# MPEG-4 Rate Control for Multiple Video Objects

Anthony Vetro, *Member, IEEE*, Huifang Sun, *Senior Member, IEEE*, and Yao Wang, *Member, IEEE*

***Abstract*—This paper describes an algorithm which can achieve a constant bit rate when coding multiple video objects. The implementation is a nontrivial extension of the MPEG-4 rate control algorithm for single video objects which employs a quadratic rate-quantizer model. The algorithm is organized into two stages: a pre- and a postencoding stage. In the preencoding stage, an initial target estimate is made for each object. Based on the buffer fullness, the total target is adjusted and then distributed proportional to the relative size, motion, and variance of each object. Based on the new individual targets and rate-quantizer relation for texture, appropriate quantization parameters are calculated. After each object is encoded, the model parameters for each object are updated, and if necessary, frames are skipped to ensure that the buffer does not overflow. A preframeskip control is exercised to avoid buffer overflow when the motion and shape information occupies a significant portion of the bit budget. The rate control algorithm switches between two operation modes so that the coder can reduce the spatial coding accuracy for an improved temporal resolution. A shape-coding control mechanism is also proposed, which provides a tradeoff between texture and shape coding accuracy. Overall, the algorithm is able to successfully achieve the target bit rate, effectively code arbitrarily shaped objects, and maintain a stable buffer level. These techniques have been adopted by the MPEG committee in July 1997 as part of the video Verification Model (VM8).**

***Index Terms*— Bit allocation, buffering policy, multiple video objects, rate control, shape coding control.**

## I. Introduction

OVER the years, rate control has been an extensively studied topic for video transmission. From the classic works on bit allocation [1]–[4], it is clear that rate control has emerged as a technology which is application specific. As a result of the strong relation between the bit allocation problem and rate control, excellent results have been obtained with regard to encoder optimization. Rate-distortion theory has been successfully applied to optimize the selection of wavelet packet bases [5], the selection of quantizers in a dependent coding framework [6], the frame type selection for MPEG encoding [7], and modes of prediction in an MPEG [8] or H.263 system [9]. In yet other works [10]–[14], the bit allocation problem has been jointly treated with the buffer control problem, thus realizing constraints set forth by the network.

The focus of this contribution is very different from the works cited above in that it does not attempt to optimize encoder performance, but defines a framework for the encoder to operate. Once a stable and robust framework for the rate control has been established, then the optimization can be performed. As an introduction, we will describe the general rate control problem, and show how different rate control algorithms have evolved by new demands posed by the encoding and transmission environment.

A common feature among conventional video coding schemes (e.g., MPEG, H.263) is that bit streams are generated through compression algorithms which output variable-length codes. In general, the use of variable-length codes realizes significant gains in compression, however, the bit stream is not directly suited for transmission over a fixed-rate channel. To make this transmission as efficient and accurate as possible, a variety of coding factors should be jointly considered: channel rate, encoding rate, and scene content. For the rate control algorithm to work well, the relationship between the coding factors and coding parameters must be determined or accurately modeled.

Rate control techniques have been studied very intensively for various standards and applications, such as videoconferencing with H.261 and H.263 [9], [15], storage media with MPEG-1 and MPEG-2 [5], [7], [8], [16], real-time transmission with MPEG-1 and MPEG-2 [12], [14], and the recent video object coding with MPEG-4 [17], [18]. For different coding schemes, different coding parameters may be employed and different constraints may be imposed. For instance, in MPEG-2, the most influential coding parameter with regard to picture quality is the quantization parameter (QP) used for texture coding. This parameter can be selected for the entire frame or change from macroblock to macroblock. In most implementations, it is selected based on a measure of buffer fullness so that the target bit rate can be obtained. Also, since the primary application for MPEG-2 is digital video broadcast, it is desirable to have a fixed GOP (group of picture) structure. By this, we mean that the anchor frame distance and $I$-frame interval are fixed within a particular GOP. In this way, the rate control algorithm cannot resort to changing the temporal coding parameter for buffer control. More on the MPEG-2 terminology and encoding process can be found in [19]. In contrast to this, the H.263 coding scheme *does* allow variable frameskip, and due to the low bit-rate conditions which may be imposed upon the encoder, it is up to the rate control algorithm to make appropriate decisions on both spatial and temporal coding parameters. This topic has been studied in [15]. Generally speaking, if the buffer is in danger of overflow, complete pictures will be discarded at the

Manuscript received October 23, 1997; revised April 20, 1998. This paper was recommended by Associate Editor M.-T. Sun.

A. Vetro and H. Sun are with the Advanced Television Laboratory, Mitsubishi Electric Information Technology Center America, New Providence, NJ 07974 USA.

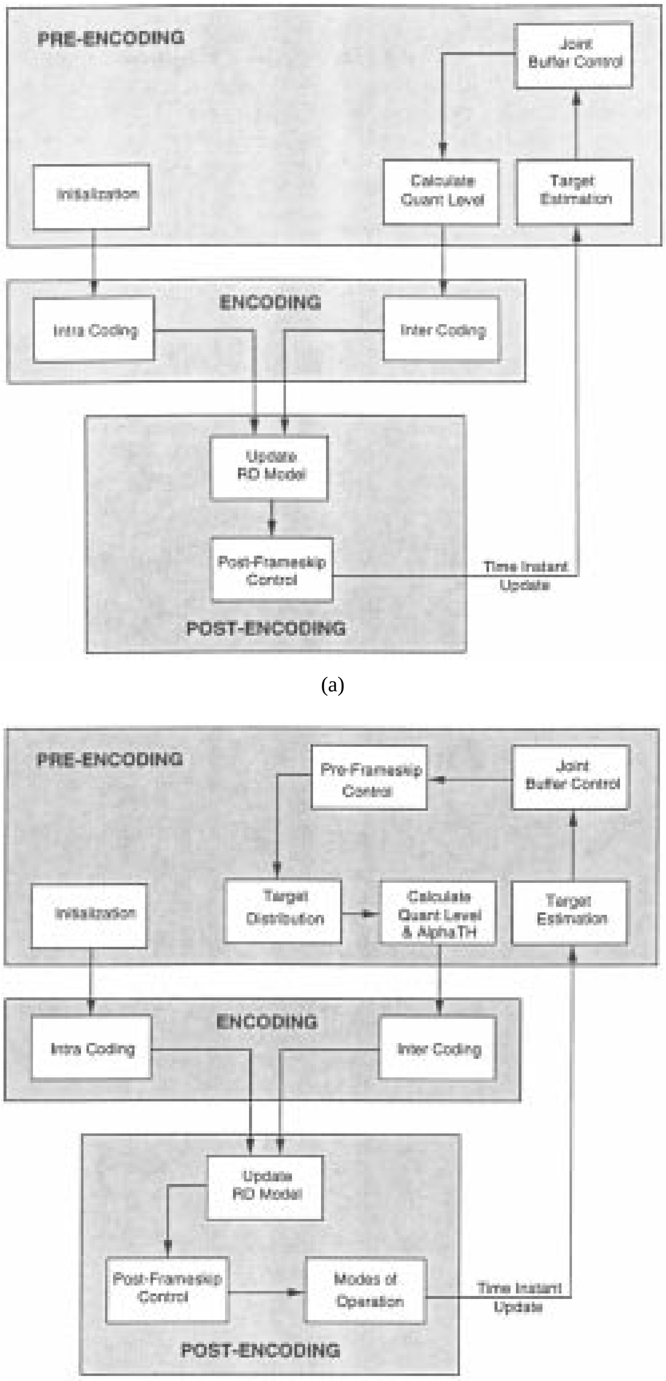
Y. Wang is with the Department of Electrical Engineering, Polytechnic University, Brooklyn, NY 11201 USA.

Publisher Item Identifier S 1051-8215(99)01191-X.

encoder. This will allow bits used to encode previous frames to be transmitted, thereby reducing the level of the buffer. In conjunction with this frame-skipping mechanism, the rate control algorithm must determine the suitable QP to obtain the desired bit rate.

Similar to the case of H.263, MPEG-4 rate control must also consider spatial and temporal coding parameters. However, since MPEG-4 also allows the coding of arbitrarily shaped objects, the encoder must consider the significant amount of bits which are used to code the shape information. This aspect of the encoder makes the rate control problem in MPEG-4 or any other object-oriented encoder unique. In fact, the rate control algorithm has a great deal of flexibility since each object may be encoded at a different frame rate. Also, additional coding parameters are introduced by MPEG-4 to control the amount of bits used to specify the shape of an object. It is the responsibility of the rate control algorithm to incorporate these new parameter decisions along with other parameter decisions (e.g., QP for texture coding each object) to ensure that the video objects are effectively coded and suitable buffer levels are maintained. The proposed rate control algorithm for multiple video objects (MVO) is an extension of the existing single video object (SVO) algorithm. The block diagrams of the SVO algorithm and the proposed MVO algorithm are given in Fig. 1(a) and (b), respectively. As shown, the MVO algorithm includes four major additions to the SVO scheme, namely, target distribution, preframeskip control, switching of operation modes, and shape-coding control (calculate AlphaTH). The first component enables individualized rate–distortion control over separate objects, whereas the second component ensures a stable buffer when the shape information occupies a large percentage of bit budget. Their addition to the SVO algorithm forms a basic framework for MVO rate control. The next two components aim at providing tradeoffs between the spatial and temporal coding resolutions and between shape and texture coding accuracy, and can enhance system performance under low bit-rate coding conditions.

