Watching Vedios with Certain and Constant Quality: PID-based Quality Control Method

Yuhang Song[1], Mai Xu[1 2] *, Shengxi Li[1], and Zulin Wang[1 2]

[1]Beihang University
37 Xueyuan Road
Beijing, 100191, China
MaiXu@buaa.edu.cn

[2]Collaborative Innovation Center of Geospatial Technology
129 Luoyu Road
Wuhan, 430079, China
wzulin@buaa.edu.cn

Abstract

In this paper, we focus on obtaining certain and constant video quality, for improving quality of experience(QoE). To achieve this objective, we propose a novel PID-based quality control (PQC) method for video coding. Specifically, a formulation is modeled for control quality of video coding with two objectives, control error and quality fluctuation. Then, we apply Laplace domain analysis on modeling the relationship between quantization parameters (QPs) and control error in this formulation. Given the relationship between QP and control error, we then propose our PQC method to solve this formulation, such that videos can be compressed with certain and constant quality. Finaly, experimental results show the effectiveness of our PQC method in both control accuracy and quality fluctuation.

1 Introduction

Along with the explosive increase of multimedia content, effective and efficient compression algorithms have been always in demand for decades. Catering for such demand, various video coding standards have been proposed and developed to compress the raw video data [1–3], in the way of minimizing bit-rates or optimizing quality with certain constraints according to different applications. Generally speaking, rate control [4, 5] aims at controlling bit-rates to meet different requirements, e.g., minimizing distortion for storage application or reducing bit-rates fluctuation for communication usage. Moreover, in some cases when perceived quality is highly crucial, quality control can be adopted to compress a video at a certain and constant quality, thus obtaining a more desirable quality of experience (QoE).

Similar to constant bit-rates control which aims at smoothing bit-rates for each frame to avoid buffer overflow or underflow, constant quality control (see Figure 1) in video coding provides smooth quality in compressed frames to avoid overall perceived quality degradation caused by some intense quality fluctuation. Toward this end, a direct way is to adopt a two-pass procedure, which pre-analyzes all the frames in advance and calculates the global quantization parameters (QPs) for each frame [6] [7] [8]. In addition, in some cases the calculated QPs are not integers, and [9] thus proposed a solution by adjusting different QPs at block level to reach the desired QP value at frame level. Obviously, the pre-analysis strategy is not applicable to real-time streaming applications, such as video conferencing and live video streaming.

Later, He *et.al.* [10] proposed a low-pass filter with geometric factor, and then applied it in quality control to achieve smoothed frame quality, meanwhile satisfying buffer status

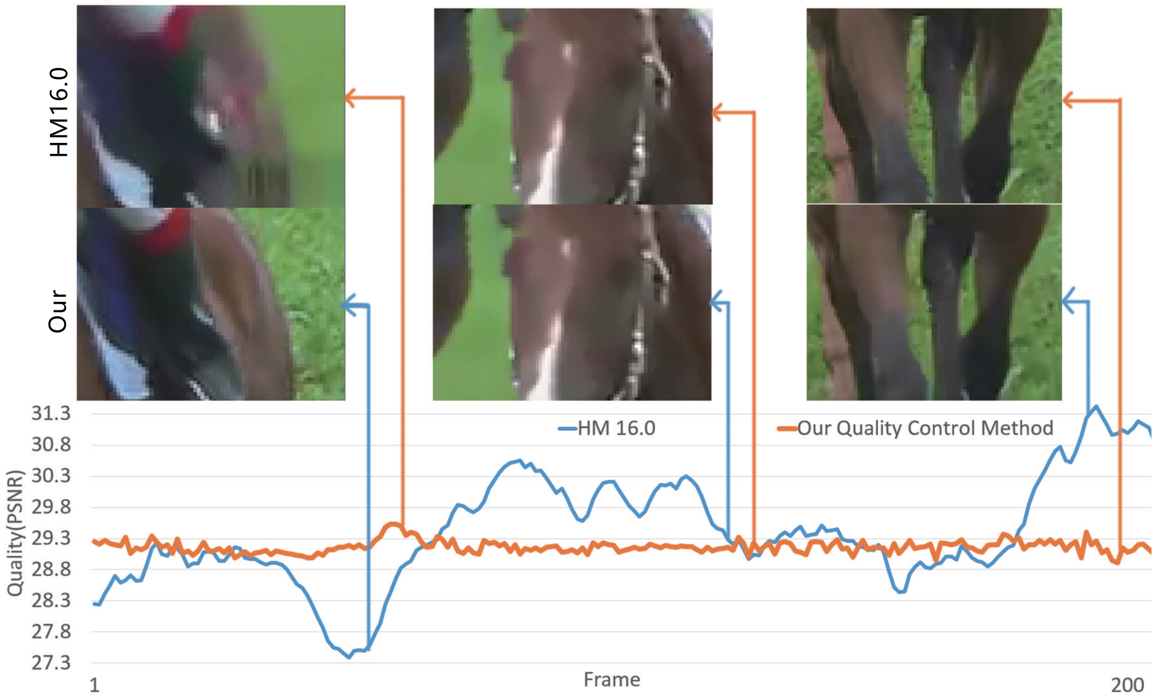

Figure 1: An example of quality control in video coding. In this figure, the state-of-the-art video coding standard high efficiency video coding (HEVC), which is implemented by the latest reference software HM 16.0, is used as an anchor. This figure shows that the fluctuation of the HEVC encoded video may result in poor perceived quality for some frames (e.g, frames 55, 120 and 190). By contrast, our quality control method is able to yield smooth quality, better than the poor quality of frames 55, 120 and 190 in HEVC. Besides, our method makes the distortion of encoded video method to the target, i.e., 29.2 dB.

