A Robust and Adaptive Rate Control Algorithm for Object-Based Video Coding

Yu Sun, *Student Member, IEEE,* and Ishfaq Ahmad, *Senior Member, IEEE*

***Abstract*—This paper proposes a rate control algorithm for single and multiple objects video coding. The algorithm exploits prediction and feedback control to achieve accurate bit rate while maximizing the picture quality and simultaneously effectively handling buffer fullness. The algorithm estimates the bit budget of a frame based on its global coding complexity, and dynamically distributes the target bits for each object within a frame according to the object's coding complexity. Exploiting a novel buffer controller based on the proportional–integral–derivative (PID) technique used in automatic control systems, the algorithm effectively reduces the deviation between the current buffer fullness and the target buffer fullness, and minimizes the buffer overflow or underflow. The algorithm dynamically adjusts several parameters to further improve the system performance. A scene-change handling method is used to deal with scene changes. The combination of prediction and feedback control improves the adaptability of the rate controller under complicated environments; it also decreases the effect of random disturbance and the deviation caused by the variance between the real system and its statistical model. 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.**

***Index Terms*—Bit allocation, MPEG-4 video coding, multiple video objects, proportional–integral–derivative (PID) buffer control, rate control.**

I. Introduction

IN OBJECT-BASED video coding, such as MPEG-4 [1], an arbitrarily shaped time-variable visual entity can be individually manipulated and combined with other similar entities to produce a scene [2]. Each object in the scene is coded individually originating its own video bitstream, and a coded scene is the multiplexing of the several video bitstreams corresponding to the video objects (VOs) constituting the scene, which can be transmitted through either constant or variable rate channels. To make the transmission as efficient and accurate as possible, various coding factors should be jointly considered, for example, encoding rate, channel rate, and scene content, etc.

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 presence of multiple video objects exacerbates the complexity of the encoding procedure as the rate controller must distribute bits among different objects according to the application requirements. The rate control (RC) problem is well studied and several solutions exist for various standards and applications, for example, storage media with MPEG-1 and MPEG-2 [3]–[7], video conference with H.261 and H.263 [8], [9]. Recently, with the advent of MPEG-4, some rate control algorithms for video object-based coding are also proposed [10]–[20].

In MPEG-2 TM5 [7], bit-allocation is accomplished in the context of the layered MPEG structure. First, at the group of pictures (GOP) GOP layer, a target bit-budget is calculated for each GOP. Within a GOP, bits are allocated to the current frame according to its picture type, the global complexity of the previous frame and the remaining number of bits assigned to the GOP. Next, within a picture, the quantization parameter (QP) for a macroblock (MB) is set and modulated based on virtual buffer fullness, an empirical "re-action parameter" and the local variance of the video signal. Since the main goal of MPEG-2 is to provide high quality in video digital broadcast, it should have a fixed group of pictures and cannot skip frames when buffer tends to overflow. Hence, the rate control algorithm of MPEG-2 can only exploit the spatial domain by selecting suitable QPs to obtain the desired bit rate. On the other hand, since H.263 is for low-bit-rate video applications and MPEG-4 is for wide range of applications (including streaming video applications), their rate control algorithms can make appropriate decisions on both spatial (QP) and temporal (frame skipping) coding parameters to achieve the target bit rate. In [10], Chiang and Zhang have proposed a rate control scheme using a quadratic rate-distortion (R-D) model that describes the relation between the QP and the required bits for coding the texture. Based on this model, they presented a rate control algorithm [11] that was scalable for various bit rates, spatial and temporal resolutions, and could be applied to both DCT and wavelet-based encoders. In this algorithm, the number of target bits per frame is initially set to a weighted sum of the number of bits used for coding the previous frame and the average number of the remaining bits per frame, and then to prevent buffer underflow and overflow, the target is scaled by a proportional factor based on the current buffer occupancy. MPEG committee for single video object (SVO) simulations has adopted this algorithm as part of the video verification Model (VM8 [12]). Vetro and Sun [13], [14] extended the above R-D model and SVO algorithm to multiple video object (MVO) rate control, they used the same method as [11] to allocate target bits to a frame, the

Manuscript received May 12, 2002; revised March 18, 2003. This paper was recommended by Associate Editor H. Sun.

Y. Sun was with the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019-0015 USA. She is now with the Department of Computer Science, University of Central Arkansas, Conway, AR 72035 USA (e-mail: yusun@mail.uca.edu).

I. Ahmad is with the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019-0015 USA (e-mail: iahmad@cse.uta.edu).

Digital Object Identifier 10.1109/TCSVT.2004.833164

TABLE I
SUMMARY OF SYMBOLS

Parameter Name		Definition
Initialization	TBR	Target bit rate for the sequence
	F	Frame rate
Target Bit Estimation	$T_{ave,t}$	Initial weighted average target bits
	T_t	Target bits for the current frame
	$T_{i,t}$	Target bits for VOP_i at time t
	$T_{texture,i}$	Target bits for texture of VOP_i
	$W_{i,t}$	Weight of VOP_i at time t
	$NW_{i,t}$	Normalized weight for VOP_i
	$SIZE_{i,t}$	Number of *MBs* or partial *MBs* in VOP_i
	$NSIZE_{i,t}$	Normalized size for VOP_i
	$VAR_{i,t}$	Variance of the motion-compensated residual for VOP_i
	$NVAR_{i,t}$	Normalized variance of VOP_i
	$P^t_{i,j}$	Luminance value of pixel j in the motion-compensated residue of VOP_i
	$\overline{P_i^t}$	Arithmetic average pixel value of VOP_i
	k	Constant power in equation (2)
	NVO	Number of VOs in a frame
	n_i^t	Number of non-transparent pixels in VOP_i
	n_P	Number of P-frame
	n_B	Number of B-frame
	n_I	Number of I-frame
	$N_{I,t}$	Remaining number of I-frames at time t
	$N_{P,t}$	Remaining number of P-frames at time t
	$N_{B,t}$	Remaining number of B-frames at time t
	$\alpha(I)_t$	Weight of I-frame
	$\alpha(P)_t$	Weight of P-frame
	$\alpha(B)_t$	Weight of B-frame

total target bits of this frame are distributed proportional to the relative size, motion and variance of each object within this frame. They also adopted a proportional buffer control method to adjust the target bits for a frame. A pre-frame-skip control is utilized to avoid buffer overflow at the low bit-rate. To provide a proper tradeoff between spatial and temporal coding, the algorithm switches between a high rate coding and low rate coding modes. This technique has been also accepted by MPEG committee for MVO simulations in VM8 [12]. In [11] and [15], Lee and Chang also developed a MVO rate control strategy, the distribution of the bit budget within a frame is proportional to the square of MAD (mean absolute difference) of each VO. These MVO algorithms [11]–[15] assume multiple video objects have the same video object plane (VOP) rate, and consider the coding complexity for each object to decide the target bits among objects within a frame, but they do not take into account the total coding complexity for a frame. Recently, Nunes and Pereira [17] proposed to perform the target bit allocation by considering coding complexity both along the coding time and among VOs in one coding time instant, aiming to minimize quality fluctuations. Ronda and Eckert regarded multi-object rate control as an optimization problem, and proposed several cost criteria as goals to be optimized, the algorithm was introduced to minimize the average distortion of objects, so as to guarantee desired qualities to the most relevant ones and to keep constant ratios among the object qualities [16]. Ribas-Corbera and Lei [9] also focused on RC for the motion-compensated intercoded frames for H.263 and MPEG-4, the target bits of a frame are first set to TBR/F (TBR is the target bitrate, F is the frame rate), and then are modified by a small value based on the buffer fullness, thus, the target number of bits for each frame is nearly constant throughout the video sequence. This means that the quality of the encoded video will vary since the complexity of the video sequence may change along time. By using Lagrange multiplier to the bit-rate limitation, the optimal QP for an MB is determined.

The above model-based schemes adopt R-D models to estimate coding properties, though simple but effective. Since the building up of any mathematical models depends on the specific channel models and statistic characteristics of video signals, these models are approximate models. The foundations and conditions of a model cannot always match with the real application environment, there always exist some deviations or errors in these model-based schemes. For example, MPEG-4 [12] has adopted a generic model to estimate R-D properties for all kinds of sequences. However, estimations may not be accurate and always have some differences between the estimated values and the real ones.

This paper proposes a rate control algorithm called Robust Adaptive with Proportional–Integral–Derivative (RAPID). The algorithm aims to achieve an accurate bit rate while maximizes the picture quality and at the same time effectively handles buffer occupancy. The algorithm estimates coding properties and predict target bit budget before encoding, and combines various feedback information to compensate for the estimated deviations after encoding, in order to reduce the effect of random disturbance and the error caused by the variance between the real system and its statistical model. The specific characteristics of the algorithm include: 1) in addition to estimating the bit budget of a frame based on its global coding complexity, the algorithm dynamically distributes the target bits

TABLE I *(Continued.)*
SUMMARY OF SYMBOLS

Target Bit Estimation	$\alpha(K)_t$	One of $\alpha(I)_t$, $\alpha(B)_t$ or $\alpha(P)_t$ corresponding to the current frame type.
	$C_{i,t}$	Coding complexity of VOP_i at the current encoding time instant t
	$C_{G,t}$	Global complexity of the current frame
	$C_{ave\_P,t}$	Average global complexity of previous n_p P-frame
	$C_{ave\_B,t}$	Average global complexity of previous n_B B-frame
	$C_{ave,t}$	One of $C_{ave\_P,t}$ or $C_{ave\_B,t}$ depending on the current frame type
	$H_{i,t-1}$	Number of motion, shape and header bits used for VOP_i at time t-1
	$R_{r,t}$	Remaining bit counts at time t
Buffer Control	B_s	Buffer size
	$B_{f,t}$	Current buffer fullness
	$B_{p,t}$	Number of bits leaving the buffer at time t
	E_t	Error signal, the difference between the target buffer fullness $B_s/2$ and $B_{f,t}$
	PID_t	The output of PID controller
	K_p	Proportional control parameter
	K_i	Integral control parameter
	K_d	Derivative control parameter
Encoding, Post-Encoding	MAD_i	Mean absolute difference of VOP_i after motion compensation
	QP_i	Quantization parameter used for VOP_i
	X_{1i}, X_{2i}	First and second order model coefficients for object i
	A_t	Number of bits used for the current frame
Feedback Adjustment	$P_{avebits,t}$	Average number of bits used for previous n_P P-frames
	$B_{avebits,t}$	Average number of bits used for previous n_B B-frames
	$I_{avebits,t}$	Average number of bits used for previous n_I I-frames
	$PSNR_{ave\_P,t}$	Average PSNR for previous n_P P-frames
	$PSNR_{ave\_B,t}$	Average PSNR for previous n_B B-frames
	$PSNR_{ave\_I,t}$	Average PSNR for previous n_I I-frames
	$PSNR_{i,t-1}$	PSNR for VOP_i at time t-1
	$PSNR_P$	PSNR of a P-frame
	$PSNR_B$	PSNR of a B-frame
	$PSNR_i$	PSNR for VOP_i
	γ	Tuning factor for $\alpha(I)_t$ and $\alpha(B)_t$
	θ	Tuning factor for $W_{i,t}$
	P_{bits}	Number of bits used for coding a P-frame
	B_{bits}	Number of bits used for coding a B-frame
	$VOP_{i,bits}$	Number of bits used for coding VOP_i
	MSE_p	Mean-Square Error for P-frame
	MSE_B	Mean-Square Error for B-frame
	U_i	Priority of VO_i
	$QP_{I,t}$	QP of I-VOP at time t
	$QP_{ave,t,}$	Average QP of inter coded VOPs before the current I-VOP
	β_I_t	Adjusting factor for $QP_{I,t}$
	$PSNR_{I,t1}$	PSNR of the last I-VOP
	$PSNR_{ave,t1}$	Average PSNR of inter coded VOPs before the last I-VOP

