A New Rate Control Algorithm for MPEG-4 Video Coding

Sun Yu and Ishfaq Ahmad

Department of Computer Science
Hong Kong University of Science and Technology
cssunyu@cs.ust.hk, iahmad@cs.ust.hk

ABSTRACT

This paper proposes a new MPEG-4 rate control algorithm for single or multiple object video sequences. The algorithm aims to achieve an accurate bit rate with the maximum picture quality while efficiently handling buffer fullness and scene change. In addition to estimating the bit budget of a frame based on its global coding complexity, the algorithm dynamically distributes the target bits for each object within a frame according to its coding complexity. Even though the VM8 solution and other algorithms adopt a simple proportional buffer controller, their control ability is rather ineffective. The proposed algorithm exploits a novel Proportional Integrated Differential (PID) buffer controller to effectively minimize the buffer overflow or underflow. The PID based controller reduces the deviation between the current buffer fullness and the target buffer fullness, mitigates the overshoots, and improves the transient response. The combined effect is a more smooth and effective buffer control. Furthermore, the algorithm defines a new and effective coding complexity of an object and dynamically optimizes several parameters. Overall, the proposed algorithm successfully achieves accurate target bit rate, provides promising coding quality, decreases buffer overflow/underflow and lowers the impact of a scene change.

Keywords: MPEG-4 video coding, rate control, bit allocation, multiple video objects, PID buffer control.

1. INTRODUCTION

MPEG-4, due to its affluent functions for supporting object-based high quality coding is at the forefront of the video compression technology and is becoming increasingly popular for present and emerging multimedia applications[1]. In MPEG-4 multimedia, a time-variable visual entity with an arbitrary shape can be individually manipulated and combined with other similar entities to produce a scene. The scene is compressed into a bitstream that can be transmitted through either constant or variable rate channels. To make the transmission as efficient and accurate as possible, a variety of coding factors should be jointly considered, for example, encoding rate, channel rate, and scene content, etc. This results in new research challenges in bit allocation and rate control schemes, which must satisfy a spectrum of application requirements.

Most visual communication applications use a fixed rate transmission channel, which means the encoder's output bit rate must be regulated to meet the transmission bandwidth. The rate controller of the encoder adjusts the quantization parameters (QPs) in order to meet the desired encoding bit rate for a source video. At the same time, the encoder must minimize the loss of the coding quality. The presence of multiple video objects exacerbates complexity of the encoding task as the rate controller must distribute bits among different objects according to the application requirements.

Typical rate controllers estimate the target bit-rate by measuring the buffer fullness. A buffer is placed between the encoder and the channel to smooth out the bit rate variation output from the encoder. The encoder generates bits and stores them in the buffer while the channel removes the bits from the buffer. When the source rate exceeds the transmission rate, the buffer temporarily stores the encoded bits so that they may be transmitted later allowing the encoding operation to continue. However, when the buffer is full, the encoder must cease generating bits by dropping frames thereby causing an interruption to the smoothness of the video. On the other hand, when the buffer is empty, the communication bandwidth is wasted and the coding quality is lower than its possible target. The buffer size is determined by the maximum delay allowed. A large buffer size tends to allow smoother video but causes longer delay, while a small buffer size guarantees low delay but may be more likely to skip frames due to overflow. Some rate control algorithms for MPEG-4 based encoding have been proposed in the past[2-7], for example, the rate control algorithms in VM8 of MPEG-4[8].

Chiang and Zhang have proposed a rate control algorithm that is scalable for various bit rates, spatial and temporal resolution, and can be applied to both DCT and wavelet-based coders[7]. This algorithm is based on a quadratic model that describes the relation between the required bits for coding the texture and the quantization parameter, the target bits of a frame is initially set to a weighted average of the number of bits used in previous frame and R/F. Vetro, Sun and Wang extended the R-D model to multiple object rate control[3], such the total target bits of a frame are distributed proportional to the relative size, motion and variance of each object. To provide a proper trade-off between spatial and temporal coding, the algorithm switches between a high rate coding mode and a low rate one. In the low rate mode, a mechanism to control the parameters for shape coding is included. Ronda, Eckert, Jaureguizar and Garcia focus on rate control for real-time applications[5]. Their algorithms rely on the modelization of the source and the optimization of a cost criterion based on signal quality parameters. Algorithms are introduced to minimize the average distortion of the objects, to guarantee desired qualities to the most relevant ones, and to keep constant ratios among the object qualities. Since their earlier work can only deal with single object rate control[7], Lee, Chang and Zhang extended it to multiple object rate control[2]. Nunes and Pereira presented a scene level and object rate control algorithm aiming at performing bits allocation for the several VOs composing a scene, encoded at different VOP rates[4].

Even though these algorithms can guarantee a relatively good coding performance, they are not efficient enough to simultaneously achieving the goals of an accurate target bit rate, high picture quality, avoiding buffer overflow/underflow, and admirably dealing with a scene change. Since MPEG-4 allows the coding of arbitrarily shaped objects, multiple objects and asynchronous VOP rate, the encoder must consider the significant amount of bits that are used to code the shape information, bits allocation among multiple objects, bits allocation for each time slot, etc. This paper proposes a new MPEG-4 rate control algorithm called Re-adjusting Adaptive with Proportional Integrated Differential (RAPID). The algorithm aims to achieve an accurate bit rate with the maximum picture quality while at the same time handling buffer fullness and scene change. The specific characteristics of the algorithm include: (a) In addition to estimating the bit budget of a frame based on its global coding complexity, the algorithm dynamically distributes the target bits for each object within a frame according to its coding complexity; (b) The algorithm exploits a Proportional Integrated Differential (PID) buffer controller to effectively minimize the buffer overflow or underflow; (c) The algorithm defines a new and effective coding complexity of an object; (d) The algorithm proposes several adaptation methods to automatically optimize parameters.

The remainder of this paper is organized as follows: In Section 2, we describe the basic philosophy of the proposed adaptive rate control algorithm for single/multiple video object. In the same section, we discuss a new buffer control method named PID controller to maintain a stable buffer level. In Section 3, we present some optimization methods that further fine tune the efficiency of the proposed algorithm. Section 4 summarizes the algorithm and describes its functionality. Section 5 includes the experimental results that demonstrate the performance of the proposed algorithm. Finally, Section 6 concludes the paper by providing some final remarks and observations.