with low delay. Besides, a PSNR adjustment was proposed in [11] to maintain the group of picture (GOP) level quality constant by empirically moving up/down QPs, according to previous frames. Instead of the GOP-based control, [12] proposed a sequence-based method, which tracks scene changes and then achieves smooth video quality adaptive to scene changes. Recently, a trellis-based method has been proposed in [13] to consider both bit budge and PSNR variance. Although the above works can reduce quality fluctuation with constant quality, they are unable to control quality to a certain level that users demand.

Most recently, Hu *et.al.* [14] have proposed to control the quality of encoded videos to a certain target for H.264. In their method, QP of each frame is adjusted by modeling consecutive frames, in the assumption of Laplacian distribution on transform coefficients. Based on the same assumption, [15] proposed a mixture model on Laplacian function to tailor distortion-quantization (D-Q) and rate-quantization (R-Q) relationships in HEVC, thus controlling quality in a constant manner. However, the certain assumptions, like Laplacian distribution of transform coefficients, are not precise for modeling various-content videos, thus leading to high inaccuracy in quality control. More importantly, those works are restricted to some specific standard or certain assumptions, and they are not effective when being applied to the latest r-λ model [16] of HEVC.

In this paper, we propose a novel quality control method for video coding, which does not rely on some specific encoder or assumption. Therefore, our method is suitable for different encoders with high control accuracy. To our best knowledge, our method is the first work on encoder-free quality control, which enables certain and constant distortion across all frames of encoded videos. Specifically, our method is inspired by the basic idea of the proportion-integral-derivative (PID) controller, thus called the PID-based quality control (PQC) method. We first propose a formulation of quality control in video coding for our PQC method, with objectives of minimization on both control error and fluctuation of distortion. Then, the relationship between QP and distortion is modeled for the formulation of our PQC method, such that the PID controller can be used to solve our formulation. Next, a PID-based solution is provided for our quality control formulation with modeled relationship between QP and distortion. As such, the distortion of video coding can be controlled to a target distortion with smooth quality fluctuation. Finally, we implement our PQC method in the latest HEVC encoder (HM 16.0), and our experiment results show that our method achieves the state-of-the-art quality control performance in terms of both control error and quality fluctuation.

2 Overview of PID Controller

The PID controller [17] is widely used to minimize the error between a desired step point and the measured process variable. Using the terms of proportional operator (P), integral operator (I) and derivative operator (D), a PID controller maintains a well trade-off among response speed, static error correction and overreacting repression. Generally speaking, the PID controller performs robustly with little overhead of computational complexity. Thus, our quality control method for video coding is also based on the PID controller.

To be more specific, assume that there exists error between the current and target positions until time $(t-1)$, which are denoted by $\{e_{t-1}, e_{t-2}, \cdots, e_0\}$. The PID controller focuses on minimizing error e_t by adjusting a control variable o_t. In the PID controller, o_t can be calculated by

$$o_t = K_p e_{t-1} + K_i \int_0^{t-1} e_\tau d\tau - K_d \frac{de_{t-1}}{dt}, \tag{1}$$

where e_{t-1}, $\int_0^{t-1} e_\tau d\tau$ and $\frac{de_{t-1}}{dt}$ are the proportion (P), integral (I) and derivative (D) values; K_p, K_i and K_d (all ≥ 0) denote their corresponding weights. As can be seen from (1), P is decided by the most recent error, whilst I accounts for long lasting previous error and D predicts error in the future. For more detials about the setting of K_p, K_i and K_d, refer to [18]. To sum up, the PID controller outputs an optimal predicted value of the control variable to minimize error e_t, using all the error incurred until $(t-1)$. Next, we propose to control quality of video coding by incorporating the PID controller.

3 PID-based quality control method

In this section, we present our encoder-free PQC method for video coding, which is achieved by predicting the optimal QP before encoding each frame. Specifically, we first establish in Section 3.1 the formulatioin of quality control in video coding. In Section 3.2 we further model the relationship between QP_t and e_t for the proposed quality control formulation. In Section 3.3, we solve the proposed quality control formualtion using the PID controller.

3.1 Formulation of quality control

In video coding, there are two main objectives for quality control:

Objective I: Minimizing the error between the actual and target quality, averaged over all frames.

Objective II: Minimizing the fluctuation of quality along with frames.

