Template for DCC Manuscripts

Author One*, Author Two†, and Author Three*

*Institution One
Street Address One
City, State, ZIP, Country
email@address

†Institution Two
Street Address Two
City, State, ZIP, Country
email@address

Abstract

The abstract describes concisely and clearly the main contributions of the paper. It should not contain math equations or citations, to ensure the abstract is self-contained and readable if converted to ASCII text. The abstract should contain about 100 to 150 words.

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] aims at controlling bit-rates to meet different requirements, e.g., minimizing distortion for storage application or reducing bit-rate fluctuation for communication usage. Moreover, in some cases when the perceived quality is highly crucial, quality control is adopted to compress a video at a certain and constant quality, thus obtaining a more desirable quality of experience (QoE).

Similar to constant bit-rate control which aims at smoothing bit-rates for each frame to avoid buffer overflow or underflow, constant quality control (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 [5] [6] [7]. In addition, in some cases the calculated QPs are not integers, and [8] 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.* [9] proposed a low-pass filter with geometric factor, and then applied it in quality control to achieve smoothed frame quality, meanwhile satisfying buffer status with low delay. Besides, a PSNR adjustment was proposed in [10] 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, [11] proposed a sequence-based method, which tracks scene changes and then achieves smoother video quality adaptive to scene changes. Recently, a trellis-based method has been proposed in [12] to consider both bit budge and PSNR variance. However, although the above works can reduce quality

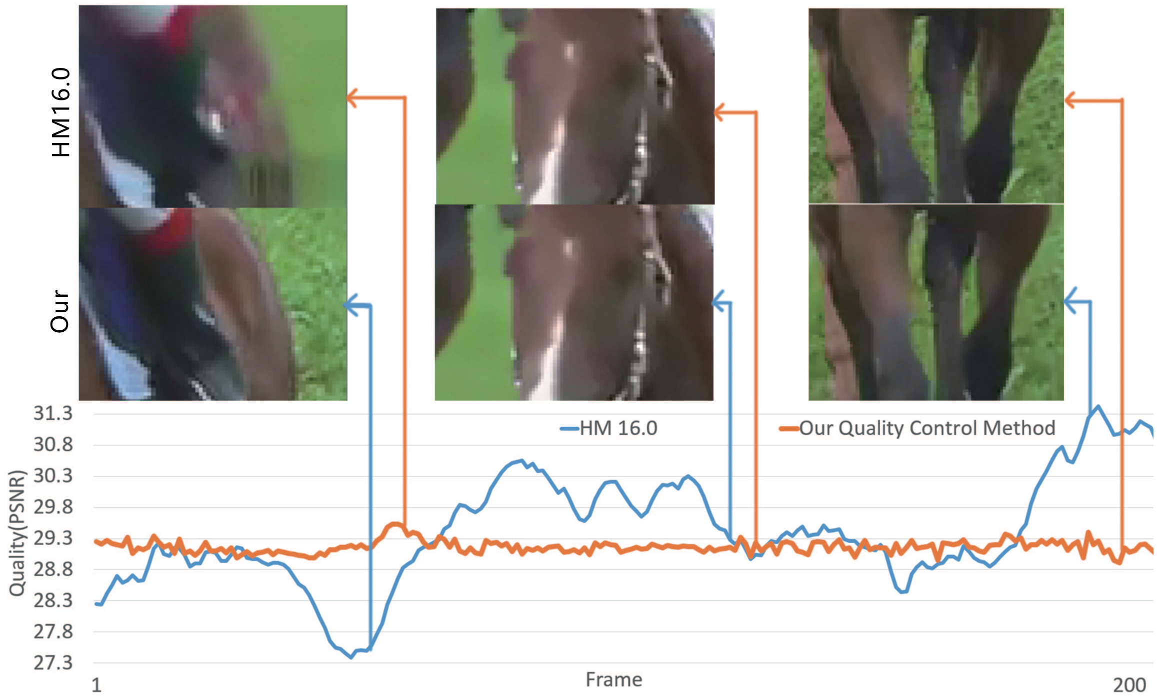


Figure 1: An example of quality control in video coding. In this figure, the state-of-the-art video coding standard high efficiency video coding (HEVC), which is implemented by the latest reference software HM16.0, is used as an anchor. This figure shows that the fluctuation of the HEVC encoded video 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 is able to make the distortion of the encoded video approach to the target, i.e., 29.2 dB.

fluctuation with constant quality, they are unable to control quality to a certain level that users demand.

