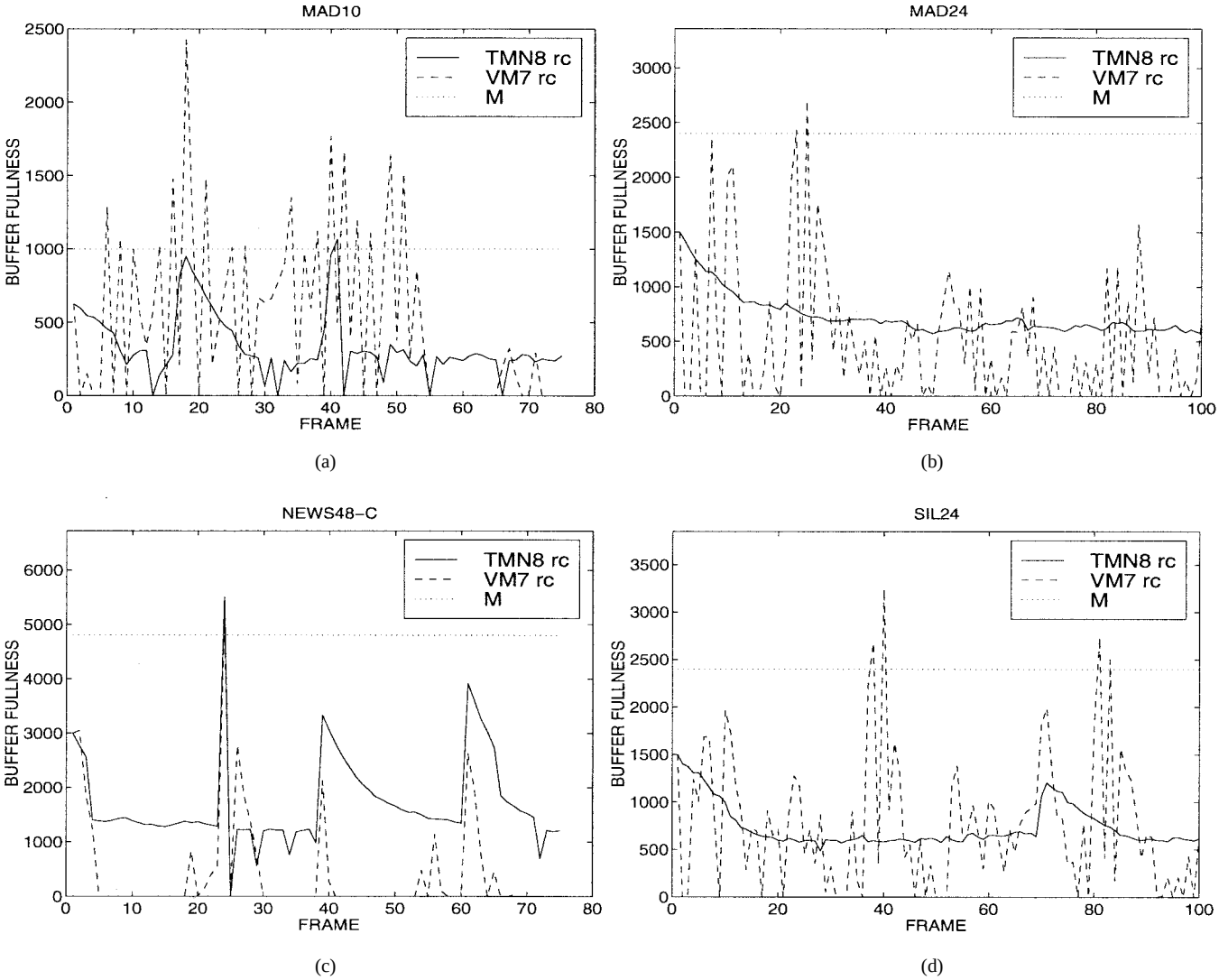
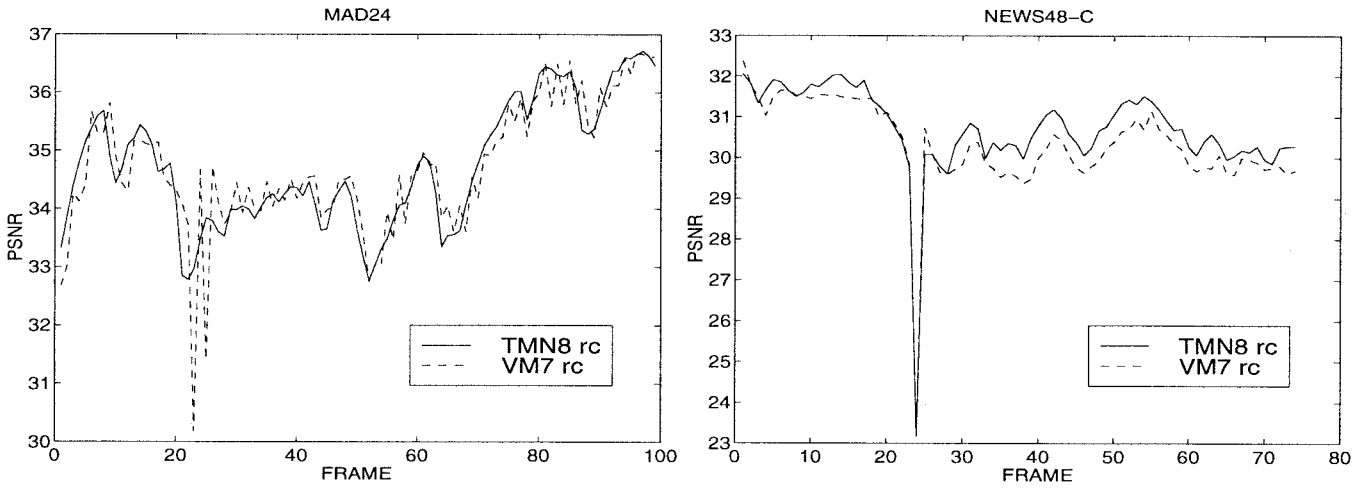


