A Context-based Fast Encoding Quad Tree Plus Binary Tree (QTBT) Block Structure Partition

Ismail Marzuki, Choi Hansol, and Donggyu Sim
Kwangwoon University
ismail@kw.ac.kr, whiteblack4@kw.ac.kr, dgsim@kw.ac.kr

Abstract

This paper proposes an algorithm to speed up block structure partition of quad tree plus binary tree (QTBT) in Joint Exploration Test Model (JEM) encoder. The proposed fast encoding of QTBT block partition employs three spatially neighbor coded blocks, such as left, top-left, and top of current block, to early terminate QTBT block structure pruning. The propose algorithm is organized based on statistical similarity of those spatially neighboring blocks, such as block depths and coded block types, which are coded with overlapped block motion compensation (OBMC) and adaptive multi transform (AMT). The experimental results demonstrate about 30% encoding time reduction with 1.3% BD-rate loss on average compared to the anchor JEM-7.1 software under random access configuration.

1. Introduction

In the middle of October 2015, an investigation for a Future Video Coding (FVC) is begun and has been cultivated by the Joint Video Exploration Team (JVET) with expectations to enhance video coding technology beyond the capabilities of HEVC [1]. During the development of the FVC, the Joint Exploration Test Model (JEM) software has been used to inspect several potential technologies brought to the JVET meetings on top of the current video coding standard software, High Efficiency Video Coding reference model (HM) [2]. According to the requirement for FVC document, JEM is expected to be able to support Wide Color Gamut (WCG), High Dynamic Range (HDR), high frame rate over 120 frames per second for temporal resolution and up to 8K×4K for spatial resolution with 8-16 coding bit depth. The development is highly expected to be able to serve the needs of various applications, such as broadcast, digital cinema with a larger screen, low-delay interactive communications, mobile communications, and so on [3]. An overview of the technical features and the key elements of JEM software have also been presented [4] as well as the complexity comparison analysis between JEM and HM software [5]. We also have several works to improve performance of JEM [6-7] and another approach for controlling bit rate for high resolution videos need also to be considered for FVC which we demonstrated on top of the HEVC [8].

Among those potential technologies designed for the FVC, a quad-tree plus binary-tree (QTBT) block structure is introduced. The QTBT in JEM software is organized to remove the concept of multiple block partitioning types (e.g., prediction unit and transform unit) in HEVC [9]. It is designed with larger Coding Tree Unit (CTU) size up to 256×256 and is set to 128×128 by default. In encoder, the QTBT will first split the larger block size of coding unit (CU) with quad tree (QT) structure (block size from 128×128 to 64×64), followed by the horizontal binary-tree (BTH) and the vertical binary-tree partitioning (BTV). Then, the QT partitioning may be performed again for smaller blocks. Each block partitioning cost will be saved and compared to other blocks in determining the best block partitioning structure with the best coding mode decisions. For JEM software, this partitioning routine consumes very high computational complexity. In this paper, we propose a new study to reduce the complexity of the recursive QTBT block partitioning in JEM-7.1. We design the proposed algorithm to early terminate the binary-tree (BT) block partitioning for time reduction with the minimum BD-rate loss. We organize some statistical similarity of spatially three neighboring blocks of the current coded block, i.e., top, left, and left-top blocks by collecting information of BT depth, overlapped block motion compensation (OBMC), and adaptive multi transform (AMT).

This paper is organized as follows. In Section 2, the proposed algorithm will be presented. In Section 3, we evaluate the proposed algorithm with several test sequences. Finally, we summarize and conclude the papers in Section 4.

2. The proposed fast QTBT encoding

As aforementioned, JEM software can have two types of block structures i.e., QT and BT. The investigation has been reported that the QTBT block structure can provide coding efficiency for the FVC. However, it leads higher computational complexity in JEM software than its previous video coding standard's block structure.

For the proposed algorithm, we first analyzed the statistical similarity information that the current block may have the same depth information as its three spatially neighbor blocks. In our studied, by just using depth similarity knowledge and employing all block structures of QT and BT to be skip from the partitioning procedure, we can save about 60% encoding time for class D. However, the BD-rate loss is very high at 10% in approximation. It is due to the fact that the current block depth has very high possibility to be the same as its neighbor coded blocks. To maintain the minimum BD-rate loss, we then employed not only depth information of spatially three neighboring blocks, but also the coded block types, such as OBMC and AMT coded block. In addition, instead of applying the algorithm for all QTBT block structure, we then decided to only early terminate the partitioning of BT blocks. This means that when the current block and one of its three neighboring blocks has the same BT depth, transform index, and is coded with OBMC, the BT block structure will be skip from the partitioning procedure of QTBT block structure. Note that the proposed algorithm is only possible if the merge and skip mode is determined as the best coding mode of the current block.

The following figure is given to illustrate the proposed algorithm presented in this paper. In the figure, the condition is placed after a block tries all coding block mode decision possibilities of encoder. If the condition is satisfied, the BTH and BTV is skip from the QTBT block structure partitioning routine. Note that coding block mode decision in JEM software consists of FRUC merge skip mode, affine merge skip mode, merge skip mode, intra PCM mode, intra mode, FRUC merge mode, affine merge mode, merge mode, inter mode, and affine mode.