2. FOUNDATIONS OF THE PROPOSED ALGORITHM

The proposed rate control algorithm consists of a number of steps. In this section, we describe the principles and foundations of these steps.

2.1. Initialization Stage

The initialization stage includes setting up of the encoding parameters and buffer size. The buffer size is initialized based on latency requirement, while the buffer fullness is initialized as the middle level of the buffer size. We assume that the required bit rate is constant, multiple VOs are synchronous with the same VOP rate, and a frame is defined as a set of VOPs of different objects with a common presentation time. The total target number of bits generated by the encoder during t_G is:

$$T_G = bit\_rate \times t_G \,.$$

The maximum number of VOPs that can be encoded during t_G is:

$$N_G = VOP\_rate \times t_G \,.$$

Thus, the numbers of I-VOPs, P-VOPs and B-VOPs in the given sequence during t_G can be computed by:

$$N_{IVOP} = Int\left(\frac{N_G + L}{L+1}\right), \quad N_{PVOP} = Int\left(\frac{N_G + K}{K+1} - N_{IVOP}\right), \quad N_{BVOP} = Int\left(\left(\frac{N_G + K}{K+1} - 1\right) \times K\right),$$

where L is the number of VOPs between two consecutive I-VOPs, K is the number of B-VOPs between two consecutive P-VOPs or P-VOP and I-VOP. Furthermore, we should know the weighted average number of bits to be output from the buffer per frame:

$$B_p = \alpha_K \times \frac{R_r}{(\alpha(I) \times N_I + \alpha(B) \times N_B + \alpha(P) \times N_P)},$$

where N_I, N_P and N_B are the numbers of I-VOPs, P-VOPs and B-VOPs which remain to be coded respectively, $\alpha(I), \alpha(B)$ and $\alpha(P)$ are their weight factors, R_r is the total number of bits available for the rest of the image sequence, α_k is $\alpha(I)$, $\alpha(B)$ or $\alpha(P)$ corresponding to the coding type of current VOP.

2.2 Initial Target Bit Estimation

Based on available bits, the perceptual efficient approach, the past history of each VO and the current time instant characteristics (coding complexity), a combination of strategies is used to estimate the initial target bits[2-5, 9].

A coding complexity of VOP_i at time t to be encoded is calculated according to the following formula:

$$C_{i,t} = \sum_{i=1}^{NVO_i} \left(\frac{\sum_{j=1}^{n_i} (P_{ij} - \bar{P}_i)^2}{n_i} \right)^{1/4} \quad \text{with} \quad \bar{P}_i = \frac{1}{n_i} \times \sum_{j=1}^{n_i} P_{ij}, \tag{1}$$

where P_{ij} is the luminance value of pixel j in the i^{th} Marco-Block MB_i of a motion-compensated residual VOP_i, $\overline{P_i}$ is the arithmetic average pixel value of MB_i, n_i is the number of non-transparent pixels in the MB_i, NVO_i is the number of non-transparent macro-blocks in the VOP_i.

The coding complexity computed by (1) naturally combines the object size (NVO_i) and average variance of each macro-block in a VOP, and, therefore, can reflect the instantaneous characteristics of this VOP. The coding complexity dictates how many bits can be appropriate for VOPs before really encoding them. This is specially useful when a VO changes its features rapidly, or when a scene change[10,11] happens, because the coding complexity of the VOP can reflect these changes. In the VM8 solution of MPEG-4, target bits are allocated to the current frame only according to the statistical information of its previous frame, without any consideration to the real complexity of the current frame. This may result in inappropriate allocation of bits to the current frame, which can lead to fluctuating and overall degraded visual quality. A global complexity of current frame at time t can be obtained by:

$$C_{G,t} = \sum_{i=1}^{M} (NW_{i,t} \times C_{i,t}),$$

where $C_{i,t}$ denotes the coding complexity of VOP_i in the current frame, $NW_{i,t}$ is the normalized weight of VOP_i which will be discussed later. M is the number of VOs in the current frame. The average global complexity of previous n frames before time t can be computed by:

$$C_{ave,t} = \frac{\sum_{n} C_{G,i}}{n}, \qquad (i = t, t\text{-}1, \ldots, t\text{-}n\text{+}1).$$

According to the type of the current frame, its target number of bits is initially set to a weighted average bitcount:

$$T_{ave,t} = \alpha_k \times \frac{R_r}{\alpha(I) \times N_I + \alpha(B) \times N_B + \alpha(P) \times N_P}.$$

The total target bit budget of the current frame to be encoded is then estimated by:

$$T_t = T_{ave,t} \times \frac{C_{G,t}}{C_{ave,t}}, \tag{2}$$

where α_k is $\alpha(B)$ or $\alpha(P)$ according to the coding type of the current VOP.

The number of target bits is estimated only for P-VOPs and B-VOPs for each time instant. We do not estimate target bits for I-VOPs, which will be explained later. This bits allocation essentially follows a basic principle: if $C_{G,t}$ is higher than $C_{ave,t}$, more bits should be allocated to the current frame than the weighted average bits $T_{ave,t}$; on the contrary, if $C_{G,t}$ is lower than $C_{ave,t}$, fewer bits should be allocated. Hence, appropriate bits can be adaptively allocated to the current frame and coding quality can be kept constant.

2.3 Target Bits Adjustment Based on the Buffer Occupancy

The initial bit target is further refined based on the buffer fullness to get a more accurate target bit estimation. The goal of the buffer control is try to keep buffer fullness in the middle level to reduce chances of buffer overflow or underflow: if buffer occupancy exceeds the middle level, the target bits are decreased to some extent; similarly, if buffer occupancy is below the middle level, the target bits are increased a little. Even though the VM8 solution and other algorithms adopt a simple proportional buffer controller, their control ability is rather ineffective. As shown in our experiment, when the complexity of a sequence changes drastically, the buffer trends to be out of control, especially in the low bit rate cases.

The proportional action can reduce the deviation between the current buffer fullness and the target buffer fullness (typically, middle level), but cannot fully eliminate this deviation. An Integral Controller has the effect of eliminating the deviation by this way: when the deviation lasts, it can automatically enhance the control strength. But it may make the transient response worse. A Differential Controller has the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. The three-mode Proportional-Integral-Differential (PID) controller[12,13] (see Figure 1) combines the advantages of each individual controller and thus, is more smooth and effective. Here we apply this technique to our buffer control problem. It can keep buffer fullness around the middle level and significantly reduce the chances of buffer overflow or underflow.

