The following publication C. -H. Fu, Y. -L. Chan, H. -B. Zhang, S. H. Tsang and M. -T. Xu, "Efficient Depth Intra Frame Coding in 3D-HEVC by Corner Points," in IEEE Transactions on Image Processing, vol. 30, pp. 1608-1622, 2021 is available at https://dx.doi.org/10.1109/TIP.2020.3046866

Efficient Depth Intra Frame Coding in 3D-HEVC by Corner Points

Chang-Hong Fu, *Member IEEE*, Yui-Lam Chan*, *Member IEEE*, Hong-Bin Zhang, Sik-Ho Tsang, *Member, IEEE,* and MengTing Xu

Abstract— **To improve the coding performance of depth maps, 3D-HEVC includes several new depth intra coding tools at the expense of increased complexity due to a flexible quadtree Coding Unit/Prediction Unit (CU/PU) partitioning structure and a huge number of intra mode candidates. Compared to natural images, depth maps contain large plain regions surrounded by sharp edges at the object boundaries. Our observation finds that the features proposed in the literature either speed up the CU/PU size decision or intra mode decision and they are also difficult to make proper predictions for CUs/PUs with the multi-directional edges in depth maps. In this work, we reveal that the CUs with multi-directional edges are highly correlated with the distribution of corner points (CPs) in the depth map. CP is proposed as a good feature that can guide to split the CUs with multi-directional edges into smaller units until only single directional edge remains. This smaller unit can then be well predicted by the conventional intra mode. Besides, a fast intra mode decision is also proposed for non-CP PUs, which prunes the conventional HEVC intra modes, skips the depth modeling mode decision, and early determines segment-wise depth coding. Furthermore, a two-step adaptive corner point selection technique is designed to make the proposed algorithm adaptive to frame content and quantization parameters, with the capability of providing the flexible tradeoff between the synthesized view quality and complexity. Simulation results show that the proposed algorithm can provide about 66% time reduction of the 3D-HEVC intra encoder without incurring noticeable performance degradation for synthesized views and it also outperforms the previous state-of-the-art algorithms in term of time reduction and ΔBDBR.**

Index Terms—**3D-HEVC, Depth map, Depth modeling mode, Multi-view video plus depth, Intra coding, Corner detection**

I. Introduction

AFTER achieving a great success in the film industry, 3D video also becomes increasingly interesting for home entertainment. The state-of-the-art 3D video is multi-view video plus depth (MVD) [1] due to its capability of providing virtual view generation via a Depth Image-Based Rendering (DIBR) technique. MVD contains several views and their corresponding depth maps. The most challenging issue is how to compress depth maps efficiently [2]. To address this problem, the depth enhanced 3D video coding standard - 3D-HEVC was launched by Joint Collaborative Team on 3D Video Coding Extension (JVT-3V) [3][4]. The test model of 3D-HEVC is known as HTM, which uses the HEVC coding framework [5] with the new depth map coding tools such as Depth Modeling Modes (DMM) [6], Depth Intra Skip (DIS) [7], Segment-wise Depth Coding (SDC) [8], and View Synthesized Optimization (VSO) [9]. 3D-HEVC inherits the quadtree-based Coding Tree Unit (CTU) structure [10] from HEVC. CTU is a basic unit and it is also called Largest Coding Unit (LCU). LCU is divided into several Coded Units (CUs). Each CU is further divided into several regions called Prediction Units (PU) for intra mode prediction.

Conventional HEVC Intra Modes (CHIM) [11] consist of planar, DC and 33 angular modes as illustrated in Fig. 1(a). These modes are inherited from texture coding. Besides CHIM, DMM in Fig. 1(b) are introduced as new intra modes for better representation of edges in depth maps. On the other hand, DIS is another customized intra prediction mode using only one neighboring reference pixel to describe the extremely homogenous region in depth maps.

SDC is an alternative residual coding technique in 3D-HEVC. It allows using only a Constant Pixel Value (CPV) to represent each segment instead of Quantized Discrete Cosine Transform (QDCT) coefficients. Generally, depth maps have noise, which makes the CPV inaccurate to represent the current block. Hence, an offset around the original CPV [12] (five candidates with offsets: 0, -1, 1, 2, -2) are evaluated to provide more accurate description. In CHIM, PU is considered as one segment and there are two segments in DMM. Since depth maps are not viewed directly, a new Rate-Distortion Optimization (RDO) cost function referred to as VSO [9] is adopted in 3D-HEVC by considering both of the synthesized view and depth map. The way to generate the synthesized view distortion can be classified into two categories: model-based VSO [13] and rendering-based VSO [14]. The latter one calculates the view synthesis distortion with higher accuracy and more complexity.

When these new depth map coding tools are integrated with the well-known quadtree CU/PU partitioning structure of HEVC, 3D-HEVC yields significant coding efficiency and provides the outstanding perceptual quality of synthesized views [3]. Simultaneously, the coding complexity is also increased drastically. Complexity analysis of 3D-HEVC have shown that depth intra mode decision and CU/PU size decision always take up a particularly high percentage of the encoding

This work is supported in part by a grant from the Research Grants Council of the HKSAR, China (Grant No. PolyU 152112/17E), the Fundamental Research Funds for the Central Universities (No. 30917011319), and in part by National Natural Science Foundation of China (No. 61301109).

C.-H. Fu and H.-B. Zhang visited the Hong Kong Polytechnic University and conducted the research work of this paper there. They and M.-T. Xu are with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, 210019 CHINA. (e-mail: enchfu@njust.edu.cn).

Y.-L. Chan* and S.-H. Tsang are with the Department of Electronic and Information Engineering, the Hong Kong Polytechnic University, Hong Kong (e-mail: enylchan@polyu.edu.hk*).

time. Therefore, some research works [15]-[41]were proposed to solve this problem, which will be thoroughly described in Section II.C.

In this work, we propose to use a two-dimensional gradient feature - Corner Points (CPs) to accelerate the CU/PU size decision and intra mode selection simultaneously. The difference between our contributions and the related schemes is to explore a good feature, CPs, for describing local complexity of blocks that could not be well predicted by the existing prediction mode. CPs have unique characteristics that can pave the way for splitting the CU with multi-directional edges into smaller sub-CUs until they all have only single directional edges. By taking advantage of the distribution of CPs, we are the first to introduce a good feature for guiding both CU/PU decision and intra mode decision, while most previous methods only focus on one of them. The rest of this paper is organized as follows. In Section II, the depth intra coding in 3D-HEVC and its fast algorithms are described. Section III shows the motivation of using two-dimensional gradient feature CP for fast depth coding. The proposed depth intra coding algorithms based on the distribution of CP are then presented, and they include Quadtree-depth Limited Strategy (QDLS), PU Partition Decision (PUD) and Fast Intra Mode Decision (FIMD). The two-step adaptive corner point selection scheme is also discussed in this section. The comparative simulation results of the proposed algorithm and state-of-the-art schemes are presented in Section IV. Finally, our conclusions are given in Section V.

II. Depth Intra Coding and its Fast algorithms in 3D-HEVC

The complexity of depth intra coding mainly springs from depth intra mode decision with tremendous candidates, and the CU/PU size decision with flexible quadtree structure.

A. *Depth Intra Mode Decision*

The implementation of intra mode decision in the current 3D-HEVC is performed for each PU, and is summarized in the following five steps:

Step 1 (DIS): Calculate the Rate-Distortion (RD) costs with two different candidate pixels for DIS using the rendering-based VSO.

Step 2 (Rough mode decision, RMD [35]): A few of effective CHIM [11] (3 or 8, for different PU sizes) are selected by the model-based VSO to form a set of Rough Modes: RM=$\{rm_0, rm_1, \ldots\}$. Then, a set of Most Probable Modes: MPM=$\{mpm_0, mpm_1, mpm_2\}$ predicted from the left and top neighboring blocks are also appended in the set. After removing the overlapping modes, the final modes are sorted by the RD cost based on the model-based VSO in the final set

$$\text{CHIM}=\{\text{RM}\cup\text{MPM}\} \quad (1)$$

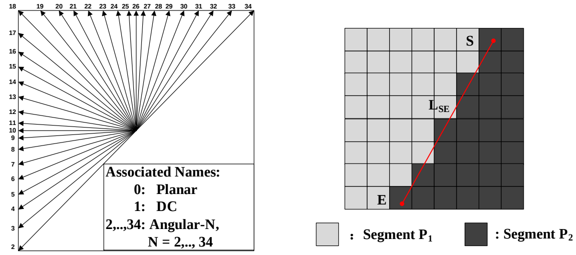
Step 3 (DMM decision): The optimal wedgelet pattern is sought for DMM by the model-based VSO from numerous candidates in

$$\text{DMM}=\{dmm_1, dmm_2, \ldots\} \quad (2)$$

and is also added into the candidate list.

Step 4 (QDCT decision): The rendering-based VSO calculations of all selective candidates are performed with the traditional QDCT residual coding.

Step 5 (SDC decision): The residual coding by SDC is checked



(a) (b)

Fig. 1. Depth intra mode in 3D-HEVC, (a) 35 HEVC traditional intra

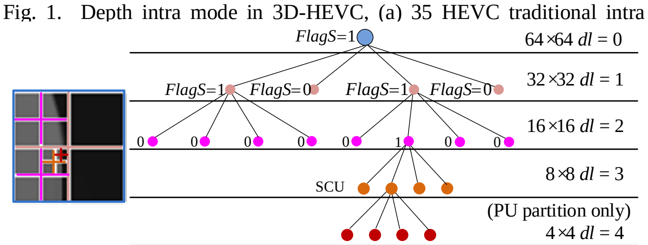


Fig. 2. Example of the quadtree coding structure and its block partitions for a given LCU.

for five offset candidates in the set of SDC as follows using the rendering-based VSO.

$$\text{SDC}=\{-2, -1, 0, 1, 2\} \quad (3)$$

Step 6: The results from Step 4 and Step 5 are compared to select the best intra mode and residual coding mode (QDCT or SDC). The corresponding cost is further compared with the DIS cost from Step 1 to obtain the final optimal intra mode.

B. CU/PU Size Decision

In the quadtree coding structure [10], a depth map is split into non-overlapping LCU that can be recursively partitioned into four equal smaller CUs. For each partitioning, the depth level (dl) is increased by 1, as shown in Fig. 2, until the quadtree reaches the smallest CU (SCU, size 8x8 and dl=3). The quadtree splitting process is determined by comparing the RD cost of the current CU and the total RD cost of its four sub-CUs as follows:

$$FlagS(blk_{dl})=\begin{cases}1, & \text{if } J_{VSO}(blk_{dl})>\sum_{k=0}^{3}J_{VSO}(blk_{dl+1}^{k})\\ 0, & \text{if } J_{VSO}(blk_{dl})\leq\sum_{k=0}^{3}J_{VSO}(blk_{dl+1}^{k})\end{cases} \quad (4)$$

where blk_{dl} is the CU at the current dl, blk_{dl+1}^{k} is the sub-CU at the next level dl+1 , k is the index of sub-CU and $FlagS(blk_{dl})$ is the splitting flag to indicate whether blk_{dl} is split ($FlagS(blk_{dl})$=1) or non-split ($FlagS(blk_{dl})$=1). As mentioned in Section I, the VSO cost $J_{VSO}(blk_{dl})$ replaces the traditional RD cost in depth map coding, which is computed by the distortion $D_{VSO}(blk_{dl})$ plus the Langrangian multiplier λ times the coding rate $R(blk_{dl})$ as

$$J_{VSO}(blk_{dl}) = D_{VSO}(blk_{dl}) + \lambda \times R(blk_{dl}) \quad (5)$$

In the next step, each CU is divided into PUs for intra-prediction process as the steps in Section II-A. It is noted that for dl from 0 to 2, PU is the same as CU. This type is named as SIZE_2N×2N, and the split process in (4) and (5) is considered as CU size decision. When dl equals to 3, there are two types of PU. One is still SIZE_2N×2N, and the other is SIZE_N×N. In that case, CU is further split into four equally sized PUs (4×4), which is the PU size decision. It is noted that blk_{dl} stands for PU in intra mode prediction, while blk_{dl} denotes CU or PU for different depth levels in CU/PU partition. Fig. 2 shows an example of the final quadtree and its CU/PU partitions.

