INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2018, VOL. 64, NO. 3, PP. 321–327
Manuscript received May 8, 2018; revised July, 2018. DOI: 10.24425/123525

Key-point Detection based Fast CU Decision for HEVC Intra Encoding

Zhe Xu, Biao Min, and Ray C.C. Cheung

***Abstract*—As the most recent video coding standard, High Efficiency Video Coding (HEVC) adopts various novel techniques, including a quad-tree based coding unit (CU) structure and additional angular modes used for intra encoding. These new techniques achieve a notable improvement in coding efficiency at the penalty of significant computational complexity increase. Thus, a fast HEVC coding algorithm is highly desirable. In this paper, we propose a fast intra CU decision algorithm for HEVC to reduce the coding complexity, mainly based on a key-point detection. A CU block is considered to have multiple gradients and is early split if corner points are detected inside the block. On the other hand, a CU block without corner points is treated to be terminated when its RD cost is also small according to statistics of the previous frames. The proposed fast algorithm achieves over 62% encoding time reduction with 3.66%, 2.82%, and 2.53% BD-Rate loss for Y, U, and V components, averagely. The experimental results show that the proposed method is efficient to fast decide CU size in HEVC intra coding, even though only static parameters are applied to all test sequences.**

***Keywords*—HEVC, Video Coding, Fast CU Decision, Corner Detection**

I. Introduction

HIGH Efficiency Video Coding (HEVC) is the successor of H.264/AVC standard with significant improvements in coding efficiency. Because of the newly adopted techniques, it can achieve 50% bit rate reduction on average with similar perceptual video quality compared with H.264/AVC [1]. However, these techniques also significantly increase the computational complexity of HEVC encoder [2].

HEVC intra encoding aims at reducing the redundancy within a single frame by taking the adjacent decoded block samples as references. Unlike macroblock (MB) structure adopted in H.264/AVC, HEVC uses a quad-tree based coding unit (CU) structure for block partitioning. A slice is partitioned into multiple coding tree units (CTUs) with the size of 64×64. CTU is the largest CU in HEVC with depth equal to 0. Each CTU can be split into four 32×32 CUs, each of which can then be split into four sub-CUs recursively to form a coding tree structure until the allowed maximum depth is reached [3]. Fig. 1 gives an example of CU partitioning in HEVC. The size 64×64 to 8×8 CUs are corresponding to depth 0-3 in quad tree structure. This CU structure allows better fleibility for block partitioning and highly increases the coding efficiency. However it also increases the computational complexity significantly. In the HEVC reference software, HM-10.0[1], search for the best block partitioning is a depth-first search method. Nearly all possible partitioning patterns are evaluated to get the optimal partitioning. It makes HEVC encoding several times more complex than H.264/AVC, which becomes a real concern. Thus, a fast algorithm is essential to reduce the complexity of intra encoding.

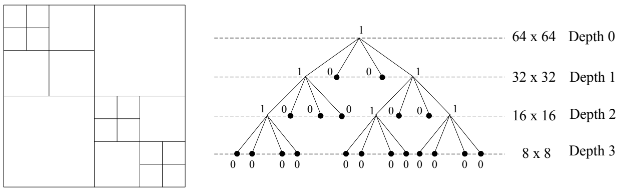


Fig. 1. An example of quad-tree based CU partitioning. Left: CU partitioning from 64×64 to 8×8; Right: corresponding quad-tree structure.

The existing reference methods can be categorized into the following two types: statistic-based methods and content-based methods. In [4] [5] [6], based on the statistical analysis of rate distortion (RD) costs in the partitioning procedure, early CU spliting and termination decision methods are proposed. In [4], the distribution of RD costs is assumed to be a Normal distribution function. A parameter called wrong decision rate, denoted as θ, is defined to obtain the dynamic threshold of RD cost in each sequence to early determine the CU termination. For early CU splitting decision, the RD cost is replaced by Hadamard cost because the RD cost at current depth is not available utile the current evaluation is finished. In [5], the RD cost is combined with its corresponding mode for early CU decision. To achieve better decision accuracy, the Bayesian decision rule is adopted with a decision loss matrix. The decision with minimum risk is selected for better balance between computational complexity reduction and coding efficiency loss. In [6], the RD cost of a CU is estimated from its relations with the Hadamard cost and is compared with the cost of its parent CU. In the mean time, the best prediction mode is searched from the coarse to the fine scale, by which only a few candidates from the 35 prediction modes are searched and the best one is selected as the candidate. Then the modes near to the selected candidate are evaluated for the final mode decision.

In [7] [8], the fast algorithm for CU size decision and prediction mode decision are combined to reduce the computational complexity. The depth level in neighboring CUs is

This work was supported by CityU Internal Grant under Project CityU 7004440 - Fast Algorithms and Efficient Architectures for High Efficient Video Coding (HEVC).

