Real-time Complexity Control for High Efficiency Video Coding

Jiunn-Tsair Fang
Ming Chuan University, Department of Electronic Engineering
No.5, Deming Rd., Taoyuan City, Taiwan, 333
e-mail: fang@mail.mcu.edu.tw

Yu-Liang Tu
National Central University, Department of Communication Engineering
No.300, Jhongda Rd., Taoyuan City, Taiwan, 320
e-mail: yltu.vaplab@gmail.com

Li-Ping Yu
National Central University, Department of Communication Engineering
No.300, Jhongda Rd., Taoyuan City, Taiwan, 320
e-mail: clare3121185@gmail.com

Pao-Chi Chang
National Central University, Department of Communication Engineering
No.300, Jhongda Rd., Taoyuan City, Taiwan, 320
e-mail: pcchang@ce.ncu.edu.tw

***Abstract*—The current video coding standard, High Efficiency Video Coding (HEVC), provides quad-tree structures of the coding unit (CU) to achieve high coding efficiency. Compared with previous standards, the HEVC encoder increases much computational complexity to levels inappropriate for applications of power-constrained devices. This work thus proposes a real-time complexity control scheme to control each frame complexity if the complexity of encoded frames is counted and its accumulated value is over the threshold. To further improve the coding efficiency, a fast CU depth decision algorithm is proposed. Experimental results show that a two-level of complexity control scheme was designed. In addition, the loss of the average BD-PSNR was about 0.23 dB and 0.27 dB as the target complexity was set to 80% and 60% of the unconstrained complexity, respectively.**

Keywords-HEVC, coding unit, complexity control

I. INTRODUCTION

The current video coding standard, High Efficiency Video Coding (HEVC), supports high resolutions of video content [1].

Comparing with the previous video standard, H.264/AVC [2], HEVC employs quad-tree structures of the coding unit (CU) to improve the coding efficiency, but the HEVC encoder also increases the computational complexity. Heavy computations hinder video applications from power-constrained mobile devices. Therefore, it becomes crucial to reduce the complexity of HEVC encoder for video applications on mobile devices.

The HEVC encoder employs the hybrid prediction coding of video compression, including the interpredition, intraprediction and a 2-D transform coding. Fig. 1 depicts the block diagram of the HEVC encoding structure to show the process of encoded bitstream in the HEVC encoder [1].

Each picture is split into block-sized regions. Each block is encoded by interprediction (except the first picture) or intraprediction. Intraprediciton uses the spatial correction from those already encoded blocks in the same picture. Interpredition uses the temporal correction from those already encoded blocks in the previous encoded pictures. The interpedction needs to search the motion vector (MV) to show the position of the reference block, and the intrapredition uses the angular similarity from the reference blocks [1].

After prediction, the difference between the original block and its prediction called residual is transform by a 2D-DCT. The transform coefficients are then scaled, quantized, entropy coded, and transmitted, as shown in Figure 1.

The encoder processes the decoding procedure to get the same data as the decoder does. That is, the encoder also reconstructs the transmitted data by inverse scaling and inverse transformed to produce the estimation of the residual signal. Some of methods, such as filter loop are also applied to smooth out artifacts created by block coding. The residual is then added to the prediction, as shown in Figure 1.

CU is the basic unit for encoding [1]. The CU uses a quadtree structure to separate its size into different candidate blocks for prediction. Figure 2 illustrates the structure of coding tree block (CTB). A CTU composes of 64×64 pixels, and each CTU can be split into four sub-TUs. There are four CU depths, which are depth 0, depth 1, depth 2, and depth 3. The corresponding CU sizes are 64×64, 32×32, 16×16, and 8×8. The best CU depth for encoding is determined by a rate-distortion optimization (RDO) process. However, it takes time to execute the RDO process. Thus, some of fast CU depth decisions were proposed to reduce the HEVC encoding computation complexity.

