Fast Algorithm of Coding Unit Depth Decision for HEVC Intra Coding

Xiaofeng Huang, Huizhu Jia*, Kaijin Wei, Jie Liu, Chuang Zhu, Zhengguang Lv, Don Xie

National Engineering Laboratory for Video Technology, Peking University, Beijing, China

{xfhuang, hzjia, kjwei, liuzimin, czhu, zglv, xdxie}@jdl.ac.cn

***Abstract*— The emerging high efficiency video coding standard (HEVC) achieves significantly better coding efficiency than all existing video coding standards. The quad tree structured coding unit (CU) is adopted in HEVC to improve the compression efficiency, but this causes a very high computational complexity because it exhausts all the combinations of the prediction unit (PU) and transform unit (TU) in every CU attempt. In order to alleviate the computational burden in HEVC intra coding, a fast CU depth decision algorithm is proposed in this paper. The CU texture complexity and the correlation between the current CU and neighbouring CUs are adaptively taken into consideration for the decision of the CU split and the CU depth search range. Experimental results show that the proposed scheme provides 39.3% encoder time savings on average compared to the default encoding scheme in HM-RExt-13.0 with only 0.6% BDBR penalty in coding performance.**

***Index Terms*— Fast CU depth decision, Neighbouring CUs, Texture complexity, QP, Weighted average.**

I. INTRODUCTION

In recent years, the demands for higher video resolution and better visual quality are increasing rapidly. The existing H.264/AVC standard may not satisfy the compression requirement of the increasing demands any more. To improve the compression efficiency, the ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) formed the Joint Collaborative Team on Video Coding (JCT-VC) and developed the new HEVC standard. In comparison to H.264/AVC, HEVC achieves approximately 50% bit-rate reduction for equal perceptual quality [1][2].

As in all previous video coding standards, HEVC also follows the classic block-based hybrid coding structure. The highly flexible hierarchy of unit representation for CU, PU and TU is introduced in HEVC to significantly improve the coding performance. CU is the basic coding processing unit and allows recursive splitting into four equally sized sub-CUs until reaching the smallest CU (SCU) of a size 8×8. HEVC supports various CU sizes ranging from the largest CU (LCU) size of 64×64 to the SCU. In CU quad-tree structure, the CU splitting and pruning process are performed to get the optimal coding tree unit (CTU) partition [3]. PU is the basic unit for prediction and its size is limited to that of CU. For intra prediction, 5 types of PU sizes, which are 64×64, 32×32, 16×16, 8×8, and 4×4, are supported. Besides CU and PU, TU is the basic unit for transform and quantization. The TU size cannot exceed that of CU and the residual quad-tree (RQT) coding further improves the coding performance [4]. In the CU splitting process, HM encoder needs to exhaust all the combinations of PU and TU types in order to get the minimal CU cost.

Recently, a number of fast algorithms [5-8] have been proposed to optimize the CU depth decision for the HEVC intra encoder. Typically, these fast CU depth decision algorithms can be classified into two categories. In the first category, works like [5][6] determine the CU depth search range before the recursive CU splitting and pruning process. The straightforward elimination of low-probability depth levels alleviates the coding complexity significantly. In [5], the correlation between the current CTU and neighbouring CTUs is fully utilized to determine the CTU depth search range. It is a rough CU depth decision method because of the depth search range decision at the CTU level and only the elimination of the least gain depth level 0 (64×64) for the most occurring prediction depth type “III” and “IV”. Work [6] uses the variance which reflects the image texture complexity as the metric for CU depth decision. The fixed threshold and the simple CU decision process limit its application to various image contents. In the second category, papers like [7][8] skip certain CUs when predefined conditions are satisfied during the fully recursive CU splitting and pruning process. Paper [7] skips the large CU based on the block structures of its sub-CUs and paper [8] skips the further splitting process based on the RD cost correlation between the parent CU and its partial sub-CUs. However, the indeterminacy of the CU depth search range limits its application to hardware implementation.

In this paper, we will propose a fast CU depth decision algorithm which belongs to the first category. Besides the texture complexity, the correlation between the current CU and neighbouring CUs which can reflect both the impact of the parameter QP and the spatial correlation is adaptively taken into consideration for the fast CU depth decision algorithm. The proposed CU-level depth decision is more adaptive and accurate than CTU-level depth decision proposed in [5]. In this paper, we simply regard 4×4 PU as a CU level, implying that there are 5 depth levels which are 64×64 (depth level 0), 32×32 (depth level 1), 16×16 (depth level 2), 8×8 (depth level 3), and 4×4 (depth level 4), respectively. The remainder of this paper is organized as follows. Section II presents our proposed fast CU depth decision algorithm. Experimental results are shown in Section III. Overall conclusions are given in section IV.

