IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 7, 2019, accepted November 22, 2019, date of publication December 2, 2019, date of current version December 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956994

Fast Depth Intra Coding Based on Decision Tree in 3D-HEVC

CHANG-HONG FU[1], (Member, IEEE), HAO CHEN[1], YUI-LAM CHAN[2], (Member, IEEE), SIK-HO TSANG[2], (Member, IEEE), HONG HONG[1], (Member, IEEE), AND XIAOHUA ZHU[1], (Member, IEEE)

[1]School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210019, China
[2]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong

Corresponding author: Chang-Hong Fu (enchfu@njust.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 30917011319, in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant PolyU 152069/18E, in part by the National Natural Science Foundation of China under Grant 61301109, Grant 61871224 and Grant 81601568, and in part by the China Scholarship Council under Grant 201806845028.

ABSTRACT The coding units (CU) partitioning in the 3D extension of the high efficiency video coding standard (3D-HEVC) is recursively conducted on different block sizes from 64×64 to 8×8. Besides, the depth coding in 3D-HEVC introduces several new coding tools for each CU to improve the coding efficiency, however, with great computational complexity. It is noted that only a small number of the CUs in recursive partitioning are encoded into the final bitstream. Among these CUs, the CUs with Intra 2N $\times$ 2N or Intra N $\times$ N as optimal modes have a very small proportion. In this paper, we thus propose an early determination of depth intra coding, where the coding stage of Intra 2N $\times$ 2N, Intra N $\times$ N or CU Splitting for the CUs could be early skipped based on several Decision Trees. Simulation results show that, with restrict the results from the tree leaves by different Gini thresholds, the proposed algorithm could save 41.14%-71.63% of the depth coding time with only a slight increase in BDBR.

INDEX TERMS 3D-HEVC, CU size decision, decision tree, Gini thresholds, depth intra coding, fast mode decision.

I. INTRODUCTION

As 3D video services are getting increasingly popular, an efficient way to better represent and compress 3D data has been actively investigated in recent years. The new multi-view plus depth (MVD) video is one of the most emerging formats for the next-generation 3D video system, in which 3D video is represented by a limited number of textures and associated depth maps. To reconstruct the 3D scene, additional virtual views can be generated via a depth image-based rendering (DIBR) technique. Since depth maps are used to provide the disparity and guide the synthesis process, they play a crucial role in the current 3D video system and should be precisely encoded.

Depth intra coding in 3D-HEVC basically adopts the flexible coding quad-tree structure in HEVC, where the block partitioning allows a coding tree unit (CTU, 64×64) recursively splitting into four coding units (CU) of equal size. In addition to the conventional 35 HEVC intra modes (CHIM), several new depth intra coding tools such as depth intra skip mode (DIS), depth modelling mode (DMM), segment-wise DC coding (SDC) and view synthesis optimization (VSO) have been designed for each CU in depth map coding. These new techniques improve coding efficiency at the expense of dramatically increasing computational complexity.

In order to reduce the complexity of depth intra coding, many fast approaches have appeared [1]–[13]. They could be roughly divided into two categories: fast mode decision methods [1]–[9] and fast coding block partition methods [10]–[13]. In [1], an edge-based DMM skipping strategy was proposed in Hadamard transform domain. In [2], [3], the process of DMM search or SDC decision is selectively skipped by comparing the rate distortion (RD) cost of the prior checked modes. In [4], the distortion calculation in DMM search is simplified as the squared Euclidean distance of variances (SEDV), and a probability-based early

The associate editor coordinating the review of this manuscript and approving it for publication was Donatella Darsena.

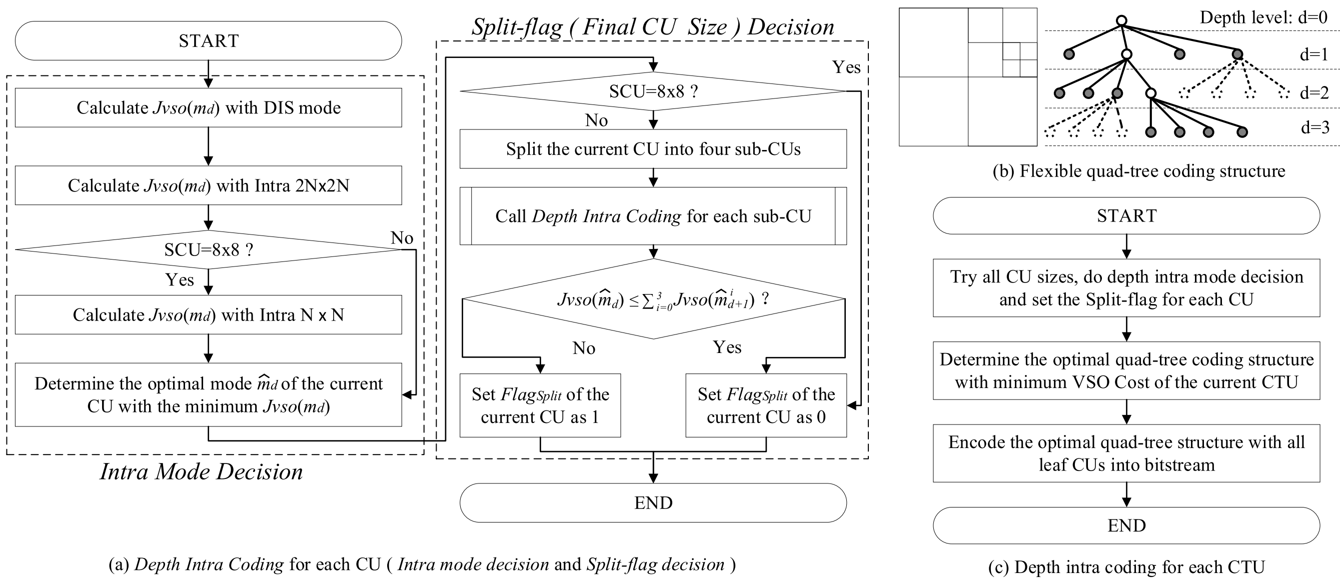


FIGURE 1. **(a) Depth intra coding for each CU; (b) Optimal coding quad-tree structure; (c) Depth intra coding for each CTU.**

decision (PBED) method was proposed by studying the correlation between rough mode cost and final mode decision. Other algorithms for block level fast intra mode decision could be found in [5]–[9], which mainly focus on the acceleration of complex modes such as DMM. On the other hand, a coding block quad-tree pruning algorithm was designed in [10] by considering the variance of blocks and the estimated distortion of single depth intra mode [15]. In [11], an inter-component coding tool was developed mainly for inter frames, where the depth coding quad-tree is limited to the coded texture quad-tree. In [12], the DIS mode correlation between coding levels is employed to speed up the partition decision processing. These methods always focus on the specific coding technique in certain coding stage, such as DMM or SDC mode decision in Intra 2N×2N or Intra N×N mode, and early decision for CU partition termination. However, there are usually some coding stages that are not useful in the final bitstream. For example, if the current CU is finally coded with DIS, then the Intra 2N×2N mode is unnecessary to be checked. In this case, skipping Intra 2N×2N can obviously achieve greater reduction than the traditional fast DMM or SDC decisions in the coding stage of Intra 2N×2N. Besides, coding mode complexity [13] and texture complexity [14] are analyzed respectively to accelerate the intra mode decision and skip unnecessary intra prediction size together. The features are manually selected in these works. Machine learning methods are seldom considered in these fast coding methods for depth maps.

