IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 30, 2020, accepted June 17, 2020, date of publication June 24, 2020, date of current version July 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004580

Fast CU Partition and Intra Mode Decision Method for H.266/VVC

QIUWEN ZHANG, (Member, IEEE), YIHAN WANG, LIXUN HUANG, AND BIN JIANG
College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

Corresponding author: Qiuwen Zhang (zhangqwen@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771432, Grant 61302118, and Grant 61702464.

ABSTRACT The Versatile Video Coding (H.266/VVC) standard has developed by Joint Video Exploration Team (JVET). Compared with the previous generation video coding standard, the H.266/VVC is more outstanding. Since the H.266/VVC introduces multi-type tree (MTT) structure including binary tree (BT) and ternary tree (TT), which brings the significant coding efficiency but increases coding complexity. Moreover, the intra prediction modes have increased from 35 to 67, which can provide more accurate prediction than H.265/High Efficiency Video Coding (HEVC). Therefore, these can improve the encoding quality, but increase computational complexity. To reduce the computational complexity, this paper designs a fast coding unit (CU) partition and intra mode decision algorithm, which includes fast CU partition based on random forest classifier (RFC) model and fast intra prediction modes optimization based on texture region features. Simulation results indicate that the proposed scheme can save 54.91% encoding time with only 0.93% increase in BDBR.

INDEX TERMS H.266/VVC, fast CU partition, intra mode decision, random forest, texture feature.

I. INTRODUCTION

With the emergence of video applications, such as 4K/8K ultra high definition (UHD), the amount of video data has exploded and the higher requirements have been placed on encoding technology [1]. The Moving Picture Experts Group (MPEG) and Video Coding Experts Group (VCEG) have established JVET to take charge of the next generation video coding standard——H.266/VVC [2]. As the beginning of 2020, the JVET has released the latest version video test model (VTM8.0) of H.266/VVC [3]. Compared with H.265/HEVC test model (HM), the VTM can maintain the subjective visual quality and enhance coding efficiency by about 40%.

The quad-tree with nested multi-type tree (QTMT) structure is introduced in H.266/VVC, which removes the differences between the concepts of coding unit (CU), prediction unit (PU) and transform unit (TU), and only adopts the concept of CU. The CU shape exists square or rectangular in QTMT structure. Firstly, the coding tree unit (CTU) is divided into quad-tree (QT) to generate QT leaf nodes. The size of the QT leaf node is from 16×16 to 128×128, the CTU size is 128×128. The minimum QT size is 16×16 and the maximum QT size is 128×128. If the leaf node size of QT is 128×128, it will not be further divided by BT and TT, because this size exceeds the root node size of maximum BT and TT (64×64). If the leaf node size of QT is not 128×128, the QT leaf node can be further divided by MTT. At this time, the QT leaf node is also the MTT root node of which the depth is 0.

When the depth of the MTT reaches the maximum allowable hierarchical depth, the further division is unconsidered. When the width of the MTT node is equal to the minimum leaf node size of BT and is less than or equal to twice the minimum leaf node size of BT, the further horizontal division is unconsidered. Similarly, when the height of the MTT node is equal to the minimum leaf node size of BT and is less than or equal to twice the minimum leaf node size of BT, the further vertical division is unconsidered. The final division of a CU needs to traverse the QT, BT and TT division at all depths. We need calculate the rate distortion (RD) cost at each depth and select the partition mode with the lowest RD. Therefore, the QTMT structure greatly increases the calculation complexity of CU partition compared with the QT partition in H.265/HEVC. Fig. 1 shows that a CTU is separated by using QTMT architecture.

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

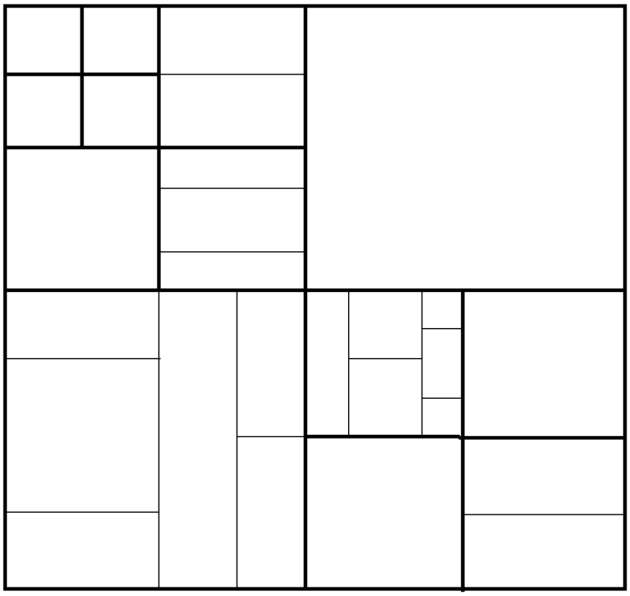

FIGURE 1. Example of QTMT coding structure.

Furthermore, the intra prediction is also one of the biggest changes, including DC, planar and 65 angle modes. Compared with H.265/HEVC which provides 35 intra prediction modes, the H.266/VVC can provide more accurate forecasts. Although these improve the encoding quality, these increase computational complexity and coding time. The H.266/VVC introduces many advanced coding tools, which greatly improves the coding efficiency of the new generation video coding standard [4], [5]. However, these new encoding tools in H.266/VVC have resulted in the significant coding complexity, thereby significantly reducing the encoding speed of the new generation video coding standard. Under the ''All intra'' configuration test condition, the coding complexity of VTM is 19 times that of HM [6]. Therefore, the H.266/VVC needs to find a compromise between the coding performance and coding complexity.

