Fast Binary Tree Partition Decision in H.266/FVC Intra Coding

Ting-Lan Lin, Hui-Yu Jiang, Jing-Ya Huang, and Pao-Chi Chang

***Abstract*—The Joint Video Exploration Team (JVET) has established the latest video compression standard, Future Video Coding (FVC). However, quadtree plus binary tree (QTBT) based coding unit (CU) structure increases noticeable computational complexity in FVC intra coding. In this paper, a fast intra CU binary tree partition decision algorithm based on spatial features is proposed. The proposed method compared to JEM5.0 can save 23% encoding time with only 0.55% BDBR increment.**

I. INTRODUCTION

Future Video Coding (FVC) [1] adopts quadtree plus binary tree (QTBT) structure to split coding tree unit (CTU) into coding unit (CU). QTBT not only removes the complex hierarchical structure of CU, prediction unit (PU), and transform unit (TU) but also supports either square or rectangle coding block to adapt various local characteristics of video. Fig. 1 illustrates QTBT based CU structure. First, CTU is partitioned by a quadtree structure. The quadtree leaf nodes are further partitioned by a binary tree structure including symmetric horizontal splitting and symmetric vertical splitting. The binary tree leaf node is named as CU, which is used for prediction and transform without any further partitioning.

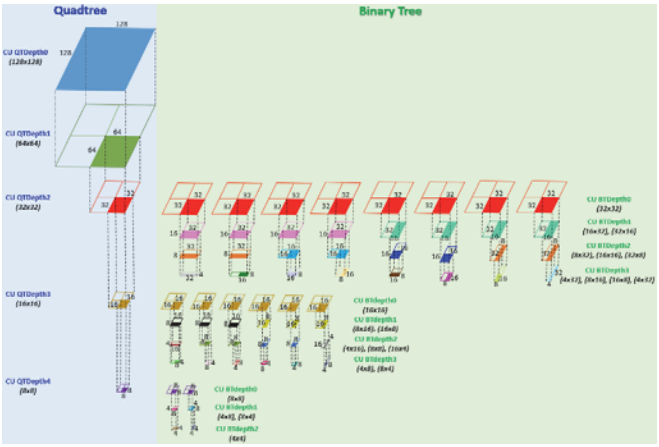

Fig. 1. QTBT-based CU structure.

II. FAST BINARY TREE PARTITION DECISION

For binary tree, we propose a fast CU partitioning strategy based on different spatial features for binary tree depth (BTDepth) and binary tree split mode (BTSplitMode).

A. Binary Tree Depth (BTDepth)

CUs with low BTDpeth are favored for reducing side information in local and smooth regions. By contrast, CUs with high BTDepth are preferred for preserving texture details in local and complex regions. Therefore, we propose using gradient variance based on Sobel operator to represent the local block complexity of BTDepth1 CUs. The gradient variance of each CU is denoted by $Variance_{GRAD}$, and defined in Eq. (1), where $GRAD$ is the gradient magnitudes of each pixel in CU, μ_1 is the mean of gradient in CU, H and W are the height and width of CU.

$$Variance_{GRAD} = \frac{1}{H \times W} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (GRAD_{i,j} - \mu_1)^2 \quad (1)$$

B. Binary Tree Split Mode (BTSplitMode)

The textures of CU are highly related to not only BTDepth but also BTSplitMode in local regions. Local region with obvious horizontal texture is favored to encode with horizontal splitting (HS). By contrast, local region with apparent vertical texture would likely encode with vertical splitting (VS). In this paper, global and local features are proposed to symbolize the direction of texture in CU. $Diff^{GRAD}_{Variance,Hor}$ and $Diff^{GRAD}_{Variance,Ver}$ are global features to represent the horizontal and vertical difference of block complexity in two sub-CUs, and calculated in Eq. (2) and Eq. (3) respectively:

$$Diff^{GRAD}_{Variance,Hor} = |Variance_{GRAD,0} - Variance_{GRAD,1}| \quad (2)$$
$$Diff^{GRAD}_{Variance,Ver} = |Variance_{GRAD,2} - Variance_{GRAD,3}| \quad (3)$$

where $Variance_{GRAD,0}$ and $Variance_{GRAD,1}$ are gradient variance in above and below sub-blocks when CU is partitioned into HS, $Variance_{GRAD,2}$ and $Variance_{GRAD,3}$ are gradient variance in left and right sub-blocks when CU is partitioned into VS.

