Chapter 15
An Efficient Fast CU Depth and PU Mode Decision Algorithm for HEVC

Zong-Yi Chen, He-Yan Chen and Pao-Chi Chang

Abstract High Efficiency Video Coding (HEVC) is a new video coding standard, which improves the coding efficiency significantly. To achieve the best performance, HEVC encoder evaluates all possible candidates to determine the best depth of coding unit (CU) and mode of prediction unit (PU). This increases substantial computational complexity that might become an obstacle for practical applications. This paper proposes a fast algorithm for CU and PU to reduce the encoding time of HEVC. By referring spatial and temporal depth information of CU and motion/texture characteristics of PU, the proposed algorithm skips rarely used depths and modes in certain situations. The experimental results show that our proposed method averagely achieves 57 % time saving in high efficiency configuration and 61 % in low complexity configuration with negligible rate-distortion loss compared with the reference software.

Keywords HEVC · Mode decision · Fast algorithm

15.1 Introduction

With increasing popularity of high resolution video format, video coding technologies which can provide a substantially higher compression capability than the existing H.264/AVC [1] standard have received increased attention. In 2010, the Joint Collaborative Team on Video Coding (JCT-VC) jointed by ITU-T VCEG and ISO/IEC MPEG started to develop the next generation video coding standard called High Efficiency Video Coding (HEVC) [2]. In HEVC, the basic encoding unit which plays a similar role as a macroblock in H.264/AVC is separated into

Z.-Y. Chen · H.-Y. Chen · P.-C. Chang (✉)
Department of Communication Engineering, National Central University, Jhongli, Taiwan
e-mail: pcchang@ce.ncu.edu.tw

A. A. Farag et al. (eds.), *Proceedings of the 3rd International Conference on Multimedia Technology (ICMT 2013)*, Lecture Notes in Electrical Engineering 278,
DOI: 10.1007/978-3-642-41407-7_15,

coding unit (CU), prediction unit (PU), and transform unit (TU), as shown in Fig. 15.1. To achieve efficient and flexible compression of video content with various resolutions, the size of CU can be ranged from 64 × 64 to 8 × 8 and a quadtree-based block partition is adopted. Figure 15.2 shows the quadtree-structured CU and the size of CU is corresponding to its depth. Each CU contains one to four PUs and the size of PU is limited to that of CU. PU is the basic unit for prediction, and all information related to prediction, e.g., motion vector, is signaled on a PU basis. The partition type of a PU depends on the prediction mode as shown in Fig. 15.3.

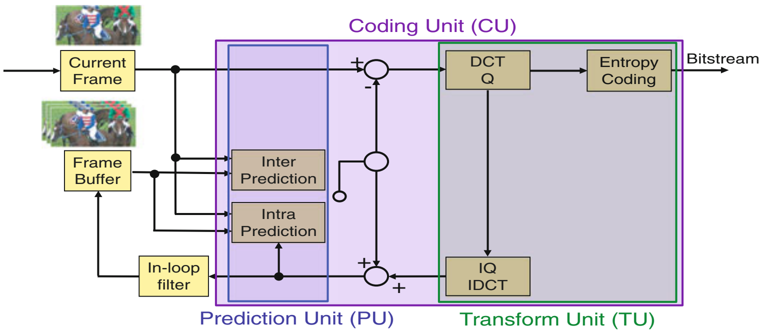


Fig. 15.1 Basic unit in HEVC

Fig. 15.2 The architecture of quad-tree-structured CU

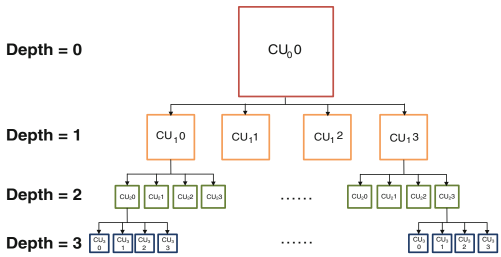
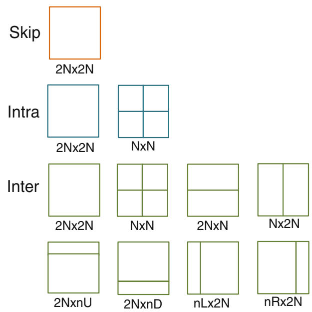


Fig. 15.3 Partition modes of a PU in HEVC



Generally, large size of CU is tend to be chosen as the best one in homogeneous regions to reduce the side information. On the other hand, small size of CU is selected to preserve the texture details in nonhomogeneous regions.