At present, many academics have presented fast methods for coding complexity of H.265/HEVC. The [7]–[14] methods can accelerate coding in H.265/HEVC based on machine learning or deep learning. A machine learning technique, the random forests, is designed in [7], where this paper proposes using the RFC model based on off-line training to skip or terminate the depth level of the current CU. A novel classifier based on CU dimension comprising an off-line trained decision tree with three hierarchical nodes is presented in [8]. This method can decrease the number of CUs to be checked by the Rough Mode Decision (RMD) and Rate Distortion Optimization (RDO) stages of intra prediction coding. A fast intra prediction modes decision scheme based on machine learning is developed in [9], which utilizes RFC model to accelerate the coding speed. To optimize the complexity allocation of CU-level with given the RD cost constraints, the machine learning methods are exploited in [10] for fast CU depth decision of H.265/HEVC. A fast CU mode decision algorithm based on the convolutional neural network (CNN) is designed in [11] to decrease the coding complexity. A fast method based on classification is presented in [12], in which the off-line and online machine learning mode of Support Vector Machine (SVM) are used for the classifier to achieve better prediction performance. A fast method for intra CU dimension decision in H.265/HEVC is introduced in [13], where each stage is based on machine learning and each CU decision layer is based on fast intra CU decision. In [14], an adaptive method for intra CU dimension decision is presented by utilizing machine learning technology based complexity classification.

The methods [15]–[18] accelerate encoding based on texture in H.265/HEVC. A fast scheme for intra CU dimension decision is presented in [15], which uses the complexity of image and an adaptive depth prediction for early split CU decision, and utilizes the Bayesian rule and quadratic discriminant analysis for early termination of the CU partitioning process. The Haar wavelet coefficients based on the texture of the CU are designed in [16] to determine whether the CU is to be continuously split. A fast intra prediction method based on texture complexity is proposed in [17], which can effectively decrease the computational complexity of H.265/HEVC. A fast intra CU partitioning scheme based on texture classification and CNN classifier is exploited [18], which decreases the calculation complexity of intra coding in H.265/HEVC by considering the heterogeneous texture features in CNN. The above fast coding methods based on H.265/HEVC are instructive for H.266/VVC. However, these methods are no longer applicable, because the MTT structure is nested in the QT structure for H.266/VVC.

In the fast H.266/VVC methods, a confidence interval based early termination method is introduced in [19] for quad-tree plus binary tree (QTBT) to identify the unnecessary partition modes of RDO. An effective partition decision method based on QTBT structure is designed in [20] to achieve a good balance between calculation complexity and RD performance, which designs a joint classifier decision tree architecture to eliminate unnecessary iterations and control the risk of mis-prediction. A fast scheme based on spatial characteristics for CU partitioning decision is introduced in [21] to decrease the complexity of QT and BT structure in H.266/VVC. Several fast intra algorithms are presented in [22] to enhance the balance between complexity and gain. A fast partition method based on Bayesian decision rules is designed in [23], which takes advantage of the correlation between the horizontal split of parent CU and sub-CU to accelerate intra coding in H.266/VVC. In [24], they determine whether to terminate the CU decomposition in advance according to the average depth information of the adjacent large coding unit (LCU). And the unnecessary RDO can be effectively eliminated by using the encoding mode of the adjacent CU to accelerate encoding in H.266/VVC. A pruning algorithm based on the PU size in advance is presented in [25] to reduce redundant MTT partitions, which aims to speed up intra coding by identifying unnecessary division directions in advance. A fast early skip intra coding method in H.266/VVC is introduced in [26], which can skip redundant MTT pruning. A fast intra partition method based on variance and gradient is developed in [27] to settle the rectangular partition issue in H.266/VVC. The basic idea of the method is mainly to use larger CUs and the sub-CUs to predict homogeneous region and complex texture region, respectively. A fast intra coding method based on H.266/VVC is exploited in [28], which integrates a low complexity CTU architecture derivation method based on statistical learning and a fast intra modes decision scheme based on gradient descent to accelerate intra coding speed.

The [29]–[33] algorithms based on deep learning or machine learning can accelerate coding in H.266/VVC. Fast CU depth decision schemes are introduced in [29], [30], which modeled the depth range as a multi-class classification problem to accelerate intra coding in H.266/VVC. An adaptive CU partition decision algorithm based on H.266/VVC is developed in [31], which uses variable CNN to optimize CU partition and avoid the calculation of full RD. A partitioning scheme of adjustable QTBT structure based on machine learning is exploited in [32], which utilizes a RFC model to determine the most likely partitioning mode for each CU. In [33], a fast CU partition decision method is presented in H.266/VVC, which is based on spatial features. A fast block partition algorithm is devised in [34] for both intra coding and inter coding to make a better trade-off between encoder complexity and coding efficiency.

In the case that the coding efficiency of H.266/VVC decrease very little, reducing coding complexity, increasing coding speed and saving coding time are important direction of current research video coding in H.266/VVC. Many coding tools are improved on the basis of H.265/HEVC and many novel technologies are also developed in H.266/VVC, which greatly improves the coding efficiency of H.266/VVC and leads to significant computational complexity. To reduce the computational complexity and encoding time, this paper proposes a fast CU partition and intra mode decision algorithm, which includes fast CU partition based on RFC model and fast intra prediction modes optimization based on texture region features. In fast CU partition method based on RFC model, the CUs are divided into smooth region, ordinary region or complex region according to the texture complexity. If the current CU belongs to smooth region, there is no need to continue split; if the current CU belongs to complex region, a well-trained RFC model is used for classification; if the current CU belongs to ordinary region, it is encoded according to the original prediction process. Moreover, the fast intra prediction modes optimization method based on the texture region characteristics calculates texture direction value of the CU and classifies the 65 prediction modes into four categories, which consist of 0°, 45°, 90° and 135°. And then we use Canny operator to calculate the gradient of each pixel in current CU. The vector of gradient is projected into the four defined directions, respectively. Further, we add the projections of the pixels of the current CU in four directions, which is defined as the energy of the current CU in four directions. The energy in the four directions is sorted from large to small, that is, {E1, E2, E3, E4}. If E2 is greater than 0.8 times E1, the direction corresponding to E1 is the main direction and the direction corresponding to E2 is the auxiliary direction, that is, the CU contains two texture directions. Otherwise, the current CU contains one texture direction, and the direction corresponding to E1 is the main direction. Therefore, the intra prediction modes corresponding to the directions are the modes which calculate the first round of the Sum of Absolute Transformed Difference (SATD) calculation. In this method, the best prediction mode is obtained and the final RD cost calculation are reduced. The proposed algorithms in this paper can decrease the calculation complexity and increase time savings.

