Chapter 33
A New Fast Encoding Algorithm Based on Motion Activity for High Efficiency Video Coding (HEVC)

Jong-Hyeok Lee, Kalyan Goswami and Byung-Gyu Kim

Abstract High efficiency video coding (HEVC) has three units such as coding unit (CU), prediction unit (PU), and transform unit (TU). It has too many complexities due to improve coding performance. We propose a fast algorithm which can be possible to apply for both CU and PU parts. To reduce the computational complexity, we propose based on rate distortion cost of CU about the parent and current levels to terminate the CU decision early. In terms of PU, we develop fast PU decision based on spatio-temporal and depth correlation for PU level. The Experimental results verify that the proposed algorithm provides up to 51.70 % of time reduction for encoding with a small loss in video quality, compared to HEVC Test Model (HM) version 10.0 software.

Keywords HEVC · Coding tree block (CTB) · Coding unit (CU) · Prediction unit (PU) · CU splitting · PU decision

33.1 Introduction

A New generation video coding standard, called a high efficiency video coding (HEVC) [1], has been developed by the Joint Collaborative Team on Video Coding (JCT-VC) group. The JCT-VC is a group of video coding experts created by ITU-T Study Group 16 (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG) in 2010.

J.-H. Lee · K. Goswami · B.-G. Kim (✉)
Department of Computer Engineering, Sun Moon University, Asan, Republic of Korea
e-mail: bg.kim@mpcl.sunmoon.ac.kr

J.-H. Lee
e-mail: ljh0607@mpcl.sunmoon.ac.kr

K. Goswami
e-mail: kalyan_goswami@mpcl.sunmoon.ac.kr

Although coding efficiency improvement is solved it has more computational complexity than the H.264/AVC [2] because of tools having high complexity and high resolution of sequences. Therefore, reduction of encoding time with convincing loss is an interesting issue.

The video encoding process in the HEVC has three unit for block structure: (a) a coding unit (CU) is basic block unit like macroblock in the H.264/AVC, (b) a prediction unit (PU) for performing motion estimation, rate-distortion optimization and mode decision, (c) a transform unit (TU) for transform and entropy coding. Encoding process encode from Coding Tree Block (CTB) having largest block size to its 4 child CUs, recursively.

In [3], a tree pruning algorithm is proposed that makes an early termination CU. Shen et al. [4] proposed an early CU size determination algorithm. In [5], a motion vectors merging (MVM) method is proposed that reduce the inter-prediction complexity of the HEVC. The MVM algorithm decide the PU partition size by using motion vectors.

This paper is organized as follows: In Sect. 33.2, the early CU splitting termination (CST) and fast PU decision method is described for proposed algorithm. Section 33.3 presents the coding performance of suggested algorithm combined by two methods. In Sect. 33.4, Concluding comments are given.

33.2 Proposed Work

33.2.1 Fast Coding Unit Splitting Termination (CST) Method

The main objective is to select the lowest RD cost before all combinations are finished. We have designed the ratio function of $\mathrm{CU_d}(i)$ at depth ‘d’ can be define as:

$$r_i = \frac{RD_{cost}(CU_d(i))}{RD_{cost}(CU_{d-1})} \quad where \quad \sum_{i=0}^{3} r_i \leq 1 \tag{33.1}$$

When a CU is split, it will be divided up into 4 child CUs. Therefore, we can define a value of ratio function for each newly created CUs. This parameter r_i is a ratio of the RD costs of the current CU and its parent CU. When the ratio function of a child is lower than its siblings, it has low amount of chance to split in the next depth level. Accordingly, this parameter can be used as a threshold for making the decision to split a current CU to next depth level.

In previous algorithm [3], if a CU is coded as SKIP mode then no further splitting of a CU. Motivating from this fact, we have explored other prediction modes of the PU. INTRA and INTER predictions have different kinds of PU types. INTER modes are 2N × 2N, 2N × N, N × 2N, N × N (presently, this mode is

Table 33.1 Classified weight factors for all prediction modes according to the motion activities

Mode	Motion activity	PU_WF (γ_i)
SKIP	Homogeneous region with motionless	0
Inter 2N × 2N	Slow motion	1
Inter 2N × N, N × 2N	Motion between slow or moderate	2
Other inter and intra	Complex motion or texture	3

not used), 2N × nU, 2N × nD, nL × 2N and nR × 2N. INTRA modes are 2N × 2N and N × N. SKIP mode is consisted of 2N × 2N size. A CU is divided into 4 CUs of lower dimensions after the completion of all mode calculations. PU modes are processed recursively from the CTB to all children CU.