In the reference software HM5.0, the encoder has to test all CU depths recursively to find the best one, and all modes of PU must be further searched in each CU. This procedure leads to significant complexity increase in the encoder. Thus, it is critical to reduce the encoding complexity to make the real-time applications practicable. From our analyses, all encoding processes except in-loop filtering are computing on a CU basis which occupies about 95 % total encoding time in the encoder, and PU including motion estimation, motion compensation also occupies 57 % encoding time in CU. Therefore, we can efficiently reduce the complexity of encoder by decreasing the candidate depths of CU and modes of PU.

Several early termination methods [3, 4] have been proposed during HEVC standardization process. In [3], if the best prediction mode in current CU depth is Skip mode, the computation of subdepths will be terminated. In [4], if coded block flag (cbf) values of luma and two chromas are zero after encoding current PU mode, the remaining modes will be skipped. A fast CU decision algorithm which contains frame level and CU level is proposed in [5]. And some other fast encoding algorithms [6, 7] are proposed continuously. But the time savings of most of the previous works can be achieved only about 40 %. In this work, we provide an efficient method to skip the rarely used CU depths and PU modes under some conditions to reduce the encoding time by 60 % approximately.

The rest of this paper is organized as follows. Section 15.2 presents the proposed fast CU depth decision. Section 15.3 describes the fast PU mode decision algorithm. Experimental results are shown in Sect. 15.4 and the conclusion is remarked in Sect. 15.5.

15.2 Fast CU Depth Decision Algorithm

In the proposed CU depth decision algorithm, the temporal correlation of CUs is first considered. Two continuous frames in a video sequence are usually highly correlated. Therefore, the depth in previous frame can be utilized to predict the depth of current frame. Table 15.1 shows the average depth distribution between

Table 15.1 Depth distribution between current and co-located CUs

Co-located CU depth	Current CU depth				
	0 (%)	1 (%)	2 (%)	3 (%)	Total (%)
0	66	23	9	2	100
1	33	41	19	7	100
2	22	28	33	17	100
3	9	18	27	46	100

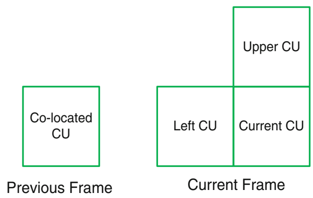


Fig. 15.4 CUs referred in the proposed algorithm

current and co-located CUs by running four sequences in different classes (B, C, D, and E) and five QPs (22, 27, 32, 37, and 42). The reference software that we adopted in our analyses is HM5.0.

We can observe that when co-located CU depth equals to zero or one, small CU size is rarely used in current CU. Similarly, large size is seldom used in current CU if co-located CU depth is three. This motivates us to skip the current CU depths when the conditional probability in Table 15.1 is below 10 % (the cases colored red with boldface). For example, when the co-located CU depth is 1, the search of depth = 3 will be skipped.

Not only object motion but details in video sequences will affect the depth determination of current CU. Thus, the spatial correlation of CUs is also taken into account for our algorithm. When an object is moving into a homogeneous area, the edge of object tends to be encoded with small CU size. Therefore, we refer to upper and left CU depths to determine whether a moving object is in the neighborhood of current CU. We propose to do full search on the current CU if any neighboring depth of CU is three. Figure 15.4 shows the CUs referred in our algorithm. And Fig. 15.5 is the flowchart of the proposed fast CU depth decision algorithm.

15.3 Fast PU Mode Decision Algorithm

In H.264, the encoder exists high probability to choose Skip mode as the best mode when the current coding mode is 16×16 and the following conditions are all satisfied [8]: (a) reference frame is previous one, (b) motion vector difference (MVD) is zero, and (c) transform coefficients are all quantized to zero. Similarly, we analyze CU depth and PU mode distribution under these three conditions in HEVC. The search of Inter $2N \times 2N$ mode in each depth is performed first. We observe that more than 92 % candidate PU modes will be selected as Skip mode when previous three conditions are met as shown in Table 15.2. Moreover, there is 99 % probability that the current CU depth will be the best one if PU satisfies these conditions as Table 15.3 shows. Hence we propose to select the best CU depth as current depth and the best PU mode as Skip mode once the conditions (a), (b), and (c) are all satisfied.

