Multimed Tools Appl (2015) 74:6623–6639
DOI 10.1007/s11042-014-1913-6

A direct non-buffer rate control algorithm for real time video compression

Yu Sun · Zhidan Feng · Reshma R. Ginnavaram

Published online: 9 March 2014

Abstract Rate control (RC) is crucial in controlling compression bitrates and encoding qualities for networked video applications. In this paper, we propose a direct non-buffer real-time rate control algorithm for video encoding, which has two unique features. First, unlike traditional algorithms which adopt buffers in rate control, the proposed algorithm does not use a buffer in rate regulation which can reduce the delay and improve real-time response. Second, we propose a new Proportional-Integral-Derivative (PID) bit controller to directly control encoding bitrates. In addition, we also develop a simple but effective method for real-time target bit allocation. To the best of our knowledge, this is the first work that conducts video rate control without using a buffer. Our extensive experimental results have demonstrated that the proposed algorithm outperforms the MPEG-4 rate control algorithm by achieving more accurate rate regulation and improving overall coding quality.

Keywords Rate control · PID bit controller · Bit allocation · Video compression

1 Introduction

In networked video applications, compressed video bitstreams have to be transmitted over heterogeneous networks that have limited and/or time-varying network bandwidths. Due to the properties of entropy coding and variable content of successive video frames, the encoding rate from an encoder is inherently variable [1]. There exists a problem when transmitting compressed streams over constrained network channels: if an encoding rate is larger than a network bandwidth, this may lead to congested networks, video data loss, and thus, cause an interruption to video smoothness. On the other hand, if an encoding rate is smaller than a network bandwidth, this may result in unnecessary quality degradation and inefficient use of available network bandwidths. Rate control (RC) must then be adopted to regulate encoding rates of a

Y. Sun (✉) · R. R. Ginnavaram
Department of Computer Science, University of Central Arkansas, Conway, AR, USA
e-mail: yusun@uca.edu

Z. Feng
IT Research Group, University of Arkansas for Medical School, Little Rock, AR, USA
e-mail: zfeng@uams.edu

video encoder in order to meet variable network bandwidths while obtaining optimal encoding quality. Therefore, it plays a crucial role in video compression and communication.

Rate control approaches normally place a buffer between an encoder and a network channel to smooth out the variable bitrate output from an encoder. A buffer stores encoding bits produced by an encoder, while a channel removes bits from the buffer and transmits them. However, when the buffer is full, the encoder must cease generating bits by dropping frames thereby causing an interruption to video smoothness. 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 use of a buffer in a video encoder definitely introduces delay. For example, if a buffer size is set to half of the target bitrate, then the maximum accumulated delay is 500 ms [3]. A large buffer size tends to allow a smoother video but causes a longer delay, while a small buffer size guarantees a low delay but may be more likely to skip frames due to overflow [1]. It is obvious that the delay would be reduced to a large degree if no buffer is used in an encoder.

Generally, most RC algorithms consist of the following five major steps: (1) Target bit allocation: given the target bitrate by the network bandwidth, estimate the initial target number of bits to code the current frame; (2) Buffer control: the initial target number of bits is further adjusted based on the buffer fullness so as to avoid buffer overflow/underflow; (3) Determine the encoding parameter - Quantization Parameter (QP) to achieve the target bits based on the Rate-Distortion (R-D) model, which describes the relationships among target bits (R), distortion (D) and QP; (4) Use QP to encode the current frame; (5) Post-encoding processing, such as collects encoding results to update a R-D model.

1.1 Related work

There are three important issues that need to be carefully addressed when designing a RC algorithm. The first one is target bit allocation, the second one is buffer control, and the third one is how to determine QP to achieve the target bitrate. RC has been extensively studied and a number of algorithms have been developed for various video coding standards and applications, such as TM5 for MPEG-2 [9], TMN8 for H.263 [2], VM8 for MPEG-4 [3], JVT-G012 [6] and JVT-W042 [4] for H.264/AVC, and JVT-W043 for H.264/SVC [5].

1.1.1 Target bit allocation

A non-real time bit allocation method has been used in some RC algorithms [3, 9, 14, 15, 19, 21, 22], including MPEG-2 TM5 and MPEG-4 VM8. This method generally allocates target bits to a frame according to the actual bits used in encoding the previous frame, the remaining number of frames to be encoded, and the remaining bits available for the whole encoding sequence. Its limitation is that it must know the total number of frames to be encoded and the total available encoding bits for a whole video sequence before compression. Obviously, it is not suitable for real-time RC since for the scenario of real-time encoding, it is impossible to know the total number of frames to be encoded and the total available encoding bits in advance. Based on target bitrates and frame rates, a real-time bit allocation method has also been adopted in [2, 4–6, 16, 17, 23], e.g. TMN8 for H.263. Some bit allocation methods, including VM8 and TMN8, don't consider frame complexities. That is to say, the target number of bits for each frame is nearly constant throughout the video sequence. This will cause fluctuating encoding quality among frames since each frame's complexity is different. To reduce quality fluctuations, some RC algorithms [14, 15] perform target bit allocation by considering frame complexity.

Although different bit allocation methods have been used in RC, one common property is that all of these methods use buffer control to adjust the initial target bits allocated to a frame.

1.1.2 Buffer control

The objective of buffer control is to maintain buffer fullness around the target buffer level in order to achieve accurate bitrates while reducing the chances of buffer overflow or underflow. If the buffer occupancy exceeds the target level, the initial target bits allocated a frame are decreased to some extent. Similarly, if it is below the target level, the initial target bits are increased by some degree. Traditional buffer control approaches, including MPEG-2 TM5, H.263 TMN8, MPEG-4 VM8, H.264/AVC JVT-G012/W042 and H.264/SVC W043, adopt a simple proportional buffer controller [2–9, 11, 12, 19–22] whose control ability is not effective enough [14, 15, 17, 23]. To overcome this weakness, we applied "Proportional + Integral + Derivative (PID)" controller in the automatic control field to video compression, and firstly proposed a novel PID buffer controller in the literature [14]. Due to its robust control ability, PID-based buffer controller has recently been used in video RC [13–17, 23–26].

1.1.3 Other RC approaches

In addition to the above works, various RC researches, which focus on different aspects and diverse applications, have been proposed in the literature. Recently, Ou et al. [11] investigated the impact of frame rate, quantization and perceptual quality of a video, and proposed a quality model which considers a spatial quality factor and a temporal correction factor. Based on this quality model, Ma et al. [8] proposed a rate model in terms of the quantization stepsize and frame rate, and applied it [11] for scalable bitstream adaptative and frame rate adaptive RC. An effective incremental RC algorithm for H.264 Scalable Video Coding (H.264/SVC) has been developed to control each SVC layer's encoding rate to meet different network bandwidths [24]. Ruan et al. proposed a RC algorithm based on the attention selectivity properties of Human Visual System [12]. Tian et al. [20] developed an accurate intra-only RC scheme which includes a new complexity measurement and a R-D model. The research in [23] integrated H.264/SVC technology with Multiple Input Multiple Output (MIMO) wireless systems and proposed a novel joint H.264/SVC-MIMO RC algorithm for wireless video applications. To improve continuity for video streaming, Tan and Chou [18] develop a frame rate optimization framework to jointly adjust the encoder frame generation rate and the decoder playout frame rate.

1.1.4 Summary

So far, all the existing research efforts of RC adopt buffers and buffer control to adjust the initial number of target bits allocated to a frame. Further, although the famous PID controller has been recently used in video RC, it has been only applied in buffer control. According to the best of our knowledge, no prior works have attempted to achieve RC without using a buffer, and none of the existing works have used PID controller in directly controlling target bits and actual encoding bits.

1.2 Overview of this work

With the increasing network bandwidths and rapidly growing demand for real-time networked video applications, such as YouTube Live, HDTV live streaming, mobile robotic-video explorations, battlefield monitoring and boundary surveillance, video compression and RC with real-time and low-delay are becoming more and more important recently. Base on this motivation, in this research, we investigate new and effective RC approaches, with the objective to improve real time response, reduce delay and enhance RC performance.