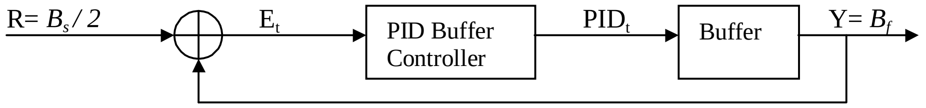


Figure 1: The PID buffer Control System

The variable E_t represents the error signal (deviation) at time t, the difference between the desired value R (half level of buffer) and the actual output Y (buffer occupancy) at time t, is defined as:

$$E_t = \frac{(B_s/2 - B_f)}{B_s/2},$$

where B_s is the buffer size, B_f is the current buffer fullness at time t. This error signal E_t is sent to the PID controller:

$$PID_t = K_p \times (E_t + K_i \times \int E_t \cdot dt + K_d \times \frac{dE_t}{dt}), \tag{3}$$

where K_p, K_i and K_d are the Proportional, Integral and Differential control parameter respectively. In the experiments, K_p, K_i, and K_d have been set to 1.0, 0.15 and 0.2 respectively for multiple object coding; and to 1.0, 0.25 and 0.3 respectively for single object coding. Then the initial target bits can be further adjusted by:

$$T_t = T_t \times (1 + PID_t)\,. \tag{4}$$

To maintain a minimum acceptable visual quality, the encoder must allocate a minimum number of bits to the current frame, that is:

$$T_t = \max\{\frac{R}{4 \times F}, T_t\}\,.$$

Similarly, to avoid buffer overflow, a maximum number of bits is given to it:

$$T_t = \min\left\{\frac{2 \times R}{F}, T_t\right\},$$

where R and F are the bit rate and frame rate required by the application.

2.4 Dynamic Target Bits Distribution among Multiple VOs

In order to maximize the overall quality of the decoded scene with a given amount of resources, it is important to effectively distribute the total target bits among multiple objects for a frame[5,14]. Normally, a rate control scheme should allocate more bits to important VOs (e.g., foreground VO) than other areas (e.g., background VOs). Visual quality should be bad if improper bits were allocated to VOs. For example, the background VOs may have excellent quality, while the foreground VOs may have low quality, or there may be unbalanced qualities among VOs. The proposed algorithm distributes the bit budget at time t for VOP_i according to the coding complexity in the following manner:

$$T_{i,t} = \frac{NW_{i,t} \times C_{i,t}}{C_{G,t}} \times T_t , \tag{5}$$

where $T_{i,t}$ represent the target bits allocated to VOP_i at time t.

2.5 Quantization Parameter Calculation

The quantziation Parameter (QP) for texture encoding is computed based on the Rate Distortion model of each VO for the corresponding VOP coding type[3,15]. Once the number of target bits $T_{i,t}$ for VOP_i is obtained, the number of target bits for coding the texture of the i^{th} object can be computed by :

$$T_{texture,i} = T_{i,t} - H_{i,t-1} ,$$

where $H_{i,t-1}$ denotes the number of bits actually used for coding the motion, shape and header for VOP_i at time t-1. $T_{texture,i}$ represents the target bits to encode texture information of VOP_i. The proposed rate control algorithm also adopt this Rate-Distortion Model[2,3]:

$$T_{texture,i} = \frac{X1_i \cdot MAD_i}{Q_i} + \frac{X2_i \cdot MAD_i}{Q_i^2} , \tag{6}$$

where MAD_i is computed using motion-compensated residual for the luminance component, Q_i denotes quantization level used for VO_i, $X1_i$ and $X2_i$ is the first and second order model coefficients.

One problem of VM8 is that Intra coded VOPs are typically encoded with lower quality than Inter coded VOPs, this result in a larger quality variations and quality decay. It indicates that the bit allocation strategy of VM8 is not very efficient. The partial reason is explained as follows: A good coding quality depends on an accurate R-D model, and the accuracy of R-D model depends on the quality and quantity of the data set used to update it. Generally speaking, more updating data points in a coding process are likely to yield a more accurate model to reflect the video contents. At the beginning of the coding process, the R-D models of all types of VOPs are very rough. Along with the coding process, more and more VOPs are selected to update these R-D models and R-D models become more and more accurate than the original ones. Though this adaptive procedure is truly successful for P-VOPs and B-VOPs, it is not very suitable for updating I-VOPs' R-D model simply because I-VOPs are quite sparse in a coding sequence. Even enough quantity of I-VOPs can be accumulated after many coded I-VOPs, most of them cannot represent the change of the coming I-VOP. Thus the R-D model of I-VOPs is less accurate than those of the inter-coded VOPs and, thus, the coding quality of I-VOPs trends to fluctuate. To avoid the above problem and achieve a constant coding quality between Intra coded VOPs and Inter coded VOPs, a novel way is adopted here: We only estimate the number of target bits and calculate QPs for B-VOPs and P-VOPs but not for I-VOPs. Instead, when coding an I-VOP, we just employ the average QP of its previous l Inter coded VOPs with some adjustment. Though this method is quite simple, it is very efficient to overcome visual quality fluctuation or degradation of I-VOPs. The QP is limited to vary between 1 and 31. To smooth quality fluctuation, QP is only allowed to change within 25% of the previous QP.

2.6 Encoding and Updating

After encoding video objects within a frame, the encoder updates the R-D model of each VO for the corresponding VOP coding type based on the encoding results of the current objects as well as the past objects. Previous QPs and corresponding texture bit counts are used in the R-D model updating. The first and second model parameters, $X1_i$ and $X2_i$, are solved by using linear regression technique[2,7]. Other parameters' adaptation is described in the next section.

2.7 Frame-Skipping Control

To effectively avoid buffer overflow, the encoder needs to examine the current buffer fullness before encoding the next frame: If the buffer occupancy exceeds 80 percentage of the buffer size, the encoder skips the encoding of the next frame, and the buffer fullness is updated by the channel output rate. Since frame skipping can significantly reduce the overall perceptual quality, a good rate control algorithm should avoid frame skipping as best as it can.