Zhe Xu, Biao Min, and Ray C.C. Cheung are with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, China (e-mail: zhexu22-c@my.cityu.edu.hk; biaomin3-c@my.cityu.edu.hk; r.cheung@cityu.edu.hk).

[1] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

adopted to constrain the current depth search range. Meanwhile, according to the RD cost of the neighboring CUs, early termination methods are proposed to decide the prediction modes and CU sizes. In [9], the hierarchical fast mode selection is proposed by restraining the search scope of intra prediction modes. The RD cost of each prediction mode indicates the gradient in the block. Based on evaluations on a subset of all prediction modes, some unlikely modes can then be skipped so the computational complexity is reduced. The gradient-based methods are also proposed in the previous works, such as [10]. The global and local edge complexities are proposed to represent the gradient in a block in different scales. At the same time, the sub-blocks are also taken into account to decide CU splitting and termination. The edge complexity in the algorithm can be represented as a metric of CU homogeneity. The threshold of the edge complexity is set by a static parameter.

In this paper, a fast HEVC intra CU decision algorithm is proposed to reduce computational complexity. Both block content and statistical analysis are adopted for CU splitting and termination decision. The gradient homogeneity in a block is represented by the corner response R. Based on Harris corner detection, a block can be categorized as a corner, edge or flat region by the corner response R. If corner points exist in a CU block, then this block is considered to be a region with strong gradients in multiple directions. Therefore, the current depth is not suitable for efficient intra prediction and the encoding procedure in this depth is early skipped for further spliting. Besides, for edge and flat blocks, it is assumed that they can be predicted efficiently by only one directional angle. The statistical feature, RD cost, is also utilized to help make early termination decision. If the current block is classified as edge or flat and the RD cost is small enough, then the CU partitioning is early terminated. Experimental results show that the proposed algorithm can achieve over 62% coding time saving on average with about 3.66% BDBR increase. The maximum time saving can be up to 80% and for certain test sequences, the coding efficiency loss is negligible. The coding efficiency can be further improved with the adaptive parameter selection in the future.

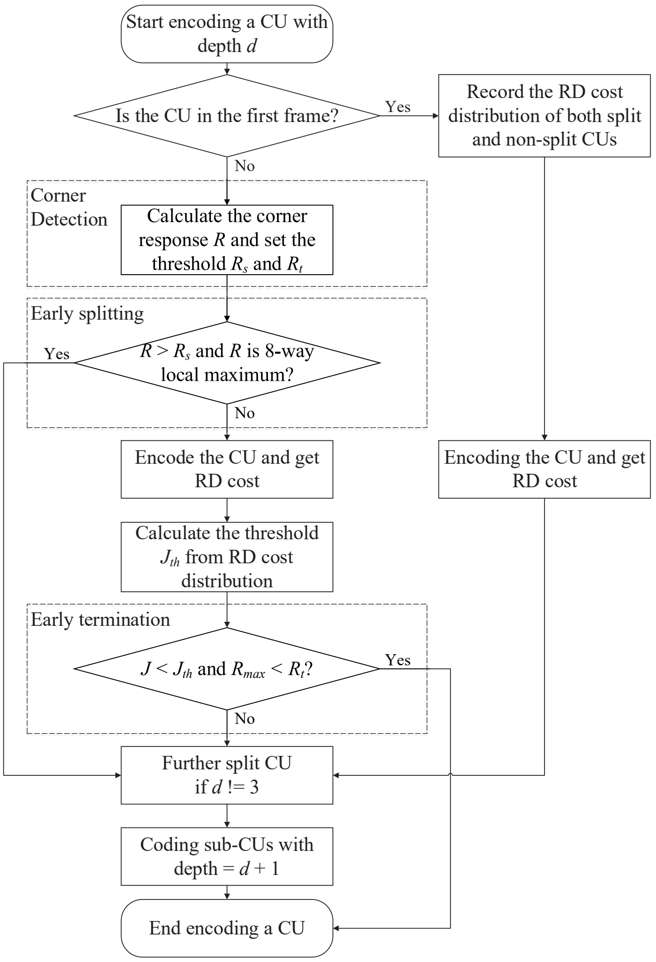
The rest of this paper is organized as follows. In Section II, the proposed fast algorithm is introduced in detail. The experimental results and comparisons with existing methods are given in Section III. Finally, Section IV gives the conclusions.

