Fast Coding Unit Decision Algorithm for HEVC

Wei-Jhe Hsu and Hsueh-Ming Hang
Department of Electronics Engineering, National Chiao-Tung University, Hsinchu, Taiwan
E-mail: hsu761001@gmail.com; hmhang@mail.nctu.edu.tw

***Abstract*— HEVC adopts a flexible Coding Unit (CU) quadtree structure. With more flexible CU size selection, the coding efficiency of HEVC increases significantly but its complexity is much higher than that of H.264/MPEG-4 AVC. To reduce computational complexity, we propose a fast algorithm, which consists of splitting decision and termination decision, in constructing the CU quadtree. This scheme is designed to be complementary to the current three fast tools included in HEVC TM5.0. In other words, when it is combined with the existing fast CU tools, it still provides additional time savings. The time reduction of our scheme is most noticeable on HD pictures. In comparison with the original HM5.0, our proposed method averagely saves about 43% encoding time and the BD rate increases by about 2.2% for the HD test sequences.**

Keywords-- HEVC; Coding unit (CU); Time reduction.

I. INTRODUCTION

Aiming at higher compression efficiency, the international JCT-VC is currently developing the next generation standard, High Efficiency Video Coding (HEVC) [1]. With a much higher encoder complexity, HEVC is able to achieve a 50% bitrate reduction compared to H.264/MPEG-4 AVC. In this paper, we design fast decision schemes in constructing the CU quadtree to reduce the computation complexity. The remaining sections of this paper are organized as follows. Section II describes briefly the process of CU quadtree decision. Section III introduces the existing fast algorithms included in HM5.0 [2]. Our proposed algorithm is described in Section IV. Section V presents the experimental results and discussions. Section VI concludes our work.

II. OVERVIEW OF HEVC CU QUADTREE

In this section, we introduce the Coding Unit (CU) quadtree decision flow defined in HM5.0 [2] with the low complexity setting. In the HEVC specifications, CU is a $2N \times 2N$ square block and *2N* can be 64, 32, 16, or 8. The largest CU is also called LCU. In HEVC, a slice in a frame is composed of many LCUs, and a large CU can be divided into four smaller CUs. Each partitioned CU can be recursively split until the smallest size CU is reached, in which 4 depths are allowed in HM 5.0. As one $2N \times 2N$ CU is processed in each depth, the encoder analyzes the rate-distortion (RD) cost of all possible prediction modes. The Prediction Unit (PU) is defined only on the leaf node of CU at each depth level. PUs can be further partitioned but the partitions are confined to be inside their CUs. The prediction modes can be the skip, the intra, or the inter modes as shown in Figure 1. Another block type defined in HEVC is Transform Unit (TU), which specifies the transform size.

At the same CU depth, the RD cost of every prediction mode is calculated, and all the costs are compared to determine the best mode for the CU at this depth. Next, the encoder compares the RD costs of the best partitioned modes at different depths. The tree structure of CU splits from top to bottom, but HEVC employs the G-BFOS algorithm to deicide the optimal structure [3], which makes pruning decision from bottom to top to reduce redundant comparisons.

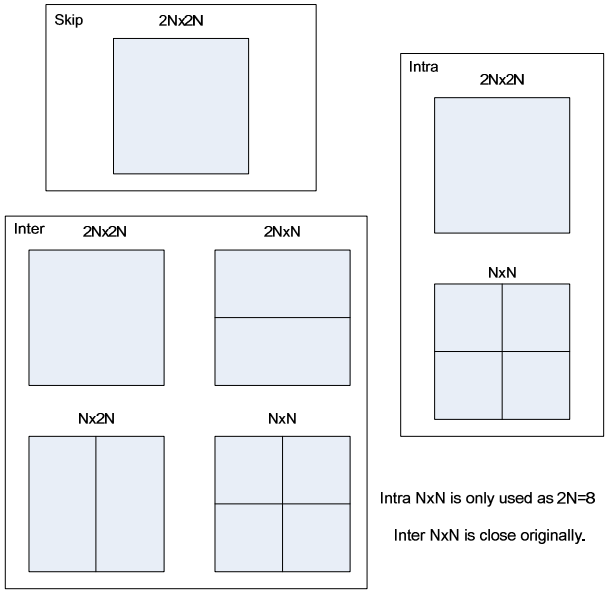

Figure 1. Possible PUs for $2N \times 2N$ CU in low complexity setting.

