Bit-Rate Control Using Piecewise Approximated Rate–Distortion Characteristics

Liang-Jin Lin, *Member, IEEE*, and Antonio Ortega, *Member, IEEE*

***Abstract*—Digital video's increased popularity has been driven to a large extent by a flurry of recently proposed international standards (MPEG-1, MPEG-2, H.263, etc.). In most standards, the rate control scheme, which plays an important role in improving and stabilizing the decoding and playback quality, is not defined, and thus different strategies can be implemented in each encoder design. Several rate–distortion (R–D)-based techniques have been proposed to aim at the best possible quality for a given channel rate and buffer size. These approaches are complex because they require the R–D characteristics of the input data to be measured before making quantization assignment decisions. In this paper, we show how the complexity of computing the R–D data can be reduced without significantly reducing the performance of the optimization procedure. We propose two methods which provide successive reductions in complexity by: 1) using models to interpolate the rate and distortion characteristics, and 2) using past frames instead of current ones to determine the models. Our first method is applicable to situations (e.g., broadcast video) where a long encoding delay is possible, while our second approach is more useful for computation-constrained interactive video applications. The first method can also be used to benchmark other approaches. Both methods can achieve over 1 dB peak signal-to-noise rate (PSNR) gain over simple methods like the MPEG Test Model 5 (TM5) rate control, with even greater gains during scene change transitions. In addition, both methods make few *a priori* assumptions and provide robustness in their performance over a range of video sources and encoding rates. In terms of complexity, our first algorithm roughly doubles the encoding time as compared to simpler techniques (such as TM5). However, complexity is greatly reduced as compared to methods which exactly measure the R–D data. Our second algorithm has a complexity marginally higher than TM5 and a PSNR performance slightly lower than that of the first approach.**

***Index Terms*— MPEG video, piecewise approximations, rate control, rate–distortion optimization.**

I. Introduction

DIGITAL techniques for recording and transmitting video signals have become popular in the last few years, as several video compression standards, such as MPEG-1 and MPEG-2 [1], [2], have been finalized and adopted, for example, in the digital video disk (DVD) and several digital broadcast TV standards. Most of these applications use constant-bit-rate (CBR) channels to deliver compressed, variable-bit-rate (VBR) bit streams, and thus the compressed data have to be stored in memory buffers at the encoder and decoder to smooth out the bit-rate variations. A bit-rate control algorithm at the encoder is necessary to ensure that the buffers at the encoder and decoder do not underflow or overflow. Even in cases where large buffers are available, the constant end-to-end delay constraints may be the dominating factor, and bit-rate control will still be required.[1] In the rest of the paper, we assume that the constraints (whether memory or delay dominated) are given in the form of a maximum encoder buffer size. In the CBR transmission case, it can be shown that, when using the same buffer sizes at the encoder and decoder, preventing encoder underflow/overflow guarantees that no *decoder* underflow/overflow will occur [3]. Thus, for CBR, it is sufficient to control the output of the encoder to avoid overflow (underflow can be avoided through bit stuffing). In the VBR transmission case, similar analyses can be made, and the result is analogous, i.e., appropriate bit-rate control at the encoder can guarantee that end-to-end delay constraints are not violated. We concentrate here on the CBR transmission case, but the techniques we develop would also be applicable, with simple modifications, to the case where VBR transmission is used and the encoder can select both source rate and channel rate (see [4] for an example of such joint optimization). Note that even in stored video applications (e.g., DVD or CD-ROM) where data are encoded off line, the bit rate still needs to be controlled correctly to prevent the decoder buffer from overflowing or underflowing during real-time playback. Because the rate control itself is not specified by the standard, and affects only the encoder, any standard compliant decoder can decode the bit stream regardless of the rate control technique used. This makes it even more important to design efficient high-performance rate control algorithms.

In addition to avoiding overflow, it is important to design bit-rate control algorithms which provide good video quality, by not only maximizing the quality of each picture frame, but also avoiding excessive variations in video quality. Quality requirements are often overlooked in designing rate

Manuscript received November 5, 1996; revised October 13, 1997 and February 10, 1998. This work was supported in part by the National Science Foundation under Grant MIP-9502227 (CAREER) and by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, with additional support from the Annenberg Center for Communication at the University of Southern California and the California Trade and Commerce Agency. This paper was recommended by Associate Editor J. Woods

L.-J. Lin is with Microtek Lab Inc., Redondo Beach, CA 90278-1226 USA.

A. Ortega is with the Integrated Media Systems Center, Department of Electrical Engineering—Systems, University of Southern California, Los Angeles, CA 90089-2564 USA.

Publisher Item Identifier S 1051-8215(98)05768-1.

[1] Let ΔN be the end-to-end delay in the system, i.e., ΔN is the time (measured in frame intervals) a particular frame remains in the system, from the time it is encoded to the time it gets decoded. In systems where no frames are dropped, ΔN is constant and, given R, the channel rate in bits per frame, it can be shown that the encoder can store at most $\Delta N \cdot R$ bits in its buffer in order to guarantee that decoder buffer underflow will not occur, *even if a larger buffer memory is available* [3], [4].

control algorithms so that only a fraction of the numerous proposed algorithms explicitly consider distortion as a factor. Thus, the focus of our paper will be the design of effective rate–distortion (R–D)-based bit-rate control algorithms. We consider both high- and low-encoding delay scenarios, and provide algorithms that are suitable for each case.

Rate–distortion techniques aim at meeting the requirement of overflow prevention while maximizing the video quality. Methods based on Lagrangian optimization [5]–[9] or dynamic programming [10] have been considered in the literature. These methods typically perform a *preanalysis* of future video frames to measure their R–D characteristics before applying a rate allocation strategy. If frame dependencies are taken into account [7], the complexity can become very high, as increasing numbers of R–D operating points have to be measured, thus making some of these methods only suitable for off-line encoding. A popular approach to reduce the complexity has been to rely on rate and distortion models, which avoid the need to measure the R–D data on all possible quantization settings [11]. Traditionally, models which allow the computation of closed-form solutions have been preferred, but in this paper, we will argue that sufficiently accurate modeling for practical coders may not be possible with closed-form solutions. Instead, in Section III, we introduce models based on sampling the R–D data and interpolating (using spline approximations) those points that have not been measured. Our models also take into account the dependencies arising in motion-compensated video coding, and we demonstrate that they can be used to significantly speed up the search procedures in R–D-based bit-rate control, with negligible penalty in video quality. In our results, we achieve average gains of close to 1-dB peak signal-to-noise rate (PSNR), with gains of over 2 dB possible for particular scenes, as compared with a simple one-pass encoder. The complexity increases by only a factor of two with respect to the one-pass encoder, but is an order of magnitude lower than that of a similar scheme which measures, rather than models, the R–D data.

The results in Section III show that preanalysis can be effectively used in combination with modeling of R–D characteristics. This preanalysis approach can be used in applications, such as stored video and broadcast video, where the encoding delay can be large, allowing the encoder to store several video frames. In general, significant encoding delay may be possible, and indeed desirable, for applications where two-way interactivity is not required.[2] For example, in the stored video case, because encoding is done once but the sequence is decoded many times, there is a clear need for algorithms that can significantly improve the video quality, even at the cost of additional compute power requirements at the encoder. It is important to note that the proposed models permit a faster implementation of general R–D based algorithms with little loss in performance. Thus our models, as will be seen in Section IV, permit benchmarking of other, faster, approaches, with a reasonably low computation cost.

Data preanalysis can only be very limited in applications requiring a low end-to-end delay. This is the case, for example, in interactive communications, as in the videophone and videoconferencing applications for which the H.261 [12] and H.263 [13] standards are designed. In low-encoding delay and low-complexity scenarios, predictive control schemes, e.g., [14] or the MPEG Test Model 5 (TM5) [15], have often been considered to be good solutions. In predictive schemes, rate allocation decisions are based on currently available information such as the buffer state or the expected rate for future blocks (which is estimated based on the rate used for previous frames). Examples include direct buffer-state feedback methods where the buffer occupancy determines the quantization setting [14]. These methods suffer in performance if the assumptions, which may be based on a particular type of sequence or scene, do not hold. Moreover, most predictive methods suffer from degradation at scene changes, since models change from scene to scene and the rate control is set to parameters based on a model that is no longer valid.

While predictive schemes are attractive due to their low computational complexity and low delay, we will propose that preanalysis and R–D criteria can be used to improve the performance, even for interactive applications. Simple preanalysis based on a single frame has been used to improve TM5 by measuring the frame and block activities from the current frame, rather than using estimates based on the previous frame. Another TM5-based method proposed in [16] uses a constant q to quantize and encode all the blocks in each frame in order to get a bit-usage profile, which is then used during the actual encoding. Another work [17] proposes measuring the entropy and using it to predict the bit rates at the macroblock level. Other model-based methods can be found in [18] and [19], but again, most of the proposed schemes are based on rate only.

Thus, our goal is *to design fast, low encoding delay, R–D-based bit-rate control techniques*. The algorithm we introduce in Section IV is based on preanalysis of a single frame, i.e., we measure the R–D characteristics of the current frame type (I, P, or B), and relies on previously encoded frames of other types to estimate the R–D characteristics of future frames. For each frame, we use the piecewise spline approximations presented in Section III. This algorithm maintains both low encoding delay and low encoding complexity, and may be of interest not just for interactive applications, but also for noninteractive applications (e.g., live video) where encoder complexity is limited. In comparison with TM5, our method can produce more stable quality, and is robust (i.e., our results are consistent for a wide variety of video sequences and channel rates), with average gains close to 1 dB in some instances, and even larger gains in specific scenes. Moreover, our scheme allows the introduction of perceptual criteria by, for example, imposing a constraint to limit the changes in quality between consecutive frames. We use the algorithm of Section III to benchmark the performance of the fast algorithm, and show that loss in performance is very slight, while the complexity is now comparable to that of the simple TM5 algorithm.