II. The Proposed Fast Algorithm

In this section, the proposed fast CU decision algorithm is presented. Firstly, we briefly introduce the HEVC rate distortion optimization (RDO) process for CU partitioning decision. In HM reference sortware, the RDO process is to find the optimal CU partitioning with minimum RD cost. The full RD cost is calculated by

$$J = SSE_{Luma} + 0.57 \times SSE_{Chroma} + \lambda \times R_{mode} \quad (1)$$

where SSE_{Luma} and SSE_{Chroma} stand the sum of squared error between the original input image block and the predicted block for luma and chroma components, respectively. λ is the Lagrangian multiplier and R_{mode} is the total coding bits cost of the intra mode for current CU block.



Start encoding a CU with depth d
Is the CU in the first frame?
Yes
Record the RD cost distribution of both split and non-split CUs
No
Corner Detection
Calculate the corner response R and set the threshold R_s and R_t
Early splitting
Yes
$R > R_s$ and R is 8-way local maximum?
No
Encode the CU and get RD cost
Encoding the CU and get RD cost
Calculate the threshold J_{th} from RD cost distribution
Early termination
$J < J_{th}$ and $R_{max} < R_t$?
Yes
No
Further split CU if d != 3
Coding sub-CUs with depth = d + 1
End encoding a CU

Fig. 2. Flowchart of the proposed fast algorithm for HEVC intra CU decision.

In HM software, there exist two processes to decide CU partitioning: CU splitting and CU pruning. CU splitting is a top-down manner to split a CU into four sub-CUs until it reaches the maximum depth. While CU pruning is a down-top manner to decide whether the CU is split or not. To determine the CU size, RD cost of the current CU J_{CU} is compared with the total RD cost of its four sub-CUs, defined as J_{split}

$$J_{split} = \sum_{i=0}^{3} J_{sub-CU_i} \quad (2)$$

If J_{split} is smaller than J_{CU}, the current CU is decided to be split and its RD cost is replaced by J_{split}. Otherwise, the CU is decided not to be split. After the exhaustive CU splitting and CU pruning process, the optimum CU partitioning in a CTU is figured out.

If the current CU is the optimal size, it is considered that the block can be predicted successfully by one intra directional mode. Otherwise, the current CU should be split into smaller sub-CUs to ensure that each sub-CU has their own reference samples and intra mode. Therefore, it is assumed that a CU can be decided to be non-split for early termination when only one major directional gradient exists. Similarly, a CU is decided to be split and the current depth evaluation is early skipped when multiple directional gradients are detected in the current CU block. In this paper, the directional gradient is represented as the intensity changes along different directions, which can be calculated by a key-point detectio algorithm called Harris corner detector [11]. The details of the proposed fast CU decision algorithm for HEVC intra encoding will be presented in the following sections. The algorithm adopts the key-point detection technique and the statistical RD cost analysis and it includes both early CU splitting and termination decision.

A. Overview of the Proposed Approach

The proposed algorithm adopts both content-based and statistic-based method. Fig. 2 shows the flowchart of the proposed algorithm. The first frame in the sequence is treated as the training frame. The RD costs of all split and non-split CUs are recorded to get the distributions. Then in the following frames, the corner response R of each pixel is calculated, which will be introduced in Section II-B. The algorithm conrains both fast CU splitting and termination decision to reduce coding complexity.

For early CU splitting decision, only the corner response R is adopted. If there exist pixles (i, j) inside the CU, whose corner response $R(i, j)$ is 8-way local maximum and satisfies

$$R(i, j) > R_s \tag{3}$$

where R_s is the threshold for splitting decision, then this CU is considered to have corner pixels within it and the corresponding magnitude of the corner response is large. Therefore, multiple gradients exist and the current CU can not be predicted correctly by only one directional angle. This type of CUs are early decided to be split and the entire encoding process at the current depth d is skipped.

For early CU termination, both statistical RD cost and corner response R are employed for decision. If the corner responses of all pixels in a CU are smaller than a threshold R_t, then the CU is considered to be an edge or flat region. This can be represented as

$$R_{max} < R_t \tag{4}$$

where R_{max} means the maximum corner responce in the CU. Then the RD cost of the current CU is considered. If the RD cost of the CU, denoted as J, is smaller than the threshold J_{th}, the current CU partitioning seach is decided to be terminated. J_{th} is calculated given a parameter called wrong decision rate, θ_S^d, and RD cost distributions obtained in the training frame. This statistical analysis will be presented further in Section II-C. The decision criterion can be represented as

$$w = \begin{cases} w_t, & J < J_{th} \ \& \ R_{max} < R_t \\ w_s, & otherwise \end{cases} \tag{5}$$

where w_t and w_s stand for early termination and regular splitting for further evaluation, respectively. The corner response indicates the gradient homogeity of the CU. If there is no large positive corner responses, it means that the current CU is considered to be homogeneous. At the same time, the RD cost represents the efficiency of current intra prediction. If the RD cost is small, current depth is already suitable. The corner response is based on the gradient detection on the pixel-level, while the RD cost is based on the whole block. So the early termination decision adopts information from both global and local scales.

