


Article

Fast Versatile Video Coding (VVC) Intra Coding for Power-Constrained Applications

Lei Chen ^{1,2}, Baoping Cheng ^{3,4}, Haotian Zhu ^{5,*}, Haowen Qin ⁵, Lihua Deng ⁶ and Lei Luo ⁷ ¹ College of Compute, National University of Defense Technology, Changsha 410073, China; cl@cae.cn² Center for Strategic Studies, Chinese Academy of Engineering, Beijing 100088, China³ Department of Electronic Engineering, Tsinghua University, Beijing 100084, China; cbp21@mails.tsinghua.edu.cn⁴ China Mobile (Hangzhou) Information Technology Company, Ltd., Hangzhou 310000, China⁵ Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; 20020102@emails.bjut.edu.cn⁶ China Telecom Research Institute, Shanghai 200122, China; denglihua@chinatelecom.cn⁷ School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; luolei@cqupt.edu.cn

* Correspondence: zhuhaotian@emails.bjut.edu.cn

Abstract: Versatile Video Coding (VVC) achieves impressive coding gain improvement (about 40%+) over the preceding High-Efficiency Video Coding (HEVC) technology at the cost of extremely high computational complexity. Such an extremely high complexity increase is a great challenge for power-constrained applications, such as Internet of video things. In the case of intra coding, VVC utilizes the brute-force recursive search for both the partition structure of the coding unit (CU), which is based on the quadtree with nested multi-type tree (QTMT), and 67 intra prediction modes, compared to 35 in HEVC. As a result, we offer optimization strategies for CU partition decision and intra coding modes to lessen the computational overhead. Regarding the high complexity of the CU partition process, first, CUs are categorized as simple, fuzzy, and complex based on their texture characteristics. Then, we train two random forest classifiers to speed up the RDO-based brute-force recursive search process. One of the classifiers directly predicts the optimal partition modes for simple and complex CUs, while another classifier determines the early termination of the partition process for fuzzy CUs. Meanwhile, to reduce the complexity of intra mode prediction, a fast hierarchical intra mode search method is designed based on the texture features of CUs, including texture complexity, texture direction, and texture context information. Extensive experimental findings demonstrate that the proposed approach reduces complexity by up to 77% compared to the latest VVC reference software (VTM-23.1). Additionally, an average coding time saving of 70% is achieved with only a 1.65% increase in BDBR. Furthermore, when compared to state-of-the-art methods, the proposed method also achieves the largest time saving with comparable BDBR loss. These findings indicate that our method is superior to other up-to-date methods in terms of lowering VVC intra coding complexity, which provides an elective solution for power-constrained applications.

Keywords: versatile video coding; quadtree with nested multi-type tree; intra coding; random forest; complexity reduction



Citation: Chen, L.; Cheng, B.; Zhu, H.; Qin, H.; Deng, L.; Luo, L. Fast Versatile Video Coding (VVC) Intra Coding for Power-Constrained Applications. *Electronics* **2024**, *13*, 2150. <https://doi.org/10.3390/electronics13112150>

Academic Editors: Chiman Kwan and Krzysztof Okarma

Received: 8 March 2024

Revised: 24 May 2024

Accepted: 29 May 2024

Published: 31 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As the next generation of the video coding standard, Versatile Video Coding (VVC) [1] has been developed by the Joint Video Exploration Team of the ITU-T Video Coding Group and the ISO/IEC Moving Picture Experts Group. The purpose of VVC standard is to provide adequate coding gain enhancement over the High Efficiency Video Coding (HEVC) standard to meet the requirements of the future video market, such as 4 K/8 K ultra-high definition, high dynamic range, virtual reality, and 360 degree video content. According to

reports, the VVC reference software (VTM) achieves a 45% bitrate reduction when compared to HEVC (HM) under standard test settings [2]. In fact, the coding efficiency improvement mainly benefits from several newly adopted innovative video coding techniques (e.g., the quadtree with nested multi-type tree (QTMT) partition structure [3], 65 directional intra prediction modes, and affine motion compensated prediction [4]), which leads to an extremely high encoding complexity increase. Based on the AHG report, the intra coding complexity of VTM increases more than 10 times over that of HM under the All-Intra test configuration [5]. However, such a high computational complexity will bring a significant challenge for power-constrained applications, such as Internet of video things (IoVT) [6]. The substantial volume of data also poses potential risk for IoVT security. Especially in terms of real-time requirements, more rapid video encoding and decoding is desired. Consequently, fast coding algorithms of VVC must be developed in order to decrease the computing complexity.

Brute-force recursive RDO searches of intra prediction modes and the partition structure of CU occupy most of the time in VVC intra coding. VVC adopts the QTMT partition structure to make CU partition shapes more flexible as shown in Figure 1. A quadtree (QT) is first used to partition a coding tree unit (CTU), and then each leaf node is partitioned by a multi-type (MT) structure until the size of the CU is the smallest. Vertical binary tree (BV), horizontal binary tree (BH), vertical ternary tree (TV), and horizontal ternary tree (TH) are four partition options of a multi-type tree (MT) structure. A QTMT partition structure is depicted in Figure 1, where different lines indicate different partition modes. Indeed, the QTMT coding block structure better suits the characteristics of various texture patterns, resulting in significant coding efficiency gain. However, such an impressive performance improvement comes at the expense of extremely high computational complexity. Based on the principle of the QTMT structure with several limitations, it can be deduced that there are, in total, more than 30,000 possible CU partition modes for a 128×128 CTU. With so many Rate-Distortion Optimization (RDO) based brute-force recursive searches, they contribute to the largest proportion of the encoding time. Therefore, a central priority is to achieve an efficient CU partition method for the fast coding algorithm.

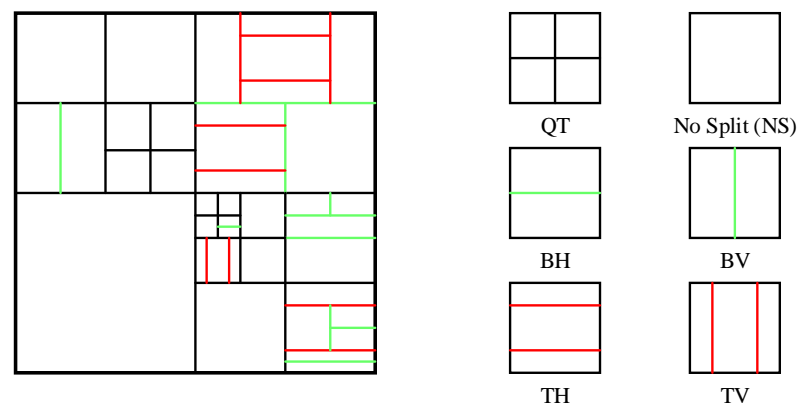


Figure 1. An illustration of the QTMT partition structure [1].

In addition, to further reduce the spatial redundancy of the intra frames, VVC designs up to 67 intra prediction modes, whereas HEVC only allows for 35. Meanwhile, several advanced prediction techniques are also proposed to further improve the intra coding efficiency, such as matrix weighted intra prediction [7], multi-reference line intra prediction [8], and intra sub-partition [9]. Evidently, the improvement of the VVC intra coding efficiency beyond HEVC requires more calculations. Therefore, instead of conducting the brute-force traversal of the expensive RDO process for all 67 modes as shown in Figure 2, the classical three-step fast intra mode decision (TS-FMD) [10] of HEVC is inherited by VVC to reduce the computational complexity of the intra mode prediction. First, VVC adopts rough mode decision (RMD) [11] to limit the number of modes for the RDO search process. In RMD,

N (usually 2 or 3) modes with the smallest Hadamard transform-based (HAD) cost are chosen from the original 35 modes of HEVC to construct a candidate list. The HAD cost is calculated as follows:

$$HAD_{Cost} = SATD + \lambda \cdot Bit_{Mode} \quad (1)$$

where $SATD$ is the sum of residual signal's absolute Hadamard transformed differences, λ represents the Lagrange multiplier, and Bit_{Mode} specifies the number of bits for encoding the intra mode information. Second, two kinds of intra mode are integrated to create a final candidate list consisting of two neighbors (left and right) of the N RMD modes and a most probable model (MPM) generated by the neighboring CUs. Finally, two or three models from the candidate list with minimal HAD-cost proceed through the RDO process to generate the final intra prediction mode. The RD-cost for each mode is calculated as follows:

$$RD_{Cost} = SSE + \lambda \cdot Bit_{Total} \quad (2)$$

where SSE is the sum of squared errors between the original and reconstructed CU, and Bit_{Total} is the bits budget to encode the selected mode. It can be seen that TS-FMD selects the best mode with about 40 times of the RMD process and 2 or 3 times of the RDO process. The TS-FMD approach can reduce the complexity of the intra prediction processing to some extent but requires additional reduction in the computing complexity.

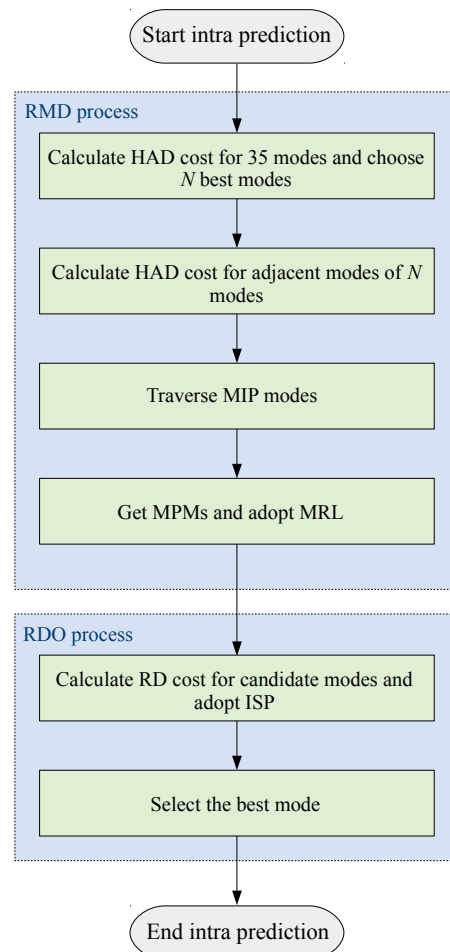


Figure 2. Flowchart of the three-step intra mode decision in VVC reference software.

To minimize the computational cost of VVC significantly, we propose a fast intra coding algorithm that integrates a more efficient CU partition strategy and a fast intra mode prediction scheme. First, a detailed statistical analysis for the CU depth distribution and intra prediction mode is presented, which motivates us to design the fast intra coding

algorithm. From the distribution of the CU depth, the RDO brute-force recursive search of the CU partition process is sped up by using random forest-based classification. The CUs are categorized as simple, fuzzy, and complex CUs based on their texture characteristics. One random forest classifier predicts the best partition modes of simple and complicated CUs directly, while another random forest identifies the early termination of the partition process for fuzzy CUs. This approach not only maximizes the potential for speed improvement but also effectively mitigates bitrate distortion (BDBR) losses, particularly for fuzzy CUs. Meanwhile, a fast hierarchical intra mode search method based on the CU texture feature is presented, and experimental results prove that our method significantly lessens the computational cost of intra mode prediction while causing negligible RD loss.

The remainder of this paper is structured as follows. Section 2 provides an overview of the related work. The statistical distribution of intra prediction and CU depth is explored in Section 3. Section 4 describes our algorithm in details. The extensive experimental results and several ablation studies are shown in Section 5, while Section 6 ends with a conclusion.

2. Related Works

The fast intra coding algorithm focuses on two main tasks, the fast CU partition mode decision and fast intra mode prediction, respectively.