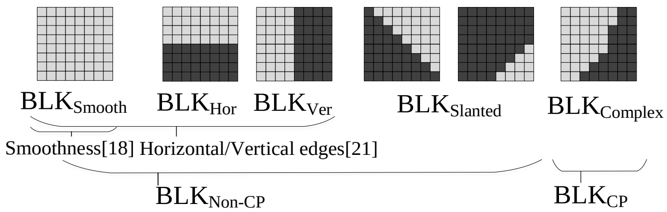


Fig. 3. Features to classify blocks for fast depth intra mode decision.

C. Fast Depth Intra Coding Algorithms

Research work on expediting depth intra coding for 3D-HEVC has been suggested in [15]-[41]. In [39], the depth map is enhanced before coding, which is not discussed in this paper. The remaining algorithms include the fast decisions on DMM, RMD, SDC, and/or CU partition.

At the early development of the 3D-HEVC standardization, DMM decision (Step 3 in Section II.A) is the most time-consuming process in depth intra coding due to a huge number of candidates. Therefore, algorithms in [15]-[17], [31]-[32] were proposed to reduce unnecessary DMM candidate list in (2). In addition, various DMM skipping schemes were investigated in [18]-[23] to skip DMM decision (i.e., the whole DMM candidate list in (2)) in PUs where the best candidate in CHIM of (1) have already predicted well enough. For illustration, we construct a set ψ of the PUs that have already well predicted by CHIM. Then the algorithms in [18]-[23] determine whether the current PU, blk_{dl}, belong to ψ using different features. If $blk_{dl} \in \psi$, it supposes to be well predicted by CHIM, and DMM can be skipped which is formulated as

$$\mathrm{DMM}(blk_{dl}) = \begin{cases} \text{DMM in (2)}, & \text{if } blk_{dl} \notin \psi \\ \emptyset, & \text{if } blk_{dl} \in \psi \end{cases} \quad (6)$$

In [18], the pixel variance of blk_{dl} is used to measure its smoothness. If the variance of blk_{dl} is smaller than a threshold and the best mode of the best rough mode is Planar mode, $blk_{dl} \in \mathrm{BLK_{Smooth}}$, where $\mathrm{BLK_{Smooth}}$ is a set of the smooth PUs, as shown in Fig. 3. Obviously, blk_{dl} in $\mathrm{BLK_{Smooth}}$ can be well predicted by smooth modes such as Planar, DC, or DIS. As result, ψ in (6) can be set as:

$$\psi = \mathrm{BLK_{Smooth}} \quad (7)$$

In the latest version of the test model [36], the algorithm in [18] has been adopted in 3D-HEVC reference software. The algorithm in [21] further identifies sets of the PUs containing horizontal and vertical edges, and the sets are denoted by $\mathrm{BLK_{Hor}}$ and $\mathrm{BLK_{Ver}}$ in Fig. 3, respectively. If $blk_{dl} \in \{\mathrm{BLK_{Hor}}, \mathrm{BLK_{Ver}}\}$, it can be predicted well by the modes of Angular-10 and Angular-26 in CHIM, as shown in Fig. 1(a). In this case, $\mathrm{BLK_{Hor}}$ and $\mathrm{BLK_{Ver}}$ can be included in ψ.

$$\psi = \{\mathrm{BLK_{Smooth}}, \mathrm{BLK_{Hor}}, \mathrm{BLK_{Ver}}\} \quad (8)$$

The features of horizontal and vertical edges help to skip DMM in both smooth and simple edge blocks, and the complexity of DMM decision was reduced significantly.

Then, the development of fast algorithms turned to the RMD (Step 2) and SDC decision (Step 5) [24]-[28], [33]-[34], [40]-[41]. The work in Park [24] skips DMM and SDC decision by an RD cost based termination method. The studies in [25]-[26] speed up the process of SDC decision by comparing the RD cost of prior checked modes. Our previous work jointly reduces the number of HEVC intra modes and DMMs by low-complexity RD cost in [27], and the reference pixels of intra prediction in [28]. Recent work in [33] reduces the number of candidates based on the ranking of RMD and MPM. The idea is similar with [27]. In [34], besides utilizing the rough cost from RMD, a low-complexity VSO metric is designed for DMM. In [38], the depth-texture correlation and edge information are jointly used for a fast depth mode decision. In [40], when the corresponding texture CU is encoded by Skip mode, the depth mode is also encoded in Skip or DIS mode. [41] is a model-level scheme for accelerating both DMM and RMD. All these methods work on the mode decision part in Section II.A.

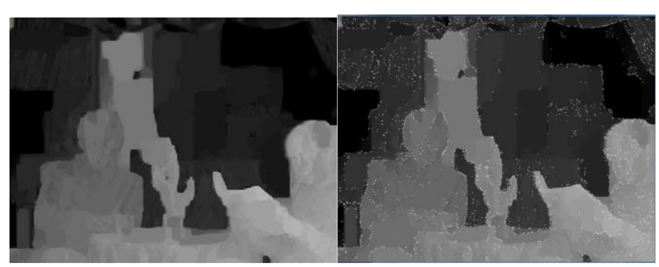
TABLE I
SUMMARY OF PREVIOUS WORKS FOR DEPTH INTRA CODING

Method	CU	PU	RMD	DMM	SDC
[15]-[23] [31][32]				√	
[24] [38]				√	√
[25] [26]					√
[27] [28] [33] [34][40][41]			√	√	
[29] [30]	√				
Proposed	√	√	√	√	√

On the other hand, the investigation in [29]-[30] was raised to speed up the process of the CU size decision of depth maps mentioned in Section II.B. In [29], the texture quadtree is used to restrict the search range of the depth quadtree. The work in [29] assumes that texture always has detailed structure than depth, but it is invalid for some regions where a depth map has more fine partition than texture as shown in the red rectangles of Fig. 5(a) and (c). In these regions, the quadtree structure estimated by [29] is not reliable. In [30] the variance and estimated synthesized view distortion of DIS (D_{DIS}) are used to determine whether the quadtree partition should be early terminated by a threshold method. The pre-estimation of quadtree by a variance feature [30] with different thresholds are also illustrated in Fig. 5(d) and (e), respectively. When threshold is larger, some LCUs may select larger block size wrongly in regions marked by the yellow square in Fig. 5(d). On the contrary, smaller threshold results in too detailed partitioning in the remaining part as shown in Fig. 5(e). Therefore, the variance feature is also not a very stable feature for CU partition.

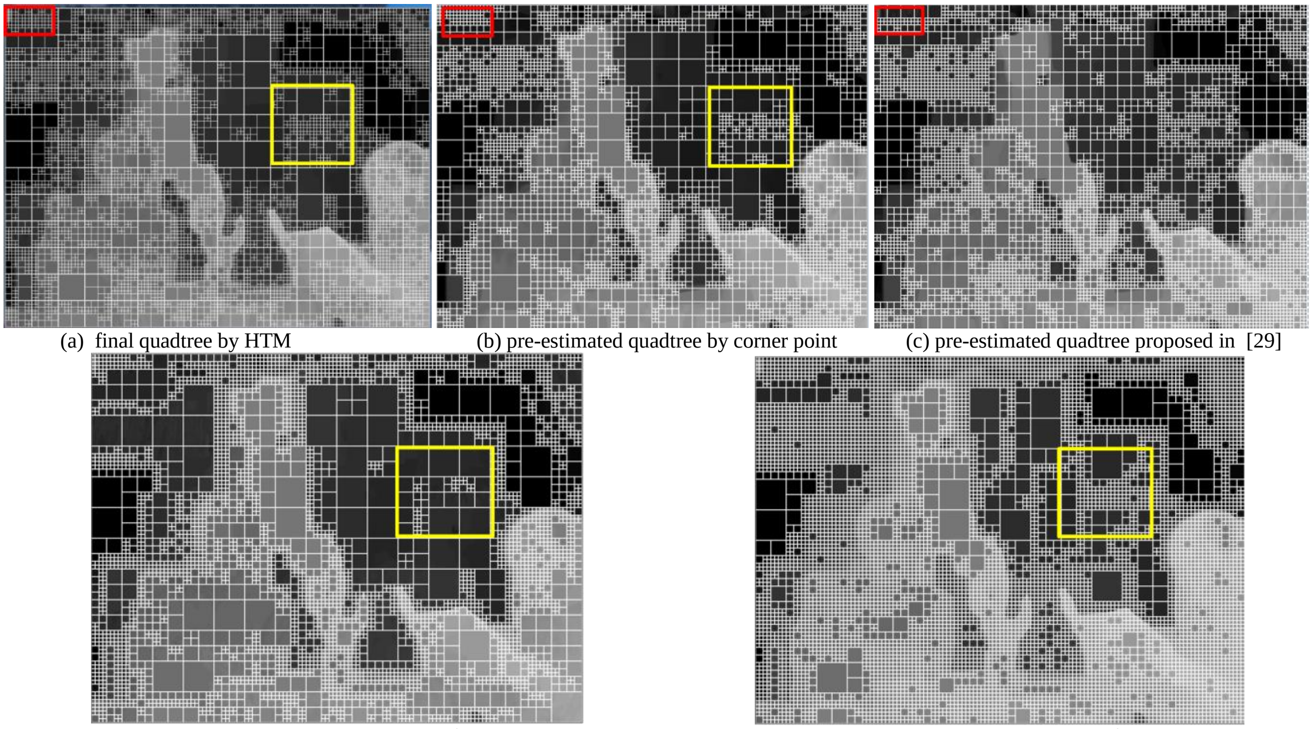
III. TWO-DIMENSIONAL GRADIENT FEATURE CP AND ITS USE IN THE PROPOSED FAST DEPTH INTRA CODING

Table I summaries the key techniques used in fast depth intra coding in Section II.C for 3D-HEVC. No matter the fast approaches accelerate the intra mode decision or the CU size decision, each of them depends on one or more features. These features such as horizontal edges, vertical edges, and variance



(a) Original depth map (b) Corner point map

Fig. 4. Original depth map and corner point map detected by KLT detector for Newspaper.



(a) final quadtree by HTM
(b) pre-estimated quadtree by corner point
(c) pre-estimated quadtree proposed in [29]
(d) pre-estimated quadtree proposed in [30] for threshold = 10.0
(e) pre-estimated quadtree proposed in [30] for threshold=1.0

Fig. 5. Quadtree of Newspaper: (a) final CU partition given by HTM ; the pre-estimated quadtree structure by (b) $FlagS_{CP}$ based on corner points , (c) texture quadtree [29], and (d) and (e) variance feature proposed in [30] with threshold values of 10.0 and 1.0, respectively.

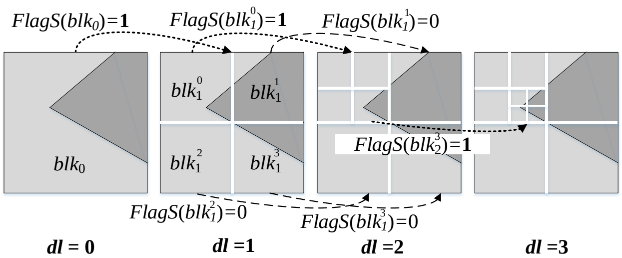


Fig. 6. Visualization of structural complexity in the conventional split process of quadtree.

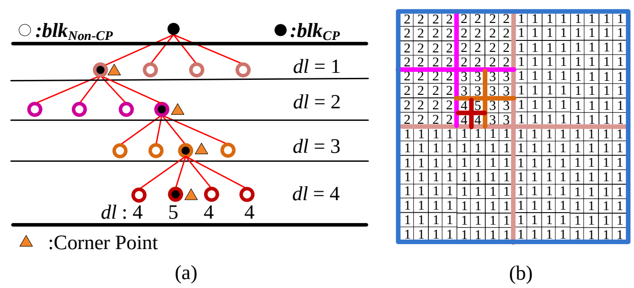


Fig. 7. Estimation of $pdl(i,j)$, (a) quadtree split by CP, and (b) the corresponding PDL matrix.

are insufficient for making proper predictions around the special multi-directional edges in depth maps.