The proposed fast CU decision algorithm contains early CU splitting and CU termination decision, so both homogeneous and non-homogeneous CUs can be fast encoded in an early stage with high accuracy. The next two sections will instroduce two key techniques adopted in our algorithm: Harris corner detector for key-point detection and statistical RD cost analysis, respectively.

B. Harris Corner Detector

A corner pixel in images can be defined as a center point with two or more dominant edge directions in the neighborhood. Corner detection is a method to find these corner pixels, typically used in a computer vision system for feature extraction. *Harris and Stephens* [11] proposed a corner detection approach which categories pixels into three types, flat, edge, and corner, as illustrated in Fig. 3. For flat patches, small shifts in any direction result in small intensity changes. For edge patches, a shift along the edge will result in small changes but it is opposite in the direction perpendicular to the edge. While for corner patches, all shifts will result in large intensity changes. Thus, the main idea to categorize a corner is to compute the gradient in all directions within a local neighborhood.

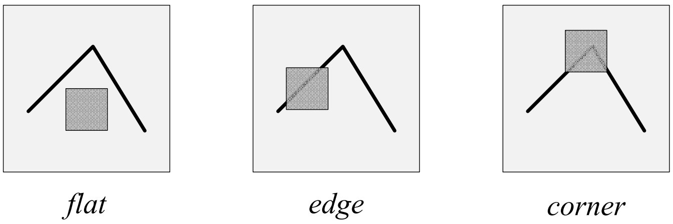


Fig. 3. Three categories in corner detection.

For a certain patch in the image I, the intensity change produced by a shift (x, y), denoted as $E(x, y)$, can be represented as

$$E(x, y) = \sum_{u,v} w(u, v)[I(u + x, v + y) - I(u, v)]^2 \tag{6}$$

where (u, v) stands for the location in the patch and $w(u, v)$ is the window function. $I(u+x, v+y)$ can then be expanded and approximated by the first order of Taylor Series

$$\begin{aligned} I(u + x, v + y) &= I(u, v) + xI_u + yI_v + O(x^2, y^2) \\ &\approx I(u, v) + xI_u + yI_v \end{aligned} \tag{7}$$

where I_u and I_v are two partial derivatives of I at location (u, v).

By substituting Equation (7) into (6), the intensity change $E(x,y)$ can be rewritten as

$$\begin{aligned} E(x,y) &\approx \sum_{u,v} w(u,v)[xI_u + yI_v]^2 \\ &= \sum_{u,v} \begin{bmatrix} x & y \end{bmatrix} w(u,v) \begin{bmatrix} I_u^2 & I_uI_v \\ I_uI_v & I_v^2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ &= \begin{bmatrix} x & y \end{bmatrix} (\sum_{u,v} w(u,v) \begin{bmatrix} I_u^2 & I_uI_v \\ I_uI_v & I_v^2 \end{bmatrix}) \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned} \tag{8}$$

In Equation (8), the middle part is a matrix describing patch patterns, denoted as *Harris matrix* A. It is rewritten as

$$\begin{aligned} A &= (\sum_{u,v} w(u,v) \begin{bmatrix} I_u^2 & I_uI_v \\ I_uI_v & I_v^2 \end{bmatrix}) \\ &= \begin{bmatrix} \overline{I_u^2} & \overline{I_uI_v} \\ \overline{I_uI_v} & \overline{I_v^2} \end{bmatrix} \end{aligned} \tag{9}$$

where $\overline{I_u^2}$, $\overline{I_v^2}$ and $\overline{I_uI_v}$ are the average values of I_u^2, I_v^2 and I_uI_v in the window area $w(u,v)$, respectively.

For the points in a flat region, the derivatives in two directions should be both small, while in a corner region, these two values should be both large relatively. Thus, the eigenvalues of metrix A, λ_1 and λ_2, are characterized as the rotationally invariant description of principal curvatures in the local region [11]. Then there are three cases needed for discussion, as illustrated in Fig. 4:

- *Flat*. If both λ_1 and λ_2 are small, which means both derivatives are small, then the windowed image region is of approximately constant intensity. The pixel is considered to be in a flat region.
- *Edge*. If one curvature is high and the other one is low, representing that only one eigenvalue is large, then only shift along one direction causes little pixel intensity change $E(x,y)$. This situation implies that the pixel is categorized into an edge region.
- *Corner*. When both eigenvalues, λ_1 and λ_2 are large, then shifts in any direction will result in large intensity change $E(x,y)$. This indicates a corner point.