There are several prior works utilizing decision tree for speeding up the coding process of HEVC [22]–[27]. However, most of them focus on the texture video coding of HEVC [22]–[25] or HEVC extension such as Screen Content Coding (SCC) [26], [27]. In this paper, we propose a more flexible coding framework by inserting the decision tree based classifiers for depth intra coding. By statistical analysis, we will first show that, for a number of CUs, the coding stage of Intra 2N×2N checking, Intra N×N checking or CU Splitting is useless. Whether to check or skip these coding stages are then considered as three binary classifications. Here, Decision Tree is employed to learn and do the classifications at each depth level. Experimental results show that, with pruning the leaves of the Decision Trees by different Gini thresholds, the proposed algorithm can achieve 41.14%-71.63% of time reduction on average with limited BDBR increase.

The rest of this paper is organized as follows. In Section II, the background of depth intra coding in 3D-HEVC is reviewed. Besides, the motivation of this paper is also described in this section. The proposed decisions based on Decision Tree are presented in Section III. The experimental results and conclusion are given in Section IV and Section V, respectively.

II. BACKGROUND AND MOTIVATION

In this Section, depth intra coding in 3D-HEVC is reviewed. Then, the motivation of this paper is introduced.

A. DEPTH INTRA CODING IN 3D-HEVC

In 3D-HEVC, a depth map is divided into several CTUs. Starting from CTU, the quad-tree partitioning allows the blocks recursively splitting into four equally sized CUs until the minimum size (SCU, 8 × 8) is reached. For each CU, as shown in Figure 1(a), it is first divided into several regions called prediction units (PU) sharing the same prediction mode, including DIS, Intra 2N × 2N and Intra N × N. Particularly, the PU of Intra N×N is available only in the SCU.

Each mode will be checked by the VSO cost calculation, and The VSO cost $J_{VSO}(m_d)$ is computed by the distortion $D_{VSO}(m_d)$ plus the Langrangian multiplier λ times the coding rate $R(m_d)$, as follows.

$$J_{VSO}(m_d) = D_{VSO}(m_d) + \lambda \times R(m_d) \quad (1)$$

where d is the current depth level and m_d is one of the intra modes in M_d, which is the set of intra modes defined as

$$M_d = \{DIS, Intra\ 2N \times 2N, Intra\ N \times N \quad (2)$$

The mode with the smallest VSO cost will be determined as the optimal PU mode, $\widehat{m}_d$, for the current CU, as follows.

$$\widehat{m}_d = argmin_{m_d \in M_d}(J_{VSO}(m_d)) \quad (3)$$

The CU splitting function is then conducted to obtain the optimal VSO cost of smaller sub-CUs. After all possible CU block sizes are performed, the quad-tree coding structure with the smallest VSO cost is decided as the optimal coding structure as shown in Figure 1(b). Whether to split, or not, for each CU in the final quad-tree is determined by the VSO cost of the current CU, $J_{VSO}(\widehat{m}_d)$, and the sum of VSO costs of its four sub-CUs, $\sum_{i=0}^{3} J_{VSO}(\widehat{m}^{i}_{d+1})$, according to

$$Flag_{Split} = \begin{cases} 0, \text{ if } J_{VSO}(\widehat{m}_d) \leq \sum_{i=0}^{3} J_{VSO}(\widehat{m}^{i}_{d+1}) \\ 1, \text{ if } J_{VSO}(\widehat{m}_d) > \sum_{i=0}^{3} J_{VSO}(\widehat{m}^{i}_{d+1}) \end{cases} \quad (4)$$

where the $Flag_{Split}$ of 0 and 1 represent Non-Split and Split, respectively, for the current CU in the final bitstream.

The final quad-tree structure is formed by the leaf nodes in gray as shown in Figure1(b). On the contrary, white nodes are further split and dashed nodes are pruned (only part of dashes nodes are shown due to space limitation). Finally, the optimal quad-tree coding structure, with optimal mode for each leaf CU, is then encoded into bitstream, as described in Figure 1(c). More details could be found in [16].

B. STATISTICAL ANALYSIS AND MOTIVATION

As shown in Figure 1(c), although all CUs with different block sizes are checked, only those CUs in the optimal coding quad-tree are encoded in the final bitstream. In order to analyze the final bitstream, we conducted some experiments on several sequences with all-intra configuration, where the coding condition is detailly described in Section V. Table 1 shows the distributions of different CU sizes and different depth intra modes at each depth level in the final bitstreams of various sequences.

As we can see, the average percentage of CUs in the final bitstream decreases sharply from 70.51% to 2.13% as the depth level increases. In other words, most CUs with small block sizes checked in the depth intra coding process are redundant. At each depth level, most of the CUs, 80% of 64×64 CU (56.92% / 70.51%) and 61% of 8×8 CU (1.31% / 2.13%), select DIS as the optimal prediction modes. The CUs with Intra 2N×2N occupy a small proportion from 13.60% to 0.67%. Besides, the SCUs with Intra N×N have a percentage of 0.15% on average. The small proportion of CUs

TABLE 1. Distribution of depth intra modes, and cu sizes in the final bitstream.

Test sequence	Block sizes	CU (%)	Depth intra mode (%)		
			DIS	INTRA 2N×2N	INTRA N×N
Kendo	64×64	58.22	42.85	15.37	/
	32×32	28.69	16.43	12.25	/
	16×16	9.79	5.83	3.97	/
	8 × 8	3.30	1.90	1.15	0.25
Poznan_Hall2	64×64	85.40	75.73	9.67	/
	32×32	11.28	8.25	3.03	/
	16×16	2.68	1.98	0.71	/
	8 × 8	0.63	0.40	0.19	0.05
Undo_Dancer	64×64	67.93	54.25	13.68	/
	32×32	21.70	15.72	5.98	/
	16×16	7.90	5.75	2.16	/
	8 × 8	2.46	1.58	0.73	0.16
Average	64×64	70.51	56.92	13.60	/
	32×32	20.56	13.90	6.66	/
	16×16	6.79	4.66	2.13	/
	8 × 8	2.13	1.31	0.67	0.15

finally encoded by Intra 2N×2N or Intra N×N also indicates that checking the VSO cost with Intra 2N×2N or Intra N×N for most CUs is unnecessary since these two intra modes are not coded in final bitstream.