2.1. Fast CU Partition Structure Decision

For the CU partition structure decision, there are three kinds of strategies, including early termination, multi-classification, and joint classification. The early-termination policy attempts to determine whether the recursive CU partition terminates after partitioning at the current depth level. It is easy to understand that the early termination mainly happens in the region of smooth texture. It maintains the RD performance well, but the reduction in computational complexity is also limited. The multi-classification strategy designs a sophisticated prediction model to directly obtain the optimal partition mode. It reduces the computational complexity significantly. However, the complex partition structure of VVC leads to the limitation of prediction accuracy and greater loss of RD performance. For the sake of improving the prediction accuracy while achieving a large amount of time saving, joint classification is designed to predict the probability of each partition mode, and one or more partition modes with highest probability are selected as the candidates of the final encoding mode.

For each of the above three strategies, three categories of common methods are used, including heuristic methods, traditional machine learning methods, and end-to-end deep learning methods. The heuristic methods first extract some handcrafted features in the encoding process (e.g., image texture complexity, rate-distortion cost, and context information), and the decision criteria are based on the preset thresholds. Its efficiency is influenced by the chosen features and decision criteria simultaneously. The greatest advantage of heuristic methods is that they are very simple but accuracy is limited. Traditional machine learning methods employ advanced classifier (e.g., decision tree) to automatically learn a decision function instead of manually designing one of the heuristic methods. However, it still needs to select some handcrafted features. On the contrary, end-to-end deep learning methods attempt to accomplish the feature extraction and decision work automatically with advanced deep neural networks.

2.1.1. Heuristic Methods

In the last few years, several heuristic methods have extensively been adopted for the fast CU partition structure decision algorithms in HEVC and VVC. By considering the spatial correlation, Gu et al. used context information to skip some CU depths that are rarely found in neighboring CUs [12]. Li et al. proposed an early intra CU size decision approach for VVC based on a configurable decision model [13]. Ni et al. proposed a texture analysis-based TT and BT partition strategy with the gradient-based intra mode decision to accelerate VVC intra coding [14]. In [15], the Bayesian decision rule was used to design an

early termination framework for CU partition based on the distribution of partition modes and RD cost. Similarly, an early termination strategy based on the HAD and RD cost was presented in [16]. Saldanha et al. proposed to remove abundant predictions by analyzing the characteristics of current block and encoding context using selected intra prediction modes [17]. In [18], Cui et al. predicted the likelihood of partition modes according to the directional gradient, and then eliminated the unlikely partitions. Fan et al. suggested that if the texture of the current CU was found to be smooth, CU partition would be terminated. Then, the gradient could be utilized to determine whether or not to split the CU by QT partition. Finally, one QTMT partition would be selected based on the difference of the variance of the sub-CUs [19].

2.1.2. Machine Learning Methods

Since CU partition can be regarded as a classification problem, both VVC and HEVC employ machine learning. Kim et al. suggested a fast CU partition approach based on Bayesian decision making that predicts whether the CU partition should be terminated via offline and online joint learning [20]. Wang et al. proposed a fast CU decision algorithm based on FSVMs and DAG-SVMs for coding complexity reduction, which divides the CU-partitioning process into two stages and symmetrically extracts some of the same CU features [21]. The fast CU size decision method was guided by the SVM classification of the complexity degree [22]. Erabadda et al. devised a weighted SVM-based CU size selection algorithm [23]. This algorithm extracts texture complexity, RD cost and context information to train the model. Shan et al. proposed a fast CU partition algorithm based on SVM and the Laplace Transparent Composite Model (LPTCM) [24]. To obtain the partition mode, the feature vectors called the Summation of Binarized Outlier Coefficient are recovered from the original frames using LPTCM and fed to the online trained SVM. The decision tree-based fast CTU partition structure in [25] explores the distribution of CU depths and designs a joint-classification framework to predict the partition probability for each partition mode. Zhang et al. designed an individual fast CU partition scheme for different CU types based on their texture complexity, while the CU with simple texture complexity was not handled efficiently [26].

2.1.3. Deep Learning Methods

Benefiting from the development of deep learning, end-to-end prediction of CU partition modes have been developed to lessen the computational complexity of video coding recently. Wu et al. proposed a hierarchical grid fully convolutional network to predict the QTMT partition structure for fast VVC intra coding [27]. Chen et al. proposed to predict CU partition modes for VVC intra coding by a CNN model, which is trained with the neighboring line pixels and quantization parameters [28]. In [29], heuristic and deep learning methods were combined. A threshold-based texture classification model was first conducted to terminate the partition process of the homogeneous CUs. Then, three different CNNs were designed to predict the partition modes for the remaining CUs. In [30], a bagged tree model was employed to predict the splitting of a CTU, and the partition problem of a 32×32 -sized CU was modeled as a 17-output classification task. Wang et al. [31] proposed a densely connected convolution neural network to predict the partition of coding units (CUs), which significantly reduces the coding complexity. Zan et al. proposed a redesigned U-NET with a quality parameter fusion network to accelerate the QTMT partition process [32]. Tang et al. proposed an adaptive CU partition choice with a pooling-variable CNN architecture, which is applicable to a wide range of CU sizes and requires only one parameter set to be configured [33]. In [34], a CNN was built for each 64×64 CU to predict a probability vector to speed up the partition process during encoding.

2.2. Fast Intra Mode Prediction

Based on the discovery that intra mode prediction is highly associated to texture features, the fast intra mode prediction algorithms focus on modeling such a relationship to accelerate the intra mode prediction process. Each block had five different edge directions, determined in [35]. A limited set of intra modes was defined for the specified edge type, reducing the complexity of the RMD process. In [36], a gradient-based strategy was presented to decrease the number of RMD and RDO candidate modes: the average gradients in the horizontal (AGH) and vertical (AGV) directions were utilized to determine the rough range of the block direction; the number of RMD and RDO modes was reduced based on the values of AGH, AGV, and AGH/AGV. In addition, Zhang et al. also proposed a gradient-based method to eliminate irrelevant orientation modes from the candidate list [37]. In order to further reduce the complexity, a mode classification based on texture features was developed to adaptively reduce the number of intra modes in chrominance. Jamali et al. not only utilized the Prewitt operator in the gradient but also tried to skip needless modes by predicting the cost of RDO [38]. In [39], progressive rough mode search (pRMS) was conducted based on the HAD cost, selecting potential modes rather than traversing all candidates. In the subsequent RDO process, fewer candidate modes were selected by pRMS. Jamali et al. succeeded in rejecting non-promising modes from further processing by utilizing the prediction modes and saving significant computations [40]. Ogata et al. [41] offered a fast intra mode decision that limits the number of possible modes by using DCT coefficients and outliers from nearby blocks.

In addition, machine learning and deep learning were also applied for intra mode prediction. Ryu et al. designed a random forest which uses four pixel points as features to reduce the number of candidates before optimizing rate-distortion [42]. Song et al. offered a fast intra mode prediction algorithm based on CNN which outputs the probability of each mode being selected as the optimal mode [43]. Meanwhile, a corner detection algorithm was developed to further minimize the number of candidate modes entering the RDO process. Ting et al. used an improved LeNet-5 CNN model to predict the probability of each mode, which achieved higher prediction accuracy [44].

3. Statistical Analysis of Intra Coding

The statistical study of VVC intra coding is useful for proposing and comprehending novel algorithms. The following study employs 26 JCT-VC test sequences encoded with VTM7.0 and placed in an all-intra configuration. These test sequences range from Class A1 to Class F with varying resolutions and contents.

3.1. CU Partition Depth Distribution

Similar to [26], the CU depth distribution of different sequences with QPs $\in \{22, 27, 32, 37\}$ is shown in Table 1. It should be noted that only one sequence of each class is analyzed based on VTM 2.0 in [26]. However, our analysis is based on all of the sequences of each class. In addition, our experiments are implemented with VTM 7.0, which includes some new advanced intra coding techniques as compared to VTM 2.0. The results are the average values for each class. $D_{i,j}$ denotes the CU depth with QT depth of $i, i \in \{1, 2, 3, 4\}$ and MT depth of $j, j \in \{1, 2, 3\}$. There are some statistical conclusions that can be observed from Table 1 as follows:

- (1) High-resolution sequences tend to be coded with a large CU size, and lower-resolution sequences prefer using a small CU size.
- (2) The percentage of large size CU increases gradually with the increasing of QP.
- (3) The percentage of CUs decreases with the decreasing of the CU size on the trend.
- (4) Nearly two thirds of regions are coded with MT CUs.

Table 1. CU depth distribution for sequences with different resolution and QPs (%).

Class	QP	$D_{1,0}$	Quadtree Depth 2				Quadtree Depth 3				Quadtree Depth 4		
			$D_{2,0}$	$D_{2,1}$	$D_{2,2}$	$D_{2,3}$	$D_{3,0}$	$D_{3,1}$	$D_{3,2}$	$D_{3,3}$	$D_{4,0}$	$D_{4,1}$	$D_{4,2}$
A1	22	20.9	20.5	19.2	11.1	8.9	2.9	5.3	5.8	3.3	1.2	0.6	0.4
A1	27	28.1	29.4	19.7	9.2	5.4	2.4	2.4	1.7	1.0	0.4	0.2	0.1
A1	32	36.8	30.8	18.1	6.8	3.3	1.7	1.2	0.6	0.3	0.2	0.1	0
A1	37	51.2	27.3	12.3	4.4	2.4	1.1	0.8	0.3	0.1	0.1	0	0
A2	22	13.4	10.6	16.4	19.3	18.1	4.6	6.6	6.1	3.0	1.1	0.5	0.3
A2	27	21.8	16.6	22.5	16.0	12.0	3.6	3.4	2.2	1.1	0.5	0.2	0.1
A2	32	24.4	18.5	23.4	15.1	11.1	2.5	2.6	1.5	0.5	0.3	0.1	0
A2	37	29.4	22.1	24.8	12.8	7.6	1.4	1.1	0.5	0.1	0.1	0	0
B	22	6.5	6.1	13.8	17.6	15.5	4.1	8.0	11.4	6.7	7.3	2.0	1.1
B	27	11.1	11.0	20.1	16.8	11.7	4.9	7.5	7.2	5.9	2.0	1.2	0.9
B	32	14.5	15.2	23.6	14.9	10.2	4.1	6.2	5.0	3.7	1.3	0.8	0.5
B	37	19.2	19.6	25.8	13.0	9.2	3.2	4.3	2.9	1.6	0.7	0.3	0.2
C	22	0	1.2	3.7	5.6	7.6	4.2	13.8	22.1	24.0	5.8	6.2	5.8
C	27	0.2	2.2	5.5	7.6	11.1	6.0	16.2	19.2	17.4	5.8	4.8	3.8
C	32	0.3	4.3	11.3	12.4	14.4	7.2	14.9	14.6	11.5	3.9	3.0	2.2
C	37	1.2	10.3	16.8	15.4	16.0	6.5	12.2	9.9	6.5	2.7	1.6	0.9
D	22	0	1.8	3.5	4.5	7.1	4.2	13.3	20.4	25.1	5.0	7.3	8.0
D	27	0	1.7	5.9	7.0	9.4	5.8	16.1	17.8	18.4	5.5	6.1	6.1
D	32	0.3	3.7	9.4	10.0	11.1	8.5	16.1	14.7	14.4	4.4	4.1	3.4
D	37	0.6	7.5	14.5	13.0	14.3	8.8	13.9	11.2	9.1	3.2	2.4	1.6
E	22	8.5	15.6	19.5	14.1	12.4	4.9	8.4	7.3	5.8	1.5	1.2	0.7
E	27	21.9	13.0	16.4	12.9	11.2	4.9	7.1	5.8	4.2	1.3	0.9	0.5
E	32	25.8	15.2	17.8	12.1	10.7	4.3	5.9	4.1	2.5	0.9	0.5	0.2
E	37	30.2	17.5	20.0	11.9	9.4	3.3	3.7	2.3	1.1	0.5	0.2	0
F	22	7.0	12.2	19.0	11.6	7.6	6.3	9.5	9.1	8.9	3.0	3.0	2.9
F	27	9.1	15.0	20.4	12.4	8.9	5.8	9.3	7.7	5.9	2.3	1.8	1.5
F	32	11.0	18.3	21.4	13.0	9.5	5.9	7.9	5.3	4.2	1.5	1.2	0.9
F	37	14.9	20.6	23.0	12.1	7.5	5.3	6.9	4.5	3.0	0.9	0.7	0.5
Average		14.6	13.9	16.7	11.9	10.1	4.6	8.0	7.9	6.8	2.3	1.8	1.5

In addition, some intuitive observations between the texture complexity and CU partition mode can be found in Figure 3 as follows:

- (1) Generally, flat regions imply a simple CU partition mode, while rich texture regions adopt a more complex CU partition structure.
- (2) Since two directions of multi-type tree partition are employed in VVC, the partition direction is related to the texture direction. Taking the edge areas as an example, the texture extends vertically with the woman’s body, and thus the majority of partitions tend to be vertical as well.
- (3) If the texture complexity among the sub-CUs is different, the current CU probably needs to be split into smaller CUs.

