Low complexity reference frame selection in QTBT structure for JVET future video coding

Sang-hyo Park
Communications & Media R&D Division
Korea Electronics Technology Institute
Gyeonggi-do, South Korea
sanghyo.park@keti.re.kr

Tianyu Dong and Euee S. Jang
Department of Computer Science
Hanyang University
Seoul, South Korea
dongtianyu@hanyang.ac.kr
esjang@hanyang.ac.kr

***Abstract*— In this paper, we propose a reference frame search method for JVET future video codec (FVC) that employs the quadtree plus binary tree (QTBT) structure. Among many new technologies proposed in FVC, QTBT poses a significant challenge since it contains the structural change of coding tree unit from HEVC. To reduce the encoding complexity of FVC with QTBT structure, we investigated some redundancy in motion estimation process—particularly, the reference frame search. In this paper, we present a method that effectively restricts the reference frame search range of general motion estimation as well as of affine motion estimation, exploiting the dependence within QTBT structure. The proposed method minimizes the maximum of the reference frame search ranges per each coding unit (CU) based on the prediction information of parents node. To be specific, the prediction direction and the index of reference frame of parent node were used. In addition, the proposed method utilizes the information of binary tree depth and of temporal layer to prevent undesired coding loss. The experimental results showed that the proposed method decreased the encoding time of motion estimation by 34% on average in comparison with joint exploration test model (JEM) 3.1, maintaining a reasonable coding efficiency (less than a 0.3% BD-rate loss).**

Keywords— FVC; JEM; video coding; video compression; motion estimation; reference frame search; encoder complexity.

I. INTRODUCTION

Motion estimation (ME) has been a pivot in video compression technology by removing temporal redundancy efficiently among consecutive pictures. To enhance the compression efficiency, recent video codecs tried to estimate motion in various shape and size. For instance, HEVC takes various motion partition strategies—square, symmetric, and asymmetric partitions—within quadtree-based variable coding unit (CU) [1]. By evaluating various CUs within the recursive quadtree structure, those various block shapes and sizes are enabled for better motion estimation.

Recently in the joint video exploration team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG11, quadtree plus binary tree (QTBT) structure was newly introduced in FVC with better coding efficiency than HEVC [2]. The principle of QTBT is basically to add binary tree at the traditional quadtree leaf nodes and to unify motion partitions and transform partitions within a CU. By adding the binary tree, in brief, ME can be executed for a very thin block (e.g., width is one eighth of height). However, due to the increased partition cases to be evaluated, the associated encoding complexity has been highlighted, which should be resolved for low-complexity encoding applications. Furthermore, the complexity of ME in FVC must be higher than that of HEVC since ME in FVC additionally attempts new techniques such as affine motion estimation (please refer to the algorithm description of FVC [3]).

In this paper, we propose a method that skips certain ME process in FVC to reduce the encoding complexity substantially. The proposed method utilizes the correlation of motion information between parent and child nodes and skips redundant ME process—particularly, reference frame search—when the special conditions met. A few researchers identified the high probability of dependence between parent and child node, but the exploited relationship was for starting point of ME [4], [5] or for skipping search points within only quadtree-based HEVC [6]-[8]. On the contrary, the proposed method allows encoder to skips significant reference frame searches based on the strong correlation between parent and child nodes among QTBT structure. To verify the efficiency of the proposed method, experiments were conducted on top of joint exploration test model (JEM 3.1). The experimental results showed that the proposed method decreased the encoding time of motion estimation by 34% on average in comparison with JEM 3.1, maintaining a reasonable coding efficiency (less than a 0.3% BD-rate loss).

II. OVERVIEW OF A CODING UNIT ENCODING IN FVC

A coding unit in FVC is a basic unit that encodes a block by inter or intra prediction modes. The size of CU can be set from 128 x 128 at maximum to 8 x 4 or 4 x 8 at minimum depending on the depth configuration as described in JEM 3.0 algorithm description [3]. The size of CU may vary not only by quadtree coding structure used in HEVC, but also by binary tree structure that halves the width or the height of the current CU. The CTB encoding process that encodes a CU or CUs in given QTBT structure is briefly described in Figure 1 as a pseudocode description. In comparison with HEVC's quadtree structure, FVC has two more tree partitioning processes:

978-1-5386-2615-3/18/$31.00 ©2018 IEEE

horizontal and vertical binary tree partitioning. Note that FVC allows not the same depth for the quadtree of HEVC. In addition, FVC may not allow, for a certain CU size, binary tree partitioning depending on the encoding configuration.