The organization of the paper is as follows. Section II provides a review of the MPEG-4 SVO rate control algorithm, and introduces some of the preliminary concepts and notations that will be used throughout the paper. Section III identifies the fundamental issues which need to be addressed in order to adapt the existing SVO algorithm to handle multiple objects. We first present the basic framework for multiple video object rate control, and then detail the proposed target distribution scheme and the improved buffering policy. In Section IV, the two enhancement components are described. First, a mechanism is introduced to make the algorithm more adaptive and robust to low and high bit-rate coding conditions. Then a method to control the shape-coding rate by dynamically varying the shape-coding parameter AlphaTH is presented. Section V presents the outcome of our simulations under a number of testing conditions. These results serve to demonstrate the effectiveness of individual components in the proposed MVO algorithm. In Section VI, we summarize the main results and provide an outlook for future directions. This paper is based on our contribution to MPEG-4 [20] which was later adopted as part of the video Verification Model (VM8) [21].



(a)

(b)

Fig. 1. Block diagrams of SVO and MVO algorithms.

## II. SVO Rate Control

The relationship between rate and quantizer for texture coding has been given a considerable amount of attention for rate control applications. For example, in [22], a model is derived from classic rate–distortion theory, and then modified to match the encoding process of practical encoders and real image data. In [16], a generic rate–quantizer model was proposed which can be adapted according to changes in picture activity. Recently, Chiang and Zhang have proposed a new rate control scheme using a quadratic rate–quantizer model [18].

The algorithm in [18] was adopted by MPEG in November 1996 for SVO simulations, and is scalable for various bit rates, spatial and temporal resolutions, and can be applied to both DCT and wavelet-based coders. This algorithm will form the basis for the proposed MVO algorithm.

Fig. 1(a) shows a high-level block diagram of the SVO rate control algorithm. As we can see, the blocks are grouped into three major stages: preencoding, encoding, and postencoding. Actually, the rate control algorithm does not impose any changes to the main encoding engine, it only provides input (such as QP) to the encoding engine based on data it has gathered in the postencoding stage. At this stage, the following information is analyzed: the QP used for the current frame, the number of texture bits which resulted from this QP, and the total amount of bits transmitted. The QP and texture bits are used to determine the model parameters, and the total amount of transmitted bits is used to update the buffer level. If the updated buffer level is too high, the postframeskip control can choose to skip an appropriate amount of frames.

In the preencoding stage, we make use of the information from the postencoding stage. First, an initial target is estimated based on available bits and the number of bits which were used by the previous frame. This estimate is then refined based on the buffer fullness, and finally, the QP for the frame is calculated. This final calculation is based heavily on the current model parameters which were determined in the most recent postencoding stage. In the following, a more detailed description of the algorithm is given.

Let $T_{\text{texture}}$ denote the encoding bit count for the texture, MAD the mean absolute difference of the texture, which is an indication of the encoding complexity, $Q$ the quantization parameter for the frame, and $X1$ and $X2$ the first- and second-order model parameters. The rate control scheme assumes that $T_{\text{texture}}$ is related to $Q$, by

$$T_{\text{texture}} = \frac{X1 \cdot \text{MAD}}{Q} + \frac{X2 \cdot \text{MAD}}{Q^2}. \tag{1}$$

With the above relationship, the algorithm can be summarized in five steps: initialization, target bit-rate calculation, quantization level calculation, updating of model parameters, and postframeskip control which is responsible for updating the time instant.

*Initialization:* During this stage, buffer-related quantities are defined and encoding parameters are initialized for use in the algorithm. A summary of the notation can be found in Table I. To code the $I$-frame, an initial QP is specified. Once this frame is coded using $T_f$ bits, we need to determine the total number of bits $\tilde{T}_r$ which are available for the remainder of the image sequence

$$\tilde{T}_r = t_s \cdot R_s - T_f. \tag{2}$$

In (2), $t_s$ is the duration of the sequence in seconds and $R_s$ is the desired bit rate for the sequence. In addition, we need to know the average number of bits to be drained from the buffer per frame

$$R_{\text{drain}} = \frac{\tilde{T}_r}{\tilde{N}_r}. \tag{3}$$

In the above equation, $\tilde{N}_r$ denotes the number of $P$-frames which remain to be coded after the $I$ frame.

*Initial Target Bit-Rate Estimation:* The target bit number for each new $P$-frame is determined in three steps. The initial estimate is determined from the number of bits remaining $T_r$ and the number of bits used for coding the previous frame $T_p$ as follows:

$$T_1 = \max\left\{\frac{R_s}{F_s}, (1 - w_p)\frac{T_r}{N_r} + (w_p)T_p\right\} \tag{4}$$

where $F_s$ denotes the frame rate of the source material, $N_r$ denotes the number of frames which remain to be coded, and the constant $w_p$ serves as a weighting factor, with a typical value of 0.1. The lower bound $R_s/F_s$ is imposed so that a minimum quality can be met.

*Joint Buffer Control:* After the initial target has been determined, it is scaled based on the current buffer level $B_c$ and the buffer size $B_s$ as described in [19]

$$T_2 = T_1 \cdot \frac{B_c + 2(B_s - B_c)}{2B_c + (B_s - B_c)}. \tag{5}$$

This scaling is performed to maintain a buffer occupancy of about 50% after coding each frame. Further changes are made to the target to avoid overflow or underflow. Specifically, the final target estimate is described by [19]

$$T = \begin{cases} (1-\delta)B_s - B_c, & \text{if } B_c + T > (1-\delta)B_s \\ R_{\text{drain}} - B_c + \delta B_s, & \text{if } B_c - R_{\text{drain}} + T < \delta B_s \\ T_2, & \text{otherwise.} \end{cases} \tag{6}$$

A typical value of $\delta$ is 0.1.

*Quantization Level Calculation:* From the previous step, we are given a target rate for the entire frame. To estimate the target bits for texture, the bits used for motion and header information of the previous frame are subtracted from the total. With these remaining bits for texture, the known model parameters $X1$ and $X2$, and the MAD, the QP for the frame can be calculated using (1). As usual, the QP is limited to vary between 1 and 31, and allowed to change within 25% of the previous QP.

*Updating the Model Parameters:* The model parameters for the rate–quantizer relationship are continually updated based on encoding results of the current frame as well as a specified number of past frames ($n$ frames total). Only bits which are relevant to the texture component are considered in this calculation, i.e., actual bits used for the header and motion are deducted from the total. The first- and second-order complexities $X1$ and $X2$ are solved for by using a least squares estimation [18]. More specifically, the $n$ QP values and corresponding bit counts from the current and past $n$ frames are used to solve a set of over complete linear equations for $X1$ and $X2$. In this estimation process, the model is calibrated by rejecting outlier data points. The rejection decision is that a data point is discarded when the error between the predicted amount of bits by the model and the actual number of bits used for a frame is more than one standard deviation among the $n$ frames. As a final point, the number of frames $n$ will change according to the MAD. If there is a scene change, i.e., MAD is a large value, a smaller value of $n$ is used.

TABLE I
SUMMARY OF NOTATION

| *Variable* | *Definition* |
|---|---|
| $R_s$ | Bit rate for the sequence |
| $F_s$ | Frame rate of the source material |
| $t_s$ | Duration of the sequence in seconds |
| $N_r$ | Number of P-frames remaining to be coded |
| $B_s$ | Buffer size |
| $B_c$ | Current buffer level |
| $B_p$ | Previous buffer level |
| $R_{drain}$ | Number of bits which drain from the buffer per coded frame |
| $T_r$ | Number of bits available |
| $T_f$ | Number of bits used for the first frame |
| $T_c$ | Number of bits used for the current frame |
| $T_p$ | Number of bits used for the previous frame |
| $T_{hdr}$ | Shape, motion, and header information of previous frame |
| $T$ | Target bits for frame (texture, motion, shape, and header) |
| $T_i$ | Target bits for object $i$ (incl. texture, motion, shape, and header) |
| $T_{texture}$ | Target bits for texture of frame |
| $T_{texture,i}$ | Target bits for texture of video object $i$ |
| $MAD$ | Mean absolute difference of the current frame after motion compensation |
| $X1, X2$ | First and second order complexities |
| $Q$ | Quantization level for the current frame |
| $Q_{lb}$ | Lower bound on quantization parameter |
| $\delta$ | Safety margin for buffer control |
| $\gamma$ | Skip margin for time instant update |
| $\beta$ | Bit threshold for frameskip control |
| $N_{post}$ | Number of frames to skip as determined by post-frameskip control |
| $N_{pre}$ | Number of frames to skip as determined by pre-frameskip control |
| $SkipTH$ | Frameskip threshold used in selecting the mode of operation |
| $AlphaTH$ | Shape rate control parameter |

*Postframeskip Control:* After encoding a frame, the total number of bits which were used $T_c$ is added to the current buffer level, and decreased from the remaining bits $T_r$. To ensure that the updated buffer level is not too high, the frameskip parameter $N_{\text{post}}$ is set to zero and incremented until the following buffer condition is satisfied:

$$B_c < \gamma B_s \tag{7}$$

where

$$B_c = B_p + T_c - R_{\text{drain}}(N_{\text{post}} + 1). \tag{8}$$

In (7), the value of $\gamma$ denotes a skip margin having a typical value of 0.8, and in (8), the parameter $B_p$ denotes the previous buffer level.

## III. MVO Rate Control: Fundamental Issues

Due to the favorable performance of the SVO algorithm and the ease of implementation, it is desirable for the MVO scheme to employ a similar framework. However, the extension is nontrivial as there are many open issues which need to be addressed. In this section, we first give an overview of the proposed MVO algorithm, and then focus on the two fundamental issues: how to select the QP for each object, and how to administer a buffering policy. The solution to these two problems provides a basic framework in which the MVO algorithm can operate.

### A. Overview of the MVO Algorithm

Fig. 1(b) shows the block diagram of the proposed MVO algorithm. In comparison to the SVO scheme, many of the blocks are the same, however, some operate on an object-based level. In the following, we describe the various components in Fig. 1(b) briefly. The four added components will be discussed in more detail in separate sections.