to each object within a frame according to its characteristics; 2) the algorithm uses a proportional–integral–derivative (PID) buffer controller to effectively minimize the buffer overflow or underflow; and 3) the algorithm proposes several adaptation methods to automatically adjust parameters and improve the forecasting accuracy.

The remainder of this paper is organized as follows. In Section II, we describe the basic philosophy of the proposed adaptive RC algorithm for single/multiple video objects. In the same section, we discuss the proposed buffer control scheme named PID buffer controller to maintain a stable buffer level. In Section III, we present some adjustment methods using feedback information to further improve the efficiency of the proposed algorithm. Section IV summarizes the algorithm and describes its functionality. Section V includes the simulation results showing the performance of the proposed algorithm. Finally, Section VI concludes the paper by providing some final remarks, observations, and future research directions.

II. 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 the algorithm. Table I summarizes the symbols employed in the algorithm.

A. Initialization Stage

The initialization stage includes setting up encoding parameters and buffer size. The buffer size is initialized based on the delay requirement specified by users, and the target buffer fullness can be set to any level of the buffer size according to applications' requirements. As VM8, the default buffer size B_s is set to half of the target bit rate, and the target buffer fullness is the middle level of the buffer size in our algorithm.

We assume that multiple VOs are synchronous with the same VOP rate, and a frame is defined as a set of VOPs of different objects presenting in one encoding time instant [16]. To encode

the first I-frame, an initial QP is given. Once the first frame has been coded, we can obtain actual bits used in coding it, the remaining available bits for encoding the rest of the image sequence, etc.

B. Initial Target Bit Estimation

According to the type of the current frame, its target number of bits $T_{\text{ave},t}$ is initially set to a weighted average bitcount:

$$T_{\text{ave},t} = \alpha(K)_t \cdot \frac{R_{r,t}}{\alpha(I)_t \cdot N_{I,t} + \alpha(B)_t \cdot N_{B,t} + \alpha(P)_t \cdot N_{P,t}} \tag{1}$$

where $N_{I,t}$, $N_{P,t}$ and $N_{B,t}$ are the number of I, P and B frames which remain to be coded respectively at the current encoding time instant t, $\alpha(I)_t$, $\alpha(B)_t$ and $\alpha(P)_t$ are their weight factors, $R_{r,t}$ is the total number of bits available for the rest of the image sequence, $\alpha(K)_t$ is $\alpha(I)_t$, $\alpha(B)_t$ or $\alpha(P)_t$ corresponding to the current frame type.

C. Target Bits Adjustment Based on the Coding Complexity

Based on the perceptual efficient approach, the past history of each VO and the current coding complexity, a combination of strategies is used to adjust the target bits [7], [11]–[17].

It is necessary to analyze the characteristics of a VOP before target bit estimation [17]. As variance-like measure is usually used in bit allocation [9], [11], [13]–[15], we propose to adopt the variance and the size of a VOP to define the coding complexity of VOP_i to be encoded at time t, $C_{i,t}$, as

$$C_{i,t} = \text{SIZE}_{i,t} \cdot (\text{VAR}_{i,t})^k$$

where

$$\text{VAR}_{i,t} = \frac{\sum_{j=1}^{n_i^t} (P_{i,j}^t - \bar{P}_i^t)^2}{n_i^t}$$

$$\bar{P}_i^t = \frac{\sum_{j=1}^{n_i^t} P_{i,j}^t}{n_i^t}. \tag{2}$$

In (2), $P_{i,j}^t$ is the luminance value of pixel j in the motion-compensated residue of VOP_i, $\overline{P_i^t}$ is the arithmetic average pixel value of VOP_i, n_i^t is the number of nontransparent pixels in VOP_i, $\text{SIZE}_{i,t}$ is the number of macro-blocks (*MBs*) or partial *MBs* in VOP_i, $\text{VAR}_{i,t}$ is the variance of the motion-compensated residue for VOP_i, the power k is a constant.

The coding complexity computed by (2) naturally combines the object size ($\text{SIZE}_{i,t}$) and the variance ($\text{VAR}_{i,t}$) of the prediction error for a VOP, and therefore, can approximately reflect the instantaneous characteristics of this VOP. Since the coding complexity of a VOP is computed based on its motion-compensated residual, when a VO changes its features, its coding complexity also updates by some degree simultaneously. To avoid very large fluctuations of coding complexities and obtain smooth coding qualities along the coding time, we hope this coding complexity only acts as fine-tuning to target bit allocation for each encoding time instant, thus its influence should not be too strong. By many experiments, we found that $k = 1/4$ can reflect the instantaneous characteristics of a VOP, meanwhile weaken the coding complexity's influence to some degree during target bit allocation. However, in the VM8 solution of MPEG-4 [12], target bits are allocated to the current frame only according to the statistical information of the 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 fluctuated and overall degraded visual quality.

To adjust coding qualities among multiple objects within a frame, the algorithm sets weight for each object. The larger the weight for an object, the more target bits should be allocated to it. Let $W_{i,t}$ be the weight for VOP_i at time t, its initial value is 1.0, meaning that each object has equal weight at the beginning of encoding. The normalized weight for VOP_i, $NW_{i,t}$, can be obtained by

$$NW_{i,t} = \frac{W_{i,t}}{\left(\sum_{j=1}^{NVO} W_{j,t}\right)},$$

Here, *NVO* is the number of VOs in a frame. $W_{i,t}$ is dynamically adjusted along the coding process.

The global complexity of the current frame, $C_{G,t}$, can be obtained by

$$C_{G,t} = \sum_{i=1}^{NVO} (NW_{i,t} \cdot C_{i,t}).$$

Then, we can calculate the average global complexity $C_{\text{ave\_}P,t}$ for previous n_p P-frames, and $C_{\text{ave\_}B,t}$ for previous n_B B-frames before time t. Here, n_p and n_B are the number of the most recently coded P and B frames used in computing $C_{\text{ave\_}P,t}$ and $C_{\text{ave\_}B,t}$ respectively.

The initial target bit budget of the current frame, T_t, is then adjusted by

$$T_t = T_{\text{ave},t} \cdot \frac{C_{G,t}}{C_{\text{ave},t}} \tag{3}$$

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

D. Target Bits Adjustment Based on the Buffer Occupancy

The bit target is further refined based on the buffer fullness so as to get a more accurate target bit estimation. The aim of buffer control is to keep buffer fullness around the target level to reduce the chances of buffer overflow or underflow: if the buffer occupancy exceeds the target level, the target bits are decreased to some extent; similarly, if it is below the target level, the target bits are increased by some degree.

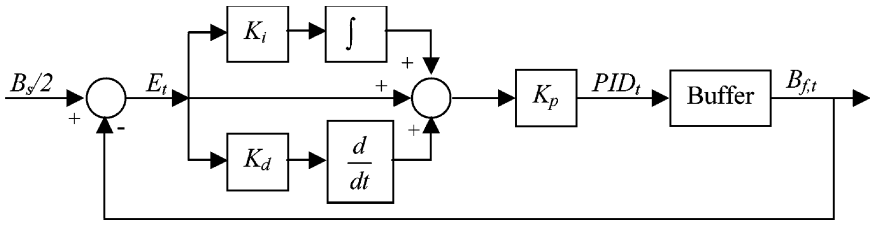


Fig. 1. PID buffer control system.

The VM8 and other algorithms adopt a simple nonlinear proportional buffer controller, whose control ability is rather less powerful. As shown in our experiments, when the complexity of a sequence changes drastically, the buffer tends to be out of control, especially in low bit rate cases.

The PID controller is by far the most popular feedback controller in the automatic control area [21], [22], and is especially suitable for unpredictable or imprecise processes to be controlled, which is one of the characteristics of video coding process since we cannot precisely predict the coming frames. The popularity of the PID technique is mainly attributed to its simplicity and good performance in a wide range of operating conditions. Here, we apply this technique to the buffer control in video coding (see Fig. 1). From the viewpoint of automatic control systems, the structure of our algorithm is a prediction plus feedback control system, but not a pure feedback system [23]. Unlike VM8, we do not use any additional means to avoid overflow or underflow since the PID buffer controller has enough control ability.

Our goal is to keep the buffer occupancy around the target buffer fullness, and minimize the deviation between the target buffer fullness and the actual buffer fullness. The error signal E_t, which measures the difference between the target buffer fullness $B_s/2$ and the actual output (current buffer fullness $B_{f,t}$) at time t, is defined as

$$E_t = \frac{\left(\frac{B_s}{2} - B_{f,t}\right)}{\frac{B_s}{2}}.$$

This error signal is sent to the PID controller

$$PID_t = K_p \cdot \left(E_t + K_i \cdot \int_0^t E_\tau \cdot d\tau + K_d \cdot \frac{dE_t}{dt}\right) \quad (4)$$

where K_p, K_i, and K_d are the proportional, integral, and derivative control parameters, respectively. The first term in (4) is the proportional action, it is the main component and can reduce the error between the current buffer fullness and the target buffer fullness, but cannot fully eliminate this error. The integral controller, the second term in (4), has the effect of eliminating the steady-state error by this way: when the error lasts, it can gradually enhance the control strength. But it may cause the transient response worsening. The derivative controller, the third term, has the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. The three-mode PID controller combines the advantages of each individual controller, and thus, improves both the transient and the steady-state response.