RD_QTBT (*x*, *y*, *width*, *height*)

{

Step 1. Prediction mode selection

Do merge/skip at (x, y) with *width* and *height* and save the cost

Do inter prediction at (x, y) with *width* and *height* and save the cost

Do intra prediction at (x, y) with *width* and *height* and save the cost

Select the best cost as *costPred* and save its prediction mode

Step 2. Horizontal binary tree partitioning

Do **RD_QTBT** (x, y, width, height / 2)

Do **RD_QTBT** (x, y + height / 2, width, height / 2)

Save the cost of sub-trees as *costHor*

Step 3. Vertical binary tree partitioning

Do **RD_QTBT** (x, y, width / 2, height)

Do **RD_QTBT** (x + width / 2, y, width / 2, height)

Save the cost of sub-trees as *costVer*

Step 4. Quadtree partitioning

Do **RD_QTBT** (x, y, width / 2, height / 2)

Do **RD_QTBT** (x + width / 2, y, width / 2, height / 2)

Do **RD_QTBT** (x, y + height / 2, width / 2, height / 2)

Do **RD_QTBT** (x + width / 2, y + height / 2, width / 2, height / 2)

Save the cost of sub-trees as *costQT*

Step 5. Determination of the best mode/tree among above steps

costBest ← **min** (*costPred*, *costHor*, *costVer*, *costQT*)

Return *costBest* and associated mode/tree data

}

Fig. 1. Pseudocode of a coding tree block (CTB) encoding process in recursive QTBT structure

To compress a CU efficiently using temporal correlation of video, FVC uses various ME techniques similar with those of HEVC such as two direction searches and bi-prediction, four reference frames searching, and sub-pixel ME using DCT-based interpolation. Moreover, new techniques such as Affine ME are equipped in JEM 3.0. Overall inter prediction process for a CU is shown in Figure 2.

To easily compare the differences between HEVC and FVC, the same terminologies of HEVC are used in Figure 3 if the basic concept of them are same. Uni-L0 means that the current CU refers a block in previous frames, whereas uni-L1 means that the current CU refers a block in future frames. Each direction performs ME in three pixel levels: integer, half-pixel, and quarter-pixel. Those ME processes are generally conducted among four reference frames at maximum in P picture, or among two frames per each direction in B picture under random access configuration, or among four frames per each direction in generalized B picture under low-delay configuration. Accordingly, the computational complexity of ME processes could be raised significantly as much as the number of reference frames increases.