The remaining of this paper is organized as follows: In Section 2, the proposed fast CU partition and intra mode decision method is described in detail, which includes fast CU partition method based on RFC model and fast intra prediction modes optimization based on texture region features. Section 3 illustrates the experiment results. Finally, the conclusion will be provided in Section 4.

II. PROPOSED ALGORITHM

The H.266/VVC uses QTMT structure to divide the CU and the intra prediction modes are increased to 67, which improves encoding efficiency of H.266/VVC and results in very high encoding complexity and longer encoding time. To decrease the coding complexity and increase time savings, this paper proposes fast CU partition and intra modes decision algorithm, which includes fast CU partition based on RFC model and fast intra prediction modes optimization based on texture region features. The proposed method can greatly increase time savings and decrease computational complexity.

A. FAST CU PARTITION BASED ON RANDOM FOREST CLASSIFIER

1) CLASSIFICATION OF CU BASED ON TEXTURE COMPLEXITY

The H.266/VVC has higher coding efficiency than the current video coding standards. However, the intra coding of high efficiency is accompanied by significant computational complexity. In order to decrease the computational burden of H.266/VVC, this paper proposes a fast CU partition method based on the RFC model. In the image coding process, the single area of image content tends to use larger CU for encoding. On the contrary, the region with rich details is generally encoded using smaller CU. This phenomenon can inspire us to use the degree of texture complexity of the current CU to determine whether the CU is directly divided or skipped, so CUs are classified based on texture complexity. In addition, the variance of the CU represents the degree of energy dispersion between two pixels of the CU. Therefore, the texture complexity of the CU is roughly measured by Standard Deviation (SD). The SD is expressed as,

$$SD = \sqrt{\frac{1}{W \times H}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}P(x,y)^2 - \left(\frac{1}{W \times H}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}P(x,y)\right)^2} \tag{1}$$

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

To more accurately represent the texture complexity of the CU, we utilize the Mean of Absolute Difference between Pixels (MADP) to define the difference between each pixel and its surrounding pixels, which is used to represent the complexity between pixels. According to the position of the pixel and the number of surrounding pixels, the pixels in the current CU are separated into three categories, namely, the quadrangular category, the boundary category and the ordinary category. The MADP of three types are obtained in the same way, so we only show the MADP of ordinary class, which is expressed as,

$$P_{MADP}(x,y) = \frac{1}{8}\begin{pmatrix} |P(x,y)-P(x-1,y-1)|+|P(x,y)-P(x,y-1)| \\ +|P(x,y)-P(x+1,y-1)|+|P(x,y)-P(x+1,y)| \\ +|P(x,y)-P(x+1,y+1)|+|P(x,y)-P(x,y+1)| \\ +|P(x,y)-P(x-1,y+1)|+|P(x,y)-P(x-1,y)| \end{pmatrix} \tag{2}$$

where $P(x, y)$ denotes the pixel value and (x, y) is the position of the pixel in the CU.

Then, we use the SD of the CU and the MADP of the pixel to calculate the relative complexity, which is defined as follow,

$$PC(x,y) = P_{MADP}(x,y) \times SD \tag{3}$$

where $P(x, y)$ represents the relative complexity, (x, y) is the position of the pixel in the CU.

The SD represents the rough texture complexity of CU block and cannot accurately describe the texture complexity. In order to gain more accurate texture complexity, we obtain the relative pixel complexity (PC) through the MADP and SD, where MADP represents the complexity between pixels, that is, the difference between the current pixel and its surrounding pixels. Compare with SD, texture complexity (TC) is based on the original SD formula, but the original pixel value is replaced by the PC value, which can accurately describe the texture complexity. According to [35], a similar method in H.265/HEVC is utilized to calculate PC, which has been verified to be reasonable. Therefore, after calculating the relative PC of the CU, the texture complexity of the CU is described again with SD formula, which is expressed as,

$$TC = \sqrt{\frac{1}{W \times H}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}PC(x,y)^2 - \left(\frac{1}{W \times H}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}PC(x,y)\right)^2} \tag{4}$$

where W denotes the width of the CU and H represents the height of the CU, TC represents the redefined texture complexity of the CU. $PC(x, y)$ represents the relative complexity.

We compare the texture complexity of the CU with the preset threshold to classify the CUs. Therefore, selecting threshold is a very critical step. Since the texture complexity of adjacent blocks are related to the current CU, we consider deriving the threshold based on the texture complexity of neighboring blocks. According to a large number of experiment results, it is reasonable that the largest and the smallest TC value in the neighboring blocks of the current CU are considered as T_{split} and T_{non_split}, respectively. If the TC value of the current CU is less than the threshold T_{non_split}, it indicates that the current CU belongs to smooth region, where the CUs in smooth region do not need to continue split; if the TC value of the current CU is greater than the threshold T_{split}, the current CU belongs to complex region; if the TC value of the current CU is between T_{split} and T_{non_split}, the current CU belongs to ordinary region, where the CUs in ordinary region are encoded according to the original prediction process.

2) RANDOM FOREST CLASSIFIER

The RFC model is constituted by the multiple decision trees, which solves the instability of a single decision tree. The decision trees are independent of each other, which can parallelize encoding to accelerate the training speed. Furthermore, the RFC model randomly selects equal sample sets, which can balance the errors of imbalanced data sets and avoid overfitting. And it randomly selects subsets of features with equal probability, which can process high dimensional data. The RFC model is utilized for regression and classification. In this paper, we mainly discuss the classification problem based on RFC model. The random forest is actually a classifier composed of multiple decision trees whose output category is determined by the output of each decision tree. The RFC model is essentially homogeneous integration method and is determined by two randomization methods, namely, Bagging and random subspace method (RSM).