Then, the target bits T_t can be further adjusted by

$$T_t := (1 + PID_t) \cdot T_t\,. \quad (5)$$

To obtain a minimum visual quality for each frame, the lower bound of the target bits imposed to each frame in VM8 is $\mathrm{TBR/F}$, TBR and F are the target bit rate and frame rate required by the application. This means each frame must obtain at least the average number of bits per frame without considering its coding complexity, and thus the total target bitrate actually allocated to frames is certainly equal or larger than the application's target bitrate. Since we think that only fewer bits are needed to maintain acceptable qualities for some frames with low complexity, we decrease this lower bound to

$$T_t := \max\left\{\frac{\mathrm{TBR}}{4 \cdot \mathrm{F}}, T_t\right\}.$$

For most applications, overflow is much worse than underflow, so maximum bits should be more strictly constrained than the minimum one. To avoid buffer overflow, the maximum number of bits is given as

$$T_t := \min\left\{\frac{2 \cdot \mathrm{TBR}}{\mathrm{F}}, T_t\right\}.$$

E. Dynamic Target Bit 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 target bits among multiple objects within a frame [17], [26]. Normally, a rate control scheme should allocate more bits to important VOs (e.g., foreground VOs) than other areas (e.g., background VOs). To obtain uniform video quality, the coding complexity and perceptual importance of a VO must be considered during bit allocation among VOs. We have chosen the normalized weight, size and variance as three factors in the target bit distribution. Therefore, as long as the target bits are given for a frame, the number of target bits for VOP_i at time t, $T_{i,t}$, is allocated by

$$T_{i,t} = \frac{NW_{i,t} \cdot (\mathrm{NSIZE}_{i,t} \cdot \mathrm{NVAR}_{i,t})}{\sum_{j=1}^{NVO} NW_{j,t} \cdot (\mathrm{NSIZE}_{j,t} \cdot \mathrm{NVAR}_{j,t})} \cdot T_t \quad (6)$$

where $\mathrm{NSIZE}_{i,t}$ and $\mathrm{NVAR}_{i,t}$ are the size and variance of VOP_i, normalized by the total size and variance of all objects, respectively.

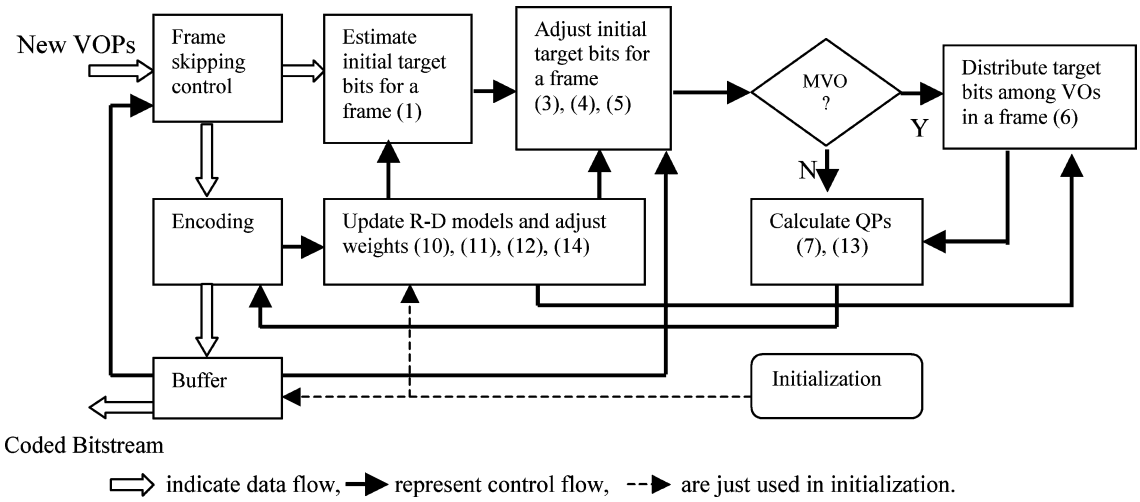


Fig. 2. Functional diagram of RAPID.

F. Quantization Parameter Calculation

The quantization parameter for texture encoding is computed based on the R-D model of each VO for the corresponding VOP type [14], [15]. Once $T_{i,t}$ is obtained, the number of target bits for coding the texture of VOP_i, $T_{\mathrm{texture,i}}$, can be computed by

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

where $H_{i,t-1}$ denotes the number of bits actually used for coding the motion, shape, and header for VOP_i at time $t-1$.The proposed algorithm also adopt this R-D model [14], [15]

$$T_{\mathrm{texture},i} = \frac{X_{1i} \cdot \mathrm{MAD}_i}{\mathrm{QP}_i} + \frac{X_{2i} \cdot \mathrm{MAD}_i}{\mathrm{QP}_i^2} \tag{7}$$

where MAD_i is mean absolute difference for VOP_i after motion compensation, QP_i denotes quantization parameter used for VOP_i, X_{1i} and X_{2i} are the first- and second-order model coefficients.

Nunes and Pereira found that intra-coded VOPs are typically encoded with lower quality than inter-coded VOPs in VM8 [17]. We also observed the similar phenomenon, which results in large quality variations and quality decay. It indicates that the bit allocation strategy of VM8 is not very efficient. The partial reason is analyzed as follows. A good coding performance relies on an accurate R-D model, and the accuracy of R-D model bases on the quality and quantity of the data set used to update it. Generally speaking, more updating data points (encoded VOPs) 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 encoded VOPs are selected to update these R-D models and thus 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. For example, if the intra period is set to one second and VOP rate is 15 VOP/s for a VO, then there is only one I-VOP among 15 VOPs. Even enough quantity of I-VOPs can be accumulated after coding many VOPs in a long sequence, most of them cannot represent the change of the coming I-VOPs. Since the shortest distance between the current I-VOP and its last I-VOP is 14 VOPs, it is possible that the incoming I-VOP is quite different to its last I-VOP. Therefore, the I-VOP's model updated gradually by previous encoded I-VOPs cannot completely reflect contents of the coming I-VOP in time. Thus, the R-D model of I-VOPs is less accurate than that of the inter-coded VOPs and, as a result, the coding qualities of I-VOPs tend to fluctuate.

TABLE II
ENCODING PERFORMANCE AT VARIOUS TARGET BITRATES FOR *COASTGUARD* SEQUENCE

PID Coefficients			Bit Rate (kbps)		# Coded frames		PSNR
Kp	Ki	Kd	Target	Actual	Target	Actual	(dB)
1.0	0.25	0.3	32	32.00	150	150	27.79
			64	63.93	150	150	30.22
			128	127.49	150	150	32.40
			192	191.62	150	150	34.47

To avoid the above problem and achieve a consistent 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 inter-coded VOPs with some adjustment. Though this method is quite simple, it is efficient to overcome visual quality fluctuation or degradation of I-VOPs.

As usual, the QP is limited to vary between 1 and 31 and only permitted to change within 25% of the previous QP. This can ensure QP would not change too much compared with its previous QP, and avoid causing huge quality fluctuation.

G. Encoding and Updating

After encoding video objects within a frame, the encoder updates the R-D model of each VO for the corresponding VOP type based on the encoding results of the current objects as well as the past objects. The first and second model parameters, X_{1i} and X_{2i}, are updated by using the linear regression technique [10], [15].

TABLE III
ENCODING PERFORMANCE USING THE FIXED PID COEFFICIENTS

Video Sequence	Bit Rate (Kbps)		# Coded frames		PSNR
	Target	Actual	Target	Actual	(dB)
Mother_daughter	96	95.50	150	150	38.89
Stefan	256	254.56	150	150	32.37
Stefan	128	127.53	150	150	28.67
Foreman_Train	256	255.61	150	150	37.45

The virtual buffer fullness is updated by

$$B_{f,t} := B_{f,t} + (A_t - B_{p,t})$$

where A_t represents the number of actual bits used for encoding the current frame. $B_{p,t}$ is the number of bits to be output from the virtual buffer per frame [12]

$$B_{p,t} = \alpha(K)_t \cdot \frac{R_{r,t}}{\alpha(I)_t \cdot N_{I,t} + \alpha(B)_t \cdot N_{B,t} + \alpha(P)_t \cdot N_{P,t}}. \tag{8}$$

Actually, the right side of (8) is the same as that of (1), because we hope the initial target bits which to be put into the buffer should roughly equal to the bits to be output from the buffer per frame, so as to keep buffer fullness around the target level and derive a useful signal of buffer fullness.

H. Frame-Skipping Control

When the number of bits in the buffer is too large, the encoder normally skips encoding frames to avoid buffer overflow, here we use the same method as VM8 [12]: the encoder needs to examine the current buffer fullness before encoding the next frame, if the buffer occupancy exceeds 80% of the buffer size, the encoder skips the next frame, and the buffer fullness is updated by subtracting $B_{p,t}$.

I. Scene-Change Handling

Scene change means the abrupt change of frame characteristics between consecutive frames [24], [25]. To better deal with scene-change problems, it is essential to detect the occurrence of a scene change before coding a frame. The scene-change detection is based on the motion estimation and MB type decision. A large number of intra MBs represents that motion estimation and compensation failed and a scene change has occurred. Therefore, if the number of intra MBs in a frame exceeds a pre-set threshold [27], the frame is regarded as a scene-change frame and its frame type is set to intra, the first I-frame following the scene-change frame is set to inter, thus the number of I-frames in the sequence would not vary.

III. FEEDBACK ADJUSTMENT OF THE PARAMETERS

As the encoding process is uncertain or cannot be precisely modeled during the prediction phase, besides exploring for more accurate models, another efficient way is using feedback information to compensate prediction errors along the coding process. To further improve the system performance, we dynamically adjust some coding parameters based on feedback information during the coding process.