[2] Note that we make a distinction between encoding delay and real-time encoding. A system can support real-time encoding (e.g., encode 30 frames per second) while having a significant encoding delay (e.g., frame i is captured at the time when frame $i - \Delta N_c + 1$ is being compressed, so that ΔN_c frames are stored at the encoder at any given time). This is the case for many hardware video encoders which operate using various degrees of pipelining. For example, the encoder can be computing the motion field for one frame while the DCT of another frame is being computed.

The paper is organized as follows. We start by formulating the problem of bit-rate control in a rate–distortion framework in Section II. Given that generating the R–D data is the major complexity factor, in Section III, we introduce models to interpolate R–D characteristics that allow significant reductions in complexity. Finally, we introduce a fast and efficient bit-rate control method where R–D characteristics predicted from past frames are used in combination with R–D data measured from the current frame (Section IV).

II. Problem Formulation

While we present experiments based on MPEG encoders, which we now briefly introduce, our proposed algorithms are general enough to be applied to other similar video coding schemes, such as H.261 or H.263. In MPEG, the input frame is segmented into blocks of 16 $\times$ 16 pixels, or macroblocks. Each macroblock can be an intrablock, which is DCT coded, or a nonintrablock, which is DCT coded after subtracting a block from the reference frame obtained through motion estimation. The intra/nonintra selection strategy is not defined in the standard, but it is constrained by the frame type. Three frame types are defined in MPEG: 1) I (intra) frames, where macroblocks can only be coded in intra mode, 2) P (predicted) frames, where each macroblock can be coded in intra, or nonintra mode, and 3) B (bidirectionally interpolated) frames, where each macroblock can be coded in intra mode, with forward prediction only, with backward prediction only, or as a bidirectionally interpolated block. The set of pictures including an I frame and all successive P and B frames up until the next I frame is called a group of pictures (GOP). When considering motion-compensated prediction, we will talk about *reference* frames, which are used to generate the prediction, and *predicted* frames. We refer to [1], [20], and [2] for more details.

MPEG encoders can assign one out of 31 possible quantization values (*mquant*) to each macroblock, thus controlling the rate–distortion tradeoff. The objective of the bit-rate control algorithm is to determine *mquant* for each macroblock to keep the output bit rate within the rate and buffer constraints, while maintaining high and stable quality. To simplify the problem, encoders typically operate at two levels. First, they perform a frame-level allocation by selecting a single parameter q for each frame. Then, specific *mquant* values are assigned to individual macroblocks within the frame. In TM5, these two steps are called *global control* and *adaptive quantization*, respectively. We concentrate here on the frame-level allocation, and use the GOP as our basic coding unit. Optimal (in a R–D sense) macroblock level selection of quantizers [21], [22] is possible given a rate budget for a frame. Thus, a complete allocation framework could include (and possibly iterate between): 1) a frame-level allocation of quantizers that generates frame rate budgets, and 2) a macroblock-level allocation for the given budget.

Let $\boldsymbol{q} = (q_1, q_2, \cdots, q_N)^T$ be the quantization choices for the N frames in a GOP, where $q(i)$ is the quantizer choice for the ith frame. The rate and distortion functions for frame i, denoted $r_i(\boldsymbol{q})$ and $d_i(\boldsymbol{q})$, respectively, can be found by computing the total number of bits and the mean-square error (MSE) of the ith frame for the given quantization choice $\boldsymbol{q}$. By using a vector expression for $\boldsymbol{q}$, we are taking into account the "dependency" of the problem, i.e., the R–D tradeoff for a given predicted/interpolated frame depends on the reference frame(s) used to generate the motion-compensated prediction [7]. The buffer occupancy $b(i, \boldsymbol{q})$ at the ith frame interval, when the GOP has been coded with quantization choice $\boldsymbol{q}$ is then

$$b(i, \boldsymbol{q}) = \max\left(b(i-1, \boldsymbol{q}) + r_i(\boldsymbol{q}) - \frac{R}{F}, 0\right), \quad \forall i = 1, \cdots, N \quad \text{with} \quad b(0, \boldsymbol{q}) = 0 \tag{1}$$

where R is the channel rate in bits/second, F is the frame rate in frames/second, and $b(0, \boldsymbol{q})$ is the buffer occupancy before the first frame is coded. Note that we use a $\max$ function in our formulation because in underflow situations the buffer occupancy never falls below zero (stuffing bits are used).

We now formulate the bit-rate control problem with two different distortion criteria. First, we consider the case where the objective is to minimize the average distortion over an entire GOP.

Formulation 1—Minimizing Average Distortion: Let $\mathcal{Q} = \{1, 2, \cdots, 31\}$ be the set of admissible quantizers[3] and let $b_{\max}$ be the prescribed maximum buffer size. Find $\boldsymbol{q}^* = (q_1^*, q_2^*, \cdots, q_N^*)^T$, with $q_i^* \in \mathcal{Q}$ for $i = 1, 2, \cdots, N$, where N is the GOP size, such that

$$\boldsymbol{q}^* = \arg \min_{\boldsymbol{q} \in \mathcal{Q}^N} \frac{1}{N} \sum_{i=1}^{N} d_i(\boldsymbol{q}) \tag{2}$$

subject to

$$b(i, \boldsymbol{q}) \le b_{\max}, \qquad i = 1, 2, \cdots N-1 \tag{3}$$

$$b(N, \boldsymbol{q}) = 0. \tag{4}$$

We impose the constraint of (4) to force the final buffer occupancy $b(N, \boldsymbol{q})$ to be zero (possibly after adding stuffing bits), and therefore maintain a constant number of bits per GOP. This constraint is necessary for recording on a digital tape recorder [5], and also allows faster searching and indexing for video streams stored in a CD-ROM or hard drive. This constraint also simplifies the optimization since it decouples the rate allocation for each GOP (all GOP's receive the same rate), and thus makes it possible to operate with a fixed encoding delay. If "constant rate per GOP" is not required, the constraints can be removed to better utilize the buffer and improve the quality.

Several methods have been proposed to solve this problem. In the simpler "independent" case, where there is no interframe dependency (e.g., I-frame-only MPEG or motion JPEG) and $r_i(\boldsymbol{q}), d_i(\boldsymbol{q})$ depend only on q_i, well-known approaches such as Lagrangian optimization [23] or dynamic programming [10] can be used to approach or achieve the optimal solution. In the more general dependent-coding case (e.g., MPEG with P and B frames), solutions tend to be more complex, even if the buffer constraints are ignored [7] because allocations in one reference frame affect the following predicted frames.

[3] A larger quantization index corresponds to a coarser quantizer, and thus higher distortion and lower rate.

The complexity increases in two ways. First, efficient ways of searching the N-dimensional space of quantization choices may not exist ($\mathcal{Q}^N$ different allocations are possible). Second, computing $r_i(\boldsymbol{q}), d_i(\boldsymbol{q})$ may require recoding *all* the frames in the GOP for each choice of $\boldsymbol{q}$ *even if only one* q_i *changed*, since changing the quantizer for a reference frame affects all of its predicted frames.

Gradient-based search techniques can be used to tackle the first source of complexity, as they provide a structured way of traversing the N-dimensional space of possible solutions and can efficiently find solutions that are close to the overall optimal one [24]. The number of values of $\boldsymbol{q}$ that have to be tested before converging is not only much smaller than that needed in exhaustive search, but it also becomes smaller when solutions for previous GOP's are used as initialization (because successive GOP's tend to be similar). Details of the algorithm can be found in [24] and [25]. To make this algorithm (or others based on R–D optimization) practical, we must still solve the remaining complexity bottleneck, namely, the computation of R–D points. This issue will be tackled in Section III.

In video coding, minimizing the average distortion does not always lead to an optimized perceptual quality. This motivates us to introduce an alternative formulation which seeks to minimize variations in distortion between frames, and therefore avoid "flicker problems" caused by abrupt changes in quality.

Formulation 2—Minimizing Distortion Variation: For each feasible value of $q_1 \in \mathcal{Q}$, find q^* such that

$$\boldsymbol{q}^* = \arg \min_{\boldsymbol{q} \in \mathcal{Q}^{N-1}} \sum_{i=2}^{N} |d_i(q_i) - d_{i-1}(q_{i-1}^*)| \tag{5}$$

subject to

$$b(i, \boldsymbol{q}) \le b_{\max}, \qquad i = 2, \cdots, N \tag{6}$$

where $b_{\max}$ is the prescribed maximum buffer size.

Note that the resulting solution depends on the initial condition q_1, which could be chosen to minimize the average distortion achieved by $\boldsymbol{q}^*$. Solutions to Formulation 2 will tend to be close to those achieved using a minimax or a lexicographic framework [26]. In a lexicographic approach, the goal is to minimize the maximum distortion among all frames in a GOP; then, given that the maximum distortion is minimized, minimize the second largest distortion, and so on. It can be shown that, for a continuous space of admissible quantizers, the optimal solution will be the one which gives a constant distortion [26], assuming that this solution does not violate the buffering constraints. If it exists this solution will also be optimal for Formulation 2.