Fig. 7. (a)–(d) Comparison of the number of bits in the encoder buffer when our TMN8 rate control (solid line) and VM7's (dashed line) are used in the MPEG4 codec. The straight, dotted line indicates the value of M, the threshold used for frame skipping. If there are more than M bits in the buffer, both rate control schemes skip frames until the buffer fullness is below M. For example, in (a), VM7 rate control skips 17 frames, while TMN8's skips one frame.

would start encoding the first P frame with the same initial conditions.

Tables III and IV are like Tables I and II, respectively, but for the comparison of TMN8 and VM7 rate control in the MPEG4 codec. Observe that, in general, our TMN8 rate control achieves a bit rate closer to the target, skips fewer frames, and has either similar or higher average PSNR. Fig. 7(a)–(d) shows plots of the number of bits in the buffer W for some of the tests, when using TMN8 rate control and VM7's. Again, our new method achieves a better buffer



(a) (b)

Fig. 8. (a), (b) Comparison of the PSNR/frame for VM7 rate control (dashed line) and our TMN8 rate control (solid line) for two of the tests in the MPEG4 codec. In (a), VM7's PSNR drops sharply twice due to frame skipping since the PSNR is computed with respect to the previous encoded frame. In (b), both methods skipped frame 24.

regulation. Finally, Fig. 8(a) and (b) compares the PSNR on a frame-by-frame basis for two of the tests. In Fig. 8(a), TMN8 rate control maintains a similar PSNR without skipping any frames (VM7 skipped two frames), and in Fig. 8(b), our PSNR is generally higher (both methods skipped one frame).

TMN8 rate control has also been tested using other target bit rates, video frame resolutions, and frame rates, and in other video sequences such as the well-known "container ship" and "coastguard." Results on the tests discussed in Sections V-A and V-B plus those on other experiments were presented in recent ITU-T meetings, and can be found in [11] and [26].[7] The subset of results presented in this paper is representative of where differences in performance were found among the different rate control methods.