As illustrated in Fig. 1(a), traditional RC algorithms set up buffers and use buffer control to adjust the initial number of target bits allocated to a frame. So far, the famous PID controller has only been used in controlling buffer and has never been used in controlling the encoding bitrate directly. In this research, we have proposed a new RC algorithm which applies the PID controller to directly control encoding bitrate without using a buffer (Fig. 1(b)). Unlike traditional RC algorithms, the new algorithm doesn't have a buffer and doesn't use "Buffer Control". Instead, it adopts "PID Bit Control" to directly regulate actual encoding bitrates to be closed to target bitrates without exploiting a buffer. The encoding bits output from a video encoder will be output to the communication channel directly. During the transmission, the encoding bits will be taken care by network management/scheduling and they might be smoothed in network buffers if needed.

Generally, the use of a buffer in a video encoder always introduces more or less delay depending on the buffer size, and this delay is always in adverse to real-time video applications. Therefore, without using a buffer, our proposed RC algorithm can reduce the delay and improve the real-time response. Our extensive experimental results have demonstrated that the proposed algorithm works effectively and outperforms VM8 RC algorithm adopted by MPEG-4 compression standard.

The major contributions of this research are summarized below:

(1) This research develops a new RC framework to regulate encoding bitrates without using a buffer
(2) This research proposes a novel PID bit controller to directly control encoding bitrates.

The remainder of this paper is organized as follows: Section 2 presents the details of the proposed RC algorithm. A summary of the algorithm is provided in Section 3. Section 4 presents the experimental results to demonstrate the performance of the proposed algorithm. Finally, Section 5 concludes this research and discusses the future work.

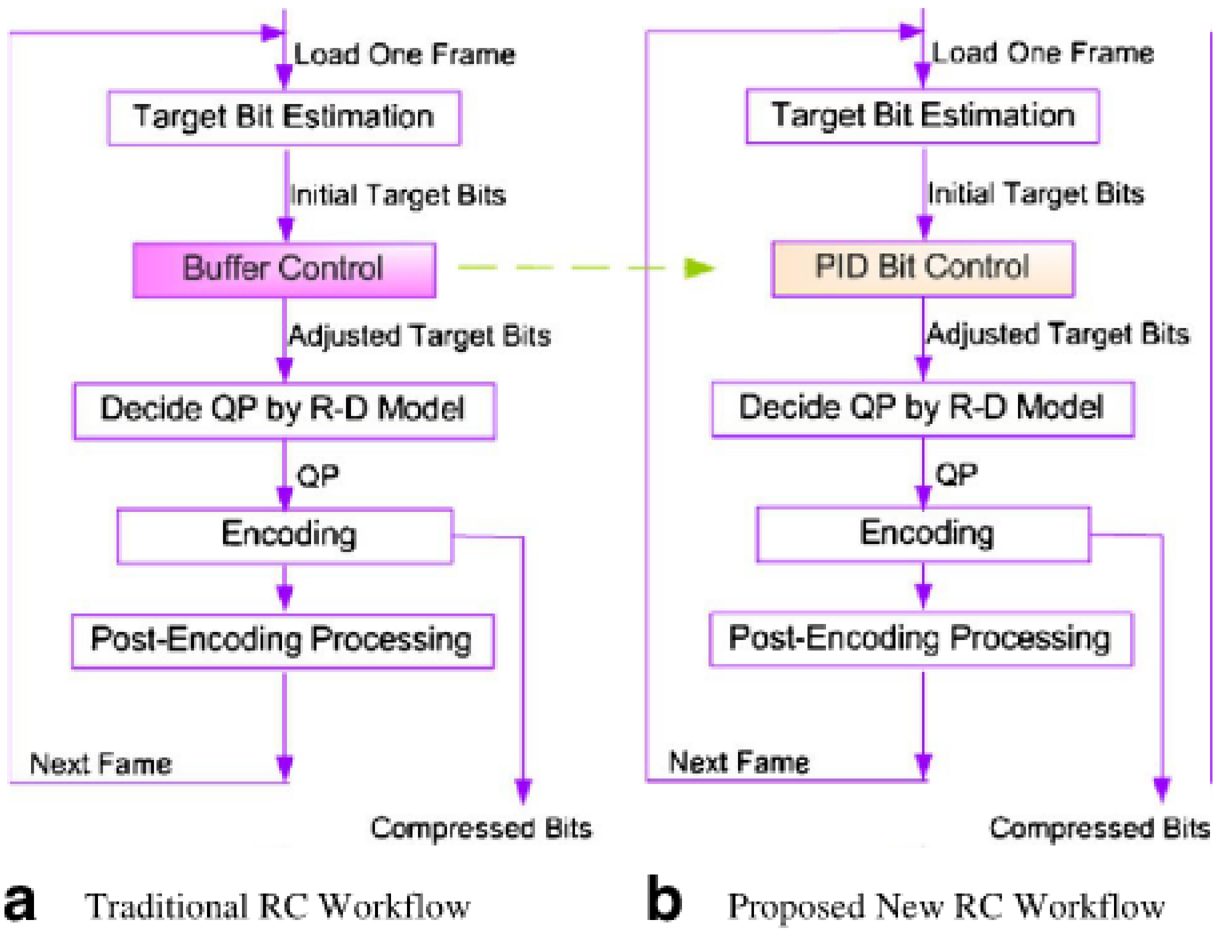


Fig. 1 Rate control workflows

2 The proposed un-buffered real time bitrate control algorithm

In this section, we present our proposed RC algorithm for video compression.

2.1 Initialization stage

The initialization stage includes setting up encoding parameters, such as initial values of QP for the first I-, B- and P-frame, model parameters, frame rate, target encoding bitrate, etc. This stage also initializes various historical values which are saved after encoding a frame, i.e. actual encoding bits and the estimated target bits for a frame.

2.2 Initial target bit estimation

The target bit allocation method proposed in VM8 RC algorithm for MPEG-4 and some other RC algorithms have to know the total number of frames to be encoded in advance (before encoding) [3, 9, 14, 15, 19, 21, 22]. In this way, the algorithms can calculate the total target bit budget available to encode all frames for a video sequence. Therefore, these algorithms are not suitable to be applied in real-time.

Here, we propose a simple but effective bit allocation method. It can be performed in real time. For every I-frame, B-frame and P-frame, based on the encoding type of the current frame, its initial target number of bits T_t is initially set to a weighted average bit count as follows:

$$T_t = \alpha(K)_t \cdot \frac{R_r}{\alpha(I)_t \cdot N_I + \alpha(B)_t \cdot N_B + \alpha(P)_t \cdot N_P} \tag{1}$$

where N_I, N_P and N_B are the numbers of I-frames, P-frames and B- frames within a second, $\alpha(I)_t$, $\alpha(B)_t$ and $\alpha(P)_t$ are their weight factors, R_r is the target bitrate which is given by the current network bandwidth, $\alpha(K)_t$ is $\alpha(I)_t$ if the current frame is an I-frame, $\alpha(K)_t$ is $\alpha(B)_t$ if the current frame is a B-frame, while $\alpha(K)_t$ is $\alpha(P)_t$ if the current frame is a P-frame. Since R_r, N_I, N_P, N_B are constants, and $\alpha(K)_t$ are already know before encoding each frame, we can see from Eq. (1) that our approach for target bit allocation only depends on the current available information, but not based on the future or look ahead information such as the total number of frames to be encoded. Thus, our algorithm is suitable for real-time control.

2.3 Target bits adjustment based on the PID bit controller

Traditional RC algorithms set up buffers and use the levels of buffer fullness to further adjust the initial number of target bits obtained in Section 2.2. Our motivation is that, as we know buffer always brings delay at some degree, if a buffer is not used during the RC process, the delay can be reduced in a large degree and the real time response will be improved a lot. Therefore, we propose to control encoding bits to be closed to the target bits directly without using a buffer. The compressed bits output from an encoder will be output to the communication channel directly. In this way, the delay will be reduced largely on the encoder side and we can achieve real time RC. During the transmission, the encoding bits will be taken care by network management/scheduling and they might be smoothed in the network buffer if needed.