Corner Point (CP) is defined as a point which has two dominant directions around its neighborhood, and used to detect a multi-directional edge. It is noted that a single directional edge is defined as the straight edge without CP and examples include a horizontal edge, a vertical edge, and a slanted edge. A typical depth frame from the sequence *Newspaper* and its corner point map detected by a KLT detector, which will be described in Section III.C, are shown in Fig. 4(a) and Fig. 4(b), respectively. As shown in Fig. 4(b), CPs are mostly distributed at the complicated boundary. As a CP is assumed to reflect the existence of multi-dimensional edges, it motivates us to consider CP as a good feature to predict the quadtree CU/PU partitioning structure and intra modes for CU/PU with multi-directional edges before the coding process.

Our previous work in [37] has initially designed a simple algorithm to estimate the quadtree CU/PU partitioning structure of depth maps based on CPs. In this paper, our contributions are as follows: (i) We further extend our work in [37] by employing CPs to form the Quadtree-depth Limited Strategy (QDLS) which is more sophisticated than the CU size decision in [37]. (ii) PU Decision (PUD) and Fast Intra Mode Decision (FIMD) are also suggested based on the distribution of CPs, (iii) We use a single two-dimensional gradient feature - Corner Points (CPs) to accelerate the CU/PU size decision and intra mode selection simultaneously. To the best of our knowledge, we are the first to introduce a good feature for both CU/PU decision and intra mode decision such that several fast algorithms for different encoding parts can be integrated by a single corner point feature. (iv) An adaptive corner point selection technique is also contrived to facilitate the proposed QDLS, PUD, and FIMD adaptive to frame content and quantization parameters with the capability of providing the flexible tradeoff between the synthesized view quality and complexity.

A. CU/PU Size Decision based on CPs

In general, each intra mode in Section II.A can only describe a block with one specific pattern. For instance, Planar mode provides good description for smooth blocks, while Angular-10 in Fig. 1 is good to predict horizontal edges. When CU/PU contains multi-directional edges, no single intra mode can provide accurate prediction. In this case, the quadtree CU/PU partitioning structure of depth coding is then employed to split

CU into smaller units until it reaches a partition that contains only one major pattern, which can be well modelled by single prediction mode.

***1) Motivation**:* For fast CU size decision, both features in [29] (the corresponding texture quadtree) and [30] (variance and D_{DIS}) do not consider the impact of multi-directional edges on the splitting process. An example of blk_0 with a multi-directional edge or a CP is used to illustrate the idea of the proposed CP based CU size decision in Fig. 6. In this example, $J_{VSO}(blk_0)$ in (4) is assumed to be a large value since it contains complicated pattern, and no single intra mode at dl of 0 can predict it well. On the contrary, $J_{VSO}(blk_1^k|k=1,2,3)$ are believed to be a small value since both of blk_1^1 ,blk_1^2 and blk_1^3 have a simple edge, which can be predicted well by one of the intra mode at dl of 1. It is highly probable that $J_{VSO}(blk_0) > \sum_{k=0}^{3} J_{VSO}(blk_1^k)$. As a result, $FlagS(blk_0)$ in (4) is set as 1 and blk_0 is further split. Similarly, $FlagS$ at dl of 1 could be also derived as $FlagS(blk_1^0)=1$ and $FlagS(blk_1^k|k=1,2,3)=0$.

It is observed in Fig. 6 that the block which contains CP (representing the multi-directional edge) always requires split. The location of CP is found to be highly correlated with $FlagS$ determined by J_{VSO} cost comparison in (4). Since CPs can be detected before the coding process, each blk_{dl} could be early classified as a set of $\mathrm{BLK_{CP}}$ or $\mathrm{BLK_{Non\text{-}CP}}$ based on

$$blk_{dl} \in \begin{cases} \mathrm{BLK_{CP}} & , \quad blk_{dl} \text{ contains CP} \\ \mathrm{BLK_{Non\text{-}CP}} & , \quad \text{otherwise} \end{cases} \tag{9}$$

Consequently, $FlagS(blk_{dl})$ in (4) can be rewritten as $FlagS_{CP}(blk_{dl})$, which depends on CP instead of J_{VSO} cost.

$$FlagS_{CP}(blk_{dl}) = \begin{cases} 1, & \text{if } blk_{dl} \in \mathrm{BLK_{CP}} \\ 0, & \text{if } blk_{dl} \in \mathrm{BLK_{Non\text{-}CP}} \end{cases} \tag{10}$$

That is to say, the early termination of the quadtree splitting process is determined by checking whether $blk_{dl} \in \mathrm{BLK_{CP}}$.

From (9) and (10), the same example in Fig.6 can be used to illustrate the splitting process based on the existence of CPs. As $blk_0 \in \mathrm{BLK_{CP}}$, $blk_1^0 \in \mathrm{BLK_{CP}}$, and $blk_1^k|k=1,2,3 \in \mathrm{BLK_{Non\text{-}CP}}$, it is drawn that $FlagS_{CP}(blk_0) = FlagS_{CP}(blk_1^0) = 1$ and $FlagS_{CP}(blk_1^k|k=1,2,3)=0$. From the above illustration, it is reasonable to replace $FlagS$ by $FlagS_{CP}$ in order to avoid the heavy computation of J_{VSO} in

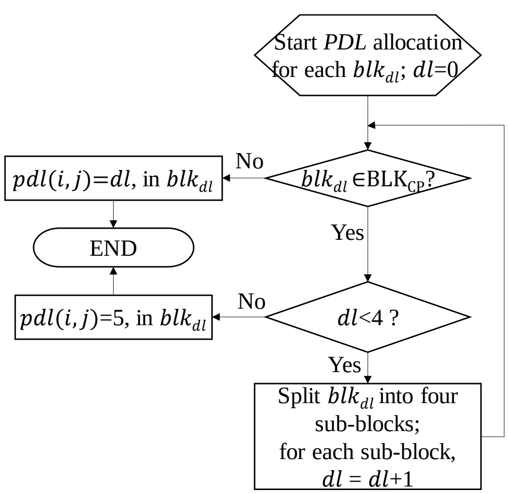


Fig. 8. The flowchart to generate the Pre-estimated Depth Level (PDL)

TABLE II
STATISTICAL RELATIONSHIP (%) BETWEEN *ODL* AND *PDL*, QP =34

Sequences	P({0,1}|0)	P({0,1}|1)	P({1,2,3}|2)	P({1,2,3}|3)
Kendo	99.61	98.02	89.56	93.98
Street	98.01	94.46	75.83	77.01
Fly	99.85	98.88	61.94	77.32
Average	**99.16**	**97.12**	**75.77**	**82.77**

(5). Fig. 5(b) shows the quadtree structure of the depth frame based on $FlagS_{CP}$ in (10), and it can be observed that this quadtree structure is very close to the optimal quadtree by using $FlagS$ in the test model as shown in Fig. 5(a). As compared with Fig. 5(c)-(e), the proposed CP feature obviously provides the best estimation as compared with the features used by [29] and [30].

2) Pre-estimated Depth Level (PDL): Although the quadtree obtained by $FlagS_{CP}$ is similar to the optimal quadtree determined by $FlagS$ in the test model, they are not exactly the same. It means that $FlagS$ cannot be simply replaced by $FlagS_{CP}$. Instead, CP is a good feature to form the pre-estimated depth level for each 4×4 block, $pdl(i,j)$, where (i,j) is the index of 4×4 blocks in LCU. Then the *PDL* matrix composed by $pdl(i,j)$ is used to guide the proposed CU/PU size decision, and it is defined as:

$$PDL = \begin{bmatrix} pdl(0,0) & \cdots & pdl(15,0) \\ \vdots & \ddots & \vdots \\ pdl(0,15) & \cdots & pdl(15,15) \end{bmatrix} \tag{11}$$

The flowchart to generate the *PDL* matrix is shown in Fig. 8. *PDL*(blk_{dl}) is defined as a set of $pdl(i,j)$ covering blk_{dl}. The example of the LCU containing a single CP in Fig. 6 and Fig.7 could be used to illustrate how to generate the *PDL* matrix according to the flowchart. At the first level of dl=0, as $blk_0 \in BLK_{CP}$, the LCU is split into four sub-blocks. For the next level dl=1, since blk_1^1, blk_1^2 and blk_1^3 do not contain corner point, the corresponding $pdl(i,j)$ are set to 1 as shown in Fig.7(b). At the same time, blk_1^0 is the further partitioned since $blk_1^0 \in BLK_{CP}$. This process continues for blocks containing CP until it reaches the smallest block size of 4×4. Finally, $pdl(i,j)$ of the 4×4 block with a CP inside is set to 5 in the *PDL* matrix of Fig. 7(b).

3) Correlation between Optimal Depth Level (ODL) and PDL: *ODL* is defined as the optimal depth level obtained by $FlagS$ in (4), which also forms the *ODL* matrix. Similar to *PDL* matrix, the *ODL* matrix is also composed of the smallest unit of each 4×4 block, $odl(i,j)$. In order to predict CU/PU size more accurately, the correlation between $odl(i,j)$ and $pdl(i,j)$ is examined statistically, and this statistical analysis is used to support the proposed algorithm. For $pdl(i,j)$ estimated by CP is "x", the conditional probability of $odl(i,j)$ to be "$y_0, y_1, \ldots, y_m$" is defined as

$$P(odl(i,j) = y_0, y_1, \ldots y_m | pdl(i,j) = x) \tag{12}$$

For the sake of simplicity, (12) is simplified to (13) in the following discussions.

$$P(y_0, y_1, \ldots y_m | x) \text{ , where } odl(i,j) = y, pdl(i,j) = x \tag{13}$$

It is noted that since the smallest CU partition is 8×8, the maximum dl is limited as 3 and dl=4 is only valid for PU partition as mentioned in Section III.A.5. As only CU partitions

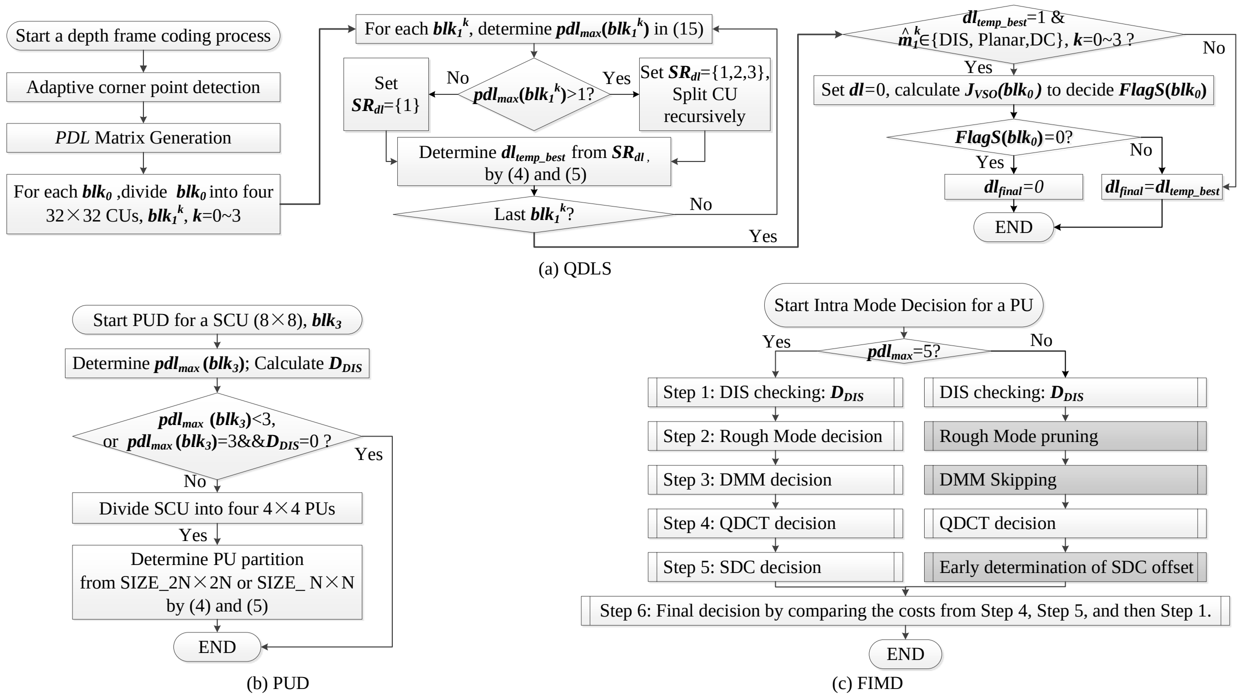


Fig. 9. Flowcharts of the proposed techniques, (a) QDLS, (b) PUD, and (c) FIMD.

are considered at the current stage, for all $pdl(i,j) \geqslant 3$, it is regarded as 3 in this sub-section. Table II tabulates four most significant probabilities that belong to $P(y_0, y_1, \ldots y_m|x)$ for the first 10 frames of three training video sequences with QP of 34. The values of $P(\{0,1\}|0)$ and $P(\{0,1\}|1)$ in Table II show that most CUs with small $pdl(i,j)$ finally choose small value of dl in the exhaustive search, which verifies the accuracy of the estimation by CPs. At the same time, the value of $P(\{1,2,3\}|2)$ and $P(\{1,2,3\}|3)$ is also remarkable in most cases.