Two major methods are classified to reduce the complexity of the HEVC encoder. One is the fast algorithm for CU depth (or PU mode) decision. The other is the complexity control method. The fast algorithm is intended to reduce the complexity of the HEVC encoder. Its main objective is to maintain the transmission quality with slight degradation but can achieve an amount of complexity reduction during the encoding process. By contrast, the complexity control method is used for effectively maintaining the transmission quality as the target complexity is downscaled. The purpose of complexity control is not only to reduce the power consumed for a given target complexity but also to effectively control the transmission quality of a power-constrained device.

978-1-5386-8003-2/18/$31.00 ©2018 IEEE

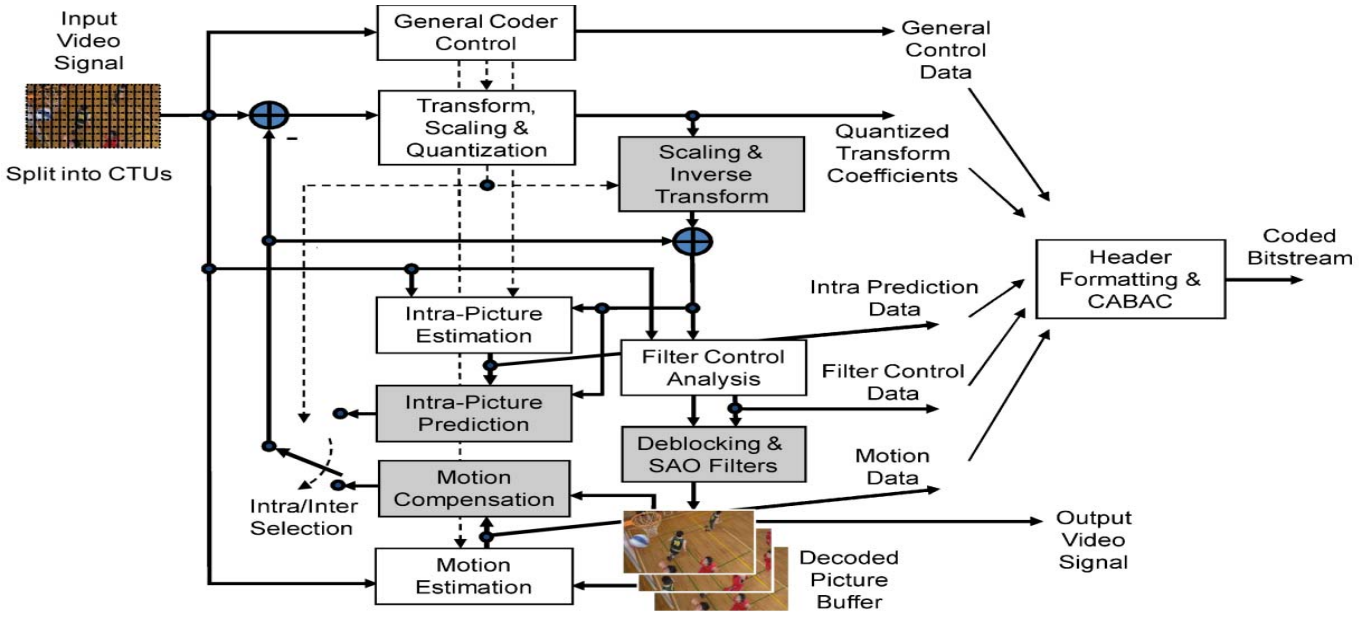

Figure 1.The encoding process of HEVC. [1].

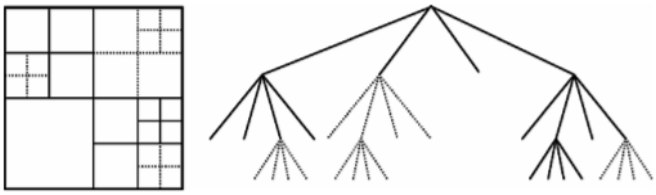

Figure 2 The CTB of HEVC. (a) CTB with its partitioning (b) The corresponding quadtree [1].