The Bagging based on Bootstrap method belongs to a typical ensemble learning algorithm, which resamples to generate multiple training sets, where the training sample sets with same sample quantity are randomly taken from the original training sample set by repeating and putting back sample. Therefore, the new Bootstrap sample sets are obtained and the number of new training sample sets is K. The RSM which is also called Attribute Bagging or Feature Bagging is also an ensemble learning method. The main idea of RSM is that a sub-attribute set is randomly and equal probabilistically extracted from all feature attribute sets, when each non-leaf node of each decision tree is splitting. And then, an optimal attribute is selected for split nodes to decrease the correlation between each classifier, which can enhance the classification accuracy. Each decision tree in forest is established by randomly extracting samples and randomly extracting attributes. The combination of multiple decision trees is called ''Random Forest'', that is, the RFC model is a combination of the Bagging and RSM.

The RFC model based on Bootstrap resampling generates the new training sample sets, where the number of training sample sets is K and the data of each sample set grows into a decision tree. At the non-leaf nodes of each tree, the features are randomly extracted based on RSM from feature attribute set which has M feature attributes, and the number of extracted features is m (m $\ll$ M). According to the non-leaf node split algorithm, the optimal attributes are selected from m features to branch growth. Finally, these decision trees are combined for voting. After the RFC model is generated, the new sample sets are taken for testing. Each decision tree in the forest independently determines the classification result. The final decision is based on the classification category with the most judgments. The classification result is expressed as,

$$H(t) = \arg\max_{Y} \sum_{i=1}^{K} I(h_i(t) = Y) \tag{5}$$

where $H(t)$ represents combined classification model, $h_i(t)$ denotes single classification tree model, t represents the feature attributes of decision tree, Y denotes the output variable, $I(\cdot)$ denotes an illustrative function, K denotes the number of the decision trees.

During the training process of the random decision tree, the training data sets multiple times are split into two sub-data sets. The branches of each decision tree are generated and the attributes are selected according to split rule that is minimum Gini coefficient. If the value of Gini coefficient is smaller, the purity of the divided subset is the higher. The decision trees are recursively constructed. The recursive process may stop at each node in the following cases: (1) the number of samples in the current node is less than given threshold; (2) the samples in the current node belongs to the same category; (3) the attribute set of the sample in the current node is empty; (4) the depth of the decision tree is greater than the pre-set value.

3) RFC TRAINING

Although the RFC model can handle UHD data, we need select truly relevant feature vectors to generalize the classification model. To measure the effectiveness of the features for classification, we select the four directions (0°, 45°, 90° and 135°) of the Gray-level co-occurrence matrix (GLCM) as the features of the RFC model. The GLCM features in RFC model training that we introduce are just used to calculate in original pixel. The calculation of feature vectors is as follows.

The Entropy (ENT) incarnates the amount of information in image. A larger value indicates a larger amount of image information, which is more likely to be divided, and a smaller value indicates a smaller amount of image information, which is less likely to be divided. The ENT is expressed as,

$$ENT = -\sum_{x=1}^{W}\sum_{y=1}^{H} P(x, y)\log_2 P(x, y) \tag{6}$$

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

The Contrast (CON) reflects the texture depth of the image. A larger value indicates a larger texture depth, which is more likely to be divided, and a smaller value indicates a smaller texture depth, which is less likely to be divided. The CON is expressed as,

$$CON = \sum_{x=1}^{W}\sum_{y=1}^{H} (x-y)^2 P(x, y) \tag{7}$$

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

The Inverse Difference Moment (IDM) reflects the magnitude of local changes in image texture. If the different regions of the texture are more smooth and change slowly, the IDM will be larger, which is more likely to be divided. If the different regions of the texture are complex and change quickly, the IDM will be smaller, which is less likely to be divided. The IDM is expressed as,

$$IDM = \sum_{x=1}^{W}\sum_{y=1}^{H} \frac{P(x, y)}{1+(x-y)^2} \tag{8}$$

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

When traversing the CU, we record the characteristics and division results of the complex CU without disturbing the normal encoding process. The resampling by the Bagging integration method, we get the multiple training sets. We randomly select the training sample sets with same sample quantity from the original sample set, and the new training sample sets are obtained by repeating and putting back sample. Therefore, we obtain the new training sample sets and the number of the training sample sets is K.

After obtaining the training sample sets, the RFC model is trained. Table 1 illustrates the relevant training parameter settings for the RFC model. According to the parameters in Table 1, Fig. 2 illustrates the process of RFC model offline training. Firstly, we get the sample set $\{S_1^*, S_2^*, \ldots, S_K^*\}$ and the testing set. Based on the Bagging integration method, we utilize the original sample set to generate the training sample sets and the corresponding decision trees $\{T_1, T_2, \ldots, T_K\}$. Secondly, each non-leaf node of the decision tree randomly selects attributes and the number of

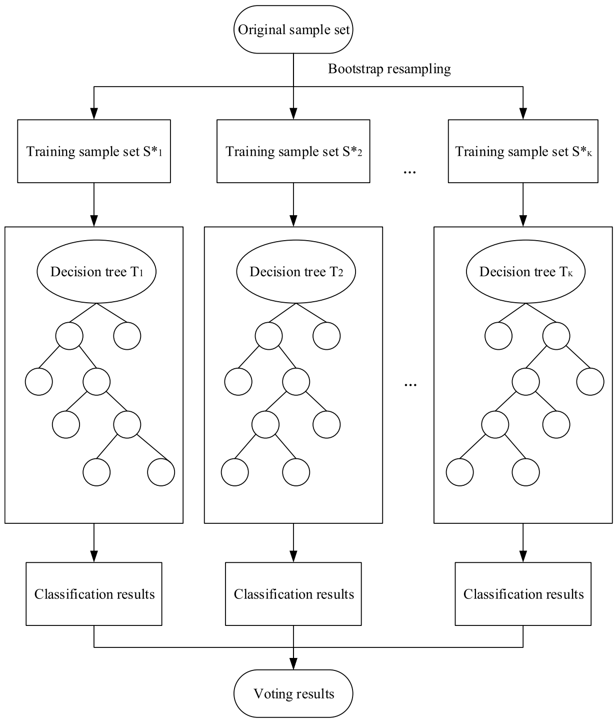

FIGURE 2. Structure of the RFC model.