3. OPTIMIZATION OF THE RATE CONTROL PARAMETERS

To further improve the system performance, some coding parameters should be considered and dynamically adjusted in the coding process. This section describes these techniques.

3.1 Weight Adjustment for VOP Types

$\alpha(I)$, $\alpha(B)$ and $\alpha(P)$ are weights of I-VOP, B-VOP and P-VOP, respectively; their initial values are set to 3.0, 0.5 and 1.0, respectively. To achieve a smooth visual quality, $\alpha(I)$ and $\alpha(B)$ are updated based on coded I- and B-VOPs, while $\alpha(P)$ is fixed to 1.0. In principle, if the average coding quality of previously coded B-VOPs (B_{PSNR}) is lower than that of previous coded P-VOPs (P_{PSNR}), we increase $\alpha(B)$ by a small amount. Then B-VOP to be coded next time can be allocated more bits, and thus improve its quality gradually to keep consistent with the quality of P-VOPs. On the contrary, if the average PSNR of the coded B-VOPs is higher than that of the coded P-VOPs, we decrease $\alpha(B)$ by a small amount to get fewer target bits for the next B-VOP, thus decrease its coding quality gradually to keep close to PSNRs of P-VOPs.

$$\alpha(B) = \frac{B_{avebits}}{P_{avebits}} \times e^{\left(\frac{P_{PSNR} - B_{PSNR}}{\gamma}\right)}, \tag{7}$$

where $P_{avebits}$ and $B_{avebits}$ denote the average number of bits used in coding previous n_P P-VOPs and n_B B-VOPs, respectively; P_{PSNR} and B_{PSNR} are their average PSNRs; $\gamma = 8$, which is determined by experiments. Similarly, $\alpha(I)$ is also updated by:

$$\alpha(I) = \frac{I_{avebits}}{P_{avebits}} \times e^{\left(\frac{P_{PSNR} - I_{PSNR}}{\gamma}\right)}. \tag{8}$$

For the reason to keep stability and rapidly reflect the influence of scene variations, (n_I + n_P + n_B) should not be too short or too long. Here a length of 30 frames is chosen to make a compromise to calculate the average values.

3.2 Weight Adjustment among Multiple Objects

Similarly, to achieve comparable and balanced quality among multiple objects within a frame, or in other words, to avoid large perceptual quality differences among multiple objects, weight for each object is further adjusted according to the PSNR difference of previous coded VOPs. $PSNR_{i,t-1}$ of VO_i (i=2..M) is compared to the $PSNR_{1,t-1}$ of VO_1, if $PSNR_{i,t-1}$ is lower than $PSNR_{1,t-1}$, the weight of the VO_i at time t, $W_{i,t}$, is increased a little, thus VO_i obtains more target bits and thus achieves a higher quality; otherwise, $W_{i,t}$ is decreased a little and achieves lower quality. We initialize $W_{i,0}$ to 1.0 for all VO_i and adopt the first object as a referential base, then the weights of other objects are updated:

$$W_{i,t} = W_{i,t-1} \times e^{\left(\frac{PSNR_{1,t-1} - PSNR_{i,t-1}}{\theta}\right)} \qquad \text{for } i > 1, \tag{9}$$

where θ = 16, which is determined by experiments. Note, W_1=1.0 forever. Then the normalized weights for all objects are calculated by:

$$NW_{i,t} = \frac{W_{i,t}}{\sum_{j=1}^{M} W_{j,t}}.$$

Obviously, a further improvement could be easily made to provide different prior levels for VOs:

$$W_{i,t} = W_{i,t-1} \times e^{\left(\frac{PSNR_{1,t-1} - PSNR_{i,t-1} + P_i}{\theta}\right)} \qquad \text{for } i > 1, \tag{9a}$$

where P_i is the priority of VO_i. P_i>0 (dB) means a higher priority while P_i<0 (dB) corresponds to a lower priority. For example, if one likes the foreground object VO_2 to have a PNSR 3 dB higher than that of the background object VO_1, one can set P_1=0.0 and P_2=3.0.

3.3 Quantization Parameter Updating for I-VOP

Since QP of I-VOP for an object is obtained directly by averaging QPs of previous *l* inter coded VOPs, to better maintain the consistent quality between I-VOP and its previous inter coded VOPs, balance adjustment is applied as following:

$$QP_{I,i} = QP_{ave,i} + \beta\_I, \tag{10}$$

where $QP_{I,i}$ is the QP of $I\text{-}VOP_i$; $QP_{ave,i}$ is average QP of *l* inter coded VOPs before $I\text{-}VOP_i$; initially, β_I = 1.0 and is updated as follows:

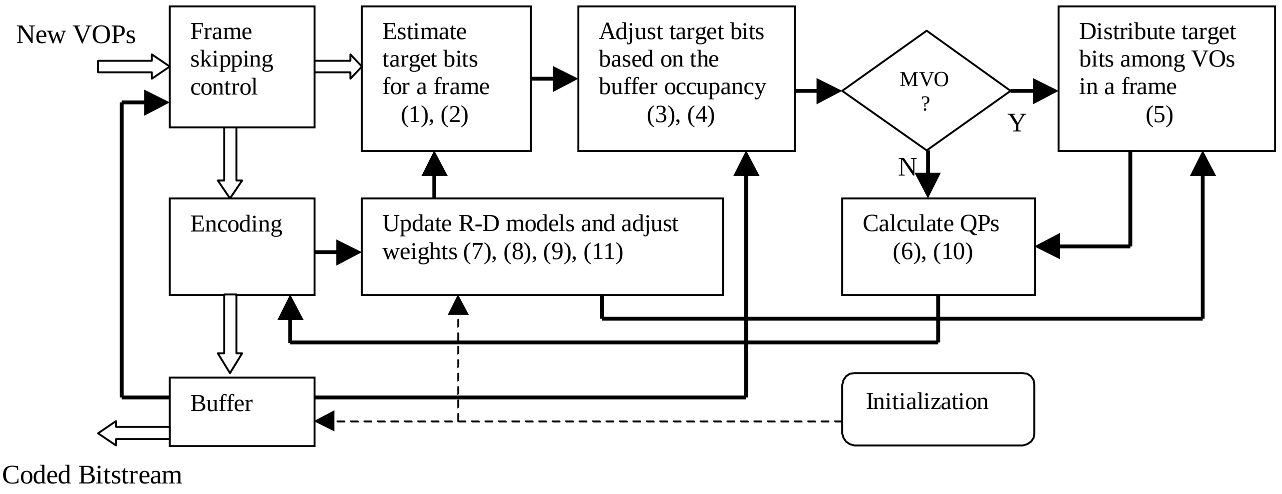
$$\beta\_I = \beta\_I + \frac{PSNR_{I,j} - PSNR_{ave.j}}{\lambda}, \tag{11}$$