Corrêa et al. separated the sequence into constrained and unconstrained frames. Information obtained from the collocated areas in the previously encoded unconstrained frames could be used to predict the number of constrained frames and their coding-tree depths. The computational complexity of HEVC can thus be estimated and controlled. [3]. Deng et al. applied the information of neighbor large CUs (LCUs) to estimate and control the depth of current encoding LCU. Based on the target complexity, they calculated and estimated the depth of LCUs to reduce the prediction error [4]. Zhang et al. created a parameter, called motion collision count (MCC), to describe the relation between MV and CU depth, and then determined the number of CU-splitting from the CU depth for complexity control [5]. Our previous work used a parabolic curve to fit complexity consumption for each P-frame within a group of pictures (GOP) under the low-delay P-frame (LDP) configuration of HEVC. Then, the ratio of complexity consumption of each frame within a GOP can be estimated, and the complexity control for frame layer can be achieved [6].

Some fast algorithms have been proposed to reduce the encoding time of HEVC. Xiong et al. proposed a pyramid motion divergence method to reduce complexity of the CU depth decision [7]. Zhang et al. used the spatial and temporal encoding parameters to increase the CU depth decision [8]. The rate-distortion (R-D) performance was analyzed to early determine the CU depth [9]. Recently, Li et al. estimated the range of the current CU depth by exploiting the temporal correlation of encoded CUs depth. They also check the coded block flag (CBF) to decide the maximum depth for the current CU and also reduce the accumulated estimation error [10]. The machine learning method has been applied for HEVC encoder. In [11], Shu et al. applied the support vector machine (SVM) for CU depth decision.

In [3-6], their complexity control schemes were to estimate the total computational complexity of the encoder, and set up the target complexity to constrain each frame computational complexity. However, these schemes have disadvantage. First, the video content is unknown, the target complexity cannot be determined in advance. Second, in the most of cases, the complexity control scheme starts as the battery charge is low. However, these schemes cannot detect the condition. This work thus provides a real-time complexity control scheme. Different to the schemes of [3-6], this work counts the complexity from each encoded frame. As the complexity of these already encoded frames reaches to the threshold, the complexity control scheme starts. In other words, the proposed method is based on the consumed complexity of the encoder, instead of based on a predefined total complexity.

To increase the coding efficiency, this work also proposed a fast CU depth decision algorithm to reduce the complexity of HEVC encoding. A parameter, average MCC, is defined to find the relation between the average MCC and the CU split ratio.

This paper is organized as follows. Section II describes the proposed method for the complexity control and fast CU depth decision algorithm. The experimental results are described in Section III. Finally, section IV provides a conclusion.

II. Complexity Control Scheme and Fast CU Algorithm

Section II describes the complexity control scheme, the fast CU algorithm, and the complexity compensation method. Following sub-sections describe these methods in detail, respectively.

A. Complexity Control Scheme

This study focuses on the low-delay P-frame (LDP) configuration of HEVC [12], because the LDP configuration is suitable for applications in low-power devices. The LDP configuration consists of one I-frame, and the rest of P-frames. Figure 3 illustrates the LDP configuration, where each GOP contains four P-frames. The QP settings for the first and third frames are QP+3, for the second frame is QP+2, and for the fourth frame is QP+1. This QP setting is the same for the remaining GOPs. Figure 1 also shows that the encoder order is the same as the picture order, meaning that this configuration is low-delay for real-time transmission.

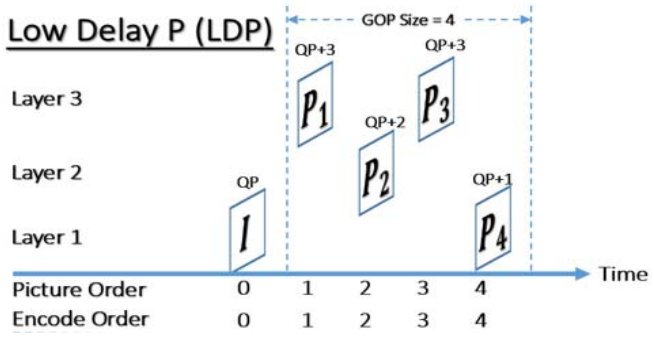

Figure 3 Hierarchical coding structure of the LDP configuration [13].

The complexity control scheme is to restrict each frame encoding complexity. This scheme starts as the complexity of encoded frames reach to the threshold. The proposed complexity control method is described as follows.