Instead of computing two eigenvalues λ_1 and λ_2 of *Harris matrix* A directly, the corner response R is proposed in [11] to select isolated corner pixels and to thin the edge pixels. R is represented as

$$\begin{aligned} &R = det(A) - k \times tr^2(A) \\ &det(A) = \lambda_1 \times \lambda_2 = \overline{I_u^2} \times \overline{I_v^2} - (\overline{I_uI_v})^2 \\ &tr(A) = \lambda_1 + \lambda_2 = \overline{I_u^2} + \overline{I_v^2} \end{aligned} \tag{10}$$

where $det(A)$ and $tr(A)$ are the determinant and the trace of matrix A, respectively. k is a parameter needed to be set in advance, typically a small value. Fig. 4 clearly shows the relations between two eigenvalues and corner response R. If the pixel is in the corner region, both λ_1 and λ_2 are large. Then R is a large positive value. Similarly, if the pixel is in the edge region, the corner response R will be a large negative value. If R is a small value, the pixel is considered to be in the flat region.

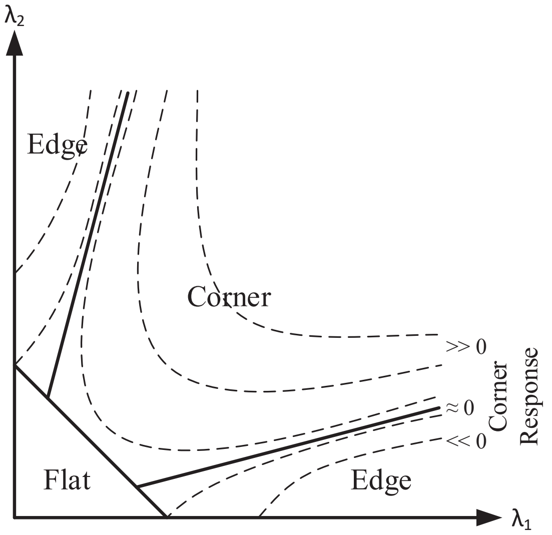


Fig. 4. Corner and edge detection. The dash lines are contour of corner response. For eigenvalues, both large λ_1 and λ_2 indicates a corner region. For corner response R, if R is a large positive value, the current pixel is assumed to be in the corner region.

C. Statistical RD Cost Analysis

According to [4], the distributions of RD costs J^d for split and non-split CUs are assumed to follow a Normal distribution function, $N(\mu_w^d, \sigma_w^d)$. For clear explanation, two classes $\{S, \bar{S}\}$ are classified, S stands for spliting CUs and $\bar{S}$ is the class containing non-split CUs. The distribution can be represented as

$$p(J^d|w^d) = \frac{1}{\sigma_w^d\sqrt{2\pi}} \times exp(-\frac{(J^d - \mu_w^d)^2}{2(\sigma_w^d)^2}) \tag{11}$$

where the superscript d is the depth of current CU and w stands for different kinds of CUs, $w \in \{S, \bar{S}\}$. $p(J^d|w^d)$ is the conditional probability density function of the RD cost J^d. μ_w^d and σ_w^d are the mean value and standard deviation of the distribution, respectively.

In general, small RD cost means that the prediction is accurate with small residuals. Thus it is considered that the mean value of RD costs of the non-split CUs, $\mu_{\bar{S}}^d$ is smaller than the mean RD cost of split CUs μ_S^d. In other word, the CU with smaller RD cost are more likely to be the non-split CU. In the proposed algorithm, the RD cost is adopted as a feature for early termination decision. If the RD cost of current CU is smaller than a threshold J_{th}^d, then the CU partitioning may be early terminated and the following depths are skipped.

Noted that when given a threshold J_{th}^d for early termination decision, split CUs whose RD costs are also smaller than J_{th}^d may be wrongly classified as the non-split CUs and the coding efficiency is affected. To better balance the computational complexity reduction and coding efficiency loss, a parameter called wrong decision rate for termination decision, θ_S^d, is used to classify split and non-split CUs. θ_S^d stands for the ratio of the split CUs to be wrongly treated as non-split CUs and is defined as

$$\theta_S^d = \int_0^{J_{th}^d} p(J^d|S^d)dJ^d \tag{12}$$

TABLE I
EXAMPLES OF J_{th}^d VALUES FOR DIFFERENT SEQUENCES GIVEN θ_S^d AND $\theta_{\bar{S}}^d$. (QP=32, DEPTH=1)

class	sequence	$\theta_{\bar{S}}^d$				θ_S^d			
		0.6	0.7	0.8	0.9	0.2	0.3	0.4	0.5
B	ParkScene	9699	12598	15913	19739	9095	11003	12834	14677
	BQTerrace	10880	14282	18312	25285	14193	16943	19887	23055
C	PartyScene	13112	15222	18956	23140	18652	22484	26070	30091
	BQMall	6419	7382	9338	13894	10973	13047	15197	18144
D	BasketballPass	6797	7788	8876	11685	8059	9777	11843	13700
	RaceHorses	8900	14822	23379	32591	13326	16475	18533	21368
	BQSquare	7042	12333	18907	27128	17120	23021	28041	33714

