Early Termination of Coding Unit Splitting for HEVC

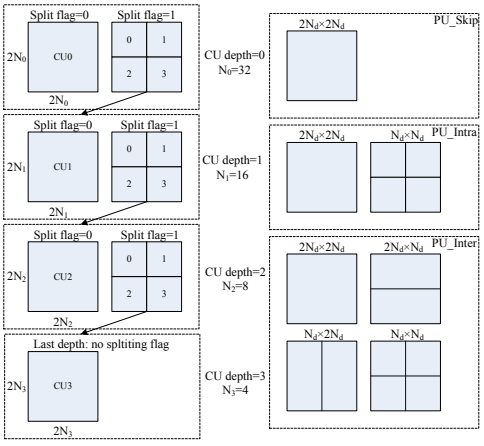
Qin Yu*, Xinfeng Zhang†, Shiqi Wang* and Siwei Ma*
*Peking University, Beijing, China
E-mail: qyu@pku.edu.cn, swang@jdl.ac.cn, swma@pku.edu.cn Tel: +86-10-62753424
†Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
E-mail: xfzhang@jdl.ac.cn Tel: +86-10-62753424

***Abstract*—The emerging high-efficiency video coding (HEVC) standard employs a new coding structure characterized by coding unit (CU), prediction unit (PU) and transform unit (TU). It improves the coding efficiency significantly, but also introduces great computation complexity on the decision of optimal CU, PU and TU sizes. To reduce the encoding complexity, we propose a CU splitting early termination scheme for inter frame coding. In the proposed scheme, the characteristics of prediction residuals are utilized to early terminate the CU splitting. Specifically, the Mean Square Error (MSE) between the prediction block and the origin block for each CU level is obtained and then compared with an adaptive threshold. The recursive CU splitting process is early terminated according to the threshold. Experimental results demonstrate that, the proposed algorithm achieves up to 34.83% total encoding time reduction with less than 0.25% BD-rate increase on average.**

I. INTRODUCTION

In the upcoming HEVC standard, more and more flexible coding tools and strategies are employed to improve the coding performance of hybrid coding structure. One of the most important coding tools in HEVC is the adoption of coding unit (CU), prediction unit (PU) and transform unit (TU) [1]. CU, the basic coding unit similar to macroblock in H.264/AVC, can have various sizes and allows recursive quad-tree splitting. Given the size of Largest Coding Unit (LCU) and the maximum hierarchical depth, CU can be expressed in a recursive quad-tree representation adapted to the picture content. Fig. 1 (a) shows maximum possible recursive CU structure in HEVC test model (HM). In HM, the tree-structured CU is limited from 8×8 to 64×64 for luma. Once the splitting of CU hierarchical tree is finished, the leaf node CUs can be further split into PUs. PU is the basic unit for prediction and it allows multiple different shapes to encode irregular image patterns as shown in Fig. 1 (b). PU size is limited to that of CU with square or rectangular shape. Besides CU and PU, TU is also defined to represent the basic unit for transform coding and quantization. The size of TU cannot exceed that of CU, but it is independent with PU size for inter coding.

The block structure of three kinds of processing units, i.e. CU, PU and TU, allows each to be coded optimally [1], which significantly improves the coding efficiency of HEVC. However, the exhaustive rate distortion cost calculation for each combination of three units brings great computation complexity to the encoder. To reduce the complexity, many fast inter mode decision algorithms have been proposed. In H.264/AVC, the spatial homogeneity and the temporal stationarity are utilized to reduce the candidate inter modes [2]. In [3, 4], motion homogeneity evaluated on a normalized motion vector field of 4×4 block is checked to find the possible optimal partition size. But this method is not suitable for HEVC because the most preferred CU splitting is identified recursively by comparing the RD cost in a quad-tree structure from LCU down to the smallest CU. In [5], I. Choi et al target on early termination of skip and selective intra mode to avoid unnecessary mode detections. However, all these algorithms are based on the macroblock level, which cannot be applied to HEVC directly. For HEVC, coded block flag (*cbf*) is used to terminate PU encoding process in [6]. If the *cbf* of an Inter PU in a CU is zero for luma and chroma except for Inter NxN PU, the next PU encoding process for the current CU is terminated. Another skip mode early termination algorithm is proposed in [7]. If skip mode is the locally optimal mode of the current CU depth, it is considered to be global optimal mode and sub-tree computation process can be skipped. Although prior methods can reduce the encoding complexity to some extent, we are still able to reduce the complexity further by considering the special block structure in HEVC.