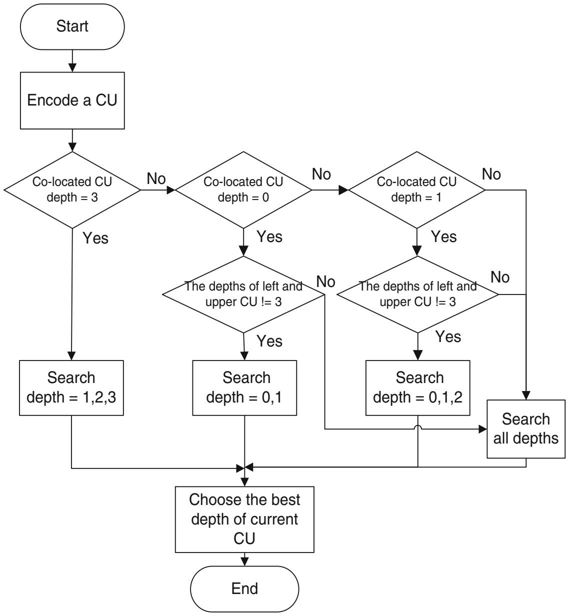


Fig. 15.5 Flowchart of the proposed fast CU depth decision algorithm

Table 15.2 PU mode distribution when (a), (b), and (c) are met

	Skip mode (%)	Inter mode (%)
ClassB_Cactus	93	7
ClassC_BQMall	91	9
ClassD_BQSquare	89	11
ClassE_vidyo3	96	4
Average	92	8

Table 15.3 CU depth distribution when (a), (b), and (c) are met

	Depth = k (%)	Depth = k + 1 (%)
ClassB_Cactus	99	1
ClassC_BQMall	99	1
ClassD_BQSquare	98	2
ClassE_vidyo3	100	0
Average	99	1

The encoder tends to choose Skip and Inter 2N $\times$ 2N modes in a stationary region. We utilize (a), (b), and (c) to represent the degree of motion and texture in a block. When (a) is satisfied, the current PU tends to be simple motion. And the motion characteristic of current PU is similar to neighboring PU if (b) is satisfied. Both conditions (a) and (b) combine with (c) can be used to represent a stationary region and over 90 % candidate PU modes will be selected as Skip and Inter $2N \times 2N$ modes under these conditions as shown in Tables 15.4 and 15.5. The probability of Skip mode is included in $2N \times 2N$ in Tables 15.4 and 15.5. Based on these observations, the proposed algorithm will only examine Skip and Inter $2N \times 2N$ modes if condition (a)(c) or (b)(c) is satisfied. Finally, the flowchart of the proposed fast PU mode decision combined with fast CU depth decision algorithm is shown in Fig. 15.6.

15.4 Experimental Results

The simulation environment is listed in Table 15.6. Other settings are set the same as the common test conditions [9] in standard meetings. A group of experiments are carried out on different sequences with five quantization parameters. Three values defined in (1), (2), and (3) are used to evaluate the performance. Two algorithms [3, 4] are compared with the proposed algorithm and all fast algorithms use the original HM as the anchor in the experiments. The results are shown in Tables 15.7 and 15.8 for high efficiency (HE) and low complexity (LC) configurations, respectively. The results of each class in Tables 15.7 and 15.8 are averages of five QPs and two sequences in the same class because of limitations of space.

$$\Delta\text{PSNR (dB)} = \text{PSNR}_{\text{proposed}} - \text{PSNR}_{\text{HM}} \tag{15.1}$$

$$\Delta\text{Bitrate (\%)} = \frac{\text{Bitrate}_{\text{proposed}} - \text{Bitrate}_{\text{HM}}}{\text{Bitrate}_{\text{HM}}} \times 100 \tag{15.2}$$

$$\Delta\text{EncTime (\%)} = \frac{\text{EncTime}_{\text{proposed}} - \text{EncTime}_{\text{HM}}}{\text{EncTime}_{\text{HM}}} \times 100 \tag{15.3}$$

From Tables 15.7 and 15.8, we can observe that the proposed fast CU depth decision algorithm achieves about 40 % time saving on average while maintaining very good RD performance. We can also observe that the proposed algorithm performs worse for videos with smaller resolutions (classC and classD). For a smaller resolution video, there exists more textures or objects in a fixed-size CU block than that in a larger resolution video, and large CUs are used less frequently in these classes. This is the reason why the time reduction of class C and class D are fewer than that of others. When comparing with ECU [3], the proposed fast CU algorithm can provide almost the same PSNR and time saving performance with just a little less rate reduction.

For the proposed fast PU mode decision algorithm, about 45 % time saving on average and little rate decrease with PSNR degradation is achieved. For videos with slow motion and texture such as class E, the proposed algorithm saves more encoding time due to more PUs satisfy our proposed conditions. And the proposed fast PU algorithm performs significantly better than CFM [4] in rate reduction and time saving with almost the same video quality.