A. Weight Adjustment for Frame Types

$\alpha(I)_t$, $\alpha(B)_t$, and $\alpha(P)_t$ are weights for I, B and P frames, respectively, they are used in target bit allocation. To achieve a smooth visual quality, after encoding an I-frame or a B-frame, $\alpha(I)_t$ and $\alpha(B)_t$ are updated, while $\alpha(P)_t$ is fixed to 1.0. The updating of $\alpha(I)_t$ and $\alpha(B)_t$ comprehensively considers several factors: currently, average bits used in encoding previous I, P or B frames, and average coding qualities of previous I, P, or, B frames. In principle, if the average coding quality of previous coded B-frames is lower than that of previous coded P-frames, we increase $\alpha(B)_t$. Then next B-frame to be coded can be allocated more bits, thus its quality is improved gradually to keep consistent with the average quality of P-frames. On the contrary, if the average PSNR of the coded B-frames is higher than that of the coded P-frames, we decrease $\alpha(B)_t$ to get fewer target bits for the next B-frame, thus decrease its coding quality gradually to keep close to the average PSNR of P-frames.

Assuming the number of bits used in encoding a frame is approximately in inverse proportion to the mean squared error (MSE) between the original frame and the reconstructed frame, namely, the more bits used in coding a frame, the less MSE is

$$B_{\text{bits}} \propto \frac{1}{\text{MSE}_B} \text{ and } P_{\text{bits}} \propto \frac{1}{\text{MSE}_P} \Rightarrow \frac{B_{\text{bits}}}{P_{\text{bits}}} \propto \frac{\text{MSE}_P}{\text{MSE}_B} \tag{9}$$

where MSE_p and MSE_B are the MSE of P and B frames, respectively, P_{bits} and B_{bits} represents the number of bits used in coding a P or B frame. From the PSNR formula

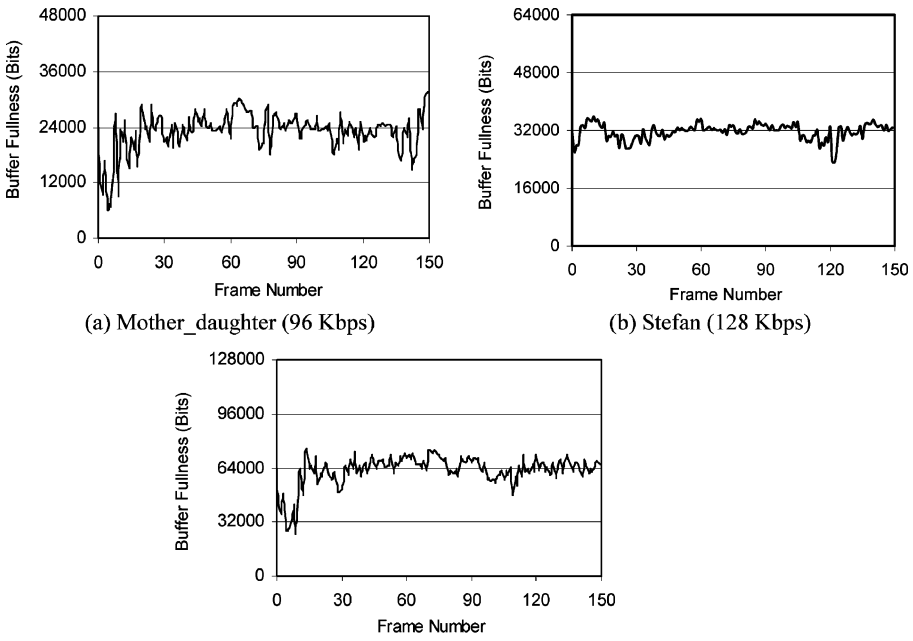
$$\text{PSNR} = 10 \cdot \log_{10} \frac{255^2}{\text{MSE}},$$

we have

$$\begin{aligned} \text{PSNR} &= 10 \cdot \left(\frac{\ln \frac{255^2}{\text{MSE}}}{\ln 10} \right) \Rightarrow \text{MSE} \\ &= \frac{255^2}{e^{\text{PSNR} \cdot (\ln 10/10)}} = \frac{255^2}{e^{\text{PSNR}/D}} \\ \text{with } D &= \frac{10}{\ln 10} \approx 4.35 \Rightarrow \frac{\text{MSE}_P}{\text{MSE}_B} \\ &= \frac{e^{PSNR_B/D}}{e^{\text{PSNR}_p/D}} = e^{(\text{PSNR}_B - \text{PSNR}_p)/D}. \end{aligned}$$

Here, PSNR_P and PSNR_B represent the PSNR of a P and a B frame, respectively. Thus, we have the following relationship from (9):

$$\frac{B_{\text{bits}}}{P_{\text{bits}}} \propto e^{(P_{\text{PSNR}} - B_{\text{PSNR}})/D}, \qquad \text{with D} \approx 4.35 \tag{9a}$$



(a) Mother_daughter (96 Kbps) (b) Stefan (128 Kbps)

(c) Foreman_Train (Scene Change at the 58th frame, 256 Kbps).

Fig. 3. Buffer fullness using the fixed PID coefficients.

TABLE IV
ENCODING RESULTS OF SINGLE OBJECT RATE CONTROL (IPPP…PPP)

Video Sequence	Rate Control Algorithm	Bit Rate (kbps)		Skipped Frames	PSNR (dB)
		Target	Actual		
Stefan (qcif)	VM8	112	112.84	0	28.09
	RAPID	112	111.77	0	28.09
Coastguard (qcif)	VM8	64	64.03	3	29.97
	RAPID	64	63.94	0	30.05
Coastguard (qcif)	VM8	112	111.99	0	32.04
	RAPID	112	112.16	0	32.05
Silent (qcif)	VM8	48	48.21	4	34.37
	RAPID	48	47.88	0	34.55
Container (qcif)	VM8	48	47.83	3	35.61
	RAPID	48	47.99	0	35.72
Hall (qcif)	VM8	32	32.04	2	34.73
	RAPID	32	31.97	0	35.02
Moible (qcif)	VM8	192	194.85	0	27.67
	RAPID	192	191.93	0	27.62

Based on the above theoretical analysis, we exploit the average bits used in coding previous frames, the difference of PSNR values and an exponential relationship to adjust $\alpha(I)_t$, $\alpha(B)_t$ as follows:

$$\alpha(B)_t = \frac{B_{\text{avebits},t}}{P_{\text{avebits},t}} \cdot e^{(\text{PSNR}_{\text{ave\_}P,t} - \text{PSNR}_{\text{ave\_}B,t})/\gamma} \quad (10)$$

$$\alpha(I)_t = \frac{I_{\text{avebits},t}}{P_{\text{avebits},t}} \cdot e^{(\text{PSNR}_{\text{ave\_}P,t} - \text{PSNR}_{\text{ave\_}I,t})/\gamma} \quad (11)$$

where $P_{\text{avebits},t}$, $B_{\text{avebits},t}$, and $I_{\text{avebits},t}$ denote the average number of bits used per frame in coding previous n_P P-frames, n_B B-frames, and n_I I-frames, respectively; $\text{PSNR}_{\text{ave\_}P,t}$, $\text{PSNR}_{\text{ave\_}B,t}$, and $\text{PSNR}_{\text{ave\_}I,t}$ are the corresponding average PSNRs. Considering the tradeoff between keeping the algorithm stability and rapidly reflecting the influence of scene's variations, we empirically choose the window size $(n_I + n_P + n_B)$ to *30*, the simulation results are not very sensitive to this specific value of the window size. If γ is too large, this adjustment is not effective; if it is too small, the effect is too strong. According to (9a), γ should roughly be 4. In our simulation, we find that $\gamma = 8$ can obtain better performance, we finally empirically choose $\gamma = 8$ for conservative reason.

TABLE V
ENCODING RESULTS OF SINGLE OBJECT RATE CONTROL (IPPP…IPPP)

Video Sequence	Algorithms	Bit Rate (Kbps)		Skipped Frames	PSNR (dB)
		Target	Actual		
Coastguard (qcif)	VM8	64	64.44	6	29.29
	RAPID	64	63.66	0	29.59
Coastguard (qcif)	VM8	128	128.70	3	31.97
	RAPID	128	127.46	0	32.23
Container (cif)	VM8	512	507.01	8	37.72
	RAPID	512	511.02	0	38.58
Bream2_1 (qcif)	VM 8	64	64.26	5	27.88
	RAPID	64	63.99	0	27.95
Bream2_1 (qcif)	VM8	192	193.62	3	35.05
	RAPID	192	191.92	0	35.31
Silent (qcif)	VM8	128	127.92	14	36.67
	RAPID	128	126.61	0	37.84
News (qcif)	VM8	64	63.66	19	32.75
	RAPID	64	63.69	0	34.00
News (qcif)	VM8	128	128.85	15	37.10
	RAPID	128	127.27	0	38.63
Mobile (qcif)	VM8	128	128.74	5	25.45
	RAPID	128	127.82	0	25.69
Mobile (qcif)	VM8	384	383.72	0	30.67
	RAPID	384	382.96	0	30.85
Train_Right (qcif)	VM8	64	64.84	10	28.27
	RAPID	64	63.68	0	29.07
Train_Right (qcif)	VM8	256	256.83	4	35.82
	RAPID	256	255.77	0	36.70