(a) (b)

Fig. 1 (a) Maximum possible recursive CU structure in HM. (LCU size= 64, maximum hierarchical depth = 4), (b) Possible PU splitting for skip, intra and inter in HM.

In this paper, we propose a new fast algorithm for CU splitting decision. Firstly, we analyze the residuals with the mean square error (MSE) of Inter 2N×2N prediction residual block at different CU depth. Then the correlation between the MSE and optimal CU splitting is investigated. According to the correlation analysis, we propose a novel early termination scheme for CU splitting. With this scheme, the unnecessary CU splitting process can be skipped in advance.

The remainder of this paper is organized as follows. Section II describes the proposed fast coding unit decision algorithm. Experimental results and analysis are shown in Section III. Finally, we conclude our paper in Section IV.

II. Proposed Early CU Termination Algorithm

In this section, a fast coding unit decision algorithm for inter frame coding is described, including experimental observations and detailed algorithm description. We start with motivating observations, which provide useful guidelines for modeling the correlation between CU splitting and the MSE of prediction residuals in the current CU level. Then, the correlation is explored to accelerate the CU splitting termination process in inter frame coding. Finally, the framework of proposed fast coding unit decision scheme is presented.

A. Motivating Observations

When coding a frame, prediction residuals can reflect the prediction accuracy. For temporally stationary and spatially homogeneous blocks, prediction residuals are relatively small, and large CU is more likely to be chosen as the optimal CU size. Because further splitting into smaller size brings no much prediction improvement but increases side information. While smaller CU partition is preferred for objects with flexible motion since the CU can be hardly predicted accurately for such cases and large prediction residuals need to be coded. Our experiments have verified this. Fig. 2 shows the prediction residuals (10 times enlarge of the original residuals) of Inter 2N×2N (N=32) luma obtained from PartyScene (832x480) and the corresponding optimal CU partition. It can be observed that, blocks with small residuals prefer to choose large CU size, while those with large residuals prefer small CU size. Therefore, if the current CU size is sufficient for accurate prediction, there is no need for further splitting. On the contrary, if the residuals of current CU size are large, further split might be necessary to get more precise prediction. Based on the above observations, we propose to utilize MSE of the prediction residual block to terminate the CU recursive splitting process in advance.

In order to explore the correlation between residuals and optimal CU size, we compress first 8 frames of the sequence PartyScene(832x480) by HM 6.0 with LCU equal to 64 and hierarchical depth equal to 4. Fig. 3 shows the MSE of Inter 2N×2N mode obtained from different CU depth. Fig. (a)~(c) are coding results of different depth, and MSE of different CU number are shown. The blue asterisk represents that the current CU size is optimal and isn't split further, while the red represents that the current CU were split to achieve its optimal size. We can also see that, the number of red asterisks are less than that of the blue ones. That's to say, when CU size is equal to 64x64, more CUs are not split than split. Among those red asterisks who are split into four 32x32, more CUs are not split than split in the next CU depth. From Fig. 3, it can be seen that when the MSE of the current CU is small, most CUs needn't be split. Therefore, it's reasonable to conclude that when the MSE of current CU is smaller than a threshold (MSE_{thres}), the quad-tree partition process could be terminated.

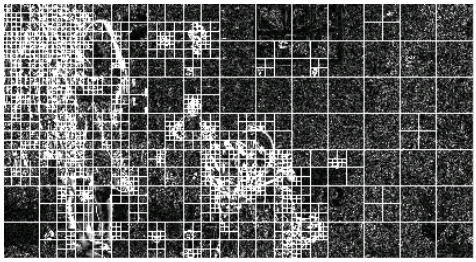
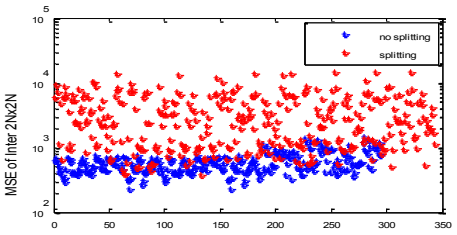
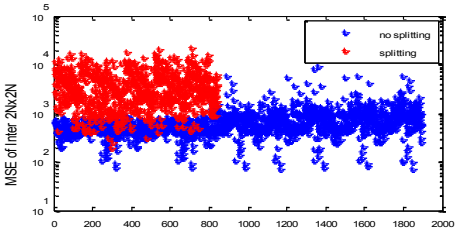


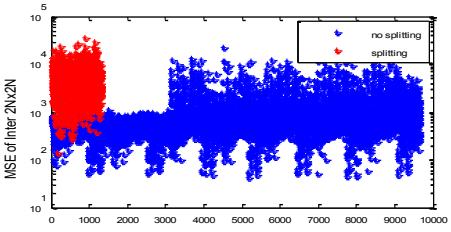
Fig. 2 Luma residuals of Inter 2N×2N (N=32) obtained from PatyScene_832x480_50 and corresponding optimal CU partition.



