Early TU Split Termination in HEVC Based on Quasi-Zero-Block

Yunyu Shi, Zhiyong Gao, Xiaoyun Zhang
Institute of Image Communication and Network Engineering
& Shanghai Key Laboratory of Digital Media Processing and Transmissions
Shanghai Jiaotong University
Shanghai, China
yunyushi@sjtu.edu.cn

***Abstract*—The new introduced concept, transform tree or residual quadtree (RQT) in HEVC standard, brings high coding performance along with high computational complexity because the optimal TU partition is estimated in a rate-distortion sense at the encoder sides by testing all kinds of partitions. This paper focus on early TU split termination in terms of the tradeoff between computation and coding quality. An early TU split termination scheme based on quasi-zero-block (QZB) is proposed, which is defined by two aspects about the quantized transform coefficients; the sum of all absolute coefficients and the number of nonzero coefficients. Besides, the selective probability of every TU depth is calculated and analyzed. Experimental results show that the proposed method (HM-QZB) could achieve 22.8% reduction in encoding time and 50.59% reduction in TU processing time compared to the HEVC test model HM10.0 encoder with about 0.04dB BD-PSNR loss in coding performance.**

Keywords-HEVC; early termination; TU split; residual quadtree (RQT); quasi-zero-block (QZB)

I. INTRODUCTION

HEVC (High Efficiency Video Coding standard) project has been conducted by JCT-VC (Joint Collaborative Team on Video Coding) since 2010, working together in a partnership between ITU-T and ISO/IEC. The main goal of HEVC video coding standard is to provide a doubling in compression efficiency with respect to existing standards, such as H.264/AVC High Profile --- in the range of 50% bit rate reduction for equal perceptual video quality [1]. Reference software, called the HEVC Test Model (HM), is being developed along with the standard. At the time of writing, the first edition of the HEVC standard has been finalized in January 2013 [2] and the current version of HM is HM10 [3].

Unlike H.264/AVC, a significant difference of HEVC is the use of a more flexible quad tree structure based on a coding tree unit (CTU), which has a size selected by the encoder and can be larger than a traditional macro block. Four different concepts, Coding Tree Unit (CTU), Coding Unit (CU), Prediction Unit (PU) and Transform Unit (TU) are introduced in HEVC. This variable-size, adaptive approach is particularly suited to larger resolutions and plays a major role in the substantial performance gains exhibited by HEVC relative to previous video coding standards[4][5]. Meanwhile, it brings high computational complexity because the optimal partitioning is estimated in a rate-distortion sense at the encoder sides by testing all kinds of partitions.

To reduce the computational complexity of the encoding process, most existing approaches reduce the number of CUs or PUs to be tested [6]. References [7] is a typical example related to fast HEVC encoding methods by proposing a fast CU size decision algorithm based on motion homogeneity checking, RD cost checking and SKIP mode checking to skip motion estimation on unnecessary CU sizes. However, early TU determination has rarely been exploited to reduce the computational complexity in the HEVC encoder. Reference [6] proposes an early TU decision method for fast video encoding in HEVC by pruning the TUs at an early stage based on the number of nonzero DCT coefficients. Two early termination schemes for both TU Merge and TU Split procedures are applied respectively for nonzero-blocks [7]. There are certain similarities among those previous methods and they are tested with earlier HM version, such as HM3.0.

This paper focus on early TU split termination in terms of the tradeoff between computational complexity and compression quality. An early TU split termination based on quasi-zero-block is proposed, where the concept of the quasi-zero-block is defined by the number of nonzero quantized transform coefficients and the sum of all absolute coefficients. Besides, the selective probability of every TU depth is calculated and analyzed. Experimental results show that the proposed method achieves a 22.8% reduction in encoding time and 50.59% reduction in TU processing time compared to the HEVC test model HM10.0 encoder with about 0.04dB BD-PSNR loss in coding performance.

The rest of the paper is organized as follows. Transform Unit structure of HEVC is the topic of Section II. Section III describes the proposed early TU termination method in detail. The experimental results of the proposed method compared with other prior similar method are presented and analyzed in Section IV. Section V concludes the paper briefly.

II. RQT STRUCTURE OF HEVC

In the main profile of HEVC, a slice is partitioned into multiple CTU which are allowed to have size from 8×8 up to 64×64 [4]. A CTU includes one luma coding tree blocks (CTB), two chroma CTBs, and syntax specifying coding data and further subdivision. Each leaf node of the coding tree is called CU with coding blocks (CB). The CU can have multiple PU and a tree of TU. Similarly, each CB is split into prediction blocks (PB) and transform blocks (TB). The shape of the PU is specified by the splitting type whereas that of TU is represented by another quad tree, called the transform tree or residual quad tree (RQT).