B. Weight Adjustment Among Multiple Objects

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

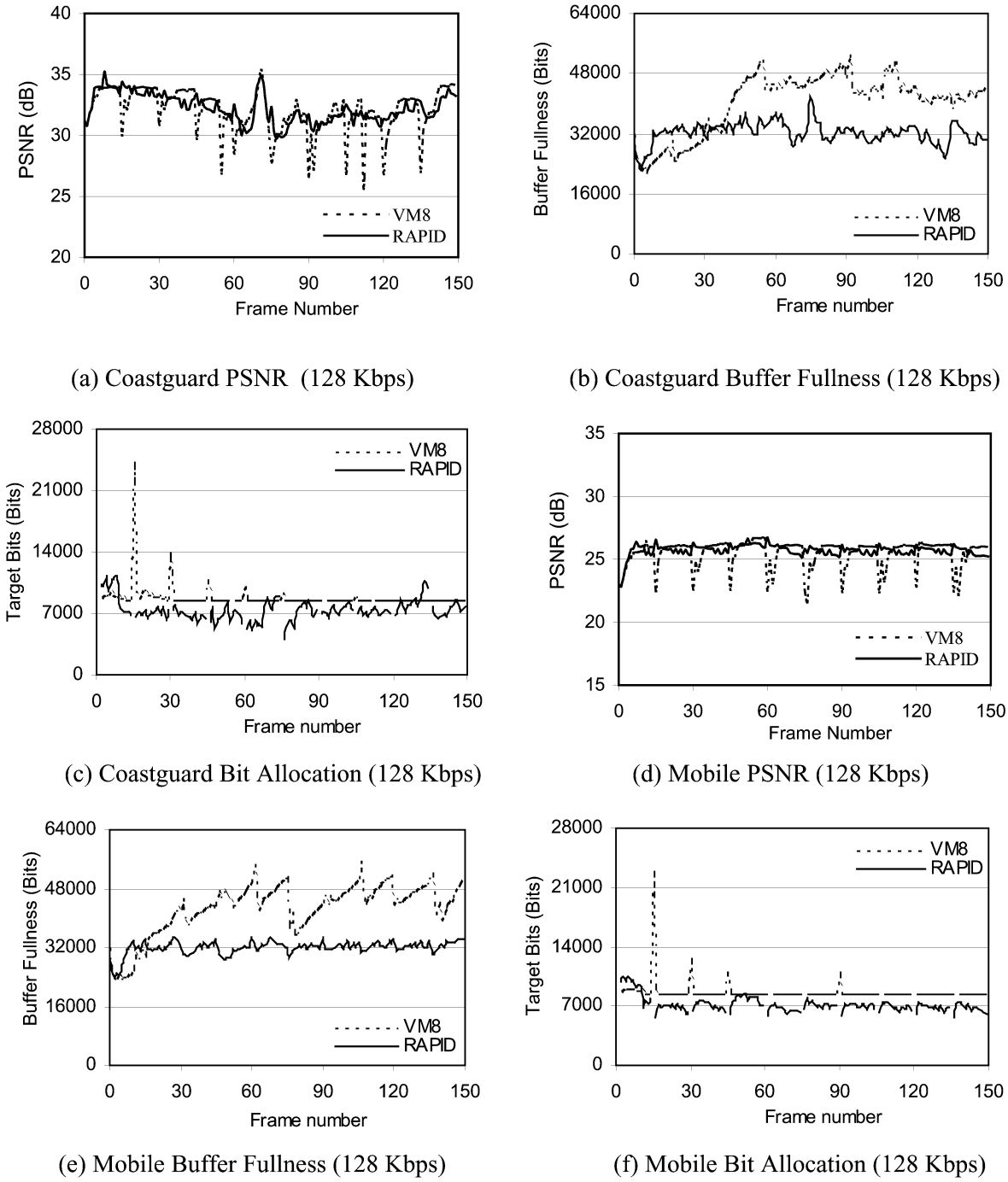


Fig. 4. Experimental results for QCIF sequences encoded at various bit rates (IPPP…IPPP).

object is further adjusted according to the PSNR difference of previous coded VOPs. We can derive the following relationship from the similar procedure as (9) and (9a):

$$\frac{\mathrm{VOP}_{i,\mathrm{bits}}}{\mathrm{VOP}_{1,\mathrm{bits}}} \propto e^{(\mathrm{PSNR}_1-\mathrm{PSNR}_i)/D}, \qquad \text{with D} \approx 4.35$$

where $\mathrm{VOP}_{i,\mathrm{bits}}$ and $\mathrm{VOP}_{1,\mathrm{bits}}$ represent the number of bits used in coding VOP_i or VOP_1, respectively. Therefore, we also exploit the difference of PSNR values and an exponential relationship to adjust the weight of $\mathrm{VOP}_i(W_{i,t})$ in (12) and (12a). We initialize $W_{i,0}$ to 1.0 for all VO_i, meaning that each object has equal weight at the beginning of encoding, and adopt the VO_1 as a referential base, its weight W_1 is 1.0 forever. $\mathrm{PSNR}_{i,t-1}$ for VOP_i ($i = 2 \ldots NVO$) at time $t-1$ is compared to the $\mathrm{PSNR}_{1,t-1}$ for VOP_1, if $\mathrm{PSNR}_{i,t-1}$ is lower than $\mathrm{PSNR}_{1,t-1}$, the algorithm improves the weight of VOP_i, thus VOP_i obtains more target bits and thus achieves a higher quality; otherwise, decreases $W_{i,t}$ to achieve lower quality. The weight for VOP_i is updated by

$$W_{i,t} = W_{i,t-1} \cdot e^{(\mathrm{PSNR}_{1,t-1}-\mathrm{PSNR}_{i,t-1})/\theta}, \qquad \text{for } i > 1. \tag{12}$$

Here, the tuning factor θ is selected to 4 theoretically and empirically. Obviously, a further improvement could be easily made to provide different priority levels for VOs

$$W_{i,t} = W_{i,t-1} \cdot e^{(\mathrm{PSNR}_{1,t-1} - \mathrm{PSNR}_{i,t-1} + U_i)/\theta}, \qquad \text{for } i > 1 \tag{12a}$$

where U_i is the priority of VO_i. $U_i > 0$ (dB) means a higher priority while $U_i < 0$ (dB) corresponds to a lower priority. For example, if one hopes the foreground object VO_2 to have a PNSR 3 dB higher than that of the background object VO_1, one can set $U_2 = 3.0$.

C. Quantization Parameter Updating for I-VOP

The QP of I-VOP for an object is obtained directly by averaging the QPs of its previous l inter-coded VOPs, since the coding type of I-VOP is intra-coded, different from inter-coded, the PSNR for I-VOP is different from PSNRs for inter-coded VOPs even if I-VOP uses the same QP as its previous inter-coded VOPs. Thus, we cannot simply use the average QP of previous inter-coded VOPs to code I-VOP. To better maintain the consistent quality between I-VOP and its previous inter-coded VOPs, we add a bias β_I_t to adjust the QP for I-VOP as follows:

$$\mathrm{QP}_{I,t} = \mathrm{QP}_{\mathrm{ave},t} + \beta\_I_t \tag{13}$$

where $\mathrm{QP}_{I,t}$ is the QP of the current I-VOP; $\mathrm{QP}_{\mathrm{ave},t}$ is the average QP of l inter-coded VOPs before the current I-VOP. Considering QP is roughly inverse proportional to PSNR and they have an approximately linear relationship in the local area, we adopt the linear adjustment here. Initially, β_I_t is 1.0 and updated as

$$\beta\_I_t := \beta\_I_t + \frac{\mathrm{PSNR}_{I,t1} - \mathrm{PSNR}_{\mathrm{ave},t1}}{\lambda} \tag{14}$$

where $t1$ is the coding time of the last I-VOP, $\mathrm{PSNR}_{I,t1}$ is the PSNR of the last I-VOP and $\mathrm{PSNR}_{\mathrm{ave},t1}$ is the average PSNR of l inter-coded VOPs before the last I-VOP, λ is a tuning parameter. When the last I-VOP's PSNR is higher than the average PSNR of its previous l inter-coded VOPs, the QP for the current 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 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. l and λ are empirically chosen to be 3 and 16, respectively, for all coding conditions, the simulation results are not very sensitive to the specific values of l and λ.

IV. Description of the RAPID Rate Control Algorithm

Here, we illustrate the RAPID algorithm in Fig. 2 and summarize it as the following steps.

Step 1) Initialize the parameters for the encoder.
Step 2) Estimate the number of initial target bits for a frame using (1).

TABLE VI
ENCODING RESULTS OF SINGLE OBJECT RATE CONTROL (IBBP…IBBP)

Video Sequence	Algorithms	Bit Rate (Kbps)		Skipped Frames	PSNR (dB)
		Target	Actual		
Coastguard (qcif)	VM8	64	65.90	5	28.87
	RAPID	64	63.38	0	29.86
Coastguard (qcif)	VM8	128	141.54	1	32.25
	RAPID	128	128.57	0	32.71
Container (cif)	VM8	128	127.77	14	31.85
	RAPID	128	126.67	0	32.95
Container (cif)	VM8	512	533.78	5	37.81
	RAPID	512	508.04	0	38.40
Bream2_1 (qcif)	VM8	64	66.16	4	27.65
	RAPID	64	63.78	0	28.39
Bream2_1 (qcif)	VM8	192	195.15	4	35.31
	RAPID	192	192.56	0	35.92
Silent (qcif)	VM8	128	128.49	8	34.92
	RAPID	128	127.92	0	36.76
News (qcif)	VM8	64	63.47	7	31.52
	RAPID	64	63.31	0	34.09
News (qcif)	VM8	128	129.70	7	35.67
	RAPID	128	127.59	0	38.42
Mobile (qcif)	VM8	128	132.25	2	25.86
	RAPID	128	126.86	0	27.10
Mobile (qcif)	VM8	384	393.17	3	31.15
	RAPID	384	382.67	0	32.17
Train_&_T_R (qcif)	VM8	64	66.69	1	27.56
	RAPID	64	64.09	0	28.14
Train_&_T_R (qcif)	VM8	256	274.13	1	36.23
	RAPID	256	255.97	0	37.11
Stefan (qcif)	VM8	112	118.14	5	27.16
	RAPID	112	111.72	0	28.01
Stefan (qcif)	VM8	256	264.65	2	31.62
	RAPID	256	254.56	0	32.37

Step 3) Adjust the initial target bits for a frame based on the coding complexity and buffer occupancy using (3), (4), and (5).
Step 4) Distribute target bits among multiple VOs in a frame using (6).
Step 5) Calculate the quantziation parameter using (7) and (13).
Step 6) Encode frame/objects.
Step 7) Update R-D Model and adjust other parameters using (10), (11), (12), and (14).
Step 8) Apply frame-skipping control, if necessary.
Step 9) Go to Step 2 until the end.

V. Simulation Results

This section presents the performance of the proposed RAPID algorithm. Simulations are based on a Momusys Codec for the MPEG-4 Video Verification Model VM8.0 [12]. 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 [9], [16]. In all experiments, the buffer size B_s is set to half of the target bit rate $\mathrm{TBR}/2$, and the initial buffer occupancy is set to half of the buffer size ($B_s/2$) after coding the first frame. The initial values of $\alpha(I)_t$, $\alpha(B)_t$, and $\alpha(P)_t$ are 3.0, 0.5, and 1.0, respectively, $\alpha(I)_t$ and $\alpha(B)_t$ are dynamically adjusted during the encoding process.