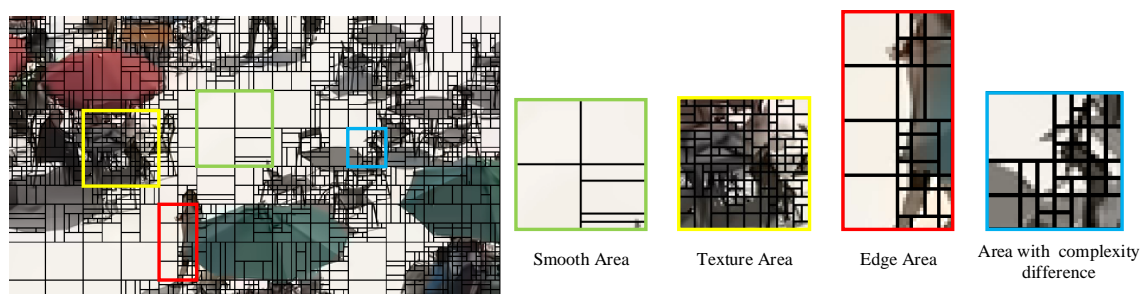


Figure 3. An example of quadtree with nested multi-type tree coding block structure.

Furthermore, statistical analysis of the relationship between the texture complexity and CU partition mode is performed as follows. The variance of a CU is defined as its texture complexity. As shown in Figure 4, by comparing the texture complexity of the current CU and its neighboring CUs, we classify CUs into three categories. If the texture complexity of the current CU is smaller than the minimum texture complexity of its neighboring CUs, it is classified as a simple CU; if the texture complexity of the current CU is greater than the maximum value of its neighboring CUs, it is classified as a complex CU. Otherwise, it is classified as a fuzzy CU. The partition probability of the three CU categories is shown in Table 2:

- (1) Nearly two-thirds of the complex CUs need to be further partitioned, especially for the videos with lower resolution.
- (2) More than 70% of simple CUs terminate the partition process at the current depth level, especially for the videos with high resolution.
- (3) The probability of early termination for fuzzy CUs is about 50%.

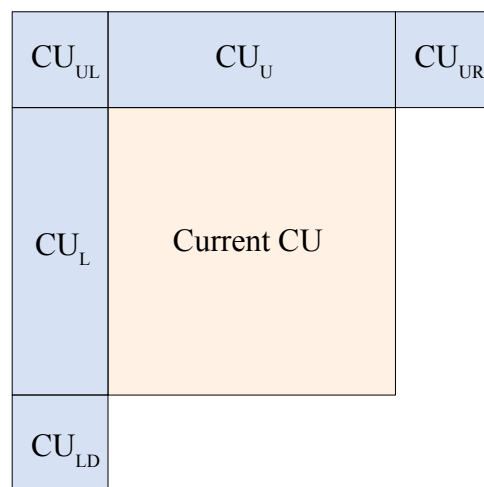


Figure 4. Neighboring CUs.

Table 2. Further partition probability of different CU (%).

Class	Simple	Fuzzy	Complex
A1	21.85	43.87	58.8
A2	26.04	50.2	65.32
B	27.12	51.92	66.44
C	30.29	50.49	73.01
D	31.96	59.52	71.04
E	25.89	52.65	64.89
F	38.13	54.29	65.39
Average	28.75	52.99	66.42

It seems that the partition process for complex and simple CUs has an obvious tendency, while the partition for fuzzy CUs is ambiguous. Therefore, we propose different acceleration schemes for the three categories of CUs, respectively.

3.2. Intra Prediction Mode Distribution

The 67 intra prediction modes of VVC can be divided into DC mode, planar mode, and angle mode. The distribution of the intra mode is presented in Table 3, and some observations can be obtained as follows:

- (1) Intra mode distribution is closely related to the sequence resolution. High-resolution sequences have more flat areas, which tend to select DC or planar mode as the best mode.
- (2) About 78.6% of the CUs best mode can be found in MPM, and it reaches 85% for some sequences with a high spatial correlation.

Table 3. Intra mode distribution of different sequences (%).

Class	Planar	DC	Angel Modes	$\frac{N_{MPM}}{N_T}$
A1	47.0	6.2	46.8	81.2
A2	41.5	7.9	50.6	84.5
B	40.6	6.4	53.0	83.8
C	25.5	4.4	70.1	76.4
D	29.3	4.2	66.5	67.9
E	32.2	4.0	63.7	78.0
F	13.0	2.5	84.5	78.7
Average	32.7	5.1	62.2	78.6

To lessen the computational cost of intra mode prediction, a feasible way is to narrow down the candidate list by removing some improbable modes. According to the first conclusion, the CUs can first be divided into two types, one is a flat CU and the other one is a non-flat CU. For flat CU, only DC and planar modes are checked to obtain the best mode. This method reduces the complexity to a certain extent, but there are still more than 60% of CUs that select directional modes.

Intuitively, the intra prediction direction is inextricably related to the texture direction. For example, as shown in Figure 5, the texture direction is 135° diagonal. The optimal intra prediction direction is more likely corresponding to it. The opposite direction of the texture direction is nearly impossible to be chosen as the optimal intra modes. It inspires us that the complexity reduction benefits from eliminating modes in unnecessary directions. The gradient is adopted to reflect the texture direction. As shown in Figure 5, it is clear that the texture direction (135° diagonal) has the smallest gradient value. By contrast, the opposite direction of the texture direction (45° diagonal) is almost impossible to be selected as the intra prediction direction, which has the largest gradient value. We investigate the probability that CUs choose the modes located in the direction with the biggest gradient value, which is shown in Table 4 and it can be found that no more than 10% of CUs select the modes in the direction with the biggest gradient value.

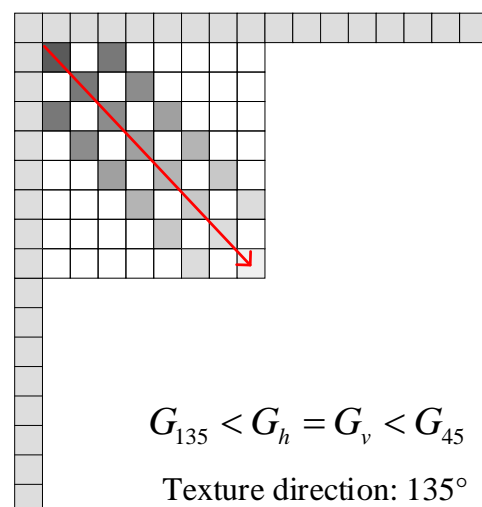


Figure 5. Relation between texture direction and gradient.

Table 4. Probability of choosing mode in directions with the largest gradient (%).

QP	Horizontal	Vertical	45	135
22	10.67	3.37	4.98	12.25
27	10.55	3.44	4.93	10.89
32	9.99	3.43	5.06	9.52
37	9.25	3.94	5.29	8.19
Average	10.11	3.52	5.06	10.21

4. Proposed Algorithm

Based on the above analysis of the statistical results in Section 3, an efficient fast algorithm for VVC intra coding is proposed, which is composed of random forest-guided classification for fast CU partition and hierarchical search-based fast intra mode prediction.

4.1. Random Forest-Guided Classification for Fast CU Partition

4.1.1. System Overview of CU Partition Structure

The proposed fast CU partition algorithm uses the random forest classifier to deal with different kinds of CU. By comparing the texture complexity between neighboring CUs, we classify CUs into three categories: simple, complex, and fuzzy. To find the best partition mode for simple and complex CUs, we utilize the same trained random forest classifier. Another random forest is utilized for fuzzy CU to decide if the partition process terminates. The overall structure is shown in Figure 6.

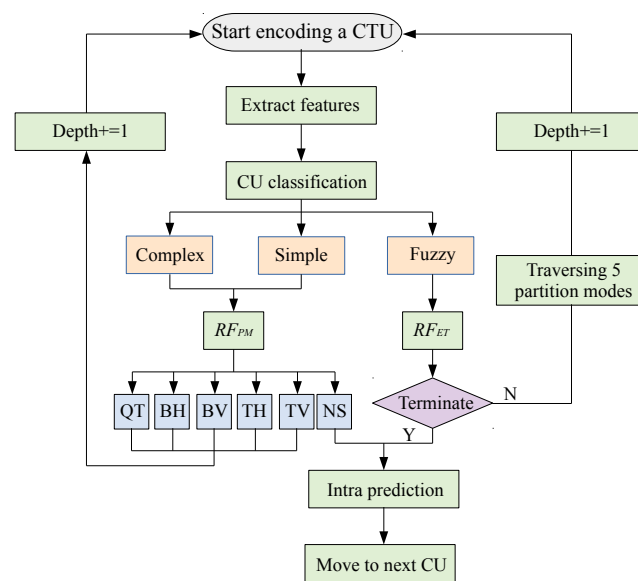


Figure 6. Flowchart of the proposed fast CU partition decision based on random forest classifier.

4.1.2. Feature Extraction

In addition to texture complexity, four other features related to the partition mode will be analyzed, including gradient information, the sub-CUs complexity difference, context information, and block information.

Texture complexity: Variance can be used to describe texture complexity as follows:

$$Var = \frac{1}{W \cdot H} \sum_{x=1}^W \sum_{y=1}^H (P(x, y) - Mean)^2 \tag{3}$$

$$Mean = \frac{1}{W \cdot H} \sum_{x=1}^W \sum_{y=1}^H P(x, y) \tag{4}$$

where W is the width of target CU, H is the height of target CU, and $P(x, y)$ is the pixel value at position (x, y) .

However, the two pictures shown in Figure 7 have the same variance but different textures, which leads to a different partition. It means that variance can only describe the global texture complexity, not the local one. So neighboring mean squared error (NMSE) is introduced to describe the local texture complexity as follows:

$$NMSE = \frac{1}{W \cdot H} \sum_{x=1}^W \sum_{y=1}^H (MADP(x, y) - Mean_{MADP})^2 \tag{5}$$

where $MADP$ is the mean of absolute difference between pixels as follows:

$$MADP_{(x,y)} = \frac{1}{8} (|P(x, y) - P(x - 1, y - 1)| + |P(x, y) - P(x - 1, y)| + |P(x, y) - P(x - 1, y + 1)| + |P(x, y) - P(x, y - 1)| + |P(x, y) - P(x, y + 1)| + |P(x, y) - P(x + 1, y - 1)| + |P(x, y) - P(x + 1, y)| + |P(x, y) - P(x + 1, y + 1)|) \tag{6}$$

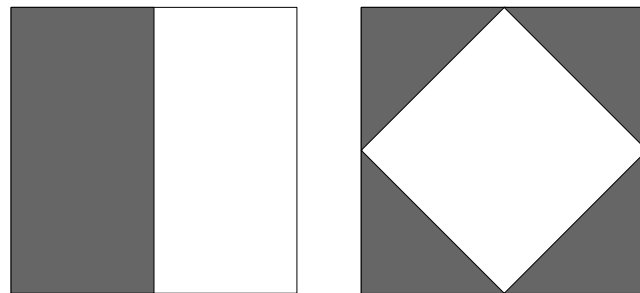


Figure 7. Different CUs with same variance.