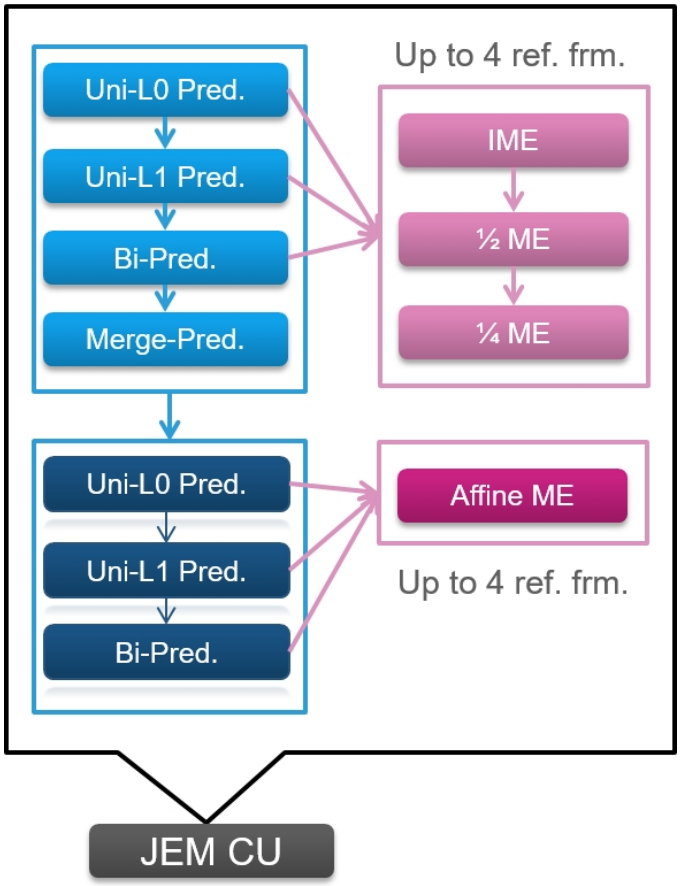


Fig. 2. Overview of inter prediction for CU in FVC

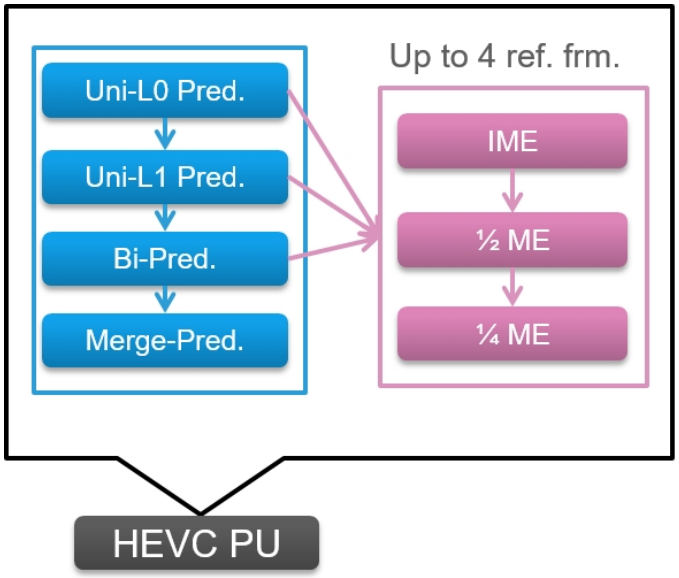


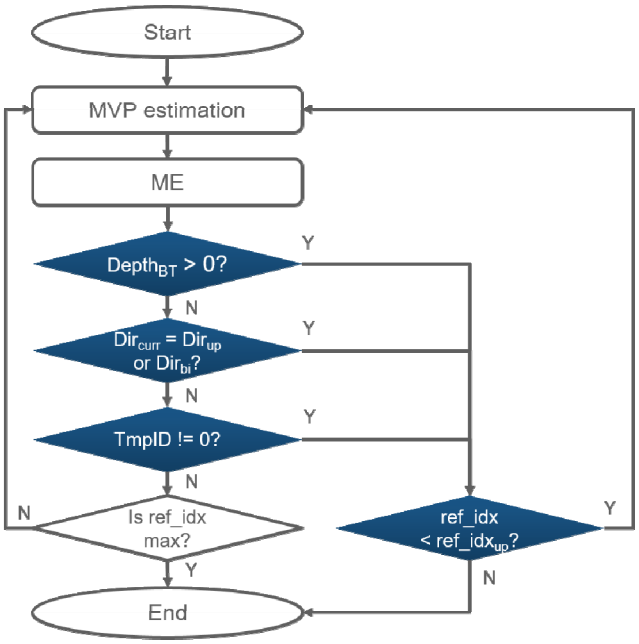
Fig. 3. Overview of inter prediction for CU in HEVC

III. PROPOSED METHOD

Based on the mechanism of CU encoding in FVC, skipping the correlated motion estimation process will efficiently decrease encoding complexity. The proposed ME skipping mechanism checks several conditions in the motion estimation processes as shown in Figure 4. To be briefly speaking, the proposed method has two sequential stages that determined to skip ME process in the certain reference frames. The first stage has four condition checking branches, and the other stage has one condition checking branch that additionally checks the room to skip ME even though the first stage is not satisfied.

The first condition in the first stage is checking whether the depth of binary tree node ($Depth_{BT}$) is larger than zero or not. According to the QTBT structure of JEM3.1, a root node of binary tree structure can split into a pair of either horizontal CUs or vertical CUs. These pairs of binary tree CUs might share special correlation character, which is assumed that these thin CUs in binary tree might tend to predict a similar motion information in nearby reference frames. If the first condition is met, the proposed method will go through the second stage; otherwise, other conditions will be checked.

The second condition in the first stage is checking whether the current prediction direction (Dir_{curr}) is the same as among parent CU's direction (whether unidirectional prediction or bidirectional). We assumed that upper CU (i.e., parent CU) may tend to have a similar characteristic in terms of prediction direction (uni-L0, uni-L1, or bidirection) with the current CU. In case that this second condition was satisfied, the next stage is ready for test. Otherwise, the next condition in the first stage will be checked.



Colored processes denotes the modified processes

Grayed processes denotes the original ME process in FVC

ref_idx ∈ {0, 1, 2, 3}

Fig. 4. Flowchart of the proposed low complexity reference frame selection method in QTBT structure for FVC

Last condition in the first stage checks whether the temporal layer of each frame (i.e., *tempID*) is not zero. Since the lowest temporal layer (*tempID* equal to zero) is encoded with the lowest quantization parameter (QP) among P or B frames, the layer is likely to be referenced with high probability, which should be encoded with high quality as much as possible for compression efficiency. In other hand, the other layer could have much room to skip ME process without losing significant coding efficiency compared to the lowest layer.

When one of the three previously condition were true, the second stage will be checked: whether the current reference frame index is smaller than the parent CU's reference index. If even the second stage were not met, then it will return to the original ME process with increased reference frame index to search motion. Until the last available reference frame is searched, the ME process will be performed.

IV. EXPERIMENTAL RESULTS

To examine the efficiency of the proposed method, the following experiments were conducted as shown in below. The experiment was deployed on a PC with Windows 7 (64-bit) operating system. The hardware specification of the PC is as follows: quad-core Intel i7 CPUs running at 4.00 GHz, with more than 16 GB random-access memory (RAM). Test video sequences were selected among common test condition (CTC) [9] recommended by JVET experts to test the performance of technical contributions on FVC.