With regard to local features, this paper denotes $Diff_{Edge,Hor}$ and $Diff_{Edge,Ver}$ as horizontal and vertical edges between the central boundary of two sub-CUs, and they are calculated in Eq. (4) and (5), where $p_{i,j}$ is the pixel value at location (i, j).

$$Diff_{Edge,Hor} = \left| \sum_{j=0}^{W-1} p_{\frac{H}{2},j} - p_{\frac{H}{2}-1,j} \right| \quad (4)$$
$$Diff_{Edge,Ver} = \left| \sum_{i=0}^{H-1} p_{i,\frac{W}{2}} - p_{i,\frac{W}{2}-1} \right| \quad (5)$$

C. Proposed Fast Binary Tree Partition Decision

For BTDepth decision, as shown in Fig. 2, a threshold $TH_{OS,BTDeth2}$ is set at BTDepth1 CU for QTDepth2 and QTDepth3 to execute only splitting into BTDepth2 (OS_BTD2)

This research was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 106-2221-E-033-010.

978-1-5386-6301-1/18/$31.00 ©2018 IEEE

on the current BTDepth1 CU with low $Variance_{GRAD}$. Threshold $TH_{ES,BTDeth3}$ is set to conduct early splitting in BTDepth3 (ES_BTD3) on the current BTDepth1 CU with high $Variance_{GRAD}$. If the $Variance_{GRAD}$ value of a BTDepth1 CU is lower than $TH_{OS,BTDeth2}$, OS_BTD2 is performed; if it is higher than $TH_{ES,BTDeth3}$, ES_BTD3 is performed. A range between two thresholds is reserved for "unsure" cases. These strategies make fast decisions with low prediction errors.

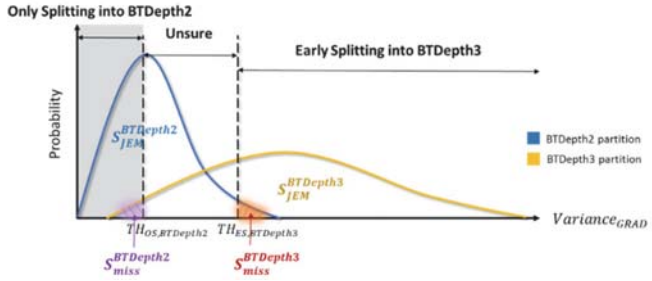

Fig. 2. Fast binary tree depth decision strategies.

In this paper, the $Variance_{GRAD}$ threshold in six training sequences are selected by adjusting the error rate due to OS_BTD2/ES_BTD3 with four QPs. Subsequently, the selected thresholds are used for fitting models.

An erroneous decision is made in the execution of OS_BTD2 when optimal CU which split into BTDepth3 ($S_{JEM}^{BTDepth3}$) is split into BTDepth2 ($S_{miss}^{BTDepth2}$). By contrast, in the execution of ES_BTD3, an erroneous decision is made when optimal CU which split into BTDepth2 ($S_{JEM}^{BTDepth2}$) is split into BTDepth3 ($S_{miss}^{BTDepth3}$). The error rates generated by OS_BTD2 and ES_BTD3 are denoted as $Error_{ES\_BTD2}$ and $Error_{ES\_BTD3}$, and they are calculated in Eq. (6) and (7) respectively.

$$Error_{OS\_BTD2}(\%) = \frac{S_{miss}^{BTDepth2}}{S_{JEM}^{BTDepth3}} \quad (6)$$

$$Error_{ES\_BTD3}(\%) = \frac{S_{miss}^{BTDepth3}}{S_{JEM}^{BTDepth2}} \quad (7)$$

We choose the error rate for $Variance_{GRAD}$ threshold being 1% can achieve good trade-off between coding performance and time saving. The adaptive threshold for fast decisions can be modeled using the thresholds selected offline from the training data with two image property parameters, image complexity and image gradient. $TH_{OS,BTDeth2}$ and $TH_{ES,BTDeth3}$ are modeled by using the two-degree polynomial functions.