attributes is m. And then, we calculate the Gini coefficient of each attribute, which selects the attribute with the smallest value of Gini coefficient as the optimal split attribute of the current node. The value of Gini coefficient is considered as split threshold, which is divided into left and right sub-tree. The RFC model is formed by each trained decision tree, which is used to discriminate the testing set T. The category with the most output in the decision tree set $\{T_1, T_2, \ldots, T_K\}$ is considered as the category of the current testing set.

TABLE 1. The training parameters of RFC model.

Parameter Name	Parameter settings
Sample set	ParkScene, RaceHorses, BQMall, Johnny
Characteristic attribute variable	ENT, CON, IDM (0°, 45°, 90°, 135°)
The number of characteristic attributes	3×4=12
The number of trees in the forest	25
Max depth	30
The number of leaf node attributes	5
The number of Classification	6

The training process of RFC model is off-line. The trained model is imported into the encoder for prediction division of CU. Therefore, the training time is not included in the encoding time of the sequences. The video sequences in Table 2 is the same as the video sequences in Table 1, where the sequences include RaceHorses, BQMall, Johnny and ParkScene. We take the former 50 frames of each video sequence, of which the former 20 frames are for RFC model training, and the last 30 frames are for RFC model testing.

TABLE 2. The prediction accuracy of RFC model.

Sequence	QP				Average
	22	27	32	37	
RaceHorses	87.53	90.91	95.23	87.92	90.15
BQMall	87.53	91.54	85.66	89.85	88.65
Johnny	87.12	86.46	91.33	85.83	87.69
ParkScene	88.23	89.91	92.22	94.89	91.31

Table 2 illustrates the average prediction accuracy of CU partition decision in each frame of each video sequence under each quantization parameter (QP). It is observed from Table 2 that the prediction accuracy of the RFC model in different sequences is between 85% and 95% with slight differences between different sequences, which is acceptable. The average of prediction accuracy is about 90%. The prediction accuracy under different QPs are also basically consistent in the same sequences, which indicates the RFC model with better consistency under different sequences and different QPs. Therefore, we conclude that the RFC model is suitable for the division and determination of the CU.

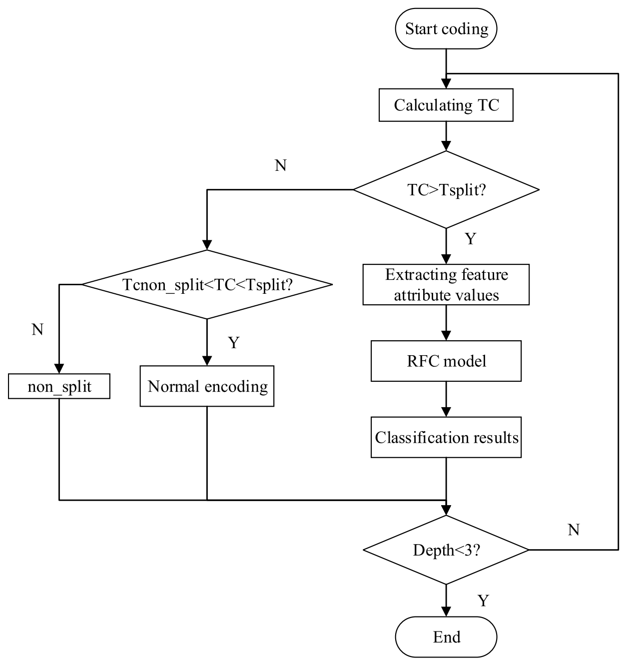

FIGURE 3. Flowchart of the proposed method based on RFC model.

The RFC model considers the CU partition as a classification issue, which is utilized to predict the CU partition result. We attain the results of the CU partition by means of based on RFC model, where the classification results include QT division, horizontal BT division, vertical BT division, horizontal TT division, vertical TT division and no division. The flowchart of the proposed fast CU partition method is shown in Fig. 3. In the CU encoding of each frame, we calculate the TC, if TC is large than T_{split}, the CU belongs to complex region, we calculate the image features of CU and determine the division results according to the trained RFC model, which can reduce the complexity of CU division while maintaining encoding performance. When the texture of the CU is very complex, if the normal partition is used to find the best partition mode, the coding complexity is high and the coding performance is reduced very little; if the RFC training model is used, the coding complexity of the recursive CU partitioning is reduced and the coding loss can be ignored. Therefore, we use the trained RFC model to decide the CU partition mode in advance which can reduce the complexity. if TC is less than T_{split} and is larger than T_{non_split}, the CU belongs to ordinary region and performs normal encoding process, otherwise, TC is less than T_{non_split} and the CU is no split. And then, we compare the depth of the CU with 3, if the depth is less 3, we compute the TC of sub-CU, and the sub-CUs continue to utilize the RFC model for predictive classification.

B. OPTIMIZATION OF THE INTRA PREDICTION MODES BASED ON TEXTURE REGION FEATURES

The number of the intra prediction modes is 67. The RDO for selecting the optimal intra modes in H.266/VVC results in the high calculation complexity. We count the time taken by each part of the coding in the frame. According to the statistical data, it concludes that the coding time generated by the RDO process exceeds the half of the total coding time. Therefore, we decrease the number of intra prediction modes to reduce complexity and coding time in the proposed intra prediction modes optimization method.