1. Set up two parameters, the initial complexity threshold (C_E), and the complexity consumption parameter (C_{EL}). The parameter C_{EL} is to record the complexity which has been consumed from those encoded frames.
2. Start encoding and counting the complexity consumption from each encoded frame, i.e., counting C_{CC} after each frame is encoded. The time consumption equals to the consumed clock divides by the CPU frequency.
3. If the complexity consumption parameter is greater than the threshold, the complexity control scheme starts. In this work, two target complexity parameters were set up. $C_{E\text{-}60}$, and $C_{E\text{-}30}$ represent the 60% and 30% of the initial complexity threshold (C_E), respectively.
4. Set up the target complexity. Different sequences may consume different frame complexity. The target complexity is set up based on the average complexity of those encoded P-frames. That is, C_s denotes the complexity of already encoded P-frames, and F denotes as the number of already encoded P-frame. So, the target complexity C_{TC} is denoted by (1)

$$C_{TC} = (C_S/F) \times R \quad (1)$$

where R is the ratio. This work designs a two-level of complexity control scheme. Thus, the target complexity in (1) has two values for the first time and second time of complexity constraint. The ratio R is 0.8 for the first time, and 0.6 for the second time of complexity constraint.

The flow chart of the complexity control scheme is plotted in Figure 4. Two steps of complexity control in this work. The target complexity for the first constraint and the second constraint are 0.8 and 0.6 of average frame complexity consumption of the non-constraint frame, respectively.

B. Fast CU Algorithm

HEVC provides CU quad-tree structures for CU. There are four depths of CU. Depth 0 represents a 64×64 block size of CU, and depth 1 represent a 32×32 block size of CU, and so on. HEVC applies a rate-distortion optimization (RDO) procedure to determine whether the parent depth of CU splits into child depth of CU, and finally determines the best depth of the CU for encoding with the best RD performance. Though the RDO procedure improves the coding efficiency, it increases computational complexity considerably. Therefore, in this work, in addition to allocating the complexity to each coding layer for complexity control, a fast algorithm for CU split decision is required to reduce the computational complexity of the HEVC encoder.

To efficiently control the complexity of HEVC encoder, a fast algorithm for CU depth decision is required. Zhang et al. created the MCC to determine the number of CU-splitting from the CU depth for complexity control [5].

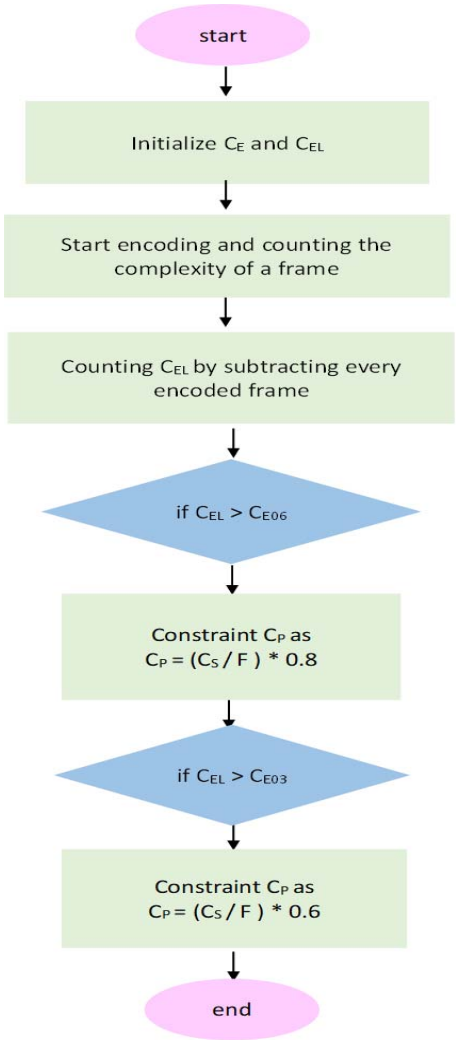

Figure 4 Flow chart of the complexity control algorithm.

The average MCC, denoted as MCC_{avg}, is the total number of MCC divided by the number of LCU in a frame. The relation between LCU depth split ratio and the average MCC was listed in Table I.