The most popular and generic feedback controller in the automatic control systems [14] is the PID controller, which is very popular mainly due to its simplicity and excellent performance in a wide range of operating conditions. It is especially suitable in controlling an unpredictable or imprecise process. Since future video frames and their characteristics cannot be predicted accurately, the video encoding process is unpredictable and imprecise. Therefore, the PID controller is suitable for controlling the process of video compression. In [14], we first applied the PID controller in video encoding and developed a robust PID buffer controller. After that, PID buffer controller has been used in video RC [13–17, 23–26]. In this research, we propose to apply the PID technique in controlling bits directly without using a buffer.

As shown in Fig. 2, the initial target bit budget obtained in Section 2.2 is further adjusted based on the PID bit controller so as to get more accurate target bit estimation. The aim of the PID bit control is to keep the actual encoding bitrate around the target bitrate so as to fully utilize the channel bandwidth while obtaining optimum encoding quality and reducing the chances of network congestion. If the number of actual encoding bits for the previous encoding frame exceeds the number of its target bits, then the budget of target bits for the current frame is decreased to some extent through the PID bit controller; otherwise, the budget of target bits is increased by some degree.

The control goal is to keep the actual encoding bits around the target bits, and minimize the deviation between the target bitrate and the actual encoding bitrate. The error signal, which measures the difference between the target number of bits T_t and the number of actual bits used for encoding the current frame A_t at time t, is defined as:

$$E_t = T_t - A_t, \tag{2}$$

where T_t is the target number of bits and A_t is the actual encoded bits at time t.

This error signal is sent to the PID bit controller and the bit adjustment PID_t can be obtained by:

$$PID_t = K_p \cdot \left(E_t + K_i \cdot \int_0^t E_\tau \cdot d\tau + K_d \cdot \frac{dE_t}{dt} \right), \tag{3}$$

where K_p, K_i and K_d are the proportional, integral, and derivative control parameters. There are three terms in Eq. (3). The first term is the *proportional component*, the primary component of the controller, which reduces the error between the actual bitrate and the target bitrate, but it cannot fully eliminate this error. The second term, *integral controller*, has the effect of eliminating the steady state error by this way: as the error lasts, it can gradually enhance the control strength. But it may cause the transient response worsening. The third term, *derivative controller*, is able to increase the system stability, reduce the overshoot and improve the transient response. Since the PID controller combines three controllers, Proportional + Integral + Derivative, it combines

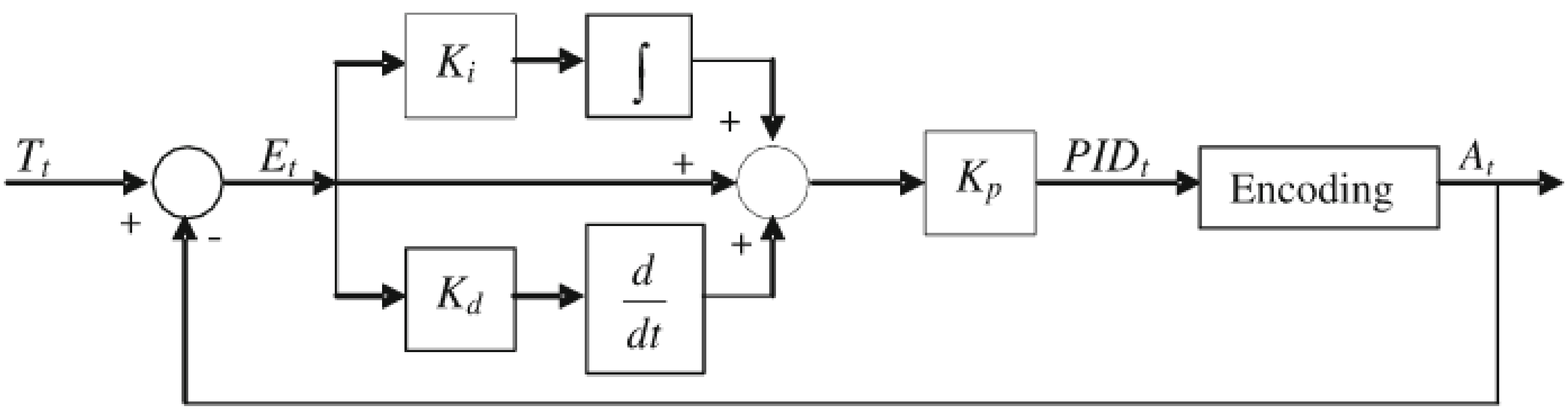


Fig. 2 The PID bit controller

the advantages of each individual controller, and thus, improves both the transient and the steady-state response [14].

Once we obtain PID_t, the target bits T_t can be further adjusted by:

$$T_t := T_t + PID_t. \tag{4}$$

To maintain a minimum acceptable visual quality, the algorithm must allocate a minimum number of bits to each frame (lower bound):

$$T_t = \max\left\{\frac{R_r}{4\cdot F}, T_t\right\}, \tag{5}$$

where R_r and F are the target bitrate and frame rate respectively. This means that, when the target bits allocated to this frame (T_t) is lower than the lower bound (one fourth of the average number of bits per frame), we still distribute the lower bound of bits to this frame. This way can make sure that each frame must obtain the minimum number of bits to maintain the minimum acceptable qualities [14].

For most applications, overflow is much worse than underflow, so each frame can not obtain too many target bits. Correspondingly, the maximum bits allocated to each frame should be more strictly constrained than the minimum one. To avoid buffer overflow, the maximum number of bits for each frame is given as [14]:

$$T_t = \min\left\{\frac{2\cdot R_r}{F}, T_t\right\} \tag{6}$$

2.4 Quantization parameter calculation

The encoding parameters QPs for texture coding are calculated based on the rate-distortion (R-D) model [3, 21]. Once T_t is obtained, the number of target bits for coding the texture of frame, $T_{texture}$, can be calculated by:

$$T_{texture} = T_t - H_{t-1}, \tag{7}$$

where H_{t-1} is the number of bits actually used for coding the motion, shape, and header for the previous encoded frame. We also adopt the following popular R-D model used in MPEG-4 and H.264 RC [3–6, 21]:

$$T_{texture} = \frac{X_1\cdot MAD}{QP} + \frac{X_2\cdot MAD}{QP^2}, \tag{8}$$

where MAD is the Mean Absolute Difference for a frame after motion compensation, X_1 *and* X_2 are the first and second-order model coefficients which are initially set to be *0* and *5000* respectively. Since MAD, X_1, X_2 and $T_{texture}$ are already known, we can easily calculate QP from Eq. (8).

To achieve a smooth coding quality among intra-coded and inter-coded frames, an effective approach has been proposed in [14]. This approach only allocates target bits and computes QPs for B-frames and P-frames but not for I-frames. The QP for an I-frame is directly obtained by using the average QP of its previous inter-coded frames with some adjustment. This method is simple but effective in smoothing coding quality. Here, we use this method in calculating QPs for I-frames as follows:

$$QP_{I,t} = QP_{ave,t} + \beta\_I_t, \tag{9}$$

where $QP_{I,t}$ is the QP of the current I-frame and $QP_{ave,t}$ is the average QP of l inter-coded frames before the current I-frame. Initially, β_I_t is 1.0 and is updated at time t as follows:

$$\beta\_I_t := \beta\_I_t + \frac{PSNR_{I,t1} - PSNR_{ave,t1}}{\lambda}, \tag{10}$$

where t_1 is the coding time of the last I-frame, $PSNR_{I,t1}$ is the PSNR of the last I-frame and $PSNR_{ave,t1}$ is the average PSNR of l inter coded frames before the last I-frame, λ is a tuning parameter.