Similarly, the recall rate of early termination decision $\theta_{\bar{S}}^d$, standing the ratio of total non-split CUs that can be early terminated, is defined as

$$\theta_{\bar{S}}^d = \int_0^{J_{th}^d} p(J^d|\bar{S}^d)dJ^d \tag{13}$$

Because $p(J^d|S^d)$ and $p(J^d|\bar{S}^d)$ are assumed to follow the Normal distribution as discussed before, the threshold can be rewritten as

$$J_{th}^d = \sigma_S^d \times \phi^{-1}(\theta_S^d) + \mu_S^d \tag{14}$$

or

$$J_{th}^d = \sigma_{\bar{S}}^d \times \phi^{-1}(\theta_{\bar{S}}^d) + \mu_{\bar{S}}^d \tag{15}$$

To figure out the relations between J_{th}^d and two parameters, σ_S^d and $\sigma_{\bar{S}}^d$, the J_{th}^d values are calculated for the first frame of different sequences. The results are shown in Table I. It can be seen that the RD costs of non-split CUs is generally smaller than those of split CUs and the threshold J_{th}^d varies in different sequences. So it is hard to set a static threshold J_{th}^d to distinguish the non-split CUs for different sequences. Instead, in our proposed algorithm, the static wrong decision rate θ_S^d is set and the threshold J_{th}^d is calculated from θ_S^d. The first frame of each sequence is set as the training frame to obtain the distribution of $p(J^d|S^d)$ and $p(J^d|\bar{S}^d)$. In the following frames, if the RD cost of a CU is smaller than J_{th}^d, the current prediction is assumed to be efficient and the CU will be further tested in the early termination decision process.

In this paper, the wrong decision rate θ_S^d is set as 0.4 for all depths, which means about 40% split CUs may be wrongly classified just based on the statistical RD cost analysis. The CUs would be treated as the non-split CUs candidates if their costs are smaller than the threshold. Table I shows that when θ_S^d is 0.4, the recall $\theta_{\bar{S}}^d$ will be over 0.7, which means most non-split CUs can be early terminated to reduce encoding time. To increase the accuracy, the corner detection method described in the previous section is also taken into consideration for early termination decision, as already introduced in Section II-A.

The proposed algorithm performs well for both homogeneous and non-homogeneous CUs. Next section will show the experimental results and comparisons with previous works.

III. EXPERIMENTAL RESULTS

A. Test Conditions

In this section, the coding efficiency and complexity reduction of the proposed algorithm are evaluated in detail. The proposed algorithm is implemented in HEVC test software HM-10.0 and tested under JCT-VC common test conditions [12]. Test conditions adopted in this paper are shown as follows:

1) The test software runs on the Intel Xeon X5657 CPU @ 3.07GHz with 12GB memory.
2) HEVC main profile is used. The performance is tested by strictly following the configurations set in the file *encode_intra_main.cfg*.
3) Four QPs: 22, 27, 32 and 37 and All Intra (AI) configuration are used for evaluation.
4) The test sequences include six classes. Totally 22 sequences are selected.
5) For early splitting decision, the thresholds R_s are set as 4×10^9, 8×10^9, 1.2×10^{10}, and 1.6×10^{10} for QP 22, 27, 32, and 37, respectively.
6) For early termination decision, θ_S^d is set as 0.4 for all depths and the threshold for corner response R_t is set to be 1×10^9.

The coding efficiency is evaluated by the BD bit rate (BDBR) using the Bjontegaard's method [13]. The encoding time is evaluated as the average time under four QPs. Then the computational complexity reduction is measured by the time saving (TS). TS is defined as

$$TS = \frac{T_{org} - T_{pro}}{T_{org}} \times 100\% \tag{16}$$

where T_{org} is the coding time of the original HM-10.0 and T_{pro} stands for computational complexity of the proposed CU decision algorithm.

B. Results and Comparisons

Table II shows the coding efficiency and complexity reduction of the proposed algorithm for AI configuration, including class A to class F. BDBR columns show the BDBR increase for three different components and TS column is the time saving calculated by Equation (16). The results show that over 62% coding complexity is reduced with about 3.6%, 2.8% and 2.5% BDBR increase for Y, U, V components under the selected thresholds, averagely. In addition, the maximum coding saving is near to 80% when dealing with *SlideShow*

TABLE II
THE PERFORMANCES EVALUATIONS OF THE PROPOSED ALGORITHM COMPARED WITH HM10.0 FOR AI CONFIGURATION.