Most recently, Hu *et.al.* [13] 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 modelling consecutive frames, in the assumption of Laplacian distribution on transform coefficients. Also based on the same assumption, [14] 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, those works are restricted to either different standards or certain assumptions, and therefore they are not effective when being applied to the latest r-λ model [4] 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. 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 out PQC method, such that the PID controller can be used to solve our formulation. Next, a PID-based solution is provided to our quality control formulation with modelled 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 [15] 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 approach for video coding is also based on the PID controller.

To be more specific, assume that there exists errors 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 errors and D predicts error in the future. For more detials about the setting of K_p, K_i and K_d, refer to [16]. To sum up, the PID controller outputs an optimal predicted value of the control variable to minimize error e_t, using all the errors 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. In addition, 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 $\mathrm{QP_t}$ and e_t can be formulated 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 $\mathbf{I}_t$ is 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 than (3) can be rewritten by

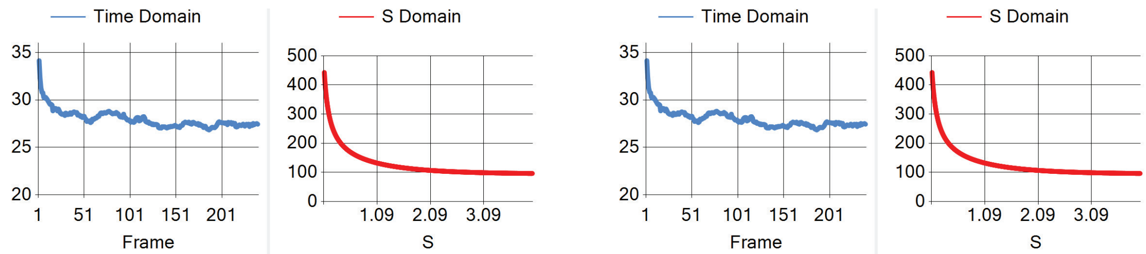
$$\Psi_{\mathbb{I}}(\mathrm{QP}_t, \mathrm{QP}_{t-1}, \cdots, \mathrm{QP}_0) = e_t. \tag{4}$$

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: Treating $\Psi_{\mathbb{I}}$ as an unknown linear time invariant(LTI) system (Input:QP_t; Output:e_t), we process time domain analysis on it. The analysis can be achieved via the following steps:

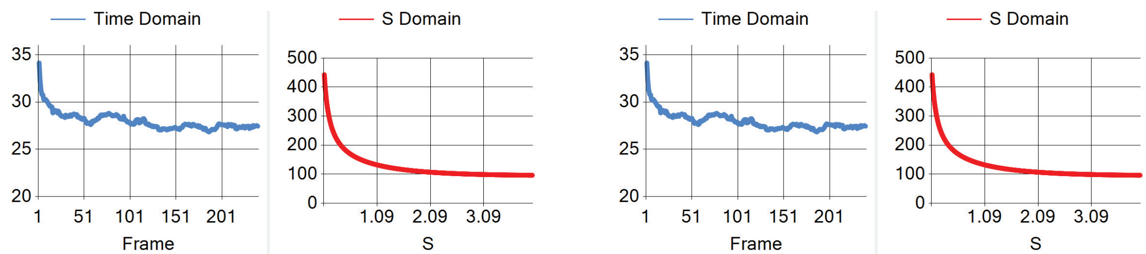
$\mathbf{Step\ 1}$: $\mathbf{Input}$ a impulse signal to the encoding system with $\mathbf{QP} = \{\mathrm{QP}_0 = \mathrm{QP}_{\mathrm{min}}, \mathrm{QP}_1 = \mathrm{QP}_{\mathrm{max}}, \mathrm{QP}_2 = \mathrm{QP}_{\mathrm{max}}, \cdots, \mathrm{QP}_t = \mathrm{QP}_{\mathrm{max}}\}$, where $\mathrm{QP}_{\mathrm{min}}$ and $\mathrm{QP}_{\mathrm{max}}$ are possibly minimal and maximal QP values, respectively.