Gradient: As mentioned earlier, the partition direction is related to the texture direction, so the gradient is also one of the features related to the partition mode. Besides the normalized gradient (G_{avg}) and maximum gradient (G_{max}), the average gradient in different directions, including horizontal, vertical, diagonal down right (DDR) and diagonal down left (DDL), should also be considered. The average gradients in the above directions are expressed as G_{HOR} , G_{VER} , G_{DDR} , and G_{DDL} , respectively. All the above gradients are formulated as Equation (7) to Equation (11):

$$G_d(x, y) = S_d \cdot P, d \in \{HOR, VER, DDR, DDL\} \tag{7}$$

$$P = \begin{bmatrix} p(x - 1, y - 1) & p(x - 1, y) & p(x - 1, y + 1) \\ p(x, y - 1) & p(x, y) & p(x, y + 1) \\ p(x + 1, y - 1) & p(x + 1, y) & p(x + 1, y + 1) \end{bmatrix} \tag{8}$$

$$G_d = \frac{1}{W \cdot H} \sum_{x=1}^W \sum_{y=1}^H |G_d(x, y)|, d \in \{HOR, VER, DDR, DDL\} \tag{9}$$

$$G_{avg} = \frac{1}{4} \sum G_d, d \in \{HOR, VER, DDR, DDL\} \tag{10}$$

$$G_{max} = \arg \max(G_d(x, y)), d \in \{HOR, VER, DDR, DDL\} \tag{11}$$

where S_d is Sobel operators.

Sub-CUs Complexity Difference: Variance difference is used to represent the texture complexity between sub-CUs, which is related to whether a sub-CU should be further

partitioned. Under five partition modes, including QT, BH, BV, TH and TV, the sub-CUs complexity difference (SCCD) is represented as follows:

$$\left\{ \begin{array}{l} SCCD_{QT} = \frac{1}{4} \sum_{i=1}^4 (var_i - \overline{var}_{QT})^2 \\ SCCD_{BH} = \frac{1}{2} \sum_{i=1}^2 (var_i - \overline{var}_{BV})^2 \\ SCCD_{BV} = \frac{1}{2} \sum_{i=1}^2 (var_i - \overline{var}_{BH})^2 \\ SCCD_{TH} = \frac{1}{3} \sum_{i=1}^3 (var_i - \overline{var}_{TH})^2 \\ SCCD_{TV} = \frac{1}{3} \sum_{i=1}^3 (var_i - \overline{var}_{TV})^2 \end{array} \right. \quad (12)$$

where var_i is the variance of the i -th sub-CU obtained by Equation (3), and \overline{var} is the average value of the sub-CUs variance.

Context Information: Select neighboring CUs complexity (NCC) and neighboring CUs depth (NCD), including the maximum, minimum, and average values, to represent the context information because the texture complexity and partition mode between neighboring CUs shown in Figure 4 are very similar due to the spatial correlation of video content. Different values of NCC are expressed as NCC_{max} , NCC_{min} , NCC_{avg} , and different values of NCD under QT and MT are shown as $NCD_{QT_{max}}$, $NCD_{QT_{min}}$, $NCD_{QT_{avg}}$, $NCD_{MT_{max}}$, $NCD_{MT_{min}}$, and $NCD_{MT_{avg}}$, respectively.

Block Information: Three important pieces of block information are selected as features to be extracted, including the width and height of CU, QT depth (D_{QT}), and MT depth (D_{MT}).

4.1.3. Random Forest Training

The random forest classifier (RFC) [45] is composed of multiple independent decision trees. The output of each decision tree has an impact on the final output, which ensures the stability of the RFC output. By randomly selecting equal sample sets, the RFC model reduces the error of imbalanced data and avoids overfitting. Meanwhile, it can process high-dimensional data by randomly selecting subsets of features with equal probability. This paper uses classification and regression tree (CART) [46] as the basic decision tree of RFC.

CART constructs binary trees using the feature and threshold that yield the smallest Gini index at each node. The Gini index of a node m indicates the impurity of data, which is expressed as follows:

$$G(D_m) = \sum_{k=1}^K p_{mk}(1 - p_{mk}) = 1 - \sum_{k=1}^K (p_{mk})^2 \quad (13)$$

$$p_{mk} = \frac{N_{mk}}{N_m} \quad (14)$$

where D_m represents the data at node m with N_m samples, K indicates the number of classification, N_{mk} specifies the number of samples belonging to class k , and p_{mk} means the proportion of class k in node m .

For each candidate split $\theta = (f, t_m)$ consisting of a feature f and threshold t_m , partition the data into $D_m^{left}(\theta)$ and $D_m^{right}(\theta)$. The Gini index is then used to calculate the quality of a proposed split of node m :

$$Q(D_m, \theta) = \frac{N_m^{left}}{N_m} G(D_m^{left}(\theta)) + \frac{N_m^{right}}{N_m} G(D_m^{right}(\theta)) \quad (15)$$

The parameters that minimize the impurity are selected in the next step.

Finally, the recursive process is continuously conducted for subsets $D_m^{left}(\theta^*)$ and $D_m^{right}(\theta^*)$ until the greatest depth permitted has been reached $G(D_m) = 1$ or $N_m = 1$:

$$\theta^* = \arg \min_{\theta} Q(D_m, \theta) \quad (16)$$

The illustration of our random forest (RF) is shown in Figure 8. For training RF, part of the sequences from the JCT-VC standard dataset is selected as the training sequence (7 out of 30, marked with * shown in Tables 5 and 6). To create the training set, we use the first 40 frames of each sequence in each class as the training set, which are encoded with the all-Intra configuration. We reported the performance on both the training sequences (BDBR: 1.2%, ΔT : 56.96%) and the testing sequences (BDBR: 1.21%, ΔT : 57.21%) when utilizing VTM7.0. Similarly, we also reported the performance on the training sequences (BDBR: 1.27%, ΔT : 56.02%) and the testing sequences (BDBR: 1.32%, ΔT : 60.11%) when utilizing VTM23.1. These results indicate that our models possess good generalization ability, despite being trained on a limited subset of frames from a small number of sequences. Table 7 shows the distribution of the CU partition. The proposed RF is trained for 26 features with 10 trees in the forest and maximum depth of 15 features and minimum samples that split a node of 20. Among the trained RFs, six are for RF_{PM} and two are for RF_{ET} . Then, the trained RFs are imported into VTM 7.0.

Table 5. Results of the proposed algorithm compared with VTM7.0 encoder (the training sequences are marked with *).

Class	Sequence	FCPD			FIMP			Overall		
		BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)
A1	Tango2 *	0.49	−0.04	60.25	0.32	−0.02	21.76	0.67	−0.06	66.84
	FoodMarket	0.44	−0.04	59.63	0.94	−0.04	19.68	1.23	−0.06	64.68
	Campfire	1.04	−0.09	56.22	0.74	−0.01	25.69	1.62	−0.11	65.95
A2	Catrobot *	1.02	−0.05	57.8	0.39	−0.02	25.79	1.25	−0.07	64.58
	DaylightRoad2	0.93	−0.07	63.62	0.68	−0.01	27.98	1.44	−0.07	71.88
	ParkRuning3	0.58	−0.03	53.52	0.55	−0.01	20.36	0.99	−0.05	59.67
B	BasketballDrive	1.8	−0.06	61.76	0.75	−0.01	23.47	2.39	−0.07	67.87
	BQTerrace *	1.42	−0.11	63.97	0.79	−0.04	28.74	1.92	−0.14	70.08
	Cactus	1.35	−0.09	63.46	0.48	−0.03	25.64	1.69	−0.11	69.43
	Kimono	1.06	−0.02	66.89	0.22	−0.01	22.71	1.1	−0.03	72.25
	ParkScene	1.82	−0.17	63.35	0.26	−0.01	27.84	1.89	−0.18	72.19
C	BasketballDrill	0.99	−0.2	53.24	0.37	−0.04	29.65	1.22	−0.24	65.63
	BQMall	1.11	−0.1	61.24	0.39	−0.04	27.84	1.39	−0.13	69.26
	PartyScene *	1.3	−0.16	51.93	0.8	−0.01	35.11	2.09	−0.14	65.64
	RaceHorsesC	0.48	−0.15	52.83	0.54	−0.05	32.32	0.93	−0.22	64.16
D	BasketballPass	1.28	−0.16	46.06	0.62	−0.02	31.11	1.6	−0.09	59.74
	BlowingBubbles *	0.83	−0.09	53.78	0.48	−0.01	29.27	1.09	−0.15	63.09
	BQSquare	0.49	−0.08	49.71	0.64	−0.06	33.38	0.93	−0.2	62.11
	RaceHorses	0.36	−0.04	41.18	0.32	−0.07	31.94	0.62	−0.15	56.29
E	FourPeople *	1.72	−0.13	62.53	0.4	−0.07	25.33	2.09	−0.19	70.57
	Johnny	1.9	−0.16	64.29	0.94	−0.06	23.66	2.67	−0.11	72.24
	KristenAndSara	2.12	−0.13	65.58	0.38	−0.06	24.98	2.29	−0.19	72.98
F	BasketballDrillText	1.53	−0.14	55.31	1.08	−0.06	25.29	2.45	−0.17	64.51
	ChinaSpeed	2.17	−0.18	54.31	0.81	−0.04	34.74	2.32	−0.19	66.19
	SlideEditing *	1.65	−0.18	48.44	1.09	−0.04	30.51	2.25	−0.17	59.74
	SlideShow	1.48	−0.18	54.78	0.71	−0.06	31.02	2.09	−0.2	66.58

Table 5. Cont.

Class	Sequence	FCPD			FIMP			Overall		
		BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)
	Training average	1.2	−0.11	56.96	-	-	-	-	-	-
	Test average	1.21	−0.11	57.21	-	-	-	-	-	-
	Average	1.21	−0.11	57.14	0.6	−0.03	27.53	1.62	−0.13	66.31

Table 6. Results of the proposed algorithm compared with VTM23.1 encoder (the training sequences are marked with *).