Table 33.1 have classified all the prediction modes according to motion activities. This kind of motion activity was explored in good approach in the H.264/AVC codec [6, 7].

Encoded CU has high chance for no splitting in SKIP mode (PU_WF is 0). To more investigate for PU_WF is 1 or 2, We have calculated a local average of the RD cost values of all encoded CUs. In this process, the final PU mode and the corresponding RD cost values are checked after encoded CU. Equations (33.2) and (33.3) perform after encoding of a CU check dimension (d) of CU and PU_WF (***wf***) in terms of satisfying ***wf*** < 3.

$$sum_d^{wf} = \left(\mathrm{avg}_d^{\mathrm{wf}} \times count_d^{wf}\right) + RDcost_d^{wf}, \tag{33.2}$$

$$avg_d^{wf} = \frac{sum_d^{wf}}{count_d^{wf} + 1} \tag{33.3}$$

An absolute motion vector (MV) is also calculated in proposed method. An average of the motion vectors can achieve from both List_0 and List_1 in the HM reference software. We only use magnitudes of the MV directions for the x and y in order to calculate simply as shown in Eq. 33.4.

$$\mathrm{MV}_{\mathrm{abs}} = 0.5 \times \left(\sum_{i=0}^{1} |MV_x(i)| + |MV_y(i)|\right) \tag{33.4}$$

By using Eq. 33.4, we can know the motion of CU with directly relationship for MV_{abs}. The CU which has low MV_{abs} value has high change that will be not divided in current hierarchy of CU. According to above mentioned method, the proposed algorithm can determine which CU need to splitting early. The CU decision part in our algorithm can be divided into two stages. The first stage is to consider the CTU case. In terms of the CTU, we do not have any information from higher levels. Hence, ratio function cannot be calculated. In the second stage, other higher level CUs are considered.

This proposed CU splitting termination (CST) algorithm is check the PU mode weighting factor initially. If PU_WF is 0 meaning SKIP mode, the decision is taken directly that there is no need splitting. If PU_WF is 1 or 2, then we cannot make any decision directly. Hence, this method is divided into two stages in order to use different checking. For the CTU case, this algorithm only use MV_{abs} and local average RD cost of equivalent dimension of PU for PU_WF = 1 and 2. If RD cost of current PU is less than average RD with same dimension and MV_{abs} is 0, Encoder is select that there is no need splitting in the CTU case for PU_WF = 1 and 2.

Otherwise, ration function, MV_{abs} and local average RD cost of suitable dimension of PU for PU_WF = 1 and 2 are used in non-CTU case. A non-CTU case is same process. Since it can use ration function with higher level's information, if the RD cost is less than average RD cost of suitable dimension of PU and ratio function (R) is less than 0.25, then it also decide that CU is no need to split.

33.2.2 Fast Prediction Unit Decision (FPU) Method

In this section, we propose an effective prediction unit selection method based on correlation and block motion complexity (BMC) that can be applied together with the CST above proposed. A natural video sequence has high spatial and temporal correlations. To analysis these correlations, we define predictor set (Ω) such as PU_1 to PU_7. Figure 33.1 shows position from current CU for each PU_i.

To define block motion complexity, we have designed weight factor for according to location from current CU as Table 33.2. Table 33.2 indicates to land more weight in spatial correlation and upper level (depth 'd−1') from current depth 'd' than other position. Based on the designed mode weight factor (Table 33.1) for motion activities, we defined a block motion complexity (BMC) [8] according to the predefined mode weight factors (γ) and a group of predictors in Ω in order to estimate the PU mode of the current CU.

The BMC has been represented as follow [8]:

$$BMC = \frac{\sum_{i=1}^{N} W(i,l) \cdot k_i \cdot r_i}{\sum_{i=1}^{N} W(i,l) \cdot k_i}, \tag{33.5}$$

$$W(i,l) = w_i + f(i,l), \tag{33.6}$$

$$f(i,l) = round(l \cdot T_{level}) \cdot T_t \cdot Tk_i \tag{33.7}$$

where N is the number of PUs equal to 7, $W(i, l)$ is function of the weight factor for each mode. γ_i is weight factor for mode of PU_i in Ω. The k_i denote which PUs have available. If current CTB is in boundary region in the frame, then above or left PU information are not available. When PU_i is available, k_i is set to 1; otherwise, k_i is equal to 0.