TABLE VII
ENCODING RESULTS OF MULTIPLE OBJECT RATE CONTROL (IPPP…PPP)

Video Sequence	Algorithms	Bit Rate (kbps)				Skipped Frames	PSNR(dB)	
		Target	Actual	VO1	VO2		VO1	VO2
News_1 (Ballet)	VM8	128	128.51	63.83	64.68	0	33.53	34.87
News_2 (Speakers)	**RAPID**	128	128.63	70.88	57.75	0	34.22	34.22
News_1 (Ballet)	VM8	256	257.83	142.94	114.89	0	38.69	39.09
News_2 (Speakers)	**RAPID**	256	256.85	148.39	108.46	0	38.94	38.92
Bream2_0 (Background)	VM8	128	128.24	26.73	101.51	0	42.34	27.08
Bream2_1	**RAPID**	128	128.29	9.10	119.19	0	38.77	27.92
Bream2_0 (Background)	VM8	256	257.67	49.81	207.43	0	44.07	31.24
Bream2_1	**RAPID**	256	256.89	9.14	247.75	0	38.80	32.37
Children2_1	VM8	384	384.29	296.08	88.21	3	32.44	40.54
Children2_2	**RAPID**	384	384.27	335.91	48.36	0	33.60	36.98
Coastguard2	VM8	112	111.98	53.12	58.86	0	31.23	31.44
Coastguard3	**RAPID**	112	111.92	54.39	57.53	0	31.24	31.25
Coastguard2	VM8	256	255.87	126.52	129.35	0	37.79	35.33
Coastguard3	**RAPID**	256	255.53	105.64	149.89	0	36.01	36.03
Container_1,	VM8	112	111.69	73.66	38.03	0	31.81	47.32
Container_5	**RAPID**	112	111.92	97.44	14.48	0	33.82	33.87

TABLE VIII
ENCODING RESULTS OF MULTIPLE OBJECT RATE CONTROL (IPPP…IPPP)

Video Sequence	Algorithms	Bit Rate (Kbps)				Skipped Frames	PSNR(dB)	
		Target	Actual	VO1	VO2		VO1	VO2
News_1 (Ballet)	VM8	128	130.13	56.09	74.04	10	32.42	32.54
News_2 (Speakers)	**RAPID**	128	127.89	56.35	71.54	0	32.84	32.86
News_1 (Ballet)	VM8	256	259.36	124.07	135.29	7	37.23	37.48
News_2 (Speakers)	**RAPID**	256	253.18	124.54	128.64	0	37.72	37.67
Bream2_0 (Background)	VM8	128	130.38	30.35	100.03	7	40.82	26.34
Bream2_1	**RAPID**	128	128.55	13.12	115.43	0	37.99	27.29
Bream2_0 (Background)	VM8	256	260.82	51.80	209.02	5	43.24	30.71
Bream2_1	**RAPID**	256	257.28	13.44	243.84	0	38.25	31.89
Children2_1	VM8	384	385.45	280.10	105.35	6	31.38	36.30
Children2_2	**RAPID**	384	383.98	313.06	70.92	0	32.62	33.83
Coastguard2	VM8	112	112.50	51.54	60.96	2	30.20	30.13
Coastguard3	**RAPID**	112	112.71	52.13	60.58	0	30.33	30.28
Coastguard2	VM8	256	257.08	119.78	137.30	0	36.69	34.59
Coastguard3	**RAPID**	256	257.35	101.77	155.58	0	35.23	35.23
Container_1	VM8	112	113.19	75.52	37.67	0	31.15	45.90
Container_5	**RAPID**	112	112.92	98.93	13.99	0	32.93	32.96

A. Robustness of the PID Buffer Controller

In automatic control systems, three PID coefficients are usually constants and determined empirically depending on application's requirements. Generally speaking, increasing the proportional coefficient K_p can intensify the control power, but too large K_p may cause the control system unstable; the integral part can eliminate the steady-state error, but may cause the system overshoot or oscillate if K_i is too large; the derivative part can reduce the overshoot and improve transient properties, but it is sensitive to noise. Therefore, it is important to select suitable values for these coefficients. By exhaustive experiments, we empirically set K_p, K_i and K_d to 1.0, 0.25, and 0.3, respectively, for various coding environments.

To examine the robustness of the PID buffer controller, we adopt the fixed PID coefficients ($K_p = 1.0$, $K_i = 0.25$, $K_d = 0.3$) to deal with various coding environments.

1) Encoding the representative sequence *coastguard* (qcif, IBBP…IBBP, 15 fps, 112 kbps, intra_period is 15 frames) at different target bitrates, results in Table II show that RAPID has realized accurate target bitrates without frame skipping using the selected PID parameters, implying that the encoding performance are not very sensitive to these coefficients at various target bitrates.
2) Encoding three representative sequences (qcif, IBBP…IBBP, 15 fps, intra_period is 15 frames) with typical characteristics: *Mother_Daughter* for slow motion, *Stefan* for fast motion, and the scene-change sequence *Foreman_Train* which the first 57 frames are from the "*Foreman*" and the remaining 93 frames are from the *"Train*", thus, scene change happens at the 58th frame. The results in Table III also indicate that we have achieved accurate target bitrates without frame skipping for different kinds of sequences. Furthermore, from Fig. 3, one can see that buffer curves are very stable, they are around the target buffer fullness with a small fluctuation.

Hence, the most important conclusion that can be obtained from these results is that the fixed PID coefficients are robust enough and not very sensitive to different kinds of sequences and target bitrates.

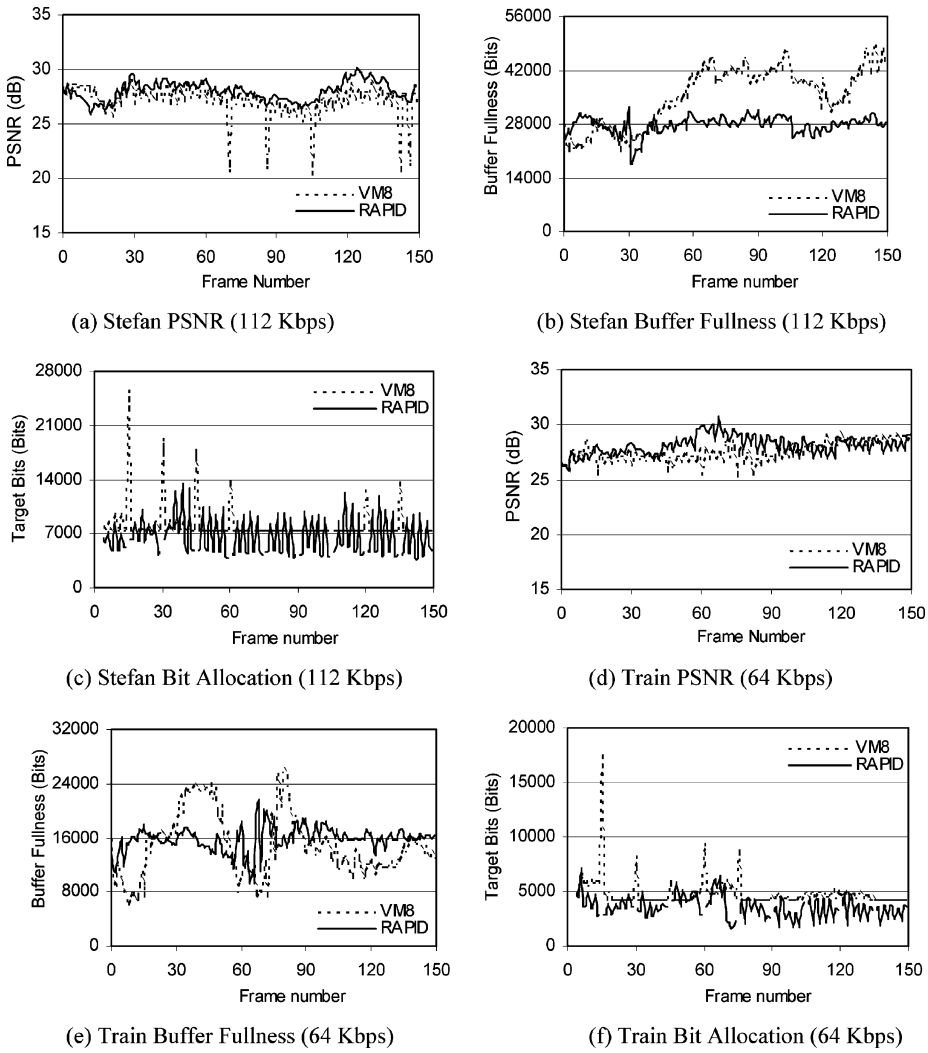


Fig. 5. Experimental results for QCIF sequences encoded at various bitrates. (IBBP…IBBP).

B. Single-Object Rate Control

We have conducted three sets of experiments for single-object RC. The target number of frames to be encoded is 150. All the sequences are encoded at 15 fps with different temporal prediction structures:

1) Only the first frame is I-frame and the remaining frames are all P-frames (IPPP…PPP).
2) Both I and P frame types are used and the intra period is set to 15 frames (IPPP…IPPP).
3) I, P and B frame types are used, the intra period is set to 15 frames, the number of B-frames is set to 2 between two P-frames or between I-frame and P-frame, the number of P-frames is set to 4 between two I-frames (IBBP…IBBP).

The structure (1) is the simplest case in RC since only P-frames needed to be controlled, and this is the general assumption in [9], [11], and [13]–[15]. Table IV shows its encoding performance for various sequences with one rectangular or arbitrary shape VO.

Table V shows the encoding results for the structure (2). Fig. 4 shows PSNR, buffer occupancy, and bit allocation curves in detail for several sequences.

Table VI shows encoding results for the structure (3) and Fig. 5 presents some example curves.

By examining the results in Tables IV–VI, it is obvious that RAPID achieves more accurate target bit rates and target frame rate with usually higher average PSNRs when compared with the VM8 solution.

Inspecting buffer fullness in these figures, our buffer curves are usually smoother and closer to the target buffer fullness when compared with those of VM8, they are always in the safe range (lower than the frame skipping threshold) of the buffer. One can see that RAPID almost overcomes the frame skipping problem from Tables IV–VI. However, VM8's buffer occupancy curves are more fluctuated, for example, in Fig. 4(b), three frames are skipped at the 54th, 91st, and 111th frames because their buffer fullness exceeds the frame skipping threshold (80%$*$buffer size), this indicates that VM8 has less control ability and results in more frame skipping cases. The skipped frames result in gaps on VM8's bit allocation curves, as shown in Fig. 4(c).