The above two objectives can be achieved by predicting the optimal QP before encoding each frame. In other words, before encoding the t-th frame, we need to estimate the best QP value for this frame, which is denoted by QP_t. Assuming that T is the target distortion and D_t is the distortion of the t-th frame, the quality control can be formulated by

$$\mathrm{QP}_t = \underset{\mathrm{QP}}{\mathrm{argmin}} \{ \underbrace{\lambda \cdot \underbrace{(D_t(\mathrm{QP}) - T)}_{\textbf{Objective I}} + (1-\lambda) \cdot \underbrace{\frac{dD_t(\mathrm{QP})}{dt}}_{\textbf{Objective II}}}_{e_t} \}, \tag{2}$$

where $(D_t(\mathrm{QP}) - T)$ models the error between the actual quality and target quality (**Objective I**), whereas $\frac{dD_t(\mathrm{QP})}{d_t}$ models the fluctuation of quality (**Objective II**). In addition, λ represents the trade off between the two objectives, and e_t denotes the overall error to be minimized.

Since PSNR is a widely used distortion evaluation metric for video coding, it is applied to model distortion D_t in (2) for this paper. However, our PQC method can be simply extended to adopting other distortion evaluation metrics, like Structural Similarity Index (SSIM) and Video Signal-to-Noise Ratio (VSNR). To solve formulation (2), we first need to model the relationship between QP_t and e_t, to be discussed in the following.

3.2 Relationship between QP_t and e_t

In a coding system, the relationship between QP_t and e_t can be modeled by the following function Ψ,

$$\Psi(\mathbf{I}_t, \mathbf{I}_{t-1}, \cdots, \mathbf{I}_0, \mathrm{QP}_t, \mathrm{QP}_{t-1}, \cdots, \mathrm{QP}_0) = e_t, \tag{3}$$

where $\mathbf{I}_t$ stands for the frame content at frame t. This formulation shows that content and QPs of all frames until currently encoded frame contribute to quality control error e_t. In fact, $\mathbf{I}_t, \mathbf{I}_{t-1}, \cdots, \mathbf{I}_0$ is a set of images from the video sequence. We denote them by a single tensor $\mathbb{I}$. Our intention here is to analyze the relationship between QP_t and e_t. Thus, given a sequence, we have a fixed $\mathbb{I}$, and then (3) can be rewritten by

$$\Psi_{\mathbb{I}}(\mathrm{QP}_t, \mathrm{QP}_{t-1}, \cdots, \mathrm{QP}_0) = e_t. \tag{4}$$

Obviously, $\Psi_{\mathbb{I}}$ describes the relationship between QP_t and e_t for given $\mathbb{I}$. To obtain $\Psi_{\mathbb{I}}$, we propose a simple and pratical way from the viewpoint of signal processing by treating $\Psi_{\mathbb{I}}$ as an unknown linear time invariant (LTI) system (Input:QP_t and Output:e_t).