After OS_BTD2, BTSplitMode decision is conducted, as shown in Fig. 3, HS and VS conditions are presented in Table I are sets to perform HS and VS. If the current CU matches HS condition, HS is conducted; if the current CU meets VS condition, VS is performed.

TABLE I. HORIZONTAL AND VERTICAL SPLITTING CONDITION.

Horizontal Splitting (HS)	Vertical Splitting (VS)
$Diff_{Variance,Hor}^{GRAD} > Diff_{Variance,Ver}^{GRAD}$ and $Diff_{Edge,Hor} > Diff_{Edge,Ver}$	$Diff_{Variance,Ver}^{GRAD} > Diff_{Variance,Hor}^{GRAD}$ and $Diff_{Edge,Ver} > Diff_{Edge,Hor}$

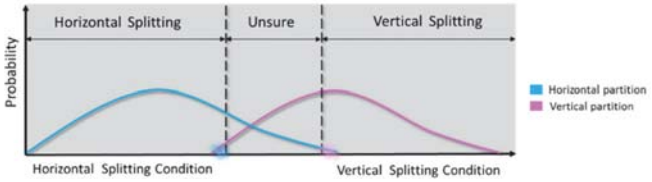

Fig. 3. Fast binary tree split mode decision strategies.

III. EXPERIMENTAL RESULTS

In the simulations, 18 sequences including six classes from A1 to E are tested. 20 frames are encoded for each sequence and the encoding configuration is Intra_main_10. The other settings are the same as the common test conditions [3] used in JVET meetings. Coding efficiency is estimated in terms of BDBR (%), and Encoding time (%) represents the time saved in comparison to JEM5.0. According to Table II, the existing algorithm [2] can save 3.6% average encoding time with a 0.023% BDBR improvement, and our proposed method can save 23% average encoding time with only 0.55% BDBR increment. It shows that our proposed method can save more encoding time with low prediction errors caused by wrong decision.

TABLE II. THE PERFORMANCE BETWEEN EXISTING ALGORITHM AND PROPOSED ALGORITHM.

Class	[2]		Proposed method	
	BDBR (%)	Encoding time (%)	BDBR (%)	Encoding time (%)
A1	-0.002	-3.400	0.329	-23.840
A2	0.048	-3.647	0.509	-25.072
B	0.005	-3.383	0.435	-23.503
C	-0.030	-4.293	0.767	-21.218
D	-0.016	-3.420	0.726	-20.504
E	-0.043	-3.145	0.576	-21.703
Average	**-0.023**	**-3.580**	**0.553**	**-22.759**

IV. CONCLUSION

In this paper, a fast intra CU binary tree partition decision in H.266/FVC based on spatial features is proposed. Adaptive threshold is derived by adjusting error rate of OS_BTD2 and ES_BTD3. The proposed algorithm compared with JEM5.0 provides 23% average time saving with only 0.55% BDBR increment.

REFERENCE

[1] J. Chen, E. Alshina, G. J. Sullivan, J. R. Ohm, and J. Boyce, "Algorithm description of Joint Exploration Test Model 4 (JEM4)," Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 3rd Meeting, Doc. JVET-C1001, Geneva, May 2016.

[2] H. Huang, S. Liu, Y. W. Huang, C. Y. Chen, and S. Lei, "AHG5: Speed-up for JEM-3.1," Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 4th Meeting, Doc. JVET-D0077, Chengdu, Oct. 2016.

[3] K. Suehring and X. Li, "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 2nd Meeting, Doc. JVET-B1010, San Diego, Feb 2016.