

Probability Density Estimation Based on the Parzen Window for Fast CU Size Decision of HEVC Intra Coding

Dokyung Lee¹, Junghyun Lee² and Jechang Jeong³⁺

^{1 2 3} Department of Electronics and Computer Engineering, Hanyang University, Seoul, Republic of Korea

Abstract. The critical problem with the practical implementation of the high efficiency video coding (HEVC) encoder is the computational complexity. Complexity increases because of its novel techniques, such as quad-tree-based block partitioning. Thus, for years, researchers have used various methods and studied multiple algorithms to reduce the computing time of the HEVC encoder. However, a fast algorithm using the Parzen window has not yet been investigated. The Parzen window is a non-parametric method that estimates probability density using statistical samples and information. The proposed method is utilized to predetermine the size of a coding unit and statistical data for the Parzen window is updated periodically. Experimental results show that the proposed algorithm reduces encoding time by 51.96 % with a small Bjøntegaard-Delta bit-rate increase of 0.56 %.

Keywords: HEVC, fast coding unit size decision, intra coding, Parzen window

1. Introduction

The latest international video coding standard, high efficiency video coding (HEVC) [1], is used for various video content and applications. HEVC is particularly useful for compression of high resolution video sequences, such as ultra-high definition video content. This high-performance standard was developed by the joint collaborative team on video coding (JCT-VC), which adopted efficient technologies such as quad-tree-based coding unit (CU) partitioning, 35 intra-prediction modes, sample adaptive offset, and advanced motion vector prediction. However, these techniques have increased the computational complexity of the HEVC encoder.

The encoding time of the HEVC standard is a critical implementation problem. Therefore, multiple studies to generate algorithms to reduce encoding time have been conducted over several years [2]-[7]. Tseng and Lai [2] proposed an algorithm for intra coding of HEVC using the standard deviation and rate distortion (RD) costs. A CU size decision algorithm, based on the Bayesian decision rule, was introduced by Cho and Kim [3]. They employed the on-line update period to renew statistical data and calculate thresholds. In our previous work [4], we utilized quadratic discriminant analysis and the Bayesian decision rule to reduce encoding time. For split decision, CU depth levels from neighboring CUs and the variance difference were used. The thresholds and data are periodically updated by the online learning phase. The support vector machine (SVM) was employed for HEVC CU partitioning process in [5]. SVM is a supervised learning method employed for classification. The authors additionally used a wrapper method based on F-score to select features for their method.

In this paper, we propose a fast CU size decision algorithm using the Parzen window (i.e., kernel density estimation) [6]. Many researchers presume that the distribution of features is Gaussian. However, this can cause degradation of coding efficiency owing to the difference between an actual and estimated distribution. Alternatively, we directly calculate the probability of features based on the Parzen window with the Gaussian kernel. Statistical samples for density estimation are updated to adapt video characteristics using an online

⁺ Corresponding author. Tel.: +82-10-5305-0369; fax: +82-2-2220-1886
E-mail address: jjeong@hanyang.ac.kr

learning phase (OLP). In previous studies such as [3] and [4], the statistics are updated every T s. However, they could not reflect the characteristics of a video sequence when a scene change occurred between OLPs. We propose an adaptive OLP decision method to solve this problem using coding information. In our proposed algorithm, we defined an early CU size decision for the HEVC encoder as a binary classification problem. Each CU in a frame can be classified into a split class, ω_s and unsplit class, ω_u . When a CU is classified into ω_s , all subsequent processes are skipped, and the block is divided into four. The CU partitioning process is finished when a CU is in ω_u .

The remainder of this paper is organized as follows. The proposed algorithm and its specification are described in Section 2. Experimental results and analysis are presented in Section 3. Finally, in Section 4, we conclude our study.