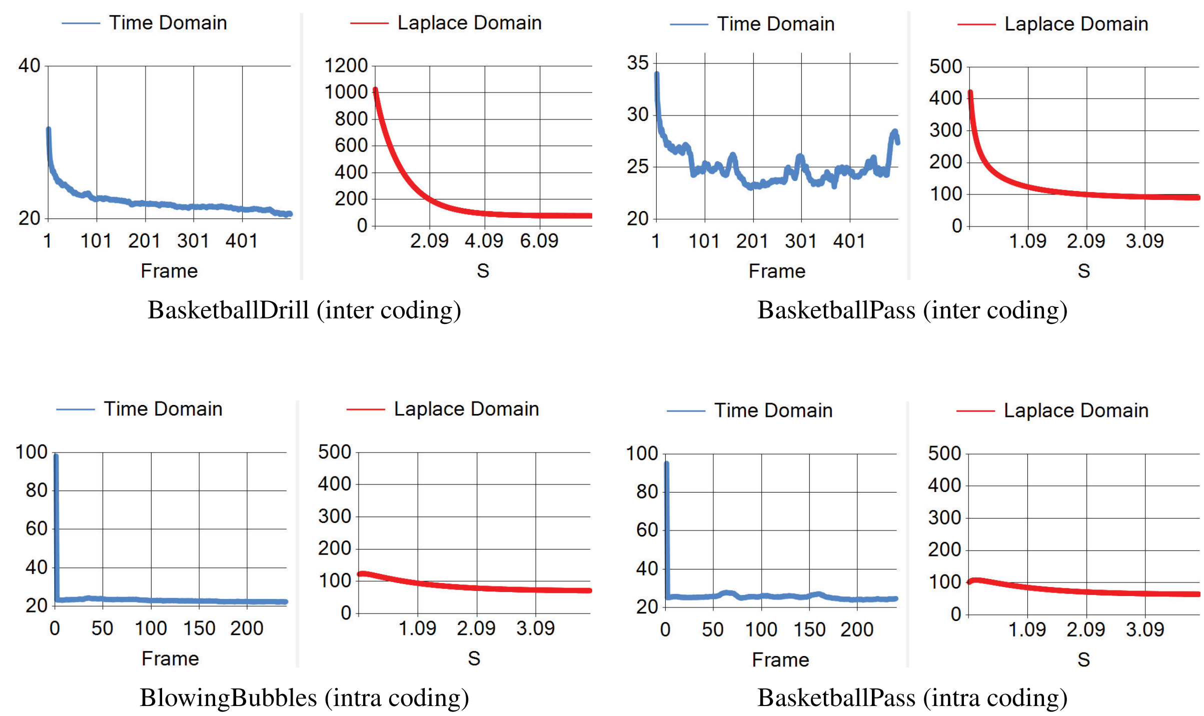

Figure 2: Analysis of pratical $\Psi_{\mathbb{I}}(\mathbf{QP})$ upon the state-of-the-art HEVC encoder with inter and intra coding. For analyzing $\Psi_{\mathbb{I}}(\mathbf{QP})$, we encoded 2 sequences by HM 16.0, using the configuration file *encoder_lowdelay_P_main.cfg* and *encoder_intra_main.cfg*. Here, the frames of each sequence were encoded with $\mathbf{QP} = \{\mathrm{QP}_0 = \mathrm{QP}_{\min}, \mathrm{QP}_1 = \mathrm{QP}_{\max}, \mathrm{QP}_2 = \mathrm{QP}_{\max}, \cdots, \mathrm{QP}_t = \mathrm{QP}_{\max}\}$, in which $\mathrm{QP}_{\min} = 0$ and $\mathrm{QP}_{\min} = 51$ in HEVC. Then, we show $\{e_1, e_2, \cdots, e_t\}$ for each encoded frame as the "Time Domain curve". We further process Laplace transform on "Time Domain curve" of $\Psi_{\mathbb{I}}(\mathbf{QP})$ to obtain "Laplace Domain curve". It is obvious that "Laplace Domain curve" has 1 and 0 pole for inter and intra coding, respectively.

First, the encoding system is input with an impulse signal $\mathbf{QP} = \{\mathrm{QP}_0 = \mathrm{QP}_{\min}, \mathrm{QP}_1 = \mathrm{QP}_{\max}, \mathrm{QP}_2 = \mathrm{QP}_{\max}, \cdots, \mathrm{QP}_t = \mathrm{QP}_{\max}\}$, where $\mathrm{QP}_{\min}$ and $\mathrm{QP}_{\max}$ are possibly minimal and maximal QP values of the encoder, respectively. Then, the system response $\Psi_{\mathbb{I}}(\mathbf{QP})$ of each video sequence is obtained for a specific video encoder (i.e.,HEVC). Next, we apply Laplace transform on the obtained $\Psi_{\mathbb{I}}(\mathbf{QP})$ to estimate the number of poles on its Laplace domain. The results are shown in Figure 2. From this figure, we can see that $\Psi_{\mathbb{I}}(\mathbf{QP})$ has 1 and 0 pole on Laplace domain for inter frame and intra frame encoding, respectively. As such, $\Psi_{\mathbb{I}}(\mathbf{QP})$ can be seen as first- and zero- order systems for inter and intra frame video coding [19] [20]. Therefore, $\Psi_{\mathbb{I}}(\mathbf{QP})$ can be represented by the differential equations, as follows,

$$\textbf{Inter Frame}: \ A_1^{\mathbb{I}} \cdot e_t + A_2^{\mathbb{I}} \cdot \frac{de_t}{dt} = \mathrm{QP}_t, \tag{5}$$

$$\textbf{Intra Frame}: \ A_0^{\mathbb{I}} \cdot e_t = \mathrm{QP}_t, \tag{6}$$

where $A_0^{\mathbb{I}}$, $A_1^{\mathbb{I}}$ and $A_2^{\mathbb{I}}$ are the coefficients derived from the data analysis in Figure 2. Note that the exact values for $A_0^{\mathbb{I}}$, $A_1^{\mathbb{I}}$ and $A_2^{\mathbb{I}}$ are not needed, since our PQC method only requires the number for orders of the differential equations, when applying the PID algorithm to solve our quality control formulation of (2).

In following section, we focus on our solution to the proposed quality control formulation (2) on the basis of e_t and QP_t relationship of (5) (6) and the PID controller of (1).

3.3 PID-based solution to the quality control formulation

In this section, we apply the PID controller to solve (2). As metioned in Section 2, the PID controller minimizes the error e_t alongside time t, via adjusting the control variable o_t. However, the PID controller can perform well, only when applied to a second-order system [21]. In other words, e_t and o_t in (1) need to satisfy the following differential equation:

$$M_2 \cdot \frac{d^2 e_t}{dt^2} + M_1 \cdot \frac{de_t}{dt} + M_0 \cdot e_t = o_t, \tag{7}$$

where M_2, M_1 and M_0 are constants. Next, we use the following way to make the modeled relationship between e_t and QP_t (i.e, (5) and (6)) satisfy the above requirement of the PID controller (i.e, (7)). As such, the PID controller can be applied to solve our quality control formulation of Section 3.1. Specifically, by applying the differential operation on (5), the following equation holds:

$$A_1^{\mathbb{I}} \cdot \frac{d^2 e_t}{dt^2} + A_0^{\mathbb{I}} \cdot \frac{de_t}{dt} + 0 \cdot e_t = \frac{d\mathrm{QP}_t}{dt}. \tag{8}$$