The reason behind the above formula is in the following. In order to lower its coding quality, the QP for the current I-frame has to be increased when the last I-frame's PSNR is greater than the average PSNR of previously l inter-coded frames. On the contrary, if the PSNR of an I-frame is lower than the average PSNR of inter-coded frames, the QP of this I-frame should be reduced, to improve the coding quality of this I-frame. By doing this, the quality of I-frame will be kept closed to those of its previously inter-coded frames. Here, l and λ are 3 and 16 respectively.

2.5 Encoding

After the encoding parameter, QP, is determined, the encoder uses this QP to encode the current frame. The encoded bits will be sent to the channel directly.

2.6 Updating

After encoding a frame, the encoder updates the R-D model based on the encoding results of the current frame as well as the past frames. X_1 and X_2 are updated by using the linear regression technique. Traditional RC algorithms need to do frame skipping control to avoid buffer overflow. Since our algorithm doesn't use buffer, there is no need to do frame skipping control here.

To achieve a smooth visual quality, after encoding a frame, the weight of this frame type needs to be updated dynamically. Here, we use the approach proposed in [14] to adjust the weights of I-frame ($\alpha(I)_t$) and B-frame ($\alpha(B)_t$) respectively while $\alpha(P)_t$ is fixed to 1.0. The update is conducted in the follow way:

$$\alpha(I)_t = \frac{I_{avebits,t}}{P_{avebits,t}} \cdot e^{\left(\frac{PSNR_{ave\_P,t} - PSNR_{ave\_I,t}}{\gamma}\right)} \tag{11}$$

$$\alpha(B)_t = \frac{B_{avebits,t}}{P_{avebits,t}} \cdot e^{\left(\frac{PSNR_{ave\_P,t} - PSNR_{ave\_B,t}}{\gamma}\right)} \tag{12}$$

where $P_{avebits,t}$, $B_{avebits,t}$ and $I_{avebits,t}$ indicate 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. $PSNR_{ave\_P,t}$, $PSNR_{ave\_B,t}$ and $PSNR_{ave\_I,t}$ are their corresponding average PSNRs. In this research, we use the same empirical values for the window size ($n_I + n_P + n_B$) and γ in [14], which are set to be 30 and 8 respectively.

The above weight adjustment comprehensively considers several factors: average bits used in encoding previous I-frames, P-frames or B- frames, and average coding qualities of previous I-frames, P-frames, or, B- frames. The basic idea is as follows. 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 the 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.

3 Summary of the proposed algorithm

The major steps of the proposed algorithm are summarized as follows:

Step 1. Initialize the parameters for the encoder.
Step 2. Estimate the number of initial target bits for a frame using Eq. (1).
Step 3. Adjust the initial target bits for a frame based on the PID bit controller using Eqs. (2), (3), (4).
Step 4. Set the upper-bound and the lower-bound for the target bits to obtain smooth encoding quality using Eqs. (5) and (6).
Step 5. Calculate a QP for a P-frame, or a B- frame using Eqs. (7), (8) and QPs for an I-frame using Eq. (9).
Step 6. Use QP obtained in step 5 to encode the current frame.
Step 7. After encoding, update the R-D model and adjust the weights using Eqs. (10), (11), (12).
Step 8. Go to step 2 to encode the next frame until the end.

4 Experimental results

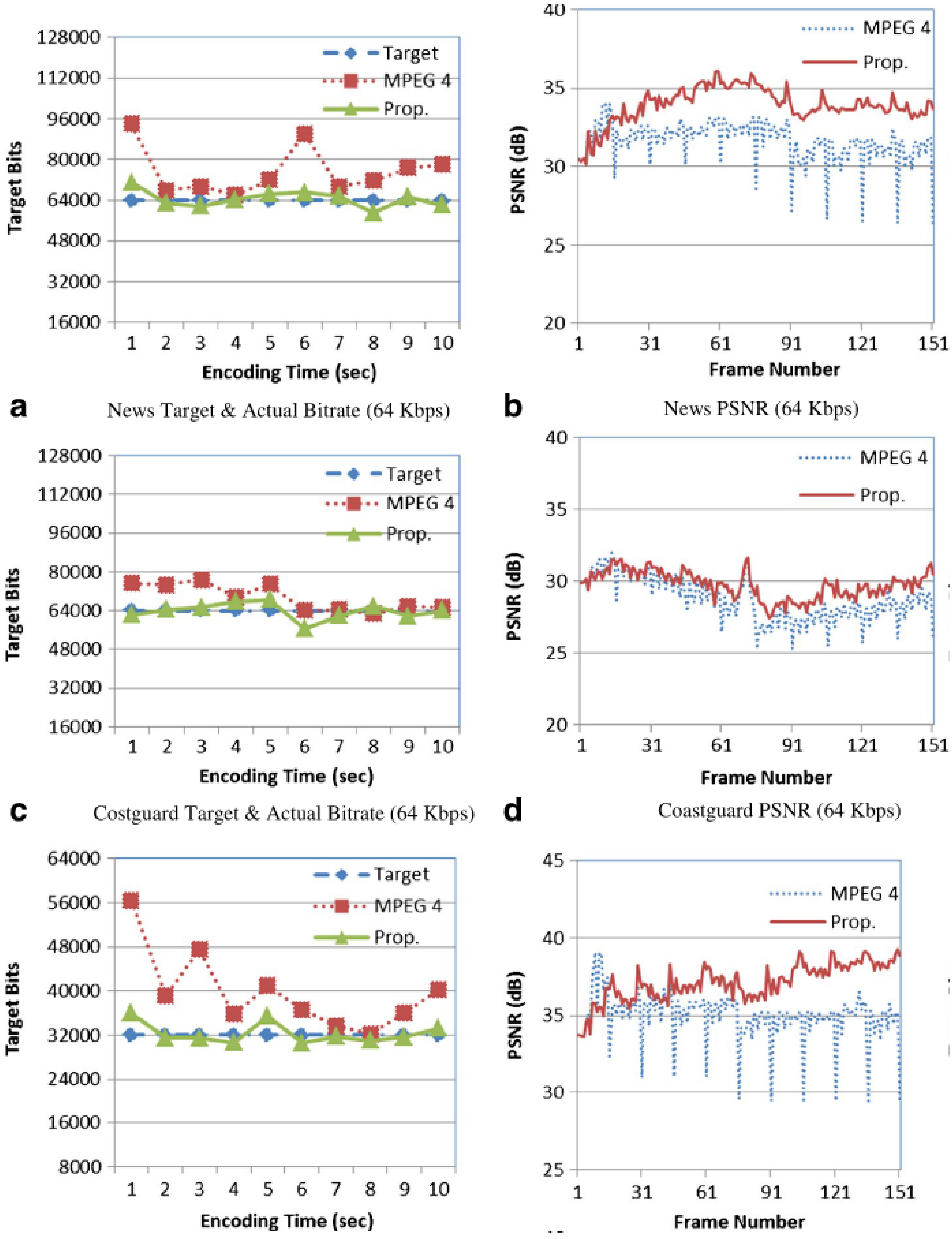
We've implemented the proposed algorithm based on the MPEG-4 encoder [10]. Numerous experiments have been conducted to evaluate the performance of the proposed algorithm. We compare our experimental results achieved here with those obtained using MPEG-4 RC algorithm. For fair comparison, both RC algorithms use the same encoder settings. Since our algorithm doesn't use a buffer and does not have frame skipping control, we disable the frame skipping control of MPEG-4 RC. This is to say, we directly compare the RC abilities between two algorithms without using frame skipping function. Please note that, before encoding, MPEG-4 RC algorithm has already known some useful information, such as the total number of frames to be encoded and the total target bits for the whole encoding sequences, while our algorithm doesn't know all these information in advance. Therefore, the difficulty level of MPEG-4 RC algorithm is lower than that of our algorithm.