2. Proposed Algorithm

2.1. Feature Selection

Optimal HEVC encoder CU partition results of the HEVC encoder are determined by comparing RD costs of each CU depth. RD costs play the most important role in the encoding process, not only because of the CU size, but also because of the prediction mode. Therefore, we select two features: the low complexity RD cost, J_{low} for the split decision and the full RD cost, J_{full} for the termination decision.

In the HEVC encoder, the rough mode decision (RMD) for reducing processing time of the intra prediction is adopted by JCT-VC. J_{low} is used for the RMD, calculated as

$$J_{low} = SATD + \lambda_{pred} \cdot B_{mode}, \quad (1)$$

where SATD denotes the absolute sum of the Hadamard-transformed residual signal. λ_{pred} is the Lagrangian multiplier, and B_{mode} represents the number of bits for a prediction mode. For terminating a partitioning process of a CU, we utilize the statistical information of full RD cost to find the optimal prediction mode and CU size. This is defined by

$$J_{full} = D + \lambda_{full} \cdot B_{full}. \quad (2)$$

Here, D is the sum of squared errors between the original and the reconstructed images. λ_{full} denotes the Lagrangian multiplier for the full RD cost calculation, and B_{full} is the number of bits needed to encode a CU.

The statistical distribution of these features in OLP are periodically updated making the proposed algorithm adapt characteristics of video sequences and enhancing accuracy.

2.2. Density Estimation Using the Parzen Window

To classify a CU into its proper class, we exploit a nonparametric density estimation method, (i.e., the Parzen window). For the Parzen window method, various kernels are used to enhance its performance. Therefore, it is also called “kernel density estimation”. We utilize this method to calculate probability for classification. The estimated density at input, x is defined as

$$p_i(x) = \frac{1}{N} \frac{1}{h_i} \sum_{n=1}^N k\left(\frac{x - s_i(n)}{h_i}\right), \quad i \in \{\omega_s, \omega_u\}, \quad (3)$$

$$k(x) = \begin{cases} 1, & \text{if } |x| \leq 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

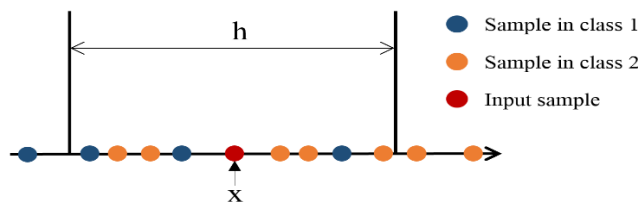


Fig. 1: Example of kernel density estimation.

Here, N denotes the number of whole samples in both ω_s and ω_u . The samples are collected during the learning phase. $s_i(n)$ represents an n -th sample of class i and h_i is a window size calculated as

$$h_i = \frac{\max_i - \min_i}{K}, i \in \{\omega_s, \omega_u\} \quad (5)$$

where \max_i and \min_i denote the maximum and minimum values of features in ω_s or ω_u . We set K to 10 through our experiments. An example of the Parzen window using the square kernel is described in Fig. 1. h denotes the window size. An appropriate class for input sample, a , can be determined as class 2, because $p_2(a)$ is larger than $p_1(a)$. Additionally, we use the Gaussian kernel function [8] for better results, thus equation (3) is redefined as

$$p_i(x) = \frac{1}{N} \sum_{n=1}^N \frac{1}{(2\pi h_i^2)^{1/2}} \exp \left\{ -\frac{(x - s_i(n))^2}{2h_i^2} \right\}, i \in \{\omega_s, \omega_u\}. \quad (6)$$

Using Eq. (6), the probability is calculated by considering both the distance from the input feature and the number of samples in the window. If the window size is relatively large, a sample that is far from the input feature may be a factor that reduces the classification accuracy. Therefore, by using the Gaussian kernel to obtain the probability, we can improve the performance of our algorithm.