$\mathbf{Step\ 2}$: $\mathbf{Obtain}$ the system response: $\Psi_{\mathbb{I}}(\mathbf{QP})$.

$\mathbf{Step\ 3}$: $\mathbf{Process}$ laplace transform on the pratical $\Psi_{\mathbb{I}}(\mathbf{QP})$, then $\mathbf{estimate}$ how many poles are there on its S domain. Above analysis for inter frames is shown in Figure 2 and the process is similiar for intra frames. If there is n poles on S domain of the



BasketballDrill BlowingBubbles



BasketballDrill BlowingBubbles

Figure 2: Analysis if pratical $\Psi_{\mathbb{I}}(\mathbf{QP})$ upon the state-of-the-art HEVC encoder with inter coding. For analysing $\Psi_{\mathbb{I}}(\mathbf{QP})$, we encoded 4 sequences by HM16.0, using the configuration file *encoder_lowdelay_P_main.cfg*. Here, the P frames of each sequence were encoded with $\mathbf{QP} = \{\mathrm{QP}_0 = \mathrm{QP}_{\mathrm{min}}, \mathrm{QP}_1 = \mathrm{QP}_{\mathrm{max}}, \mathrm{QP}_2 = \mathrm{QP}_{\mathrm{max}}, \cdots, \mathrm{QP}_t = \mathrm{QP}_{\mathrm{max}}\}$, in which $\mathrm{QP}_{\mathrm{min}} = 0$ and $\mathrm{QP}_{\mathrm{min}} = 51$ 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" to obtain "S Domain curve", which represents $\Psi_{\mathbb{I}}(\mathbf{QP})$ in S domain. It is obvious that "S Domain curve" had and only had one one pole, which is 0.

pratical $\Psi_{\mathbb{I}}(\mathbf{QP})$, it means $\Psi_{\mathbb{I}}$ is a n order system, further means it can be described with a n order differential equation [**?**]. Results showed that there is 1 and 0 pole for inter and intra frames respectively. So $\Psi_{\mathbb{I}}$ can be described with one order and two order differential equation respectively:

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

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

where $A_0^{\mathbb{I}}$, $A_1^{\mathbb{I}}$ and $A_2^{\mathbb{I}}$ are the coefficients derived from data analysis in Figure 2. We do not have to obtain these coefficients precisely for each $\mathbb{I}$, because our intention here is to figure out that the relationship between QP_t and e_t is modelled by a n order differential equation, and further determine n. This n is crucial when we introduce PID controller in the following section.

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 [17]. 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 modelled 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 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) satisfies 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,

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

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

for controlling quality of video coding. Obviously, we can see that QP_t is only related with the 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.

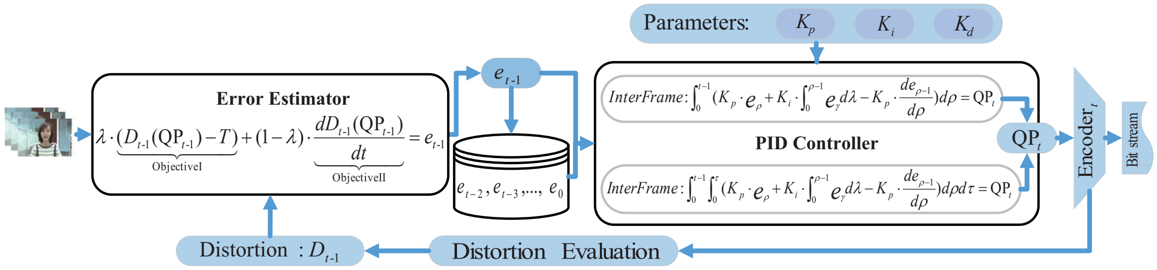


Figure 3: Framework of our PQC method.