To solve the above-formulated problems, it will be necessary to evaluate the code length and distortion for all possible quantization settings. In the next two sections, we introduce two approaches to approximate the R–D data. In Section III, we apply the piecewise spline approximation methods to speed up the algorithm of [24] in solving Formulation 1. In Section IV, a faster method is developed which combines the spline approximations with prediction mechanisms. We demonstrate how this approach can be used to obtain solutions for both Formulations 1 and 2.

III. Approximation of Rate–Distortion Functions

Modeling of rate and distortion characteristics has been frequently used within bit-rate control schemes [11], [17], [27], [19], [28], [29]. Gaussian, Laplacian, or generalized Gaussian distributions are typical choices, thus leading to exponential or logarithmic expressions. Most of the models only consider the rate function, and often implicitly assume that the distortion is a linear function of the quantization scale. In addition, most models do not take into account the dependencies that arise in the choice of quantizers for the reference frames and the predicted frames [7]. Even when dependencies are taken into account, as in [11], some nonlinear effects typical in video coding are ignored. For example, for typical intra/inter selection rules, prediction is turned off (and intra coding is used) if the quality of the reference frame is too low, thus eliminating the dependency if coarse quantization is used on the reference frames. Because the accuracy of the models directly affects the results of the bit-rate control, our goal is to design models which: 1) account for both rate and distortion; 2) make a minimum of *a priori* assumptions on the frame characteristics; and 3) take into account the effect of frame dependencies.

A. Intraframe Approximation Method

We consider first the problem of approximating frame-level R–D characteristics, i.e., estimating $r(q)$ and $d(q)$ for any given value of q without actually having to quantize and encode the data for all values of q.[4] To compare the accuracy of the various models, we use the MPEG-2 encoder of [31], and compute the MSE and code length for all quantization settings (from 1 to 31) and all frame types (for P and B frames, a constant quantization scale of ten is used for the corresponding reference frame). The relative error for each model is calculated by

$$\text{relative\_error} = \left| \frac{\text{estimated\_value} - \text{original\_value}}{\text{original\_value}} \right|. \tag{7}$$

1) Exponential Models: Source rate models of the form

$$r(q) = \alpha + \beta \cdot \log\left(\frac{1}{q}\right) \tag{8}$$

where α and β are two parameters which may also depend on q, have been proposed in the literature [29]. Curve fitting can be used to derive appropriate values for α and β. The results show that relative errors for these models are too large (average errors of up to 40% for typical I frames and up to 150 and 400% for P and B frames, respectively) to be useful in our rate control context.

Alternative rate models, which achieve better performance, can be defined as follows [32]:

$$r(q) = \alpha + \frac{\beta}{q^\gamma} \tag{9}$$

[4] The methods we propose can also be applied to macroblock-level R–D modeling [30], [25], and can thus be used to speed up the block-level rate allocation within a frame (e.g., for the algorithms of [23] and [22]).

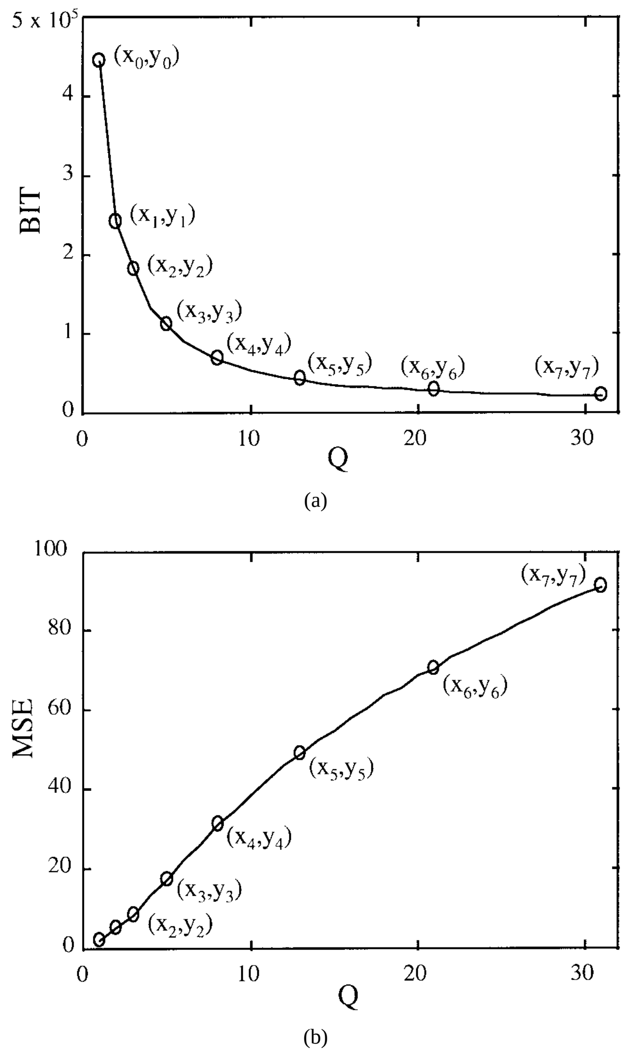


Fig. 1. Control points for typical (a) rate and (b) distortion curves. Each control point (x_i, y_i) represents the measured rate or distortion value y_i if the quantization scale is set to x_i. (a) Rate, $r(q)$ (bits). (b) Distortion (MSE), $d(q)$.

where a third parameter γ controls the curvature of the function. The average and maximum relative errors after fitting are shown in Table I (top table, column labeled as "opt.expon"). The results are better, but the error is still large, in particular for B and P frames in low-activity video sequences.

These models show relatively large errors, even if the "optimum" parameters of the model have been computed. Since model fitting is best when all of the data are used (i.e., when we measure the rate at all values of q and then fit the parameters), the complexity of this approach can still be high. Fixing the models with some *a priori* empirical values or adaptively adjusting them based on measured data is a possibility, but will result in increased relative error. Additionally, these models assume that distortion (measured by MSE) is proportional to q, which will result in large errors for $d(q)$. In what follows, we introduce interpolation functions for both $r(q)$ and $d(q)$ which allow us to trade off complexity and accuracy.

2) Interpolation Functions: In order to increase the accuracy of the models, we encode the video data and measure the R–D functions, but *only for a small set of quantization scales*, which we call "control points." We use M control points, defined as $(x_i, y_i), i = 0 \cdots M-1$, where x_i represents the quantization scale (for MPEG, $x_i \in \{1, 2, \cdots, 31\}$), and y_i represents the actual measured rate or distortion (see Fig. 1). Piecewise cubic or linear interpolation is then used to estimate the rate and distortion for the remaining q's. Let f_i be the function (either r_i or d_i) which we seek to approximate. Cubic interpolation provides an approximation which possesses first-order continuity at the control points. In this case, the values of f_i in between two consecutive control points x_i and x_{i+1} can be approximated as

$$f_i(x) = a_i \cdot x^3 + b_i \cdot x^2 + c_i \cdot x + d_i \tag{10}$$

where $i = 0 \cdots M-2$. There are $M-1$ polynomials, each corresponding to one segment. For each polynomial, the four parameters a_i, b_i, c_i, d_i can be derived from the four control points $(x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}), (x_{i+2}, y_{i+2})$ by imposing the following two constraints: 1) the interpolated function should take the same values as the original one at the control points, and 2) the first-order derivative should be continuous at the control points.

In order to capture the exponential-like decay property of the rate function, we choose our control points such that $x_i = x_{i-1} + x_{i-2}$ (e.g., for MPEG, the control points are $\{1, 2, 3, 5, 8, 13, 21, 31\}$.) For typical video sequences at the standard rate (e.g., CIF at 1.152 Mbits/s), some of the quantizers (e.g., $q = 1, \cdots, 4$) are rarely used, and so only five or six control points are required in most cases.

Our algorithms show significant reductions in error compared to the exponential model for $r(q)$, with cubic spline interpolation outperforming linear interpolation (see Table I). For $d(q)$, we include only the comparison between linear and cubic interpolation functions, which shows little difference between the two techniques. Thus, in what follows, we will use cubic interpolation for $r(q)$ and linear interpolation for $d(q)$.

B. Interframe Dependency Model

The intraframe approximations introduced in the previous section can be used to model P and B frames for a given choice of quantizer for their reference frame(s). However, additional modeling is required to fully approximate the dependency and take into account the changes in R–D as the reference frame quantizer varies. To simplify the computation, motion estimation is based on the original frames so that reevaluating the rate and distortion after a quantization change does not require recomputing the motion vectors.