A. RQT structure of HEVC

The TU is a basic representative block having residual or transform coefficients for applying the integer transform and quantization. A RQT has its root at the CU level, and is also structured by successive signaling of the syntax element *split_transform_flag* in a recursive manner. In the current version of HEVC standard, there's only square shape RQT (SRQT).

In HEVC, both the PU size and the TU size can reach the same size of the corresponding CU. This leads to the fact that the size of TU may be larger than that of the PU in the same CU, i.e., residuals from different PUs in the same CU can be transformed together. For example, when the TU size is equal to the CU size, the transform is applied to the residual block covering the whole CU regardless of the PU splitting type. Note that this case exists only for inter coded CU, since the prediction is always coupled with the TU splitting for intra coded CU [4].

Although the coding efficiency in HEVC can be improved by using various transform block sizes (from 32×32 down to 4×4 samples), the computational complexity increased dramatically in terms of the transform kernel size and the transform coding structure [6].

The maximum depth of transform tree is closely related to the encoding complexity. To provide the flexibility on this feature, HEVC specifies two syntax elements in the SPS which control the maximum depth of transform tree for intra coded CU and inter coded CU, respectively. The case when the maximum depth of transform tree is equal to 1 is denoted as implicit TU splitting since there is no need to transmit any information on whether the TU is split. In this case, the transform size is automatically adjusted to be fit inside the PU rather than allowing transform across the boundary [4].

B. TU split determination based on RD cost

As described above, the most suitable TU partition is estimated in a rate-distortion sense at the encoder sides by exhaustively testing all kinds of TU partition. In the example shown in Fig. 1, the RD cost evaluation is performed a number of times within each quad tree structure: once for the TU depth=0, 4 times for the depth=1, and 16 times for the depth=2 except if the size of the current CU is 64×64 or 8×8.

III. Early TU Termination Based on QZB

A zero-block (ZB) means that the quantized transforms coefficients of a TU are all zero. For these blocks, the processing of inverse quantization and inverse transform are unnecessary. Luma ZB contributes a reasonable amount as shown in Table 1. These values are obtained from 20 frames of two HD videos respectively, under the random access condition tested by HM10.0 with some fast search options, which are detailed in Reference [5].

A. Definition of quasi-zero-block

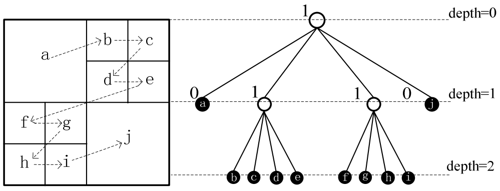

Figure 1. Example of transform tree and block partitioning (black node: determined TU).

Subtle difference between the reference video frame and its corresponding compressed version could not be observed by human eyes according to human inherent visual redundancy. So, a non-zero-block could be regarded as a ZB if the features between them are very similar. The key problem here is how to make sure the small difference. In this paper, the concept of quasi-zero-block (QZB) is proposed based on human redundant subjective information, which may be useful for reducing encoding time.

Table I. ZB and QZB Percentage (%)

Video	QP	Pred Mode	ZB	QZB	Δ
Basket ball Drive	22	Intra	47.75	68.19	20.44
		Inter	36.66	59.48	22.82
	27	Intra	69.13	85.68	16.55
		Inter	41.50	67.24	25.74
	32	Intra	80.05	92.25	12.19
		Inter	51.45	76.31	24.86
	37	Intra	87.23	95.94	8.71
		Inter	57.10	82.31	25.20
	Avg.	Intra &Inter	58.86	78.42	19.57
Park Scene	22	Intra	22.11	33.53	11.42
		Inter	48.77	61.70	12.92
	27	Intra	38.83	53.10	14.26
		Inter	57.39	69.00	11.61
	32	Intra	57.25	74.21	16.69
		Inter	63.11	76.71	13.60
	37	Intra	74.44	89.13	14.69
		Inter	68.46	84.66	16.20
	Avg.	Intra &Inter	53.83	67.75	13.92
Avg.		Intra &Inter	56.35	73.09	16.74

The concept of the ZB is determined by the sum of all absolute quantized transform coefficients (AbsSum for short). Obviously, AbsSum of a ZB is always zero. So, a QZB could be defined by two aspects; AbsSum of all quantized transform coefficients and the number of nonzero quantized transform coefficients (NNZ for short). The transform block could be regarded as a QZB if these two values are less than their respective threshold.