TABLE I. RELATION BETWEEN THE LCU DEPTH SPLITTING RATIO AND THE AVERAGE MCC

Sequence	QP	MCC_Avg	P_split(%)
Kimono1	27	51.75	76.1
	32	33.89	63.2
	37	23.19	49.1
	42	12.05	29.7
RaceHorses	27	64.6	96.1
	32	49.91	92.3
	37	37.19	84.6
	42	22.96	66.7
Vidyo3	27	4.24	41.7
	32	2.48	23.9
	37	2.02	18.4
	42	1.17	12.1

From the result of TABLE I, it shows that higher average MCC value has higher chance for this LCU to split into depth 1 of CU. By contract, lower average MCC value has lower chance for this LCU to split into depth 1 of CU. This is because high average MCC value means that the sequence content have more objects of movement, and therefore, the LCU layer has more chance to be encoded in the deeper depth of CU.

Based on the above analysis, simulations were designed. Several thresholds were set up to determine whether or not the current encoding LCU splits into Depth 1 of CU. The algorithm is as follows.

1. If the average MCC is not greater than 20, then the threshold equals to its average MCC times 0.7.
2. If the average MCC is greater than 20, but is less than 50, then the threshold equals to its average MCC times 0.5.
3. If the average MCC is greater than 50, then the threshold equals to its average MCC times 0.3.

C. Complexity Compensation

The actual complexity consumption may not be equivalent to the allocated complexity. Following conditions may be happened during the HEVC encoding. First, some LCUs may early terminate the coding process by using skip mode to save the complexity. An algorithm is added to distribute the extra complexity unused to the rest frames. Second, as the distributed complexity is exhausted, but the process is still going, this procedure can continue until it ends. However, the extra used complexity will be deduced from the rest frames.

As above description, when a frame consumes a complexity different from that which is allocated, the complexity allocation to the next frame is adjusted by adding the complexity compensation C_{CMP}, which is calculated using the expression

$$C_{CMP} = (A_{F_i} - C_{F_i}) / (n - n_i) \quad (2)$$

where A_{F_i} is the actual complexity consumption of the ith frame F_i, n is the total number of frames in the GOP, n_i is the ith frame, and $n - n_i$ represents the remaining uncoded frames in this GOP.

The complexity offset $A_{F_i} - C_{F_i}$ is used to adjust the complexity allocation. If the allocated complexity is more than the actual complexity consumption, then this term is negative, and the complexity allocation for the remaining frames is reduced. Conversely, when this term is positive, the complexity allocation for the remaining frames is increased. Finally, this complexity offset is divided among the remaining uncoded frames for equal distribution.

III. EXPERIMENTAL RESULTS

Simulations were designed to show the real-time complexity control scheme, and also designed to show the PSNR performance under the complexity constraint. TABLE II lists the environment for simulations. Four test sequences represent different classes with different video sizes.

TABLE II. ENVIRONMENT FOR SIMULATIONS

Reference software	HM 12.1
Sequence	Class A_ PeopleOnStreet (2560*1600)
	Class B1_Kimono1 (1920*1024)
	Class C_BasketballDrill (832*448)
	Class E_Vidyo1
FramesToBeEncoded	57
Configuration file	Low Delay P (IPPP…)
QP	22,27,32,37
CPU	Intel(R) Core(TM) i7-6700 CPU@ 3.40GHz
RAM	8.0 G bytes
System	Windows 10 - 64bit

Figures 5 and 6 shows the complexity consumption of each frame under the complexity control scheme. For better representation, Figure 5 shows the sequences of Classes A and B1, and Figure 6 shows the sequences of Classes C and E. Both figures were plotted under QP 22 and QP 37, respectively.

It shows that the sequence of Class A_PeopleOnStreet consumed the largest amount of complexity because it has the biggest size of frames. By contrast, Class E_Vidyo1 consumed the least complexity. All sequences have been constrained the complexity twice, starting in the 25th, and the 51th frame, respectively. Both figures show that the complexity for these two complexity constraint reduced about 80%, and 60%, respectively. The result is compatible to our design.

Figure 5 Complexity control for sequences of Classes A and B1.

Figure 6 Complexity control for sequences of Classes C and E.