where $PSNR_{I,j}$ is the PSNR of last $I\text{-}VOP_j$ and $PSNR_{ave,j}$ is the average PSNR of *l* inter coded VOPs before last I-VOP_j; *l*=3 in the experiment; λ is 4 for single object and 16 for multiple object. The is because if an I-VOP's PSNR is higher than the average PSNR of its previous *l* inter coded VOPs, The QP for I-VOP should be increased in order to lower its coding quality. Otherwise, if the PSNR of an I-VOP is lower than the average PSNR of *l* inter coded VOPs, The QP of I-VOP's should be decreased in order to increase its coding quality. This adjusts the quality of I-VOP to be closer to those of its previous inter coded VOPs.

4. THE RAPID Rate Control Algorithm

Here, we summarize the previous sections as the RAPID algorithm. The algorithm has the following steps:

1) Initialize the parameters for the encoder.
2) Estimate the number of target bits for a frame using Equation (1), (2).
3) Adjust target bits for a frame based on the buffer occupancy using Equation (3), (4).
4) Distribute target bits among multiple VOs in a frame using Equation (5).
5) Calculate the Quantziation Parameter using Equation (6), (10).
6) Encode frame/objects.
7) Update R-D Model and adjust other parameters using Equation (7), (8), (9), (11).
8) Apply frame-skipping control, if necessary.



⇨ indicate data flow, ➔ are control flow and , - - ▶ are just used in initialization.

Figure 2: The functional diagram of RAPID.

5. EXPERIMENTAL RESULTS

This section presents the performance of the proposed RAPID algorithm. We conducted two sets of experiments: one for encoding a single object with rectangular or arbitrary shape, and the second for encoding multiple objects. The results achieved here are compared with those achieved using the VM8 rate control algorithm suggested by the MPEG-4 visual standard. Since a skipped VOP is represented in the decoded sequence by repeating the previously coded VOP according to MPEG-4 core experiments, the PSNR of a skipped VOP is computed by using the previous encoded VOP[5,16]. It is obvious that the PSNR of a skipped VOP is typically much lower than that of a normal one.

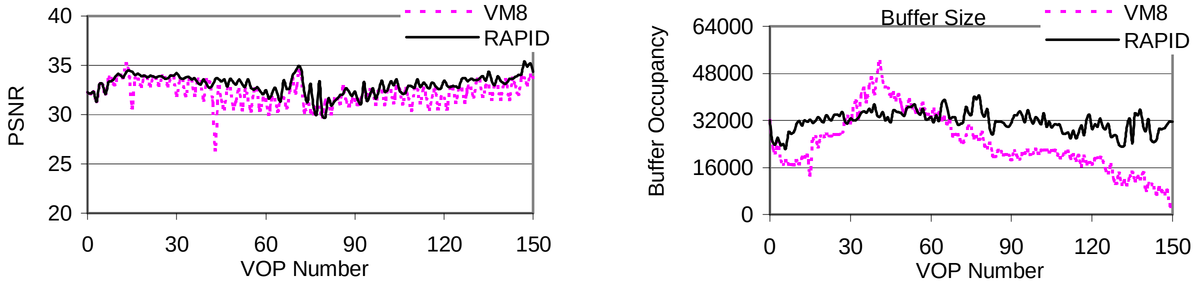
5.1 Single Object Rate Control

The results of encoding various testing sequences using I-VOP, P-VOP and B-VOP for one rectangular or arbitrary shape VO are reported in Table 1. For instance, Figure 3a and 4a illustrate PSNR curves and Figure 3b and 4b show the corresponding buffer occupancy curves for two sequences respectively.

In these experiments, the Intra period is set to one second; the number of B-VOPs is set to 2 between two P-VOPs or between I-VOP and P-VOP; the number of P-VOPs is set to 4 between two I-VOPs. The initial values of $\alpha(I)$, $\alpha(B)$ and $\alpha(P)$ are 3.0, 0.5, and 1.0, respectively; the values of $\alpha(I)$ and $\alpha(B)$ are dynamically adjusted during the encoding process. All sequences are encoded at 15 frames/sec (fps). Each sequence in Table 1 is tested using a relatively higher bit-rate and a lower bit-rate.

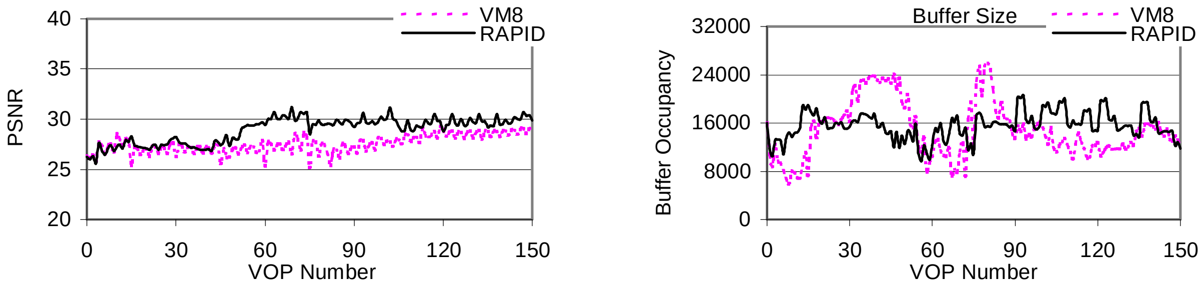
Table 1: Single VO rate control using I-VOPs, B-VOPs, and P-VOPs.