1) Formulation of Interframe Dependency: Consider the first P frame in a GOP and its reference I frame (the same analysis applies if the reference is a P frame). Let q_I and q_P be the quantization choices for the I and P frames, respectively. The rate and distortion functions for the P frame will have the form $d_P(q_I, q_P)$ and $r_P(q_I, q_P)$, so that variations with both q_I *and* q_P have to be modeled. We can extend the idea of the previous section and measure $d_P(q_I, q_P)$ and $r_P(q_I, q_P)$ at selected control points in the 2-D parameter space. A straightforward approach would be to select the same M

TABLE I

RELATIVE ERRORS FOR INTRAFRAME APPROXIMATION FUNCTIONS; THE ERRORS ARE MEASURED FOR THE THREE FRAME TYPES (I, P, B), AND THREE APPROACHES ARE COMPARED: 1) BEST MATCH EXPONENTIAL MODEL (opt. expon), 2) PIECEWISE LINEAR (pw. linear), AND 3) PIECEWISE CUBIC (pw. cubic)

	Intra-frames Relative Errors for $r(q)$					
	average error			maximum error		
	opt.expon	pw.linear	pw.cubic	opt.expon	pw.linear	pw.cubic
Claire	5.77%	2.27%	0.65%	28.46%	28.89%	7.57%
Football	14.95%	4.30%	0.90%	77.09%	15.43%	6.32%
Miss America	26.75%	3.07%	1.16%	100.03%	26.05%	9.86%
Susie	21.15%	3.35%	1.24%	68.00%	23.65%	7.13%

	Intra-frames Relative Errors for $d(q)$			
	average error		maximum error	
	pw.linear	pw.cubic	pw.linear	pw.cubic
Claire	0.72%	0.55%	2.95%	4.37%
Football	0.95%	0.55%	6.13%	7.01%
Miss America	0.65%	0.43%	4.63%	3.84%
Susie	0.84%	0.51%	5.95%	6.10%

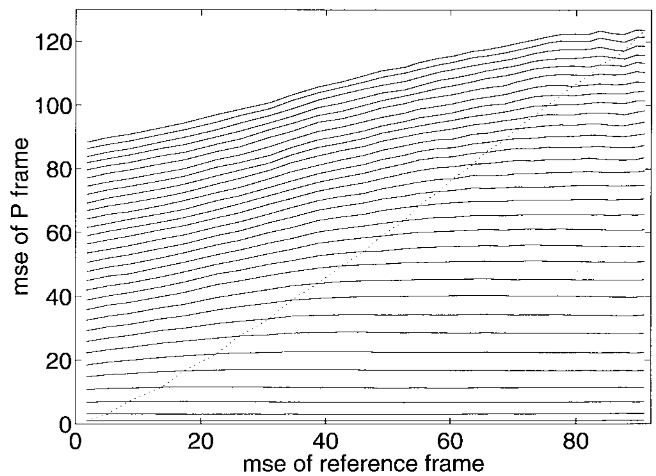
control points used in the previous section for both q_I and q_P (total of $M \times M$ control points). For each choice of q_I, the method is exactly the same as that described before. However, this approach is complex because, in order to compute the data for each additional control point along the q_I axis, both the I and P frames have to be recompressed and reconstructed. This complexity is much higher than that involved in computing the data along the q_P axis (requiring only quantization and encoding for the P frame). To cope with this problem, we introduce a model for interframe dependency which only requires two control points along the q_I axis.

Consider Fig. 2, which plots $d_P(q_I, q_P)$ as a function of $d_I(q_I)$, the MSE for the corresponding I frame, for all possible choices of q_I and q_P. These experimental results indicate that, for a fixed q_P, increasing $d_I(q_I)$ (i.e., increasing q_I) results in roughly linear increases in $d_P(q_I, q_P)$. However, d_P does not increase further beyond the point where q_I and q_P are equal. This linear-constant model can be partly justified based on typical mode selection mechanisms within MPEG (e.g., those used in [31]). Typically, the interframe mode is used on a block as long as the energy in the prediction residue is below the energy in the original block. However, as q_I increases, so does the energy in the residue, and when this energy is greater than that of the original block, the coding mode is changed to intra (at that point there is no longer a dependency with respect to q_I, and thus d_P as a function of q_I is constant). More detailed analyses to justify this model can be found in [25].

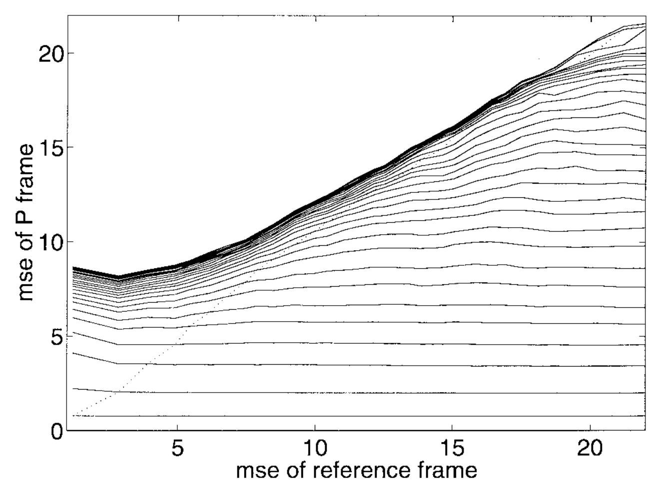
We thus propose the following I–P dependency model (refer to Fig. 3). Let $q_P = C$ be constant, then, as motivated above, we can model $d_P(q_I, q_P = C)$ as a one-dimensional, linear/constant function of the variable $d_I(q_I)$. The function is linear with respect to $d_I(q_I)$ for $q_I \leq C$, and becomes a constant function for $q_I > C$:

$$d(q_I, C) = \begin{cases} \alpha - \beta \cdot [d_I(C) - d_I(q_I)], & \text{if } q_I \leq C \\ \alpha, & \text{if } q_I > C. \end{cases} \quad (11)$$

The two model parameters α and β can be determined by encoding and measuring the distortion at two values of q_I. As shown in Fig. 3, if the two values are chosen to be 5 and 13, and the same spline model with six control points (as in Section III-A) is used along the q_P axis, the set of



(a)



(b)

Fig. 2. MSE for the P frames from two video sequences, plotted as a function of the MSE for their reference I frames. Each solid line represents the MSE for a fixed q_P as q_I changes. The dotted line indicates the boundary where the quantizers for the predicted and reference frames are equal. (a) Football sequence. (b) Miss America sequence.

12 control points becomes {(5,3), (5,5), (5,8), (5,13), (5,21), (5,31), (13,3), (13,5), (13,8), (13,13), (13,21), (13,31)}. To interpolate d_p at an arbitrary quantization setting, say (10,10),

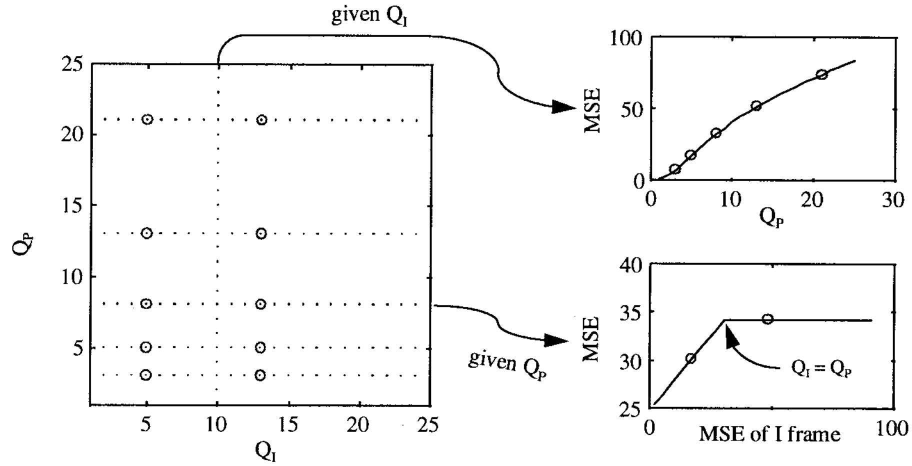


Fig. 3. Reconstruction of approximated distortion P frame. The dots indicate all of the admissible operating points in the 2-D space. The circled dots indicate those operating points at which rate and distortion are actually measured. Two different techniques are used to interpolate along the q_P (top, right) and the q_I (bottom, right) axes.

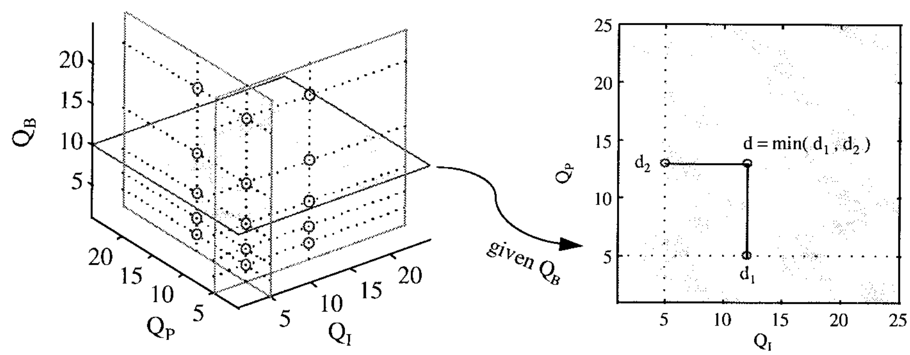


Fig. 4. Reconstruction of approximated distortion B frame. The dots indicate all of the admissible operating points in the 3-D space. The circled dots indicate those operating points at which rate and distortion are actually measured.

the interframe model is applied four times with C set to $\{$5, 8, 13, 12$\}$, so that the distortion values are estimated at (10, 5), (10, 8), (10, 13), (10, 21). Piecewise interpolation based on these four points is then used to derive the distortion at (10, 10).

For $r_P(q_I, q_P)$, we have observed for several video sequences that, for quantization scales between 3 and 24, the interframe dependency is reasonably low. Hence, the following simple piecewise linear model is used (assuming that $r_P(q_I, C)$ has been measured for q_I set to x_1 and x_2) as shown in (12), at the bottom of the page, for: 1) $q_I \leq x_1$, 2) $x_1 < q_I < x_2$, and 3) $q_I \geq x_2$, respectively.