Class	Sequence	FCPD			FIMP			Overall		
		BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)
A1	Tango2 *	0.57	−0.05	62.28	0.35	−0.03	23.12	0.92	−0.07	69.5
	FoodMarket	0.58	−0.07	62.34	1.01	−0.05	21.22	1.49	−0.08	67.65
	Campfire	1.19	−0.3	58.90	0.79	−0.02	27.23	1.86	−0.13	70.06
	Drums	1.32	−0.16	60.60	0.82	−0.03	28.17	1.73	−0.14	68.28
A2	Catrobot1 *	1.09	−0.06	59.86	0.43	−0.04	27.24	1.38	−0.08	67.56
	DaylightRoad2	1.10	−0.10	66.30	0.8	−0.03	29.24	1.46	−0.08	74.9
	ParkRuning3	0.73	−0.07	56.22	0.59	−0.03	22.31	1.28	−0.06	69.11
	TrafficFlow	0.61	−0.06	54.73	0.47	−0.02	20.89	1.23	−0.05	66.82
B	BasketballDrive	1.93	−0.11	64.46	0.8	−0.03	25.34	2.42	−0.08	70.51
	BQTerrace *	1.48	−0.13	66.05	0.85	−0.05	30.35	1.97	−0.15	75.22
	Cactus	1.48	−0.14	66.16	0.52	−0.04	27.44	1.71	−0.13	76.92
	Kimono	1.19	−0.07	69.59	0.25	−0.03	24.41	1.45	−0.04	74.44
	ParkScene	1.93	−0.21	66.04	0.3	−0.03	29.57	1.92	−0.19	74.64
	MarketPlace	1.38	−0.15	64.66	0.71	−0.03	27.88	1.78	−0.13	69.34
	RitualDance	1.47	−0.14	64.21	0.79	−0.03	25.66	1.84	−0.13	66.28
C	BasketballDrill	1.13	−0.24	55.93	0.41	−0.05	31.34	1.49	−0.25	67.89
	BQMall	1.24	−0.14	63.93	0.45	−0.05	29.45	1.41	−0.14	72.07
	PartyScene *	1.37	−0.17	54	0.83	−0.03	36.39	2.13	−0.15	68.26
	RaceHorsesC	0.63	−0.19	55.52	0.57	−0.06	34.28	1.27	−0.23	66.36
D	BasketballPass	1.39	−0.20	48.75	0.64	−0.03	32.38	1.62	−0.09	66.72
	BlowingBubbles *	0.91	−0.11	55.85	0.54	−0.03	31.44	1.14	−0.17	67.12
	BQSquare	0.63	−0.11	52.41	0.66	−0.07	34.67	1.07	−0.23	64.78
	RaceHorses	0.52	−0.07	43.88	0.36	−0.09	33.43	1.04	−0.15	59.32
E	FourPeople *	1.79	−0.14	64.6	0.42	−0.09	27.61	2.15	−0.19	72.81
	Johnny	1.21	−0.20	66.98	0.96	−0.07	25.52	2.71	−0.13	74.59
	KristenAndSara	2.23	−0.17	68.27	0.4	−0.08	26.43	2.31	−0.2	75.41
F	BasketballDrillText	1.67	−0.18	58.00	1.1	−0.07	27.57	2.49	−0.19	67.98
	ChinaSpeed	2.28	−0.22	56.99	0.85	−0.05	36.78	2.34	−0.2	68.28
	SlideEditing *	1.71	−0.19	50.51	1.19	−0.05	32.82	2.28	−0.19	61.91
	SlideShow	1.59	−0.22	57.57	0.79	−0.07	33.2	2.12	−0.21	69.34
	Training average	1.27	−0.12	59.02	-	-	-	-	-	-
	Test average	1.32	−0.15	60.11	-	-	-	-	-	-
	Average	1.31	−0.14	59.86	0.66	−0.05	28.78	1.73	−0.14	69.47

Table 7. Partition distribution of training set.

Partition Type	QP 22	QP 27	QP 32	QP37
NS	5,079,696	3,492,693	3,243,862	2,353,272
QT	771,644	505,940	442,246	302,200
BH	2,444,758	1,480,714	1,178,438	744,162
BV	2,348,766	1,433,616	1,147,124	723,560
TH	595,808	366,073	349,944	242,860
TV	586,848	380,857	353,132	241,424

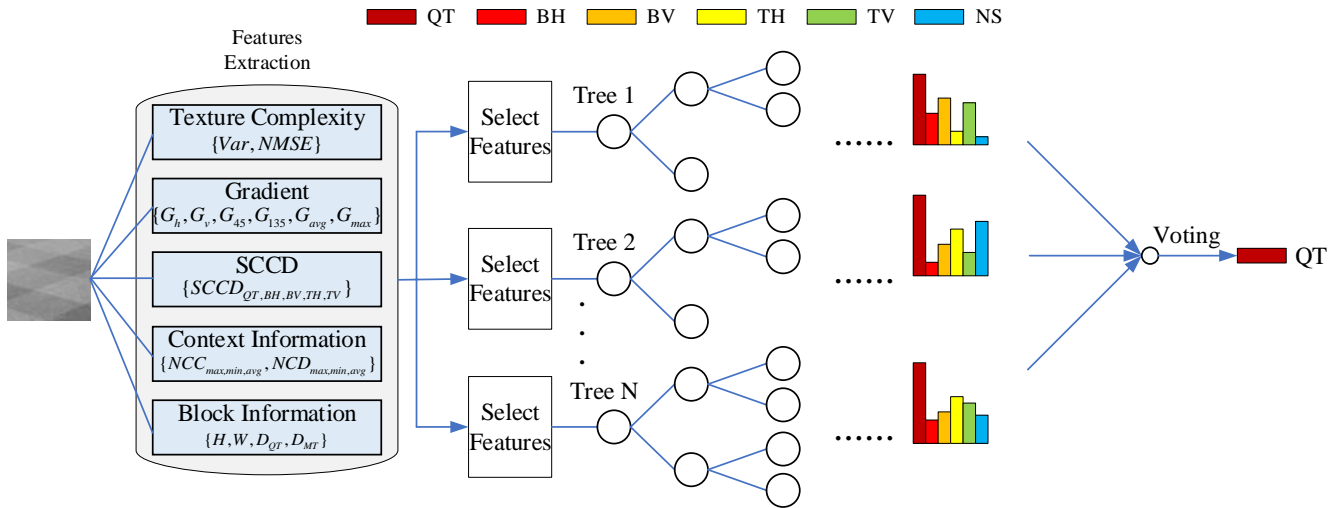


Figure 8. Illustration of the random forest RF_{PM} or RF_{ET} .

4.2. Fast Intra Mode Prediction

In [25], Yang et al. proposed the one-dimensional gradient decent search method to reduce the RMD search range. Based on our above analysis in Section 3, we employ three texture features to design a hierarchical search method to further reduce the search range as shown in Figure 9.

First, flat CUs tend to choose the planar or DC mode as the best intra mode. Thus, variance threshold τ_f of a CU is carefully tuned to divide the CUs into two categories, including flat and non-flat. For flat CUs, the RMD process is skipped, and only DC and planar modes are checked in the RDO process to obtain the best intra mode. In our experiments, τ_f is set as $0.3 \times QP$.

Second, the optimal intra mode direction is almost consistent with the texture direction. In other words, the prediction orientation deviating from the texture direction is rarely chosen as the best intra mode. Thus, the direction information of CU is conducted to narrow down the search range of the coarse search process.

- (1) The 65 directional prediction modes are divided into four groups. Prediction modes from 10 to 26 are categorized as the horizontal group (C_h), prediction modes from 42 to 58 are categorized as the vertical group (C_v), prediction modes from 2 to 10 and 58 to 66 are categorized as the 45° diagonal group (C_{45}), and prediction modes 26 to 42 are categorized as the 135° diagonal group (C_{135}).
- (2) The gradients of four directions are computed, including horizontal (G_{HOR}), vertical (G_{VER}), diagonal down right (G_{DDR}), and diagonal down left (G_{DDL}). The maximum value of $G_{HOR}, G_{VER}, G_{DDR}, G_{DDL}$ is denoted as G_{max} . Only if G_{max}/G_i is larger than a threshold τ_d is the corresponding prediction mode group added into the coarse search range, where the $i \in \{HOR, VER, DDR, DDL\}$. One exception is that all 67 prediction modes are added into the search range only if all G_{max}/G_i are smaller than τ_d . In our experiments, τ_d is set to 1.5.

Third, the optimal prediction modes of the neighboring CUs are usually similar, which is represented as the MPM list. It is very helpful for choosing the initial point for the coarse search. The RMD process is conducted on the six modes in the MPM list. The mode with the smallest HAD cost is chosen as the initial search point for the coarse search process. If the initial search point is not within the search range, all 65 modes are included in the search range.

Fourth, the optimal mode M_c of the coarse search is set as the initial point for the fine search, and the search range is from $M_c - 2$ to $M_c + 2$, and the step size is 1. Finally, the two modes with the smallest HAD cost after the fine search are checked through the RDO process. The mode with the smallest RDO cost is chosen as the optimal intra prediction mode. The planar mode is not included in the two modes, and the RDO cost of the planar mode also needs to be compared.

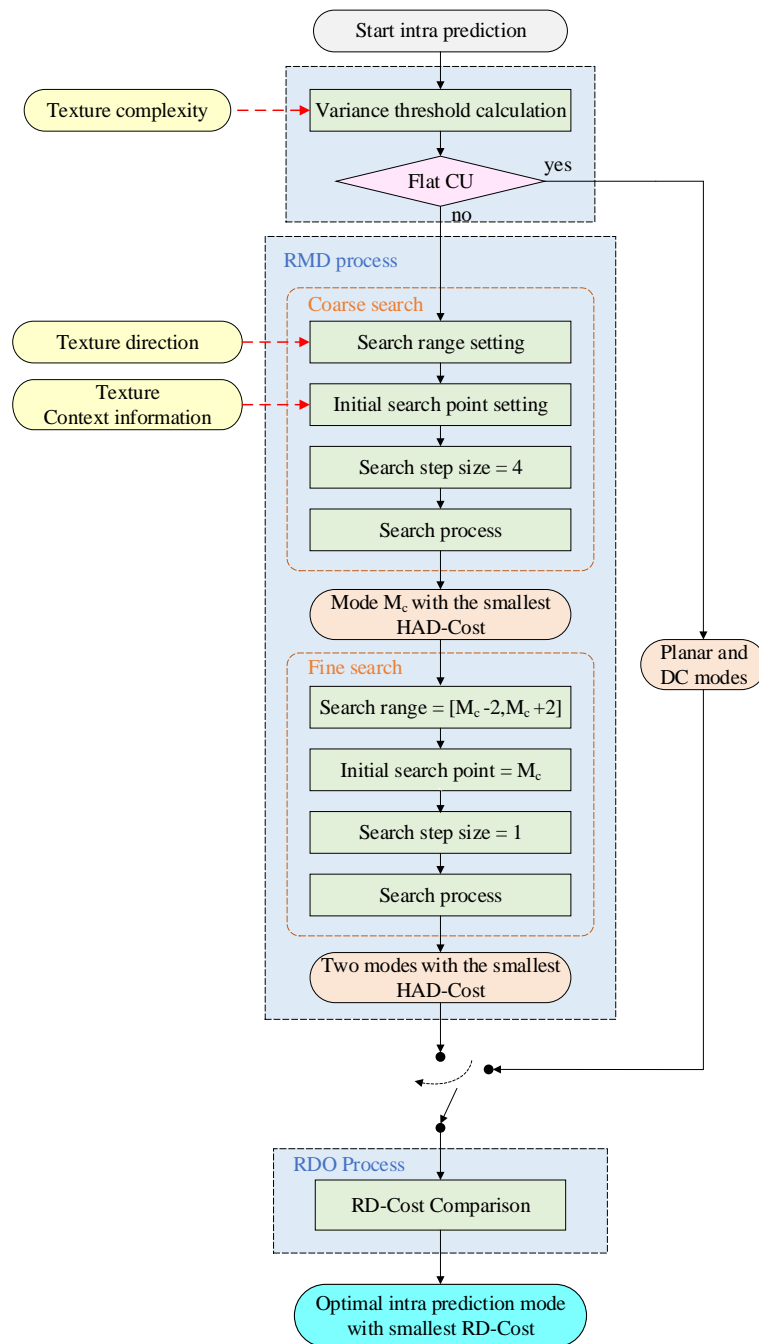


Figure 9. Flowchart of the hierarchical search method for fast intra mode prediction.

5. Experimental Results

We implement the proposed method on top of both VTM 7.0 and VTM 23.1. In the evaluation process, we use all of the video sequences from JCT-VC standard test set and fix quantization parameters to 22, 27, 32, 37, named as QP values. For quantitative analysis, we employ BDBR and BDPSNR for the rate-distortion assessment, as well as the time-saving rate for estimating the complexity reduction. The time-saving rate ΔT is defined as

$$\Delta T = \frac{T_{VTM} - T_{Fast}}{T_{VTM}} \quad (17)$$

where T_{VTM} is the encoding time of the original VTM 7.0 or VTM 23.1, and T_{Fast} is the encoding time of our improved fast method. We practice our method on a laptop with Intel Core i7-9750H and 2.6 GHz GPU.