Table 15.4 Inter mode distribution under (a) and (c)

	2N × 2N (%)	2N × N (%)	N × 2N (%)	N × N (%)	2N × nU (%)	2N × nD (%)	nL × 2N (%)	nR × 2N (%)
ClassB_Cactus	93	3	4	0	0	0	0	0
ClassC_BQMall	94	2	4	0	0	0	0	0
ClassD_BQSquare	89	4	7	0	0	0	0	0
ClassE_vidyo3	94	2	4	0	0	0	0	0
Average	92	3	5	0	0	0	0	0

Table 15.5 Intermode distribution under (b) and (c)

	2N × 2N (%)	2N × N (%)	N × 2N (%)	N × N (%)	2N × nU (%)	2N × nD (%)	nL × 2N (%)	nR × 2N (%)
ClassB_Cactus	97	1	2	0	0	0	0	0
ClassC_BQMall	96	1	3	0	0	0	0	0
ClassD_BQSquare	92	3	5	0	0	0	0	0
ClassE_vidyo3	96	1	3	0	0	0	0	0
Average	95	2	3	0	0	0	0	0

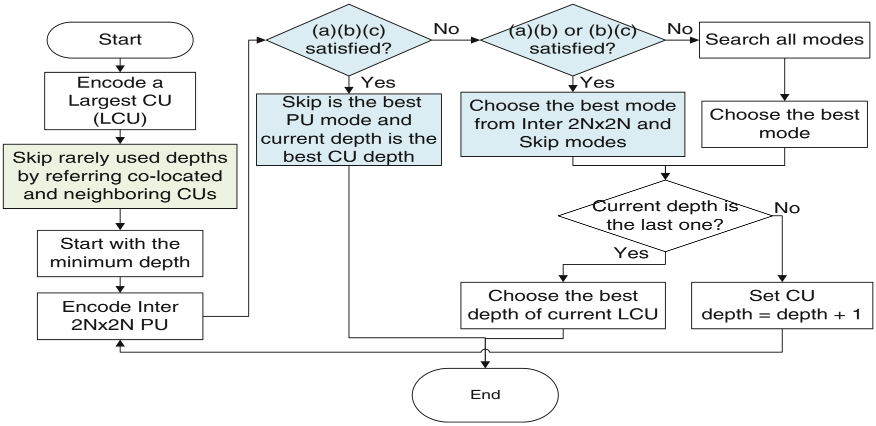


Fig. 15.6 Flowchart of the proposed overall algorithm

Table 15.6 Simulation environment

HM5.0		
Sequence	ClassB (1920 × 1080)	BasketballDrive, BQTerrace
	ClassC (832 × 480)	BasketballDrill, RaceHorses
	ClassD (416 × 240)	BlowingBubbles, BasketballPass
	ClassE (1280 × 720)	Vidyo1, Vidyo4
Configuration file	Low delay P (IPPP…)−High Efficiency (HE)	
	Low delay P (IPPP…)−Low Complexity (LC)	
MotionSearch	EPZS	
SearchRange	64	
FramesToBeEnclosed	100	
QP	22, 27, 32, 37, 42	
GOPSize	4	

Finally, the overall performance of the proposed algorithm is compared with the combination of ECU [3] and CFM [4]. The proposed algorithm achieves better PSNR and time saving performance than the reference algorithm (ECU + CFM) but with less rate reduction. We can observe that the proposed algorithm saves much more encoding time than the reference algorithm for ClassC (9.59 % in LC case). The two sequences in ClassC exist complex textures and objects such that CU depth is less early terminated at large CU stage in ECU and zero blocks occurs less in CFM. Compared with the original HM encoder, the average PSNR degradation of the proposed algorithm is no more than 0.12 dB and the rate reduction is up to 1.08 % on average. The encoding time can be saved up to 75 %. On average, 57 and 61 % encoding time reduction are achieved respectively for HE