III. EXISTING FAST ALGORITHMS IN HM5.0

As described in the last section, the HEVC encoder computes a large amount of RD costs (of various modes) to select the best CU size, PU partition, and TU structure. The encoder spends a huge amount of computations on PUs and TUs in a CU quadtree to identify the lowest RD cost. Therefore, many researchers propose efficient methods to build quadtree nodes to reduce the complexity. There are 3 existing schemes in the literature, namely, fast encoder setting (FEN) [2], early CU termination (ECU) [4], and cbf fast mode decision (CFM) [5]. FEN has been included in the earlier HM version, and it is turned on in the original setting. Thus, we only describe the more recent 2 schemes in this section.

A. Early CU Termination (ECU)

ECU is a fast CU decision method using early termination based on the optimal PU mode, and this approach was proposed by Choi et al. [4]. Based on the analysis of condition probability of the CU depth selection, the CU depth check is skipped for all the next sub-CUs when the RD cost of the skip mode is minimum in the current CU. ECU yields

approximately 42% time reduction in encoding time with negligible loss on the luma BD-rate [6] in HM3.1.

B. Cbf Fast Mode Decision (CFM)

The RD costs for all allowed PUs in each depth are examined to ensure the optimal prediction, but the exhaustive search takes a lot of time. The coded block flag (cbf) is a good indicator to estimate the benefit of using prediction. After the residual block is transformed and quantized with a suitable TU structure, if all the coefficients in this residual block are zero, the cbf is set to 0, which means the prediction is sufficient (no residual coefficients coding). Otherwise, cbf is 1. Gweon et al. proposed a CFM algorithm [5] that uses this cbf property, and reduces about 41.2% computational complexity with the luma BD-rate loss 0.85% in HM3.2. The core idea of CFM is checking three cbf values (1 luma and 2 chromas) for every PU partition. If all of them are zero, then the rest of the PU options in the current depth are skipped.

IV. Our Algorithm Design

Because HM5.0 [2] includes FEN, ECU [4], and CFM [5] for speeding up the encoder procedure, our aim is to design additional fast algorithms from different perspectives. The principle of our new tool should be different from those three existing tools, and the added tool should not reduce the performance of the existing schemes and is compatible with the CU quadtree structure in HM5.0.

A. Related Work

CU depth estimation is another possible way to reduce the complexity in CU quadtree expansion. In [7], J. Leng et al. accelerate the encoding procedure of HEVC by using the correlation of related CUs. The encoder uses the size information of neighboring CUs and the processed depth-ratio in the previous frame to limit the permissible processed depth. It achieves averagely 45% time reduction but without the RD performance evaluation. In [8], a complexity-control method is proposed by G. Correa et al., which performs the time analysis and adjusts the number of fast encoding frames of each picture group. Recording the deepest depth used in the unit of LCU in the previous frame, the encoder finds the best tree structure within the limit of the recorded depth in each LCU in the current frame.

B. Our Proposed Algorithm

In this paper, we propose the recursive splitting decision and termination decision based on an extension of the CU level algorithm in [7]. The CU-level fast decision is based on the observation that in the temporal and spatial neighborhoods, the motion and texture characteristics of those picture patches are similar. Therefore, we can predict the candidate CU depth by checking the size of its neighbor CUs (spatial) and co-located CU (temporal). Figure 2 shows the relation between the referred neighboring CUs and the current coded CU. The co-located CU means that the previous frame CU has the same position as the current encoded CU. Our algorithm executes recursively for the side length of CU equal to 64, 32, or 16.

The *splitting decision* is designed for reducing the unnecessary operations in a large size CU. When the CU RD analysis begins at the current depth and all the following conditions are satisfied, the PU mode search in the current depth will be skipped except for the $2N \times N$ / $N \times 2N$ inter modes, and then it jumps into the next depth directly.

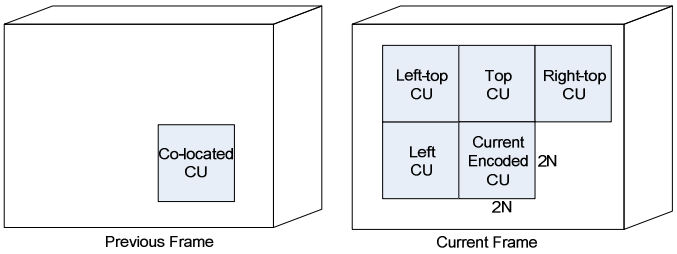

Figure 2. Reference CUs and the current encoded CU

Splitting decision conditions:
1. The co-located CU has smaller CUs.
2. All neighboring CUs have smaller CUs.
3. The current encoding frame is not I frame.