4) Quadtree-depth Limited Strategy (QDLS)*:* According to the statistics shown in Table II, PDL can be used to limit the search range of depth levels, SR_{dl}, in CU size selection. It provides the condition that only $y_0, \ldots y_m$ are evaluated for CUs with the given $pdl(i,j)$ in a straightforward way, as follows:

$$SR_{dl} = \begin{cases} \{0,1\}, & \text{if } pdl(i,j) = 0 \text{ or } 1 \\ \{1,2,3\}, & \text{if } pdl(i,j) = 2 \text{ or } 3 \end{cases} \tag{14}$$

However, $P(\{1,2,3\}|2)$ and $P(\{1,2,3\}|3)$ are not as high as $P(\{0,1\}|0)$ and $P(\{0,1\}|1)$ in some sequences such as *Fly*. Moreover, it is found that median value dl=1 is important and checked for all cases in (14). Therefore, we rearrange the order of CU size decision from top-down to median-first in the implementation. The CU size decision at dl=1 is set at the starting point and the largest CU size dl=0 is checked at the last step to overcome the relatively low accuracy of $P(\{1,2,3\}|2)$ and $P(\{1,2,3\}|3)$ in some sequences.

The proposed CU size decision is named Quadtree-depth Limited Strategy (QDLS), which limits SR_{dl} based on PDL matrix as the following steps. At the same time, the flowchart of QDLS is shown in Fig.9(a) for illustration.

Step 1 (Initialization at dl=1): Each input LCU blk_0 is divided into four 32×32 CU, blk_1^k, where k=0,1,2,3

Step 2 (Determination of $pdl_{max}(blk_1^k)$): For each k, i.e, each sub-CU, the maximum value of $pdl(i,j)$ in $PDL(blk_1^k)$, $pdl_{max}(blk_1^k)$, is determined as

$$pdl_{max}(blk_1^k) = \max_{pdl(i,j) \in PDL(blk_1^k)} \{pdl(i,j)\} \tag{15}$$

Step 3 (Selection of SR_{dl}): Based on $pdl_{max}(blk_1^k)$, SR_{dl} of each LCU blk_0 is set as

$$SR_{dl} = \begin{cases} \{1,2,3\}, & \text{if } pdl_{max}(blk_1^k) > 1 \\ \{1\}, & \text{if } pdl_{max}(blk_1^k) \leq 1 \end{cases} \tag{16}$$

Step 4 (VSO cost evaluation based on SR_{dl}): Temporarily the best dl, dl_{temp_best}, is then determined by minimizing the VSO cost based on (4) and (5) for $dl \in SR_{dl}$.

Step 5 (Determination of the final dl): if dl_{temp_best}=1 is determined at Step 4 and the corresponding intra mode, $\hat{m}_1^k$, is DIS, Planar or DC, the VSO cost of dl= 0, $J_{VSO}(blk_0)$, is further checked and compared with $J_{VSO}(blk_1^k)$ to determine $FlagS(blk_0)$. The final dl, dl_{final}, for CU size decision is then determined by

$$dl_{final} = \begin{cases} 0, & \text{if } dl_{temp\_best} = 1 \,\&\, \hat{m}_1^k \in \{DIS, Planar, DC\} \,\&\, FlagS(blk_0) = 0 \\ dl_{temp\_test}, & \text{otherwise} \end{cases} \tag{17}$$

TABLE III STATISTICAL ANALYSIS OF PU PARTITION FOR SCUS WITH EACH pdl_{max}

Sequences	PERCENTAGE OF SIZE_N×N for SCUs with pdl_{max} =x (%)			
	x=0	x=1	x=2	x⩾3
Kendo	0.00	1.56	1.83	14.36
Street	0.00	7.59	6.42	16.74
Fly	0.00	2.78	3.70	20.53
Average	**0.00**	**3.98**	**3.98**	**17.21**

TABLE IV DISTRIBUTION OF SIZE_N×N SCUS AMONG DIFFERENT pdl_{max}

Sequences	pdl_{max} (%)			
	0	1	2	⩾3
Kendo	0.00	0.02	0.24	99.74
Street	0.00	1.15	3.81	95.04
Fly	0.00	0.03	0.26	99.71
Average	**0.00**	**0.40**	**1.44**	**98.16**

5) PU Partition Decision (PUD) for SCU: As aforementioned, the PU partition is the same as CU partition (called SIZE_2N×2N) for dl=0, 1, 2. When dl=3 for a SCU (8×8), two possible PU partitions are available, SIZE_2N×2N, and SIZE_N×N (4×4). The VSO cost evaluation of SIZE_N×N induces further complexity. Similar to the CU size decision in QDLS, the proposed PUD for SCU in this section is designed to skip unnecessary SIZE_N×N evaluations in PU partition based on the *PDL* matrix.

In general, SIZE_N×N tends to be used for complex regions. Since the *PDL* matrix can roughly reflect the structural complexity, it is reasonable to assume that the PU partition size may have strong correlation with the *PDL* matrix. Several probabilities are measured in Table III and Table IV to investigate the correlation. As each SCU contains four $pdl(i,j)$ values, $pdl_{max}(blk_3)$ covered by SCU is used to make the PU size decision. First, for SCUs given different $pdl_{max}(blk_3)$, the probabilities of SIZE_N×N are illustrated in Table III. The results show that for the SCUs with the corresponding $pdl_{max}(blk_3)$ as 0, 1 and 2, only 0.03%, 2.78%, and 3.07% of them select SIZE_N×N as PU size on average. On the contrary, 17.01% of SCUs with $pdl_{max}(blk_3)\geq 3$ utilize SIZE_N×N. Obviously, SIZE_N×N is negligible in SCUs with small $pdl_{max}(blk_3)$. Second, the distribution of SIZE_N×N among CUs with different $pdl_{max}(blk_3)$ is tabulated in Table IV for further verification. It is shown that for all the SCUs utilizing SIZE_N×N, only 0.02%, 0.14% and 0.97% of them have the corresponding $pdl_{max}(blk_3)$ as 0, 1 and 2.

According to the above analysis, SIZE_N×N could be skipped for SCUs with $pdl_{max}(blk_3)$=0, 1, or 2 in the PU partition decision. Besides, for some blocks with $pdl_{max}(blk_3)$=3, which are smooth enough, the chance of using SIZE_N×N is also very small. It is found that when one 8×8 CU contains only single depth value, it can be well described by DIS in the first step of the mode decision instead of being further partitioned. Therefore, we propose to make an early evaluation of DIS and use the corresponding D_{DIS} as a feature to represent the smoothness of 8×8 CU. The condition for skipping SIZE_N×N of SCUs is proposed as:

$$pdl_{max}(blk_3) < 3 \text{ or } (pdl_{max}(blk_3) = 3 \,\&\, D_{DIS} = 0) \quad (18)$$

The flowchart of PUD is depicted in Fig. 9(b). It is also noted that the effect of PUD is partially overlapped with QDLS. When $pdl_{max}(blk_3)$ =0 or 1, partition is already terminated at dl=1 by QDLS. Consequently, PU decision is automatically skipped.

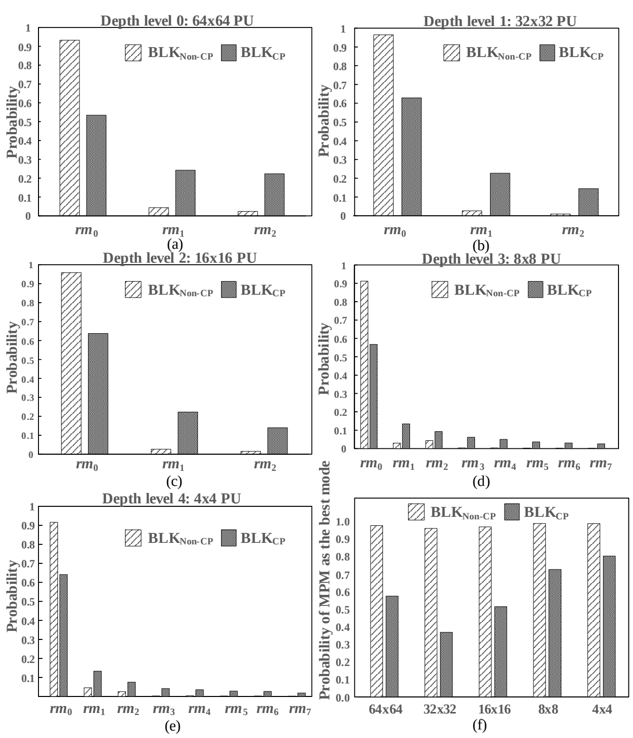


Fig. 10. Conditional probabilities of RMs for $\mathrm{BLK_{Non\text{-}CP}}$ and $\mathrm{BLK_{CP}}$ at *dl* of (a) 0, (b) 1, (c) 2, (d) 3, and (e) 4; Conditional probabilities of MPM for $\mathrm{BLK_{Non\text{-}CP}}$ and $\mathrm{BLK_{CP}}$ at different *dl*s in (f).

B. Fast Intra Mode Decision based on CPs

In Section III.A, we reveal that CP is a good feature to decide the splitting process of CU/PU size by making use of $\mathrm{BLK_{CP}}$. At the same time, $\mathrm{BLK_{Non\text{-}CP}}$ can be used to accelerate the intra mode decision in each PU, since the PUs belong to $\mathrm{BLK_{Non\text{-}CP}}$ should be well predicted by single intra mode. Three methods are proposed for fast intra mode decision in $\mathrm{BLK_{Non\text{-}CP}}$ including CHIM pruning, DMM decision skipping and early determination of SDC coding. The flowchart of FIMD is shown in Fig. 9(c), where the left branch is the default flow. The three proposed methods are associated with Steps 2, 3 and 5, respectively, in the right branch of Fig. 9(c) when the current PU belongs to $\mathrm{BLK_{Non\text{-}CP}}$.

1) Rough Mode Pruning (RMP): In (1), CHIM is composed of RM and MPM. The final CHIM modes are sorted by the model-based VSO cost from rm_0 to rm_7, where rm_0 has the lowest rough cost. In this section, it is observed that the prediction accuracy of RM and MPM varies notably of PUs in $\mathrm{BLK_{Non\text{-}CP}}$ and $\mathrm{BLK_{CP}}$.This phenomenon can be demonstrated by the conditional probabilities of the final rm_i to be selected as the optimal intra mode in Fig. 10.

From Fig. 10(a)-(e), over 90% of $\mathrm{BLK_{Non\text{-}CP}}$ selects the first candidate rm_0 to be the optimal intra mode. On the contrary, this probability is only 60% in $\mathrm{BLK_{CP}}$. Obviously, the probability distribution of CHIM is extremely biased to rm_0 in $\mathrm{BLK_{Non\text{-}CP}}$. Besides, about 98% of $\mathrm{BLK_{Non\text{-}CP}}$ select one of MPM as the optimal intra mode, while the percentage is only around 60% for $\mathrm{BLK_{CP}}$ as illustrated in Fig. 10(f). It shows that MPM predicted from spatial neighbors are quite suitable for $\mathrm{BLK_{Non\text{-}CP}}$. The reason behind is that the spatial orientation among neighboring PUs is more consistent in $\mathrm{BLK_{Non\text{-}CP}}$.