Table 15.7 Results for high efficiency (HE) configuration

HE	△ PSNR (dB)	△ Bitrate (%)	△ EncTime (%)	△ PSNR (dB)	△ Bitrate (%)	△ EncTime (%)	△ PSNR (dB)	△ Bitrate (%)	△ EncTime (%)
	ECU [3]			CFM [4]			ECU + CFM		
ClassB	−0.03	−1.37	−42.59	−0.04	−0.57	−33.53	−0.11	−3.23	−55.72
ClassC	−0.03	−0.66	−24.23	−0.04	−0.16	−26.41	−0.12	−1.24	−39.95
ClassD	−0.04	−0.71	−30.12	−0.07	−0.27	−24.59	−0.17	−1.59	−44.58
ClassE	−0.04	−1.21	−60.97	−0.03	−0.18	−40.53	−0.14	−2.42	−72.62
Average	−0.03	−0.99	−39.48	−0.05	−0.29	−31.27	−0.13	−2.12	−53.22
	Proposed Fast CU			Proposed Fast PU			Proposed Overall		
ClassB	−0.03	−0.41	−43.71	−0.05	−1.39	−43.67	−0.08	−1.61	−59.35
ClassC	−0.03	0.31	−30.31	−0.05	−0.44	−31.70	−0.08	−0.09	−47.36
ClassD	−0.04	0.21	−28.67	−0.08	−0.65	−35.52	−0.12	−0.08	−48.71
ClassE	−0.03	−0.17	−52.02	−0.06	−0.99	−63.79	−0.10	−0.61	−70.70
Average	−0.03	−0.02	−38.68	−0.06	−0.87	−43.67	−0.10	−0.60	-56.53

Table 15.8 Results for low complexity (LC) configuration

LC	PSNR (dB)	△ Bitrate (%)	△ EncTime (%)	△ PSNR (dB)	△ Bitrate (%)	△ EncTime (%)	△ PSNR (dB)	△ Bitrate (%)	△ EncTime (%)
	ECU [3]			CFM [4]			ECU + CFM		
ClassB	−0.04	−1.89	−44.13	−0.04	−0.69	−33.75	−0.13	−3.74	−57.83
ClassC	−0.04	−0.81	−31.02	−0.05	−0.40	−31.11	−0.15	−1.90	−44.80
ClassD	−0.06	−0.88	−31.58	−0.07	−0.61	−30.44	−0.21	−2.16	−49.03
ClassE	−0.05	−1.70	−62.83	−0.03	−0.08	−42.68	−0.14	−2.88	−75.07
Average	−0.05	−1.32	−42.39	−0.05	−0.44	−34.49	−0.16	−2.67	−56.68
	Proposed Fast CU			Proposed Fast PU			Proposed Overall		
ClassB	−0.06	−0.97	−42.80	−0.05	−1.48	−44.13	−0.11	−2.42	−60.23
ClassC	−0.05	0.24	−35.95	−0.06	−0.64	−39.26	−0.10	−0.27	−54.39
ClassD	−0.07	0.23	−30.82	−0.09	−0.85	−38.48	−0.16	−0.44	−53.33
ClassE	−0.07	−0.31	−52.17	−0.06	−1.32	−67.27	−0.13	−1.18	−75.39
Average	−0.06	−0.20	−40.44	−0.07	−1.08	−47.29	−0.12	−1.08	−60.84

and LC cases. Although the rate-distortion performance of the proposed overall algorithm is slightly worse than the reference algorithm, higher encoding time can be saved.

15.5 Conclusion

In this paper, we propose a fast algorithm for HEVC to skip rarely used depths of CU and modes of PU in certain situations. In CU, the depth information of co-located block is employed to reduce the depth candidates and the upper and left blocks are referred to determine whether to search all depths or not. In PU, the combination of motion and texture characteristics including reference previous one, zero block, and zero MVD are utilized. On average, approximately 60 % reduction in encoding time compared to HM5.0 encoder can be yielded with only negligible rate-distortion losses.

References

1. H.264: International Telecommunication Union (2003) Recommendation ITU-T H.264: advanced video coding for generic audiovisual services. ITU-T
2. JCT-VC (2012) High efficiency video coding (HEVC) test model 8 encoder description. JCTVC-J1002
3. JCT-VC (2011) Coding tree pruning based CU early termination. JCTVC-F092. 6th JCT-VC meeting, Torino
4. JCT-VC (2011) Early termination of CU encoding to reduce HEVC complexity. JCTVC-F045. 6th JCT-VC meeting, Torino
5. Leng J, Sun L, Ikenaga T, Sakaida S (2011) Content based hierarchical fast coding unit decision algorithm for HEVC. IEEE CMSP 1:56–59
6. Tan LH, Liu F, Tan YH, Yeo C (2012) On fast coding tree block and mode decision for high-efficiency video coding (HEVC). IEEE ICASSP 825–828
7. 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
8. Lee J, Jeon B (2004) Fast mode decision for H.264. IEEE ICME 2:1131–1134
9. JCT-VC (2011) Common HM test conditions and software reference configurations. JCTVC-G1200. 7th JCTVC meeting. Geneva, CH