Therefore, for a number of CUs, skipping one or more coding stages of Intra 2N×2N checking, Intra N×N checking and CU splitting can significantly reduce the computational complexity and do not affect the final bitstream.

III. PROPOSED ALGORITHMS

As shown in the statistical analysis in Section II, not all coding stages of Intra 2N×2N, Intra N×N and CU splitting for most CUs have contributions to the final bitstream. In this Section, we propose to adopt several decision tree based classifications to decide whether to check or skip the coding stages of Intra 2N×2N, Intra N×N and CU splitting in the depth intra coding process for all CU sizes.

A. DECISION TREE BASED DEPTH INTRA CODING FRAMEWORK AND ITS CONTRIBUTIONS

In the proposed algorithm shown in Figure 2, we propose a more flexible coding framework by inserting the decision tree based classifiers before checking Intra 2N×2N, Intra N×N, and CU splitting. Compared with the prior work, the contributions of this paper can be summarized as:

1) The new coding structure of depth intra coding considers all coding stages one by one in the cascaded way which has the following designs: (a) DIS mode is separated from the mode decision part due to its high percentage in the final bitstream and low complexity, (b) The CU partitioning process is illustrated in the way of split-flag decision and uncoupled from the intra mode decision, as depicted in Figure 2, and (c) The cascaded structure makes it easy to involve decision trees in the encoding process compared to the usual practice where intra mode decision is included in the CU partition

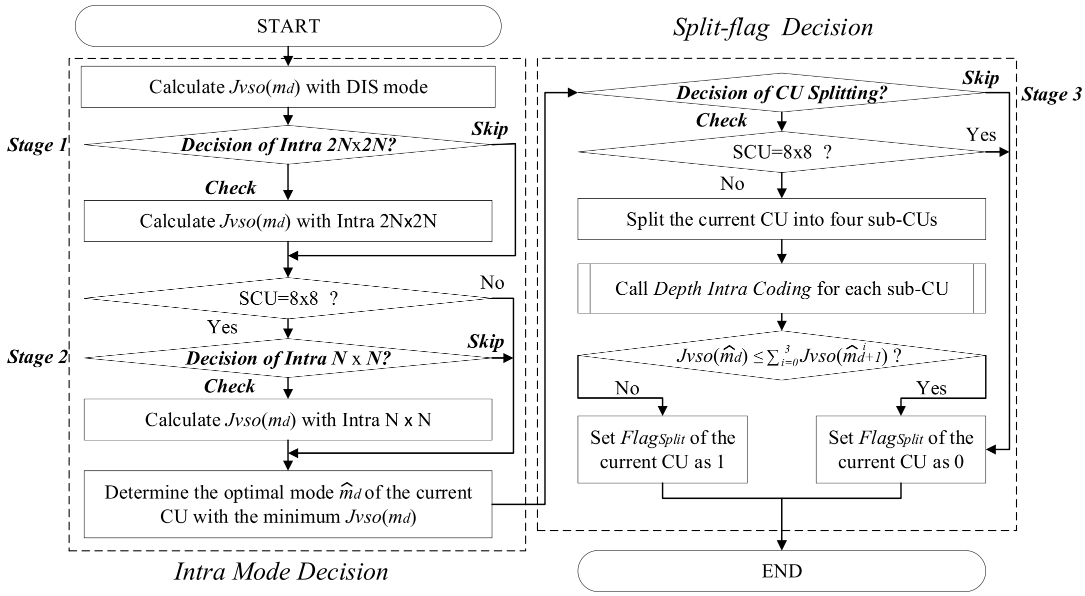


FIGURE 2. Proposed flow of depth intra coding for each CU.

TABLE 2. Selected coding stages.

i	1	2	3
Coding Stage	Intra 2N×2N	Intra N×N	CU Splitting

process. Three binary classifications are then proposed at different stages.

2) The proposed cascaded structure facilitates the use of the most direct features from the best $J_{VSO}(\widehat{m}_d)$ so far. These direct features vary as a CU goes through different classifiers, which provides more precise intermediate coding information of a CU to the classifiers, resulting in accurate decisions.

3) The CU sample selection from different partition levels of depth intra coding and the influence on the training data distribution will also be discussed. It is noted that the sample selection part is crucial to any machine learning method including a decision tree.

4) Gini values are involved to improve flexibility. The proposed algorithm could provide a different tradeoff between BDBR and time saving.

For the sake of convenience, we name the coding stages of Intra 2N×2N, Intra N×N and CU splitting as coding stages 1, 2, 3, as shown in Table 2. For coding stage $i(i = 1, 2, 3)$, we need to determine whether to check or skip it at each depth level. Therefore, three binary classifiers are utilized for each CU size.

As illustrated in Figure 2, the bold part is from the proposed decisions, while the others are original steps in 3D-HEVC. Before conducting coding stage $i(i = 1, 2, 3)$, a corresponding classifier is utilized to determine whether to check or skip the coding stage i. If the decision result is to check, the current CU will conduct the coding stage i; if the decision result is to skip, the coding stage i for the current CU will be skipped.

It is noted that these three classifiers are conducted on different ranges of depth levels. The decision of coding stage 1 can be conducted for all CU sizes. However, the decision of coding stage 2 can only be conducted for SCUs, since Intra N×N is only available for SCUs. Besides, the decision of coding stage 3 is conducted from the depth level of 0 to depth level of 2, where CU partitioning is allowed to be conducted for depth maps in 3D-HEVC.

B. SAMPLE COLLECTION

It is well-known that the sample collection and feature selection are both important to train a classifier in terms of prediction accuracy. We first discuss the sample collection. The samples here are the CUs in depth intra coding for all CU sizes. Theoretically, the sample space needs to be comprehensive and can cover all possible samples of the coding data. However, in the proposed sample collection, only the CUs which are coded as the final bitstream are collected as samples for training in Stage 1 and Stage 2. As shown in Figure1(b), there are 85 $(1 + 4 + 16 + 64 = 85)$ possible nodes in each CTU, if the full quadtree is exhaustively searched. On the contrary, only 10 leaf nodes (gray nodes in the example) are decided as the final quadtree by the RD optimization shown in Figure1(a).