Video Sequence	Algorithms	Bit Rate (Kbps)		# Coded VOPs		PSNR (dB)
		Target	Actual	Target	Actual	
Coastguard (qcif)	VM8	64	64.66	150	145	28.87
	RAPID	64	63.85	150	150	30.34
Coastguard (qcif)	VM8	128	141.54	150	149	32.25
	RAPID	128	127.40	150	150	33.08
Cotainer (cif)	VM8	128	127.77	150	136	31.85
	RAPID	128	127.27	150	150	33.20
Cotainer (cif)	VM8	192	196.14	150	139	33.31
	RAPID	192	190.44	150	150	34.72
Cotainer (cif)	VM8	512	533.78	150	145	37.81
	RAPID	512	507.92	150	150	39.63
Bream2_1 (qcif)	VM8	64	65.37	150	145	27.73
	RAPID	64	64.09	150	150	28.71
Bream2_1 (qcif)	VM8	192	194.85	150	146	35.24
	RAPID	192	191.91	150	150	36.11
Silent (qcif)	VM8	64	65.08	150	137	31.76
	RAPID	64	63.36	150	150	34.04
Silent (qcif)	VM8	128	128.49	150	142	34.92
	RAPID	128	128.13	150	150	37.70
Silent (qcif)	VM8	180	166.32	150	149	38.18
	RAPID	180	173.62	150	150	39.15
News (qcif,)	VM8	64	63.47	150	143	31.52
	RAPID	64	63.84	150	150	34.27
News (qcif)	VM8	128	129.70	150	143	35.67
	RAPID	128	126.32	150	150	39.11
Mobile (qcif)	VM8	128	126.50	150	148	25.86
	RAPID	128	127.02	150	150	27.34
Mobile (qcif)	VM8	384	379.69	150	147	31.15
	RAPID	384	383.93	150	150	32.82
Train_&_T_R (qcif)	VM8	64	66.69	150	149	27.56
	RAPID	64	63.16	150	150	28.92
Train_&_T_R (qcif)	VM8	256	274.13	150	149	36.23
	RAPID	256	255.97	150	150	37.11



(a) PSNR Curves (b) Buffer Occupancy

Figure 3: The results for the Coastguard sequence (QCIF) encoded at 128 kbps, 15fps (IBBP…IBBP).



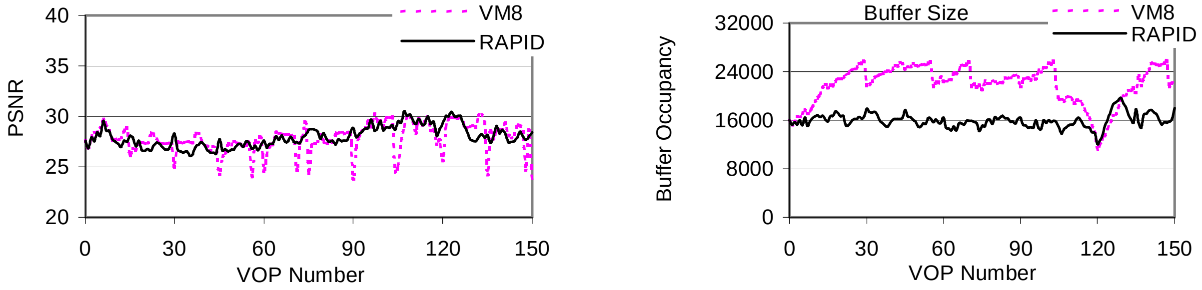
(a) PSNR Curves (b) Buffer Occupancy

Figure 4: The results for the Train_&_Tunnel_Right sequence (QCIF) encoded at 64 kbps, 15fps (IBBP…IBBP).

Table 2: Single VO rate control, only I-VOPs and P-VOPs are used in coding.

Video Sequence	Algorithms	Bit Rate (Kbps)		# Coded VOPs		PSNR (dB)
		Target	Actual	Target	Actual	
Coastguard (qcif)	VM8	64	64.44	150	144	29.29
	RAPID	64	63.78	150	150	29.61
Coastguard (qcif)	VM8	128	128.70	150	147	31.97
	RAPID	128	127.64	150	150	32.24
Cotainer (cif)	VM8	192	191.71	150	131	32.67
	RAPID	192	190.93	150	150	34.38
Cotainer (cif)	VM8	512	507.01	150	142	37.72
	RAPID	512	511.02	150	150	38.58
Bream2_1 (qcif)	VM 8	64	64.26	150	145	27.88
	RAPID	64	64.14	150	150	27.97
Bream2_1 (qcif)	VM8	192	193.62	150	147	35.05
	RAPID	192	192.12	150	150	35.30
Silent (qcif)	VM8	64	63.98	150	131	33.00
	RAPID	64	63.64	150	150	34.05
Silent (qcif)	VM8	128	127.92	150	136	36.67
	RAPID	128	126.99	150	150	38.26
Silent (qcif)	VM8	180	175.45	150	150	39.70
	RAPID	180	176.55	150	150	40.22
News (qcif)	VM8	64	63.66	150	131	32.58
	RAPID	64	63.77	150	150	34.18
News (qcif)	VM8	128	128.85	150	135	37.10
	RAPID	128	128.12	150	150	38.73
Mobile (qcif)	VM8	128	128.74	150	145	25.45
	RAPID	128	127.94	150	150	25.75
Mobile (qcif)	VM8	384	383.72	150	150	30.67
	RAPID	384	383.63	150	150	30.86
Train_Right (qcif)	VM8	64	64.84	150	140	28.27
	RAPID	64	63.76	150	150	28.84
Train_Right (qcif)	VM8	256	256.83	150	146	35.82
	RAPID	256	254.85	150	150	36.67

Table 2 shows the encoding results of single VO which only I-VOPs and P-VOPs are used. Figure 5a and 5b shows PSNR curves and buffer curves for sequence Bream2_1 respectively. The Intra period is set to one second. Initially, $\alpha(I)$=3.0 and $\alpha(P)$=1.0.



(a) PSNR Curves

(b) Buffer Occupancy

Figure 5: The results for the Bream2_1 sequence (QCIF) encoded at 64kbps, 15fps without using B-VOPs (IP…IP).

By examining the results in Table 1 and Table 2, it is obvious that the RAPID achieves more accurate target bit rate and target frame rate with higher average PSNR as compared to the VM8 solution. From Figure 3a, 4a and 5a, we observe that in the VM8 algorithm, intra coded VOPs typically have lower qualities than inter coded VOPs or there are large fluctuations between them, indicating a less efficient bit allocation strategy. From Figure 3b, 4b and 5b, one can see the buffer occupancy curves of RAPID are quite stable; they are around 50% of the buffer size with a small variation. However, by examining the buffer occupancy curves produced by VM8, it is evident that VM8 has less control ability and results in more frame skipping cases.