The *termination decision* prevents the encoder from building a large tree with a lot of computational complexity owing to small CUs. If the encoder has already finished the CU mode decision in the current depth, the termination decision is determined by the following conditions. When all the following conditions are satisfied, the mode selection process, whose depth is greater than the current depth, will not be conducted.

Termination decision conditions:
1. The co-located CU does not have any smaller CU.
2. 3 or more neighboring CUs do not have any smaller CU.
3. The current encoding frame is not I frame.

Some reference CUs in Fig. 2 may not exist due to the picture boundary or the coding frame order. If only one reference CU Is lost, the fast decision scheme still works under the same rules. When losing more than one reference CUs, the current CU is processed using the original procedure without our proposed schemes. Also, the splitting decision and the termination decision are inactive in I frames because a mismatched CU size in an intra frame often results in a significant amount of PSNR drop or bit rate increase.

We test 8 HD HEVC sequences (32 frames per sequence) with the proposed splitting decision and termination decision. Then, we check their Time savings (TS), luma (Y) BD rate, BD PSNR [6], and the RD curves to evaluate the performance of the **basic algorithm** (described in the above) and the average numerical values are shown in Table I. The RD curves indicate that the coding loss increases as QP gets larger, such as 32 and 37. As listed in Table I, although we achieve 50% time saving (TS), the coding loss is too high. Therefore, we like to modify the method to maintain an appropriate complexity and to improve its RD performance. Note that the time saving (TS) index is defined by Eq.(1).

$$TS(QP) = \frac{Time_{original}(QP) - Time_{fast}(QP)}{Time_{original}(QP)} \times 100\% \quad (1)$$

The fast algorithm suggested in [8] defines two types of frames: the unconstrained frames (*Fu*) and the constrained ones (*Fc*). The CUs in an *Fc* frame is coded by the fast algorithm, and the CUs in *Fu* is processed in the ordinary way. Each *Fu* is followed by *Nc* constrained frames (*Fc*) as illustrated by Figure 3.

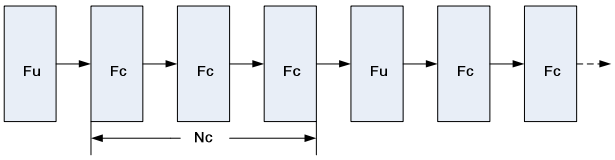

Figure 3. Example of Nc=3

Our proposed fast algorithm produces sub-optimal frames. The lower PSNR propagates and accumulates due to inaccurate predictions. Therefore, we try different *Nc* values to check the PSNR loss and pick up a tolerable bound for the PSNR drop or bit rate increase. In the experiments, *Nc* is set to 3, 6, 9, 12, and 15. Over 75% sequences have PSNR drops under 0.1dB. Because the RD performance drop is QP dependent, we find a least square liner regression, Eq.(2), to fit our collected data.

$$N_c = round(-0.32 \times QP + 14.94), \quad QP < 46 \quad (2)$$

Thus, we mark the frames with *Fu* and *Fc*, and adjust the *Nc* value based on Eq.(2). The RD performance with the *Nc* control scheme is much better than that of the original fast algorithm as shown in Table I, and we also control the average (AVG.) time saving is close to 40%.

Table I Performance of 8 HD test sequences with/without *Nc* control

Performance Index	Basic Scheme	With Nc Control
AVG. TS (QP=22)	39.65%	34.26%
AVG. TS (QP=27)	48.56%	38.70%
AVG. TS (QP=32)	53.91%	41.37%
AVG. TS (QP=37)	59.46%	41.95%
AVG. TS	50.39%	39.07%
Y BD rate (%)	5.495	1.807
Y BD PSNR (dB)	-0.135	-0.044

We notice that the computational time reduction of low QP is smaller than that of high QP. We also know that the small sized CUs are often used in low QP cases. The splitting decision occurs often in the region of clustered small sized CUs. Maybe, we can use only one inter prediction after the splitting decision to reduce the complexity further. There are two possible inter modes examined originally after the splitting decision, $2N \times N$ and $N \times 2N$. We observe that the shape is highly dependent on the sizes of neighboring CUs. If the number of small CU on the left-referenced CU is larger than that on the top-referenced CU, the encoder examines the RD cost of $2N \times N$ at the current depth. Otherwise, we check only the $N \times 2N$ prediction. We test 8 HD sequences (64 frames per sequence) in Table II to show the time saving improvement by this $2N \times N / N \times 2N$ pre-selection.