We have conducted two sets of experiments with different temporal prediction structures: (1) I-frames, P-frames and B- frames are used with an intra-period of *15*, *2* B-frames are inserted between two reference frames (IBBP…IBBP); (2) Only I- and P-frames are used with an intra-period of *15* (IPP…IPP). PID coefficients K_p, K_i and K_d are empirically set as 0.3, 0.25 and 0.1 respectively. The initial values of $\alpha(I)_t$, $\alpha(B)_t$ and $\alpha(P)_t$ are 3.0, 0.5, and 1.0, respectively. The values of $\alpha(I)_t$ and $\alpha(B)_t$ are dynamically adjusted during the encoding process. All standard test sequences are in *QCIF* (176×144) format and are encoded at 15 frames/s (fps). 150 frames in total or 10 s of sequences are used in testing.

Table 1 Overall performance comparison (Intra Period=15, Encoding Structure IBBP…IBBP)

Sequences	Target bitrate (Kbps)	Actual encoding bitrate (Kbps)		Rate control accuracy (%)		PSNR (dB)		
		MPEG-4	Proposed	MPEG-4	Proposed	MPEG-4	Proposed	PSNR Gain
News	128	132.37	127.10	3.41 %	0.70 %	34.41	38.21	3.8
	96	105.74	95.22	10.15 %	0.81 %	33.37	36.16	2.79
	64	75.68	64.49	18.25 %	0.76 %	31.58	33.91	2.33
Coastguard	256	272.85	251.96	6.58 %	1.58 %	34.70	35.31	0.61
	192	209.53	188.65	9.13 %	1.74 %	33.18	33.72	0.54
	64	69.31	63.68	8.29 %	0.50 %	28.76	29.74	0.98
Akiyo	64	69.18	63.90	8.09 %	0.15 %	37.89	39.96	2.07
	48	53.58	47.20	11.62 %	1.66 %	36.35	38.51	2.16
	32	39.82	32.16	24.44 %	0.49 %	34.99	37.13	2.14
Mobile	512	550.11	505.02	7.44 %	1.36 %	32.91	33.75	0.84
	384	412.20	380.54	7.35 %	0.90 %	30.71	31.89	1.18
	128	142.14	129.75	11.05 %	1.37 %	25.56	26.59	1.03
Stefan	384	403.92	381.58	5.19 %	0.63 %	33.71	34.76	1.05
	256	268.76	254.40	4.99 %	0.63 %	30.85	32.05	1.2
	112	126.17	113.11	12.66 %	0.99 %	27.45	27.89	0.44
Mother_Daughter	128	135.80	126.20	6.09 %	1.41 %	38.79	39.44	0.65
	64	73.70	63.80	15.15 %	0.32 %	34.89	36.54	1.65
	32	42.13	32.16	31.64 %	0.50 %	32.93	34.04	1.11
Train	256	270.26	253.48	5.57 %	0.98 %	33.70	36.11	2.41
	128	131.21	127.73	2.51 %	0.21 %	29.68	32.04	2.36
	64	68.49	65.580	7.01 %	2.47 %	27.14	28.84	1.7
Table	256	273.93	243.03	7.00 %	5.07 %	36.87	36.94	0.07
	128	146.56	125.19	14.50 %	2,20 %	33.21	33.81	0.6
	96	103.99	94.57	8.33 %	1.49 %	31.61	32.74	1.13
Hall	192	215.68	186.51	12.33 %	0.29 %	38.81	39.57	0.76
	128	139.28	125.57	8.81 %	1.99 %	35.30	37.52	2.22
	64	75.84	64.01	18.50 %	0.02 %	31.81	34.39	2.58
Silent	128	210.44	126.04	57.38 %	1.53 %	36.58	35.78	−0.8
	64	108.53	63.55	69.59 %	0.79 %	32.99	33.11	0.12
	32	60.92	31.81	90.38 %	0.61 %	30.83	30.36	−0.47
Children2	192	200.12	189.14	4.22 %	1.49 %	37.79	39.13	1.34
	128	135.78	126.67	6.07 %	1.04 %	35.34	36.79	1.45
	96	100.94	95.27	5.15 %	0.76 %	33.63	35.24	1.61
Bream2	512	522.93	511.91	2.13 %	0.08 %	39.68	40.03	0.35
	384	395.37	386.05	2.96 %	0.53 %	36.67	36.48	−0.19
	256	265.31	257.83	3.64 %	0.72 %	32.39	32.78	0.39

The overall performance results for the temporal prediction structure (1) are reported in Table 1. As we can see that each sequence in Table 1 is tested using different bitrates, i.e.: a relatively higher bit-rate and a lower bit-rate.



a News Target & Actual Bitrate (64 Kbps)

b News PSNR (64 Kbps)

c Costguard Target & Actual Bitrate (64 Kbps)

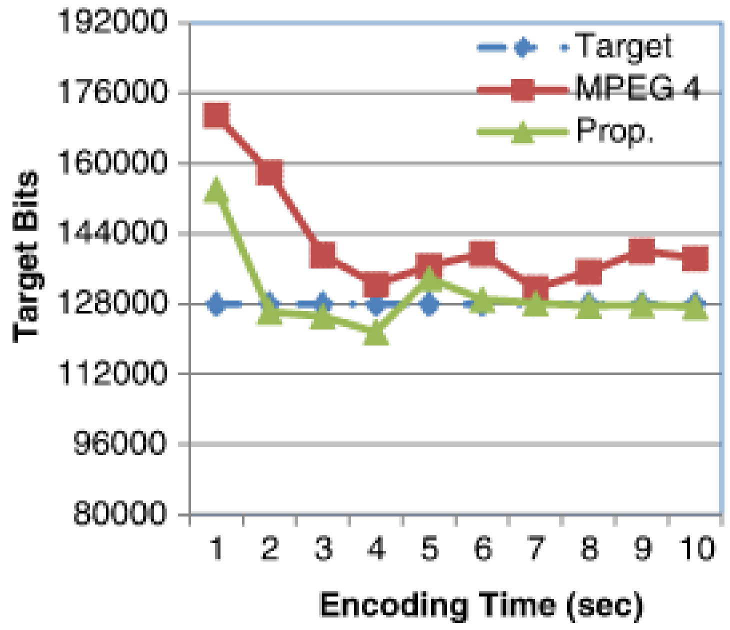
d Coastguard PSNR (64 Kbps)

e Akiyo Target & Actual Bitrate (32 Kbps)

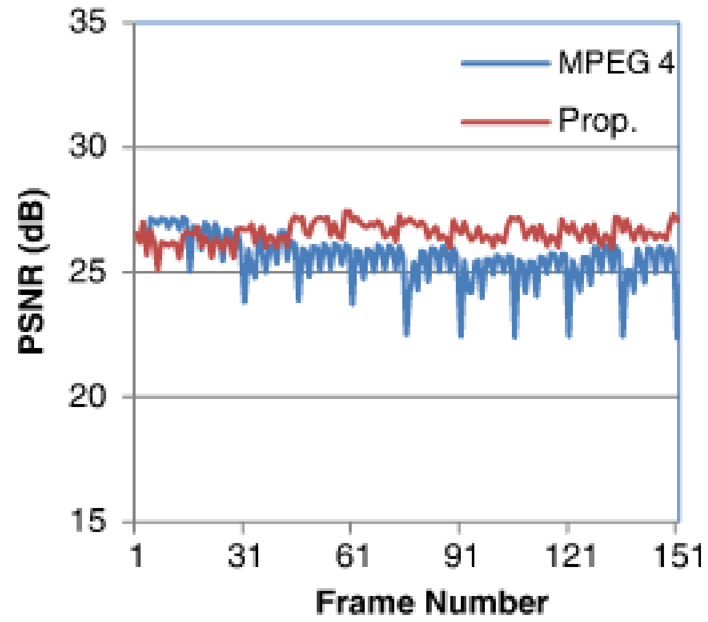
f Akiyo PSNR (32 Kpbs)

Fig. 3 Experimental results for video seuqences encoded at various bitrates (IBBP…IBBP)