For B frames, the distortion function can be written as $d_B(q_I, q_P, q_B)$, where q_B, q_I, q_P are the quantization scales for the B, I, and P frames involved. *A priori*, one would have to consider a three-dimensional set of parameters. To simplify, we evaluate two 2-D models as illustrated in Fig. 4. We first set $q_I = c$ (where c is one of the interframe control points), and evaluate the dependency with respect to the P frame by using the 2-D model for P frames described above to model $d_1(c, q_P, q_B)$. We then fix $q_P = c$ and apply the same model to find $d_2(q_I, c, q_B)$. Finally, $d(q_I, q_P, q_B)$ is defined as $\min(d_1(c, q_P, q_B), d_2(q_I, c, q_B))$. This procedure can be intuitively justified given the strategy for selecting "forward" or "backward" motion vectors in the MPEG encoder, where the lowest energy predictor is chosen. The same model is also used for the rate function, where there are a total of 18 control points to be measured if the same set of control points as in the example above is used.

2) Model Compliance Tests: We use the MPEG-2 encoder implementation of [31] to test the accuracy of the approximation model. As before, we encode the frames to measure the MSE and code length, for every quantization setting. Based on the function values at the predefined control points ($\{$1, 2, 3, 5, 8, 13, 21, 31$\}$ for intracoded frame, $\{$5, 13$\}$ for interframe dependency), we build the model using the procedures just

$$r_P(q_I, C) = \begin{cases} r_P(x_1, C) \\ \dfrac{r_P(x_1, C)(d_I(x_2) - d_I(q_I)) + r_P(x_2, C)(d_I(q_I) - d_I(x_1))}{d_I(x_1) - d_I(x_2)} \\ r_P(x_2, C) \end{cases} \tag{12}$$

TABLE II
RELATIVE MODELING ERRORS FOR PREDICTIVE CODING MODEL; THE RESULTS ARE GIVEN FOR QUANTIZATION SCALES IN THE RANGE FROM 3 TO 24

	Predictive Coding Model Errors							
	BITS				MSE			
	average error		maximum error		average error		maximum error	
	linear	cubic	linear	cubic	linear	cubic	linear	cubic
Claire	9.34%	2.49%	40.90%	33.02%	3.03%	0.88%	12.30%	12.30%
Football	5.38%	0.66%	14.01%	8.41%	1.81%	0.39%	5.59%	6.60%
Miss America	12.39%	3.27%	43.54%	45.82%	2.89%	0.89%	11.03%	11.03%
Susie	11.04%	2.92%	39.33%	15.88%	4.30%	1.24%	15.88%	15.88%

	Bi-directional Prediction Coding Model Errors							
	BITS				MSE			
	average error		maximum error		average error		maximum error	
	linear	cubic	linear	cubic	linear	cubic	linear	cubic
Football B1	5.08%	3.74%	22.14%	22.72%	3.08%	2.61%	17.56%	17.56%
Football B2	5.73%	4.43%	23.43%	23.64%	3.13%	2.58%	13.92%	14.73%

described, and calculate the estimated rate and distortion values. Both linear and cubic interpolation for intraframe approximation are tested. The relative error is then calculated by (7). Finally, the average and maximum relative errors are calculated over the typical operating range of quantization scales, which is from 3 to 24. The results are shown in Table II and Figs. 5 and 6, and demonstrate reasonably small errors for P frames, but somewhat larger ones for B frames. In our examples from the "football" sequence, we show the model approximations achieved for $r_I(q)$ [Fig. 5(a)], d_P as a function of d_I [Fig. 5(b)], and $d_B(q_I, q_P, C)$ for a given constant $q_B = C$ (Fig. 6).

C. Bit-Rate Control with Interpolated R–D

The proposed models are general, and can be applied to any bit-rate control scheme which requires R–D data. Indeed, the appropriate way of demonstrating their effectiveness is to show that a particular R–D based bit-rate control scheme does not lose in performance when it uses interpolated R–D data instead of measured data. Thus, we test the models introduced in Sections III-A and III-B with the gradient-based algorithm proposed in [24]. The basic idea is to replace the actual data with those obtained from the models in the rate control algorithm. Once the algorithm has converged to a solution $\boldsymbol{q}$ for the GOP, we can apply this quantizer selection to encode the GOP.

Due to the errors in the model, some of the constraints (e.g., constant rate per GOP as in [24]) may not be strictly met by the solution. However, because modeling errors are only significant for the B frames, which consume the fewest bits, the buffer constraints are normally still satisfied. To mitigate the effect of relatively large model errors in B frames, we can introduce a second pass in the algorithm. First, we use the approximations to select the quantization settings; then, after encoding the I and P frames using those settings, we calculate the total number of bits remaining for the B frames, which we denote R_B. Using this available bit budget, the bit allocation for B frames is then reoptimized. The additional optimization procedure does not cost much in terms of computation because all of the reference frames (I and P) are fixed and all of the M_B B frames are independent of each other. Denote by $r_{Bi}(q_i)$ and $d_{Bi}(q_i)$, respectively, the rate and distortion functions for the ith B frame. Our goal is to select the quantization scales for each B frame $(q_0, q_1, \cdots, q_{M_B-1})$, so as to[5]

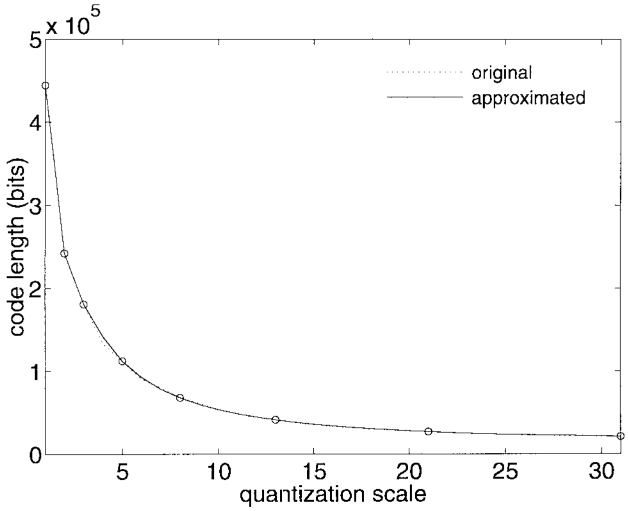
$$\text{minimize} \sum_{i=0}^{M_B-1} d_{Bi}(q_i) \text{ subject to } \sum_{i=0}^{M_B-1} r_{Bi}(q_i) \leq R_B. \tag{13}$$

This problem can be solved efficiently using Lagrangian optimization [23].

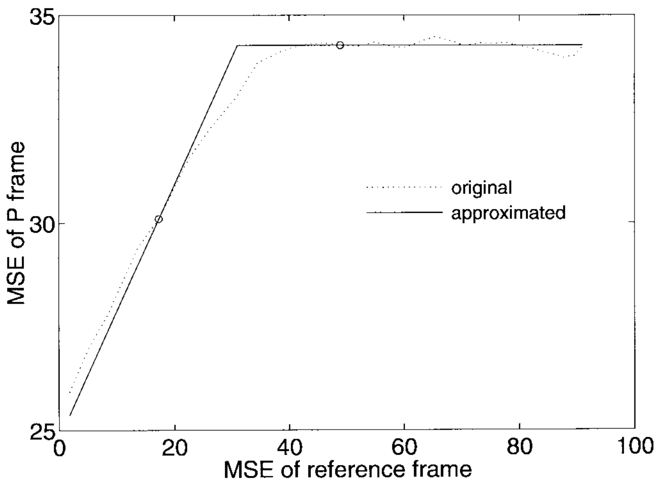
We use two standard MPEG video sequences, "football" and "table tennis," in CIF format at 1.152 Mbits/s, and compare four algorithms: 1) mdl: gradient-based method with approximated R–D characteristics; 2) mrb: method 1) with additional bit-reallocation for B frames using the Lagrangian method; 3) org: gradient-based method with the original R–D as in [24]; and 4) $tm5$: the TM5 algorithm [15] implemented in [31]. A GOP of size 6 (IBBPBB) was chosen in our experiments. The results are shown in Fig. 7 and Table III. The computation complexities provided are relative to an encoder using TM5, and are estimated based on the subroutines in [31], where: 1) 13 multiplications and 29 additions are required for each 1-D size-8 DCT operation, and 2) the full-search method is used for motion estimation. We assume that there is sufficient memory to hold all intermediate data including the motion vectors, reconstructed reference frames, DCT coefficients, etc., so that, for example, motion estimation or DCT computation only has to be done once during the evaluation of R–D data on the control points. Note that the relative increase in complexity with respect to TM5 will become larger if a fast motion estimation algorithm is used since motion estimation is responsible for most of the complexity in the encoding process (e.g., 90% of computations when using full search and TM5).

Our results show that the proposed models reduce the computation to just 15–20% of the original cost in [24], with very little loss in PSNR. If reallocation for the B frames is

[5] Note that reallocating bits for the B frames does not affect the performance for the other frames in the GOP.