978-1-4799-6139-9/14/$31.00 ©2014 IEEE

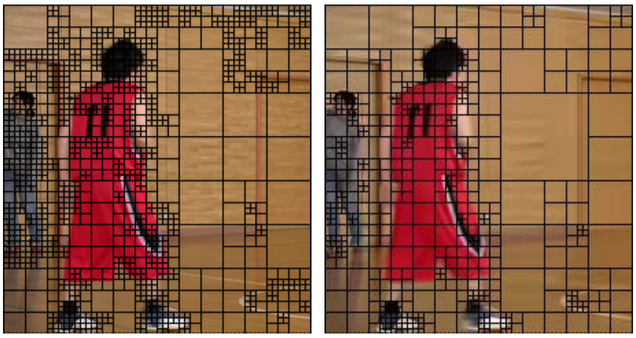
II. Proposed Fast CU Depth Decision Algorithm

This section presents a fast CU depth decision algorithm in detail, which includes the analysis based on observations and the algorithm procedure. The observations and the analysis instruct us to devise a novel and reliable fast CU depth decision algorithm.

A. *Observations and Analysis*

In the HEVC intra encoder, the final CU depth has a strong correlation with the texture complexity of the input image, which can be illustrated as shown in Fig. 1. In Fig. 1, large CUs are preferred for the flat region, while small CUs are chosen for the complex region in order to minimize the prediction error. Thus the variance which reflects the texture complexity can be used as a main factor for the fast CU depth decision. The calculation of variance value is shown as (1), where N represents the size of the CU, and $Y(i,j)$ is the luminance value of the pixel at (i,j) location.

$$Var = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} Y(i,j)^2 - \left(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} Y(i,j)\right)^2 \tag{1}$$



(a) 2nd frame (QP 22) (b) 2nd frame (QP 37)

Fig. 1 CU depth structure of BasketballPass with different QPs (black square represents CU)

Although the variance-based fast CU depth decision algorithm proves to be reasonable and effective, it should be further explored because parameters such as QP affect the final CU depth structure greatly. As shown in Fig. 1, large CUs instead of small CUs are finally selected in the same region with the increase of QPs. This observation inspires us to take the parameter QP into consideration for the fast CU depth decision algorithm. A simple and straightforward method is to use a threshold such as in [6] which is a function of QP. However, it is quite difficult to use such an approach because of the inaccurate mathematical modelling of the diverse input video scenes. In order to resolve this problem, the already-determined neighbouring CU depth levels are fully utilized in our proposed fast CU depth decision algorithm. The neighbouring CU depths are decided by the CU splitting and pruning process which is correlated with the parameter QP. Obviously, the spatial neighbouring correlation will also be considered if the neighbouring CUs are taken into account [9]. As illustrated above, the CU texture complexity and the neighbouring CU depth levels are intentionally taken into consideration for the proposed fast CU depth decision algorithm. The texture complexity, the impact of the parameter QP and the spatial correlation are fully utilized in our proposed algorithm. The fast CU depth decision algorithm is detailed in the next section.

B. *Fast CU Depth Decision Algorithm*

The overall procedure of the proposed fast CU depth decision algorithm is described in Fig. 2. In Fig. 2, the texture complexity level value and the predicted depth level value are calculated for the current CU, respectively. The split level value is calculated by the weighted average of these two values. Then, the split decision is judged according to the split level value. If split, the current CU depth is excluded and further sub-CU depth decision is executed; if not split, the CU depth search range is decided based on the split level value and the current CU depth level. The CU splitting and pruning process of the current CU and its sub-CUs in the CU search depth range is performed to get the optimal CU depth structure. Finally, if the current CTU is not finished, the next z-scan order CU is obtained to operate as illustrated above. The texture complexity level calculation, the predicted depth level calculation, the split level calculation and the CU depth search range decision process are illustrated below.