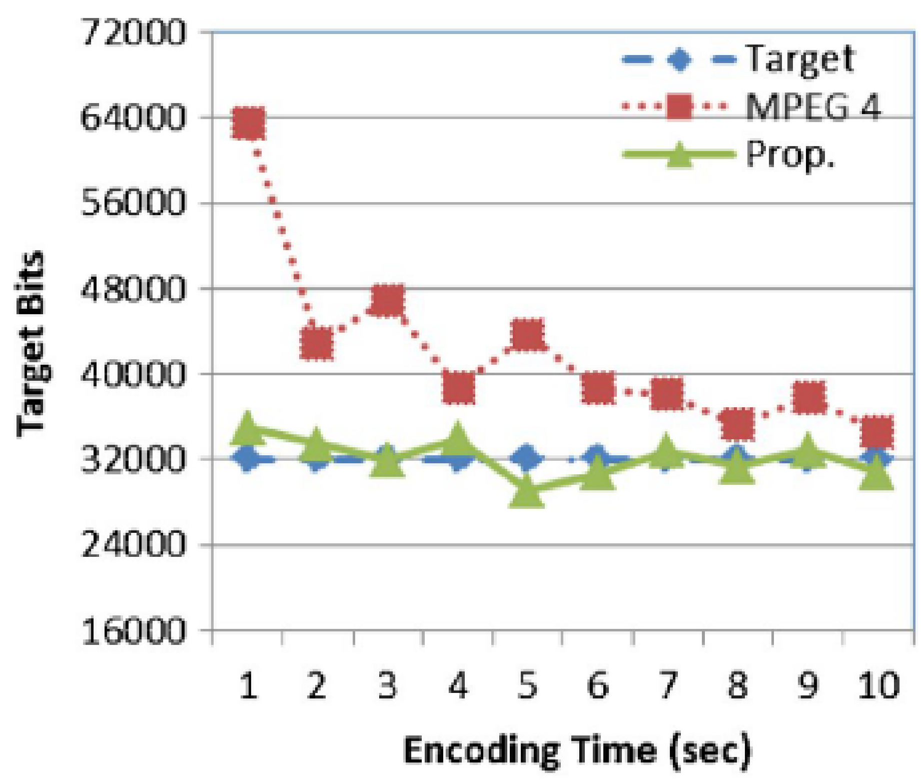
The RC accuracy in Table 1 is measured by: *(1-|(T-A)/T|)× 100 %*, where *T* and *A* are the average target bitrate and average actual coding bitrate. The lower percentage value represents the higher RC accuracy. By examining the results in Table 1, it is clear that the proposed algorithm achieves more accurate target bitrates, higher RC accuracies with higher average



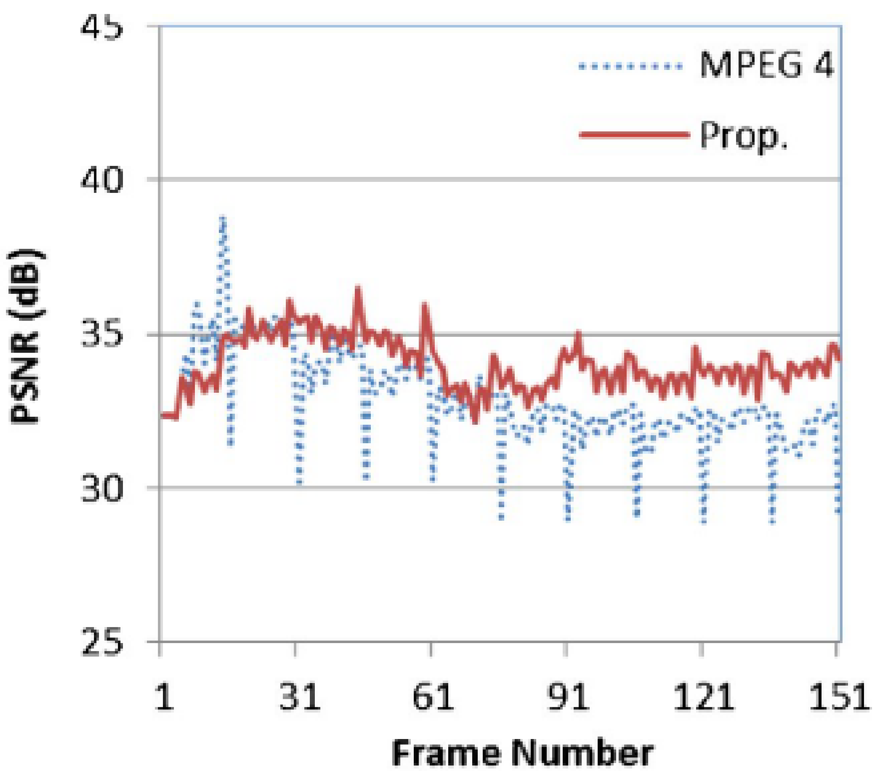
g Mobile Target & Actual Bitrate (128 Kbps)



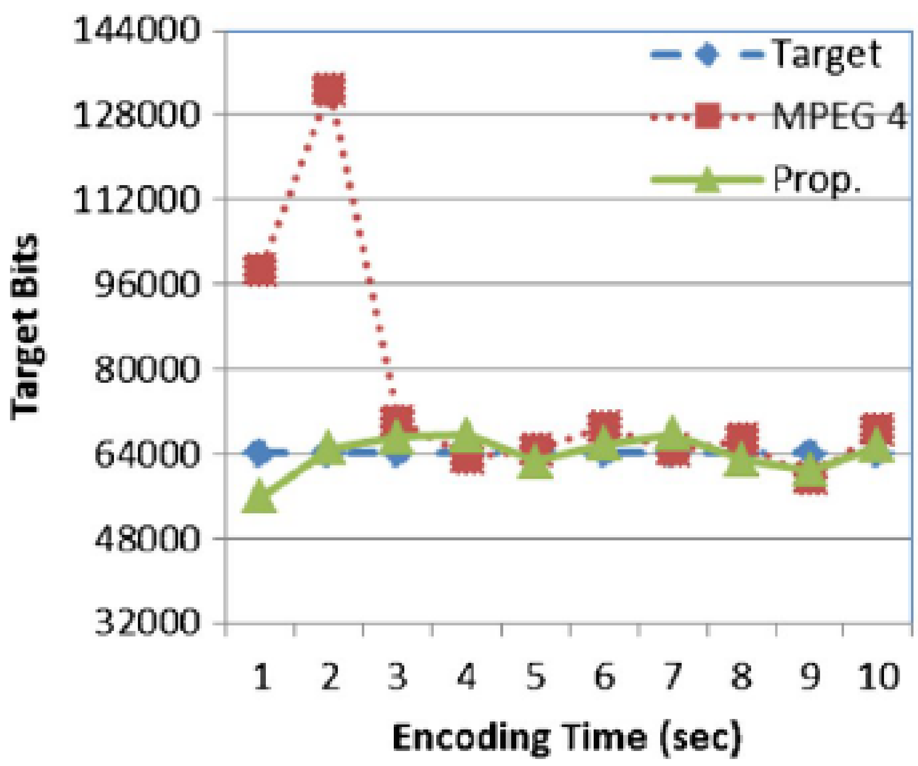
h Mobile PSNR (128 Kpbs)