Thus, we can see that (8) meets the requirement (7) of the PID controller with $M_2 = A_1^{\mathbb{I}}$, $M_1 = A_0^{\mathbb{I}}$ and $M_0 = 0$. As a result, we have

$$\frac{d\mathrm{QP}_t}{dt} = o_t. \tag{9}$$

Then, (9) can be rewritten as follows,

$$\textbf{Inter Frame}: \ \mathrm{QP}_t = \int_0^t o_\tau d\tau. \tag{10}$$

Similarly, on the basis of (6) and (7), we have the following equation for the intra frames of video coding:

$$\textbf{Intra Frame}: \ \mathrm{QP}_t = \int_0^t \int_0^\tau o_\rho d\rho d\tau. \tag{11}$$

Finally, by replacing o_t with (1), the QP values of each frame can be estimated as follows,

$$\textbf{Inter Frame}: \ \mathrm{QP}_t = \int_0^t o_\tau d\tau = \int_0^{t-1} (K_p e_{\rho-1} + K_i \int_0^{\rho-1} e_\gamma d\gamma - K_d \frac{de_{\rho-1}}{d\rho}) d\rho, \tag{12}$$

$$\textbf{Intra Frame}: \ \mathrm{QP}_t = \int_0^t \int_0^\tau o_\rho d\rho d\tau = \int_0^{t-1} \int_0^\tau (K_p e_{\rho-1} + K_i \int_0^{\rho-1} e_\gamma d\gamma - K_d \frac{de_{\rho-1}}{d\rho}) d\rho d\tau, \tag{13}$$

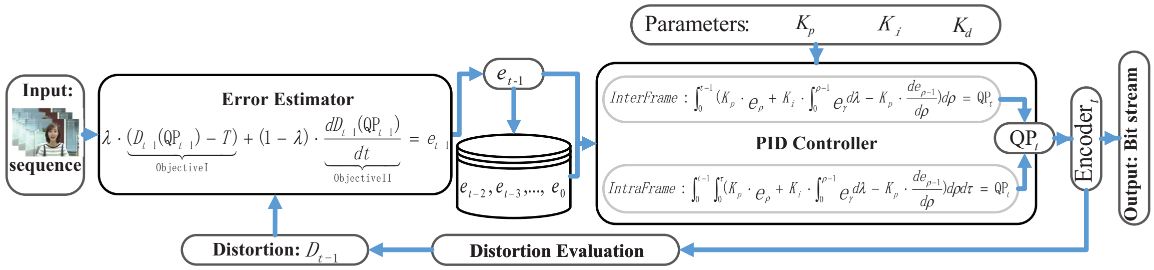

Figure 3: Framework of our PQC method.

for controlling quality of video coding. Obviously, we can see that QP_t is only related to control error $\{e_0, e_1, \cdots, e_{t-1}\}$ and parameters K_p, K_i and K_d, such that our method can be seen as an encoder-free quality control method. Figure 3 summarizes the overall framework of our PQC method.

4 Experimental results

In this section, experimental results are presented to vevify the effectiveness of our PQC method. Details about the parameter setting are presented in Section 4.1. In Section 4.2, we compare the quality control performance of our PQC and [15] in terms of the control error and quality fluctuation. In addition, we evaluate the rate-distortion performence of our PQC and [15] methods, whose results are provided in Section 4.3. Finally, to verify the simplicity and practicality of our PQC method, Section 4.4 briefly evaluate the computional complexity of our PQC method.

4.1 Parameter Setting

All 16 video sequences from the standard test sequence database [22] were utilized for comparison in our experiment. Moreover, we implemented our PQC method on the latest HM 16.0 platform to compress all test sequences. At the same time, the state-of-the-art quality control method [15] was utilized for comparison, which was also implemented on HM 16.0 for fair comparison. It is also worth pointing out that our PQC method can be applied to other encoders, as our PQC method only relies on QP and visual quality. For both methods, the low delay IPPP structure was chosen, using the configuration file *encoder_lowdelay_P_main.cfg*. Two QP values, i.e., $\mathrm{QP} = 32$ and $\mathrm{QP} = 37$, were chosen in our experiments, and other parameters were set by default. Specifically, the default HM 16.0 was used to compress all 16 sequences at $\mathrm{QP} = 32$ and $\mathrm{QP} = 37$. Then, the distortion for each compressed sequences was obtained in terms of Y-PSNR, and then it was set as the target quality for both our and [15] methods.

For the paramters related to our PQC method, we followed the PID paramter-tuning method in [18] to find proper values of K_p, K_i and K_d. As for HM 16.0 platform, we set

$K_p = 2.12$, $K_i = 0.10$, $K_d = 0.60$.[1] Next, we emperically set $\lambda = 0.8$ to balance error and fluctuation. Since λ represents the trade-off between error and fluctuation of quality control in video coding, it (is modifiable)/(varies) according to different applications.

4.2 Quality Control Performence

Control error and quality fluctuation are two crucial objectives emphased in our PQC method. Thus, we compare the quality control performence between our PQC and the conventional [15] methods, in terms of total control error and quality fluctuation alongside frames. As can be seen from Tabel 1, the 3-5th and 8-10th columns report the control error of our and [15] methods. which is the difference between actual and target Y-PSNR. It is obvious that that our method significant reduces the control error when compared with [15], in a magnitude of 10^2.