5.2 Multiple Object Rate Control

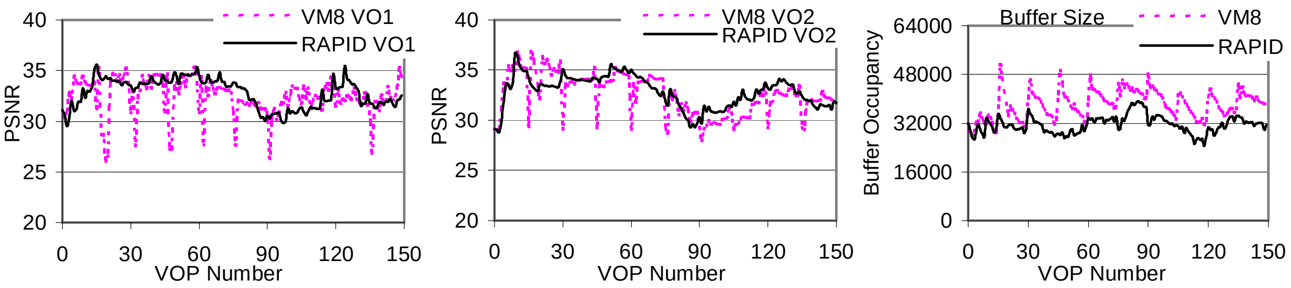
The results for multiple VO encoding are shown in Table 3. The Intra period is set to 0.5 second, and B-VOP is not used. Initially, $\alpha(I)$ = 3.0 and $\alpha(P)$ = 1.0. $\alpha(I)$ is updated during the encoding process. All sequences are QCIF format and are encoded at 30 fps.

With the same conditions, the VM8 solution skips much more frames than RAPID (see Table 3), indicating that its buffer control ability is relatively less efficient and bit allocation is not very accurate. This is crucial for low bit rates where the bit resources are scarce. These results also show that quality differences among VOs of RAPID are smaller than those of VM8, this illustrates the merit of the proposed automatic adaptation methodology.

Figure 6 and 7 show the PSNR and buffer fullness curves for RAPID and VM8. Note that, large PSNR degradations of I-VOPs exist in VM8 solutions, which cause quality fluctuation. The buffer fullness of RAPID is around 50% of the buffer size with a smaller variation, thus is more stable than the VM8 solution.

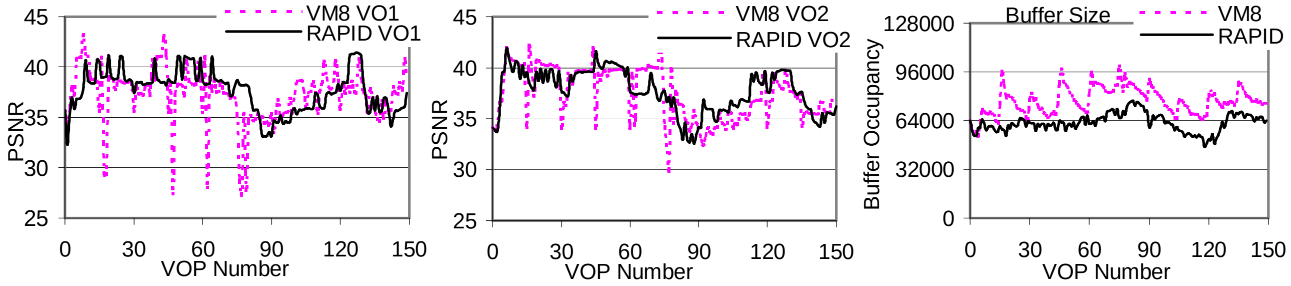
Table 3: Multiple object rate control, both I-VOP and P-VOP are adopted.

Video Sequence	Algorithms	Bit Rate (Kbps)				# Coded VOPs		PSNR(dB)	
		Target	Actual	VO1	VO2	Target	Actual	VO1	VO2
News_1 (Ballet) News_2 (Speakers)	VM8	128	130.13	56.09	74.04	150	140	32.42	32.54
	RAPID	128	127.97	56.50	71.47	150	150	32.78	32.84
News_1 (Ballet) News_2 (Speakers)	VM8	256	259.36	124.07	135.29	150	143	37.25	37.48
	RAPID	256	255.52	124.07	131.45	150	150	37.61	37.68
Bream2_0 (Background) Bream2_1	VM8	128	130.38	30.35	100.03	150	143	40.82	26.34
	RAPID	128	127.85	13.27	114.58	150	150	38.03	27.17
Bream2_0 (Background) Bream2_1	VM8	256	260.82	51.80	209.02	150	145	43.24	30.71
	RAPID	256	255.59	14.96	241.63	150	150	38.64	31.81
Children2_1 Children2_2	VM8	192	192.81	136.49	56.32	150	145	26.21	31.31
	RAPID	192	191.81	156.26	35.55	150	150	27.43	29.27
Children2_1 Children2_2	VM8	384	384.70	291.49	92.91	150	146	31.63	34.95
	RAPID	384	383.62	310.03	73.59	150	150	31.79	34.90



(a) PSNR curves of VO1. (b) PSNR curves of VO2. (c) Buffer Occupancy.

Figure 6: The results for the News sequence (QCIF) with 2VOs encoded at 128 kbps, 30fps (IP…IP).



(a) PSNR curves of VO1. (b) PSNR curves of VO2. (c) Buffer Occupancy

Figure 7: The results for the News sequence (QCIF) with 2VOs encoded at 256 kbps, 30fps (IP…IP).

The results given in Table 4 are under a special condition that only first VOP is I-VOP and the remaining VOPs are all P-VOPs. This is the simplest case in rate control. All sequences are QCIF format and are encoded in 30 fps. The results in Table 4 also indicate that the performance of RAPID is better than or at least equal to the VM8 solution.

Table 4: Multiple object rate control, only P-VOPs are used in test sequences except first I-VOP.