Class	Sequences	TS (%)	BDBR-Y (%)	BDBR-U (%)	BDBR-V (%)
A	PeopleOnStreet	65	4.8	3.6	2.3
	Traffic	65	4.8	2.6	2.4
B	Kimono	56	3.4	3.0	2.3
	ParkScene	57	4.0	4.9	2.8
	BQTerrace	59	4.5	2.5	1.9
	Cactus	57	3.9	2.6	2.9
	BasketballDrive	58	1.9	0.2	0.2
C	PartyScene	62	1.9	1.4	1.3
	BQMall	65	4.0	2.5	2.4
	BasketballDrill	55	2.9	1.9	1.8
	RaceHorses	64	4.4	4.7	4.5
D	BQSquare	61	2.0	1.1	1.2
	RaceHorses	57	3.5	3.6	3.8
	BlowingBubbles	54	1.8	1.4	1.4
	BasketballPass	63	2.5	1.7	2.2
E	Johnny	67	6.0	5.5	5.2
	KristenAndSara	69	5.3	5.2	4.7
	FourPeople	67	5.3	3.6	2.8
F	BasketballDrillText	55	2.8	2.1	1.9
	ChinaSpeed	67	2.9	1.1	1.3
	SlideEditing	67	2.2	1.6	1.5
	SlideShow	79	5.8	4.9	4.9
Average		**62.2**	**3.66**	**2.82**	**2.53**

TABLE III
THE PERFORMANCE COMPARISON WITH EXISTING ALGORITHMS FOR AI CONFIGURATION

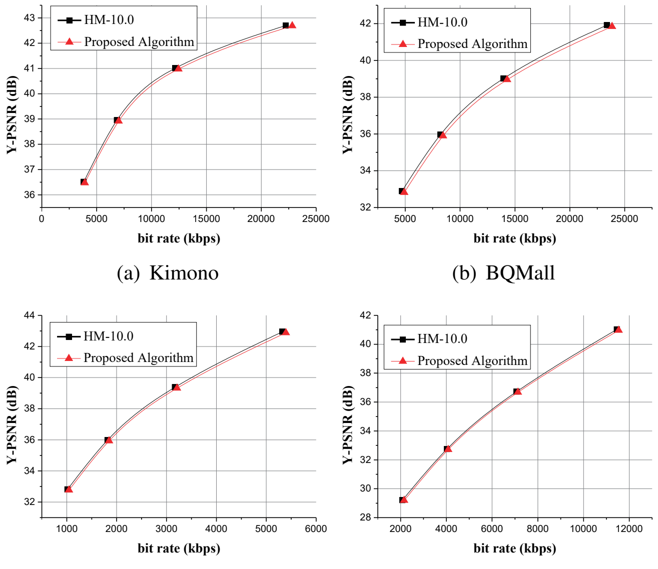
class	sequence	proposed		Cho et al. [4]		Liu et al. [14]		Lim et al. [15]	
		BDBR-Y (%)	TS (%)	BDBR-Y (%)	TS (%)	BDBR-Y (%)	TS (%)	BDBR-Y (%)	TS (%)
A	PeopleOnStreet	4.8	65	1.0	44	4.1	63	2.7	58
	Traffic	4.8	65	1.2	54	3.4	60	1.4	56
B	Kimono	3.4	56	1.2	65	1.0	64	0.3	50
	ParkScene	4.0	57	0.9	47	2.5	60	0.1	51
	BQTerrace	4.5	59	1.0	60	3.3	61	1.4	55
	Cactus	3.9	57	1.1	48	3.5	60	1.4	53
	BasketballDrive	1.9	58	2.0	61	3.8	65	1.2	51
C	PartyScene	1.9	62	0.3	46	3.0	51	1.5	55
	BQMall	4.0	65	1.0	50	4.0	57	2.8	56
	BasketballDrill	2.9	55	0.6	53	5.9	56	3.5	55
	RaceHorses	4.4	64	1.4	64	3.0	55	1.6	54
D	BQSquare	2.0	61	0.4	54	1.8	51	3.8	52
	RaceHorses	3.5	57	1.3	54	3.8	55	2.8	53
	BlowingBubbles	1.8	54	0.2	49	3.9	55	1.5	52
	BasketballPass	2.5	63	0.9	50	3.9	60	2.8	54
E	Johnny	6.0	67	2.3	65	5.6	72	2.8	53
	KristenAndSara	5.3	69	2.7	70	4.2	68	4.4	55
	FourPeople	5.3	67	0.9	51	4.8	63	3.1	57
Average		**3.7**	**61.2**	**1.1**	**54.7**	**3.6**	**59.8**	**2.2**	**53.9**

sequence. This means the proposed algorithm can significantly reduce computational complexity and the coding efficiency loss is still acceptable.