Figure 4 illustrates the quality fluctuation of our PQC, [15] and default HM 16.0 methods, with the curve of Y-PSNR alongside frames. As can been seen from this figure, our PQC method produces the most smooth Y-PSNR curve. Similar results can also be found for other sequences, as can be verified from the 6th and 11th columns in Table 1, in which the quality fluctuation is measured by standard deviation of Y-PSNR for each video sequence. In average, our PQC method reduces the quality fluctuation for around 75% compared with [15]. Therefore our PQC method achieves the best performence in both control error and quality fluctuation.

4.3 Rate-Distortion Performence

In this section, we focus on the performance of three methods in terms of rate-distortion (R-D) performence. As reported in the 7th and 12th columns in Table 1, at similar quality, our method consumes slightly more bit-rates (around 5% bit-rate increase) than the conventional HM 16.0 for most sequences, which is the cost of smoothing on quality fluctuation. However, the fluctuation is significantly reduced in our method. By contrast, the bit-rates of [15] are much higher than that of our PQC method. The reason is that [15] generally yields better quality than the target quality with more consumed bit-rates. Therefore, compared with [15], our PQC method not only has a better control accuracy but also consumes a far less total bit rate.

4.4 Evaluation on Computational Complexity

The complexity of our PQC and [15] methods is evaluated by the consumed time, during which the tested method complete all the operations and is ready for encoding the next frame. Both method is tested on the same machine[2]. The results show that our PQC method take XX ms in average to complete all the operations, yet the [15] method take XX ms under the same conditions. The reason can be inferred from the final equations of our PQC

[1]It is worth pointing out that the values of K_p, K_i and K_d are not unique, and they are acceptable in a wide range due to the robustness of PID controller.

[2]CPU: Intel i7 4790k, 4.36GHz, 4cores, 8threads; Memory: DDR4, 3300GHz

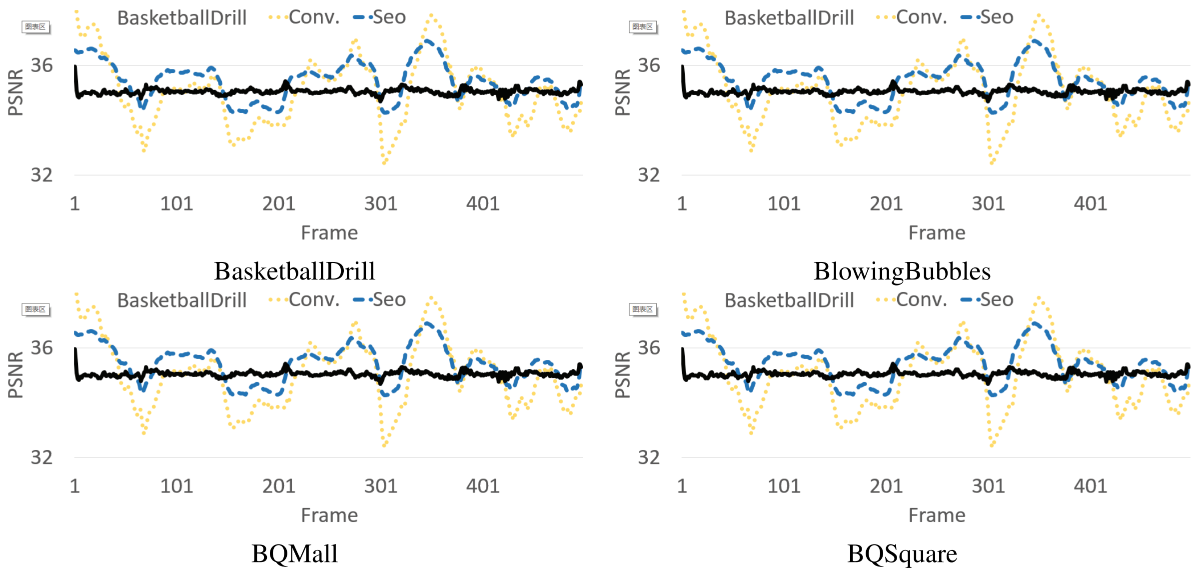

Figure 4: Quality fluctuation of our PQC, conventional [15] and the default HM 16.0 methods. All use the configuration file *encoder_lowdelay_P_main.cfg*. The quality is measured by Y-PSNR. Similar fluctuation results can be found for other encoded video sequences, and quantified fluctuation is provided in Table 1.

method (12) and (13). Both equations can be achieved within 10 times of floating-point addition operations and 10 times of floating-point multiplication operations. In other words, our PQC method rarely increases the computional complexity, as is also verified by above experiment.

5 Conclusion

In this paper, we have proposed a new method, called PQC method, to optimally control the quality of video coding. First, we established a formulation for optimal quality control, which considers both error and fluctuation of quality control for video coding. For the established formulation, the relationship between QP and control error was modeled in our PQC method. Then, a PID-based solution was developed to solve our quality control formulation, based on the modeled relationship between QP and control error alongside encoded frames. Since our PQC method only alternates QP values of encoded frames for achieving desirable control error and quality fluctuation, it can be seen as an encoder-free method. We further implemented our PQC method in the latest HEVC encoder, i.e., HM 16.0. The experimental results show that the control error and quality fluctuation of our PQC method are better than those of the conventional method for HEVC.