2.3. Online Learning Phase

The statistical data for classification should be periodically updated to prevent misclassifications. The learning phase is the key role of supervised learning algorithms. Therefore, in our proposed algorithm, the OLP is determined adaptively based on the statistics J_{low} and the best intra mode resulting from a rough mode decision (M_{rmd}). Additionally, J_{low} and M_{rmd} are derived from coding tree units (CTUs) which are 64×64 CUs since the data of other CUs are affected by our fast CU size decision. Our OLP decision algorithm is described in Fig. 2. First, we calculate a difference in the ratio of RD cost () which is defined as follows,

$$D_j = \frac{|J_{low,A}^L - J_{low,A}^N|}{J_{low,A}^L}, \quad (7)$$

where, A denotes a CTU address. $J_{low,A}^L$ and $J_{low,A}^N$ are RD costs by RMD of OLP and a current frame (non-OLP), respectively. We define a current CTU as a different CTU when D_j is larger than 0.8 which means 80 %. If D_j is smaller than 0.8 and larger than 0.4, we consider the additional information (M_{rmd}). The proposed method checks similarity based on the directionality of the intra prediction mode. When an RMD intra mode of a current CTU (M_{rmd}^N) and a CTU from OLP (M_{rmd}^L) are in DC or Planar mode, or the difference between both modes is less than or equal to 3, we consider that a current CTU is similar to that of an OLP. Then, we can obtain a number of different CTUs (N_d) and similar CTUs (N_s). When the HEVC encoding process of a frame in non-OLP is finished, the proposed algorithm calculates the ratio of different CUs in a current frame. If the ratio is larger than a threshold (T_{OLP}), our method decides that the next frame is OLP.

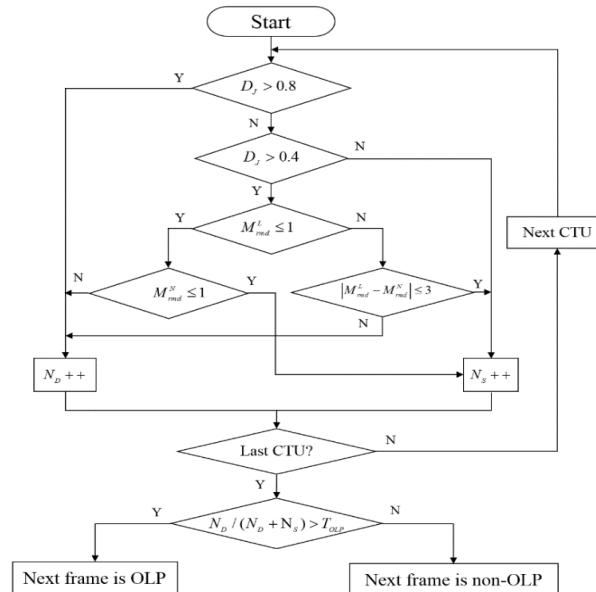


Fig. 2: A flowchart of OLP decision.

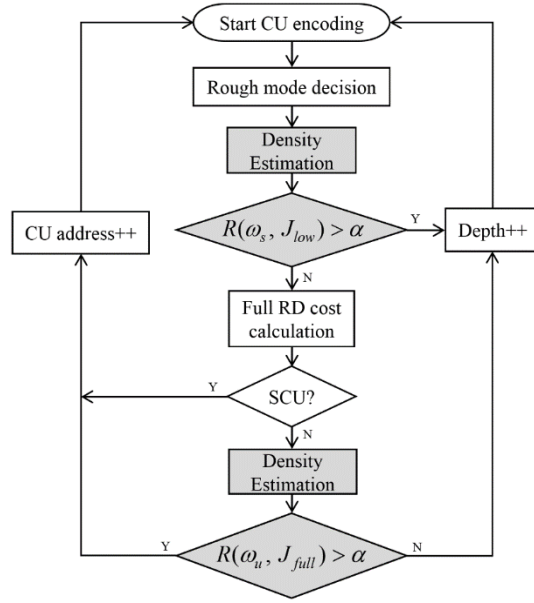


Fig. 3: Block diagram of the proposed algorithm for non-OLP frames.