In this example, only 10 gray leaf nodes in the final quadtree are collected as samples of this CTU. The reason behind is that the distribution of features such as VSO cost, distortion and bits in the optimal leaf nodes is quite different from those unemployed white nodes and dashed nodes. For example, white nodes which will be further split into sub-CUs usually have much larger RD cost than the gray nodes in the same depth level. Collecting all nodes including unemployed nodes will affect the desired data distribution.

TABLE 3. Samples collection.

Decisions	CU Samples	Numbers	Types	Samples Descriptions
Coding stage 1	CUs in the final bitstream	248 268	*Skip*	$\widehat{m}_d \neq Intra\ 2N \times 2N$ after checking Intra 2N×2N
			Check	$\widehat{m}_d = Intra\ 2N \times 2N$ after checking Intra 2N×2N
Coding stage 2	CUs in the final bitstream	134 228	*Skip*	$\widehat{m}_d \neq Intra\ N \times N$ after checking Intra N×N
			Check	$\widehat{m}_d = Intra\ N \times N$ after checking Intra N×N
Coding stage 3	CUs with $Flag_{Split} = 1$ and CUs with $Flag_{Split} = 0$, (d=0~2)	192 564	*Skip*	$Flag_{Split} = 0$ after checking all sub-CUs split from the current CU
			Check	$Flag_{Split} = 1$ after checking all sub-CUs split from the current CU

TABLE 4. Importance of different features.

		$J_{VSO}(\widehat{m}_d)$	$D_{VSO}(\widehat{m}_d)$	$R(\widehat{m}_d)$	Var	SAG_{Hor}	SAG_{Ver}	$NumCheck_{neighbors}$	$Flag_{texture}$
Stage 1	64x64	0.003	0.837	0.145	0.000	0.009	0.006	0.000	N/A
	32x32	0.013	0.716	0.265	0.000	0.006	0.000	0.000	N/A
	16x16	0.009	0.797	0.194	0.000	0.000	0.000	0.000	N/A
	8x8	0.000	0.936	0.063	0.000	0.000	0.000	0.000	N/A
Stage 2	8x8	0.027	0.730	0.244	0.000	0.000	0.000	0.000	0.000
Stage 3	64x64	0.000	0.167	0.817	0.000	0.000	0.000	0.015	0.000
	32x32	0.000	0.132	0.806	0.000	0.000	0.000	0.061	0.001
	16x16	0.000	0.254	0.706	0.000	0.000	0.000	0.041	0.000

Since the trained decision tree is desired to only model the intra mode decision of gray leaf nodes in Stage 1 and Stage 2, samples are only collected from optimal gray nodes instead of all of them.

Based on the discussion above, the rules of sample collection for Stage 1 and Stage 2 are concluded in Table 3. For Stage 1, positive samples are those gray nodes in Figure 1(b), where optimal mode, $\widehat{m}_d$, does not belong to $Intra\ 2N \times 2N$, that means $Intra\ 2N \times 2N$ could be skipped safely in Stage 1. These positive samples are labeled as "Skip". On the contrary, those gray nodes where the final optimal mode belongs to $Intra\ 2N \times 2N$ are labeled as "Check", i.e. negative samples which should not be skipped by Stage 1. The philosophy of labeling samples is almost the same at Stage 2. The only difference is that the gray nodes at depth level 3 are collected since $Intra\ N \times N$ is only available for SCUs as mentioned before. As a result, the number of samples at Stage 2 is much smaller than that of Stage1 as shown in Table 3.

For Stage 3, since CU partitioning is only considered on nodes from the depth levels of 0 to 2, the gray leaf nodes in the depth levels of 0 to 2 are collected as positive samples. In these nodes, no further split is required with $Flag_{Split} = 0$. The positive samples are labeled as "*Skip*". On the contrary, the white nodes with $Flag_{Split} = 1$ are collected as negative samples and labeled as "*Check*". In other words, CU splitting cannot be skipped at these nodes.

C. FEATURE SELECTION

The precision of the mode and CU splitting decisions in a classification task is highly dependent on the feature space used to train the classifiers. Therefore, the features should be carefully selected. In prior work of fast depth intra coding, various features could be extracted even they do not belong to the machine learning approach. The features include the edge information [1], utilization of rough mode [3][4], variance of blocks [10], texture quadtree [11] etc. The disadvantage is that they are not directly related to the rate-distortion optimization function, which determines the final mode and partition. As mentioned in (3) and (4), the optimal mode $\widehat{m}_d$ or the split information $Flag_{Split}$ for each CU is directly related with the minimum VSO cost, $J_{VSO}(\widehat{m}_d)$. However, it is impossible to utilize $J_{VSO}(\widehat{m}_d)$ as a feature directly in the previous fast algorithms [1], [3], [4], [10], [11] since the calculation of RDO is very time consuming and not available until the coding process is finished. When the encoding flow is re-organized as the cascaded structure in Figure 2, this design makes it possible to use the best $J_{VSO}(\widehat{m}_d)$ so far without allowing too much complexity of the original encoder. It is considered as the best $J_{VSO}(\widehat{m}_d)$ before checking the target mode, and it varies as a CU goes through different classifiers. For example, in the coding stage 1, $J_{VSO}(\widehat{m}_d = DIS)$ is evaluated at the beginning of the encoding flow and it is separated from the mode decision part. $J_{VSO}(\widehat{m}_d = DIS)$ can be used as the feature in stage 1. The best $J_{VSO}(\widehat{m}_d)$ so far is then dynamically updated when it goes through different classifiers. Therefore, $J_{VSO}(\widehat{m}_d)$ before the coding stage $i (i = 1, 2, 3)$ is selected as one of the features. In addition to $J_{VSO}(\widehat{m}_d)$, the two separate parts of the $J_{VSO}(\widehat{m}_d)$ calculation, the distortion $D_{VSO}(\widehat{m}_d)$ and bits $R(\widehat{m}_d)$, are also utilized.

Besides, some common features are utilized for training together with RDO features. The resulting *feature_importance* from the machine learning package Scikit-learn [19] is shown in Table 4. *Var* means the CU variance representing the smoothness of the current CU. SAG_{Hor} and SAG_{Ver} are calculated by the sum of absolute gradient of CU in horizontal and vertical respectively. $NumCheck_{neighbors}$ counts the number of *Check* decisions made in neighbor blocks (*Above, AboveLeft, AboveRight, BelowLeft, Left*). $Flag_{texture}$

TABLE 5. Selected features and descriptions.