Video Sequence	Algorithms	Bit Rate (kbps)				# Coded VOPs		PSNR(dB)	
		Target	Actual	VO1	VO2	Target	Actual	VO1	VO2
News_1 (Ballet)	VM8	128	128.51	63.83	64.68	150	150	33.53	34.87
News_2 (Speakers)	RAPID	128	128.87	72.02	56.85	150	150	34.22	34.29
News_1 (Ballet)	VM8	256	257.83	142.94	114.89	150	150	38.69	39.10
News_2 (Speakers)	RAPID	256	257.94	147.82	110.12	150	150	38.87	38.94
Bream2_0 (Background)	VM8	128	128.24	26.73	101.51	150	150	42.34	27.08
Bream2_1	RAPID	128	127.81	9.11	119.18	150	150	38.71	27.94
Bream2_0 (Background)	VM8	256	257.67	49.81	207.43	150	150	44.07	31.24
Bream2_1	RAPID	256	255.59	10.80	245.79	150	150	40.25	32.34
Children2_1	VM8	192	192.29	143.38	48.91	150	144	27.11	34.16
Children2_2	RAPID	192	192.00	171.40	20.60	150	150	28.43	29.68
Children2_1	VM8	384	384.29	296.08	88.21	150	148	32.44	40.56
Children2_2	RAPID	384	384.13	343.75	40.38	150	150	33.83	36.35

As the VM8 algorithm is very sensitive to initial values of QP, unsuitable values of QP can result in many frame skipping, while RAPID is quite robust, which can work with a wide range of initial QP values without any frame skipping. In all experiments, initial values of QP are always selected for optimizing the VM8 solution, and then these initial values of QP are also used in RAPID. As a result, RAPID is more robust and can handle scene change by quickly adjusting unsuitable values of QP to adapt the new scene. In some cases the frame skipping activity is very frequent in VM8 solution, especially when the target bit rate is very low. But, RAPID can deliver good performances without any frame skipping under the same conditions. This indicates that the control range of target bit-rate of RAPID is wider than that of VM8.

6. CONCLUSIONS

In this paper, we proposed a rate control scheme for efficient bit allocation for MPEG-4 video coding. We proposed a number of ideas: For example, our scheme considers the coding complexities of both object and frame and then performs bit allocation among frames and among VOs within a frame based on coding complexities. A PID buffer control mechanism is used to adjust the global bit rate. Finally, the algorithm performs adjustments for I-VOP as well as among multiple VOs within a frame. The performance results for both single VO and multiple VOs encoding authenticate that RAPID outperforms the VM8 solution by: (a) providing more accurate rate regulation; (b) achieving better picture quality; (c) reducing quality fluctuation; (d) balancing PSNR among both frames and multiple VOs; (e) allowing higher priority to favorite VOs; (f) maintaining a more stable buffer level; (g) covering a wide bit-rate control range; (h) in additional, tolerating unsuitable initial QPs and scene change.

REFERENCES

1. R. Koenen, Ed., "Overview of the MPEG-4 standard," Doc. ISO/IEC JTC1/SC29/WG11 N2725 Seoul, Korea, Mar.1999.
2. Hung-Ju Lee, Tihao Chiang, and Ya-Qin Zhang, "Scalable Rate Control for MPEEG-4 Video," *IEEE Trans.On Circuits and Systems for Video Technology*, VOL. 10, NO. 6, pp. 878-894, Sep. 2000.
3. Anthony Vetro, Huifang Sun and Yao Wang, "MPEG-4 Rate Control for Multiple Video Objects," *IEEE Trans. On Circuits and Systems for Video Technology*, VOL. 9, NO. 1, pp. 186-199, Feb. 1999.
4. Paulo Nunes, Fernando Pereira, "Scene Level Rate Control Algorithm for MPEG-4 Video Coding," in *Visual Communications and Image Processing, Proc. SP*IE 4310, pp.194-205, 2001.
5. Jose I.Ronda, Martina Eckert, Fernando Jaureguizar, and Narciso Garcia, "Rate Control and Bit Allocation for MPEG-4," *IEEE Trans. On Circuits and Systems for Video Technology*, VOL.9, NO. 8, pp.1243-1258, Dec. 1999.
6. J.I.Ronda, M. Eckert., S. Rieke, F.Jaureguizar, and A. Pacheco, "Advanced Rate control for MPEG-4 Coders," in *Proc. SPIE Visual Commun. & Image Processing*, pp. 383-394, San Jose, CA. Jan. 1998.
7. T.Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate-distortion modeling," *IEEE Trans. On Circuits and Systems for Video Technology*, VOL.7, NO. 1, pp. 246-250, Feb.1997.
8. MPEG-4 video verification model V8.0, ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG97/N1796, July 1997, Stockholm, Sweden.
9. ISO/IEC JTC1/SC29/WG11, MPEG93/457 "MPEG video test model 5," Draft, Apr. 1993.
10. Sanggyu Park, Youngsun Lee, and Hyunsik Chang, "A New MPEG-2 Rate Control Scheme Using Scene Change Detection," *ETRI Journal*, VOL 18, NO. 2, pp. 61-74, Jul. 1996.
11. Li-Jun Luo, Cai-Rong Zou, and Zhen-Ya He, "A New Algorithm on MPEG-2 Target Bit-Number Allocation at Scene Changes," *IEEE Trans. On Circuits and Systems for Video Technology,* VOL.7, NO. 5, pp. 815-819, Oct. 1997.
12. Vance J.VanDoren, "Tuning Fundamentals, Basics of Proportional-Integral-Derivative Control," *Control Engineering*, March 1998, http://www.controleng.com/archives/1998/ctl0301.98/03a305.htm.
13. "Tuning a PID (Three-Mode) Controller, " http://www.newportinc.com/Products/Techncal/techncal.htm.
14. H.-J.Lee, T.Chiang, and Y.-Q.Zhang, "Multiple-VO rate control and B-VO rate control," Doc. ISO/IEC JTC1/SC29/WG11 M2554, Stockholm, Sweden, July 1997.
15. "Description of rate control and texture coding core experiments on coding efficiency in MPEG-4 video," Doc. ISO/IEC JTC1/SC29/WG11 N2258, Stockholm, Sweden, July 1997.
16. Jordi Ribas-Corbera, Shawmin Lei, "Rate Control in DCT Video Coding for Low-Delay Communications," *IEEE Trans. On Circuits and Systems for Video Technology,* VOL.9, NO. 1, pp. 172-185, Feb. 1999.