*Initialization:* The initialization process is not very different from the SVO process described before. Most of the notation is unchanged, but many of the variables are extended to vector quantities so that each object can maintain its own set of parameters.

*Initial Target Bit-Rate Estimation:* To estimate an initial total target bit rate, the solution given by (4) can be used. Alternatively, the target can be made object based by allocating the bit rate for the $i$th object proportional to the bit rate used for the $i$th object of the previous frame $R_{p,i}$

$$T_1 = \sum_{i \in \mathcal{M}} T_i, \quad \text{with}$$

$$T_i = \max_{i \in \mathcal{M}} \left\{ \frac{R_s}{m \cdot F_s}, (1 - w_p) \frac{T_r}{m \cdot N_r} + (w_p) R_{p,i} \right\}. \tag{9}$$

In the above equation, $\mathcal{M} = \{0, 1, \cdots, m\}$ is the set of video object (VO) id's. An increase in the value of $w_p$ will skew the individual targets more proportional to $R_{p,i}$. A value of $w_p = 0.25$ was used in our experiments. It should be noted that the initial estimate does not need to be very accurate, and either of the above two methods can be used.

*Joint Buffer Control:* For the MVO algorithm, the scaling procedure of (5) and the overflow/underflow adjustments of (6) can be performed in the same way. However, as an added precaution for excess shape information at low bit rates, the safety margin $\delta$ is increased to 0.25.

*Target Distribution:* In this step, the output target of the joint buffer control is distributed among each of the arbitrarily shaped VO's to yield the target bit number $T_i, i \in \mathcal{M}$ for individual objects. The proposed solution for this problem will be discussed in the next subsection.

*Quantization Level Calculation:* Given the values of $X1_i$, $X2_i$, $\text{MAD}_i$ and $T_{\text{texture},i}$, the appropriate values of $Q_i$ can easily be found. The target number of bits for the texture of the $i$th object is defined as

$$T_{\text{texture},i} = T_i - T_{hdr,i} \tag{10}$$

where $T_{hdr,i}$ represents the amount of shape, motion, and header bits used for the $i$th object of the previous frame. In our implementation, motion is always coded losslessly, while the shape can be coded losslessly or lossy. Under normal circumstances, the algorithm requires no change to the quantization level calculation of Section II, only that the correct object-based parameters be used, by replacing $T_{\text{texture}}$ in (1) with

$$T_{\text{texture},i} = \left[ \frac{X1_i \cdot MAD_i}{Q_i} + \frac{X2_i \cdot MAD_i}{Q_i^2} \right]. \tag{11}$$

*Shape-Coding Parameter (AlphaTH) Calculation:* This block is used to determine AlphaTH, the parameter that controls shape distortion in MPEG4. The adjustment of this parameter can provide a tradeoff between texture and shape coding accuracy. For now, we assume that it is fixed to zero, which leads to lossless shape coding. This block will be discussed in more detail in Section IV-B.

*Updating the Model Parameters:* Using (11), the object-based complexities $X1_i$ and $X2_i$ are determined just as before in Section II, except that $T_{\text{texture},i}$ is used rather than $T_{\text{texture}}$.

*Postframeskip Control:* As mentioned in Section I MPEG-4 allows each object to be coded at a different frame rate. In the proposed algorithm, we impose the restriction to code each object at the same frame rate. This is done to avoid problems with composition. In other words, when two objects are coded at different frame rates, it is very likely that undefined pixels will be present in the composite image sequence. Although a large amount of savings can be achieved by coding objects at different frame rates, a method to overcome the composition problem is required.

With the above assumption, the method of postframeskip control is basically the same as the SVO algorithm. The only new consideration is that the buffer level is now updated with shape bits in addition to bits used for texture, motion, and header information.

*Preframeskip Control:* At high bit rates, the number of shape bits is small, and shape can be considered side information. However, at low bit rates, this is no longer the case. The large percentage of the shape information can cause buffer overflow, even with the skipping of frames exercised in the postencoding stage. To anticipate potential buffer overflow, an additional frameskip control is added in the preencoding stage. This improved buffering policy for handling excess side information will be discussed in Section III-C.

*Mode of Operation:* Ideally, the various parameters in the proposed rate control scheme should be adapted based on the coding environment, e.g., high rate versus low rate. This is accomplished by switching between two operation modes. The discussion on this block is reserved until Section IV-A.

### B. Target Distribution Among Objects

Generally speaking, an object-based coder attempts to code each object with a different quantization parameter. This is done to exploit the fact that each object need not be coded with the same precision to achieve comparable quality. For example, a stationary background coded with a QP of 25 may have a higher quality decoded output than a more complex moving object that was coded with a QP of 18. To accomplish the task of finding appropriate QP values for every object in the scene, it is necessary to extend the SVO algorithm to analyze object-based data and distribute the total target bit for a frame among multiple object.

The bit allocation problem has been treated in many papers. For the macroblock level case, Pickering and Arnold propose a perceptually efficient VBR rate control algorithm [26]. In this work, it is suggested that a perceptual masking factor be used to classify blocks, where the masking factor was determined based on a spatial derivative, an activity factor, and a motion factor. In related works [23], [24], measures such as the variance, contrast, and size were incorporated to locate areas of interest, and even predict the quality that one may obtain upon coding. Other research has tried to exploit facial models to apply different degrees of spatial and temporal scalability to different areas of the scene in videotelephony applications [25]. In the statistical multiplexing (StatMux) problem [27], it has been shown that adjustments can be made on the quantization parameter of several encoders to ensure that the channel capacity is being efficiently utilized. This method depends highly on the statistical variation among several programs, and attempts to achieve uniform quality among every program.

In the proposed target distribution algorithm, a combination of philosophies from the perceptually efficient approach and the StatMux approach is used to distribute the target. At the same time, it is very important that all of the factors used are easily computed. The three measures that we have chosen for target rate distribution are the size, motion and a variance-like measure, the MAD$^2$. In [29], it was suggested that the MAD$^2$ is a better model of the variance than simply the MAD. So, for a given target, the target for object $i$ is given by

$$T_i = T \cdot (w_s \mathrm{SIZE}_i + w_m \mathrm{MOT}_i + w_v \mathrm{VAR}_i) \tag{12}$$

where $\mathrm{SIZE}_i$, $\mathrm{MOT}_i$, and $\mathrm{VAR}_i$ are the size, motion, and MAD$^2$ of object $i$, normalized by the total SIZE, MOT, and VAR of all objects, respectively. Here, the motion magnitude of the $i$th object, $\mathrm{MOT}_i$ is the sum of the absolute values of each motion vector component within object $i$, and the size of the object $\mathrm{SIZE}_i$ is simply the number of macroblocks or partial macroblocks within the object. The weights $\{w_s, w_m, w_v\} \in [0, 1]$ and satisfy: $w_s + w_m + w_v = 1$. Typical values of the weighting factors will be discussed in Section IV.

Once the total bit number for each object is determined, the available bits for texture can be derived by subtracting the bits used for motion, shape, and other side information. Then the quantization parameter can be determined using the rate–quantizer model for each object, as described before. For now, we assume that the shape and motion information is coded losslessly.

### C. Improved Buffering Policy

Under low bit-rate coding conditions, it is very likely that the buffer overflows when using the buffering policy of the SVO algorithm. The reason is that shape information tends to use a considerable percentage of the bit rate, and is not accounted for until the actual bits have been spent. If too many bits have already been spent, the buffer will overflow. The only remedy, which is an “after effect,” that the existing scheme can provide is to skip additional frames before coding the next frame.

In this section, an improved buffering policy is described to compensate for the effects of large side information. With this, appropriate adjustment to QP are made before too many texture bits have been spent. Also, appropriate temporal adjustments are made to anticipate the usage of the additional shape bits. These spatial and temporal adjustments are achieved by introducing a new block to the algorithm, preframeskip control, which has an effect on how the quantization levels are calculated and how the time instant is updated. Overall, a much more stable buffer occupancy can be achieved.

*Preframeskip Control:* Often, in low bit-rate coding conditions, the target which emerges from the joint buffer control may not be enough to even code the motion, shape and header information, let alone the texture. In the preencoding stage, a positive target for the texture is needed so that a suitable quantization parameter can be determined. However, it is possible that all of the target bits are used by information other than the texture. In that case, there must be a mechanism to effectively alert other parts of the system that there is some deficiency in the number of allocated bits. Among those system components which are affected are the target distribution and QP calculation, as well as the time instant update.

The most obvious remedy is to allow more frames to be skipped during the postencoding stage so that the buffer control will allow more bits to be allocated for the next frame to be coded. As a result, the value $N_{\mathrm{pre}}$ is determined so that additional frames will be skipped in the next postencoding stage. Specifically, let $s = T - T_{hdr}$ be the difference between the target and the amount of bits used in the previous frame for the shape, motion, and header; $N_{\mathrm{pre}}$ is determined by the following algorithm:

```
while $(s < \beta)$
    $N_{\mathrm{pre}} = N_{\mathrm{pre}} + 1$
    $s = s + R_{\mathrm{drain}}$
```

where $\beta \geq 0$ is a bit threshold.

It should be emphasized that no attempt is made to skip additional frames in the preencoding stage. This action is reserved for the postencoding stage; the changes to the postframeskip control are discussed below.

*Quantization Level Calculation:* The object-based QP’s are determined in the same manner as before. However, adjustments on the QP are made based on the new information that has been extracted in the preframeskip control. In the event that $N_{\mathrm{pre}}$ is greater than zero, the quantization parameter should be lower bounded so that the actual bits used for coding the texture information is not excessive. Letting $Q_{lb}$ denote this bound, the QP which is used for a particular object is constrained in the range $[Q_{lb}, 31]$. A typical value of $Q_{lb}$ is 28.