4 Experimental results

4.1 Setting

In this section, experimental results are presented to vevify the effectiveness of our PQC method. We use all 16 video sequences from the standard video database [18]. In addition, we implemented our PQC method on the latest HM16.0 platform in our experiment. It also needs to point out that our PQC method can be qpplied to other frameworks. Furthermore, the low delay IPPP structure was chosen, using the configuration file *encoder_lowdelay_P_main.cfg*. The parameters were set by default. Besides, the state-of-the-art quality control method [14] is utilized for comparison in this paper, which is also implemented on HM16.0 for fair comparison.

Some of key parameters related to our PQC method are: $K_p = 2.12$, $K_i = 0.1$, $K_d = 0.6$ and $\lambda = 0.8$. PID paramters tuning method to determine K_p, K_i and K_d was represent in [16]. But it need to point out that the value of K_p, K_i and K_d is not unique, for they are valid in a wide range (about an order of magnitude for each paramter) due to the robustness of PID controller. As for λ, since it represents the trade off between the two objectives of quality control, it is modifiable according to different applications. We only present $\lambda = 0.8$ as an example.

4.2 Quality Control Performence

Quality error and quality fluctuation are two crucial objectives when evaluating quality control. If the actual quality are equal to the target quality assigned for each frame, there is no quality error or fluctuation. On the other hand, the large quality error or fluctuation may cause the poor perceived quality as shown in Figure 1. Therefore, the comparisons of the two objectives for default HM 16.0, [14] method and our PQC method are conducted to validate the effectiveness of our PQC method. Specifically, HM 16.0 do not take quality control into consideration. As a result, the default HM 16.0 is used to compute the target quality. The difference between [14] method and our PQC methods exists in minimizing both objectives.

In details, we tested all 16 video sequences from JCT-VC test set [18] and utilize PSNR as a measurement of quality. Table 1 presents the overall quality control performence of

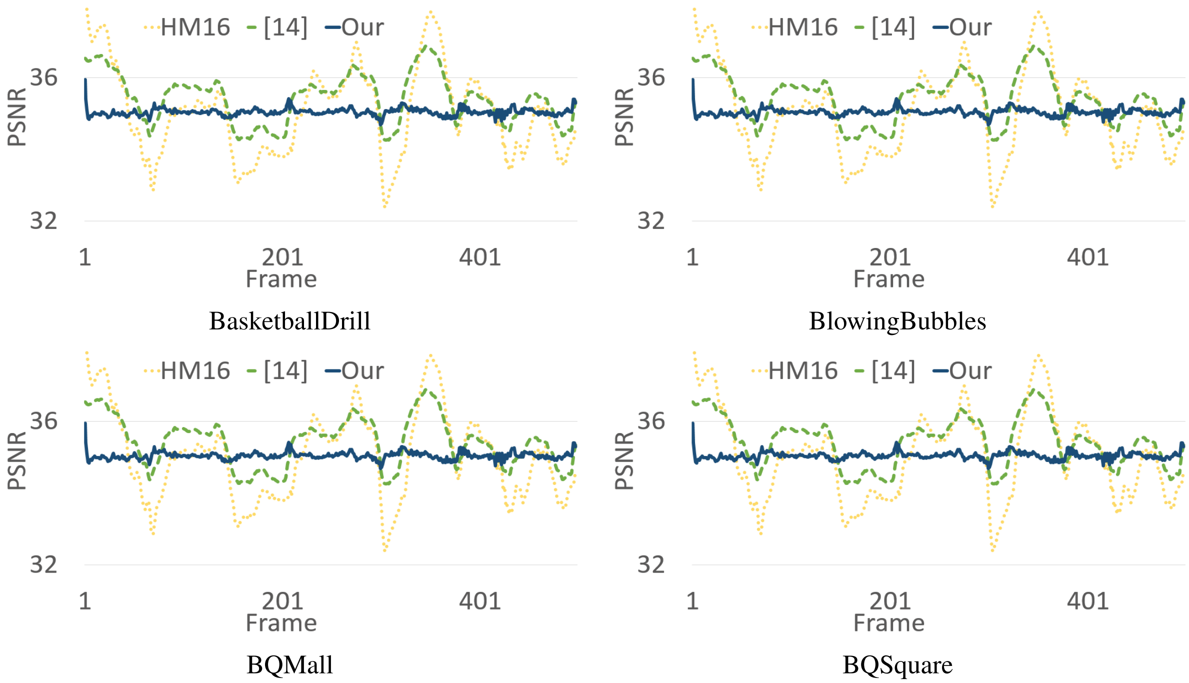


Figure 4: Quality fluctuation of conventional HM16.0, [14] and our PQC method. The quality is measured by PSNR. Similar fluctuation results can be found for other encoded video sequences, and all of them are summarized in Table 1.