Due to the extreme bias to rm_0 and MPM observed in $\mathrm{BLK_{Non\text{-}CP}}$, some redundant rough mode candidates can be removed safely. In the proposed RMP, the relationship between RM and MPM is exploited to further ensure the accuracy of estimation. For $\mathrm{BLK_{Non\text{-}CP}}$, several rules are applied to prune the candidate set as

$$\mathrm{CHIM}^* = \begin{cases} \{rm_0\} & , \quad \text{if } rm_0 \in \mathrm{MPM} \\ \{rm_0, rm_1\} & , \quad \text{else if } rm_1 \in \mathrm{MPM} \\ \{rm_0, rm_2\} & , \quad \text{else if } rm_2 \in \mathrm{MPM} \\ \{rm_0, \mathrm{MPM}\} & , \quad \text{else if } 0 \leq dl \leq 2 \\ \{rm_0, rm_1, rm_2, \mathrm{MPM}\} & , \quad \text{else if } 3 \leq dl \leq 4 \end{cases} \quad (19)$$

Then, the proposed RMP is given by

TABLE V STATISTICAL ANALYSIS OF DMM FOR $BLK_{Non\text{-}CP}$ AND BLK_{CP} (%)

Sequences		32x32	16x16	8x8	4x4
Kendo	$P(DMM\|BLK_{Non\text{-}CP})$	3.07	2.07	0.92	0.57
	$P(DMM\|BLK_{CP})$	35.46	27.90	17.18	10.59
Street	$P(DMM\|BLK_{Non\text{-}CP})$	3.18	1.26	0.68	0.35
	$P(DMM\|BLK_{CP})$	16.47	11.09	7.72	4.09
Fly	$P(DMM\|BLK_{Non\text{-}CP})$	2.85	4.88	2.38	0.38
	$P(DMM\|BLK_{CP})$	29.57	29.51	26.12	15.77
Average	**$P(DMM\|BLK_{Non\text{-}CP})$**	**3.03**	**2.73**	**1.33**	**0.44**
	$P(DMM\|BLK_{CP})$	**27.17**	**22.84**	**17.01**	**10.15**

$$\mathrm{RMP}(blk_{dl}) = \begin{cases} \text{CHIM}^* \text{ in (19)}, & \text{if } blk_{dl} \in \text{BLK}_{\text{Non-CP}} \\ \text{CHIM in (1)}, & \text{otherwise} \end{cases} \quad (20)$$

2) DMM skipping algorithm (DMMSA): Although the features in [18] and [21] help to skip DMM in smooth and edge blocks accurately as shown in (7) and (8), they cannot recall all the PUs where DMM evaluation are redundant. For example, those blocks containing slanted edges, could not be signed by a simple horizontal feature or/and vertical feature. Redundant DMM evaluations are still performed in the PUs that belong to $BLK_{Slanted}$ in Fig. 3, where $BLK_{Slanted}$ is a set of the PUs with slanted edges. It is further observed that $BLK_{Slanted}$ in Fig. 3 could also be well predicted by corresponding angular modes in CHIM. This motivates us to also include $BLK_{Slanted}$ in ψ of (6) to further reduce the redundant DMM evaluation, as (21).

$$\psi = \{BLK_{Smooth}, BLK_{Hor}, BLK_{Ver}, BLK_{Slanted}\} \quad (21)$$

To detect $blk_{dl} \in \psi$ in (21), CP again is a good feature that can distinguish blk_{dl} with a straight edge from the blocks with multi-dimensional edges, which is ignored by all existing features in the literature.

Since $BLK_{Non\text{-}CP}$ implies no multi-dimensional edge, it is equivalent to ψ in (21), which can then be written as

$$\psi = BLK_{Non\text{-}CP} = \{BLK_{Smooth}, BLK_{Hor}, BLK_{Ver}, BLK_{Slanted}\} \quad (22)$$

In order to verify the classification accuracy of CPs, the conditional probability of DMM utilization is calculated for each class. The results for different block sizes are summarized in Table V, where $P(DMM|BLK_{Non\text{-}CP})$ and $P(DMM|BLK_{CP})$ represent the conditional probability of DMM to be the best intra mode for the PUs belongs to $BLK_{Non\text{-}CP}$ and BLK_{CP}, respectively. As shown in Table V, for $BLK_{Non\text{-}CP}$, the average probability of DMM to be the optimal mode is only 3.03%,2.73% 1.33%, and 0.44%, in different block sizes. On the contrary, about 27.17%, 22.84%, 17.01% and 10.15% of BLK_{CP} select DMM as the best mode. This analysis confirms that almost all $BLK_{Non\text{-}CP}$ can be predicted well by CHIM and the time consuming DMM decision can be skipped safely in these PUs. The proposed algorithm DMMSA is given by

$$\mathrm{DMMSA}(blk_{dl}) = \begin{cases} \emptyset, & \text{if } blk_{dl} \in \text{BLK}_{\text{Non-CP}} \\ \text{DMM in (2)}, & \text{otherwise} \end{cases} \quad (23)$$

3) Early determination of SDC offset (EDSDC): After *RMD* and *DMM* decision, some candidates are selected for further determination of residual coding mode (QDCT or SDC). In Step 5 of Section II.A, the best offset is determined from (3) by evaluating the rendering-based VSO cost function. The number of candidates is limited compared to DMM, but the rendering-based VSO cost function is still time consuming. The algorithm in [26] designs a RD-cost based method, in which the offsets of 2 and -2 can be skipped if the offset of 0 is superior to the offsets of 1 and -1. However, there are still three offsets to be checked for each selective candidate mode. In fact, the multiple offset candidates in SDC are designed as refinement for the regions where the selective intra mode cannot provide a well prediction for the current block [12].

TABLE VI
STATISTICAL ANALYSIS OF SDC OFFSET FOR $BLK_{Non\text{-}CP}$ AND BLK_{CP}(%)

PU	Cases	0	1	-1	2	-2
64x64	$BLK_{Non\text{-}CP}$	**99.1**	0.4	0.5	0.0	0.0
	BLK_{CP}	66.0	12.1	6.1	12.1	3.7
32x32	$BLK_{Non\text{-}CP}$	**98.1**	0.9	0.7	0.1	0.2
	BLK_{CP}	76.3	11.2	5.3	5.0	2.2
16x16	$BLK_{Non\text{-}CP}$	**97.7**	1.0	0.8	0.2	0.2
	BLK_{CP}	82.2	7.8	5.0	3.0	1.9
8x8	$BLK_{Non\text{-}CP}$	**96.8**	1.3	1.3	0.3	0.4
	BLK_{CP}	86.9	5.3	4.4	1.8	1.7

As $BLK_{Non\text{-}CP}$ is likely to be well predicted by single intra mode, it is reasonable to consider skipping some candidates of SDC in $BLK_{Non\text{-}CP}$. The distributions of the optimal SDC offsets for $BLK_{Non\text{-}CP}$ and BLK_{CP} are illustrated in Table VI. It can be verified that the overwhelming majority about 97% of PUs choose the offset "0" to encode residual signals for $BLK_{Non\text{-}CP}$. In contrast, the percentage is only about 66%~87% for BLK_{CP}. Based on the statistics, an early determination of the SDC offset is proposed for only checking the offset 0 for $BLK_{Non\text{-}CP}$, which is given by

$$\mathrm{EDSDC}(blk_{dl}) = \begin{cases} \{0\}, & \text{if } blk_{dl} \in \text{BLK}_{\text{Non-CP}} \\ \text{SDC in (3)}, & \text{otherwise} \end{cases} \quad (24)$$

C. Corner Point Detection

The way to select CPs properly is of great importance for the proposed techniques. In this paper, KLT operator [42], [43] is used to detect CPs in a depth frame. The covariance matrix of each sample pixel and its neighbors is formed as *M*.

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (25)$$

where $w(x,y)$ is the weight function at pixel coordinates *x* and *y* of the depth frame, and I_x and I_y are the horizontal and vertical gradients, respectively. Based on the classical Principal Component Analysis (PCA) theory, the two eigenvectors of *M* (quadric form) are the two principal directions of the pixel at location (x,y) of the depth frame and their eigenvalues can

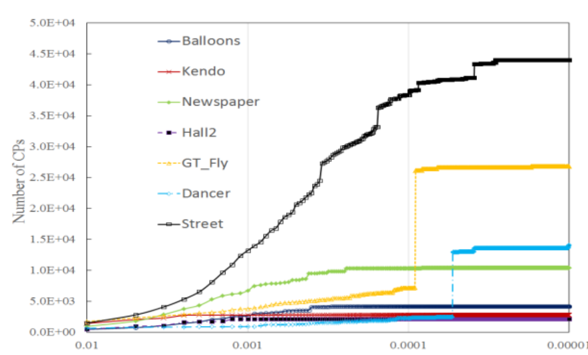


Fig. 11. Number of CPs by different values of threshold T.

reflect the degrees of change in their directions. Since each real CP should have at least two dominant directions. The smaller one of the two eigenvalues is noted as $\lambda_{M(x,y)}$. When $\lambda_{M(x,y)}$ is greater than a pre-defined threshold T, this sample is regard as a CP. Obviously, a proper threshold T is important for CP detection since it determines to the level of tolerated gradient change for both sides of boundaries.

In our proposed work, the extracted CPs are used to estimate the quadtree structure and intra mode, which highly depend on the video content and quantization parameter (QP) of a given frame. Therefore, it is better to generate CPs adaptive to the video content and QP. To solve this problem, a two-stage CP selection algorithm is designed. To be more specific, all possible CP candidates are detected first, and then QP is used for refinement. Fig. 11 shows the number of CPs detected by the KLT operator with different thresholds. It can be seen that the number of CPs increases as T decreases for most sequences. However, the increasing trends are different for sequences. We select three sequences *Kendo*, *Street* and *Fly*, which have different characteristics in Fig. 11. They are used to train the threshold T. Based on the extensive simulation results on these training sequences, it is found that when T is smaller than 0.0001, the continuous depth value change would be unexpectedly detected as CPs by the KLT operator. As a result, the value of T is set to 0.0001 in the first step. It is verified later that the threshold T also works well for remaining sequences. This threshold T helps to reject most of the unexpected CPs introduced by continuous depth changes and detected all possible CP candidates. It is noted that the temporal and inter-view correlations of depth video are much lower than those of texture video [44]. As a result, it is difficult to adjust T adaptively by the video content of temporal or inter-view frames. However, the number of CP candidates is still adaptive to video content. For example, in Fig. 11, the number of CP candidates is about 2000 for *Hall2* with mostly smooth regions, while it is 10000 in *Newspaper*, which has more detailed texture in depth maps.

Then, the number of CP candidates selected in the first step is defined as N_C. All these N_C CPs are sorted by their corresponding eigenvalues values $\lambda_{M(x,y)}$ [42]. The second step is to choose proper number of CPs from the sorted CP candidates according to QP. In the Section III, it is found from the proposed techniques that less number of CPs means more blocks could be accelerated in CU/PU size decision and intra mode decision at the expense of quality degradation. When QP is smaller than or equal to 36, the quality of the synthesis view is more concerned in the RD-cost evaluation. All N_C CP candidates are reserved as CPs to ensure the RD performance. When QP is larger than 36, the number of final CPs is reduced by a ratio. Only the first $FNum_{CP}$ CP candidates with higher eigenvalues are selected as CPs. The final number of CPs, $FNum_{CP}$, is trained with extensive experiments by trading off the coding performance and complexity as follows.

$$FNum_{CP}=\begin{cases} N_C, & QP \le 36 \\ N_C \times \frac{5-(QP-37)\%3}{3\times 2^{(\lfloor \frac{QP-37}{3} \rfloor +1)}}, & \text{otherwise} \end{cases} \tag{26}$$

where % is the modular arithmetic operator and $\lfloor . \rfloor$ is the floor function. The design of $FNum_{CP}$ is also motivated by the process of determining the value of λ in the RD function of mode decision [45]. In the selection of λ, when QP is increased by 3, λ is doubled to reduce the weighting of quality distortion. The proportion of CPs is kept in similar way. When QP is increased by 3, the number of the final CPs is reduced by half. The ratio $FNum_{CP}/Nc$ is 1/2, 1/4, and 1/8 when QP increases from 39, 42, to 45. By this way, we can adaptively choose the suitable number of CPs as an approximately exponential descent function according to QP.