*Postframeskip Control:* In the SVO case, the number of shape bits is zero, and other bits pertaining to the motion and header information are relatively small compared to the texture bits. Because of this, the rate–quantizer model is able to accurately predict the distortion given some rate and vice versa, leading to a stable buffer which can always be compensated for by using the condition given by (7). For low-bit rate coding of MVO’s, buffer levels are less predictable due to the relatively large amount of shape information. Since the number of bits used for shape may have a dramatic influence on the buffer levels, some means of compensation needs to be considered. The first action which can be taken is to make the postframeskip control more robust by considering the total amount of bits $T_p$ that were spent on the previous frame rather than only considering the current buffer level. This is accomplished by replacing (7) and (8) with

$$\tilde{B}_c + T_p - R_{\text{drain}} < \gamma B_s \tag{13}$$

and

$$\tilde{B}_c = B_p + T_c - R_{\text{drain}}(N_{\text{post}} + 1). \tag{14}$$

Note that the above condition complements the motivation of the preframeskip control to account for excess header bits in the skipping mechanism. This is true since $T_c$ would include the large percentage of shape bits for low-bit rate simulation.

As a second action, the value of $N_{\text{pre}}$ should be taken into account. Once the value of $N_{\text{post}}$ has been found, the sum $N_{\text{tot}} = N_{\text{post}} + N_{\text{pre}}$ is formed, and the buffer is ultimately updated according to

$$B_c = B_p + T_c - R_{\text{drain}}(N_{\text{tot}} + 1). \tag{15}$$

Essentially, the use of $N_{\text{pre}}$ in the above equation represents the error in the frameskip from the previous postencoding stage. Since the safety margin was increased, the error was absorbed by coding the current frame with a lower spatial quality. Although the above techniques do not guarantee that buffer overflow will not occur, the simulation results in Section V-A provide strong evidence that it is unlikely.

## IV. MVO Rate Control: Enhancement Issues

In the previous section, the basic elements of an MVO rate control scheme were considered. Additionally, some helpful tools such as the preframeskip control and an improved buffer condition were discussed. These elements together can provide reasonable quality and maintain a stable buffer. To further improve the system performance, the proposed MVO algorithm also include two additional components, which are discussed in this section.

MPEG-4 bit streams are expected to be used in a variety of coding environments. In most instances, it will not be specified whether the environment is considered low bit rate, high bit rate, or somewhere in between. Since coding decisions may change according to the environment, it is desirable to have a mechanism to detect and keep an update of such changes. To this effect, we propose two different *modes of operation*: one for encoding at low bit rates, and another for encoding at high bit rates.

In coding a set of arbitrarily shaped video objects at a high bit rate, it seems appropriate to code the shape of each object losslessly. On the other hand, when the shape of each object needs to be specified in a low bit-rate coding environment, the percentage of bits used for shape information may be excessive if coded losslessly. In this case, it is quite probable that the number of bits which remain for texture coding is small or inadequate. Also, a significant decrease in the temporal resolution may be experienced. Therefore, we propose a shape-coding control to reduce the amount of bits used for shape coding so that more bits can be used for texture coding and/or more frames can be coded. Since this type of control would only be invoked under low bit-rate conditions, it makes sense to utilize the mode of operation for this purpose. More on these two topics is discussed in the following subsections.

### A. Modes of Operation

In video coding applications, the environment in which the encoder is forced to operate can depend on a number of factors, e.g., channel rate, encoding rate, and scene content. As these factors change, the various control parameters in the rate control algorithm should make appropriate adaptations as well. In [15], some experimental studies have been performed to choose between coding at a low frame rate with high quality versus a high frame rate with low quality. Here, we would like to vary the control parameters to exercise a similar type of spatiotemporal control, i.e., should more frames be coded with a coarser QP, or should fewer frames be coded with a finer QP. In order to avoid the excessive complexity associated with a fine granularity of adaptation, we propose to switch between two operation modes, depending on the current temporal coding resolution. Because the frameskip parameter basically reflects this information, the rate control algorithm determines the mode of operation as follows:

```
if ($N_{\text{tot}} > SkipTH$)
    Operate in $LowMode$
else
    Operate in $HighMode$.
```

In the current implementation, the skip threshold was set to two. In a more advanced scheme, the skip threshold can be viewed as a tolerance parameter, where the actual frame rate would be allowed to deviate from the target frame rate by a certain percentage. The complexity of this scheme is negligible since we would only need to keep a record of the actual frame rate.

If we are in LowMode, we know that the encoder has skipped a minimum number of frames. To prevent the coder from continuing to skip excessive frames, the current frame should be coded with a coarser quantizer. Therefore, LowMode should impose a lower bound on the calculated quantization parameter. This lower bound $Q_{lb}$ is the same as that used in the preframeskip control of the previous section. Although the bounds are the same, the purpose is very different. When using $Q_{lb}$ from the preframeskip control, the algorithm is attempting to compensate for the deficiency in the target; on the other hand, when using $Q_{lb}$ from the LowMode decision, the algorithm is attempting to increase the temporal resolution for the remainder of the sequence. This approach proves to be very effective in controlling the frame rate and associated spatial quality.

Besides imposing constraints on the coding parameters, the algorithm can also define a heuristic for target distribution. When encoding at high bit rates, the availability of bits allows the algorithm to be flexible in its target assignment to each VO. Under these circumstances, it is reasonable to impose homogeneous quality among each VO. Therefore, the inclusion of VAR$_i$ is essential to the target distribution, and should carry the highest weighting. On the other hand, when the availability of bits is limited, it is very difficult (if not impossible) to achieve homogeneous quality among the VO. Also, under low bit-rate constraints, it is desirable to spend fewer bits on the background and more bits on the foreground. Usually, the background has a smaller relative motion to the foreground, consequently, the significance of the variance should be decreased and the significance of the motion should be increased. Based on the above arguments and experimental trial and error, the weights used in our experiments were: $w_m = 0.6, w_s = 0.4, w_v = 0.0$ for LowMode and $w_m = 0.25, w_s = 0.25, w_v = 0.5$ for HighMode. In testing the algorithm, we have found that the encoder performance is not very sensitive to the specific weighting factors as long as the heuristic discussed above is followed.

### *B. Decisions on Shape Rate Control Parameters*

According to [21], rate control and rate reduction of shape information can be achieved through size conversion of the alpha plane. The possible conversion ratios (CR) are 1, 1/2, or 1/4. In other words, a 16 × 16 macroblock (MB) may be downconverted to an 8 × 8 or a 4 × 4 block. Each macroblock containing relative shape information for the object can be down-converted for coding, then reconstructed to the original size. A conversion error is calculated for every 4 × 4 pixel block (PB). The conversion error is defined as the sum of absolute differences between the value of a pixel in the original PB and the reconstructed PB. If the conversion error is larger than 16 × AlphaTH, then this PB is referred to as an *Error PB*. If there is one Error PB in the macroblock, then the CR for the macroblock is increased, with the maximum value being 1.

From the above discussion of shape coding, it is evident that the value of AlphaTH has a considerable effect on the number of bits which will be spent on the shape information. To control the number of bits for shape coding, we propose to vary the value of AlphaTH based on the current mode of operation and the output of the preframeskip control. Specifically, AlphaTH is adapted according to

```
if $(LowMode \text{ OR } N_{\text{pre}} > 0)$
    $AlphaTH = \min\{AlphaTH + AlphaINC, AlphaMAX\}$
else
    $AlphaTH = \max\{0, AlphaTH - AlphaDEC\}$
```

where $AlphaINC$ and $AlphaDEC$ denote constants which increment or decrement the current value of $AlphaTH$.

Using the above algorithm, AlphaTH may vary between 0 and AlphaMAX, where AlphaTH = 0 implies lossless shape coding. A discussion on choosing appropriate values for AlphaMAX, AlphaINC, and AlphaDEC is provided in Section V-B. Once AlphaTH is chosen by the above algorithm, the shape coding is executed. The significance of this scheme is that: 1) it is flexible in that it does not make a hard decision on the AlphaTH, but directs it in a favorable way, and 2) it provides a way of reducing the bit rate used for shape. Although this will increase the distortion for shape, more bits will be available for texture coding and/or an increase in the temporal resolution.

## V. Simulation Results

In this section, experimental results are provided to evaluate the performance of individual components within the proposed MVO algorithm. From this, we will justify the need for certain components, and discuss the impact of others on the coding quality. All simulations are based on the VM8.0 software. The first set of experiments is aimed at demonstrating the usefulness of the preframeskip control to regulate the buffer occupancy. Two sets of simulations are produced: one which does not employ the improved buffering policy presented in Section III-B (MVO1), and one which does (MVO2). In both cases, the enhancement tools discussed in Section IV are disabled. The same weighting factors are used for the target distribution in both algorithms: $w_m = 0.25, w_s = 0.25, w_v = 0.5$. In the next set of experiments, the modes of operation are added to MVO2 and simulations are performed using various values of AlphaTH which remain fixed for every frame. The goal here is to investigate the tradeoff in bits between shape and texture and analyze the resulting object quality. This algorithm is referred to as MVO3 with fixed AlphaTH. Finally, a comparison is made between and MVO3 with lossless shape coding and MVO3 with dynamic shape rate control.

### *A. Comparison of Buffer Occupancy in MVO1 and MVO2*

To illustrate the impact of the improved buffering policy, a variety of testing conditions were considered at both low and high bit rates. The number of objects in each scene were as follows: Akiyo(2), Container(6), News(4), and Coastguard(4). Obviously, the number of objects in the scene has some impact on the difficulty to control the buffer level; however, the complexity of each shape has significant bearing as well. For the purpose of algorithm testing, the buffer size $B_s$ was set to half the rate $R_s/2$, and the initial buffer level was set to $R_s/4$. This means that, after coding the first $I$-frame, the buffer occupancy was 50%.