(a) Depth=0, N=32



(b) Depth=1, N=16



(c) Depth=2, N=8

Fig. 3. MSE of Inter 2N×2N obtained from different CU depth.

Generally, the mean MSE of the no splitting CUs' (blue dots) can be defined as a threshold for classification, but it is not appropriate to be applied here. As shown in Fig. 3, a few CUs that didn't split have extremely large MSE. These singular points may significantly enlarge the mean MSE and influence the

accuracy of the early termination. Taking this point into account, to get a robust threshold value, we employ the *K*-th smallest MSE of no splitting CUs as the threshold. *K* is defined as:

$$K=round(k\times NUM_{nsi}) \quad (1)$$

where NUM_{nsi} represents the number of CU in the *i*th depth that need no splitting and *k* is a adjustable factor between 0 and 1. By adjusting *k*, we can change the correctness and error probability of the classification, which plays an important role in balancing encoding complexity reduction and coding efficiency.

TABLE I.
CORRECTNESS AND ERROR PROBABILITY UNDER DIFFERENT THRESHOLDS

Sequence	*k*	0.5		0.6		0.7	
	QP	*P(R)* (%)	*P(W)* (%)	*P(R)* (%)	*P(W)* (%)	*P(R)* (%)	*P(W)* (%)
Class B 1080p	22	33.2	4.0	36.8	5.2	42.9	7.4
	27	23.4	1.4	39.1	3.5	45.8	5.3
	32	26.2	1.7	41.5	3.9	48.8	5.6
	37	28.3	2.3	43.4	4.8	50.9	6.5
Class C WVGA	22	22.7	3.8	24.3	4.8	28.1	7.1
	27	15.3	1.7	27.6	5.3	31.7	7.0
	32	17.6	2.6	30.7	6.6	35.6	8.7
	37	21.5	4.5	32.8	8.4	38.9	10.8
Class D WQVGA	22	20.7	0.8	23.7	1.5	29.2	2.5
	27	6.3	0.3	23.6	2.3	30.7	3.9
	32	13.6	0.8	28.4	4.6	32.4	6.9
	37	13.8	2.7	31.7	9.3	37.4	12.7
Class E 720p	22	35.1	2.9	41.2	4.9	47.0	7.3
	27	31.7	1.5	44.7	3.9	50.6	5.4
	32	35.3	2.4	45.6	4.4	53.5	5.8
	37	37.4	2.8	48.1	5.0	56.3	6.3

To verify the effectiveness of threshold, statistic probabilities of correct and wrong classifications are introduced. The correct and wrong classification probability are denoted as *P(R)* and *P(W)* defined as:

$$P(R)= NUM_{(MSE<MSE_{thres})\&\&ns}/NUM_{CU}\times 100\% \quad (2)$$

$$P(W)= NUM_{(MSE<MSE_{thres})\&\&s}/NUM_{CU}\times 100\% \quad (3)$$

where $NUM_{(MSE<MSE_{thres})\&\&ns}$ and $NUM_{(MSE<MSE_{thres})\&\&s}$ are the number of CUs that need splitting and no splitting respectively when MSE is less than the threshold; NUM_{CU} represents the number of total CUs. Table I tabulates the percentage of correct and wrong classification for HEVC common test sequences in Class B, C, D and E. These sequences are encoded with low delay B he10, and QP is set to 22, 27, 32 and 37 respectively. From our statistics results, it can be seen that only less than 10% CUs are coded with the wrong CU size. Another observation is that as the increasing of *k*, the correctness as well as the error probability increases, which results in more time reduction and more efficiency penalty. This observation suggests that our scheme is effective in balancing the encoding complexity and the compression performance.