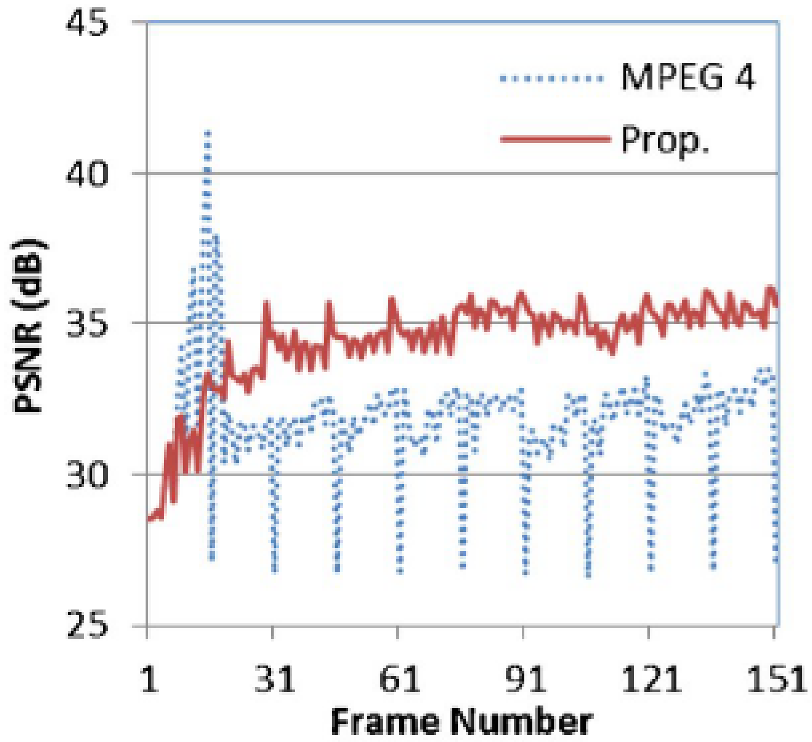
i Mother_Daughter Target & Actual Bitrate (32 Kbps)



j Mother_Daughter PSNR (32 Kbps)



k Hall Target & Actual Bitrate (64 Kbps)



l Hall PSNR (64 Kbps)

Fig. 3 (continued)

PSNRs when compared with the MPEG-4 solution. In some cases, MPEG-4 algorithm is out of control and cannot regulate the actual encoding bitrates to achieve the given target bitrates. For example, when the given target bitrate of *SILENT* sequence is 128 Kbps, the actual encoding bitrate achieved by MPEG-4 RC is *210.4* Kbps with an accuracy of *57.38 %*, while

Table 2 Overall performance comparison (Intra-Period=15, Encoding Structure IPPP…IPPP)

Sequences	Target bitrate (Kbps)	Actual encoding bitrate (Kbps)		Rate control accuracy (%)		PSNR (dB)		
		MPEG-4	Proposed	MPEG-4	Proposed	MPEG-4	Proposed	PSNR Gain
Akiyo	64	69.08	62.30	7.94 %	2.66 %	41.06	41.58	0.52
	48	55.93	48.25	16.52 %	0.53 %	39.69	39.61	−0.08
Coastguard	128	129.27	126.90	1 %	0.86 %	32.04	32.18	0.14
	96	98.15	95.22	2.23 %	0.81 %	30.88	31.02	0.14
News	96	106.41	93.60	10.84 %	2.50 %	36.21	36.60	0.39
	64	75.37	62.70	17.76 %	2.03 %	34.07	34.20	0.13
Mobile	256	260.54	254.91	1.77 %	0.43 %	28.45	28.57	0.12
	192	197.70	192.32	2.97 %	0.17 %	27.15	27.27	0.12
Stefan	256	260.23	253.73	1.65 %	0.89 %	31.53	31.66	0.13
	192	193.04	191.08	0.54 %	0.48 %	29.89	30.05	0.16
Mother_Daughter	64	72.73	62.53	13.65 %	2.3 %	37.26	37.46	0.20
	32	38.23	31.32	19.46 %	2.13 %	34.52	34.74	0.22
Train	128	132.87	127.63	3.80 %	0.29 %	31.79	32.36	0.57
	64	70.91	64.39	10.80 %	0.60 %	28.73	28.99	0.26
Hall	64	77.51	63.86	21.11 %	0.21 %	35.09	35.54	0.45
	32	41.58	32.37	29.92 %	1.17 %	31.73	31.81	0.08

our actual encoding bitrate is *126.04* Kbps and the accuracy is *1.53 %*. This demonstrates that MPEG-4 is out of control and our algorithm works very well.

For the temporal prediction structure (IBBP…IBBP), Fig. 3 presents some frame-by-frame encoding results in terms of the bitrate and PSNR. In these figures, the curve indicated by "MPEG-4" represents the actual encoding bitrate obtained by MPEG-4 RC while the curve of "Prop." represents the actual encoding bitrate achieved by our proposed agloriithm. The horizontal line of "Target" represent the target encoding bitrate which is a constant. From Fig. 3: (a), (c),(e),(g),(i),(k), we can observe that the curves of actual encoding bitrates of MPEG-4 are more fluctuated while our curves are usually smoother and closer to the target bitrates. Namely, the acutal encoding bitrates of our algorithm are closer to the target bitrates when compared with those obtained by MPEG-4 RC algorithms. Sometimes, for MPEG-4 RC, the differences between the actual encoding bitrates and target bitrates are large. This indicates that MPEG-4 RC has less control ability than our RC algorithm. From Fig. 3: (b),(d),(f),(h),(j),(i), we can see that the PSNR curves of our algorithm are smoother and generally higher than those PSNR curves of MPEG-4 RC. These demonstrate that our algorithm can achieve better encoding quality with lower quality fluctuation among frames. All of these indicate that the overall performance of our RC algorithm is better than that of MPEG-4 RC.

For the prediction structure (2) where only I-frames and P-frames are used, Table 2 shows the overall RC performance and Fig. 4 shows frame-by-frame PSNRs, target bitrates and actual encode bitrates in details. From all the experimental results showed in Table 2 and Fig. 4, we can draw the same conclusion that our proposed algorithm is very effective and can achieve more accurate target bitrates, higher RC accuracies with higher average PSNRs than MPEG-4 RC solution.