our PQC method and the other two anchors, in terms of the Control Error and Quality Fluctuation over all 16 test sequences at two different target PSNR values (total up to 32 experiments), as reported in the 5th, 6th, 8th and 9th column of Table 1. In this table, the target PSNR were set to be the averaged PSNR by compressing the same sequences at fixed QP values with conventional HM16.0 encoding. The QP values are 37 and 32, as reported in the first row of Table 1. And the Control Error was computed by the relative error between the target PSNR and the averaged PSNR encoded with each method. Furthermore, the Quality Fluctuation was measured by the mean square error of PSNR of each encoded video. As can be seen from this table, our PQC method offers smaller Control Error and Quality Fluctuation on compressing all 16 sequences at different target PSNR values. Figure 4 further shows 4 of the 32 experiments with PSNR-Frame curve.

4.3 Rate-Distortion Performence

This section verifies the performance of three methods in terms of rate and distortion performence. As reported in 3th and 7th columns in Table 1, our method shares a very close averaged PSNR with the conventional HM16.0 due to our well performence in terms of quality control, but at the cost of consuming more bit rate. The total bit rate of our PQC method is evenly 0.7 and 0.1 Mbps ($\mathrm{QP} = 32$ and $\mathrm{QP} = 37$ respectively) more that of the conventional HM16.0, as reported in the last row of Table 1. However, some encoded videos even shows that our PQC method cost less bit rate than the conventional HM16.0, but still maintains a very close averaged PSNR to it. Comparing to [14], our PQC method not only has a better controlled averaged PSNR but also has a far less total bit rate.

4.4 Evaluation on Computational Complexity

Since the simplicity and efficiency of our PQC method, our method can be accomplished within 10 times of floating-point addition operations and 10 times of floating-point multiplication operations at each frame, as can be verified from (12) and (13). So there is no obvious extra computation time for our PQC method compared to the conventional HM16.0.

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 modelled in our PQC method. Then, a PID-based solution was developed to solve our quality control formulation, based on the modelled relationship between QP and control error alongside encoded frames. Since our PQC method only alternates QP values of 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., HM16.0. The experimental results show that our PQC method advances state-of-the-art quality control of HEVC in terms of both control error and quality fluctuation.

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.

[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] S. Li, M. Xu, Z. Wang, and X. Sun, "Optimal bit allocation for ctu level rate control in hevc," 2016.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] K. J. strm and T. Hgglund, “Advanced pid control,” 2006.

[16] 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.

[17] 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.