To evaluate the RD performance of the proposed algorithm, RD curves of the original HM reference software and proposed algorithm are shown in Fig. 5, where only Y component is given. Four sequences with different resolutions, Kimono, BQMall, BasketballPass and BlowingBubbles are selected for evaluation. Fig. 5 shows that the proposed algorithm achieves RD performance close to that of the original HM reference software for different QP configurations and sequences. Therefore, the RD performance degradation of the proposed algorithm is small.

To further evaluate the performance of the proposed algorithm, experimental results are compared with those in previous works. Three algorithms, Cho *et al.*'s [4], Liu *et al.*'s [14] and Lim *et al.*'s [15] are selected for detailed comparison. HM-6.0, HM-12.0 and HM-14.0 are employed in [4], [14], [15], respectively. According to [16], coding results for different HM versions are similar for HM does not involve significant changes among these versions. Therefore the comparison with other works under different HM versions is acceptable. Table. III gives the performance comparisons in terms of BDBR increase and time saving. Results of Class F are not provided in [4] and [15] so only Class A to E are presented for fair comparison. Results show that the major advantage of the proposed algorithm is that it can achieve more complexity reductions compared with other works.

In [4], RD cost is the major feature for early CU splitting and termination decision. Our algorithm can reduce about



(a) Kimono

(b) BQMall

(c) BasketballPass

(d) BlowingBubbles

Fig. 5. RD curves of four individual sequences.

6.5% more coding complexity. Compared with [15], the proposed algorithm can also save more than 7% coding time while the BDBR does not increase much. [14] proposed a CNN-based fast CU partitioning method. Our result has almost the same BDBR increase with [14] while ours reduces more encoding time. Generally, our performance is slightly better than [14].

IV. Conclusions

In this paper, a fast CU size decision algorithm for HEVC intra encoding is proposed, with both content and statistic analysis adopted. The block partitioning is treated as the detection of singular gradient or multiple gradients in the block, using Harris corner detection method. If a block is classified as flat or edge region and the RD cost is smaller than the threshold, it is decided to be non-split because it is assumed that current depth is already suitable to effciently predict the current block. It is indicated that multiple gradients exist if the corner pixels are detected. As a result, the block is early decided to be split and the search for the optimal encoding mode will be skipped in the current depth. The experimental results shwo that the proposed algorithm is effective in terms of complexity reduction achieving over 62% time saving, which surpasses the existing works. From the prospective of the coding efficiency, the BDBR increase is acceptable, while it is especially small for certain sequences. The performance in terms of coding efficiency could be improved with the adaptive parameters for different sequence contents in the future work.

References

[1] G. Sullivan, J. 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, 2012.

[2] 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, 2012.

[3] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1792–1801, 2012.

[4] S. Cho and M. Kim, "Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 9, pp. 1555–1564, Sept 2013.

[5] Z. Xu, B. Min, and R. C. Cheung, "A fast inter cu decision algorithm for hevc," *Signal Processing: Image Communication*, vol. 60, pp. 211–223, 2018.

[6] H. Zhang and Z. Ma, "Fast intra mode decision for high efficiency video coding (hevc)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 660–668, April 2014.

[7] L. Shen, Z. Zhang, and P. An, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 207–213, 2013.

[8] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An effective CU size decision method for HEVC encoders," *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 465–470, Feb 2013.

[9] J. Siast, J. Stankowski, and M. Domański, "Hierarchical fast selection of intraframe prediction mode in hevc," *International Journal of Electronics and Telecommunications*, vol. 62, no. 2, pp. 147–151, 2016.

[10] B. Min and R. C. C. Cheung, "A fast cu size decision algorithm for the hevc intra encoder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 892–896, May 2015.

[11] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Citeseer, 1988, p. 50.

[12] F. Bossen, "Common test conditions and software reference configuration," *document JCTVC-L1100, Geneva*, Jan. 2013.

[13] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves." *Document VCEG-M33*, April, 2001.

[14] Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, and D. Wang, "Cu partition mode decision for hevc hardwired intra encoder using convolution neural network," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5088–5103, Nov 2016.

[15] K. Lim, J. Lee, S. Kim, and S. Lee, "Fast pu skip and split termination algorithm for hevc intra prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 8, pp. 1335–1346, Aug 2015.

[16] T. Zhang, M. T. Sun, D. Zhao, and W. Gao, "Fast intra-mode and cu size decision for hevc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1714–1726, Aug 2017.