When a frame is in OLP, RD costs (J_{full} , J_{low}) and the number of CU of each class (ω_s , ω_u) are stored in memory for classification. Additionally, the window size, hi in Eq. 5, is calculated as the encoding process of the frame is finished.

2.4. Proposed CU Decision Algorithm

The proposed algorithm of a frame in non-OLP is described in Fig. 3. The SCU denotes the smallest CU (8×8 CU). Additional processes by the proposed algorithm are seen in the greyed boxes. α is a user parameter that is related to the accuracy of the proposed method. CU address increases per a zigzag scan order. A current CU can be classified into ω_s or ω_u , based on following equation.

$$R(i, x) = \frac{p_i(x)}{p_{\omega_s}(x) + p_{\omega_u}(x)}, i \in \{\omega_s, \omega_u\}. \quad (8)$$

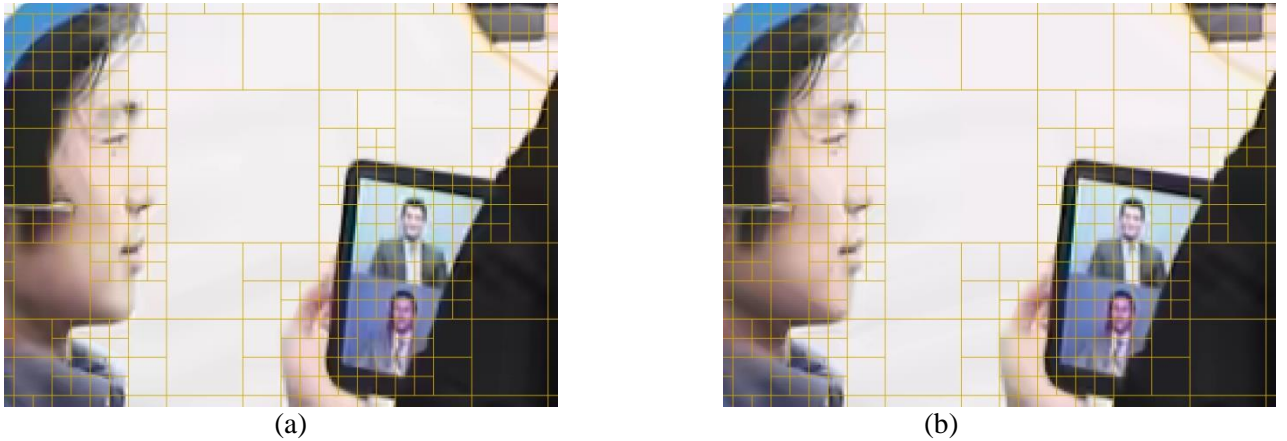


Fig. 4: CU partition results of *KristenAndSara* test sequence by (a) HM Original and (b) $\alpha=0.8$.

Here, $p_i(x)$ denotes estimated probability, from Eq. (6). For a split decision, i and x are ω_s and J_{low} , respectively. Additionally, i and x are ω_u and J_{full} , respectively, when the proposed algorithm decides if a CU is terminated. Following the RMD of intra prediction, J_{low} of a CU becomes the input of the density estimation, and the proposed algorithm decides whether the CU is pre-partitioned. When a current CU is classified into ω_s , it means that $R(\omega_s, J_{low})$ is larger than α , and the depth level of the CU is increased. If the current CU is not split early, the full RD cost is calculated and the same method is also applied to early

termination decision. When a partitioning of the current CU is terminated early, the CU address increases based on a zigzag scan order.