We set JEM 3.1 as an anchor to compare the efficiency of the proposed method, and the proposed method was built on top of JEM 3.1. This experiment was conducted under low-delay encoding configuration, and four different QPs (22, 27, 32, and 37) were used. These video sequences and their encoding results are listed as shown in Table 1. BD-rate means bitrate reduction ratio in the assumption that the peak signal-to-noise ratio is equal between the anchor and the proposed method. ME time is computed as the ME time of the proposed method divided by the ME time of the anchor, and similarly, total time is computed as the total encoding time of the proposed method divided by the total encoding time of the anchor.

TABLE I. RESULT OF PROPOSED METHOD ON LOW-DELAY CONFIGURATION

Sequence		BD-rate			Time	
		Y	U	V	ME time	Total time
1920 x 1080	Kimono	0.23%	0.42%	0.35%	62%	87%
	ParkScene	0.33%	0.41%	0.16%	67%	89%
	Cactus	0.34%	0.70%	0.48%	66%	87%
832 x 480	BasketballDrill	0.23%	0.02%	-0.21%	66%	88%
	BQMall	0.24%	0.27%	0.72%	66%	87%
	PartyScene	0.48%	0.36%	0.35%	71%	91%
416	BasketballPass	0.17%	-1.15%	0.45%	68%	88%

x 240	BlowingBubbles	0.38%	0.09%	-0.29%	71%	90%
1280 x 720	FourPeople	0.20%	0.29%	1.10%	64%	84%
	Johnny	0.40%	1.28%	-1.08%	65%	84%
	KristenAndSara	0.24%	0.31%	-0.18%	65%	84%
Average		**0.29%**	**0.27%**	**0.17%**	**66%**	**87%**

As the result shows in Table 1, compared with the original JEM encoding procedure, the total encoding time of the proposed method was reduced to 87% on average, and the ME time was reduced to 66% on average. The best efficiency of the proposed method appeared at 1280 x 720 sequences, showing 84% encoding time compared to the anchor. In terms of compression performance, the proposed method only dropped the BD-rate of Y color component by 0.29% on average. Interestingly, two 416 x 240 sequences (BasketballPass and BlowingBubbles) showed a better compression efficiency in U or V component. It is assumed that the performance gain may occur by reducing bits for reference frame index since the nearer reference frame index may be selected by the proposed method.

V. CONCLUSION

In this paper, a low complexity reference frame selection method is presented by adding several branches to skip reference frame search of FVC. As shown in the experimental results, the proposed method reduced ME time and accordingly the total encoding time substantially while sustaining reasonable compression performance compared to the anchor. Since it is reported that the encoding complexity of JEM is significant in FVC standardization project, the proposed method could contribute to relieve the encoding complexity of FVC for low complexity encoding application such as mobile devices. As virtual reality video or 360-degree video may need much higher video resolution than those common test sequences, the proposed method could be useful to easily accelerate the encoding complexity of FVC.

REFERENCES

[1] Sullivan, G. J., Ohm, J. –R. Han, W. –J., and Wiegand, T., "Overview of the High Efficiency Video Coding(HEVC) Standard," IEEE Trans. Circuits Syst. Video Technol., pp.1649-1668, 2012.

[2] Sullivan, G., and Ohm, J.-R., "Meeting notes of the 3rd meeting of the Joint Video Exploration Team (JVET)," *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, JVET-C1000, 2016.

[3] Chen,J., Alshina, E., Sullivan, G. J., Ohm, J. –R., and Boyce, J., "Algorithm Description of Joint Exploration Test Model 3," *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, JVET-C1001, 2016.

[4] Park, S., and Jang, E. S., "An Efficient Motion Estimation Method for QTBT Structure in JVET Future Video Coding," *Proc. Data Compression Conference (DCC)* , 2017.

[5] Pan, Z., Lei, J., Zhang, Y., Sun, X., and Kwong, S., "Fast Motion Estimation Based on Content Property for Low-Complexity H.265/HEVC Encoder," IEEE Trans. Broadcast., pp.675-684, 2016.

[6] Park, S., and Jang, E. S., "Comments on 'Fast Motion Estimation Based on Content Property for Low-Complexity H.265/HEVC Encoder,'" IEEE Trans. Broadcast., June 2017.

[7] Nalluri, P., Alves, L. N., and Navarro, A., "Complexity reduction methods for fast motion estimation in HEVC," Signal Process.-Image Commun., pp.280-292, 2015.

[8] Yang, S., Shim, H. J., and Jeon, B., "Motion vector inheritance method for fast HEVC encoding," Proc. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2014.

[9] Suehring, K., and Li, X., "JVET common test conditions and software reference configurations," *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, JVET-B1010, 2016.