5.1. Performance Evaluation of Proposed Algorithm

Table 5 shows the test performance of the proposed fast CU partition decision (FCPD), fast intra mode prediction (FIMP), and the overall method. It can be seen that FCPD can save the coding time by 66.89% at most, 41.18% at least, and 57.14% on average, with only 1.21% BDBR increasing and 0.11 dB BDPSNR decreasing on average. Meanwhile, the results also show that the performance between testing sequences and training sequences (the sequences with *) are almost the same, which indicates that our two random forest models have good generalization ability.

FIMP can save coding time by 35.11% at most, 19.68% at least, and 27.53% on average, with only 0.6% BDBR increasing and 0.03 dB BDPSNR decreasing on average, which proves that within acceptable loss, FIMP significantly improves efficiency by eliminating some needless modes. In Table 5, we also calculate the overall performance of the proposed fast algorithm composed of FCPD and FIMP. The experimental results show that approximately 66.31% of the coding time is saved with a tolerable 1.62% BDBR performance loss on average. It can be seen that the proposed method significantly reduces the coding complexity, making it more suitable for power-constrained application scenarios. In addition, as shown in Table 6, we test all sequences on top of VTM23.1, and the results show that our algorithm also achieves the maximum time saving within the acceptable loss range of BDBR.

5.2. Comparison with Others

We compare our algorithm with three of the most advanced methods, namely, those of Ni 2022 [14], Wang 2023 [21], and Li 2024 [22]. As shown in Table 8, the $T/B = TS/BDBR$ denotes the measurement for the trade-off between the time savings and BDBR performance. While the proposed method may not achieve the optimal balance, it stands out in maximizing the encoding time savings among other advanced techniques, while keeping the BDBR loss within an acceptable range. It can be seen that our algorithm offers substantial time savings while maintaining an acceptable level of BDBR loss. The relative time-saving improvements are 27.38%, 15.45%, and 13.32%, respectively. Meanwhile, the relative BDBR losses incurred are 1.21%, 0.58%, and 0.52%, respectively.

A quantitative analysis comparing our model with deep learning-based methods is presented in Table 9. This includes methods from Zan 2023 [32], Wu 2022 [27], and Chen 2023 [28]. It can be seen that our algorithm on top of VTM 23.1 achieves the maximum time saving of 69.85% with the minimum BDBR loss of 1.65% (the time saving of the proposed method on top VTM 7.0 is slightly reduced to 66.69%). The particularly high T/B ratio (69.85) demonstrates the superiority of our model in achieving the trade-off between time savings and BDBR loss in comparison to deep learning-based methods.

Table 8. Performance comparison between the proposed algorithm and state-of-the-art algorithms.

Class	Sequence	Ni 2022 [14] (VTM11.0)			Wang 2023 [21] (VTM10.0)			Li 2024 [22] (VTM7.0)			Ours (VTM7.0)			Ours (VTM23.1)		
		BDBR (%)	TS (%)	$\frac{T}{B}$	BDBR (%)	TS (%)	$\frac{T}{B}$	BDBR (%)	TS (%)	$\frac{T}{B}$	BDBR (%)	TS (%)	$\frac{T}{B}$	BDBR (%)	TS (%)	$\frac{T}{B}$
A1	Tango2	0.58	47.35	81.68	/	/	/	0.99	56.85	57.42	0.67	66.84	99.76	0.92	69.5	75.54
	FoodMarket	0.52	50.29	96.71	1.14	54.85	48.11	0.97	53.53	55.19	1.23	64.68	52.59	1.49	67.65	45.40
	Campfire	0.26	43.58	167.62	1.25	57.16	45.73	0.86	49.67	57.76	1.62	65.95	40.71	1.86	70.06	37.67
	Drums	/	/	/	1.37	58.04	42,36	/	/	/	/	/	/	1.73	68.28	39.47
A2	Catrobot1	0.37	41.43	111.97	/	/	/	1.12	55.99	49.99	1.25	64.58	51.66	1.38	67.56	48.96
	DaylightRoad2	0.26	33.29	128.04	1.23	52.29	42.51	0.97	54.53	56.22	1.44	71.88	49.92	1.46	74.9	51.30
	ParkRuning3	0.27	41.57	153.96	1.31	55.45	42.32	1.09	53.57	49.15	0.99	59.67	60.27	1.28	69.11	53.99
	TrafficFlow	/	/	/	0.92	54.39	59.12	/	/	/	/	/	/	1.23	66.82	54.33
B	BasketballDrive	0.36	44.73	124.25	/	/	/	/	/	/	2.39	67.87	28.40	2.42	70.51	29.14
	BQTerrace	0.3	40.05	133.50	1.03	54.98	53.38	1.29	52.41	40.63	1.92	70.08	36.5	1.97	75.22	38.18
	Cactus	0.4	42	105	1.28	56.81	44.38	/	/	/	1.69	69.43	41.08	1.71	76.92	44.98
	Kimono	0.34	43.03	126.56	1.36	57.47	42.26	1.22	57.56	47.18	1.1	72.25	65.68	1.45	74.44	51.34
	ParkScene	0.37	40	108.11	/	/	/	0.95	59.22	62.34	1.89	72.19	38.20	1.92	74.64	38.88
	MarketPlace	0.58	40.41	69.67	0.91	51.89	57.02	/	/	/	/	/	/	1.78	69.34	38.96
	RitualDance	0.43	42.55	98.95	/	/	/	/	/	/	/	/	/	1.84	66.28	36.02
C	BasketballDrill	1.16	47.88	41.28	0.76	49.38	64.97	0.94	50.68	53.91	1.22	65.63	53.80	1.49	67.89	45.56
	BQMall	0.52	45.43	87.37	1.04	50.66	48.71	/	/	/	1.39	69.26	49.83	1.41	72.07	51.11
	PartyScene	0.32	41.5	129.69	0.85	55.92	65.79	0.95	53.66	56.48	2.09	65.64	31.41	2.13	68.26	32.05
	RaceHorsesC	0.29	44.9	154.83	1.09	56.24	51.6	0.91	53.75	59.07	0.93	64.16	69	1.27	66.36	52.25
D	BasketballPass	0.44	42.02	95.5	0.94	52.43	55.78	/	/	/	1.6	59.74	37.34	1.62	66.22	40.88
	BlowingBubbles	0.39	40.66	104.26	0.93	53.61	57.65	0.91	54.88	60.31	1.09	63.09	57.88	1.14	67.12	58.88
	BQSquare	0.36	40.3	111.94	0.86	53.28	61.95	1.16	55.55	47.89	0.93	62.11	66.78	1.07	64.78	60.54
	RaceHorses	0.37	41.54	112.27	1.07	57.39	53.63	1.01	49.45	48.96	0.62	56.29	90.79	1.04	59.32	57.04
E	FourPeople	0.62	39.57	63.82	/	/	/	0.89	58.72	65.98	2.09	70.57	33.77	2.15	72.81	33.87
	Johnny	0.62	44.16	71.23	/	/	/	0.96	59.33	61.80	2.67	72.24	27.06	2.71	74.59	27.52
	KristenAndSara	0.55	41.19	74.89	/	/	/	1.23	57.62	46.85	2.29	72.98	31.87	2.31	75.41	32.65
Average		0.44	42.47	96.52	1.07	54.4	50.84	1.02	54.83	53.75	1.51	66.69	44.17	1.65	69.85	42.33

Table 9. Performance comparison with deep learning methods.

Class	Sequence	Wu 2022 [27] (VTM7.0)			Chen 2023 [28] (VTM14.0)			Zan 2023 [32] (VTM7.0)			Ours (VTM7.0)			Ours (VTM23.1)		
		BDBR (%)	TS (%)	$\frac{T}{B}$	BDBR (%)	TS (%)	$\frac{T}{B}$	BDBR (%)	TS (%)	$\frac{T}{B}$	BDBR (%)	TS (%)	$\frac{T}{B}$	BDBR (%)	TS (%)	$\frac{T}{B}$
A1	Tango2	1.52	66.71	43.89	1.98	52.45	26.49	/	/	/	0.67	66.84	99.76	0.92	69.5	75.54
	FoodMarket	1.57	50.97	32.46	1.18	52.91	44.84	/	/	/	1.23	64.68	52.59	1.49	67.65	45.40
	Campfire	2.20	64.08	29.13	1.45	59.86	41.28	/	/	/	1.62	65.95	14.71	1.86	70.06	37.67
	Drums	/	/	/	/	/	/	/	/	/	/	/	/	1.73	68.28	39.47
	Average (A1)	1.76	60.59	34.43	1.54	55.07	35.76	1.89	63.97	33.85	1.17	65.82	56.26	1.50	68.87	45.91
A2	Catrobot1	2.37	65.4	27.59	1.75	43.27	24.73	/	/	/	1.25	64.58	51.66	1.38	67.56	48.96
	DaylightRoad2	1.8	71.2	39.56	1.67	50.49	30.23	/	/	/	1.44	71.88	49.92	1.46	74.9	51.30
	ParkRuning3	1.26	58.94	46.78	0.79	48.37	61.23	/	/	/	0.99	59.67	60.27	1.28	69.11	53.99
	TrafficFlow	/	/	/	/	/	/	/	/	/	/	/	/	1.23	66.82	54.33
	Average (A2)	1.81	65.18	36.01	1.40	47.38	33.84	2.00	68.82	34.41	1.23	65.38	53.15	1.34	69.60	51.94
B	BasketballDrive	1.32	76.68	58.09	2.22	50.38	22.69	/	/	/	2.39	67.87	28.40	2.42	70.51	29.14
	BQTerrace	2.98	62.57	21.00	2.61	47.38	18.15	/	/	/	1.92	70.08	36.50	1.97	75.22	38.18
	Cactus	2.02	70.67	34.99	2.03	46.57	22.94	/	/	/	1.69	69.43	41.08	1.71	76.92	44.98
	Kimono	2.09	74.18	35.49	/	/	/	/	/	/	1.1	72.25	65.68	1.45	74.44	51.34
	ParkScene	2.16	65	30.09	/	/	/	/	/	/	1.89	72.19	38.20	1.92	74.64	38.88
	MarketPlace	/	/	/	1.25	49.22	39.38	/	/	/	/	/	/	1.78	69.34	38.96
	RitualDance	/	/	/	1.83	49.82	27.22	/	/	/	/	/	/	1.84	66.28	36.02
	Average (B)	2.11	69.82	33.09	1.99	48.67	24.46	2.35	73.33	31.20	1.80	70.36	39.09	1.87	72.48	38.76
C	BasketballDrill	3.65	57.35	15.71	3.29	50.38	15.31	/	/	/	1.22	65.63	53.80	1.49	67.89	45.56
	BQMall	2.33	65.65	28.18	2.47	58.55	23.70	/	/	/	1.39	69.26	49.83	1.41	72.07	51.11
	PartyScene	2.07	59.27	28.63	1.85	49.33	26.66	/	/	/	2.09	65.64	31.41	2.13	68.26	32.05
	RaceHorsesC	1.52	64.92	42.71	1.73	50.40	29.13	/	/	/	0.93	64.16	68.99	1.27	66.36	52.25
	Average (C)	2.39	61.80	25.86	2.34	52.17	22.29	2.50	67.06	26.82	1.41	66.17	46.93	1.58	68.65	43.45
D	BasketballPass	2.33	59.25	25.43	2.22	51.84	23.35	/	/	/	1.6	59.74	37.34	1.62	66.22	40.88
	BlowingBubbles	3.15	54.63	17.34	1.67	53.27	31.90	/	/	/	1.09	63.09	57.88	1.14	67.12	58.88
	BQSquare	1.95	59.64	30.58	2.91	55.85	19.19	/	/	/	0.93	62.11	66.78	1.07	64.78	60.54
	RaceHorses	2.68	60.96	22.75	1.93	55.08	28.54	/	/	/	0.62	56.29	90.79	1.04	59.32	57.04
	Average (D)	2.53	58.62	23.17	2.18	54.01	24.78	2.16	64.58	29.90	1.06	60.31	56.90	1.22	64.36	52.75
E	FourPeople	2.80	70.00	25.00	2.68	57.10	21.31	/	/	/	2.09	70.57	33.77	2.15	72.81	33.87
	Johnny	2.82	68.11	24.15	2.93	56.60	19.32	/	/	/	2.67	72.24	27.06	2.71	74.59	27.52
	KristenAndSara	2.88	67.92	23.58	3.00	50.21	16.74	/	/	/	2.29	72.98	31.87	2.31	75.41	32.65
	Average (E)	2.83	68.68	24.27	2.87	54.64	19.04	3.11	73.75	23.71	2.35	71.93	30.61	2.39	74.27	31.08
Average		2.19	65.53	29.92	2.07	51.79	25.02	2.33	68.76	29.51	1.51	66.69	44.17	1.65	69.85	42.33