B. Proposed Fast Coding Unit Decision Algorithm

In our proposed scheme, MSE is utilized to guide the CU splitting process. We summarize the whole process of the proposed scheme in Fig. 4. Firstly, the first two inter frames are coded with the conventional mode decision process; and the MSE of Inter 2N×2N prediction residual block are obtained from CUs that need no splitting in different depth. To reduce the hardware expenses, we divide the MSE below 200 into 200 bins evenly, and MSE beyond 200 as one bin. Once one MSE is within the scope of the bin, the value of the bin is increased by one. Then the value of k is set according to target application and the value of MSE_{thres} corresponding to each CU depth can be finally determined.

For mode decision process of other inter frames, MSE is initialized with a large value. After that SKIP mode and Inter 2N×2N are computed. Then we calculate the MSE of Inter 2N×2N when the current CU size is larger than 8×8, for 8×8 CU cannot be divided into smaller ones. After we sequentially checked other remaining modes in current CU, MSE is compared to MSE_{thres}. If MSE is smaller than MSE_{thres}, the recursive splitting process is terminated; otherwise, we proceed to the next CU depth.

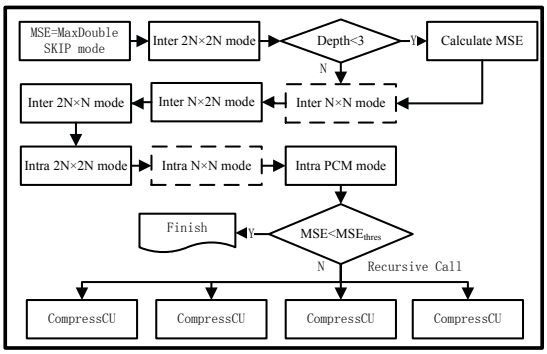


Fig. 4. Architecture of the proposed scheme.

III. EXPERIMENTAL RESULTS

The proposed algorithm was implemented on HM6.0. The experiments were implemented with random access he10 and low delay B he10 setting [8]. The test sequences are common test sequences of HEVC, varying from WQVGA to 1600p. The testing platform used is Inter(R) Xeon(R) X5450-3GHz with eight cores, 8 GB RAM. Experiments were done on the common test sequences with quantization parameters 22, 27, 32 and 37 as specified by [8]. The coding performance is measured by BD-rate and encoder complexity is measured by time saving ΔT. BD-rate is calculated with the method descried in [9], and ΔT is calculated as:

$$\Delta T = \frac{T_{anchor} - T_{proposed}}{T_{anchor}} \times 100\% \quad (4)$$

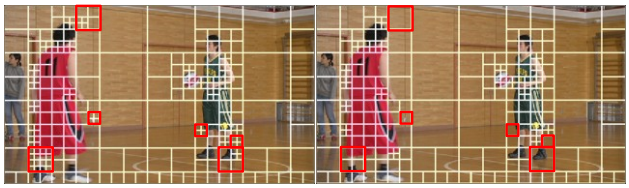
where T_{anchor} and $T_{proposed}$ are the total encoding time of anchor and the proposed encoder respectively.

We use median as the threshold. That's to say *k* is set to 0.5. The coding performance (measured by BD-rate) and encoder complexity (measured by ΔT) compared to original HM6.0 are shown in Table II. The values are averaged over all sequences from their corresponding class.

TABLE II
BD-RATE AND ENCODING COMPLEXITY REDUCTION

	Random Access HE10				Low delay B HE10			
	Y	*U*	*V*	*ΔT*	*Y*	*U*	*V*	*ΔT*
Class A	0.1%	0.1%	0.1%	22.4%				
Class B	0.3%	0.2%	0.1%	28.4%	0.2%	0.3%	0.4%	25.6%
Class C	0.2%	0.4%	0.4%	23.0%	0.3%	0.9%	1.0%	20.0%
Class D	0.2%	0.3%	0.3%	17.0%	0.3%	0.1%	-0.3%	15.0%
Class E					0.1%	-0.1%	0.0%	34.83%
Overall	0.2%	0.3%	0.2%		0.3%	0.3%	0.3%	
Average ΔT	24%				25%			

From Table II, it can be seen that our proposed fast CU decision algorithm achieves 15% to 34.83% encoding complexity reduction with negligible bitrate increase. For high resolution and motionless sequences, which prefer larger CU, more coding time can be saved. Fig. 5 depicts the optimal CU partition of BasketballPass(416×240) with HM6.0 and our proposed algorithm. Different partition regions are marked with red line. The CUs, whose MSE less than MSE_{thres} in Fig. 5(b), are determined to be split in Fig. 5(a). Although the partition decision is different, our scheme can still maintain negligible loss, because the coding efficiency improvement is marginal when MSE is less than MSE_{thres}. Therefore, this algorithm has little negative influence on the coding efficiency and it is able to speed up the encoding procedure significantly.