Definition: A TB is a QZB if NNZ and AbsSum of quantized transform coefficients $c_{i,j}$ ($i, j = 0..nS-1$) satisfy the conditions as follows:

$$\begin{cases} NNZ \le T_N \\ AbsSum \le T_S \end{cases}, \quad \begin{cases} NNZ = \sum_{i=0}^{i<nS} \sum_{j=0}^{j<nS} \left(c_{i,j} \,?1,0 \right) \\ AbsSum = \sum_{i=0}^{i<nS} \sum_{j=0}^{j<nS} \left| c_{i,j} \right| \end{cases}$$

, where nS represents the size of the current TB, T_N and T_S are two thresholds.

Table 1 also gives the proportion of QZB tested by the same test conditions for ZB values, where $T_N = 3$, $T_S = Log_2 nS$. The average proportion of ZB and QZB are more than 56% and 73% respectively, and the increase from ZB to QZB is about 16.74%. In the general, the proportion of ZB and QZB always increase when the QP increases. The distribution of ZB and QZB percentages doesn't change in a consistent way with variable QP for different videos. Take Basketball Drive for an example. With different kind of prediction mode, the proportion of ZB and QZB in intra-mode is higher than that in inter-mode. However the increase amount from ZB to QZB within inter-mode is about 1.2-3 times than that within intra-mode.

B. Early TU split termination

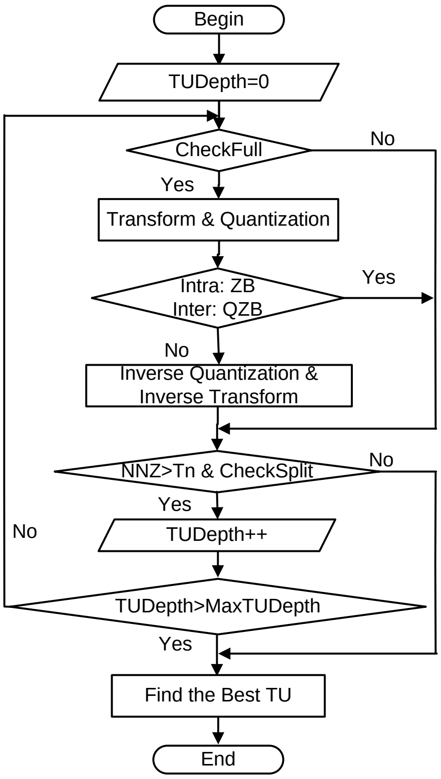

Figure 2. Simplified diagram of TU split determination.

In this section, an early luma TU split termination scheme is proposed based on QZB defined as above.

Fig.2 simply depicts the TU split determination processing in the proposed algorithm. Check Full and Check Split represent a TU is checked as the same size of the current CU or by being split into four equal-size sub-TU. When the size of the current CU is 64×64, Check Full is always false for that the maximum size of TU is 32×32. The TU is checked recursively until reaching the maximum TU depth and the best TU is determined by the minimum RD cost calculated within each quad tree structure.

The difference between the proposed flowchart and the original version from HM10.0 encoder is at the position of two kinds of judgments. One is the luma zero-block determination. The QZB takes the place of ZB in the inter-prediction mode, while ZB is also retained for intra-prediction mode in order to minimize quality loss of I-slice. The other different judgment is whether to check the sub-TU by estimating the NNZ in addition to original condition for both intra- and inter-prediction modes.

Calculation of NNZ and AbsSum hardly don't increase computation complexity for AbsSum is calculated to determine ZB in the previous encoder.

Correspondingly, it's very important to determine the values of these two thresholds T_N and T_S . The selection of T_N depends on the conclusion of Reference [6], where exploits the strong relationship between the determined TU size and NNZ and finds the TU size at the root node is very likely to be the determined TU size when NNZ is small. The value of T_S is found in an experimental way and is adaptive to the TU size ($Log_2 nS$) according to the AbsSum value of every determined TU given by HM10.0.

C. Statistic of TU depth

TABLE II. SELECTION RATIO OF THREE TU DEPTHS (%)

Video	QP	depth=0	depth=1	depth=2
Basketball Drive	22	38.04	43.28	18.68
	27	44.62	41.24	14.14
	32	46.80	41.43	11.77
	37	48.29	42.78	8.93
	Avg.	44.44	42.18	13.38
Park Scene	22	11.25	42.61	46.14
	27	12.80	44.90	42.30
	32	15.63	48.43	35.94
	37	19.01	52.39	28.60
	Avg.	14.67	47.08	38.25
Avg.		29.55	44.63	25.81

The maximum depth of transform tree is adjustable and is specified in the slice header syntax. The values in the HM are set according to the test scenario and are always set to 3 for both intra- and inter-block.