Table II Performance of $2N \times N$ / $N \times 2N$ pre-selection

Performance Index	Without	With Pre-selction
AVG. TS (QP=22)	34.28%	40.18%
AVG. TS (QP=27)	38.75%	41.95%
AVG. TS (QP=32)	42.01%	44.15%
AVG. TS (QP=37)	41.53%	42.75%
AVG. TS	39.14%	42.26%
Y BD rate (%)	2.089	2.050
Y BD PSNR (dB)	-0.052	-0.051

In this section, we firstly propose the basic algorithm for fast CU size decision, and then we design two useful additional tools to enhance its coding performance and to increase time reduction, respectively. The flowchart of our entire algorithm is drawn in Figure 4.

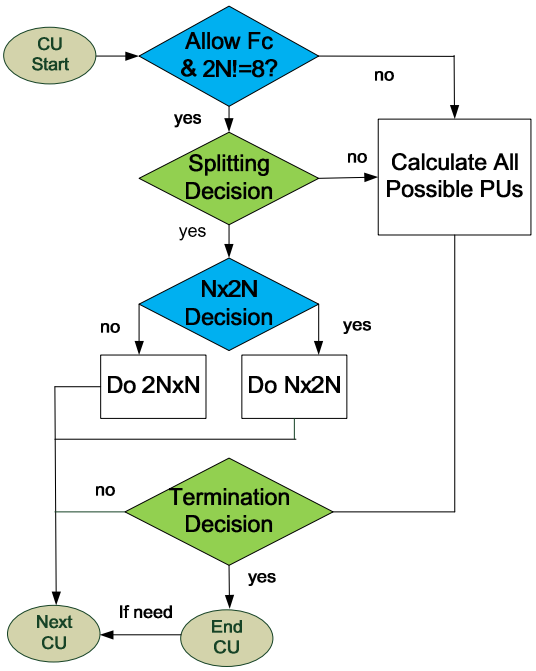

Figure 4. Proposed algorithm flowchart for processing an CU

V. Simulation Results

To verify the performance of our proposed fast algorithm, we implement it on the software HM5.0 [2], which is the reference software of HEVC encoder. The experiment settings are given in Table III, and we set GOP=1 for the simple environment (fixed QP). The time saving (TS) index is defined by Eq.(1).

A. Performance of our proposed method

The performance of our proposed fast decision scheme is listed in Table IV and Table V (100 frames per sequence). The original (reference) scheme is the original HM5.0 without ECU and CFM. We select several picture sizes to evaluate our proposed schemes. They are classified as B(1920x1080),

C(832x480), D(416x240), and E(1280x720) in the HEVC standard test sequences.

Table III Experiment conditions

Configuration: Low delay P with low complexity
MaxCUsize: 64×64
MaxCUdepth: 4
Search mode: EPZS pattern
Search range: ± 64
Reference frame: The previous frame
Group of Picture (GOP): 1
Used QP values : 22, 27, 32, and 37
Sequence type: IPPP (Only the first frame is I frame.)
MRG: 1, FEN: 1, ECU: 0/1, CFM: 0/1.

Table IV Performance of our proposed algorithm for HD sequences

HD Sequences		QP=22 TS	QP=27 TS	QP=32 TS	QP=37 TS	AVG. TS	Y BD rate
B	Kimono	42.0%	45.1%	45.0%	39.9%	43.0%	3.1%
	ParkScene	39.7%	36.4%	37.7%	36.8%	37.7%	2.2%
	Cactus	39.7%	39.7%	42.7%	41.0%	40.8%	2.1%
	BasketballDrive	38.7%	45.3%	47.7%	45.7%	44.4%	2.8%
	BQTerrace	43.1%	36.7%	36.9%	41.8%	39.6%	0.9%
E	Vidyo1	43.6%	48.3%	52.8%	51.9%	49.2%	1.8%
	Vidyo3	40.6%	45.1%	49.0%	46.3%	45.2%	2.6%
	Vidyo4	37.0%	43.1%	47.4%	46.4%	43.5%	2.4%
AVG.		40.5%	42.5%	44.9%	43.7%	**42.9%**	**2.2%**

Table V Performance of our proposed algorithm for other sequences

Other Sequences		QP=22 TS	QP=27 TS	QP=32 TS	QP=37 TS	AVG TS	Y BD rate
C	BasketballDrill	34.4%	33.4%	33.0%	31.1%	33.0%	3.4%
	BQMall	35.3%	31.6%	31.3%	29.6%	32.0%	2.7%
	PartyScene	36.1%	33.2%	30.5%	27.7%	31.9%	0.6%
	RaceHorses	34.3%	32.3%	31.0%	28.4%	31.5%	1.9%
D	BasketballPass	26.5%	25.2%	24.5%	23.7%	24.9%	1.3%
	BlowingBubbles	27.5%	23.4%	22.7%	23.5%	24.3%	0.9%
	BQSquare	28.5%	24.5%	22.1%	20.0%	23.8%	0.3%
	RaceHorses	27.3%	24.5%	23.1%	22.2%	24.3%	1.0%
AVG.		31.3%	28.5%	27.3%	25.8%	28.2%	1.5%