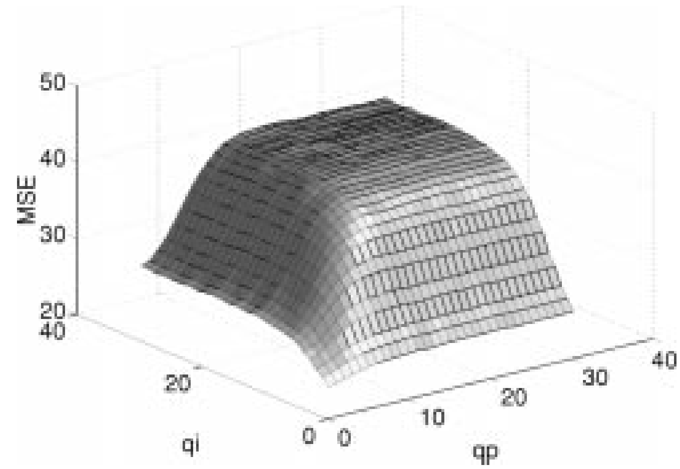


(a)

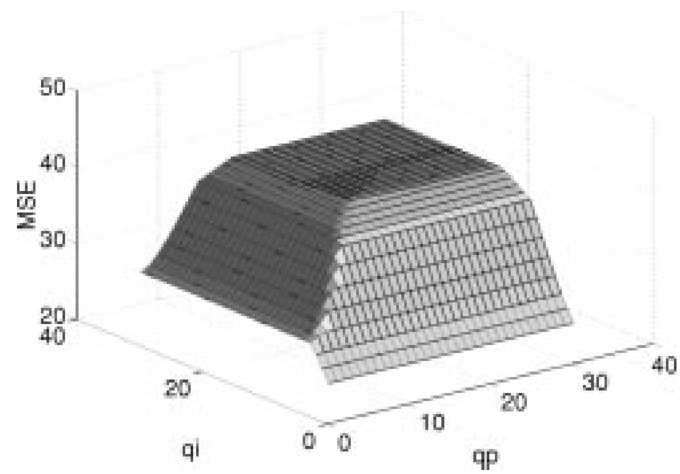


(b)

Fig. 5. (a) Rate function of an I frame in the football sequence. The circles indicate the control points, which are chosen to capture the exponential-decay property of the rate function. (b) The dotted line is the MSE of a P frame in the football sequence, with respect to the MSE of its reference frame. The quantization scale of the P frame q_P is fixed at 8. The curve is approximated by a linear-constant function, indicated by a solid line. The circles indicate the two control points, at $q_I = 5$ and $q_I = 13$. The corner point is at $q_I = q_P = 8$.



(a)



(b)

Fig. 6. (a) Original measured data and (b) reconstructed with B-frame model, of a B frame in the football sequence, as a function of q_I and q_P, with q_B fixed at 10.

used, the same PSNR as in the original method [24] can be achieved with minimal additional computation. The proposed algorithm has clear advantages over TM5 in handling scene changes, and in general in being robust enough to be used with different sequences and rates. While scene changes tend to mask compression artifacts, it can be noted that TM5 takes several frames to adjust to the new frame characteristics (see the "table tennis" sequence in Fig. 7(b), with a scene change in frames 66–67), while our algorithm "learns" much faster the characteristics of the new scene.

IV. Rate Control with Predicted R–D Characteristics

The interpolated R–D models presented in the previous section can be used to speed up R–D-based rate control algorithms with preanalysis, to ensure robust rate control with good quality, independently of specific video contents and channel rates. The computational complexity is independent of the specific rate control algorithm because the number of R–D evaluations (or control points) is fixed. Thus, this method is suitable for either off-line DVD program encoding or, with appropriate pipelining hardware, for real-time TV broadcasting. The cost paid for the increased quality and robustness is both complexity (less than a factor of two increase as compared to TM5) and encoding delay (one GOP).

For two-way interactive communication applications, a delay of one GOP is no longer admissible. Thus, we now introduce the use of predicted R–D characteristics to reduce the encoding delay to a single frame. This approach could also be useful in noninteractive applications where complexity or encoding delay are limited. This section will also serve to demonstrate the applicability of the R–D models of Section III for benchmarking: the performance of a particular fast algorithm can be compared to that attained with an R–D optimized method. While in benchmarking applications off-line computation is possible, and thus one could also use the original R–D data, our interpolated R–D method makes faster benchmarking possible, thus allowing tests to be

TABLE IV
AVERAGE PSNR AND FIRST-ORDER DIFFERENCE OF MSE FOR THE TEST SEQUENCES. GRADIENT & MODEL: GRADIENT METHOD WITH R–D APPROXIMATED BY THE MODEL WITH ADDITIONAL BIT-REALLOCATION FOR B FRAMES; PREDICTION & MINIMUM MSE: PREDICTED R–D WITH MINIMUM MSE; PREDICTION & SMOOTH MSE: PREDICTED R–D WITH SMOOTH MSE

	Bicycle				Cheer			
	GOP 6		GOP 15		GOP 6		GOP 15	
	PSNR	Diff	PSNR	Diff	PSNR	Diff	PSNR	Diff
Gradient & Model	27.05	46.03	n/a	n/a	26.59	35.47	n/a	n/a
Prediction & Minimum MSE	26.93	11.52	27.04	9.59	26.35	11.75	26.54	11.15
Prediction & Smooth MSE	26.87	9.32	27.02	7.57	26.21	6.55	26.42	5.42
Test Model 5	26.37	27.07	26.49	27.11	25.86	25.05	26.06	25.86

	Football				Flower			
	GOP 6		GOP 15		GOP 6		GOP 15	
	PSNR	Diff	PSNR	Diff	PSNR	Diff	PSNR	Diff
Gradient & Model	33.17	6.25	n/a	n/a	27.07	24.02	n/a	n/a
Prediction & Minimum MSE	33.14	5.26	33.18	4.14	26.78	15.47	27.39	14.15
Prediction & Smooth MSE	32.99	7.39	33.12	5.31	26.69	12.02	27.30	10.59
Test Model 5	32.43	11.29	32.49	11.01	25.91	14.04	26.70	15.01

	Mobile				Table Tennis			
	GOP 6		GOP 15		GOP 6		GOP 15	
	PSNR	Diff	PSNR	Diff	PSNR	Diff	PSNR	Diff
Gradient & Model	25.27	31.21	n/a	n/a	32.81	8.53	n/a	n/a
Prediction & Minimum MSE	24.72	19.59	25.54	14.88	32.65	7.80	33.48	6.93
Prediction & Smooth MSE	24.77	15.06	25.55	9.75	32.42	6.92	33.18	5.41
Test Model 5	23.96	30.02	25.07	21.44	31.25	8.62	32.13	7.58

conducted over longer sequences (while still giving a reliable approximation to the true R–D optimal solution).

To motivate our algorithm, we note that, except at scene changes, the contents of consecutive frames tend to change slowly over short periods of time (e.g., within a GOP). Thus, when encoding a GOP, it is reasonable to assume that the R–D characteristics of future, not yet coded, frames are similar to those of the most recently coded frame of the same type. In this section, we propose an algorithm which uses the intraframe model considered earlier (Section III-A), but where we now assume that R–D data are only measured for the current frame, while models based on already encoded frames are used for the remaining frames in the GOP. We further simplify the procedure by not taking into account the dependencies in the coding.

A. Control Procedures

In the new control procedure we still consider GOP's as the basic optimization unit. Because of the R–D prediction, all frames of the same type will have the same R–D data, and thus the formulation can be further simplified. As in the TM5 algorithm, we also do not explicitly consider the buffer constraints in the new formulation. The number of frames for each frame-type within a GOP is denoted as N_I, N_P, N_B. The total number of bits allocated for the GOP can be represented as

$$B = B_0 + (N_I + N_P + N_B) \cdot \frac{R}{F} \tag{14}$$

where B_0 is the number of bits left (or overused if it is negative) from the previous GOP, R is the channel rate in bits per second, and F is the frame rate in frames per second. The encoding procedure is as follows. For each frame, we first measure and approximate its R–D functions $r(q)$ and $d(q)$. To avoid any further preanalysis, the R–D data of future frames are estimated using the data from the most recently coded frame of the same type. For example, the latest P frame model is used for all future P frames remaining in the GOP. Therefore, we need to keep three sets of R–D data for the future frames, denoted as $r_I(q), d_I(q)$, $r_P(q), d_P(q)$, $r_B(q), d_B(q)$, for I, P, B frames, respectively. With these R–D data, we optimize quantization scales for all frames in the GOP, but only the quantization scale selected for the current frame is actually used in the encoding. After the current frame is encoded, we count the actual number of bits consumed by the frame, subtract it from B in (14), and then remove the current frame from the GOP (so the number of frames is decreased by one). The procedure, using the updated values of B and changing the GOP structure, is repeated one frame at a time until all of the frames in the GOP have been encoded. The two different optimization criteria shown in Section II are used.