It is found that the optimal prediction mode of CU has a high correlation with the pixel similarity in the corresponding direction. In other words, the content difference in the direction of the optimal mode is the smallest in the most prediction modes. Based on this principle, the texture direction of CU is used to eliminate some prediction modes that are unlikely to be the most mode, thereby simplifying the intra prediction process. Thus, the intra prediction modes have strong correlation with texture features in H.266/VVC. The texture feature represents the nature of the corresponding thing in the image, and the texture direction is an important texture feature. In most cases, the prediction mode of CU is almost consistent with the texture direction. For regions with flat texture, horizontal or vertical mode is often used for prediction coding. For areas with rich edges and details, more accurate angle modes are used for prediction coding. In order to clarify the inherent rules of intra prediction mode selection, several video sequences with different categories and texture features are selected for statistical analysis. The rules of intra prediction modes selection in H.266/VVC are obtained by experiments. For the video sequences with uniform and flat features, the probability of selecting Planar, DC and vertical mode is larger than other prediction modes. For example, the test sequence ParkScene with general degree of motion, uniform and flat feature has a higher probability of choosing Planner, DC and vertical mode than other prediction modes, which are 28.7%, 25.8% and 7.6%. For the video sequences with complex foreground and monotonous background features, the video sequences with complex texture and non-uniform distribution features and the video sequences with non-uniform distribution and more vigorous motion features, through the analysis of texture features and data, we found that the probability of selecting Planar, DC, horizontal and vertical mode, such as the FourPeople, BQMall and BasketballPass video sequences, is higher compared with other modes. Therefore, we classify the prediction mode set according to the direction value of the texture in CU and divide 35 intra prediction modes into four categories close to the above direction, including 0°, 45°, 90° and 135°. Moreover, combined with the frequency of mode = (0, 1, 10 and 26) that is statistically calculated, we add Planar and DC mode to each category based on experimental statistics to enhance the accuracy of the method. Fig. 4 shows the classification of prediction modes.

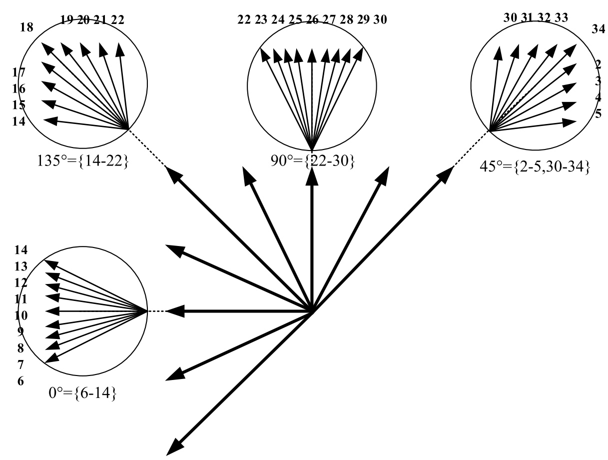

FIGURE 4. The classification of prediction modes.

TABLE 3. The categorization of intra prediction modes.

Texture direction	Mode set
0°	0, 1, 6-14
45°	0, 1, 2-5, 30-34
90°	0, 1, 22-30
135°	0, 1, 14-22

Table 3 illustrates the classification of intra prediction modes. It can be seen from Table 3 that the texture direction is correspond to mode set. Specifically, the texture direction 0° is correspond to mode "0, 1, 6-14", the texture direction 45° is correspond to mode "0, 1, 2-5, 30-34", the texture direction 90° is correspond to mode "0, 1, 22-30", the texture direction 135° is correspond to mode "0, 1, 14-22". This paper proposes a fast intra modes prediction method by utilizing Canny operator to compute the gradient of each pixel. The calculation of Canny operator is simple, so the gradient can be calculated quickly. The horizontal and vertical gradient components are expressed as,

$$G_x(x,y) = P(x+1,y) - P(x,y) + P(x+1,y+1) - P(x,y+1) \tag{9}$$

$$G_y(x,y) = P(x,y) - P(x,y+1) + P(x+1,y) - P(x+1,y+1) \tag{10}$$

where $G_x(x,y)$ and $G_y(x,y)$ denote the gradient components of the current pixel in the horizontal and vertical, respectively. $P(x,y)$ represents the pixel value of the CU at (x,y) position.

To reduce the amount of calculation, we utilize the absolute value operation instead of the square operation to roughly calculate the amplitude. The amplitude and angle are expressed as,

$$Amp_{x,y} = |G_x(x,y)| + |G_y(x,y)| \tag{11}$$

$$\theta_{x,y} = \arctan(\frac{G_y(x,y)}{G_x(x,y)}) \tag{12}$$

where $Amp_{x,y}$ denotes the gradient amplitude of the Canny operator, $\theta_{x,y}$ represents the gradient angle of the Canny operator.

The texture directions of the CU are obtained, which calculates based on each pixel. In this method, we compute the gradient vector of each pixel, which project each pixel vector to 0°, 45°, 90° and 135°, respectively. Therefore, we can get the component of each pixel vector in each direction. The projection is expressed as,

$$d = |\vec{a}|\cos\beta = \frac{\vec{a}\cdot\vec{b}}{|\vec{b}|}$$
$$\vec{a}\cdot\vec{b} = x_a x_b + y_a y_b \tag{13}$$

where d represents the $\vec{a}$ projected to $\vec{b}$, $|\vec{a}|$ denotes the modulus of $\vec{a}$ and $|\vec{b}|$ denotes the modulus of $\vec{b}$. (x_a, y_a) and (x_b, y_b) represent the coordinates of $\vec{a}$ and $\vec{b}$, respectively. β denotes the angle between $\vec{a}$ and $\vec{b}$.