VI. Conclusions

We presented a new rate control method for operating typical DCT video coders for low-delay video communications. Our method, TMN8 rate control, is based on an analysis of the rate and distortion in this type of coders and on a Lagrangian optimization that minimizes distortion for a given bit budget constraint. This analysis provides formulas that add some new insights to quantizer control theory. For example, they indicate that, in order to minimize the MSE of a motion-compensated frame, the range of quantization step sizes should be small at high bit rates, increase with the range of macroblock variances from medium to low bit rates, and decrease again at very low bit rates. Our method can be used for real-time encoding because only a few operations are required per macroblock and each macroblock is encoded only once. We implemented our TMN8 rate control in H.263 and MPEG4 codecs, and compared its performance to the rate control in TMN7 and that in VM7 using low-delay (small) buffers. In comparison to TMN7 and VM7 rate control, our method met the target bit rate more accurately, skipped fewer frames, encoded the sequences with similar or higher visual quality and PSNR, and kept a lower, steadier buffer occupancy and delay while rarely suffering from buffer underflow. In particular, since TMN8 rate control skipped fewer frames, the video sequences are encoded with better motion continuity, which is important to avoid lip-sync problems and for specific applications such as sign-language communications.

[7]This and other related documents of the ITU Video Experts Group, Study Group 16, Question 15, are available by anonymous ftp to standard.pictel.com/video-site.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable suggestions and their colleagues in ITU-T and MPEG for many valuable discussions. They are also grateful to G. Côté and P. List for implementing TMN8 rate control in the publicly available encoders of H.263+ and providing comments regarding computational complexity. They also thank the members of the Digital Video Department at Sharp Labs, in particular, S. Daly, J. Li, and K. Matthews for many valuable comments and discussions throughout the development of this work.

References

[1] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Processing*, vol. 3, pp. 533–545, Sept. 1994.
[2] 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. 12–19, Feb. 1996.
[3] L.-J. Lin, A. Ortega, and C.-C. J. Kuo, "Rate control using spline-interpolated R-D characteristics," in *Proc. VCIP*, Orlando, FL, Mar. 1996, pp. 111–122.
[4] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, pp. 47–58, Apr. 1991.
[5] ITU-T/SG15, Video codec test model, TMN7, Nice, Feb. 1997.
[6] ITU-T/SG15, Video codec test model, TMN8, Portland, June 1997.
[7] E. D. Frimout, J. Biemond, and R. L. Lagendik, "Forward rate control for MPEG recording," in *Proc. SPIE Visual Commun. Image Processing*, Cambridge, MA, Nov. 93, pp. 184–194.
[8] A. Nicoulin, M. Mattavelli, W. Li, A. Basso, A. Popat, and M. Kunt, "Image sequence coding using motion-compensated subband decomposition," in *Motion Analysis and Image Sequence Processing*, M.