As we can see from the plots in Figs. 2 and 3, the buffer occupancy for the MVO2 algorithm is quite stable over the broad range of testing conditions and is always under 100%. The occupancy has a mean of approximately 50% and variations of about ±20%. From these results, it is safe to say that the buffer has very little chance of overflow/underflow. This type of behavior is also demonstrated by the MVO1 algorithm under high bit-rate conditions. However, in examining the buffer occupancy plots produced by the MVO1 algorithm under low bit rate conditions, it is evident that the algorithm has less control over the level of the buffer. In every low bit-rate simulation, the buffer experiences at least one overflow; in the case of *Container* at 10 kbits/s, every frame which is coded results in an overflow of the buffer.

As expected, MVO2 outperforms MVO1 under low bit-rate conditions and provides similar performance under high bit-rate conditions. Under the low bit-rate conditions, the object-based QP's which are calculated by the MVO1 only consider the texture information of the individual object. In other words, the preencoding stage of the MVO1 does not try to allocate bits for the texture and shape jointly. As a result, the texture coding is unaware of the possibility that a relatively

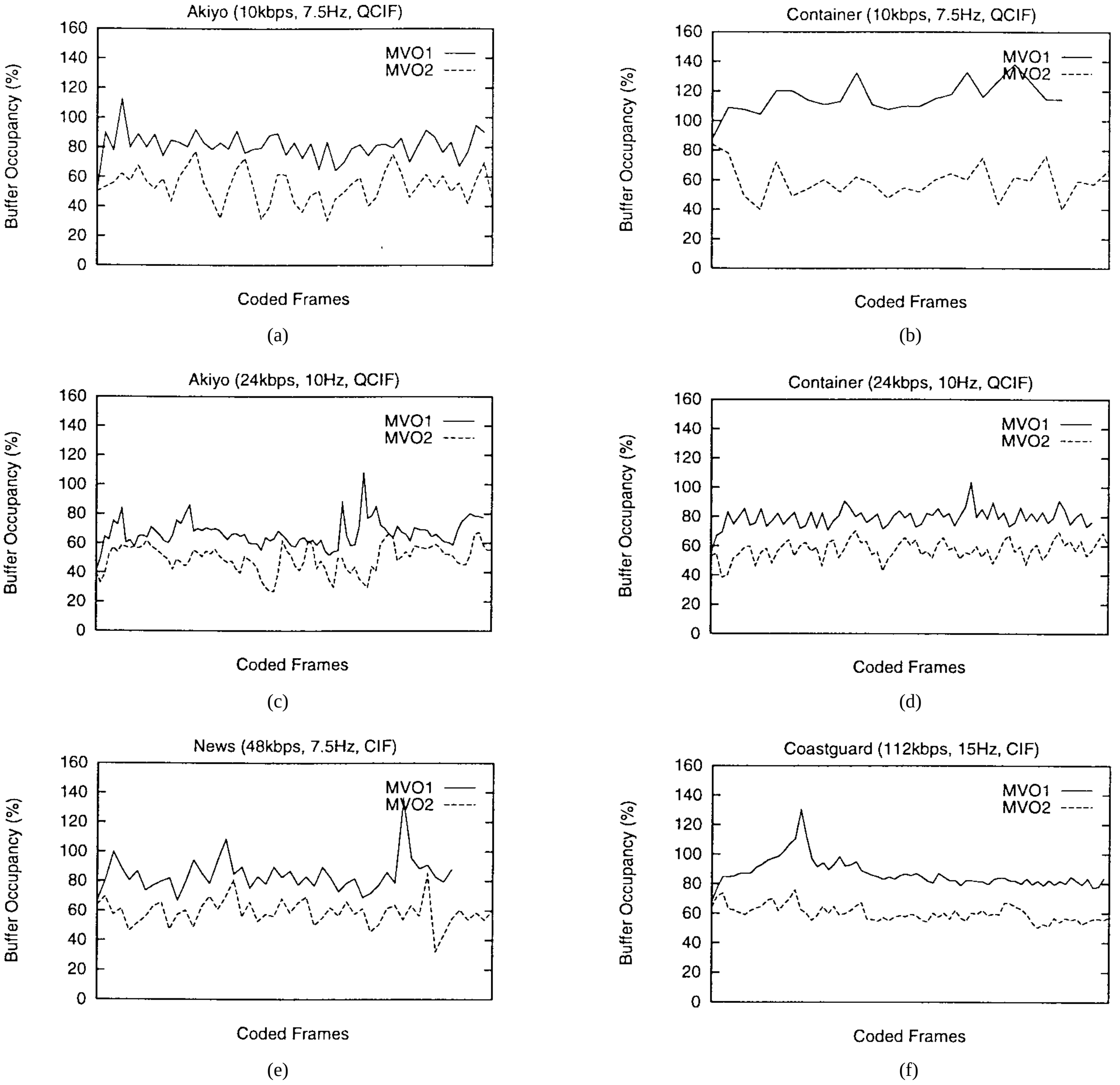


Fig. 2. Comparison of buffer occupancy plots for low bit-rate testing conditions.

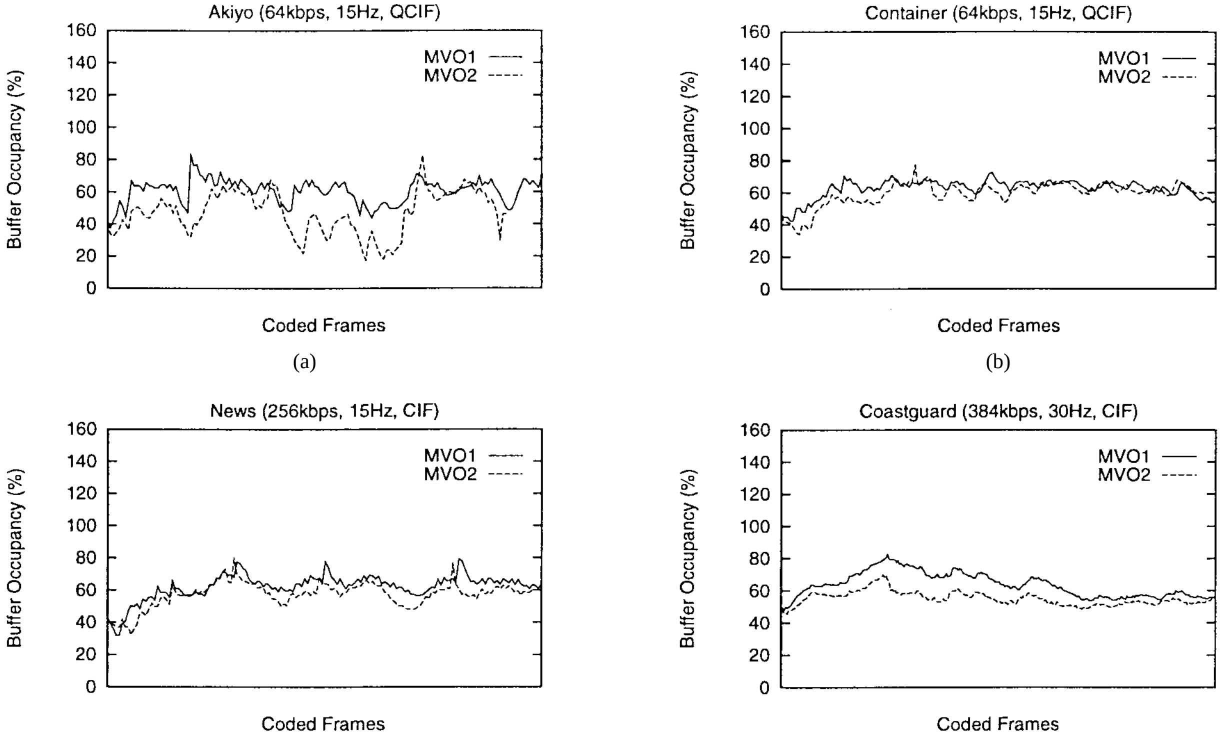
large number of bits may used for shape coding. When these bits for texture and shape are added together (along with motion and header bits), the total can easily become excessive, and at this point, the only means of compensation is to skip more frames. Often, buffer overflow has already occurred.

### B. Analysis of MVO3 with Fixed AlphaTH

To gain insight regarding the nature of the shape information and to understand its impact on the rate control algorithm, a number of tests are conducted on two video objects (VO2 and VO3) of the *Coastguard* test sequence. VO2 is a small boat with some motion and relatively complex shape, while VO3 is a larger background landscape with simple shape. In all simulations, the value of AlphaTH does not change from frame to frame; the value stays fixed.

In our first experiment, we examine the tradeoff in bits for texture and shape at various AlphaTH. The plots are shown in Figs. 4 and 5 for VO2 and VO3, respectively. As expected, the number of shape bits occupies a large percentage of the total bits at low bit rates, and also, the number of shape bits decreases with larger AlphaTH, while the number of texture bits increases. An interesting phenomenon, which is somewhat unexpected, is that the average bits/frame is increased for shape when the bit rates become lower. The reason is that the temporal resolution is reduced, therefore, the change in shape from one coded frame to the next is increased. Since the shape coding in MPEG-4 uses intercoding techniques, the bits are expected to increase. The last point that we would like to make regarding these plots is that they demonstrate a change in the bit requirement for shapes of varying complexity. More specifically, the bit requirement for VO2 is significantly larger than that for VO3.