B. Comparsion with ECU and CFM

We simulate both ECU [4] and CFM [5] in HM5.0 with our experiment settings on 8 HD sequences (64 frames per sequence and GOP=1), and the results are listed in Table VI. The first three schemes in Table VI are (a) our proposed algorithm, (b) ECU and CFM, and (c) ECU, CFM, and our proposed algorithm altogether. Although the time savings ability of Scheme (a) overlaps with that of Scheme (b), the combined Scheme (c) can still provide additional savings over Scheme (b), 56% vs. 37%. The RD curve in Figure 5 implies that Scheme (c) has a higher coding loss at the low bitrate region. Therefore, we restrict the use of our proposed algorithm to the cases of QP=22 and 27 in Scheme (c). This is the so-called Scheme (d). The entries shown for Scheme (d) are the average of 8 HD sequences with 100 frames per sequence. Scheme (d) offers about 51% time reduction with the increment of about 2.02% BD rate, which is quite a bit better than Scheme (c).

VI. CONCLUSIONS

In this paper, we propose a fast algorithm based on the neighboring CU size information for the HEVC encoder. The algorithm saves encoding time quite efficiently at both low and high QPs, and it can be combined with the existing ECU [4] and CFM [5] schemes to achieve an overall higher time saving. Based on the multiple sequences test, we demonstrate the proposed algorithm can reduce 42.9% encoding time with 2.2% bitrate increment for HD sequences.

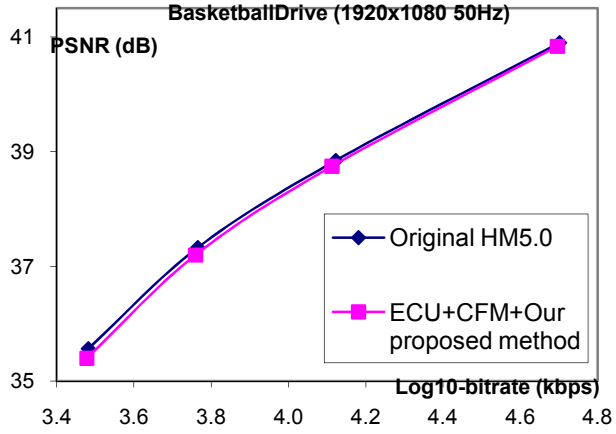

Figure 5. RD curve of scheme (c) in one worse case

Table VI Performance of different schemes for HD sequences

Scheme	QP=22 TS	QP=27 TS	QP=32 TS	QP=37 TS	AVG. TS	Y BD rate
(a)	40.18%	41.95%	44.15%	42.75%	42.26%	2.05%
(b)	15.63%	30.53%	44.48%	56.93%	36.89%	0.51%
(c)	46.37%	53.62%	60.62%	66.54%	56.79%	3.27%
(d)	46.86%	54.09%	44.73%	57.16%	**50.71%**	**2.02%**

VII. ACKNOWLEDGMENT

This work was supported in part by the NSC, Taiwan under Grants 98-2221-E-009-076-MY3 and by the Aim for the Top University Project of National Chiao Tung University, Taiwan.

REFERENCES

[1] T. Wiegand et al., "Special section on the joint call for proposals on High Efficiency Video Coding (HEVC) standardization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1661–1666, 2010.

[2] JCT-VC, "High Efficiency Video Coding (HEVC) Test Model 5 (HM 5) Encoder Description", JCT-VC document, JCTVC-G1102, January 2012.

[3] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," *IEEE Trans. Image Process.*, vol. 3, no. 3, pp. 327–331, May 1994.

[4] K. Choi et al., "Coding tree pruning based CU early termination," JCT-VC document, JCTVC-F092, Jul. 2011.

[5] R. H. Gweon et al., "Early termination of CU encoding to reduce HEVC complexity," JCT-VC document, JCTVC-F045, Jul. 2011.

[6] G. Bjontegaard, "Calculation of Average PSNR Differences between RD-curves," Document VCEG-M33, Apr. 2001.

[7] J. Leng et al., "Content based hierarchical fast coding unit decision algorithm for HEVC," *International Conference on Multimedia and Signal Processing*, pp. 56–59, 2011.

[8] G. Correa et al., "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Trans. on Consumer Electronics*, Vol. 57, No. 4, Nov. 2011.