TABLE VII SEQUENCES FOR EXPERIMENTS

Sequences	Frame rate	Spatial Resolution	View Number	Usage[a]
Kendo	30	1024×768	3	**T-set**
Street	25	1920×1088	3	**T-set**
Fly	25	1920×1088	3	**T-set**
Newspaper	30	1024×768	3	V-set
Balloons	30	1024×768	3	V-set
Hall2	25	1920×1088	3	V-set
Dancer	25	1920×1088	3	V-set
Shark	30	1920×1088	3	V-set

a T-set: Training, V-set: Validation

IV. SIMULATION RESULTS

The proposed QDLS, PUD, and FIMD have been implemented in the 3D-HEVC reference software HTM version 16.1 [36] to verify their performances. The experiments were carried out on three video sequences at XGA resolution (1024x768): *Kendo*, *Balloons*, *Newspaper*, and five video sequences at 1080p resolution (1920x1088): *Dancer*, *Hall2*, *Street*, *Fly*, and *Shark* specified by JCT-3V in which 200 to 250 frames were coded for each sequence. Table VII tabulates the usage of all the sequences. As mentioned in Section III.C, *Kendo*, *Street* and *Fly* are utilized to train the corner point selection model. The results are further validated by other sequences, *Newspapers*, *Balloons*, *Dancer*, *Hall2, and Shark*. All sequences are classified as Training-set (T-set) and Verification-set (V-set), respectively. The average value of each set is also shown in the following results.

Since the proposed algorithms focus on depth intra coding, all the frames were coded as I-frame under the all intra (AI) configuration specified by the common test condition defined in [46]. The QP of coding depth maps was set to 34, 39, 42 and 45. The original HTM-16.1 was implemented as anchor. Some existing approaches were implemented for comparison. All fast algorithms which have been adopted by the test model were enabled in the simulations. For each video sequence, three views of the depth map and texture image (V_{d0}, V_{d1}, V_{d2}, V_{t0}, V_{t1} and V_{t2}) were encoded and six synthesized views ($Syn_{0.25}$, $Syn_{0.5}$ $Syn_{0.75}$, $Syn_{1.25}$, $Syn_{1.5}$ and $Syn_{1.75}$) were generated by the decoded depth maps and texture images. Six virtual synthesized views from the uncompressed depth maps and textures were then used as the anchor of synthesized view for PSNR calculation. The Bjontegaard Delta bitrate (ΔBDBR) [47] was preformed to measure the coding performance, which comes from the total bitrate of texture and depth video, and the PSNR of synthesized views. ΔT (%) was used to evaluate the time reduction of the proposed algorithm, which is defined as

$$\Delta T = \frac{Time_{proposed} - Time_{HTM}}{Time_{HTM}} \times 100\% \tag{27}$$

where $Time_{proposed}$ represents the encoding time of depth coding by the proposed algorithm whereas $Time_{HTM}$ denotes

TABLE VIII Performance of QDLS, PUD and FIMD Compared with HTM-16.1

Sequences	QDLS					PUD					FIMD				
	ΔT (%) for different QPs				ΔBDBR (%)	ΔT (%) for different QPs				ΔBDBR (%)	ΔT (%) for different QPs				ΔBDBR (%)
	34	39	42	45		34	39	42	45		34	39	42	45	
Kendo	-34.7	-35.8	-45.5	-56.7	0.27	-13.0	-14.5	-18.9	-22.2	0.00	-26.7	-35.9	-42.8	-49.8	0.29
Street	-25.6	-42.5	-53.1	-63.0	0.24	-10.2	-13.4	-19.5	-23.0	0.01	-34.5	-47.1	-47.5	-50.6	0.26
Fly	-54.1	-62.0	-63.7	-67.0	0.18	-12.9	-15.8	-22.8	-26.8	0.00	-50.3	-54.5	-49.1	-47.1	0.20
T-Avg.	**-38.1**	**-46.8**	**-54.1**	**-62.2**	**0.23**	**-12.0**	**-14.6**	**-20.4**	**-24.0**	**0.00**	**-37.2**	**-45.8**	**-46.5**	**-49.2**	**0.25**
Newspaper	-18.0	-18.4	-26.6	-40.5	0.17	-5.7	-7.7	-13.2	-17.9	0.02	-16.4	-27.2	-35.8	-45.8	0.45
Balloons	-26.5	-29.4	-38.3	-52.8	0.15	-9.6	-12.2	-16.4	-20.1	0.01	-21.1	-32.5	-40.8	-50.3	0.45
Hall2	-55.1	-62.9	-70.3	-76.4	1.13	-19.0	-22.2	-27.2	-30.1	0.02	-42.5	-50.4	-50.8	-52.6	1.14
Dancer	-52.8	-54.6	-58.3	-61.8	1.08	-16.7	-20.1	-23.7	-27.0	0.03	-39.9	-47.3	-45.1	-47.1	0.35
Shark	-25.9	-31.0	-43.1	-55.2	0.46	-12.5	-15.0	-20.9	-23.7	0.02	-26.6	-37.0	-43.3	-48.4	0.65
V-Avg.	**-35.7**	**-39.3**	**-47.3**	**-57.3**	**0.60**	**-12.7**	**-15.4**	**-20.3**	**-23.8**	**0.02**	**-29.3**	**-38.9**	**-43.2**	**-48.8**	**0.61**
Average	**-36.6**	**-42.1**	**-49.9**	**-59.2**	**0.46**	**-12.4**	**-15.1**	**-20.3**	**-23.8**	**0.02**	**-32.2**	**-41.5**	**-44.4**	**-48.9**	**0.48**

TABLE IX Percentage of CU with Different *pdl*(*i*,*j*) for Different QPs

Sequence	QP=34, *pdl*(*i*,*j*) (%)				QP=39, *pdl*(*i*,*j*) (%)				QP=42, *pdl*(*i*,*j*) (%)				QP=45, *pdl*(*i*,*j*) (%)			
	0	1	2	≥ 3	0	1	2	≥ 3	0	1	2	≥ 3	0	1	2	3
Kendo	27.40	20.06	20.41	32.13	30.80	25.14	21.42	22.65	43.68	27.36	15.96	13.00	57.26	23.59	11.57	7.58
Street	7.90	23.32	28.46	40.33	30.55	26.48	20.35	22.62	52.34	21.04	13.92	12.70	69.96	15.05	8.33	6.67
Fly	42.67	22.37	14.92	20.04	63.15	15.77	10.02	11.05	74.69	12.37	6.70	6.24	82.72	8.83	4.71	3.75
T-Avg.	**25.99**	**21.92**	**21.26**	**30.83**	**41.50**	**22.46**	**17.26**	**18.77**	**56.90**	**20.26**	**12.19**	**10.65**	**69.98**	**15.82**	**8.20**	**6.00**
Newspaper	6.42	11.87	18.41	63.30	9.62	15.85	25.13	49.40	17.88	24.71	26.85	30.55	35.10	27.83	20.62	16.45
Balloons	22.57	16.74	20.35	40.34	26.35	20.86	23.17	29.62	38.40	24.38	19.80	17.42	54.17	24.37	12.35	9.11
Hall2	52.95	22.70	13.68	10.67	64.14	19.36	9.85	6.65	74.81	14.96	6.47	3.76	86.35	8.52	3.27	1.86
Dancer	61.20	16.42	11.13	11.25	77.35	11.36	6.30	4.99	85.53	7.56	4.04	2.87	88.51	6.47	3.13	1.89
Shark	4.31	14.29	33.97	47.42	11.75	26.66	34.99	26.59	24.30	38.60	23.23	13.87	46.10	32.80	13.65	7.45
V-Avg.	**29.49**	**16.40**	**19.51**	**34.60**	**37.84**	**18.82**	**19.89**	**23.45**	**48.18**	**22.04**	**16.08**	**13.69**	**62.05**	**20.00**	**10.60**	**7.35**
Average	**28.18**	**18.47**	**20.17**	**33.19**	**39.21**	**20.19**	**18.90**	**21.70**	**51.45**	**21.37**	**14.62**	**12.55**	**65.02**	**18.43**	**9.70**	**6.84**

that of the original HTM reference software. Besides, as discussed before, the threshold of KLT detetor is set to 0.0001 for all simulations. All experiments were conducted on a PC with the platform of 3.3GHz CPU and 16GB RAM.

A. *Performance of individual algorithms*

1) *Performance of QDLS*

The coding performance in terms of ΔT and ΔBDBR of the proposed QDLS is tabulated in Table VIII. This table shows that QDLS can save the encoding time by 36.6%-59.2% on average. Besides, QDLS can save more encoding time as QP increases for all sequences. It is consistent with the $pdl(i,j)$ distribution for different QPs as shown in Table IX. When QP is large, less CPs are detected. Consequently, more CUs are assigned with small values of $pdl(i,j)$ as demonstrated in Table IX. From this table, the average percentages of $pdl(i,j)=0$ increase from 28.18% to 65.02% as QP increases from 34 to 45, while the average percentage of $pdl(i,j) \geq 3$ significantly decreases from 33.19% to 6.84%. It is noted that since blocks in different depth levels have various sizes, counting the number of basic 4×4 unit covered by blocks of each $pdl(i,j)$ is more meaningful than counting the number of blocks directly. Due to adoption of the median-first strategy in QDLS, dl of 1 is checked no matter what $pdl(i,j)$ is. Then if $pdl(i,j)$ is equal to 0 or 1, the further split to dl of 2 and 3 will be directly skipped. Obviously, high percentage of $pdl(i,j)=0$ and $pdl(i,j)=1$ indicates more time saving. As a result, the time saving of QDLS in Table VIII increases from 36.6% to 59.2% as QP increases from 34 to 45.

In particular, QDLS demonstrates more time saving in *Hall2* and *Fly* since they contain simple contents and less CPs are detected in the first step of CP detection. On the other hand, the time reduction becomes smaller if the input depth sequence includes complex content such as *Street* and *Newspaper*. In Table VIII, it can also be found that the average increase in ΔBDBR is only 0.46%, which is not noticeable. It is noted that the increase in ΔBDBR is a little higher for *Hall2*. In this sequence, the depth values gradually change in some regions, which is difficult for the feature CP to model. In contrast, there are complex contents in the corresponding texture video. When the gradual change region cannot be detected by CPs, the coding loss of the depth map will introduce visible influence on the quality of the synthesized view.

2) *Performance of PUD*

It is also shown in Table VIII that 12.4%-23.8% of ΔT on average is achieved by the proposed PUD. The results confirm that PUD is more efficient at extremely high QPs. It is because the low bitrate case has more SCUs with $pdl(i,j)$ of 0, 1 and 2 where the checking of SIZE_NxN is skipped by PUD. At the same time, the increase in ΔBDBR is only 0.02% on average, which can be neglected in terms of coding efficiency.

3) *Performance of FIMD*

Table VIII also shows the performance of the proposed FIMD. On average, the time reduction is 32.2%-48.9% with 0.48% of ΔBDBR increase as compared with HTM-16.1, which has adopted the algorithms of [15], [18], [25], and [26].

FIMD mainly includes RMP, DMMSA and EDSDC. The encoding time reductions achieved by these techniques are also drawn in Fig. 12. When QP becomes higher, less CPs are detected and more PUs are classified as $BLK_{Non\text{-}CP}$. As a result, RMP tends to select fewer candidates and achieves larger time reduction. Table X further shows the percentage of PUs, where DMM are evaluated in DMMSA. In this table, Gu's method [18] adopted by HTM-16.1 is also shown for comparison. It employs a threshold related to QP for DMM skipping. Similarly, the number of CPs in the proposed DMMSA also depends on QP.