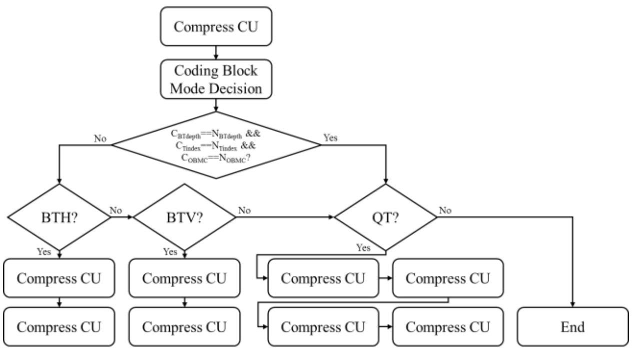

Figure 1: Flowchart of the proposed algorithm

3. Experimental results

In comparison against JEM-7.1, the proposed algorithm was tested by following the common test conditions for JEM which employs video test sequences of Class A1 (Tango2 and CampfireParty), Class A2 (CatRobot and Rollercoaster2), Class B (Kimono and BasketballDrive), Class C (BasketballDrill, BQMall, PartyScene, and RaceHorses), and Class D (BasketballPass, BQSquare, BlowingBubbles, and RaceHorses). All test sequences were coded in full frames with quantization parameter values 22, 27, 32, and 37 under 'Random Access configuration' [10]. The proposed algorithm was evaluated on Windows 10 (64-bits) OS over 3.00GHz Intel (R) Core(TM) i7-5960X CPU with 32GB RAM.

Table 1 shows the RD performance and encoding time of the proposed algorithm for JEM fast QTBT block partition. In our experiment, we found that total encoding time of the proposed algorithm reduced about 30.28% with BD-rate loss about 1.30% on average compared to JEM-7.1 reference software. In this regard, the proposed algorithm argues to be able to decrease encoding complexity with negligible BD-rate loss.

Table 1: Overall RD Performance and Encoding Time of the Proposed Algorithm

Sequence (Resolution)	BD-rate (piecewise cubic)		
	Y	U	V
Class A1 (3840×2160)	1.06%	0.18%	1.45%
Class A2 (3840×2160)	1.52%	1.40%	1.16%
Class B (1920×1080)	1.36%	0.18%	0.37%
Class C (832×480)	1.26%	0.59%	0.81%
Class D (416×240)	1.33%	0.26%	0.44%
Average of BD-rate	1.30%	0.49%	0.78%
Encoding Time Reduction	30.28%		

4. Conclusion

The proposed algorithm attempts to reduce the complexity of QTBT block partitioning structure in JEM software. The proposed algorithm is designed to early terminate the BT block structure pruning, both BTH and BTV block structure. When the proposed algorithm is satisfied, only QT block structure will be processed to save ending time. The proposed algorithm demonstrates in decreasing average encoding time about 30% with negligible average BD-rate loss about 1.30% for Luma and about 0.49% and 0.78% for both Chroma components under random access configuration.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2018R1A2B2008238)

References

[1] J. Chen, E. Alshina, G. J. Sullivan, J. R. Ohm, and J. Boyce, "Algorithm Description of Joint Exploration Test Model 1," JVET-A1001, 2015.

[2] Joint Exploration Test Model (JEM) version 7.1, [Online]: https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftwar

e/tags/HM-16.6-JEM-7.1

[3] Requirements for Future Video Coding (H.FVC), w15340, 2017, [Online]: http://mpeg.chiariglione.org/sites/default/files/files/standards/parts/docs/w15340.docx

[4] I. Marzuki and D. Sim, "Overview of Potential Technologies for Future Video Coding (FVC) in JEM Software: Status and Review," IEIE Transaction on Smart and Computing, vol. 7, no. 1, pp. 22-35, 2018.

[5] X. Li, X. Wu, I. Marzuki, Y. J. Ahn, and D. Sim, "Complexity Analysis of HM and JEM Encoder Software," 2016 The Korean Institute of Broadcast and Media Engineers Summer Conference, pp. 313-315, 2016.

[6] I. Marzuki and D. Sim, "Modification of Deblocking Filter Decision in JEM Reference Software," 2017 년도 대한전자공학회 하계종합학술대회, pp. 514-516, 2017.

[7] X. Li, I. Marzuki, S. Jang, and D. Sim, "Adaptive Loop Filter Complexity Reduction Based on Local Gradient," ICEIC 2017 International Conference on Electronics, Information, and Communication, pp. 571-573, 2017.

[8] J. An, Y. W. Chen, K. Zhang, H. Huang, Y. W. Huang, S. Lei, "Block Partitioning Structure for Next Generation Video Coding", MPEG doc. m37524 and ITU-T SG16 Doc. COM16-C966, 2015.

[9] I. Marzuki, Y. J. Ahn, and D. Sim, "Tile-level Rate Control for Tile-parallelization HEVC Encoders," J Real-Time Image Proc., 2017. doi: 10.1007/s11554-017-0720-5

[10] K. Suehring and X. Li, "JVET Common Test Conditions and Software Reference Configurations," JVET-G1010, 2017.