Decisions	ID	Features	Feature Descriptions
Decision for Coding stage i, (i=1,2,3)	1	$R(\widehat{m}_d)$	$R(\widehat{m}_d)$ with the current optimal mode $\widehat{m}_d$ before checking the coding stage i
	2	$D_{VSO}(\widehat{m}_d)$	$D_{VSO}(\widehat{m}_d)$ with the current optimal mode $\widehat{m}_d$ before checking the coding stage i
	3	$J_{VSO}(\widehat{m}_d)$,	$J_{VSO}(\widehat{m}_d)$ with the current optimal mode $\widehat{m}_d$ before checking the coding stage i

TABLE 6. Test sequences.

Type	Resolution	Test Sequence	Frame Rate	No. of Frames
Training	1024x768	Newspaper[T]	30	300
Evaluation	1024x768	Balloons[E]	30	300
		Kendo[E]	30	300
		Newspaper[T, E]	30	300
	1920x1088	GT_Fly[E]	25	250
		Poznan_Hall[E]	25	200
		Poznan_Street[E]	25	250
		Shark[E]	30	300
		Undo_Dancer[E]	25	250

T, Sequences used to collect training samples;
E, Sequences used to evaluate the performance of the proposed methods;

represents whether the stage is necessary to check in the corresponding texture CU. It is verified in Table 4 that the RDO features are dominant in terms of importance.

Therefore, the final features selected for each classifier in this paper are listed in Table 5. For the decision of the coding stage $i (i = 1, 2, 3)$, the best VSO cost $J_{VSO}(\widehat{m}_d)$ so far and its separate part $D_{VSO}(\widehat{m}_d)$, $R(\widehat{m}_d)$, before the coding stage i, are all collected as features for the decision of the coding stage i. The decision for the coding stage i can then be decided by the direct features, $J_{VSO}(\widehat{m}_d)$, $D_{VSO}(\widehat{m}_d)$, and $R(\widehat{m}_d)$ before conducting the coding stage i for the current CU. It is noted that these features are intermediate values in depth coding and selecting these features do not introduce additional complexity.

D. DECISION TREE LEARNING

Based on the common test condition (CTC) specified in [17], 8 sequences in Table 6 are selected for the evaluation of various depth coding algorithms. The sequence *Newspaper* is selected for sample collection. To avoid redundant samples, we selected a single frame for each second, i.e. the 1^{st}, 31^{th}, 61^{th}, …, 271^{st} frames for sample collection, given the frame rate is 30fps. To obtain the ground truth labels, the coding model HTM-16.1 [18], with All-Intra configuration described in Section IV, is used to conduct the original depth coding in 3D-HEVC. Besides, all 8 sequences are used for evaluating the performance of the final classification.

The Scikit-learn [19], a popular machine learning package, is utilized for offline training. Written as a python package in Scikit-learn, the well-known CART algorithm [20] is used to construct the Decision Tree. It is noted that each leaf node of the final tree has a Gini value representing for the impurity of the final classification decision. The lower the Gini value is, the more accurate the leaf decision is. The Gini value for each node is calculated as follows:

$$Gini = 1 - \sum_{k=1}^{2} p_k^2 = 1 - \sum_{k=1}^{2} \left(\frac{N_k}{N}\right)^2 \tag{5}$$

where k, 0 or 1, refers to the *Skip* or *Check* labels in sample collection, p_k is the proportion of the samples with k label of the current leaf node, which is calculated by the number of samples with k labels, N_k, and the total number of samples, N, of the current leaf node.

Figure 3 shows an example of the final tree for the decision of coding stage 1 at the depth level of 2. It is noted that the leaf nodes with *Check* result refer to the decision of checking the corresponding coding stage, which is the same as the original process. Thus, only the leaf nodes with result *Skip* are focused here. As we can see, each leaf node with result *Skip* has a Gini value varying from 0.024 to 0.5. The Gini value will be considered as a metric of the confidence level to adopt the result of decision trees into the proposed stages in Figure 2.

In this paper, we set a threshold *TH* for the Gini value of the leaf node with *Skip* result. If the Gini of the leaf node is less than *TH*, we then adopt the leaf node result, *Skip*, into the proposed flow in Figure 2. The proposed decision from restricting the leaf node results is shown as follows:

$$Decision = \begin{cases} Skip, & if\ leaf\ result\ is\ Skip\ with\ Gini \leq TH \\ Check, & Otherwise \end{cases} \tag{6}$$

The performance evaluation of the proposed decision in (6) with different thresholds *TH* will be discussed in the next Section.

IV. SIMULATION RESULTS

The proposed decision tree based algorithm for depth intra coding has been implemented in HTM-16.1 [18]. The original depth intra mode decision in HTM-16.1 is an anchor for comparison with the state-of-the-art algorithms [4], [7], [8], [11], [13], [14] and the proposed algorithm. The quantization parameters (QP) were set as 25, 30, 35, and 40 for texture views and 34, 39, 42, and 45 for the corresponding depth views. All results in Table 7-9 are the average simulation results of these QPs. Sequences for evaluation are all 8 sequences listed in Table 6. These sequences were tested under the common test condition (CTC) specified in [17]. The coding configuration is All-intra. The experimental work was implemented on the platform with the CPU of Intel(R) Core i7-4790 CPU @ 3.60GHz and RAM 16.0GB. To study the performances of the proposed algorithm compared with the state-of-the-art algorithms, coding results including complexity reduction and coding efficiency are taken into account. The average of encoding time saving ΔT

TABLE 7. Performances of the proposed algorithm with different Gini-Thresholds compared with HTM-16.1 (%).

Test sequence	Proposed Algorithm							
	$TH = 0.1$		$TH = 0.2$		$TH = 0.3$		$TH = 0.4$	
	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT
Balloons	-0.02	-32.82	+0.10	-46.56	+0.26	-52.53	+0.45	-60.83
Kendo	+0.03	-36.12	+0.10	-51.27	+0.30	-58.21	+0.70	-66.06
Newspaper	+0.03	-23.96	+0.02	-36.77	+0.31	-44.33	+0.67	-53.44
GT_Fly	-0.01	-43.01	+0.07	-65.50	+0.51	-71.07	+2.37	-77.14
Poznan_Hall2	-0.05	-61.10	+0.40	-79.12	+1.26	-82.32	+1.97	-85.73
Poznan_Street	+0.01	-37.32	+0.11	-61.38	+0.31	-69.40	+0.68	-76.24
Shark	+0.04	-43.05	+0.03	-64.04	+0.28	-70.23	+0.75	-75.38
Undo_Dancer	+0.01	-51.78	+0.15	-69.01	+0.69	-74.24	+1.25	-78.26
Average	-0.00	-41.14	+0.12	-59.21	+0.49	-65.29	+1.10	-71.63

TABLE 8. Comparison between the proposed and the state-of-the-art algorithms (%).