3. Experimental Results

To evaluate the performance of the proposed algorithm, we implemented it on an HEVC test model (HM 16.15) [9]. The test sequences and all intra configuration specified by JCT-VC [10] are used for simulation. The user parameter α varies from 0.7 to 0.8. K and T_{OLP} are set to 10 and 0.2, respectively, based on experimental results. The quantization parameters are 22, 27, 32, and 37. The proposed algorithm is compared to recent algorithms in terms of the Bjøntegaard-Delta bit-rate (BR), Bjøntegaard-Delta peak signal-to-noise ratio (PSNR) (BP) [11], and time savings (ΔT), calculated as,

$$\Delta T = \frac{T_{HM16.15} - T_{Proposed}}{T_{HM16.15}} \times 100(\%) \quad (9)$$

where $T_{HM16.15}$ and $T_{Proposed}$ represent the encoding time of HM 16.15 and the proposed algorithm, respectively.

The CU partitioning results of HM16.15 and the proposed algorithm when α is 0.8 are described in Fig. 3. CUs in the low frequency area (i. e., background) are relatively larger than CUs on the edges and in the complex area. The proposed algorithm classifies CUs accurately, because the CU partitioning structure of the proposed algorithm is similar to that of HM 16.15.

Experimental results of the effects of accuracy, α , are listed in Table I. As accuracy increases, coding efficiency improves, in terms of the BR (%) and BP (dB). The more accurately the proposed classification is applied, the longer the encoding time takes. The proposed algorithm demonstrates the best performance for Class A, because the resolution of Class A is relatively large, having statistical data. Class D possesses a relatively small resolution. Therefore, inaccurate classification can occur more than other sequences.

Comparison results of related algorithms are shown in Table II, in terms of BR (%) and ΔT (%). The proposed algorithm's BR and encoding time are better than [2] (CU decision only). The proposed algorithm ($\alpha = 0.7$) saves more than 1.73 % of encoding time compared to [3] ($\alpha = 0.1$) with a better coding efficiency. Compared to our previous work [4], the proposed algorithm ($\alpha = 0.7$) exhibits a lower coding loss and has a faster encoding time. The proposed algorithm also displays superior coding performance compared with [5], but the time saved is minor.

Table 1: Overall performance of the proposed algorithm for various α values

Classes	Test sequences	$\alpha = 0.7$			$\alpha = 0.75$			$\alpha = 0.8$		
		BR	BP	ΔT	BR	BP	ΔT	BR	BP	ΔT
Class A (2560x1600)	NebutaFestival	0.36	-0.03	70.32	0.27	-0.02	66.60	0.14	-0.01	58.45
	PeopleOnStreet	0.45	-0.03	46.15	0.37	-0.02	40.92	0.32	-0.02	42.02
	SteamLocomotiveTrain	0.61	-0.02	59.67	0.39	-0.01	50.79	0.22	-0.01	48.87
	Traffic	0.29	-0.02	46.55	0.21	-0.01	43.72	0.13	-0.01	41.68
Class B (1920x1080)	BasketballDrive	0.60	-0.02	57.93	0.42	-0.01	53.32	0.23	-0.01	49.14
	BQTerrace	0.53	-0.03	57.72	0.42	-0.02	53.14	0.36	-0.02	54.54
	Cactus	0.34	-0.01	46.79	0.30	-0.01	44.49	0.24	-0.01	43.04
	Kimono	1.18	-0.04	62.62	0.96	-0.03	58.25	0.35	-0.01	51.80
	ParkScene	0.18	-0.01	40.75	0.17	-0.01	36.13	0.08	0.00	37.20
Class C (WVGA)	BasketballDrill	0.59	-0.03	48.64	0.51	-0.02	42.60	0.47	-0.02	43.63
	BQMall	0.57	-0.03	48.19	0.48	-0.03	42.92	0.47	-0.03	43.55
	PartyScene	0.36	-0.03	46.64	0.31	-0.02	41.32	0.28	-0.02	43.28
	RaceHorses	0.37	-0.02	43.90	0.23	-0.01	36.21	0.20	-0.01	37.25
Class D (WQVGA)	BasketballPass	0.71	-0.04	53.80	0.57	-0.04	48.49	0.47	-0.03	48.12
	BlowingBubbles	0.28	-0.02	44.00	0.18	-0.01	39.96	0.16	-0.01	39.27
	BQ square	0.64	-0.05	45.53	0.60	-0.05	44.35	0.54	-0.04	46.04
	RaceHorses	0.68	-0.04	41.43	0.58	-0.04	38.74	0.42	-0.03	36.86
Class E (720p)	FourPeople	0.51	-0.03	53.33	0.32	-0.02	49.41	0.27	-0.02	49.14
	Johnny	1.05	-0.04	62.63	0.90	-0.04	59.46	0.85	-0.04	58.16
	KristenAndSara	0.86	-0.04	62.56	0.77	-0.04	59.13	0.65	-0.03	59.76
Average		0.56	-0.03	51.96	0.45	-0.02	47.50	0.34	-0.02	46.59