Equation 33.6 is an adaptive weighting factor function. The value of w_i is weight factor for PU_i. It has property of $\sum_{i=1}^{N} w_i = 1$. A value of \$l\$ denotes the

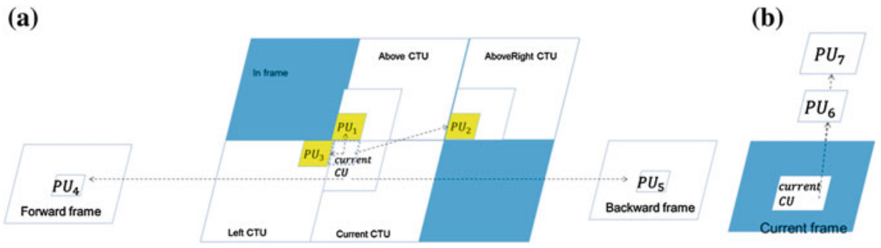

Fig. 33.1 The position of adjacent PUs from current CU (CU_0) for **a** spatio-temporal and **b** depth correlation

Table 33.2 Weight factor for PUs (default setting and changing weight factor adaptively according to level of temporal layer)

Default setting	Index (i) in classified PUs	1	2	3	4	5	6	7
	w_i	0.2	0.2	0.2	0.05	0.05	0.2	0.1
Changes of weight factor for PUs	Index (i) in classified PUs	1	2	3	4	5	6	7
	w_i	0.2	0.2	0.2	0.1	0.1	0.1	0.1

temporal decomposition layer. The HEVC consist of three layers in 8 GOP according to distance between frames. Accordingly, temporal layer can represent $\text{Level}_{1,2,3,4}$ having 8, 4, 2, and 1 distance. If current frame close with reference frame, it has strong correlation than long distance case. Therefore, setting weight factors is changed when PU has 1 and 2 distance adaptively as second row in Table 33.2. $\text{Level}_{1,2,3,4}$ is set to 0.02 and is used for normalization of temporal level as l. Tk_i is a control value for changing the weight factors. Tk_i is a control value used to increase or decrease weight factors. It is set to 1, −2, or 0. PUs having the temporal correlations with PU_4 and PU_5 are set to be 1. Also, for a PU_6, it is set to be −2. The others are set to 0.

$$\begin{cases} BMC < Th_1, & Slow\ motion, \\ Th_1 \le BMC < Th_2, & Medium\ motion, \\ Th_2 \le BMC, & Complex\ motion\ or\ texture \end{cases} \qquad (33.8)$$

where Th_1 and Th_2 are set to 1 and 3 as the mode weight factor for slow motion and complex motion or texture in PU_WF of Table 33.1. If the BMC is slow motion, the proposed method performs SKIP mode and Inter 2N × 2N. In case of medium motion, mode decision process performs modes of slow motion, Inter 2N × N and N × 2N. When motion is complex, all modes are performed, then mode search process selects the best one.

Table 33.3 The performance summarization of original HM reference software 10.0 and the proposed algorithm

	CST			FPU [8]			MVM [5]			Proposed		
	ΔBit (%)	ΔY − PSNR	ΔT (%)	ΔBit (%)	ΔY − PSNR	ΔT (%)	ΔBit (%)	ΔY − PSNR	ΔT (%)	ΔBit (%)	ΔY − PSNR	ΔT (%)
Avg. of Class A	0.5	−0.03	−15.86	0.61	−0.03	−38.35	0.87	−0.03	−22.98	1.93	−0.09	−43.66
Avg. of Class B	−0.4	−0.03	−15.34	0.38	−0.02	−45.81	0.50	−0.01	−20.11	0.20	−0.06	−51.70
Avg. of Class C	0.66	−0.05	−25.56	0.64	−0.04	−41.06	0.70	−0.02	−25.48	2.80	−0.14	−48.42
Avg. of Class D	0.44	−0.07	−17.04	1.06	−0.05	−44.76	0.91	−0.02	−23.12	3.42	−0.18	−51.09
Total average	0.3	−0.04	−18.45	0.67	−0.04	−42.50	0.75	−0.02	−22.92	2.09	−0.12	−48.72