From a large number of tests, we find that VM8 is sensitive to initial values of QP, unsuitable initial values of QP can result in frame skipping, while RAPID is robust to initial QPs, which

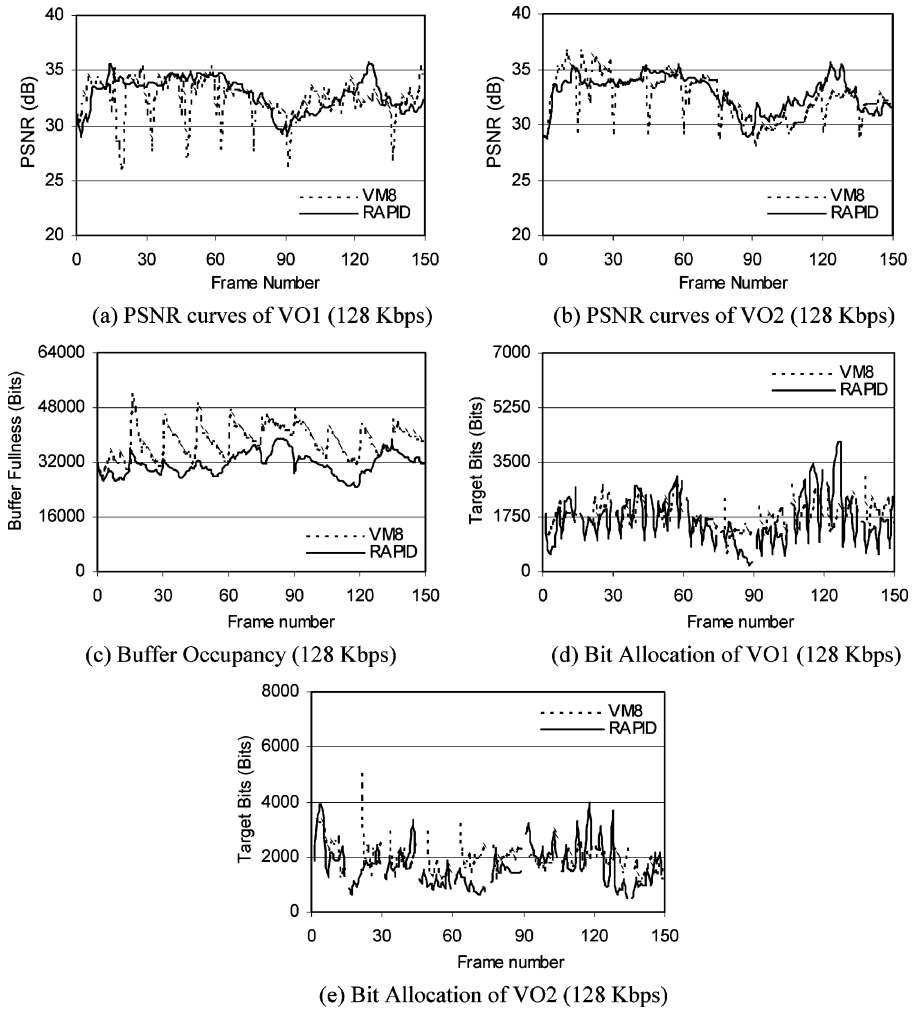


Fig. 6. Performance curves for *News* sequences (QCIF, 2VOs, 30 fps, 128 kbps, IP…IP).

can work within a wide range of initial QP values mostly without frame skipping. In our experiments, initial values of QP are always selected for optimizing VM8, and then these initial values of QP are also used in RAPID. In some cases the frame skipping activity is frequent in VM8 solution, especially when the target bit rate is very low. However, RAPID can deliver good performance without or fewer frame skipping under the same conditions.

Besides the objective and quantitative comparisons of simulation results, subjective tests also exhibit improvement due to the significant reduction of frame skipping, and the motion continuity is maintained.

Investigating the bit allocation curves, RAPID fluctuates more than VM8 does, which is due to the different bit allocation strategies: the target bits of the current frame in VM8 is set to 5% of the number of bits used for coding the previous frame plus 95% of the average number of the remaining bits per frame, without taking into account the real complexity of the current frame, thus the target number of bits for each frame does not vary according to its characteristics, this may cause visual quality varying since the complexity of the video sequence may change along the coding time, while RAPID considers the current frame's complexity during target bit estimation, and allows the target bits among frames varying according to the complexities, trying to smooth the quality fluctuation.

From Fig. 4(a) and (d), we observe that in the VM8 algorithm, intra-coded frames typically have lower qualities than those of inter-coded frames, and there are large fluctuations in PSNR curves. This may due to the less efficient bit allocation strategy of VM8. For example, in Fig. 4(c), even though the 15th I-frame can obtain obvious higher target bits (24376 bits) than its nearby inter-coded frames, its PSNR is 29.99 dB, significantly lower than its neighbor inter-coded frames (33.93 and 32.81 dB for the 14th and 16th frames). In addition, we notice that, in some cases, the target bits for I-frames become fewer and fewer along the coding process due to insufficient remaining bits available, sometimes they are almost equal to the target bits for the nearby inter-coded frames. Especially when the target number of bits for both I-frames and inter-coded frames are fewer than the lower bound, VM8 impose the lower bound of target bits to them, hence both I-frames and inter-coded frames obtain the same number of target bits, this may cause larger quality degradation for I-frames. Meanwhile the PSNR curves of RAPID are smoother, indicating RAPID can handle I-frames more efficiently. This is because we consider the frame's complexity during target bit allocation, and give up estimating target bits for I-frames; instead we just directly predict their QPs by (13)

TABLE IX
PERFORMANCE OF SCENE-CHANGE PROCESSING

Video Sequence	Rate Control Algorithm	Bit Rate (kbps)		# Coded frames		PSNR (dB)
		Target	Actual	Target	Actual	
Coastguard-Mother_daughter (IPPP…PPP, scene change at 66th frame)	VM8	64	64.33	150	145	35.00
	RAPID	64	64.30	150	150	35.33
Foreman_Train (IPPP…PPP, scene change at 82th frame)	VM8	128	129.91	100	93	34.96
	RAPID	128	127.93	100	100	35.21
Mobile_Stefan (IPPP…IPPP, scene change at 108th frame)	VM8	256	256.35	150	148	29.14
	RAPID	256	255.61	150	150	29.51
Stefan_Mobile (IPPP…IPPP, scene change at 108th frame)	VM8	256	256.59	150	141	30.39
	RAPID	256	254.41	150	150	30.74

and (14). Due to no target bit allocations for I-frames, the bit allocation curves of RAPID are not continuous and gaps occur at I-frames' positions, as shown in Fig. 4(c).

C. Multiple Object Rate Control

For MVO RC, the target number of VOPs to be encoded is 150, all the sequences are in QCIF format and encoded at 30 VOP/s with different temporal prediction structures.

1) Only first VOP is I-VOP and the remaining VOPs are all P-VOPs (IPPP…PPP).
2) Both I-VOP and P-VOP are used, the intra period is set to 15 VOPs (IPPP…IPPP).

Table VII shows results for structure (1), while Table VIII and Fig. 6 present results coding in structure (2).

The results for MVO encoding with both structures also indicate that the performance of RAPID is better than or at least equal to the VM8 solution, similar to the situations in the single object case.

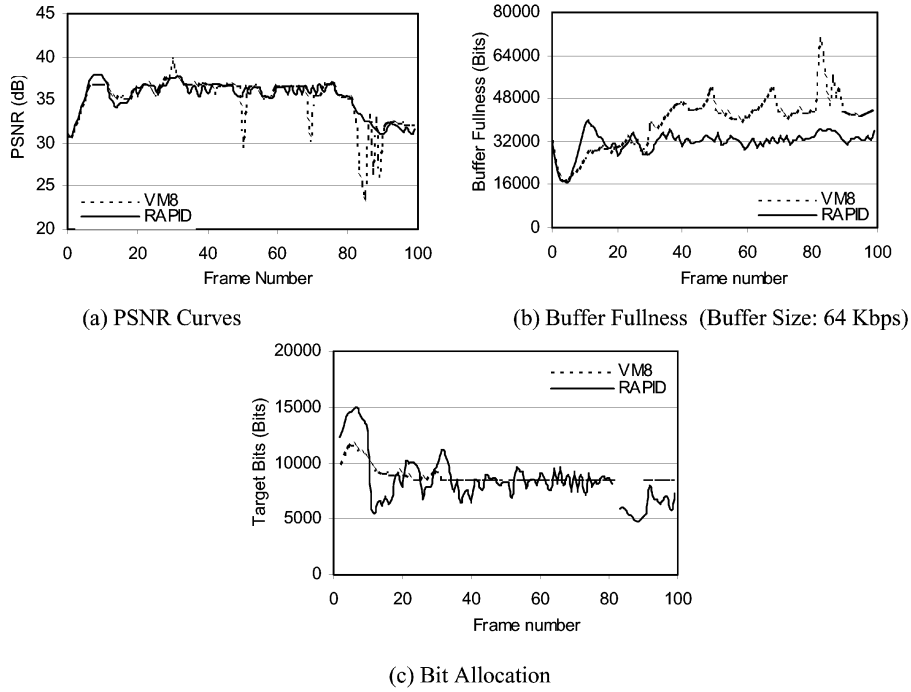
One may notice that in some cases, due to the big PSNR gap between two objects in VM8, the PSNR of one object in VM8 is much higher than that of the same object in RAPID, while the other object's PSNR in VM8 is lower than the same object's in RAPID, thus RAPID effectively decreases quality gaps between objects. For example, VO1 in the *Container* sequence is a moving big boat while VO5 is a very small moving American flag whose size is only one MB, one can see from Table VII, when the target bitrate is 112 kbps, the average PSNR of VO1 is as low as to 31.81 dB, and VO5's PSNR is as high as 47.32 dB using VM8 RC, the quality difference between these two objects is very large (15.51 dB); however, under the same settings, the VO1's PSNR is 33.82 dB while VO5's PSNR is 33.87 dB using RAPID, and the quality gap has been effectively reduced to 0.05 dB. Thus, RAPID obtains a balanced coding quality, indicating our weight adjustment among MVOs is very useful. In other examples (*Bream* and *Children*), RAPID also tries to avoid that background objects have excellent qualities while foreground objects have low qualities.

D. Scene-Change Processing

In order to test scene-change handling abilities of VM8 and RAPID, combined QCIF sequences are used, and the frame rate is 15 fps. For example, for the combined sequence *"Mobile-Stefan",* the first 107 frames are from the "Mobile" and the remaining 43 frames are from the *"Stefan"*, thus, scene change happens at the 108th frame. If the number of intra macroblocks exceeds *30%* of the total number of macroblocks in a frame [27], this frame is regarded as a scene-change frame. Examining the results in Table IX, we can see that RAPID can better deal with scene change without frame skipping as compared with VM8. In Figs. 7(a) and 8(a), VM8 performs poorly at the scene-change frame and its subsequent frames, since it only utilizes information obtained from previously coded frames in estimating target bits for the current frame, when a scene change occurs, information obtained from previous coded frames is no longer suitable for the current frame and causes visual quality degradation in the frames following the scene change. However, the visual qualities at the scene-change frame and its following frames in RAPID are improved. In addition, one can see the buffer overflows at the scene-change frame in Fig. 7(b) for VM8. As a result, RAPID generally obtains higher average PSNRs than VM8 through the whole combined sequences. These results show RAPID improves the ability to deal with scene change and can get better visual quality. Figs. 7(c) and 8(c) are the bit allocation figures for the scene-change sequences.