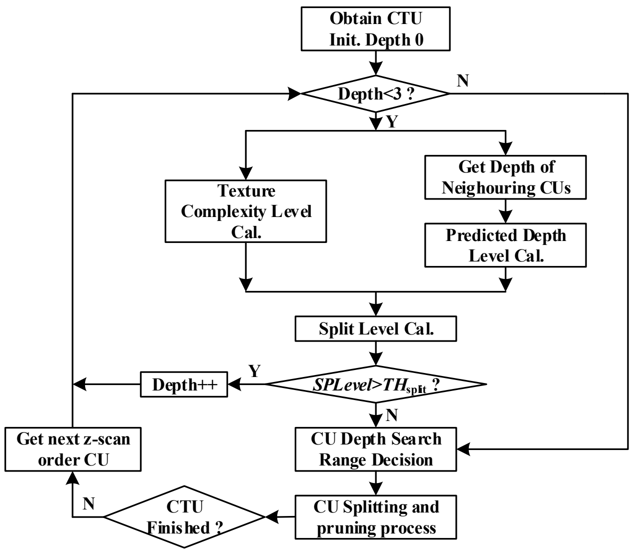


Fig. 2 The overall procedure of the fast CU depth decision algorithm

1) *Texture Complexity Level Calculation.* As in [6], the CU variance value is utilized to calculate the texture complexity level. In this paper, we quantize the variance value into five texture complexity levels, which are 0, 2, 4, 6, and 8, respectively. The texture complexity level 0 corresponds to the extremely flat region and the texture complexity level 8 corresponds to the extremely complex region. The texture complexity levels 2, 4, 6 represent the intermediate states which reflect the transition from the flat region to the complex region. The calculation of texture complexity level (*TCLevel*) is shown in (2).

$$TCLevel = \begin{cases} 0 & Var < TH_0 \\ 2 & Var < TH_1 \\ 4 & Var < TH_2 \\ 6 & Var < TH_3 \\ 8 & else \end{cases} \quad (2)$$

The four threshold values of TH_0, TH_1, TH_2, and TH_3 in (2) are in the increasing order and are derived based on the statistics as in [6].

2) *Predicted Depth Level Calculation.* In Fig. 2, the calculation of the predicted depth level is based on the neighbouring CU depth levels. As shown in Fig. 3, the depth levels of the LEFT, ABOVE-LEFT, ABOVE and ABOVE-RIGHT CUs in blue colour are utilized for the predicted depth level calculation. The chosen depth levels of the LEFT and ABOVE CUs are the maximum depth levels of all the neighbouring CUs. The depth levels of the ABOVE-LEFT and ABOVE-RIGHT CUs are the depth levels of the neighbouring CUs. This is more adaptive and accurate than [5], which only utilizes the maximum depth levels of the neighbouring CTUs.

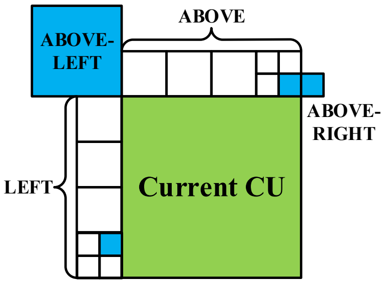


Fig. 3 Neighbouring CUs utilized for the predicted depth level calculation

TABLE I
WEIGHTING FACTORS OF NEIGHBOURING CUS

	LEFT	ABOVE	ABOVE-LEFT	ABOVE-RIGHT
β_i	0.3	0.3	0.2	0.2

The predicted depth level (*PDLevel*) is calculated by the weighted average of the four neighbouring CU depth levels. The formula is shown as (3).

$$PDLevel = \sum_{i=1}^{4} (NeiD_i - CurD + 4) \bullet \beta_i \quad (3)$$

In (3), the $NeiD_i$ represents the depth levels of the neighbouring CUs which are LEFT, ABOVE, ABOVE-LEFT, and ABOVE-RIGHT CUs, respectively. The *CurD* is the depth level of the current CU. The constant value 4 is used to make the value ($NeiD_i$-$CurD$) into a positive number, so the value of ($NeiD_i$-$CurD$+4) is ranging from 0 to 8. The β_i are the weighting factors of the four neighbouring CUs and defined in Table I according to the correlation between the current CU and its neighbouring CUs. It is obvious that the LEFT and ABOVE CUs impose a greater difference than the ABOVE-LEFT and ABOVE-RIGHT CUs. The *PDLevel* value is ranging from 0 to 8, which has the same data range as *TCLevel* value.