(a) HM6.0 (b) Proposed

Fig. 5. Optimal CU splitting with HM6.0 and proposed scheme.

As described before, the value of k can be set accordingly, thus we can balance the coding performance and the time reduction. Fig. 6 shows the relationship between coding performance, coding time reduction and the threshold k, which are the average of class C and D. From Fig. 6, we can see that with the increasing of k, the time reduction varies from 17% to 29%, and the BD-Rate varies from 0.1% to 0.6%. From Fig. 6, we can conclude that it is proper for *k* to be set to 0.5, for it is an inflection point for this curve.

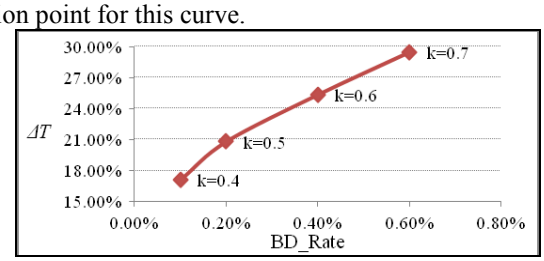


Fig. 6. BD-Rate and complexity reduction for different k value

To verify the compatibility with the existing CU splitting early termination algorithm in HM6.0 [7], another set of experiments with that algorithm on are done. Experimental results show that the coding time reduction is 34.12% and the corresponding BD-Rate is 0.55% for class C and D if we combine the algorithm in this paper and in HM6.0. Our method works well with the existing fast algorithm in HM6.0.

IV. CONCLUSION

We proposed a novel coding unit early-termination algorithm for the HEVC encoding. This algorithm takes advantage of the correlations between the MSE of prediction residuals and the splitting property in the current CU level in order to terminate recursive CU splitting process as early as possible. Simulation results show that, with the proposed method, approximately a quarter of the encoding running time can be saved while preserving the comparable coding performance. In summary, our algorithm is compatible with the existing HEVC standard and provides more flexibility to reduce the encoding complexity by defining additional early-termination criteria to avoid unnecessary search efforts.

V. ACKNOWLEDGEMENT

This research is supported by the 973 program (2009CB320903), the 863 program (2012AA011505, 2011BAH08B01) and the National Science Foundation of China (60833013, 61103088), which are gratefully acknowledged.

REFERENCES

[1] W.-J. Han, J. Min, I.-K. Kim, E. Alshina, A. Alishin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. M. Hong, M.-S. Cheon, N. Shlyakhov, K. McCann, T. Davies, J.-H. Park, "Improved Videl Compression Efficiency Through Flexible Unit Representation and Corresponding Extension of Coding Tools," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1709-1720, Dec. 2010.

[2] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, R. Susanto and C. C. Kuo, "Fast intermode decision in H.264/AVC Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 6, pp. 953-958, Jul. 2005.

[3] Z. Liu, L. Shen, and Z. Zhang, "An efficient intermode decision algorithm based on motion homogeneity for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 1, pp. 128-132, Jan. 2009.

[4] T. Zhao, H. Wang, S. Kwong and C.-C. Jay Kuo, "Fast mode decision based on mode adaptation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 5, pp. 697-705, May. 2010.

[5] I. Choi, J. Lee, and B. Jeon, "Fast coding mode selection with rate-distortion optimization for MPEG-4 Part-10 AVC/H.264," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1557-1561, Dec. 2006.

[6] JCT-VC, "Early termination of CU encoding to reduce HEVC complexity", *JCTVC-F045*, JCT-VC Meeting, Torino, July 2011.

[7] JCT-VC, "Coding tree pruning based CU early termination", *JCTVC-F092*, JCT-VC Meeting, Torino, July 2011.

[8] F. Bossen, "Common test conditions and software reference configurations", JCTVC-H1100, 8th JCT-VC Meeting, San Jose, CA, USA, 1-10 February, 2012.

[9] G. Bjontegaard, "Calculation of average PSNR difference between RD-curves," in Proc. ITU-T Q.6/SG16 VCEG 13th Meeting, Austin, TX, Apr. 2001, Doc. *VCEG-M33*.