The texture direction of each pixel is perpendicular to the gradient direction. The vector of gradient is $(Amp_{x,y}\cdot\cos\theta_{x,y}, Amp_{x,y}\cdot\sin\theta_{x,y})$. The unit vectors of 0°, 45°, 90° and 135° are $(1,0)$, $(\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2})$, $(0,1)$ and $(\frac{\sqrt{2}}{2},-\frac{\sqrt{2}}{2})$, respectively, and the module of the unit vectors are 1 for 0°, 45°, 90° and 135°. The calculation of the texture direction of each pixel for 0°, 45°, 90° and 135° is expressed as,

$$P_{0^\circ_{x,y}} = Amp_{x,y}\times\sin\theta_{x,y} \tag{14}$$

$$P_{45^\circ_{x,y}} = Amp_{x,y}\times\sin\theta_{x,y}\times\frac{\sqrt{2}}{2} - Amp_{x,y}\times\cos\theta_{x,y}\times\frac{\sqrt{2}}{2} \tag{15}$$

$$P_{90^\circ_{x,y}} = Amp_{x,y}\times\cos\theta_{x,y} \tag{16}$$

$$P_{135^\circ_{x,y}} = Amp_{x,y}\times\cos\theta_{x,y}\times\frac{\sqrt{2}}{2} + Amp_{x,y}\times\sin\theta_{x,y}\times\frac{\sqrt{2}}{2} \tag{17}$$

where $P_{0^\circ_{x,y}}$, $P_{45^\circ_{x,y}}$, $P_{90^\circ_{x,y}}$ and $P_{135^\circ_{x,y}}$ denote the projection value of the each pixel in 0°, 45°, 90° and 135°, respectively.

According to the projection of the gradient of all pixels of the CU in four directions, we judge the texture direction of the CU. Therefore, we add the projections of the pixels of the CU in four directions. The energy in each direction is expressed as,

$$P_d = \sum_{x=1}^{W}\sum_{y=1}^{H}|P_{d_{x,y}}|, (d = 0^\circ, 45^\circ, 90^\circ, 135^\circ) \tag{18}$$

where W represents the width of the CU and H denotes the height of the CU. $P_{d_{x,y}}$ represents $P_{0^\circ_{x,y}}$, $P_{90^\circ_{x,y}}$, $P_{90^\circ_{x,y}}$ and $P_{135^\circ_{x,y}}$, respectively.

We sort the energy of the four directions from large to small, that is, {E1, E2, E3, E4}. The direction with the greatest energy is the main direction. Moreover, if E2 is greater than 0.8 times E1, the current CU contains two texture directions, where E2 is the auxiliary texture direction. Otherwise, the intra prediction modes corresponding to the direction with the greatest energy execute the first round of SATD calculation. Therefore, the modes which execute the first round of SATD calculation are reduced by at least half compared with the original 35 modes. Furthermore, the second round of SATD calculation is performed by computing the adjacent modes of the modes obtained in the first round of SATD calculation. If the adjacent modes belong to the mode obtained by the first SATD calculation, which no longer execute the second round of SATD calculation. Therefore, if there are two texture directions, the RD calculation mode is reduced by half, and if there is only one texture direction, the mode is reduced by three quarters. In addition, the concept of Most Probable Modes (MPMs) also exists in VTM. Since the best candidate mode has high probability to be selected from MPMs list, this paper further simplify the candidate mode set. After RMD processing, we sort the entire set of candidate modes according to SATD value from small to large. The RD cost calculation on the modes before MPMs (including MPMs) are performed, and the modes after MPMs are eliminated, excluding Planar, DC, horizontal and vertical mode. Thus, the best prediction mode is obtained and the final RD cost calculation are reduced, which can reduce about 30% encoding time in the experimental results.

III. EXPERIMENTAL RESULTS

To assess the proposed overall method, the simulation tests are performed on the H.266/VVC. The test platform is Inter Core i5-6500CPU 3.2 GHz. The configuration file is "All intra". The compression property of the proposed overall method is assessed by employing the Bjontegaard Delta Bitrate (BDBR) [36]. The coding complexity is measured by reducing the coding runtime "ΔT", which is defined as,

$$TS = \frac{Time_{anchor} - Time_{proposed}}{Time_{anchor}} \tag{19}$$

where T_{anchor} represents the coding time of the anchor method (VTM4.0). The anchor method mentioned in this

TABLE 4. Results of the proposed individual method.

Test sequence BDBR(%) TS(%)		FPAC BDBR	FPAC TS	FMOR BDBR	FMOR TS
Class A1	Tango2	0.55	42.35	0.56	32.12
	FoodMarket4	0.49	39.96	0.61	30.88
	Campfire*	0.59	35.36	0.53	27.46
Class A2	Catrobot1	0.68	40.87	0.55	31.39
	DaylightRoad 2	0.71	36.11	0.57	33.21
	ParkRunning 3*	0.64	41.97	0.59	29.17
Class B	Kimono	0.69	43.53	0.69	27.85
	ParkScene	0.65	40.23	0.62	27.68
	BQTerrace	0.68	39.41	0.57	26.31
Class C	PartyScene	0.79	38.56	0.65	31.71
	RaceHorsesC	0.81	39.79	0.49	32.15
	BasketballDri ll	0.91	38.52	0.45	28.34
Class D	BlowingBubb les	0.65	39.42	0.49	29.31
	RaceHorses	0.65	41.85	0.54	30.41
	BQSquare	0.72	38.13	0.45	33.62
Class E	Johnny	0.76	36.57	0.47	30.12
	FourPeople	0.69	38.63	0.55	29.17
	KristenAndSa ra	0.73	31.58	1.12	31.45
	Average	**0.69**	**39.05**	**0.58**	**30.13**

TABLE 5. Results of the proposed overall scheme.