Test sequence	Zhang's [4]		Mora's [11]		Saldanha's [7]		Sanchez's [8]		Shen's [13]		Zuo's [14]		Proposed (TH=0.2)	
	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT
Balloons	+0.40	-38.53	+5.47	-45.13	+0.21	-26.5	+0.36	-34.1	+0.1	**-59**	+0.31	-37.0	**+0.10**	-46.56
Kendo	+0.46	-40.24	+4.19	-54.73	+0.3	-23.2	+0.37	-33.9	+0.3	**-63**	**-0.1**	-40.8	+0.10	-51.27
Newspaper	+0.87	-35.51	+4.40	-45.64	+0.26	-23.2	+0.46	-35.4	+0.3	**-55**	+0.61	-36.2	**+0.02**	-36.77
GT_Fly	+3.01	-41.99	+6.06	-49.08	N/A	N/A	+0.12	-40.6	**+0.0**	-64	+0.19	-53.1	+0.07	-65.50
Poznan_Hall2	+0.89	-49.45	+2.77	-63.59	+0.39	-30.2	+0.43	-38.8	+0.4	-65	+1.3	-54.4	**+0.40**	**-79.12**
Poznan_Street	+0.27	-43.83	+1.52	-50.44	+0.15	-30.0	+0.22	-41.7	+0.2	-61	+0.27	-37.1	**+0.11**	**-61.38**
Shark	+0.91	-41.94	+3.87	-52.21	+0.13	-28.4	+0.11	-36.2	N/A	N/A	+0.19	-46.1	**+0.03**	**-64.04**
Undo_Dancer	+0.31	-43.40	+0.79	-46.59	N/A	N/A	+0.12	-38.5	+0.3	-61	+0.25	-53.7	**+0.15**	**-69.01**
Average	+0.89	-41.86	+3.63	-50.93	+0.24	-26.9	+0.27	-37.4	+0.2	**-61**	+0.38	-44.8	**+0.12**	-59.21

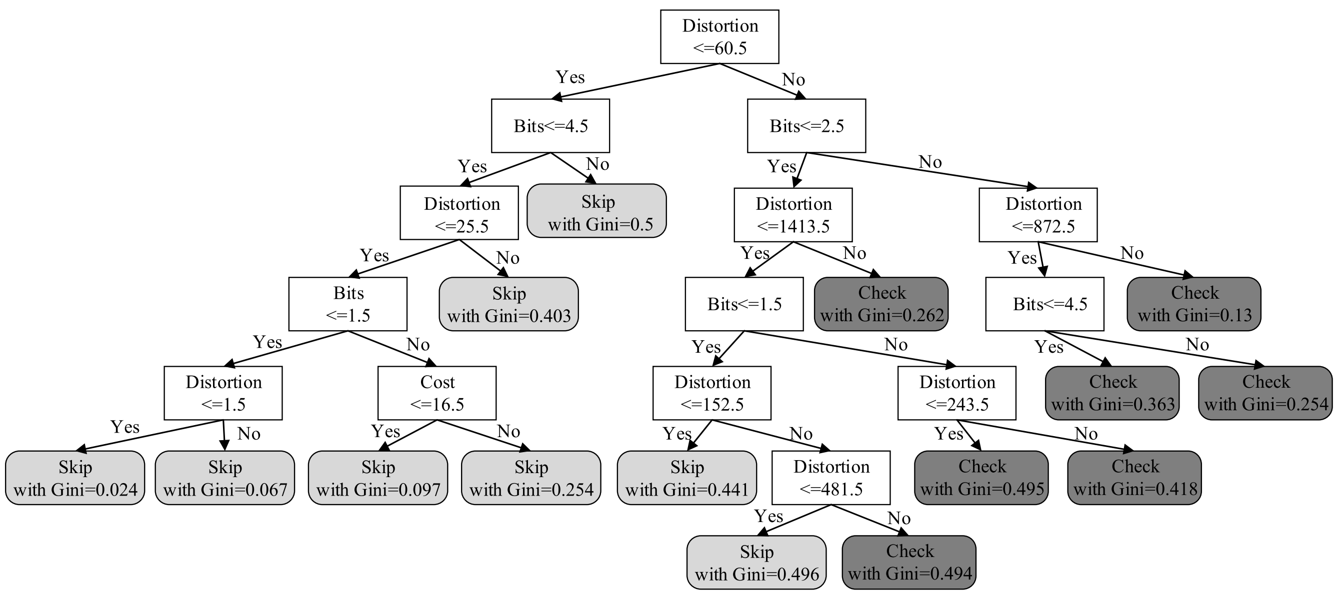


FIGURE 3. An example of decision tree: the final tree for the decision of Intra 2N×2N (Stage 1) at the depth level of 2.

for four different QPs is used to evaluate the complexity reduction. And the coding efficiency is evaluated by the BDBR [21] which is calculated by the PSNR of synthesized views and the total bitrate of depth and texture videos.

A. PERFORMANCE OF THE PROPOSED ALGORITHM

Table 7 shows the results of the proposed algorithm with different Gini thresholds, ($TH = 0.1, 0.2, 0.3, 0.4$). From this table, it can be seen that the proposed algorithm has a remarkable time reduction, ranging from 41.14% to 71.63%, with a small BDBR increase. For $TH = 0.1$, there is no BDBR increase on average. With TH increases up to 0.4, the BDBR increase is 1.10%, which is still acceptable especially considering the significant time reduction of 71.63%.

Besides, the performance comparison with the state-of-the-art algorithms is tabulated in Table 8, where the

TABLE 9. Contribution of each decision stage with Gini-Threshold of 0.2 compared with HTM-16.1 (%).

Sequences	Intra 2N×2N		Intra N × N		CU Splitting	
	ΔBDBR	ΔT	ΔBDBR	ΔT	ΔBDBR	ΔT
Balloons	-0.01	-20.2	-0.02	-14.8	+0.05	-35.2
Kendo	+0.02	-22.2	-0.00	-15.5	+0.09	-41.7
Newspaper	+0.06	-14.7	+0.02	-11.9	+0.07	-27.3
GT_Fly	+0.02	-29.8	+0.01	-18.2	+0.06	-57.9
Hall2	-0.04	-40.6	-0.00	-21.8	+0.34	-70.8
Street	+0.02	-25.4	+0.01	-15.9	+0.08	-53.9
Shark	-0.02	-28.6	-0.00	-19.1	-0.00	-56.4
Dancer	+0.01	-35.3	-0.01	-18.5	+0.12	-59.5
Average	**+0.01**	**-27.1**	**-0.00**	**-17.0**	**+0.10**	**-50.3**

Gini-Threshold of the proposed algorithm is set as 0.2 and the best performance is bolded for each row. It is demonstrated that the proposed algorithm outperforms all the other algorithms in almost all sequences in ΔBDBR and in half of them in ΔT simultaneously. In addition, the overall performance of Shen's algorithm [13] is close to the proposed one with similar time saving but more increase in BDBR. However, it is noted that four training sequences are utilized to train the statistics in [13] and only three of them are test sequences. The result of *Shark* is not available in [13]. Similarly, two sequences are used for training in [7]. They are marked as "N/A" in the table. In contrast, only one random selected training sequence is required in the proposed algorithm. It demonstrates the advantage of using decision tree over the empirical classification based on observation and statistics.