VI. CONCLUSION

In this paper, we have proposed a rate control scheme for efficient bit allocation for MPEG-4 video coding, which includes a number of ideas: our scheme considers the coding complexities for both objects and frames, and then performs bit allocation among frames and among objects within a frame based on coding complexities; A PID buffer control mechanism is used to promote the control ability; More important, the algorithm performs a lot of feedback adjustments to improve the forecasting accuracy, such as: weight adjustment for frame types, weight adjustment among multiple objects, QP adjustment for I-frames. The performance results for both single VO and multiple VOs encoding authenticate that RAPID outperforms the VM8 solution by: 1) providing more accurate rate regulation; 2) achieving better picture quality; 3) depressing quality fluctuation; 4) balancing PSNRs among both frames and multiple VOs; 5) maintaining a more stable buffer level and reducing frame skipping; and 6) improving the capability to deal with scene change. In this paper, some parameters are fixed and set empirically. Regarding future work directions, we will continue our research on developing intelligent methods to automatically estimate these parameters from the data to be encoded, such as dynamically deciding the sliding window size, adaptively changing control

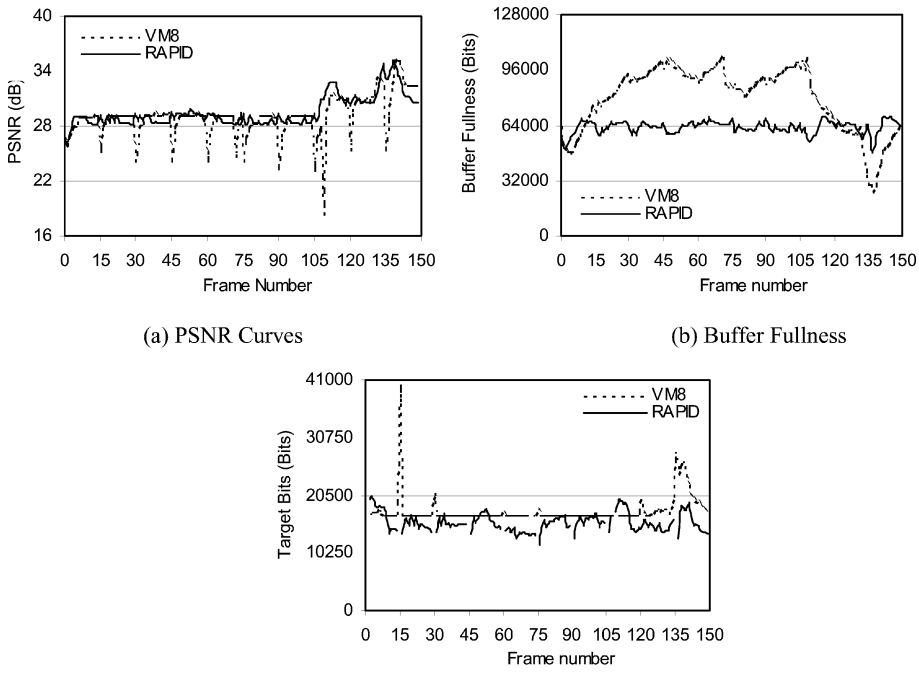


(a) PSNR Curves

(b) Buffer Fullness (Buffer Size: 64 Kbps)

(c) Bit Allocation

Fig. 7. Encoding performance of the combined "Foreman and Train" sequence with scene change at the 82th frame (QCIF, 128 kbps, IPPP…PPP, the intra-period is 150 frames).



(a) PSNR Curves

(b) Buffer Fullness

(c) Bit Allocation

Fig. 8. Encoding performance of the combined "Mobile and Stefan" sequence with scene change at the 108th frame (QCIF, 256 kbps, IPPP…IPPP, the intra-period is 15 frames).

parameters along the coding procedure, exploring more accurate models and better adaptation methods, and developing more advanced rate control structure.

REFERENCES

[1] *Overview of the MPEG-4 Standard*, Doc. ISO/IEC JTC1/SC29/WG11 N2725, R. Koenen, Ed., Mar. 1999.

[2] P. Nunes and F. Pereira. (1999, May) Object-based rate control for the MPEG-4 visual simple profile. Proc. Workshop Image Analysis for Multimedia Interactive Services (WIAMIS'99), Berlin, Germany. [Online] Available: http://amalia.img.lx.it.pt/~fp/artigos/WIAMIS99.DOC

[3] 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.

[4] 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.

[5] L.-J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 446–459, Aug. 1998.
[6] B. Tao, B. W. Dickinson, and H. A. Peterson, "Adaptive model-driven bit allocation for MPEG video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 147–157, Feb. 2000.
[7] *"MPEG Video Test Model 5," Draft*, ISO/IEC JTCI/SC29/WG11, MPEG93/457, Apr. 1993.
[8] K. Oehler and J. L. Webb, "Macroblock quantizer selection for H.263 video coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Oct. 1997, pp. 365–368.
[9] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 172–185, Feb. 1999.
[10] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate-distortion modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 246–250, Feb. 1997.
[11] H.-J. Lee, T. Chiang, and Y.-Q. Zhang, "Scalable rate control for very low bitrate video," in *Proc. 1997 Int. Conf. Image Processing*, vol. 2, Oct. 1997, pp. 768–771.
[12] *MPEG-4 Video Verification Model V8.0*, ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG97/N1796, July 1997.
[13] *Coding of Moving Pictures and Associated Audio MPEG 97/M1631*, ISO/IEC JTC1/SC29/WG11, Feb. 1997.
[14] A. Vetro, H. Sun, and Y. Wang, "MPEG-4 rate control for multiple video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 186–199, Feb. 1999.
[15] H.-J. Lee, T. Chiang, and Y.-Q. Zhang, "Scalable rate control for MPEG-4 video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 878–894, Sept. 2000.
[16] J. I. Ronda, M. Eckert, F. Jaureguizar, and N. Garcia, "Rate control and bit allocation for MPEG-4," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1243–1258, Dec. 1999.
[17] P. Nunes and F. Pereira, "Scene level rate control algorithm for MPEG-4 video coding," in *Proc. SPIE, Visual Commun. Image Process.*, vol. 4310, 2001, pp. 194–205.
[18] Y. Sun and I. Ahmad, "A new rate control algorithm for MPEG-4 video coding," in *Proc. SPIE, Visual Commun. Image Process.*, vol. 4671, San Jose, CA, Jan. 2002, pp. 698–709.
[19] P. Nunes and F. Pereira, "Rate control for scenes with multiple arbitrarily shaped video objects," in *Proc. Picture Coding Symp. (PCS'97)*, Berlin, Germany, Sept. 1997, pp. 303–308.
[20] 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 Process.*, San Jose, CA, Jan. 1998, pp. 383–394.
[21] A. F. D' Souza, *Design of Control System*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[22] C. L. Phillips and R. D. Harbor, *Basic Feedback Control Systems*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[23] J. P. Leduc and O. Poncin, "Quantization algorithm and buffer regulation for universal video codec in the ATM Belgian broadband experiment," in *Proc. Fifth Eur. Conf. Signal Processing (EUSIPCO)*, Barcelona, Spain, 1990, pp. 873–876.
[24] S. Park, Y. Lee, and H. Chang, "A new MPEG-2 rate control scheme using scene change detection," *ETRI J.*, vol. 18, no. 2, pp. 61–74, Jul. 1996.
[25] L.-J. Luo, C.-R. Zou, and Z.-Y. He, "A new algorithm on MPEG-2 target bit-number allocation at scene changes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 815–819, Oct. 1997.
[26] *Multiple-VO Rate Control and* B-VO *Rate Control*, Doc. ISO/IEC JTC1/SC29/WG11 M2554, July 1997.
[27] E. C. Reed and F. Dufaux, "Constrained bit-rate control for very low bit-rate streaming-video applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 7, pp. 882–889, July 2001.

Yu Sun (S'04) received the B.S. and M.S. degrees in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 1996, and the Ph.D. degree in computer science and engineering from The University of Texas at Arlington in 2004.

From 1996 to 1998, she was a Lecturer in the Department of Computer Science, Sichuan Normal University, China. Since August 2004, she has been an Assistant Professor in the Department of Computer Science, The University of Central Arkansas, Conway. Her main research interests include video compression, multimedia communication, and image processing.

Ishfaq Ahmad (S'88–M'92–SM'03) received the B.Sc. degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 1985, and the M.S. degree in computer engineering and the Ph.D. degree in computer science from Syracuse University, Syracuse, NY, in 1987 and 1992, respectively.

He is currently a Full Professor of computer science and engineering in the Computer Science and Engineering Department, The University of Texas at Arlington (UTA). His recent research focus has been on developing parallel programming tools, scheduling and mapping algorithms for scalable architectures, heterogeneous computing systems, distributed multimedia systems, video compression techniques, and web management. His research work in these areas has been published in over 125 technical papers in refereed journals and conferences, Prior to joining UTA, he was an Associate Professor in the Computer Science Department at Hong Kong University of Science and Technology, Hong Kong, where he was also the Director of the Multimedia Technology Research Center, an officially recognized research center that he conceived and built from scratch. The center was funded by various agencies of the Government of the Hong Kong Special Administrative Region as well as local and international industries. With more than 40 personnel including faculty members, postdoctoral fellows, full-time staff, and graduate students, the center engaged in numerous R&D projects with academia and industry from Hong Kong, China, and the U.S. Particular areas of focus in the center are video (and related audio) compression technologies and videotelephone and conferencing systems. The center commercialized several of its technologies to its industrial partners worldwide.

Prof. Ahmad has participated in the organization of several international conferences and is an Associate Editor of *Cluster Computing*, *Journal of Parallel and Distributed Computing*, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *IEEE Concurrency*, and *IEEE Distributed Systems Online*. He was the winner of Best Paper Awards at Supercomputing'90 (New York), Supercomputing'91 (Albuquerque, NM), and the 2001 International Conference on Parallel Processing (Spain).