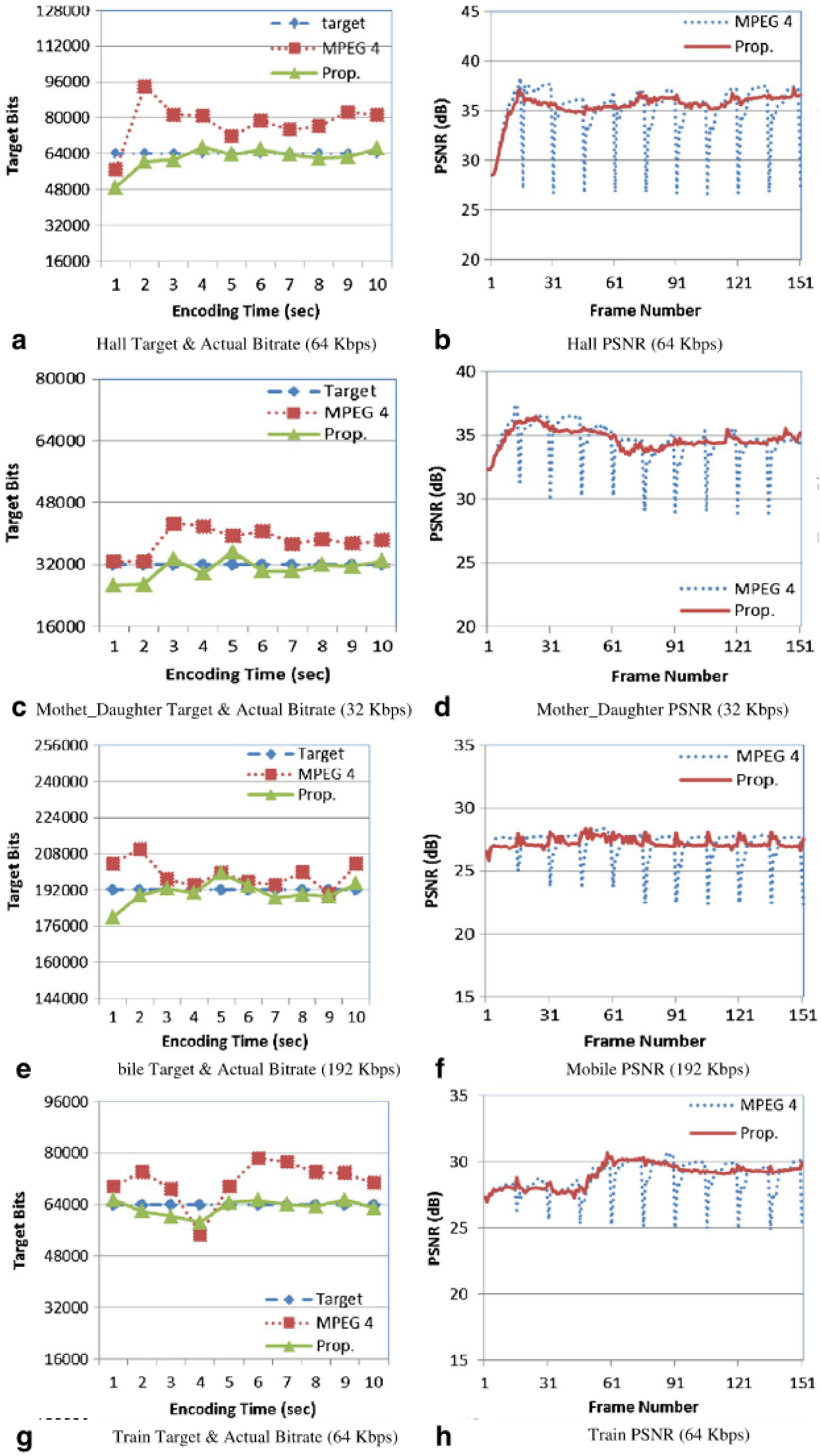


Fig. 4 Experimental results for video seuqences encoded at various bitrates (IPPP…IPPP)

5 Conclusion

In this paper, we have developed a new un-buffered real-time rate control algorithm for video coding. In addition to the unique framework which does not use a buffer in rate control, the proposed algorithm introduces several efficient rate control methods, including applying a PID controller in direct rate regulation and real-time target bit allocation. Our numerous experimental results have demonstrated that, when compared with MPEG-4 rate control algorithm, our proposed algorithm achieves more accurate rate regulation, obtains higher coding quality, decreases delay and thus improves real-time response.

Regarding future work directions, aiming at further enhancing the overall performance of video rate controller, we will extend this research to H.264/AVC and the new compression standard H.265. In addition, we will also continue our research on investigating impact of our non-buffer algorithm, developing more advanced rate control structure, and exploring better real-time target bit allocation method which considers the frame complexity, etc.

Acknowledgement This work is partially supported by NASA EPSCoR 2012 Award (No. NNX13AD32A) and faculty sabbatical leave fund from Uni. of Central Arkansas.

References

1. Chen Z, Ngan KN (2007) Recent advances in rate control for video coding. Signal Process Image Commun 22(1):19–38
2. Corbera JR, Lei S (1999) Rate control in DCT video coding for Low-delay communications. IEEE Trans Cir Syst Video Technol 9(1):172–185
3. Lee HJ, Chiang T, Zhang YQ (2000) Scalable rate control for MPEG-4 video. IEEE Trans Cir Syst Video Technol 10(6):878–894
4. Leontaris A, Tourapis A-M (2007) Rate Control Reorganization in the Joint Model (JM) Reference Software. JVT-W042, 23rd meeting, San Jose, California, USA
5. Leontaris A, Tourapis A-M (2007) Rate Control for the Joint Scalable Video Model (JSVM). JVT-W043, 24th meeting, San Jose, USA
6. Li ZG, Pan F, Lim KP, Feng GN (2003) Adaptive basic unit layer rate control for JVT. JVT-G012, 7th meeting, Pattaya, Thailand
7. Liu Y, Li ZG, Soh YC (2007) A novel rate control scheme for low delay video communication of H.264/AVC standard. IEEE Trans Circ Syst Video Technol 17(1):68–78
8. Ma Z, Xu M, Ou Y-F, Wang Y (2012) Modeling of rate and perceptual quality of compressed video as functions of frame rate and quantization stepsize and its applications. IEEE Trans Circ Syst Video Technol 22(5):671–682
9. MPEG-2 Test Model 5 (1993) Doc. ISO/IEC JTCI/SC29/WG11/93-400
10. MPEG-4 Video Verification Model V8.0 (1999) ISO/IEC JTC1/SC29/WG11 N3093 (1999) Coding of Moving Pictures and Audio, Maui
11. Ou Y-F, Ma Z, Wang Y (2011) Perceptual quality assessment of video considering both frame rate and quantization artifacts. IEEE Trans Circ Syst Video Technol 21(3):286–298
12. Ruan R, Hu R and Li Z (2011) A novel rate control algorithm of video coding based on visual perceptual characteristic. Proceedings of the 6th International Conference on Computer Science & Education (ICCSE)
13. Shen L, Zhang Z, Zhang W (2006) A New scheme for SNR-FGS with spatial scalability based on H.264/AVC. J Optoelectron Laser 17(8):948–952
14. Sun Y, Ahmad I (2004) A robust and adaptive rate control algorithm for objects based video. IEEE Trans Circ Syst Video Technol 14(10):1167–1182

15. Sun Y, Ahmad I (2005) Asynchronous rate control for multi-object videos. IEEE Trans Cir Syst Video Technol 15(8):1007–1018
16. Sun Y, Ahmad I, Li D, Zhang YQ (2006) Region-based rate control and bit allocation for wireless video transmission. IEEE Trans Multimed 8(1):1–10
17. Sun Y, Zhou Y, Feng Z, He Z (2009) Incremental rate control for H.264/AVC video compression. Inst Eng Technol (IET) Image Process 3(5):286–298
18. Tan E, Chou C-T (2012) A frame rate optimization framework for improving continuity in video streaming. IEEE Trans Multimed 14(3):910–922
19. Tao B, Dickinson WB, Peterson HA (2000) Adaptive model-driven bit allocation for MPEG video coding. IEEE Trans Circ Syst Video Technol 10(1):147–157
20. Tian L, Zhou Y, Sun Y (2012) Novel rate control scheme for intra frame video coding with exponential rate-distortion model on H.264/AVC. Elsevier J Vis Commun Image Represent 23(6): 873–882
21. Vetro A, Sun H, Wang Y (1999) MPEG-4 rate control for multiple video objects. IEEE Trans Circ Syst Video Technol 9(1):186–199
22. Wang H, Kwong S (2008) Rate-distortion optimization of rate control for H.264 with adaptive initial quantization parameter determination. IEEE Trans Circ Syst Video Technol 18(1):140–144
23. Yang J, Sun Y, Wu Y, Sun S (2012) Joint H.264/SVC-MIMO rate control for wireless video applications. Inst Eng Technol (IET) Image Process 6(1):43–52
24. Yang J, Sun Y, Zhou Y, Sun S (2013) Incremental rate control for H.264 AVC scalable extension. Int J Multimed Tools Appl (Springer) 64(3):581–598
25. Zhou Y, Sun Y, Feng Z, Sun S (2009) New rate-distortion modeling and efficient rate control for H.264/AVC video coding. Elsevier J Signal Process - Image Commun 24(5):345–356
26. Zhou Y, Sun Y, Feng Z, Sun S (2011) PID-based bit allocation strategy for H.264/AVC rate control. IEEE Trans Circ Syst II 58(3):184–188

Yu Sun 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. In August 2004, she joined in the Department of Computer Science, University of Central Arkansas, Conway, AR, USA. Now, she is an Associate Professor. Her main research interests include video compression, multimedia communication, and image processing. She is a voting member of IEEE Multimedia Communications Technical Committee (MMTC), was a co-editor of *IEEE Distributed Systems Online* and serves on technical program committees over 30 international conferences.

Zhidan Feng received the B.S. and M.S. degrees in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 1995. From 1997 to 2005, he was a research associate at the Department of Computer Science, Hong Kong University. He was also a software developer at Acxiom Corporation from 2006 to 2009. Since 2009, he joined IT research group, University of Arkansas for Medical Sciences (UAMS). Now, he is a senior software architect there.

Reshma R. Ginnavaram received her B.S. degree in Computer Science from the Jawaharlal Nehru Technological University of India in 2009. She received her M.S. degree in Applied Computing from the Department of Computer Science, University of Central Arkansas, Conway, Arkansas in 2011. Now, she is working for Textura Corporation.