1) Minimizing Average Distortion: We first consider Formulation 1. Since we have ignored the effect of dependencies to build our predicted R–D models, we take advantage of the monotonicity property. This property [7], confirmed by our MPEG coding experiments, indicates that a better quality in the reference frame (I and P) will lead to a better total coding efficiency. Hence, it is reasonable to restrict the admissible operating points to the range $q_I \leq q_P \leq q_B$. The total (estimated) MSE is

$$D(q_I, q_P, q_B) = N_I \cdot d_I(q_I) + N_P \cdot d_P(q_P) + N_B \cdot d_B(q_B) \tag{15}$$

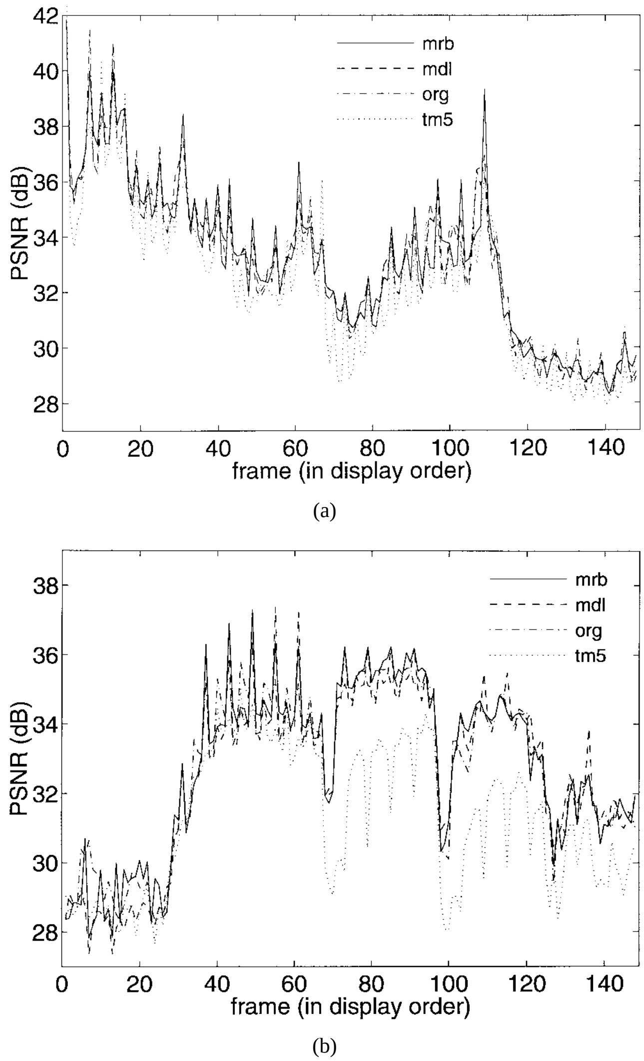


Fig. 7. PSNR for each frame in the "Football" and "Table tennis" sequences. mrb: gradient-based method using the approximated R-D by the proposed model, with additional bit-re-allocation for B frames; mdl: gradient-based method using the approximated R-D only; org: gradient-based method using the original measured R-D; $tm5$: Test model 5 algorithm. (a) Football. (b) Table tennis.

TABLE III
AVERAGE PSNR AND COMPUTATION COMPLEXITY WITH DIFFERENT ENCODING METHODS. THE COMPLEXITY IS RELATIVE TO THE TEST MODEL 5 ALGORITHM

	Football		Table Tennis	
	PSNR	Complexity	PSNR	Complexity
mdl	33.12	1.68	32.23	1.71
mrd	33.17	1.70	32.81	1.73
org [24]	33.17	8.87	32.74	11.35
tm5	32.43	1.00	31.25	1.00

and the goal of our optimization becomes to minimize $D(q_I, q_P, q_B)$ subject to

$$N_I \cdot r_I(q_I) + N_P \cdot r_P(q_P) + N_B \cdot r_B(q_B) \leq B \tag{16}$$

$$q_I \leq q_P \leq q_B \tag{17}$$

where B is the total number of bits available for a GOP. Because there are only three independent variables and there is no interframe dependency involved, it can be efficiently solved by the Lagrange multiplier method proposed in [23].

Algorithm 1—Minimum MSE:

Step 1) Initialize N_I, N_P, N_B, and set the total bit budget for a GOP using (14).

Step 2) Read the current frame, and compute its DCT transform (after motion-compensated prediction if it is a P or B frame). Let X be the current frame type (X is I, P or B).

Step 3) Evaluate and approximate $r(q)$ and $d(q)$ for the current frame, using the intraframe approximation method of Section III-A. Use the results to update $r_X(q)$ and $d_X(q)$.

Step 4) Minimize the total MSE in (15) subject to the constraints in (16) and (17). The solution is denoted as (q_I^*, q_P^*, q_B^*).

Step 5) Use q_X^* to encode the current frame.

Step 6) Calculate the actual number of bits consumed by the current frame, and subtract it from B. Decrease the counter corresponding to the current frame type N_X by one.

Step 7) If the current frame is the last frame of GOP, assign B to B_0, advance to next GOP, and go to Step 1). Otherwise, advance to the next frame and go to Step 2).

2) Minimizing Distortion Variation: The optimization criterion in Formulation 2 aims at minimizing the difference in MSE between consecutive frames, and often leads to a more stable playback quality. When using predicted R–D characteristics, the formulation can be further simplified. We use a two-step optimization process, where we first minimize the MSE difference. Based on the current frame type, we pick one variable in $\{q_I, q_P, q_B\}$ as a primary variable. For example, suppose the current frame is an I frame, the primary variable is q_I. Given $q_I = x$, the quantization scales for P frames and B frames (denoted as $y^*(x)$ and $z^*(x)$, respectively) are derived by minimizing the MSE difference

$$y^*(x) = \arg \min_y [d_P(y) - d_I(x)] \tag{18}$$

$$z^*(x) = \arg \min_z [d_B(z) - d_I(x)]. \tag{19}$$

As in the minimum MSE case, we also add a constraint

$$d_I \leq d_P \leq d_B \tag{20}$$

to force the quality of the reference to be better than that of predictive frame, which in general gives better performance due to the monotonicity property. Then, in the second step, the solution for the I frame, denoted as q_I^*, is derived by minimizing the difference between the total bits (generated by the model) and the total bit budget B

$$|[N_I \cdot r_I(x) + N_P \cdot r_P(y^*(x)) + N_B \cdot r_B(z^*(x))] - B| \tag{21}$$

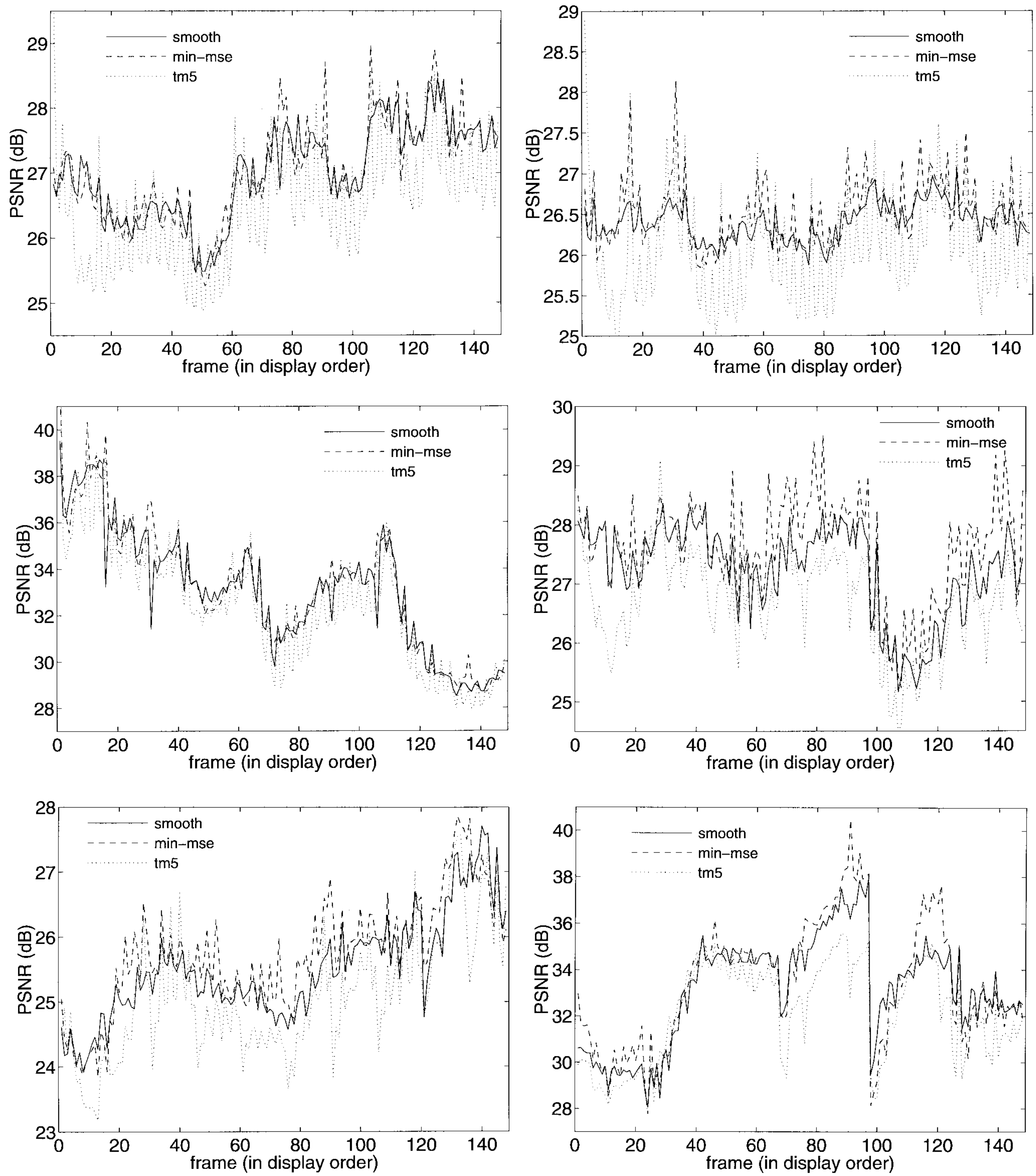


Fig. 8. PSNR of image frames of the six video sequences, encoded using GOP size 15. In each figure, smooth: optimizing by smooth MSE criterion; min-mse: optimizing by minimum MSE criterion; tm5: Test Model 5 algorithm.

over all possible x's (or values for q_I). If the current frame type is P or B, the solution of q_P^* or q_B^* can be derived by a similar procedure.