Table 2: Comparison to state-of-the-art algorithms

Algorithms	BR	ΔT
Proposed ($\alpha = 0.7$)	0.56	51.96
Proposed ($\alpha = 0.75$)	0.45	47.50
Proposed ($\alpha = 0.8$)	0.34	46.59
[2] (CU decision)	0.50	40.00
[3] ($\alpha = 0.1$)	0.60	50.20
[4] ($\gamma = -0.5, S_{OLP} = 1$)	0.58	50.35
[5]	1.17	52.99

4. Conclusion

In this study, an early split and termination decision for CU of the HEVC encoder were proposed, based on the Parzen window and a learning system. We utilized the Parzen window to determine the proper class of CUs and selected the Gaussian kernel to improve performance. To adapt the characteristics of video sequences, we exploited the OLP to update the statistical information needed for the proposed algorithm. Simulation results demonstrated that our algorithm reduces encoding time by 53.77 % with negligible coding efficiency loss (0.61 %) when α is 0.7. Compared to other algorithms, the proposed method shows superior performance in terms of time savings and the coding efficiency.

5. Acknowledgements

This work was supported by the ICT R&D program of MSIP/IITP. [2014-0-00670, Software Platform for ICT Equipment]

6. References

- [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.* 22 (12) (2012) 1649–1668.
- [2] C. Tseng and Y. Lai, Fast coding unit decision and mode selection for intra-frame coding in high-efficiency video coding, *IET Image Process.*, 10 (3) (2016) 216–221.
- [3] S. Cho and M. Kim, Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding, *IEEE Trans. Circuits Syst. Video Technol.* 23 (9) (2013) 1555–1564.
- [4] D. Lee and J. Jeong, Fast intra coding unit decision for high-efficiency video coding based on statistical information, *Signal Process.: Image Commun.* 55 (2017) 121–129.
- [5] D. Liu, X. Liu and Y. Li, Fast CU Size Decisions for HEVC Intra Frame Coding Based on Support Vector Machines, *2016 IEEE 14th Intl. Conf. on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl. Conf. on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, Auckland, 2016, pp. 594–597.
- [6] H. Zhang and Z. Ma, Fast Intra Mode Decision for High Efficiency Video Coding (HEVC), *IEEE Trans. Circuits Syst. Video Technol.*, 23 (4) (2014) 660–668
- [7] L. Shen, Z. Liu, X. Zhang, W. Zhao and Z. Zhang, An Effective CU Size Decision Method for HEVC Encoders, *IEEE Transactions on Multimedia*, 15 (2) (2013) 465–470.
- [8] C. M. Bishop, Pattern Recognition and Machine Learning, *Springer*, USA: Springer, 2006.
- [9] HM 16.15 Reference Software [Online]. Available: <http://hevc.kw.bbc.co.uk/trac/browser/tags/HM-16.15>.
- [10] F. Bossen, Common Test Conditions and Software Reference Configurations, document JCTVC-K1100, in: *Joint Collaborative Team on Video Coding (JCT-VC) of ISO/IEC and ITU-T*, Shanghai, China, October, 2012.
- [11] G. Bjøntegaard, Calculation of Average PSNR Differences between RDCurves, document VCEG-M33, Austin, TX, USA, April, 2001