References

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

Table 1: Comparison of quality control performence for HM 16.0, [15] and our method.

Video	Method	QP = 32					QP = 37				
		Avg. PSNR (dB))	Control Error (dB)	Control Error (%)	Quality Fluctuation (dB)	Bit Rate (Mbps)	Avg. PSNR (dB)	Control Error (dB)	Control Error (%)	Quality Fluctuation (dB)	Bit Rate (Mbps)
$BasketballDrill$	HM16	33.93	–	–	0.73	8.5	31.59	–	–	0.72	4.4
	[15]	34.77	0.8424	2.48	0.20	11.9	32.68	1.0923	3.46	0.35	6.9
	Our	33.92	**0.0046**	**0.01**	**0.13**	9.3	31.60	**0.0030**	**0.01**	**0.11**	4.8
$BasketballDrive$	HM16	35.05	–	–	1.25	32.7	32.73	–	–	1.31	16.6
	[15]	36.43	1.3818	3.94	0.71	51.3	34.27	1.5373	4.70	0.83	25.5
	Our	35.05	**0.0054**	**0.02**	**0.18**	34.5	32.73	**0.0053**	**0.02**	**0.20**	17.3
$BasketballPass$	HM16	33.57	–	–	2.29	4.2	30.54	–	–	2.18	2.1
	[15]	35.94	2.3761	7.08	0.93	7.7	32.11	1.5757	5.16	1.05	3.4
	Our	33.56	**0.0052**	**0.02**	**0.18**	4.7	30.54	**0.0084**	**0.03**	**0.17**	2.4
$BlowingBubbles$	HM16	30.12	–	–	1.36	3.6	27.27	–	–	1.31	1.5
	[15]	31.91	1.7880	5.94	0.45	5.9	28.44	1.1680	4.28	0.60	2.4
	Our	30.11	**0.0038**	**0.01**	**0.16**	3.9	27.27	**0.0024**	**0.01**	**0.17**	1.7
$BQMall$	HM16	33.87	–	–	2.26	9.4	31.02	–	–	2.40	4.7
	[15]	35.35	1.4711	4.34	0.82	14.0	32.34	1.3220	4.26	0.93	6.8
	Our	33.87	**0.0033**	**0.01**	**0.07**	10.6	31.02	**0.0007**	**0.00**	**0.06**	5.4
$BQSquare$	HM16	30.02	–	–	0.86	3.4	27.15	–	–	1.09	1.3
	[15]	32.36	2.3419	7.80	0.27	6.9	28.87	1.7275	6.36	0.34	2.3
	Our	30.01	**0.0037**	**0.01**	**0.16**	3.4	27.15	**0.0041**	**0.02**	**0.13**	1.4
$BQTerrace$	HM16	32.92	–	–	0.94	23.6	30.65	–	–	1.16	8.4
	[15]	34.01	1.0927	3.32	0.66	44.6	31.93	1.2877	4.20	0.80	15.3
	Our	32.92	**0.0017**	**0.01**	**0.06**	26.9	30.65	**0.0010**	**0.00**	**0.05**	9.3
$Cactus$	HM16	34.46	–	–	0.38	26.5	32.27	–	–	0.46	13.0
	[15]	35.18	0.7174	2.08	0.18	39.9	32.73	0.4609	1.43	0.22	18.7
	Our	34.46	**0.0032**	**0.01**	**0.06**	30.0	32.27	**0.0025**	**0.01**	**0.06**	12.8
$FourPeople$	HM16	37.12	–	–	0.71	4.4	34.53	–	–	0.78	2.3
	[15]	38.36	1.2390	3.34	0.15	6.9	34.46	0.0695	0.20	0.14	2.7
	Our	37.12	**0.0021**	**0.01**	**0.07**	4.6	34.53	**0.0009**	**0.00**	**0.04**	2.8
$Johnny$	HM16	38.34	–	–	0.82	1.9	35.94	–	–	1.35	1.0
	[15]	39.11	0.7768	2.03	0.32	2.6	36.62	0.6898	1.92	0.19	1.3
	Our	38.34	**0.0022**	**0.01**	**0.08**	1.9	35.94	**0.0016**	**0.00**	**0.08**	1.0
$Kimono1$	HM16	36.58	–	–	1.20	12.0	33.95	–	–	1.60	5.9
	[15]	39.07	2.4849	6.79	0.22	12.3	36.68	2.7344	8.05	0.26	6.2
	Our	36.61	**0.0303**	**0.08**	**0.15**	12.2	34.03	**0.0797**	**0.23**	**0.14**	6.0
$KristenAndSara$	HM16	38.42	–	–	0.62	3.3	35.86	–	–	1.01	1.7
	[15]	39.25	0.8277	2.15	0.22	5.1	36.13	0.2638	0.74	0.16	2.2
	Our	38.43	**0.0019**	**0.00**	**0.06**	3.6	35.86	**0.0013**	**0.00**	**0.05**	2.0
$ParkScene$	HM16	33.72	–	–	0.60	13.6	31.19	–	–	0.65	5.9
	[15]	35.01	1.2838	3.81	0.18	20.7	32.23	1.0413	3.34	0.17	8.5
	Our	33.72	**0.0086**	**0.03**	**0.16**	13.7	31.19	**0.0058**	**0.02**	**0.15**	6.0
$PartyScene$	HM16	30.06	–	–	1.44	15.6	27.18	–	–	1.38	6.6
	[15]	30.71	0.6561	2.18	0.37	18.9	27.29	0.1116	0.41	0.37	7.7
	Our	30.06	**0.0002**	**0.00**	**0.23**	16.7	27.18	**0.0042**	**0.02**	**0.10**	7.7
$RaceHorses$	HM16	32.03	–	–	1.37	3.0	29.16	–	–	1.23	1.4
	[15]	33.66	1.6308	5.09	0.73	4.4	29.85	0.6888	2.36	0.61	1.8
	Our	32.02	**0.0074**	**0.02**	**0.23**	3.1	29.17	**0.0093**	**0.03**	**0.11**	1.5
$RaceHorses$	HM16	32.92	–	–	2.62	10.2	29.89	–	–	2.33	4.6
	[15]	34.14	1.2177	3.70	1.13	14.9	30.40	0.5093	1.70	0.94	5.8
	Our	32.92	**0.0085**	**0.03**	**0.73**	13.4	29.90	**0.0133**	**0.04**	**0.16**	5.8
Average	HM16	33.94	–	–	1.22	11.0	31.31	–	–	1.31	5.1
	[15]	35.33	1.3830	4.13	0.47	16.7	32.31	1.0175	3.29	0.50	7.3
	Our	33.94	**0.0057**	**0.02**	**0.18**	11.7	31.32	**0.0090**	**0.03**	**0.13**	5.2