B. CONTRIBUTION OF EACH PROPOSED EARLY DECISION

To show the contribution of each decision, we separate the proposed algorithm into three parts. Each part only conducts one proposed decision of the three coding stages: Intra 2N×2N, Intra N×N, and CU splitting. Since the results of the proposed algorithm with *TH* of 0.2 shown in Table 8 are superior both in BDBR and time reduction, we set *TH* as 0.2 in evaluating each individual decision. The coding results of the three separate parts are shown in Table 9. As we can see, all three parts keep the BDBR increase at a low level. The early decision for CU splitting part can achieve 50.3% remarkable complexity reduction. If the decision is "*Skip*" at the coding stage of CU splitting, all recursive computation for smaller sub-CUs at higher depth level would not be conducted. Besides, the early decisions of skipping 2N×2N and Intra N×N can also achieve, respectively, 27.1% and 17.0% of time reduction. It is noted that the early decision of Intra 2N×2N or Intra N×N part is designed for CUs that are more likely to be coded in the final bitstream, as mentioned in sample collection in Section III B. Although the time reduction for these two parts is not as great as the decision for CU splitting, they provide a much reliable BDBR as shown in Table 9. Combining all these three early decisions together, a better result could then be achieved. It is interesting to note that in both TABLE 7 and TABLE 9, the coding performance of BDBR of the proposed algorithm is superior to the original encoder for some sequences while taking less coding time. The reason is as follows. In the RD optimization process of the original encoder, $D_{VSO}(m_d)$ in (1) contains both the distortion of the synthesized view D_{syn} and the depth map D_{dep}. While in the calculation of BDBR, only D_{syn} is considered since only the quality of the synthesized view is concerned. Besides, D_{syn} is estimated by model-based view synthesis distortion (VSD) without performing time-consuming view synthesis in some steps of mode decision. It also results in the sub-optimization in the original encoder. A similar phenomenon could be found for the sequence *Kendo* of Zuo's method [14] in TABLE 8.

V. CONCLUSION

In this paper, we propose an early determination algorithm for depth intra coding, in which whether to skip or check the coding stages of Intra 2N×2N, Intra N×N and CU splitting for the current CU is considered as several binary decisions. After conducting the decision tree based learning algorithm, each proposed decision for the current CU could obtain a result with Gini value from the tree leaves. By restricting the leaf results with different Gini-thresholds, the proposed process can save 41.14% to 71.63% of the depth coding time with negligible BDBR loss.

REFERENCES

[1] C.-S. Park, "Edge-based intramode selection for depth-map coding in 3D-HEVC," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 155–162, Jan. 2015.

[2] J. Y. Lee, M. W. Park, and Y. Jin, *3D-CE2 Related: Fast SDC DC Offset Decision*, document JCT3V-I0084, Joint Collaborative Team on 3D Video Coding Extension Development, 9th Meeting, Sapporo, Japan, Jul. 2014.

[3] H.-B. Zhang, C.-H. Fu, W.-M. Su, S.-H. Tsang, and Y.-L. Chan, "Adaptive fast intra mode decision of depth map coding by low complexity RD-Cost in 3D-HEVC," in *Proc. IEEE Int. Conf. Digit. Signal Process.*, Singapore, Jul. 2015, pp. 487–491.

[4] H.-B. Zhang, C.-H. Fu, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Probability-based depth intra-mode skipping strategy and novel VSO metric for DMM decision in 3D-HEVC," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 28, no. 2, pp. 513–527, Feb. 2018.

[5] G. Sanchez, M. Saldanha, L. Agostini, and C. Marcon, "Depth modeling modes complexity control system for the 3D-HEVC video encoder," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, 2017, pp. 1021–1025.

[6] Q. Zhang, Q. Yang, Y. Chang, W. Zhang, and Y. Gan, "Fast intra mode decision for depth coding in 3D-HEVC," *Multidimensional Syst. Signal Process.*, vol. 28, no. 4, pp. 1203–1226, Oct. 2017.

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

[8] G. Sanchez, L. Agostini, and C. Marcon, "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.

[9] C.-H. Fu, Y.-W. Zhao, H.-B. Zhang, Y.-L. Chan, and W.-C. Siu, "Depth modelling mode decision for depth intra coding via good feature," in *Proc. IEEE Int. Conf. Image Process.*, Beijing, China, Sep. 2017, pp. 4018–4022.

[10] M. Kim, N. Ling, and L. Song, "Fast single depth intra mode decision for depth map coding in 3D-HEVC," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, Turin, Italy, Jun./Jul. 2015, pp. 1–6.

[11] E. G. Mora, J. Jung, M. Cagnazzo, and B. Pesquet-Popescu, "Initialization, limitation, and predictive coding of the depth and texture quadtree in 3D-HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 9, pp. 1554–1565, Sep. 2014.

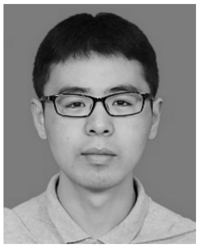
[12] H. Chen, C.-H. Fu, Y.-L. Chan, and X. Zhu, "Early intra block partition decision for depth maps in 3D-HEVC," in *Proc. 25th IEEE Int. Conf. Image Process.*, Athens, Greece, Oct. 2018, pp. 1777–1781.