TABLE III. THE BD-PSNR PERFORMANCE

Video	BD-PSNR (dB)	80%BD-PSNR (dB)	60%BD-PSNR (dB)
PeopleOnStreet	-0.23	-0.34	-0.31
Kimono1	-0.02	-0.05	-0.04
BasketballDrill	-0.28	-0.44	-0.62
Vidyo1	-0.1	-0.09	-0.1
AVERAGE	**-0.16**	**-0.23**	**-0.27**

Table III lists the Bjøntegaard delta peak signal-to-noise rate (BD-PSNR) [14] performance for these four test sequences. The BD-PSNR decreases as the complexity constraint increases

IV. CONCLUSION

This work proposes a real-time complexity control scheme for the HEVC encoder, and also provides a fast CU algorithm to improve the coding efficiency. The real-time complexity control scheme is based on the complexity consumption of those encoded frames. Simulation results show that the proposed real-time complexity control scheme can reflect the complexity of those encoded frame so that the complexity control scheme can be more adequate for the application in the power-constrained device.

REFERENCE

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649 – 68, Dec. 2012.

[2] T. Wiegand, G. J. Sullivan, G. Bjonte, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.* vol. 13, no.7, pp. 560–576, Aug. 2003.

[3] G. Corrêa, P. Assuncao, L. Agostini, and L. A. da Silva Cruz, "Complexity Control of High Efficiency Video Encoders for Power-Constrained Devices," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1866–1874, Nov. 2011.

[4] X. Deng, M. Xu, L. Jiang, X. Sun, and Z. Wang, "Subjective-driven complexity control approach for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 91-107, Jan. 2016.

[5] Y. Zhang, S. Huang, H. Li, and H. Chao, "An optimally complexity scalable multi-mode decision algorithm for HEVC," in *Proc. IEEE Int. Conf. Image Processing* (*ICIP*), Melbourne, VIC, pp. 2000-2004, Sep. 2013.

[6] J.-T. Fang, Z.-Y. Chen, C. R. Lai, and P.-C. Chang, "Computational complexity allocation and control for inter-coding of high efficiency video coding with fast coding unit split decision," *Journal of Visual Communication and Image Representation*, vol 40, Part A, pp. 34–41, Oct. 2016.

[7] J. Xiong, H. Li, Q. Wu, and, F. Meng, "A fast HEVC inter CU selection method based on pyramid motion divergence," *IEEE Trans. Multimedia* vol. 16, no. 2, pp. 559–564, 2014.

[8] S. Ahn, B. Lee, and M. Kim, "A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding," *IEEE Trans. Circuits Syst. Video Technol.* vol. 25, no. 3, pp. 422 – 435, 2015.

[9] T. Zhang, M.-T. Sun, D. Zhao, and W. Gao, "Fast Intra-Mode and CU Size Decision for HEVC," *IEEE Trans. Circuits Syst. Video Technol* , vol. 27, no. 8, pp. 1714-1726, Aug. 2017.

[10] Y. Li, G. Yang, Y. Zhu, X.-L Ding, and X.-M Sun, "Adaptive Inter CU Depth Decision for HEVC Using Optimal Selection Model and Encoding Parameters," *IEEE Transactions on Broadcasting,* vol. 63, no. 3, Jun. 2017.

[11] L. Zhu, Y. Zhang, S. Kwong, "Fuzzy SVM-Based Coding Unit Decision in HEVC," *IEEE Transactions on Broadcasting,* Nov. 2017.

[12] JCT-VC, "High Efficiency Video Coding (HEVC) Test Model 12 (HM12) Encoder Description," *JCTVC-N1002,* 14th JCTVC meeting, Vienna, AT, July. 2013.

[13] J. Vanne, M. Viitanen, T. D. Hämäläinen, A. Hallapuro, "Comparative rate-distortion-complexity analysis of HEVC and AVC Video Codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, No. 12, pp. 1885-1898, Dec. 2012.

[14] G. Bjøntegaard, "Calculation of average PSNR differences between RD-Curves," ITU-T SG16 Q.6 Document, VCEG-M33, Austin, US, Apr. 2001.