Algorithm 2—Smooth MSE:

Step 1) Initialize the value of N_I, N_P, N_B and the total bit budget of a GOP using (14).

Step 2) Read the current frame, and compute its DCT transform (after motion-compensated prediction if it is a P or B frame). Let X be the current frame type (X is I, P or B).

Step 3) Follow the same procedure as in Algorithm 1, Step 3) to derive and update R–D data.

Step 4) Derive the solution q_X^* by using the above double-loop optimization procedure.

Step 5) Follow the same procedure as in Algorithm 1, Steps 5) and 6).

Step 6) If the current frame is the last frame of GOP, assign B to B_0, advance to next GOP, and go to Step 1). Otherwise, advance to next frame, and go to Step 2).

B. Experimental Results

We encode the six MPEG test video sequences using the algorithm. The results are shown in Table IV. The PSNR of encoded image frames for GOP size 15 is shown in Fig. 8. Results for the algorithm of Section III, the gradient-search procedure with approximated R–D plus reoptimization on the B-frames, are also shown in the table for comparison. Note that we use linear interpolation for distortion approximation and cubic interpolation for rate. If both the rate and distortion models use linear interpolation, the PSNR's will be 0.05–0.2 dB lower on average. Also note that the computational complexity of the new algorithm is similar to that of TM5, with only eight additional quantization and encoding operations per frame. Compared to other operations like motion estimation or DCT, the additional overhead is not significant, with further speedups being achievable by using a parallel hardware implementation. Note that our results are also very close to those achieved in Section III, thus indicating that the potential for even further gains using R–D techniques is limited. Even if the differences in PSNR compared to a simple algorithm like TM5 are small on average, our algorithm has the advantage of being robust (it works well for different rates and video sequences and at scene changes), and also being amenable to the introduction of perceptually based criteria as part of the optimization process [33].

V. Conclusion

In this paper, we have followed the framework of deterministic rate–distortion optimization techniques with preanalysis, and have formulated the bit-rate control problem as a constrained optimization problem. We proposed an approximation model which reduces the computational complexity of R–D based methods to a practical level without degrading the quality. We also introduced a fast R–D-based algorithm suitable for low-delay encoding, and have shown promising results in the simulations. Additional work to incorporate subjective quality measures into the cost functions is currently underway [33].

Acknowledgment

The authors wish to thank the anonymous reviewers for their comments and suggestions which helped to improve the presentation and organization of this paper. They also acknowledge the generous support of Prof. C.-C. J. Kuo at the University of Southern California.

References

[1] D. LeGall, "MPEG: A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, pp. 46–58, Apr. 1991.
[2] *ISO/IEC 13818 (MPEG-2): Generic Coding of Moving Pictures and Associated Audio Information*, Nov. 1994.
[3] A. R. Reibman and B. G. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 361–372, Dec. 1992.
[4] 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. Select. Areas Commun.*, vol. 15, pp. 1016–1028, Aug. 1997.
[5] S.-W. Wu and A. Gersho, "Rate-constrained optimal block-adaptive coding for digital tape recording of HDTV," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, pp. 100–112, Mar. 1991.
[6] J. Choi and D. Park, "A stable feedback control of the buffer state using the controlled langrange multiplier method," *IEEE Trans. Image Processing*, vol. 3, pp. 546–558, Sept. 1994.
[7] 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.
[8] D. W. Lin, M.-H. Wang, and J.-J. Chen, "Optimal delayed-coding of video sequences subject to a buffer-size constraint," in *Proc. SPIE Visual Commun. and Image Processing '93*, Cambridge, MA, Nov. 1993, pp. 223–234.
[9] J. Lee and B. W. Dickinson, "Joint optimization of frame type selection and bit allocation for MPEG video encoders," in *Proc. ICIP'94*, Austin, TX, 1994, vol. II, pp. 962–966.
[10] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Trans. Image Processing*, vol. 3, pp. 26–40, Jan. 1994.
[11] K. M. Uz, J. M. Shapiro, and M. Czigler, "Optimal bit allocation in the presence of quantizer feedback," in *Proc. ICASSP'93*, Minneapolis, MN, vol. V, pp. 385–388. Apr. 1993.
[12] *ITU-T Recommendation H.261: Video Codec for Audiovisual Services at* $p \times 64$ *kbits*, Mar. 1993.
[13] *DRAFT ITU-T Recommendation H.263: Video Coding for Low Bitrate Communication*, July 1995.
[14] C.-T. Chen and A. Wong, "A self-governing rate buffer control strategy for pseudoconstant bit rate video coding," *IEEE Trans. Image Processing*, vol. 2, pp. 50–59, Jan. 1993.
[15] MPEG-2, *Test Model 5 (TM5) Doc. ISO/IEC JTC1/SC29/WG11/93-225b*, Test Model Editing Committee, Apr. 1993.
[16] G. Keesman, I. Shah, and R. Klein-Gunnewiek, "Bit-rate control for MPEG encoders," *Signal Process.: Image Commun.*, vol. 6, pp. 545–560, Feb. 1995.
[17] E. D. Frimout, J. Biemond, and R. L. Lagendijk, "Forward rate control for MPEG recording," in *Proc. SPIE Visual Commun. Image Processing '93*, Cambridge, MA, Nov. 1993, pp. 184–194.
[18] J. Zdepsky, D. Raychaudhuri, and K. Joseph, "Statistically based buffer control policies for constant rate transmission of compressed digital video," *IEEE Trans. Commmun.*, vol. 39, no. 6, pp. 947–957, June 1991.
[19] W. Ding and B. Liu, "Rate-quantization modeling for rate control of MPEG video coding and recording," in *Proc. IS&T/SPIE Digital Video Compression '95*, San Jose, CA, Feb. 1995, pp. 139–150.
[20] J. Mitchell, W. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*. New York: Chapman and Hall, 1997.
[21] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 6, pp. 182–190, Apr. 1996.
[22] 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 Digital Video Compression '95*, San Jose, CA, Feb. 1995, pp. 129–138.
[23] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1445–1453, Sept. 1988.
[24] L.-J. Lin, A. Ortega, and C.-C. J. Kuo, "A gradient-based rate control algorithm with applications to MPEG video," in *Proc. ICIP'95*, Washington, DC, vol. III, pp. 392–395, 1995.
[25] L.-J. Lin, "Video bit-rate control with spline approximated rate-distortion characteristics," Ph.D. dissertation, Univ. Southern California, May 1997.
[26] D. T. Hoang, "Fast and efficient algorithms for text and video compression," Ph.D. dissertation, Brown Univ., May 1997.
[27] J.-J. Chen and H. M. Hang, "A transform video coder source model and its application," in *Proc. ICIP'94*, Austin, TX, 1994, vol. II, pp. 962–966.
[28] J. Katto and M. Ohta, "Mathematical analysis of MPEG compression capability and its application to rate control," in *Proc. ICIP'95*, Washington, DC, 1995, vol. II, pp. 555–559.
[29] H.-M. Hang and J.-J. Chen, "Source model for transform video coder and its application," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 287–311, Apr. 1997.

[30] L.-J. Lin, A. Ortega, and C.-C. J. Kuo, "Rate control using spline-interpolated R-D characteristics," in *Proc. SPIE Visual Commun. Image Processing '96*, Orlando, FL, Mar. 1996, pp. 111–122.
[31] "MPEG-2 encoder v. 1.1a, MPEG Software Simulation Group" [Online]. Available WWW: http://www.mpeg.org/tristan/MPEG/MSSG
[32] W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 12–20, Feb. 1996.
[33] L.-J. Lin and A. Ortega, "Perceptually based video rate control using pre-filtering and predicted rate-distortion characteristics," in *ICIP'97*, Santa Barbara, CA, Sept. 1997.

Liang-Jin Lin (S'95–M'97) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Hsinchu, Taiwan, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, in 1997.

In 1989, he joined Ulead Systems, and developed the PhotoStyler image processing software. From 1993 to 1996, he was a Research Assistant at the Signal and Image Processing Institute, University of Southern California. He is currently a Researcher at Microtek Lab, Inc. His current research interests include image processing, video coding, rate control, 3-D graphics, and geometric computations.

Antonio Ortega (S'91–M'95) was born in Madrid, Spain, in 1965. He received the Telecommunications Engineering degree from the Universidad Politecnica de Madrid (UPM), Madrid, Spain, in 1989, and the Ph.D. degree in electrical engineering from Columbia University, New York, NY, in 1994.

He was a Research Assistant in the Image Processing Group at UPM in 1990. At Columbia, he was a Graduate Research Assistant at the Center for Telecommunications Research from 1991 to 1994, and was supported by a scholarship from the Fulbright Commission and the Ministry of Education of Spain. Since September 1994, he has been an Assistant Professor in the Electrical Engineering—Systems Department at the University of Southern California (USC). At USC, he is also a member of the Integrated Media Systems Center, an NSF Engineering Research Center, and of the Signal and Image Processing Institute. His research interests are in the areas of image and video compression and communications.

In 1995, Dr. Ortega received the NSF Faculty Early Career Development (CAREER) award. In 1997, he was awarded the USC School of Engineering Junior Research Award. He received the Leonard G. Abraham IEEE Communications Society Award (along with C.-Y. Hsu and A. R. Reibman). He is an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and a member of the IEEE Signal Processing Society Technical Committee on Multimedia Signal Processing. He is Technical Cochair of the 1998 Workshop on Multimedia Signal Processing.