Table 2 shows the selection percentage of different TU depth (expressed by 0, 1 and 2) generated by HM10.0 given four QPs (22, 27, 32, 37).

Generally speaking, the determined proportion of depth=0 increases when the QP increases, while the opposite situation exists about depth=2. The contributions of different TU depths are changeable greatly for different videos. For an example, the selection probability of depth=2 is only 13.38% and is much lower than that of the other two depths for

Basketball Drive. However, the minimum percentage is from depth=0, only 14.67% for Park Scene.

On the average, the selection of every TU depth is about 29.55%: 44.63%: 25.81%, which could be regarded as a statistic basis for the setting of the maximum RQT depth in terms of encoding complexity. And the results may explain the relevant tests of the Reference [4], where gives the coding efficiency loss when the transform tree depth is reduced from 3 to 2 and from 3 to 1.

IV. Experimental Results

A. Test conditions

The JCT-VC common test conditions [9] are used to investigate the performance of the proposed method. All tests are performed by setting configurations in HM10.0 reference software. Test conditions are summarized as follows.

- HEVC main profile is used.
- Four QP are used: 22, 27, 32, and 37.
- Random access configuration is used.
- Several fast encoding modes are enabled such as ECU, CFM and ESD [5].
- A total of 5 video sequences whose resolution is 1920×1080 are evaluated.
- 50 frames of those video sequences are tested respectively.
- Encoding times are obtained on an X86 computer containing Inter Core i7-2600 clocked at 3.40GHz and using VS2008.

For measuring the performance of the proposed method, the widely well known BD-rate and BD-PSNR are used in this section [10]. The input of the measurement method includes the bit rate and the combined PSNRYUV, which is calculated as the weighted sum of the PSNR per picture of both luma and chroma components. The output includes a single average difference in bit rate and in PSNRYUV that takes into account the tradeoffs between luma and chroma component fidelity [10].

$$PSNR_{YUV} = (6 \times PSNR_Y + PSNR_U + PSNR_V)/8$$

The reduction of processing time is represented by ΔTimeEnc and ΔTimeTU. ΔTimeEnc indicates the whole encoding time difference between the proposed method and HM10.0, while ΔTimeTU is for the TU processing time between them. They are defined as follows with one QP value.

$$\Delta TimeEnc = \frac{TimeEnc(HM) - TimeEnc(\Pr oposed)}{TimeEnc(HM)} \times 100\%$$

$$\Delta TimeTU = \frac{TimeTU(HM) - TimeTU(\Pr oposed)}{TimeTU(HM)} \times 100\%$$

The final results of ΔTimeEnc and ΔTimeTU are calculated by averaging the values under four QP conditions respectively as shown in Table 3.

B. Experimental results

Table 3 gives the results of the conducted experiments about the previous method (HM-NNZ for short) and the proposed method (HM-QZB), where the anchor is the HM10.0. The RD curves of two videos are shown in Fig. 3.

HM-NNZ is an early TU determination method modified from the Reference [6], where the NNZ is selected as a threshold to stop the RD cost evaluation below the root node. NNZ is set to 3 in this experiment and is as the same as the Reference [6]. HM-NNZ could be regarded as the extension of the Zero-Block-Inheritance early termination for TU split from the Reference [7], that is to say, no more evaluation on further TU partition is executed if the current TU is found to be a zero-block.

The values of two thresholds, T_N and T_S, could directly influence the compression performance of the proposed method, HM-QZB. The lower the thresholds values, the higher coding quality; while the higher the thresholds, the less computational complexity. T_N and T_S are set to 3 and $Log_2 nS$ as explained in the preceding section for this experiment. And in the HM-NNZ and HM-QZB, the maximum depth of RQT is set to 2 for both intra- and inter-prediction mode according to the above analysis.

It is noticeable from the Table 3 that HM-NNZ could reduce 18.16% encoding time and 44.64% TU processing time with 0.02dB PSNR decrease. Although HM-QZB generates about 0.04dB quality loss, it could further reduce about 22.82% encoding time and 50.59% TU processing time with subtle rate increase (less than 2%).

Compared with HM10.0, HM-NNZ works well for Basketball Drive (21.86% encoding time reduction with negligible quality loss, -0.01dB), the reason maybe found from the selection probability statistic of TU depths in Section 3.3, where it's much fewer possibility to select depth=2. So, the reduction of RQT maximum depth from 3 to 2 brings tiny influence on the encoding quality. The opposite situation could be explained in the same way for Park Scene.