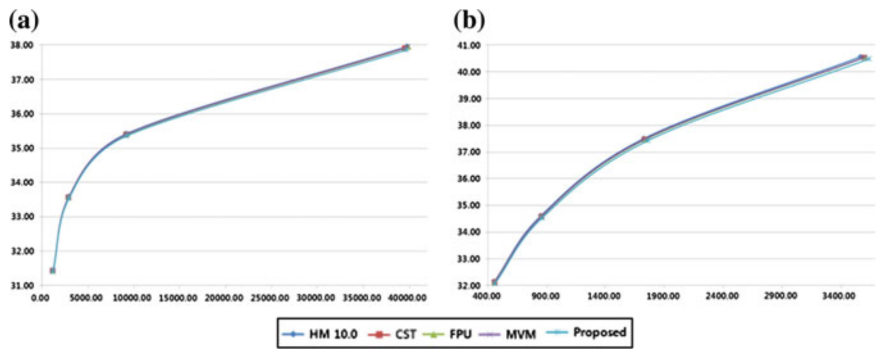

Fig. 33.2 Rate-distortion (*RD*) *curves* for **a** BQTerrace and **b** BasketballDrill sequences for Class B and Class C in random access, main condition

33.3 Experimental Results

The proposed algorithm was implemented on HM 10.0 (HEVC reference software). Test conditions were random access using RA-Main. Standard sequences with 50 frames were used from two or three sequences per each Class with various QP values (22, 27, 32, 37). Details of the encoding environment can be seen in JCTVC-L1100 [9].

Table 33.3 shows results for comparisons between the original HM 10.0 software and the proposed algorithm without any fast options. Moreover, we have considered 3 other algorithms (namely CST, FPU and MVM) which are shown in this table. The time reduction performance of the proposed method is almost 48.72 % on average with some loss as 2.09 % and 0.12 (dB) losses both in bit-rate and Y-PSNR, respectively. It can be noted from the table, that our proposed algorithm gives superior result in terms of time reduction compared with other fast algorithm. However, it suffers from marginal bit rate and PSNR loss. But if we consider the time reduction, then the loss is negligible.

Figure 33.2 shows the RD performance. In Fig. 33.2a, it is shown that for the bigger QP values larger loss are generated for bit-rate. However, the proposed method is very similar to the original HM 10.0 software. There is negligible loss of bit-rate in Fig. 33.2b.

Although our algorithm has some loss for bit-rate and PSNR, it has significant time saving performance 43.66 %, minimum, and 51.70 %, max. In performance of Class B, the proposed algorithm can achieve very little loss as 0.2 % of bit-rate and 0.06 (dB) of PSNR, and 51.70 % of good time saving performance without any fast options.

Even if our proposed algorithm has some loss, it shows performance more than a other fast algorithm about time reduction. Our method also has good structure that can be running with other fast options in HEVC standard.

33.4 Conclusions

We have proposed a new fast coding algorithm both of early CU splitting termination (CST) and fast PU decision (FPU). The CST algorithm are used the RD costs for different CU dimensions and motion complexity for PU level. The FPU method is based on spatial, temporal and depth correlation information and adaptive weighting factor design using temporal distance. By combine with CST and FPU, our algorithm achieved, on average, a 48.72 % of time saving over the original HM 10.0 software with 2.09 % of bit-rate loss. Our algorithm is useful to where needed much time reduction like real-time video encoding systems.

Acknowledgments This research was supported by the KCC (Korea Communications Commission), Korea, under the ETRI R&D support program supervised by the KCA (Korea Communications Agency) (KCA-2012-11921-02001).

References

1. Sullivan GJ, Ohm JR, Han WJ, Wiegand T (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circuits Syst Video Technol 22(12):1649–1668
2. Wiegand T, Sullivan GJ (2007) The H.264/AVC video coding standard. IEEE Signal Process Mag II:148–153
3. Choi K, Jang ES (2012) Fast coding unit decision method based on coding tree pruning for high efficiency video coding. Opt Eng Lett
4. Shen L, Liu Z, Zhang X, Zhao W, Zhang Z (2013) An effective CU size decision method for HEVC encoders. IEEE Trans Multimedia 15(2):465–470
5. Sampaio F, Bampi S, Grellert M, Agostini L, Mattos J (2012) Motion vectors merging: low complexity prediction unit decision heuristic for the inter prediction of HEVC encoders. In: International conference on multimedia and Expo, July 2012, pp 657–662
6. Hosur PI, Ma KK (1999) Motion vector field adaptive fast motion estimation. In: International conference on information, communications and signal processing (ICICS'99)
7. Zeng H, Cai C, Ma KK (2009) Fast mode decision for H.264/AVC based on macroblock motion activity. IEEE Trans Circuits Syst Video Technol 19(4):1–11
8. Lee JH, Park CS, Kim BG, Jun DS, Jung SH, Choi JS (2013) Novel fast PU decision algorithm for the HEVC video standard. In: IEEE International conference on image processing, pp 1982–1985
9. Bossen F (2013) Common test conditions and software reference configurations. In: Joint collaborative team on video coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 12th meeting, Jan 2013