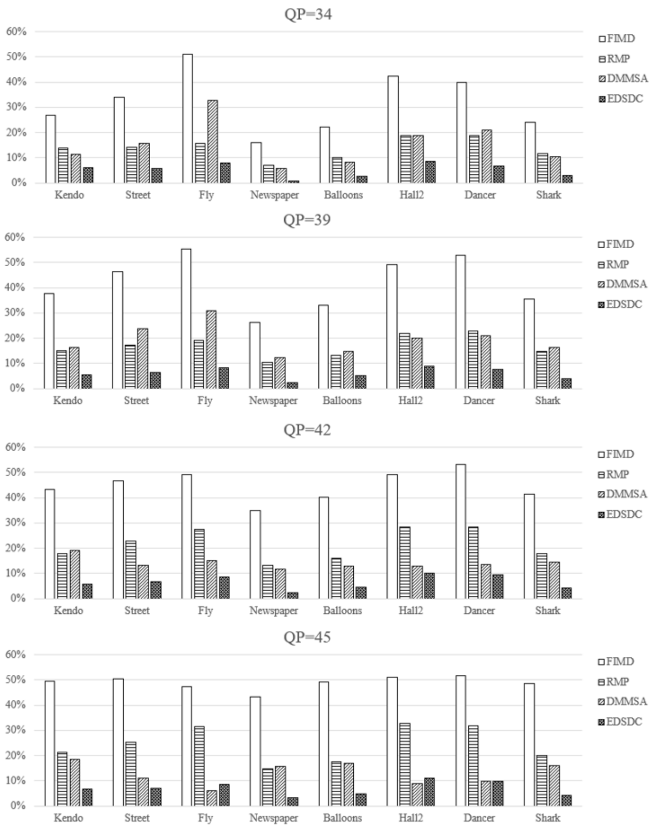


Fig. 12. Encoding time saving of FIMD, which is composed of RMP, DMMSA, and EDSDC, for different QPs.

When QP increases, both methods calculate less number of DMM evaluations as illustrated in Table X. It is also consistent with the trend of the increase in time reduction with higher QP in Fig. 12. It is noted that the time saving in Fig. 12 is calculated by comparing the proposed DMMSA to HTM-16.1 with Gu's method [18] enabled which skips DMM evaluations for most PUs already in some sequences with high QP. As a result, the time savings of DMMSA in *Dancer*, *Fly* and *Hall2* even get smaller with larger QPs as shown in Fig. 12. Table XI presents the percentage of PUs, where the offset of SDC are checked in EDSDC. As shown in the table, SDC offsets are checked in more PUs when Lee's method [26] is used. Obviously, EDSDC could provide more time reduction than Lee's method [26].

B. Performance of the overall algorithm

In addition to HTM16.1, some state-of-the-art methods have been used for comparison with our algorithm which includes all of QDLS, PUD, and FIMD. To the best of our knowledge, no work in the literature suggested any feature for expediting both of the CU/PU decision and intra mode decision. It is also one of our contributions to integrate several fast decisions for different encoding parts by making use of the proposed two-dimensional gradient feature - corner point. We compared with the Park's method in [21], Gu's [19], Zhang's [23], Mora's [29], Saldanha's [40], Sanchez's [41], and Zhang's [34] for the evaluation of our algorithm. The comparison results are tabulated in Table XII. The proposed algorithm can achieve 66.56% of ΔT with 1.08% of ΔBDBR increase on average. It is superior to the existing algorithms. Among them, the algorithms in [19], [21] and [23] focus on the DMM decision. They share the similar idea of the method [18] in HTM16.1. In

TABLE X Ratio of PUs (%) of which DMM are evaluated for [18] and DMMSA

Sequences	Gu's method [18]				Proposed DMMSA			
QP	34	39	42	45	34	39	42	45
Kendo	42.8	43.0	33.7	28.1	26.8	19.9	12.2	7.6
Street	63.0	58.5	28.3	19.3	33.7	19.7	11.5	6.3
Fly	51.2	37.5	23.6	16.0	17.4	9.9	5.8	3.6
T-Avg.	**52.3**	**46.3**	**28.5**	**21.1**	**25.9**	**16.5**	**9.8**	**5.8**
Newspaper	59.7	57.0	41.6	35.0	49.8	39.0	26.2	15.5
Balloons	45.6	45.0	35.4	30.6	32.8	25.0	16.0	8.8
Hall2	34.3	30.7	17.3	11.3	10.1	6.6	3.9	1.9
Dancer	66.6	53.4	21.1	9.2	9.9	4.7	2.8	1.9
Shark	65.5	61.0	45.4	42.1	38.5	25.2	14.4	8.0
V-Avg.	**54.3**	**49.4**	**32.2**	**25.6**	**28.2**	**20.1**	**12.7**	**7.2**
Average	**53.6**	**48.3**	**30.8**	**23.9**	**27.4**	**18.8**	**11.6**	**6.7**

TABLE XI Ratio of PUs (%) of which Different SDC Offset are Checked for [26] and Proposed EDSDC

Sequences	Lee's method [26] with [25] enabled				Proposed EDSDC with [25] enabled			
Offsets	-1	1	2	-2	-1	1	2	-2
Kendo	23.2	23.2	5.5	4.5	15.6	15.6	5.9	4.9
Street	26.7	26.7	15.4	8.0	15.1	15.1	11.0	5.1
Fly	21.7	21.7	6.5	3.2	9.3	9.3	6.8	3.5
T-Avg.	**23.9**	**23.9**	**9.1**	**5.2**	**13.4**	**13.4**	**7.9**	**4.5**
Newspaper	27.0	27.0	9.4	8.3	22.7	22.7	9.9	8.9
Balloons	24.3	24.3	7.8	5.1	17.7	17.7	8.3	5.5
Hall2	18.9	18.9	2.4	1.9	6.9	6.9	2.6	2.2
Dancer	18.8	18.8	3.4	2.1	5.6	5.6	3.6	2.3
Shark	23.0	23.0	8.6	4.4	17.8	17.8	13.8	7.2
V-Avg.	**22.4**	**22.4**	**6.3**	**4.4**	**14.2**	**14.2**	**7.6**	**5.2**
Average	**23.0**	**23.0**	**7.4**	**4.7**	**13.9**	**13.9**	**7.7**	**4.9**

addition, HTM16.1 has already removed some DMMs and reduced the number of wedgelet pattern candidates. Hence, the time reduction of these methods is very limited. Saldanha's [40], Sanchez's [41], and Zhang's [34] also focus on the intra mode decision part, including DIS, rough mode, and DMM. The time reduction of all these block-level schemes are limited as shown in Table XII. Mora's [29] achieves significant time reduction of over 50% with large ΔBDBR 3.63% by terminating CU partitioning process earlier. In contrast, the proposed framework achieves 16% more time reduction, while the increase in ΔBDBR is only kept around 1%. In [29], the quadtree-depth level for the depth map are restricted by the co-located texture CUs. Due to the mismatch between the depth map and texture image, this method sometimes damages the depth map quality in the mismatch regions. As a result, the quality of synthesized views is degraded significantly. Besides, the proposed algorithm provides similar performance in T-set and V-set. The increase in ΔBDBR for V-set is slightly higher than T-set. But it could still be concluded the corner point selection trained by T-set also works well for other sequences.

As mentioned in Section II, the features used in [19], [21], [23], [29], [30], [34], [40], and [41] either speed up the quadtree decision or intra mode decision. All these fast encoding strategies only focus on a specific part of the encoding process, such as CU decision or mode decision, or more specifically in a subcategory, DMM decision. To demonstrate the superiority of the proposed algorithm, Table XIII further shows the comparison between our algorithm and some of the recent fast depth intra coding algorithms in [19], [21], [23], [26], [28]-[30] , [34], [40], and [41] according to the categories with the adoption of different fast encoding strategies. It is observed that the proposed QDLS provides good balance between time saving and ΔBDBR in the category of CU decision, as compared with [29]-[30]. No previous algorithm is specifically designed for fast PU decision. In intra mode decision, the

TABLE XII Performance of the Overall Algorithm Compared with Previous Works (%). N/A in [40] means these sequences are used in training only, but not for testing.

Sequences	Mora's [29]		Park's [21]		Gu's [19]		Zhang's [23]		Saldanha [40]		Sanchez [41]		Zhang's[34]		Proposed	
	ΔBDBR	ΔBDBR	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT
Kendo	4.19	-54.73	0.13	-11.57	0.63	-8.78	0.97	-9.02	0.30	-23.2	0.37	-33.9	0.43	-38.23	0.70	-64.54
Street	1.52	-50.44	0.00	+5.31	0.74	-9.15	0.79	-10.94	0.15	-30.0	0.22	-41.7	0.29	-39.67	0.66	-67.96
Fly	6.06	-49.08	0.08	-7.84	0.50	-10.51	0.51	-10.16	N/A	N/A	0.12	-40.6	0.21	-43.14	0.45	-76.53
T-Avg.	**3.92**	**-51.41**	**0.07**	**-4.7**	**0.62**	**-9.48**	**0.76**	**-10.04**	**0.23**	**-26.6**	**0.24**	**-38.7**	**0.31**	**-40.35**	**0.60**	**-69.68**
Newspaper	4.40	-45.64	0.24	-5.45	1.50	-11.52	1.31	-10.61	0.26	-23.2	0.46	-35.4	0.89	-33.85	0.71	-49.81
Balloons	5.47	-45.13	0.22	-9.19	1.01	-9.62	1.28	-14.12	0.21	-26.5	0.36	-34.1	0.67	-35.49	0.75	-58.90
Hall2	2.77	-63.59	0.52	-17.13	0.71	-12.83	1.46	-16.55	0.39	-30.2	0.43	-38.8	0.75	-48.33	2.89	-80.35
Dancer	0.79	-46.59	0.18	-6.65	0.62	-8.24	0.94	-15	N/A	N/A	0.12	-38.5	0.49	-41.57	1.24	-71.57
Shark	3.87	-52.21	0.00	+2.85	0.89	-9.35	0.85	-10.04	0.13	-28.4	0.11	-36.2	0.33	-34.16	1.25	-62.84
V-Avg.	**3.46**	**-50.63**	**0.23**	**-7.11**	**0.95**	**-10.31**	**1.17**	**-13.33**	**0.25**	**-27.1**	**0.30**	**-36.6**	**0.63**	**-38.68**	**1.37**	**-64.69**
Average	**3.63**	**-50.93**	**0.17**	**-6.21**	**0.83**	**-10.00**	**1.01**	**-12.10**	**0.24**	**-26.9**	**0.27**	**-37.4**	**0.51**	**-39.31**	**1.08**	**-66.56**

TABLE XIII Performance of Proposed Algorithms Compared with Previous Works in different categories

Category		Works	ΔBDBR(%)	ΔT(%)
CU Decision		Kim's [30]	0.15	-28.97
		Mora's [29]	3.63	-50.93
		Proposed QDLS	**0.46**	**-46.93**
PU Decision		**Proposed PUD**	**0.02**	**-17.90**
Mode Decision	RM Decision	**Proposed RMP**	**0.17**	**-19.11**
	DMM Decision	Gu's [19]	0.83	-10.00
		Zhang's [23]	1.01	-12.10
		Park's [21]	0.17	-6.21
		Proposed DMM	**0.13**	**-15.50**
	SDC Decision	Lee's[26]	0.00	-7.40
		Proposed SDC	**0.02**	**-6.14**
	Overall Intra Mode Decision	Our [28]	0.94	-37.21
		Saldanha [40]	0.24	-26.90
		Sanchez [41]	0.27	-37.40
		Zhang's [34]	0.51	-39.31
		Proposed FIMD (RMP+DMM+SDC)	**0.48**	**-41.75**
CU+PU+ Mode decision		**Proposed Overall**	**1.08**	**-66.56**

proposed FIMD is composed of RMP, DMM, and SDC and they also perform well in all sub-categories. For instance, the proposed DMM skipping scheme remarkably outperforms other fast DMM decision schemes in [19], [21] and [23]. The comparative results further validate that the feature CP works better than other features such as variance [19] and horizontal/vertical edges [21]. For the intra mode decision part, the proposed FIMD outperforms most of them [28] [40], [41], [34]. Table XIII verifies that the proposed algorithms based on CPs extracted from depth maps are effective and superior to the existing algorithms in different coding categories for the 3D-HEVC encoder. Therefore, it can be concluded that a two-dimensional gradient feature, CP, can benefit both CU/PU decision and intra mode decision. Besides, the proposed algorithm could provide the tradeoff between ΔBDBR and time reduction by adjusting the number of CPs as shown in Fig.13. As the number of CPs decreases, the proposed algorithm could provide more time saving with more drop in the synthesized view quality (a decrease in ΔBDBR).