3) *Split Level Calculation.* The split level (*SplitLevel*) is calculated by the weighted average of the *TCLevel* and *PDLevel* value. The formula is shown as (4). The α value is the weighting factor, which is assigned to 0.4 because the *TCLevel* value only considers the texture complexity and the *PDLevel* value takes both the impact of the parameter QP and the spatial correlation into account.

$$SplitLevel = \alpha \times TCLevel + (1-\alpha) \times PDLevel \quad (4)$$

The split decision is based on the *SplitLevel* value. If the *SplitLevel* is greater than a threshold TH_{split}, the split is executed and the current CU depth level will be excluded in the subsequent CU splitting and pruning process. The TH_{split} value is correlated with the current CU depth level, and is assigned to 4 for the CU depth level 0 and assigned to 5 for the CU depth level 1 and 2 because of the increasing occurring probability of the CU depth levels.

4) *CU depth search range decision.* The CU split decision process excludes the low-probability CU depth levels, which alleviates the computational complexity. However, the CU depth search range should be determined before the CU splitting and pruning process.

$$SearchRange = \begin{cases} 2 & SplitLevel < TH_{range} \\ 3 & SplitLevel <= TH_{split} \end{cases} \quad (5)$$

As shown in (5), the CU depth search range is determined according to the *SplitLevel* value. If the *SplitLevel* value is smaller than the TH_{range}, the minimum CU depth level and the maximum depth level in the CU splitting and pruning process are assigned to the current CU depth level (*CurD*) and the (*CurD*+1), respectively. This implies that only two CU depth levels are recursively searched. The threshold value TH_{range} is assigned to 2, which represents that the current CU is flat and trends to be a large CU. If the *SplitLevel* value is in the range between TH_{range} and TH_{split}, the minimum CU depth level and the maximum CU depth level are assigned to (*CurD*) and (*CurD*+2), respectively. As shown in Fig. 2, when the current CU depth level is 3, the current CU depth level (3) and the depth level (4) are assigned straightforwardly.

This section illustrates the proposed algorithm in detail. Our algorithm starts from the observations, which inspires us to take the texture complexity, the impact of the parameter QP and the spatial correlation into consideration. Firstly, the procedure calculates the texture complexity level (*TCLevel*) and the predicted depth level (*PDLevel*), respectively. Secondly, the split decision is based on the *SplitLevel* value which is derived by the weighted average of the *TCLevel* value and the *PDLevel* value. Finally, the CU depth search range is determined based on the *SplitLevel* value. The minimum CU depth level and maximum CU depth level in the CU splitting and pruning process are assigned based on the CU depth search range decision. The CU splitting and pruning process are performed to get the optimal CU depth structure. Experimental results will be shown in Section III.

III. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed fast CU depth decision algorithm, the proposed algorithm is implemented on the HEVC test model HM-RExt-13.0. The configuration is “All Intra Main” which is one of the common test conditions. The proposed algorithm is evaluated with QPs 22, 27, 32 and 37 and using sequences recommended by the JCT-VC. For comparison, we also implemented the fast CU depth decision algorithm proposed in papers [5][6] on the test. The performance of the proposed algorithm is measured by the BDBR defined in [10] and the complexity reduction (ΔT) defined in (6).

$$\Delta T = \frac{Time_{HMRExt13.0} - Time_{proposed}}{Time_{HMERExt13.0}} \times 100\% \qquad (6)$$

The performance comparisons of the proposed algorithm and the algorithm in [5][6] are shown in Table II. As shown in Table II, the proposed algorithm achieves about 39.3% complexity reduction with 0.6% BDBR penalty compared to the HM-RExt-13.0.

TABLE II
PERFORMANCE COMPARISONS

Class	Sequence	Proposed		Algorithm in [5]		Algorithm in [6]	
		ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)
A	Traffic	43.2	0.7	14.4	0.1	58.9	4.7
	PeopleOnstreet	37.5	0.7	14.2	0.1	54.3	3.5
B	BasketballDrive	46.8	0.6	23.6	0.6	67.1	7.0
	BQTerrace	42.3	0.5	20.3	0.5	55.1	2.4
	Cactus	42.9	0.8	15.9	0.1	57.5	4.6
	Kimono	50.9	1.2	49.2	0.5	**66.8**	**17.1**
	ParkScene	43.8	0.8	16.2	0.1	56.8	3.8
C	BasketballDrill	36.1	0.6	10.8	0.0	59.7	5.4
	BQMall	33.9	0.8	11.8	0.1	48.2	3.8
	PartyScene	31.3	0.4	10.9	0.0	40.4	0.6
	RaceHorses	38.9	0.5	11.7	0.0	51.0	2.5
D	BasketballPass	33.3	0.2	9.7	0.0	49.5	2.6
	BlowingBubbles	24.9	0.1	9.9	0.1	37.6	1.0
	BQSquare	30.1	0.0	9.8	0.0	37.9	0.4
	RaceHorses	26.8	0.2	9.2	0.0	37.9	1.6
E	FourPeople	38.7	0.8	16.0	0.2	59.1	4.7
	Johnny	47.1	0.8	29.5	0.7	70.7	7.4
	KristenAndSara	44.5	0.7	26.5	1.1	67.5	5.7
F	ChinaSpeed	41.3	0.6	18.8	3.8	61.3	3.0
	SlideEditing	36.3	0.9	13.0	0.0	48.4	2.1
	SlideShow	55.2	0.9	**44.4**	**13.4**	74.2	4.6
Average		**39.3**	**0.6**	**18.4**	**1.02**	**55.2**	**4.2**