I. Sezan and R. L. Lagendijk, Eds. Norwell, MA: Kluwer Academic, 1993, pp. 225–256.
[9] J. Ribas-Corbera and D. Neuhoff, "On the optimal motion vector accuracy for block-based motion-compensated video coders," in *Proc. IS&T/SPIE Dig. Video Comp.: Alg. & Tech.*, San Jose, CA, Jan.–Feb. 1996, pp. 302–314.
[10] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer Academic, 1992.
[11] J. Ribas-Corbera and S. Lei, "Rate control for low delay video communications," ITU-T/SG16, Q15-A-20, Portland, June 1997.
[12] Telenor codec, "ITU-T/SG-15, video codec test model, TMN5," Telenor Research, Jan. 1995.
[13] MoMuSys codec, "MPEG4 verification model version 7.0," ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG 97, Bristol, U.K., Mar. 1997.
[14] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 246–250, Feb. 1997.
[15] Video Group, "Text of ISO/IEC 14496-2 MPEG4 video VM—Version 7.0," *ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG 97/W1642*, Bristol, U.K., Mar. 1997.
[16] F. Bellifemine, A. Capellino, A. Chimienti, R. Picco, and R. Ponti, "Statistical analysis of the 2D-DCT coefficients of differential signal for images," *Signal Processing: Image Commun.*, vol. 4, pp. 477–488, 1992.
[17] F. Moscheni, F. Dufaux, and H. Nicolas, "Entropy criterion for optimal bit allocation between motion and prediction error information," in *Proc. VCIP*, Nov. 1993, pp. 604–612.
[18] J. Ribas-Corbera and D. L. Neuhoff, "Optimizing block size in motion-compensated video coding," *J. Electron. Imaging*, vol. 7, pp. 155–165, Jan. 1998.
[19] T. M. Cover and J. M. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[20] 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, pp. 287–298, Apr. 1997.
[21] D. A. Pierre, *Optimization Theory with Applications*. New York: Dover, 1986.
[22] B. Tao, H. A. Peterson, and B. W. Dickinson, "A rate-quantization model for MPEG encoders," *in Proc. ICIP*, Santa Barbara, CA, vol. I, Oct. 1997, pp. 338-341.
[23] A. Y. K. Yan and M. L. Liou, "Adaptive predictive rate control algorithm for MPEG videos by rate quantization method," in *Proc. Picture Coding Symp.*, Berlin, Germany, Sept. 1997, pp. 619–624.
[24] J. Ribas-Corbera and S. Lei, "Comparison of TMN8 and VM7 rate control for low-delay video communications," ITU-T/SG16, Q15-B-28, Sunriver, Sept. 1997.
[25] A. Puri and R. Aravind, "Motion-compensated video with adaptive perceptual quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, pp. 351–378, Dec. 1991.
[26] J.-J. Chen and H.-M. Hang, "Source model for transform video coder and its application—Part II: Variable frame rate coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 299–311, Apr. 1997.
[27] K. H. Yang, A. Jacquin, and N. S. Jayant, "A normalized rate-distortion model for H.263-compatible codecs and its application to quantizer selection," in *Proc. ICIP*, Santa Barbara, CA, vol. II, Oct. 1997, pp. 41–44.
[28] Video Group, "Text of ISO/IEC 14496-2 MPEG4 video VM—Version 8.0," *ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG 97/W1796*, Stockholm, Sweden, July 1997.
[29] S. Daly, K. Matthews, and J. Ribas-Corbera, "Face-based visually-optimized image sequence coding," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Chicago, IL, Oct. 1998, vol. III, pp. 443–447.
[30] A. Ortega and K. Ramchandran, "Forward-adaptive quantization with optimal overhead cost for image and video coding with applications to MPEG video coders," in *Proc. IS&T/SPIE Dig. Video Comp.: Alg. & Tech.*, San Jose, CA, Feb. 1995, pp. 129–138.

Jordi Ribas-Corbera (S'91–M'92) received the Enginyer Tècnic de Telecomunicacions degree from the Escola d' Enginyeria La Salle, Barcelona, Spain, in 1990, the M.S. degree in electrical engineering from the University of California, Irvine, in 1992, and the Ph.D. degree in electrical engineering (Systems) from the University of Michigan, Ann Arbor, in 1996.

He joined the digital video department at Sharp Labs of America, Camas, WA, in 1996 where he is currently a Member of Technical Staff in the video coding and communications group. In summer 1994, he worked in the advanced video processing laboratory at the NTT Human Interface Labs, Yokosuka, Japan. His research interests are in image processing and information theory, especially data compression, video processing and analysis, and computer vision.

Dr. Ribas-Corbera received the Young Investigator Award in the 1997 IS&T/SPIE International Conference on Visual Communications and Image Processing. He currently represents Sharp in the ISO/MPEG and ITU-T/LBC standardization efforts. He is a member of SPIE.

Shawmin Lei (S'87–M'88–SM'95) received the B.S. and M.S. degrees from the National Taiwan University, Taipei, R.O.C., in 1980 and 1982 and the Ph.D. degree from the University of California, Los Angeles, in 1988, all in electrical engineering.

From 1982 to 1984, he was an Instructor of Electrical Engineering at Naval Academy, Taiwan. From August 1988 to October 1995, he was with Bellcore, Red Bank, NJ, where he had worked in both video compression and wireless communication areas. He is currently with Sharp Laboratories of America, Camas, WA, where he is a Manager of the Video Coding and Communication Group. His current research interests include video/image compression, coding, processing and communication, multimedia communication, wireless communications, data compression, and error control coding. He is the author or coauthor of more than 40 technical papers and has been awarded three patents.

Dr. Lei received the Best Paper Award (corecipient) of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 1993.