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

Table 1: Experimental results

Video	Method	QP = 32 Avg. PSNR (dB))	QP = 32 Control Error (%)	QP = 32 Quality Fluctuation (dB)	QP = 32 Bit Rate (Mbps)	QP = 37 Avg. PSNR (dB)	QP = 37 Control Error (%)	QP = 37 Quality Fluctuation (dB))	QP = 37 Bit Rate (Mbps))
BasketballDrill	HM16	33.93	–	0.73	8.5	31.59	–	0.72	4.4
	[14]	34.77	2.48	0.20	11.9	32.68	3.46	0.35	6.9
	Our	33.92	**0.01**	**0.13**	9.3	31.60	**0.01**	**0.11**	4.8
BasketballDrive	HM16	35.05	–	1.25	32.7	32.73	–	1.31	16.6
	[14]	36.43	3.94	0.71	51.3	34.27	4.70	0.83	25.5
	Our	35.05	**0.02**	**0.18**	34.5	32.73	**0.02**	**0.20**	17.3
BasketballPass	HM16	33.57	–	2.29	4.2	30.54	–	2.18	2.1
	[14]	35.94	7.08	0.93	7.7	32.11	5.16	1.05	3.4
	Our	33.56	**0.02**	**0.18**	4.7	30.54	**0.03**	**0.17**	2.4
BlowingBubbles	HM16	30.12	–	1.36	3.6	27.27	–	1.31	1.5
	[14]	31.91	5.94	0.45	5.9	28.44	4.28	0.60	2.4
	Our	30.11	**0.01**	**0.16**	3.9	27.27	**0.01**	**0.17**	1.7
BQMall	HM16	33.87	–	2.26	9.4	31.02	–	2.40	4.7
	[14]	35.35	4.34	0.82	14.0	32.34	4.26	0.93	6.8
	Our	33.87	**0.01**	**0.07**	10.6	31.02	**0.00**	**0.06**	5.4
BQSquare	HM16	30.02	–	0.86	3.4	27.15	–	1.09	1.3
	[14]	32.36	7.80	0.27	6.9	28.87	6.36	0.34	2.3
	Our	30.01	**0.01**	**0.16**	3.4	27.15	**0.02**	**0.13**	1.4
BQTerrace	HM16	32.92	–	0.94	23.6	30.65	–	1.16	8.4
	[14]	34.01	3.32	0.66	44.6	31.93	4.20	0.80	15.3
	Our	32.92	**0.01**	**0.06**	26.9	30.65	**0.00**	**0.05**	9.3
Cactus	HM16	34.46	–	0.38	26.5	32.27	–	0.46	13.0
	[14]	35.18	2.08	0.18	39.9	32.73	1.43	0.22	18.7
	Our	34.46	**0.01**	**0.06**	30.0	32.27	**0.01**	**0.06**	12.8
FourPeople	HM16	37.12	–	0.71	4.4	34.53	–	0.78	2.3
	[14]	38.36	3.34	0.15	6.9	34.46	0.20	0.14	2.7
	Our	37.12	**0.01**	**0.07**	4.6	34.53	**0.00**	**0.04**	2.8
Johnny	HM16	38.34	–	0.82	1.9	35.94	–	1.35	1.0
	[14]	39.11	2.03	0.32	2.6	36.62	1.92	0.19	1.3
	Our	38.34	**0.01**	**0.08**	1.9	35.94	**0.00**	**0.08**	1.0
Kimono1	HM16	36.58	–	1.20	12.0	33.95	–	1.60	5.9
	[14]	39.07	6.79	0.22	12.3	36.68	8.05	0.26	6.2
	Our	36.61	**0.08**	**0.32**	6.2	34.03	**0.23**	**0.54**	1.2
KristenAndSara	HM16	38.42	–	0.62	3.3	35.86	–	1.01	1.7
	[14]	39.25	2.15	0.22	5.1	36.13	0.74	0.16	2.2
	Our	38.43	**0.00**	**0.06**	3.6	35.86	**0.00**	**0.05**	2.0
ParkScene	HM16	33.72	–	0.60	13.6	31.19	–	0.65	5.9
	[14]	35.01	3.81	0.18	20.7	32.23	3.34	0.17	8.5
	Our	33.72	**0.03**	**0.16**	13.7	31.19	**0.02**	**0.15**	6.0
PartyScene	HM16	30.06	–	1.44	15.6	27.18	–	1.38	6.6
	[14]	30.71	2.18	0.37	18.9	27.29	0.41	0.37	7.7
	Our	30.06	**0.00**	**0.23**	16.7	27.18	**0.02**	**0.10**	7.7
RaceHorses	HM16	32.03	–	1.37	3.0	29.16	–	1.23	1.4
	[14]	33.66	5.09	0.73	4.4	29.85	2.36	0.61	1.8
	Our	32.02	**0.02**	**0.23**	3.1	29.17	**0.03**	**0.11**	1.5
RaceHorses	HM16	32.92	–	2.62	10.2	29.89	–	2.33	4.6
	[14]	34.14	3.70	1.13	14.9	30.40	1.70	0.94	5.8
	Our	32.92	**0.03**	**0.73**	13.4	29.90	**0.04**	**0.16**	5.8
Average	HM16	33.94	–	1.22	11.0	31.31	–	1.31	5.1
	[14]	35.33	4.13	0.47	16.7	32.31	3.29	0.50	7.3
	Our	33.94	**0.02**	**0.18**	11.7	31.32	**0.03**	**0.13**	5.2