Compared to the algorithm in [5], our proposed algorithm shows much more reliability and much more complexity reduction for all sequences. The algorithm proposed in [5] shows a singular point for the artificial sequence SlideShow. This is because that the spatial correlation algorithm proposed in [5] no longer satisfies the feature of the SlideShow sequence. The complexity reduction of our algorithm is by adaptively eliminating more CU depth levels, while work [5] only eliminates the least gain depth level 0 (64×64) for the most occurring prediction depth type “III” and “IV”.

Compared to the algorithm in [6], our proposed algorithm shows much more reliability and less BDBR increment for all sequences. The so much BDBR increment in [6] is because that the final CU depth structure is predetermined by the simple comparison of the CU variance and the fixed threshold value. And paper [6] shows a singular point for Kimono sequence, this is because that the threshold value is not appropriate for this sequence. Our proposed algorithm takes the texture complexity, the impact of the parameter QP and spatial correlation into consideration, and shows being both reliable and efficient for all sequences.

IV. CONCLUSION

This paper presents a fast CU depth decision algorithm to alleviate the computational complexity for the HEVC intra encoder. The proposed algorithm is inspired by the observations. Besides the texture complexity, the correlation between the current CU and neighbouring CUs which reflects both the impact of the parameter QP and spatial correlation are adaptively taken into consideration for the algorithm. The proposed fast CU depth decision algorithm proves to be reliable and efficient for various input sequences. Experimental results show that our proposed algorithm achieves about 39.3% complexity reduction with only 0.6% BDBR increment.

ACKNOWLEDGMENT

This work is partially supported by grants from the Chinese National Natural Science Foundation under contract No.61171139, and National High Technology Research and Development Program of China (863 Program) under contract No.2012AA011703.

REFERENCES

[1] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, “Comparison of the coding efficiency of video coding standards-Including High Efficiency Video Coding (HEVC),” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1668-1683, Dec. 2012.

[2] 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. 1649-1668, Dec. 2012.

[3] Seunghyun Cho and Munchurl Kim, “Fast CU Splitting and Pruning for Suboptimal CU Partitioning in HEVC Intra Coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 9, pp. 1555-1564, Sept. 2013.

[4] Tung Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand, “Transform Coding Techniques in HEVC,” *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 978-989, Dec. 2013.

[5] Liquan Shen, Zhaoyang Zhang, and Ping An, “Fast CU size decision and mode decision algorithm for HEVC intra coding,” *IEEE Trans. Consumer Electronics*, vol. 59, no. 1, pp. 207-213, Feb. 2013.

[6] T. Nishikori, T. Nakamura, T. Yoshitome, and K. Mishiba, “A Fast CU Decision Using Image Variance in HEVC Intra Coding,” in *Proc. of IEEE ISIEA*, pp. 52-56, Sept. 2013.

[7] Han Huang, Yao Zhao, Chunyu Lin, and Huihui Bai, “FAST BOTTOM-UP PRUNING FOR HEVC INTRAFRAME CODING,” in *Proc. of Visual Communications and Image Processing (VCIP)*, pp. 1-5, Nov. 2013.

[8] Hao Zhang and Zhan Ma, “Fast Intra Mode Decision for High Efficiency Video Coding (HEVC),” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 4, pp. 660-668, April 2014.

[9] Yongfei Zhang, Haibo Wang, and Zhe Li, “Fast Coding Unit Depth Decision Algorithm for Interframe Coding in HEVC,” in *Proc. of Data Compression Conference (DCC)*, pp. 53-62, March 2013.

[10] G. Bjontegaard, "Calculation of average PSNR difference between RD-curves," *13th VCEG-M33 Meeting*, Austin, TX, Apr. 2-4, 2001.