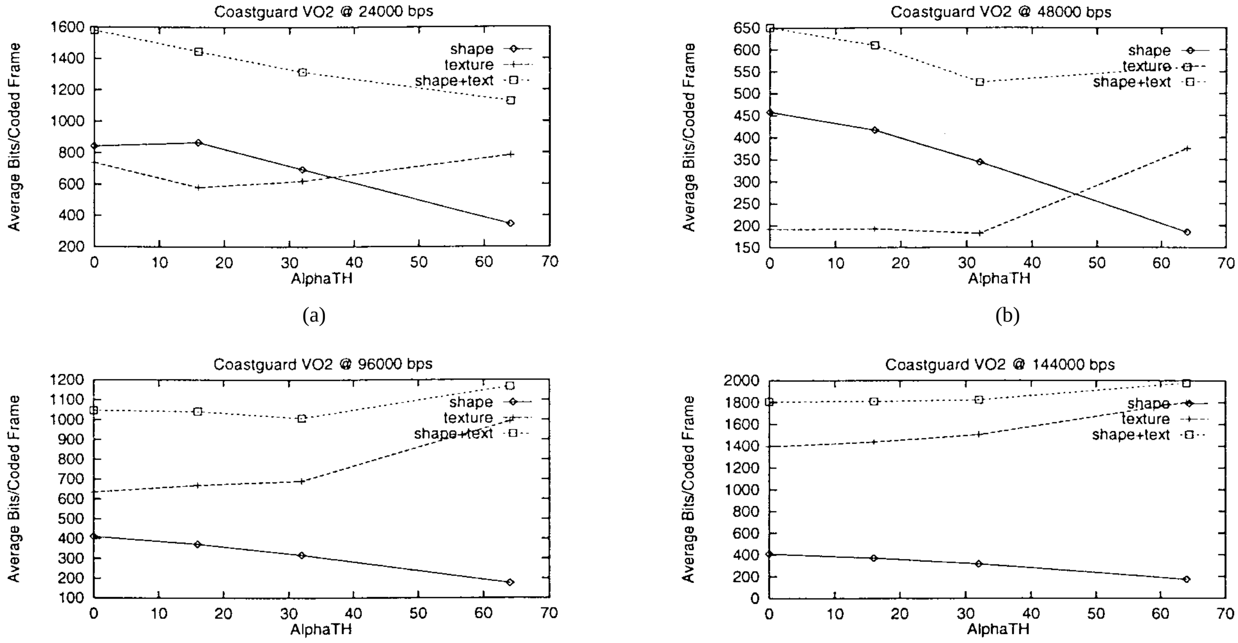
Now that the impact of shape information on the bit rate is understood, we move toward analyzing the distortion of the shape at various bit rates and AlphaTH's. More importantly, we are interested in the effect shape distortion has on the



(a) (b)

(c) (d)

Fig. 3. Comparison of buffer occupancy plots for high bit-rate testing conditions.

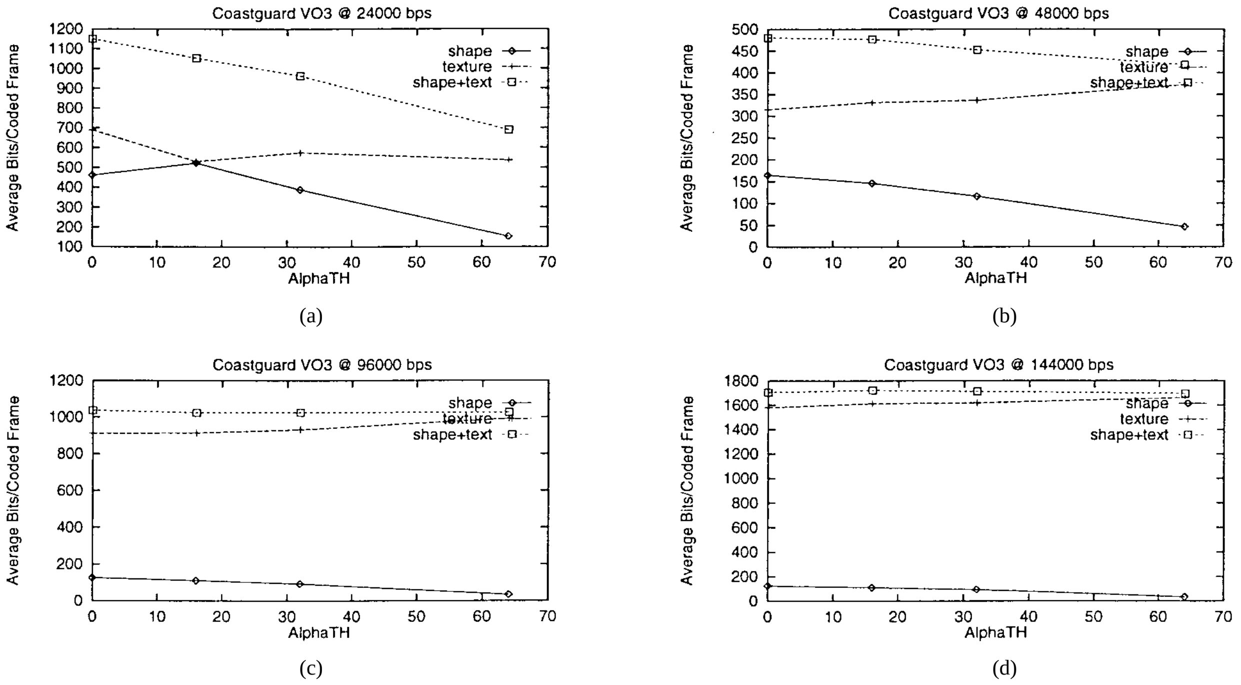


(a) (b)

(c) (d)

Fig. 4. Illustration of tradeoffs in shape and texture bits for VO2 of *Coastguard* sequence at different bit rates and AlphaTH values.

overall video quality. Fig. 6 illustrates the object-based $R$–$D$ curves for each object. The most important conclusion that can be reached from this plot is that an AlphaTH greater than zero can provide slightly higher PSNR at lower bit rates. Another conclusion which can be extracted from these plots is that simpler shapes are more resilient to distortions brought on by lossy shape coding. However, at the highest bit rate (144 kbits/s), both plots agree that lossless shape coding is the best in terms of coding efficiency. These plots are useful in that they also provided some indication on how to choose the parameter values which are required by the shape rate control algorithm. We see that using AlphaTH $=$ 0, 16, and 32 yield very similar results, whereas AlphaTH $=$ 64 leads to inferior performance under all bit rates. Based on this observation,



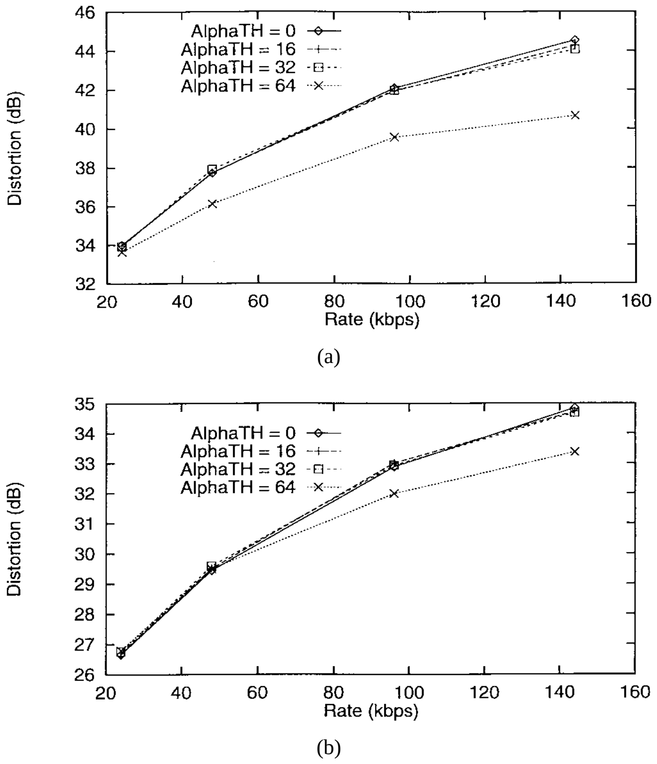
(a) (b)

(c) (d)

Fig. 5. Illustration tradeoffs in shape and texture bits for VO3 of *Coastguard* sequence at different bit rates and AlphaTH values.

for the proposed MVO algorithm with dynamic shape coding control, the following values will be used: AlphaMAX = 36, AlphaINC = 12, AlphaDEC = 12. The maximum value is chosen so that the shape will not be overdistorted, and the increment/decrement values allow one to choose the number of intermediate steps between lossless and maximum distortion shape coding.

Figs. 5 and 6 show that varying the AlphaTH parameter does not lead to significant changes in PSNR. Next, we examine its impact on the temporal resolution. Since the number of shape bits is decreased with increasing AlphaTH, it can be expected that the number of coded frames will increase; Table II supports this notion. With regard to QP, Table III shows the average QP which was used for each object for every testing condition. From this table, we see two interesting things. For one, at 24 kbits/s, the QP's for every object are approximately equal. This is due to the limited avalaiblity of bits for the texture and possibly the lower bound $Q_{lb}$ imposed by the preframeskip control. But, as the bit rate increases and the constraints are lifted, we observe that lower QP's are automatically assigned to the more interesting foreground objects (VO0 is water and VO1 is another boat). Second, as the value of AlphaTH increases, the change in QP for every object decreases slightly—more so for the low bit rates. This, in conjunction with Table II, allows us to conclude that bits that were previously used for shape are now used to increase the temporal resolution when it is deficient. This increase is visually noticeable, leading to an improvement in visual quality that is greater than that indicated by the marginal gains in PSNR.



(a)

(b)

Fig. 6. *R–D* curves for VO2 and VO3 of *Coastguard* sequence. See text for explanation of PSNR calculation.