5.3. Ablation Study

We conduct ablation experiments to analyze the impact of major components in the proposed algorithm. For the FCPD scheme, we first investigate the prediction accuracy of the two random forests, including RF_{PM} and RF_{ET} . Figure 10 illustrates the prediction accuracies of RF_{PM} and RF_{ET} for different sequences under four QP settings. It can be observed that the prediction accuracies of RF_{PM} and RF_{ET} are both beyond 90%. The prediction accuracies are also stable for different sequences and different QP settings. Therefore, the FCPD scheme can efficiently maintain the RD performance. Moreover, in order to evaluate the effectiveness of the employed features, the importance proportion of the five types of features in the random forest classification is shown in Table 10. It can be observed that the contribution of each feature is beyond 10%, and the sub-CUs complexity difference contributes the most (beyond 30%). It demonstrates that all of the employed features are effective and representative.

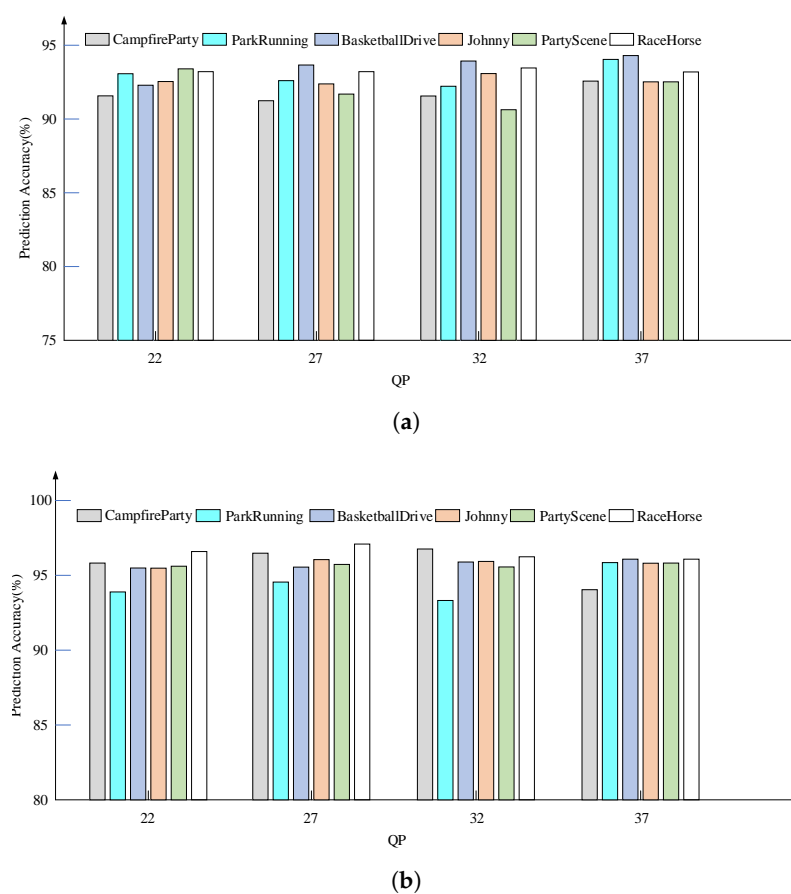


Figure 10. The accuracy of the two random forest classifiers. (a) RF_{PM} ; (b) RF_{ET} .

Table 10. The importance of features.

Features	Proportion	
	RF_{PM}	RF_{ET}
Block Information	0.17	0.2
Texture Complexity	0.1	0.11
Gradient Information	0.16	0.16
Sub-CUs Complexity difference	0.39	0.33
Context Information	0.18	0.2

The proposed FCPD scheme is composed of accelerating of the simple, fuzzy, and complex CU partition decision. Figure 11 reports the time saving of each component.

FCPD improves complex CUs the most, which saves 30–60% encoding time. Meanwhile, predicting the optimal partition mode for simple CUs and the early termination process for fuzzy CUs also bring considerable complexity reduction, about 10–25% encoding time saving, respectively.

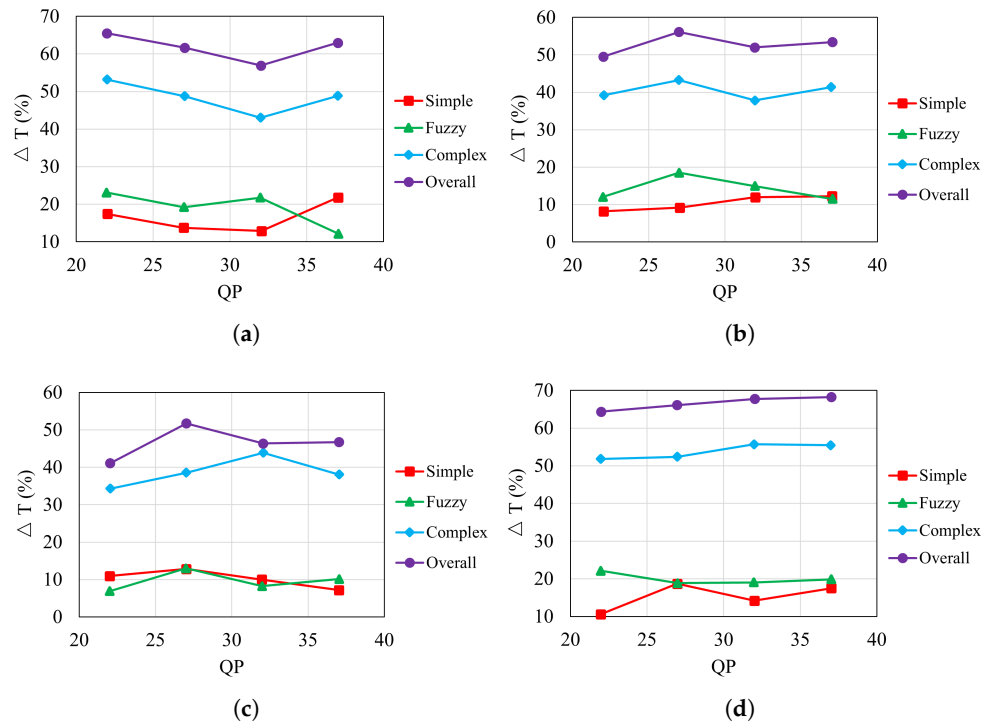


Figure 11. Influence of different components in FCPD for different sequences. (a) *BasketballDrive*; (b) *RaceHorses*; (c) *BasketballPass*; (d) *Johnny*.

For the FIMP scheme, we investigate the prediction accuracy, which indicates the ratio of the CUs that the predicted intra modes by FIMP are same as the original VTM encoder. Overall, as shown in Figure 12, the average accuracy of different sequences is around 90% and the accuracy is slightly higher with the increasing of QPs.

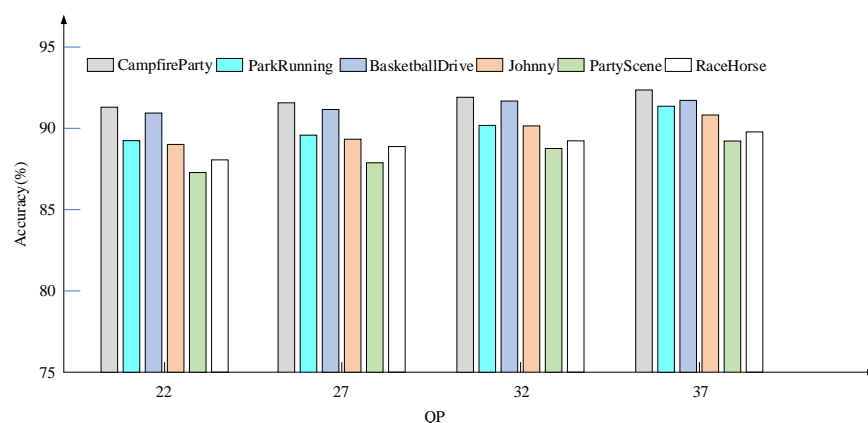


Figure 12. The accuracy of FIMP.

To the proposed FCPD, FIMP, and overall algorithm, we compare the RD performance and time saving under four different QP values as shown in Figure 13. Figure 13a shows that the RD performance results of FCPD, FIMP, and the overall algorithm are almost comparable to the VTM encoder. Figure 13b indicates that the proposed algorithm maintains similar complexity reduction over different QP values.

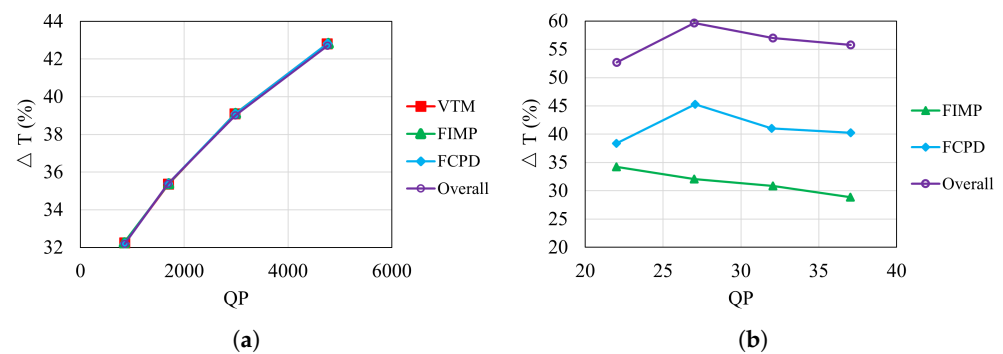


Figure 13. Performance results of the proposed FCPD, FIMP, and overall algorithm compared with VTM-7.0 encoder of RaceHorseC. (a) RD curves; (b) time saving under different QPs.

6. Conclusions

We propose a fast VVC intra coding method for power-constrained applications. The proposed method is divided into two parts. First, a fast CU partition scheme is proposed based on two random forest classifiers. The CUs are categorized as simple, fuzzy, and complex CUs based on their texture characteristics. One random forest classifier predicts the best partition modes of simple and complicated CUs directly, while another random forest identifies the early termination of the partition process for fuzzy CUs. This approach not only maximizes the potential for speed improvement but also effectively mitigates bitrate distortion (BDBR) losses, particularly for fuzzy CUs. Second, a hierarchical search method based on texture features is proposed to accelerate the intra mode prediction process. Experimental findings indicate that our algorithm obtains significant complexity reduction with acceptable loss compared with the latest VVC reference software (VTM 23.1). Further, when compared to state-of-the-art methods, the proposed method also achieves the largest time saving with comparable BDBR loss.

Author Contributions: Writing—original draft, L.C.; Writing—review and editing, B.C.; Writing—review and editing, H.Z.; Visualization, H.Q.; Supervision, L.D.; Project administration, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61501074.

Data Availability Statement: The data presented in this study are available on reasonable request from the corresponding author.