[13] L. Shen, K. Li, G. Feng, P. An, and Z. Liu, "Efficient intra mode selection for depth-map coding utilizing spatiotemporal, inter-component and inter-view correlations in 3D-HEVC," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4195–4206, Sep. 2018.
[14] J. Zuo, J. Chen, H. Zeng, C. Cai, and K.-K. Ma, "Bi-layer texture discriminant fast depth intra coding for 3D-HEVC," *IEEE Access*, vol. 7, pp. 34265–34274, 2019.
[15] J. Y. Lee, M. W. Park, and C. Kim, *3D-CE1: Depth Intra Skip (DIS) Mode*, document JCT3V-K0033, Joint Collaborative Team on 3D Video Coding Extension Development, 11th Meeting, Geneva, Switzerland, Feb. 2015.
[16] Y. Chen, G. Tech, K. Wegner, and S. Yea, *Test Model 11 of 3D-HEVC and MV-HEVC*, document JCT3V-K1003, Joint Collaborative Team on 3D Video Coding Extension Development, 11th Meeting, Geneva, Switzerland, Feb. 2015.
[17] K. Muller and A. Vetro, *Common test conditions of 3DV core experiments*, document JCT3V-G1100, Joint Collaborative Team on 3D Video Coding Extension Development, San Jose, CA, USA, Jan. 2014.
[18] *3D-HEVC Reference Software: HTM-16.1*. Accessed: Dec. 3, 2019. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-16.1/
[19] *Decision Trees Scikit-Learn 0.19.1 Documentation*. Accessed: Dec. 3, 2019. [Online]. Available: http://scikit-learn.org/stable/modules/tree.html
[20] D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man Cybern.*, vol. 21, no. 3, pp. 660–674, May 1991.
[21] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD Curves*, document ITU-T Video Coding Experts Group (VCEG), 2001.
[22] G. Correa, P. A. Assuncao, L. V. Agostini, and L. A. da Silva Cruz, "Fast HEVC encoding decisions using data mining," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 4, pp. 660–673, Apr. 2015.
[23] M. Grellert, S. Bampi, G. Correa, B. Zatt, and L. A. da Silva Cruz, "Learning-based complexity reduction and scaling for HEVC encoders," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Calgary, AB, Canada, Apr. 2018, pp. 1208–1212.
[24] N. Westland, A. S. Dias, and M. Mrak, "Decision trees for complexity reduction in video compression," in *Proc. IEEE Int. Conf. Image Process.*, Taipei, Taiwan, Sep. 2019, pp. 2666–2670.
[25] D. Zhang, X. Duan, and D. Zang, "Decision tree based fast CU partition for HEVC lossless compression of medical image sequences," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process.*, Nanjing, China, Oct. 2017, pp. 1–6.
[26] F. Duanmu, Z. Ma, and Y. Wang, "Fast mode and partition decision using machine learning for intra-frame coding in HEVC screen content coding extension," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 517–531, Dec. 2016.
[27] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Fast HEVC to SCC Transcoder by early CU partitioning termination and decision tree-based flexible mode decision for intra-frame coding," *IEEE Access*, vol. 7, pp. 8773–8788, 2019.

CHANG-HONG FU (M'07) received the B.Eng. (Hons.) and Ph.D. degrees from The Hong Kong Polytechnic University, Hong Kong, in 2002 and 2008, respectively.

He was a Postdoctoral Research Fellow with The Hong Kong Polytechnic University, from 2008 to 2011. He joined the Nanjing University of Science and Technology, as an Associate Professor, in 2011. He was a Short-term Visiting Scholar with The University of Sydney and the University of Maryland, in 2014 and 2016, respectively. He was a Visiting Scholar with Dolby Laboratories, Inc., in 2019. He has authored and coauthored over 40 research articles in international journals and conferences. His current research interests include multimedia technologies, signal processing, image and video compression, 3DTV/3DV, multiview video coding, depth coding, machine learning for video coding, future video coding standards, and biomedical application of video processing.

Dr. Fu was a Publication Chair of the 2019 IEEE 2nd International Conference on Multimedia Information Processing and Retrieval.



HAO CHEN received the B.Eng. and master's degrees from the Nanjing University of Science and Technology, in 2016 and 2019, respectively.

He was a Research Assistant with The Hong Kong Polytechnic University, in 2018. His current research interests include 3D video coding, multiple view video, image signal processing, video transcoding, and video summarization.

YUI-LAM CHAN (S'94–A'97–M'00) received the B.Eng. (Hons.) and Ph.D. degrees from The Hong Kong Polytechnic University, Hong Kong, in 1993 and 1997, respectively.

He joined The Hong Kong Polytechnic University, in 1997, where he is currently an Associate Professor with the Department of Electronic and Information Engineering. He is actively involved in professional activities. He has authored over 130 research articles in various international journals and conferences. His current research interests include multimedia technologies, signal processing, image and video compression, video streaming, video transcoding, video conferencing, digital TV/HDTV, 3DTV/3DV, multiview video coding, machine learning for video coding and future video coding standards, including screen content coding, light-field video coding, and 360-degree omnidirectional video coding.

Dr. Chan was the Special Sessions Co-Chair and the Publicity Co-Chair of the 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, and the Technical Program Co-Chair of the 2014 International Conference on Digital Signal Processing. He serves as an Associate Editor for the IEEE Transactions on Image Processing. He was the Secretary of the 2010 IEEE International Conference on Image Processing.

SIK-HO TSANG (M'10) received the Ph.D. degree from The Hong Kong Polytechnic University (PolyU), Hong Kong, in 2013.

He was a Postdoctoral Fellow, from 2013 to 2016, and involved in numerous industrial projects for video coding and transcoding. He is currently a Research Fellow with PolyU. He has authored numerous international journals, conferences, and patents. His current research interests include video coding, such as HEVC, VVC, multiview video plus depth coding, screen content coding, and immersive video coding, including light field coding and 360-degree video coding. His research interests also include machine learning and deep learning.

Dr. Tsang serves as a Reviewer for international journals, including the IEEE Transactions on Image Processing, the IEEE Transactions on Circuits and Systems for Video Technology, and the *Journal of Signal Processing: Image Communication* (Elsevier).

HONG HONG (M'10) received the Ph.D. degree in electrical engineering from Nanjing University, Nanjing, China, in 2010.

In 2014, he was a Visiting Scholar with the Institute of Biomedical Engineering and Technology, The University of Sydney, Sydney, Australia. He is currently an Associate Professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing. His current research interests include biomedical applications of microwave technology, audio signal processing, and radar signal processing.

Dr. Hong served as a Steering Committee Member and a TPC Member for the IEEE MTT-S International Microwave Biomedical Conference, from 2018 to 2019. He also serves as a member for the IEEE MTT-S Biological Effect and Medical Applications of RF and Microwave (MTT-10). He is a Guest Editor of the IEEE JOURNAL OF ELECTROMAGNETICS, RF AND MICROWAVES IN MEDICINE AND BIOLOGY.

XIAOHUA ZHU (M'07) received the Ph.D. degree in communication and information system from the Nanjing University of Science and Technology, Nanjing, China, in 2002.

He is currently a Professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology. He has authored and coauthored four books, and more than 100 articles. He has submitted 18 patent applications. His current research interests include radar systems, radar signal theory, and digital signal processing.

Dr. Zhu is the Director of the Radar and High-speed Digital Signal Processing Laboratory, Nanjing University of Science and Technology. He was a ten-time recipient of the Ministerial and Provincial-Level Science and Technology Award.