The significant difference of HM-QZB is the QZB detection with respect to HM-NNZ. From the observation in Section 3.1, the increase of luma inter QZB to ZB is about 24.66% and 13.58% on the average, respectively for Basketball Drive and Park Scene. So HM-QZB is more effective for Park scene than for Basketball Drive in terms of compression performance, 1.36% versus 2.53% BD-rate; while reduces less encoding time, 19.13% versus 26.58%.

V. Conclusions

In this paper, an early TU split termination scheme is proposed based on QZB detection. Experimental results indicate that the proposed method could achieve 22.82% encoding time reduction and 50.59% TU processing time reduction with marginal quality loss (about 0.04dB). To some extent it's effective for a fast HEVC encoder in terms of the tradeoff between computational complexity and encoding performance.

The proposed method could be combined with other early determination schemes for CU and PU in the future. And the thresholds in this paper are also worthy studying by some human visual features, such as Just-Noticeable-Model.

ACKNOWLEDGMENT

The work was supported in part by National Natural Science Foundation of China key project (No.61133009), National Natural Science Foundation of China project (No. 61301116) and the Shanghai Key Laboratory of Digital Media Processing and Transmissions (12DZ2272600).

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[2] JCT-VC, “High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Last Call),” JCTVC- L1003_v34, Jan. 2013.

[3] JCT-VC, “High Efficiency Video Coding (HEVC) Test Model 10 (HM10) Encoder Description,” JCTVC-L1002_v3, Jan. 2013.

[4] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, “Block Partitioning Structure in the HEVC Standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1697-1706, Dec. 2012.

[5] F. Bossen, B. Bross, K. Suhring, and D. Flynn, “HEVC Complexity and Implementation Analysis,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp 1685-1696, Dec. 2012.

[6] Kiho Choi, Euee S. Jang, “Early TU decision method for fast video encoding in high efficiency video coding,” Electronics Letters, vol. 48, no. 12, pp. 689-691, 2012.

[7] Su-Wei Teng, Hsueh-Ming Hang, and Yi-Fu Chen, “Fast mode decision algorithm for residual quadtree coding in HEVC,” 2011 IEEE Visual Communications and Image Processing (VCIP), pp. 1-4, 6-9 Nov. 2011, Tainan.

[8] Liquan Shen, Zhi Liu, Xinpeng Zhang, Wenqiang Zhao, and Zhaoyang Zhang, “An Effective CU Size Decision Method for HEVC Encoders,” IEEE Transaction on Multimedia, vol. 15, no. 2, pp. 465-470, Feb. 2013.

[9] JCT-VC, “Common test conditions and software reference configurations,” JCTVC- K1100, Oct. 2012.

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

[11] 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 Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1669-1684, Dec. 2012.

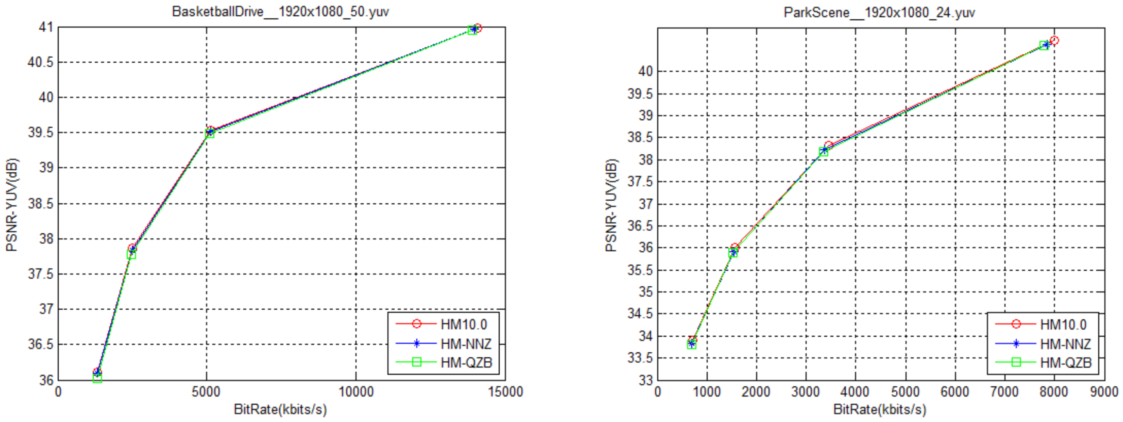

Figure 3. Selected rate-distortion curves.