Conflicts of Interest: Lihua Deng is employed by China Telecom Research Institute. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

VCC	versatile video coding
HEVC	high-efficiency video coding
CU	coding unit
QTMT	quadtree with nested multi-type tree
BDBR	Bjontegaard delta bit rate
BDPSNR	Bjontegaard delta peak signal-to-noise rate
QT	quadtree
CTU	coding tree unit
MT	multi-type
BV	vertical binary tree
BH	horizontal binary tree

TV	vertical ternary tree
TH	horizontal ternary tree
RDO	rate-distortion optimization
TS-FMD	the classical three-step fast intra mode decision
RMD	rough mode decision
MPM	most probable model
SVM	support vector machine
LPTCM	Laplace transparent composite model
CNN	convolutional neural network
AGH	the average gradients in the horizontal
AGV	the average gradients in the vertical
pRMS	progressive rough mode search
NMSE	neighboring mean squared error
DDR	diagonal down right
DDL	diagonal down left
SCCD	sub-CUs complexity difference
NCC	neighboring CUs complexity
NCD	neighboring CUs depth
RFC	random forest classifier
CART	classification and regression tree
RF	random forest
FCPD	fast CU partition decision
FIMP	fast intra mode prediction

References

1. Bross, B. Versatile Video Coding (Draft 1). In Proceedings of the Joint Video Exploration Team (JVET), San Diego, CA, USA, 10–20 April 2018.
2. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [[CrossRef](#)]
3. Liu, S.; Brass, B.; Chen, J. Versatile Video Coding (Draft 2). In Proceedings of the 11th JVET Meeting, Ljubljana, Slovenia, 10–18 July 2018; Volume 22, pp. 1649–1668.
4. Lin, S.; Chen, H.; Zhang, H.; Maxim, S.; Yang, H.; Zhou, J. Affine Transform Prediction for Next Generation Video Coding. In *Huawei Technologies, International Organisation for Standardisation Organisation Internationale De Normalisation ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, ISO/IEC JTC1/SC29/WG11 MPEG2015/m37525*; ISO: Geneva, Switzerland, 2015.
5. Bossen, F.; Li, X.; Suehring, K. AHG report: Test model software development (AHG3). In Proceedings of the Joint Video Exploration Team (JVET), San Diego, CA, USA, 10–20 April 2018.
6. Chen, C.W. Internet of Video Things: Next-Generation IoT With Visual Sensors. *IEEE Internet Things J.* **2020**, *7*, 6676–6685. [[CrossRef](#)]
7. Bross, B.; Chen, J.; Liu, S. Versatile Video Coding (Draft 5). In Proceedings of the Joint Video Exploration Team (JVET), Geneva, Switzerland, 19–27 March 2019.
8. Chang, Y.; Jhu, H.; Jiang, H.; Zhao, L.; Zhao, X.; Li, X.; Liu, S.; Bross, B.; Keydel, P.; Schwarz, H.; et al. Multiple Reference Line Coding for Most Probable Modes in Intra Prediction. In Proceedings of the 2019 Data Compression Conference (DCC), Snowbird, UT, USA, 26–29 March 2019; p. 559.
9. De-Luxn-Hernandez, S.; Valeri, G.; Ma, J.; Tung, N.; Schwarz, H.; Marpe, D.; Wiegand, T. An Intra Subpartition Coding Mode for VVC. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1203–1207.
10. Piao, Y.; Min, J.; Chen, J. Encoder Improvement of Unified Intra Prediction. In Proceedings of the Joint Collaborative Team on Video Coding (JCT-VC), Guangzhou, China, 7–15 October 2010.
11. Chen, J.; Chen, Y.; Karczewicz, M.; Li, X.; Liu, H.; Zhang, L.; Zhao, X. Coding tools investigation for next generation video coding based on HEVC. In Proceedings of the Applications of Digital Image Processing XXXVIII, San Diego, CA, USA, 10–13 August 2015.
12. Gu, J.; Tang, M.; Wen, J.; Han, Y. Adaptive Intra Candidate Selection With Early Depth Decision for Fast Intra Prediction in HEVC. *IEEE Signal Process. Lett.* **2018**, *25*, 159–163. [[CrossRef](#)]
13. Li, Y.; Yang, G.; Song, Y.; Zhang, H.; Ding, X.; Zhang, D. Early Intra CU Size Decision for Versatile Video Coding Based on a Tunable Decision Model. *IEEE Trans. Broadcast.* **2021**, *67*, 710–720. [[CrossRef](#)]
14. Ni, C.T.; Lin, S.H.; Chen, P.Y.; Chu, Y.T. High Efficiency Intra CU Partition and Mode Decision Method for VVC. *IEEE Access* **2022**, *10*, 77759–77771. [[CrossRef](#)]

15. Fu, T.; Zhang, H.; Mu, F.; Chen, H. Fast CU Partitioning Algorithm for H.266/VVC Intra-Frame Coding. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019; pp. 55–60.
16. Lei, M.; Luo, F.; Zhang, X.; Wang, S.; Ma, S. Look-Ahead Prediction Based Coding Unit Size Pruning for VVC Intra Coding. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 4120–4124.
17. Saldanha, M.; Sanchez, G.; Marcon, C.; Agostini, L. Fast Partitioning Decision Scheme for Versatile Video Coding Intra-Frame Prediction. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Sevilla, Spain, 10–21 October 2020; pp. 1–5.
18. Cui, J.; Zhang, T.; Gu, C.; Zhang, X.; Ma, S. Gradient-Based Early Termination of CU Partition in VVC Intra Coding. In Proceedings of the 2020 Data Compression Conference (DCC), Snowbird, UT, USA, 24–27 March 2020; pp. 103–112.
19. Fan, Y.; Chen, J.; Sun, H.; Katto, J.; Jing, M. A Fast QTMT Partition Decision Strategy for VVC Intra Prediction. *IEEE Access* **2020**, *8*, 107900–107911. [[CrossRef](#)]
20. Kim, H.; Park, R. Fast CU Partitioning Algorithm for HEVC Using an Online-Learning-Based Bayesian Decision Rule. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 130–138. [[CrossRef](#)]
21. Wang, F.; Wang, Z.; Zhang, Q. FSVM- and DAG-SVM-Based Fast CU-Partitioning Algorithm for VVC Intra-Coding. *Symmetry* **2023**, *15*, 1078. [[CrossRef](#)]
22. Li, M.; Wang, Z.; Zhang, Q. Fast CU size decision and intra-prediction mode decision method for H.266/VVC. *EURASIP J. Image Video Process.* **2024**, *7*, 7. [[CrossRef](#)]
23. Erabadda, B.; Mallikarachchi, T.; Kulupana, G.; Fernando, A. Content Adaptive Fast CU Size Selection for HEVC Intra-Prediction. In Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 11–13 January 2019; pp. 1–2.
24. Shan, Y.; Yang, E. Fast HEVC intra coding algorithm based on machine learning and Laplacian Transparent Composite Model. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 2642–2646.
25. Yang, H.; Shen, L.; Dong, X.; Ding, Q.; An, P.; Jiang, G. Low-Complexity CTU Partition Structure Decision and Fast Intra Mode Decision for Versatile Video Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 1668–1682. [[CrossRef](#)]
26. Zhang, Q.; Wang, Y.; Huang, L.; Jiang, B. Fast CU Partition and Intra Mode Decision Method for H.266/VVC. *IEEE Access* **2020**, *8*, 117539–117550. [[CrossRef](#)]
27. Wu, S.; Shi, J.; Chen, Z.; HG-FCN: Hierarchical Grid Fully Convolutional Network for Fast VVC Intra Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 5638–5649. [[CrossRef](#)]
28. Chen, J.-J.; Chou, Y.-G.; Jiang, C.-S. Speed Up VVC Intra-Coding by Learned Models and Feature Statistics. *IEEE Access* **2023**, *11*, 124609–124623. [[CrossRef](#)]
29. Zhang, Y.; Wang, G.; Tian, R.; Xu, M.; Kuo, C.C.J. Texture-Classification Accelerated CNN Scheme for Fast Intra CU Partition in HEVC. In Proceedings of the 2019 Data Compression Conference (DCC), Snowbird, UT, USA, 26–29 March 2019; pp. 241–249.
30. Li, Y.; Li, L.; Fang, Y.; Peng, H.; Ling, N. Bagged tree and ResNet-based joint end-to-end fast CTU partition decision algorithm for video intra coding. *Electronics* **2022**, *11*, 1264. [[CrossRef](#)]
31. Wang, T.; Wei, G.; Li, H.; Bui, T.; Zeng, Q.; Wang, R. A Method to Reduce the Intra-Frame Prediction Complexity of HEVC Based on D-CNN. *Electronics* **2023**, *12*, 2091. [[CrossRef](#)]
32. Zan, Z.; Huang, L.; Chen, S.; Zhang, X.; Zhao, Z.; Yin, H.; Fan, Y. Fast QTMT Partition for VVC Intra Coding Using U-Net Framework. In Proceedings of the 2023 IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia, 8–11 October 2023; pp. 600–604.
33. Tang, G.; Jing, M.; Zeng, X.; Fan, Y. Adaptive CU Split Decision with Pooling-variable CNN for VVC Intra Encoding. In Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP), Sydney, Australia, 1–4 December 2019; pp. 1–4.
34. Tissier, A.; Hamidouche, W.; Vanne, F.G.J.; Menard, D. CNN Oriented Complexity Reduction Of VVC Intra Encoder. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Virtual Conference, 25–28 October 2020; pp. 3139–3143.
35. da Silva, T.L.; Agostini, L.V.; da Silva Cruz, L.A. Fast HEVC intra prediction mode decision based on EDGE direction information. In Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, 27–31 August 2012; pp. 1214–1218.
36. Zhang, T.; Sun, M.T.; Gao, W. Fast Intra-Mode and CU Size Decision for HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 1714–1726. [[CrossRef](#)]
37. Zhang, D.; Chen, Y.; Izquierdo, E. Fast intra mode decision for HEVC based on texture characteristic from RMD and MPM. In Proceedings of the 2014 IEEE Visual Communications and Image Processing Conference, Valletta, Malta, 7–10 December 2014; pp. 510–513. [[CrossRef](#)]
38. Gwon, D.; Choi, H.; Youn, J.M. HEVC fast intra mode decision based on edge and SATD cost. In Proceedings of the 2015 Asia Pacific Conference on Multimedia and Broadcasting, Bali, Indonesia, 23–25 April 2015; pp. 1–5.
39. Zhang, H.; Ma, Z. Fast Intra Mode Decision for High Efficiency Video Coding (HEVC). *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 660–668. [[CrossRef](#)]
40. Jamali, M.; Coulombe, S. Fast HEVC Intra Mode Decision Based on RDO Cost Prediction. *IEEE Trans. Broadcast.* **2019**, *65*, 109–122. [[CrossRef](#)]

41. Ogata, J.; Ichige, K. Fast Intra Mode Decision Method Based on Outliers of DCT Coefficients and Neighboring Block Information for H.265/HEVC. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5.
42. Ryu, S.; Kang, J. Machine Learning-Based Fast Angular Prediction Mode Decision Technique in Video Coding. *IEEE Trans. Image Process.* **2018**, *27*, 5525–5538. [[CrossRef](#)]
43. Song, N.; Liu, Z.; Ji, X.; Wang, D. CNN oriented fast PU mode decision for HEVC hardwired intra encoder. In Proceedings of the 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, Canada, 14–16 November 2017; pp. 239–243.
44. Ting, H.; Fang, H.; Wang, J. Complexity Reduction on HEVC Intra Mode Decision with modified LeNet-5. In Proceedings of the 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Hsinchu, Taiwan, 18–20 March 2019; pp. 20–24.
45. Breiman, L. Random Forest. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
46. Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. Classification and Regression Trees. *Eur. J. Oper. Res.* **1985**, *19*, 144.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.