Since the complexity and processing time reduction are especially important for embedded systems, we will address this issue in the following discussion to show the practicality of the proposed algorithm. Some algorithms may introduce additional difficulty in the design of parallel processing. For example, the work in [28] utilizes the pixels information of neighboring blocks for fast mode decision. The dependence among neighboring blocks in [28] could make parallel processing more difficult. Besides, the algorithms in [29][40] employ the quadtree and mode information of the corresponding texture video as features. These features restrict

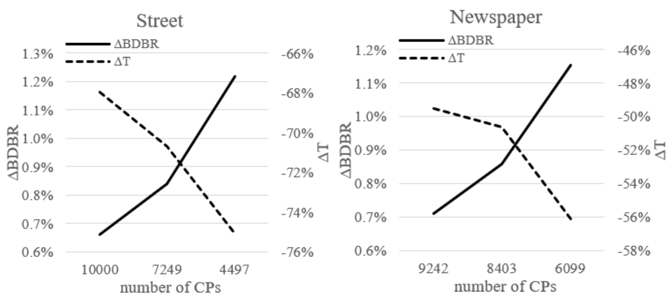


Fig.13: Tradeoff between the ΔBDBR and time reduction.

the coding order of texture and depth video and result in additional memory communication between texture and depth coding. In the proposed algorithm, there is no data dependency between the current block being processed and its neighboring block, which implies that it will not prevent the use of parallel processing. Besides, the data independence of the texture and depth coding of the proposed algorithm results in no additional memory communication between them. Generally, the proposed algorithm will not introduce further difficulty in parallelism and memory communication. The only additional memory requirement is for the storage of CPs. The CP information is stored in the form of PDL matrix, as shown in (11). For each 4×4 block, one $pdl(i,j)$ from 0 to 5 is stored and 3 bits are required for each $pdl(i,j)$. Therefore, the size of PDL matrix of the whole frame is 3× (frame width/4)×(frame height/4) bits. When the PDL matrix is generated for each LCU, the size is only 3×16×16=768 bits, which is negligible.

V. Conclusions

In this paper, a good feature - corner point in computer vision has been introduced for speeding up the depth intra coding in 3D-HEVC. Three fast techniques are developed by this good feature. They include quadtree-depth limited strategy, PU partition decision and fast depth intra mode decision. These proposed techniques study the correlations between the CPs and encoding modes based on the statistical analysis, given by the assumption that a CP can detect multi-directional edges and highly related to CU and PU block size and dominant intra mode. We are the first to introduce a good feature for both CU/PU decision and intra mode decision. The proposed algorithm can provide a significant time reduction while maintaining ΔBDBR performance. Considering the spatial discontinuity by CPs, our techniques can preserve the sharp edges for key scenario or perceptible objects by using small CUs. The main contribution of this paper is that the good feature/structure for the depth map in computer version is brought into video coding. Actually, this has widespread application prospect where further research works can be inspired by our algorithm, such as adaptive bit-allocation, region-of-interesting coding, and 3D shape coding.

References

[1] K. Müller, P. Merkle, *et al.*, "3D video formats and coding methods," in proc. of ICIP, Hong Kong, pp. 2389-2392, Sept. 2010.

[2] "Call for proposals on 3D video coding technology," ISO/IEC/SC29/WG11/MPEG2011/N12036, Geneva, Switzerland, March, 2011.

[3] K. Muller, H. Schwarz, *et al.*, "3D high-efficiency video coding for multi-view video and depth data," IEEE Trans. on Image Processing, Vol. 22, no. 9, pp. 3366-3378, Sep. 2013.

[4] G. Tech, Y. Chen, *et al.*, "Overview of the multiview and 3D extensions of high efficiency video coding," IEEE Trans. on Circuits and Syst. for Video Tech. vol. 26, no. 1, Jan. 2016.

[5] G.J. Sullivan, J. Ohm, *et al.* "Overview of the high efficiency video coding (HEVC) standard," IEEE Trans. on Circuits and Syst. for Video Tech., vol.22, no.12, pp.1649-1668, Dec. 2012.

[6] K. Muller, P. Merkle, *et al.* "3D video coding with depth modeling modes and view synthesis optimization," in proc. of APSIPA ASC, pp. 1-4, 2012.

[7] J.Y. Lee, M. W. Park, *et al.* "3D-CE1: Depth Intra Skip (DIS) Mode", ITU-T SG16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 11th meeting, document: JCT3V-K0033, Geneva, Switzerland, Feb. 2015.

[8] F. Jager, "Simplified depth map intra coding with an optional depth lookup table", in proc. of IC3D, Dec. 2012.

[9] B. Oh, J. Lee, *et al.* "3D-CE8.h results on view synthesis optimization," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/ WG 11, 1st meeting, SE, July 2012.

[10] D. Marpe, *et al.* "Video compression using nested quadtree structures, leaf merging, and improved techniques for motion representation and entropy coding," IEEE Trans. on Circuits and Syst. for Video Tech., vol. 20, no. 12, pp. 1676– 1687, Dec. 2010.

[11] J. Lainema, F. Bossen, *et al.* "Intra coding of the HEVC standard," IEEE Trans. on Circuits and Syst. for Video Tech., vol.22, no.12, pp.1792-1801, Dec. 2012.

[12] Y. Chen, G. Tech, *et al.* "Test model 10 of 3D-HEVC and MV-HEVC," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-J1003, 10th Meeting, Strasbourg, FR, 18–24 Oct. 2014.

[13] B.T. Oh, K.J. Oh, "View synthesis distortion estimation for AVC- and HEVC-compatible 3-D video coding," IEEE Trans. on Circuits and Syst. for Video Tech., vol.24, no.6, pp.1006-1015, June 2014.

[14] G. Tech, H. Schwarz, *et al.* "3D video coding using the synthesized view distortion change," in proc. of Picture Coding Symposium (PCS), 2012, pp.25-28, 7-9 May 2012.

[15] P. Merkle, K. Muller, *et al.* "Simplified wedgelet search for DMM modes 1 and 3", ITU-T SG 16WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-B0039, Shanghai, China.

[16] M.M. Zhang, C. Zhao, *et al.* "A fast depth-map wedgelet partition scheme for intra prediction in 3D video coding," in proc. of ISCAS, Beijing, China, pp. 2852-2855, May 2013.

[17] C.H. Fu, H.B. Zhang, *et al.* "Fast wedgelet pattern decision for DMM in 3D-HEVC," in proc. of DSP, Singapore, pp. 447-481, 2015.

[18] Z. Gu, J. Zheng, *et al.* "Fast intra prediction mode selection for intra depth map coding," document JCT3V-E0238, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Aug. 2013

[19] Z. Gu, J. Zheng, *et al.* "Fast bi-partition mode selection for 3D HEVC depth intra coding", in proc. of ICME, pp.1-6, 2014.

[20] G. Sanchez, M. Saldanha, *et al.* "Complexity reduction for 3D-HEVC depth maps intra-frame prediction using simplified edge detector algorithm," in proc. of ICIP, pp. 3209-3213, Oct. 2014.

[21] C. Park, "Edge-based intra mode selection for depth-map coding in 3D-HEVC," IEEE Trans. on Image Processing, Vol. 24, no. 1, Jan. 2015.

[22] T. da Silva, L. Agostini, *et al.* "Complexity reduction of depth intra coding for 3D video extension of HEVC," in proc. of VCIP, pp.229-232, Dec 2014.

[23] Q. W. Zhang, N. N. Li, *et al.* "Effective early terminate algorithm for depth map intra coding in 3D-HEVC," IEEE Electronics Letters, vol. 50, no. 14, pp. 994-996, Jul. 2014.

[24] C. Park, "Efficient intra-mode decision algorithm skipping unnecessary depth-modelling modes in 3D-HEVC," IEEE Electronics Letters, vol. 51, no. 10, pp. 756-758, May. 2015.

[25] Z. Y. Gu, J. H. Zheng, *et al.* "Fast intra SDC coding for 3D-HEVC intra coding," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-I0123, 9th Meeting, Sapporo JP, July 2014.

[26] J. Y. Lee, M. W. Park, *et al.* "3D-CE2 related: Fast SDC DC offset decision," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-I0084, 9th Meeting, Sapporo JP, Jul. 2014.

[27] H.B. Zhang, C.H. Fu, *et al.* "Adaptive fast intra mode decision of depth map coding by low complexity RD-Cost in 3D-HEVC," in proc. of DSP, Singapore, pp. 487-491, 2015.

[28] H.B. Zhang, C.H. Fu, *et al.* "Efficient depth intra mode decision by reference pixels classification in 3D-HEVC," in proc. of ICIP, Quebec, Canada, pp.961-965, 27-30 Sept. 2015.

[29] E.G. Mora, J. Jung, *et al.* "Initialization, limitation, and predictive coding of the depth and texture quadtree in 3D-HEVC," IEEE Trans. on Circuits and Syst. for Video Tech., vol. 24, No.9, pp. 1554-1565, Sept. 2014.

[30] M. Kim., N. Ling., *et al.* "Fast single depth intra mode decision for depth map coding in 3D-HEVC", in proc. of ICMEW, p.1-6, 2015.

[31] H. Lei, A. Wang, *et al.* "Early termination algorithm for the depth modeling mode in three-dimensional extension of high-efficiency video coding," Journal of Electronic Imaging, vol. 25, no. 5, pp. 050503, 2016.

[32] J. Lei, Z. Sun, *et al.* "Simplified search algorithm for explicit wedgelet signalization mode in 3D-HEVC," in proc. of ICME, Hong Kong, China, pp.805-810, 2017.

[33] G. Sanchez, L. Agostini, *et al.* "Complexity reduction by modes reduction in RD-List for intra-frame prediction in 3D-HEVC depth maps," in proc. of ISCAS, Baltimore, MD, USA, pp.1-4, 2017.

[34] H. B. Zhang, C. H. Fu, *et al.* "Probability-based depth intra mode skipping strategy and novel VSO metric for DMM decision in 3D-HEVC," IEEE Trans. on Circuits and Syst. for Video Tech., vol. 28, No.2, pp. 513-527, 2018.

[35] L. Zhao, L. Zhang, *et al.* "Fast mode decision algorithm for intra prediction in HEVC," in proc. of VCIP, pp. 1- 4, 2011.

[36] 3D-HEVC software HTM-16.1 [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn 3DVCSoftware /tags/HTM-16.1

[37] H.B. Zhang, Y.L. Chan, *et al.* "Quadtree decision for depth intra coding in 3D-HEVC by good feature," in proc. of ICASSP, Shanghai, China, pp.1481-1485, 2016.

[38] Q. Zhang, N. Zhang, *et al.* "Fast depth map mode decision based on depth–texture correlation and edge classification for 3D-HEVC," Journal of Visual Communication and Image Representation, vol. 45(C), pp. 170-180, May, 2017.

[39] Z. Peng, H. Han, *et al.* "Joint processing and fast encoding algorithm for multi-view depth video," EURASIP Journal on Image and Video Processing, vol. 2016, no. 1, pp. 24, 2016.

[40] M. Saldanha, G. Sanchez, *et al.* "Block-level fast coding scheme for depth maps in three-dimensional high efficiency video coding," J. Electron. Imag., Art. no. 010502, vol. 27, no. 1, Feb. 2018,.

[41] G. Sanchez, L. Agostini, *et al.* "A reduced computational effort mode-level scheme for 3D-HEVC depth maps intra-frame prediction," J. Vis. Commun. Image Represent., vol. 54, no. 7, pp. 193–203, Jul. 2018.

[42] J.B. Shi, and C. Tomasi, "Good features to track," in proc. of CVPR, Seattle, WA, pp. 593-600, June 1994.

[43] C. Harris, and M. Stephens, "A combined corner and edge detector". in proc. of Fourth Alvey Vision Conference, Manchester, pp. 147-152,1998.

[44] C. Zhu, Y. Zhao, *et al.* "3D-TV system with depth-image-based rendering-architectures, techniques and challenges," Springer-Verlag New York, 2013 [doi: 10.1007/978-1-4419-9964-1].

[45] C. Rosewarne, B. Bross, "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 7," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/ WG 11 JCTVC-Y1002, October 2016.

[46] K. Muller and A. Vetro, "Common test conditions of 3DV core experiments," in Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) of ITU-T VCEG and ISO/IEC MPEG, documentJCT3V-G1100, Jan. 2014.

[47] G. Bjontegaard, "Calculation of average PSNR differences between RD curves," ITU-T Video Coding Experts Group (VCEG), 2001.