[2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[3] D. Mukherjee, J. Bankoski, A. Grange, J. Han, J. Koleszar, P. Wilkins, Y. Xu, and R. Bultje, "The latest open-source video codec vp9-an overview and preliminary results," in *Picture Coding Symposium (PCS), 2013*. IEEE, 2013, pp. 390–393.

[4] B. Li, H. Li, L. Li, and J. Zhang, "Domain rate control algorithm for high efficiency video coding," *IEEE transactions on Image Processing*, vol. 23, no. 9, pp. 3841–3854, 2014.

[5] L. Shengxi, X. Mai, R. Yun, M. Chengzhang, and Z. Wang, "Optimizing subjective quality in hevc-msp: An approximate closed-form image compression approach," in *Data Compression Conference (DCC), 2013*. IEEE.

[6] Y. Yu, J. Zhou, Y. Wang, and C. W. Chen, "A novel two-pass vbr coding algorithm for fixed-size storage application," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 345–356, 2001.

[7] L.-J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 446–459, 1998.

[8] J. Lee and B. W. Dickinson, "Temporally adaptive motion interpolation exploiting temporal masking in visual perception," *IEEE Transactions on image processing*, vol. 3, no. 5, pp. 513–526, 1994.

[9] Z. He, Y. K. Kim, and S. K. Mitra, "Low-delay rate control for dct video coding via ρ-domain source modeling," *IEEE transactions on Circuits and Systems for Video Technology*, vol. 11, no. 8, pp. 928–940, 2001.

[10] Z. He, W. Zeng, and C. W. Chen, "Low-pass filtering of rate-distortion functions for quality smoothing in real-time video communication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 8, pp. 973–981, 2005.

[11] F. De Vito and J. C. De Martin, "Psnr control for gop-level constant quality in h. 264 video coding," in *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.* IEEE, 2005, pp. 612–617.

[12] B. Xie and W. Zeng, "A sequence-based rate control framework for consistent quality real-time video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 56–71, 2006.

[13] K.-L. Huang and H.-M. Hang, "Consistent picture quality control strategy for dependent video coding," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 1004–1014, 2009.

[14] S. Hu, H. Wang, and S. Kwong, "Adaptive quantization-parameter clip scheme for smooth quality in h. 264/avc," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1911–1919, 2012.

[15] C.-W. Seo, J.-H. Moon, and J.-K. Han, "Rate control for consistent objective quality in high efficiency video coding," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2442–2454, 2013.

[16] S. Li, M. Xu, Z. Wang, and X. Sun, "Optimal bit allocation for CTU level rate control in HEVC," *Transactions on Circuits and Systems for Video Technology*, 2016.

[17] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE transactions on control systems technology*, vol. 13, no. 4, pp. 559–576, 2005.

[18] P. Cominos and N. Munro, "PID controllers: recent tuning methods and design to specification," *IEE Proceedings-Control Theory and Applications*, vol. 149, no. 1, pp. 46–53, 2002.

[19] D. V. Widder, *Laplace Transform (PMS-6).* Princeton University Press, 2015.

[20] R. C. Dorf and R. H. Bishop, "Modern control systems," 1998.

[21] C. Zhao, D. Xue, and Y. Chen, "A fractional order PID tuning algorithm for a class of fractional order plants," in *IEEE International Conference Mechatronics and Automation, 2005*, vol. 1. IEEE, 2005, pp. 216–221.

[22] F. Bossen, "Common test conditions and software reference configurations," *Document: JCTVC-G1200, Joint Collaborative Team on Video Coding*, Nov. 2011.