Note that lossy shape coding will result in undefined pixels on the object boundary. If these pixels are ignored, the shape distortion is not accountable, and even very high values of AlphaTH still provide high quality for the pixels which are defined. On the other hand, if these pixels are incorporated into the calculation by setting them to a constant value, a severe drop in PSNR will be experienced, even when the shape distortion is visually acceptable. In our simulation, undefined pixels are simply replaced with a gray value of 128. This simple method of treating undefined pixels has the effect of undermining the margin of improvement achievable with

TABLE II
COMPARISON OF CHANGE IN TEMPORAL RESOLUTION FOR VARYING AlphaTH AND BIT RATE. MAX NUMBER OF FRAMES CODED = 150 SINCE ORIGINAL 10 s 30 Hz SEQUENCE WAS CODED AT 15 Hz

| AlphaTH | Bit-Rates (kbps) | | | |
|---|---|---|---|---|
| | 24 | 48 | 96 | 144 |
| 0 | 50 | 123 | 149 | 150 |
| 16 | 52 | 125 | 149 | 150 |
| 32 | 55 | 130 | 150 | 150 |
| 64 | 59 | 133 | 150 | 150 |

TABLE III
COMPARISON OF CHANGE IN QP FOR VARYING AlphaTH AND BIT RATE

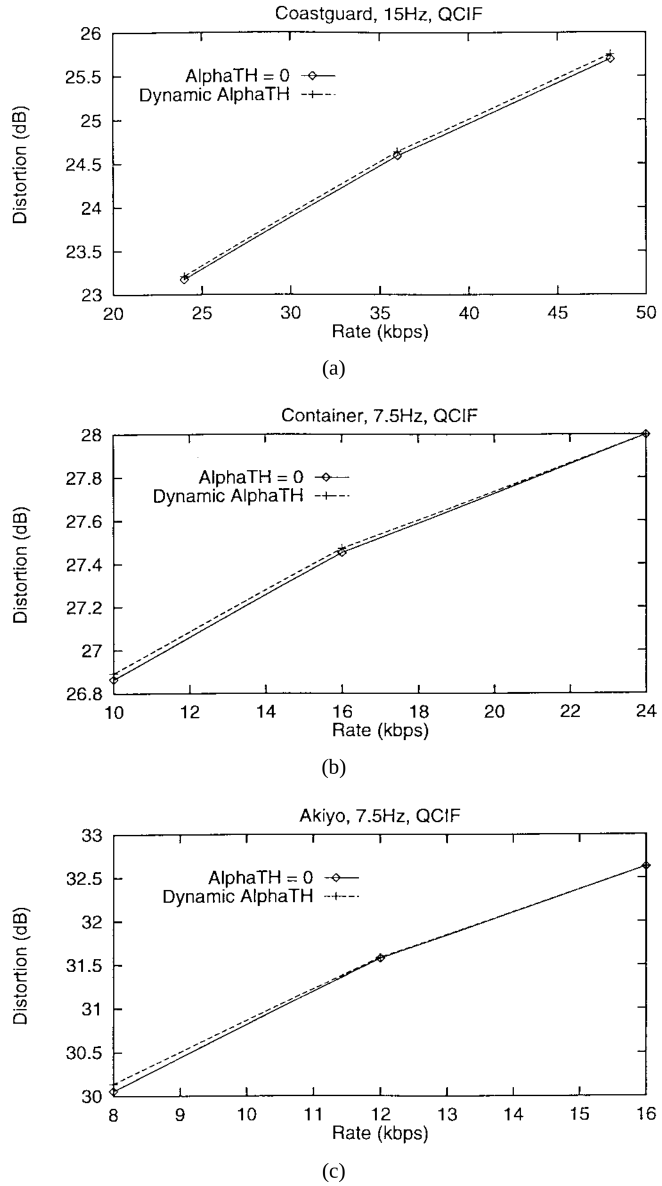
| AlphaTH | VO | Bit-Rates (kbps) | | | |
|---|---|---|---|---|---|
| | | 24 | 48 | 96 | 144 |
| 0 | 0 | 30.6 | 29.3 | 15.5 | 10.5 |
| | 1 | 30.6 | 22.0 | 11.2 | 7.6 |
| | 2 | 30.2 | 16.6 | 8.2 | 5.6 |
| | 3 | 30.5 | 26.3 | 14.6 | 10.2 |
| 16 | 0 | 30.5 | 29.2 | 15.0 | 10.4 |
| | 1 | 30.5 | 21.6 | 11.0 | 7.6 |
| | 2 | 30.1 | 16.0 | 8.1 | 5.6 |
| | 3 | 30.5 | 26.0 | 14.1 | 10.3 |
| 32 | 0 | 30.5 | 29.2 | 14.5 | 10.3 |
| | 1 | 30.3 | 20.9 | 10.8 | 7.4 |
| | 2 | 29.7 | 15.4 | 8.0 | 5.5 |
| | 3 | 30.5 | 25.9 | 14.3 | 10.2 |
| 64 | 0 | 30.5 | 28.9 | 16.6 | 11.4 |
| | 1 | 30.4 | 20.1 | 10.7 | 7.5 |
| | 2 | 30.2 | 15.5 | 7.9 | 5.6 |
| | 3 | 30.5 | 24.6 | 14.3 | 10.2 |

shape-coding control. For improved performance, one can apply more advanced interpolation techniques or postfiltering the decoded alpha plane, as proposed in [35] for compositing multiple objects.

### C. Performance of MVO3 with Dynamic Shape Rate Control

In the previous subsection, a detailed analysis of the object-based coding results with a fixed AlphaTH was presented. In this section, we compare the MVO3 which uses lossless shape coding and the MVO3 which uses the shape rate control algorithm proposed in Section IV-A. This is to evaluate the impact of the proposed shape rate control.

The *R–D* curves for the *Coastguard*, *Container*, and *Akiyo* sequences are plotted in Fig. 7. The simulations were performed on QCIF images at 15 Hz for *Coastguard* and 7.5 Hz for *Container* and *Akiyo*. Only the low bit-rate case is considered. At higher bit rates than the ones shown, the dynamic AlphaTH simulations performed exactly equal, i.e., lossless shape coding was automatically employed since the algorithm was always operating in HighMode. This convergence at higher bit rates is evident from the plots.



(a)

(b)

(c)

Fig. 7. *R–D* curves for *Coastguard, Container,* and *Akiyo* sequences. See text for explanation of PSNR calculation.

At the low bit rates, it is evident that choosing the AlphaTH dynamically incurs marginal improvements in terms of the PSNR values. Although the average gain in PSNR is small, a consistent increase in temporal resolution ranging from 3 to 5% was noted. This trend agrees with that observed in the fixed AlphaTH simulations, and is a visually notable improvement.

## VI. CONCLUDING REMARKS

In this paper, the problems associated with rate control for coding multiple video objects were addressed. This type of algorithm is useful in supporting the object-based functionalities which are central to the emerging MPEG-4 standard. A number of rate control tools have been proposed to provide a framework for efficient coding of multiple video objects at a wide range of bit rates and various spatial and temporal resolutions. The algorithm has been described in stages so that individual additions to the existing SVO scheme are justified. First, the necessary extensions for the SVO algorithm to operate within an object-based coder are discussed. The major addition was the target bit allocation among objects. Next, an improved buffering policy was introduced to handle the excess side information incurred by the shape information. Through preencoding stage analysis on the available bit rate after motion and shape coding and the buffer status, spatial and temporal coding resolutions are adjusted so that a stable buffer can be achieved. Lastly, we presented a mechanism for adapting the rate control parameters based on the mode of operation (which depends on the temporal coding resolution) and a scheme for dynamically adjusting the shape coding parameter. These assist the algorithm in adapting to different coding conditions, and achieve an appropriate tradeoff between spatial and temporal coding resolutions and between shape and texture distortion. Our results show that for low bit-rate simulations, moderate gains in temporal resolution can be achieved, while maintaining a similar same spatial coding quality. Overall, the algorithm does not experience any buffer overflow/underflow, and the video sequence is coded with reasonable quality.

As a general note, it should be mentioned that the rate control problem is simple when the texture bits comprise a large percentage of the total rate, i.e., high bit-rate condition. In this case, almost every frame is coded, the shape is coded without loss, and the bits generated by shape and motion can simply be considered overhead. Therefore, the major impact on quality is in the texture distribution of bits among each object. Since this is a common factor for all of the simulated algorithms (MVO1–MVO3), their performance is similar, and an optimization should focus on the distribution of total bits for texture among different objects. This topic has been studied in [30]. However, the situation changes drastically for low bit-rate coding conditions. A significant amount of complexity is added since the shape bits no longer comprise a small percentage of the total rate. Also, the need to skip frames to satisfy buffer constraints has emerged. In this case, the preframeskip control which is responsible for spatial and temporal adjustments becomes a necessary addition. For improved performance, especially the visual quality, mechanisms should be provided for trading off spatial coding accuracy for improved temporal resolution. Finally, distortion in shape should be allowed to achieve a good tradeoff between shape and texture coding accuracy. This is expected to provide large gains, and should serve as a focal point for optimization in the low bit-rate case. Our simulation results show that the shape rate control part does not have a significant impact on the coding efficiency in terms of the PSNR, however, a notable increase in temporal resolution, and consequently visual quality, can be gained.

As mentioned in Section I, the algorithm is not by any means optimal. However, we believe that the proposed framework will serve as a solid foundation for further performance improvement. Although MPEG-4 will be a standard in November 1998, encoder optimization will still be a very active topic. The proposed MVO scheme can achieve suitable tradeoffs between spatial and temporal coding resolutions, and between shape and texture coding. For improved performance, an algorithm should be devised to guarantee that optimal decisions are made or to verify that current decisions are near optimal. To arrive at these optimal or near-optimal decisions, it is necessary to have good $R$–$D$ models for describing the shape and texture of an object. Although the rate–quantizer model used in the present work is quite adequate for texture coding, effective $R$–$D$ models for shape still need to be developed. It is expected that such a model will depend on geometric attributes of the shape. In addition to individual $R$–$D$ models, a good understanding of the perceptual weighting for shape and texture distortion is also required to exercise joint shape and texture rate control. Also, some means to overcome the composition problem should be developed so that different objects can be encoded at different frame rates. Although this would require a more complex buffering scheme, the potential savings are enormous. Lastly, the rate control algorithm can take the responsibility to control the coding modes decisions, and jointly consider those decisions with the rest of the algorithm. This has been done with MPEG-2 [8] and H.263 [32]; preliminary results for MPEG-4 have been reported in [33] and [34]. Finally, the method of rate reduction for shape (size conversion of the alpha plane) is specific to MPEG-4, and the proposed shape-coding control is based on this method. Different control mechanisms will be needed for other lossy shape-coding methods that allow for rate–distortion control, e.g., the method in [31].

## ACKNOWLEDGMENT

The authors would like to thank the participants of core experiment Q2 for verifying the performance of their algorithm, especially T. Chiang and H. J. Lee of the Sarnoff Corporation for their useful discussions. They also thank the anonymous reviewers for providing insightful comments.

## REFERENCES

[1] J. Max, “Quantizing for minimum distortion,” *IRE Trans. Inform. Theory*, vol. IT-6, pp. 7–12, Mar. 1960.

[2] A. Segall, “Bit allocation and encoding for vector sources,” *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 162–169, Mar. 1976.

[3] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice Hall, 1984.

[4] 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.

[5] K. Ramchandran and M. Vetterli, “Best wavelet packet bases in a rate-distortion sense,” *IEEE Trans. Image Processing*, vol. 2, pp. 160–175, Apr. 1993.

[6] K. Ramchandran, A. Ortega, and M. Vetterli, “Bit allocation for dependent quantization with application to multi-resolution and MPEG video coding,” *IEEE Trans. Image Processing*, vol. 3, pp. 533–545, Sept. 1994.

[7] J. Lee and B. W. Dickenson, “Rate-distortion optimized frame type selection for MPEG encoding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 501–510, June 1997.

[8] H. Sun, W. Kwok, M. Chien, and C. H. J. Ju, “MPEG coding performance improvement by jointly optimizing coding mode decision

and rate control,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 449–458, June 1997.

[9] T. Weigand, 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 Technol.*, vol. 6, pp. 182–190, Apr. 1996.

[10] J. Choi and D. Park, “A stable feedback control of the buffer state using the controlled Lagrangian multiplier method,” *IEEE Trans. Image Processing*, vol. 3, pp. 546–558, Sept. 1994.

[11] J. J. Chen and D. W. Lin, “Optimal bit allocation for video coding video signals over ATM networks,” *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1002–1015, Aug. 1997.

[12] C. Y. Hsu, A. Ortega, and A. R. Reibman, “Joint selection of source and channel rate for VBR transmission under ATM policing constraints,” *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1016–1028, Aug. 1997.

[13] 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.

[14] W. Ding, “Joint encoder and channel rate control of VBR video over ATM networks,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 266–278, Apr. 1997.

[15] F. C. Martins, W. Ding, and E. Feig, “Joint control of spatial quantization and temporal sampling for very low bit rate video,” in *Proc. ICASSP*, vol. 4, May 1996, pp. 2072–2075.

[16] 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.

[17] A. Vetro and H. Sun, “Joint rate control for coding multiple video objects,” in *Proc. IEEE Workshop Multimedia Signal Processing*, Princeton, NJ, June 1997, pp. 181–186.

[18] T. Chiang and Y.-Q. Zhang, “A new rate control scheme using quadratic rate-distortion modeling,” *IEEE Trans. Circuits Syst. Video Technol.*, Feb. 1997.

[19] MPEG Test Model 5, ISO/IEC JTC/SC29/WG11 Document, Apr. 1993.

[20] A. Vetro and H. Sun, “Joint rate control for multiple video objects based on quadratic rate-distortion model,” ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG 97/M1631, Sevilla, Spain, Feb. 1997.

[21] MPEG-4 video verification model v8.0 ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG97/N1796, Stockholm, Sweden, July 1997.

[22] 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.

[23] P. Fleury, J. Reichel, and T. Ebrahimi, “Image quality prediction for bit rate allocation,” in *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, Oct. 1996, pp. 339–342.

[24] X. Marichal, T. Delmot, C. DeVleeschouwer, V. Warscotte, and B. Macq, “Automatic detection of interest areas of an image or of a sequence of images,” in *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, Oct. 1996, pp. 371–374.

[25] J. B. Lee and A. Eleftheriadis, “Spatio-temporal model-assisted compatible coding for low and very low bit rate video-telephony,” in *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, Oct. 1996, pp. 429–432.

[26] M. R. Pickering and J. F. Arnold, “A perceptually efficient VBR rate control algorithm,” *IEEE Trans. Image Processing*, vol. 3, pp. 527–532, Sept. 1994.

[27] L. Wang and A. Vincent, “Joint rate control for multi-program video coding,” *IEEE Trans. Consumer Electron.*, vol. 42, pp. 300–305, Aug. 1996.

[28] C. H. Lin and J. L. Wu, “Content-based rate control scheme for very low bit-rate video coding,” *IEEE Trans. Consumer Electron.*, vol. 43, pp. 123–133, May. 1997.

[29] H. J. Lee, T. Chiang, and Y. Q. Zhang, “Multiple-VO rate control and B-VO rate control,” ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG 97/M2554, Stockholm, Sweden, July 1997.

[30] J. I. Ronda, M. Eckert, S. Rienke, F. Jaureguizar, and A. Pacheco, “Advanced rate control for MPEG-4 coders,” in *Proc. Visual Commun. Image Processing (VCIP ‘98)*, SPIE, vol. 3309, Jan. 1998, pp. 383–394.

[31] G. M. Schuster and A. K. Katsaggelos, “An optimal boundary encoding scheme in the rate-distortion sense,” *IEEE Trans. Image Processing*, vol. 7, pp. 13–26, Jan. 1998.

[32] ——, “Fast and efficient mode and quantizer selection in the rate-distortion sense for H.263,” in *Proc. Visual Commun. Image Processing (VCIP ‘96)*.

[33] Y. K. Chen, A. Vetro, H. Sun, and S. Y. Kung, “Optimizing intra/inter coding mode decisions,” in *Proc. Int. Symp. Multimedia Inform. Processing*, Taipei, Taiwan, Dec. 1997, pp. 561–568.

[34] ——, “Optimizing $16 \times 16/8 \times 8$ coding mode decisions,” ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG 98/M3035, San Jose, CA, Feb. 1998.

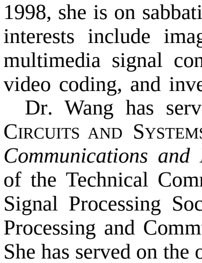
[35] C. S. Boon, S. Kadono, and J. Takahashi, “A blending method for objects composition,” ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG 98/M3161, San Jose, CA, Feb. 1998.

**Anthony Vetro** (S’92–M’96) simultaneously received the B.S. and M.S. degrees in electrical engineering from Polytechnic University, Brooklyn, NY, in June 1996. He is currently pursuing the Ph.D. degree at Polytechnic University.

From 1994 to 1996 he was a Teaching Assistant for the Department of Electrical Engineering at Polytechnic University. In 1996, he joined the Advanced Television Laboratory of Mitsubishi Electric Information Technology Center America, Inc., New Providence, NJ, where he is currently a Principal Member of the Technical Staff. His current research interests include digital video coding with emphasis on motion estimation and rate control, compressed-domain processing, motion-based segmentation, and multimedia signal processing.

**Huifang Sun** (S’83–M’85–SM’93) received the B.S. degree in electrical engineering from Harbin Engineering Institute, Harbin, China, in 1967 and the Ph.D. degree in electrical engineering from the University of Ottawa, Ottawa, Canada, in 1986.

From 1982 to 1986, he was with the Electrical Engineering Department at the University of Ottawa. In 1986, he joined Fairleigh Dickinson University, Teaneck, NJ, as an Assistant Professor and was consequently promoted to Associate Professor in Electrical Engineering. From 1990 to 1995, he was with the Sarnoff Corporation (formerly David Sarnoff Research Center), Princeton, NJ, as a Member of the Technical Staff and was consequently promoted to Technology Leader of Digital Video Technology where his activities were MPEG video coding, and Grand Alliance HDTV development. He joined the Advanced Television Laboratory, Mitsubishi Electric Information Technology Center America, Inc. in 1995 as a Senior Principal Technical Staff where his activity is advanced television development. In 1997, he was promoted to Deputy Director. He holds six U.S. patents and has several pending, and has authored or co-authored more than 60 journal and conference papers.



**Yao Wang** (S’86–M’90) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1983 and 1985, respectively, and the Ph.D degree in electrical engineering from the University of California at Santa Barbara in 1990.

Since 1990, she has been on the faculty of Polytechnic University, Brooklyn, NY, and is presently an Associate Professor of Electrical Engineering. From 1992 to 1996, she was a Consultant with AT&T Bell Laboratories, Holmdel, NJ. Since 1997, she continued as a Consultant with AT&T Labs Research, Red Bank, NJ. In 1998, she is on sabbatical leave at Princeton University. Her current research interests include image and video compression for unreliable networks, multimedia signal content analysis, motion estimation and object-oriented video coding, and inverse scattering for medical imaging.

Dr. Wang has served as Associate Editor for IEEE Transactions on Circuits and Systems for Video Technology and for *Journal of Visual Communications and Image Representation* since 1995. She is a member of the Technical Committee on Multimedia Signal Processing of the IEEE Signal Processing Society and the Technical Committee on Visual Signal Processing and Communications of the IEEE Circuits and Systems Society. She has served on the organizing/technical committees of several international conferences and workshops, and as Guest Editor for several special issues related to image and video coding.