Test sequence BDBR(%) TS(%)		Proposed algorithm BDBR	Proposed algorithm TS
Class A1	Tango2	0.97	56.95
	FoodMarket4	0.84	52.53
	Campfire*	0.78	48.77
Class A2	Catrobot1	1.08	54.55
	DaylightRoad2	0.78	53.63
	ParkRunning3*	0.81	52.67
Class B	Kimono	1.03	55.52
	ParkScene	1.33	57.78
	BQTerrace	1.07	55.21
Class C	PartyScene	0.96	56.65
	RaceHorsesC	0.92	56.71
	BasketballDrill	0.97	59.65
Class D	BlowingBubbles	0.69	58.61
	RaceHorses	0.76	54.67
	BQSquare	0.73	56.12
Class E	Johnny	0.85	51.46
	FourPeople	0.85	54.34
	KristenAndSara	1.32	52.62
	Average	**0.93**	**54.91**

paper refers to VTM4.0. $T_{proposed}$ denotes the coding time of the proposed method.

Table 4 shows the encoding performance of the proposed individual algorithm compared with the anchor algorithm. The proposed overall method consists of fast CU partition method based on a RFC model (FPAC) and fast intra prediction modes optimization method based on texture region features (FMOR). It can be observed from Table 4 that the average time savings is 39.05%, and the BDBR is increased by 0.69% in FPAC method. The simulation results demonstrate that the FPAC scheme can quickly divide the current CU. It can be observed from Table 4 that the average time savings is 30.13%, and the BDBR is increased by 0.58% in FMOR method. Therefore, the proposed FMOR method can effectively save coding time, and the loss of RD performance is negligible.

Table 5 shows the coding performance of the proposed overall scheme, which incorporates FPAC and FMOR method. In this paper, the FPAC method avoids unnecessary RDO calculations and the FMOR scheme reduces the calculation of many unnecessary prediction modes, which can reduce 54.91% coding time compared with the anchor method. At the same time, the algorithm increases the BDBR by 0.93%, which is acceptable. Thus, the proposed overall algorithm can effectively increase time savings and have a good RD performance.

Fig.5 shows the RD performance of the proposed overall method compared with VTM4.0 in the test videos. It can be observed from Fig. 5 that this proposed overall scheme can realize the consistent property in terms of RD performance compared with VTM.

Table 6 shows the coding property of the proposed overall algorithm compared with the latest fast methods. To assess the coding property of the proposed overall scheme, we compare the proposed overall scheme with the latest fast methods of H.266/VVC, including CTTD [26], FPIC [27], CPFD [28] and FCPD [34] method. These are fast and efficient method for H.266/VVC encoder. The proposed method and CTTD, FPIC, CPFD and FCPD method are compared with the same anchor method (VTM4.0). Table 6 shows the proposed overall method can save 55.78% of coding time with only increasing 0.96% BDBR compared with the anchor method. We can see from Table 6 that CTTD, FPIC, CPFD and FCPD method have good RD performance, but the coding time savings of the state-of-the-art methods are less than the proposed algorithm. Among the five methods, the BDBR of the FCPD and the proposed method are the smallest, while the BDBR of the CPFD method is the largest. The proposed algorithm saves more coding time, and the FCPD method saves the least coding time. It is observed from Table 6 that the average BDBR of the CTTD, FPIC, CPFD and FCPD method increase by 1.84%, 1.39%, 2.81% and 0.70%, respectively. Compared with the proposed overall algorithm, the average BDBR of CTTD, FPIC and CPFD method increase by 0.88%, 0.43% and 1.85%, respectively. Furthermore, the time savings of the proposed overall method is better than the CTTD, FPIC, CPFD and FCPD method. It can be seen from Table 6 that the time savings of CTTD, FPIC, CPFD and FCPD method increase by 17.43%, 3.62%, 18.11% and 21.02% compared with the proposed overall method. Though the average BDBR of the proposed method with negligible bit loss is larger than FCPD method, but the time savings of the proposed method outperforms the FCPD method.

TABLE 6. The coding performance of the proposed overall method compared with anchor method and previous works.

Test sequence		Proposed (VTM4.0)		CTTD [26] (VTM4.0)		FPIC [27] (VTM4.0)		CPFD [28] (VTM4.0)		FCPD [34] (VTM4.0)	
BDBR(%) TS(%)		*BDBR*	*TS*	*BDBR*	*TS*	*BDBR*	*TS*	*BDBR*	*TS*	*BDBR*	*TS*
Class B 1920×1080	Kimono	1.03	55.52	1.27	38.35	1.72	66.59	2.22	35.05	0.52	35.05
	ParkScene	1.33	57.78	1.41	39.23	1.28.	56.28	2.33	33.29	0.63	43.29
	BQTerrace	1.07	55.21	1.36	41.11	1.16	49.44	2.61	38.11	0.81	35.29
Class C 832×480	PartyScene	0.96	56.65	1.48	40.23	0.28	41.71	2.18	37.22	0.34	33.18
	RaceHorsesC	0.92	56.71	1.81	39.02	0.84	52.07	2.49	36.13	0.65	30.53
	BasketballDrill	0.97	59.65	1.73	39.04	1.91	53.05	2.31	39.62	1.30	29.62
Class D 416×240	BlowingBubbles	0.69	58.61	1.67	40.33	0.49	43.90	2.15	32.14	0.23	27.71
	RaceHorses	0.76	54.67	2.08	40.05	0.54	44.93	2.46	33.24	0.30	26.21
	BQSquare	0.73	56.12	1.33	38.17	0.17	32.34	2.18	38.23	0.22	26.02
Class E 1280×720	Johnny	0.85	51.46	3.19	34.85	3.07	62.55	4.46	40.92	1.29	40.72
	FourPeople	0.85	54.34	2.35	35.89	2.55	62.18	3.98	40.09	1.18	42.29
	KristenAndSara	1.32	52.62	2.39	33.95	2.56	60.82	4.32	47.95	0.94	47.20
	Average	**0.96**	**55.78**	**1.84**	**38.35**	**1.39**	**52.16**	**2.81**	**37.67**	**0.70**	**34.76**

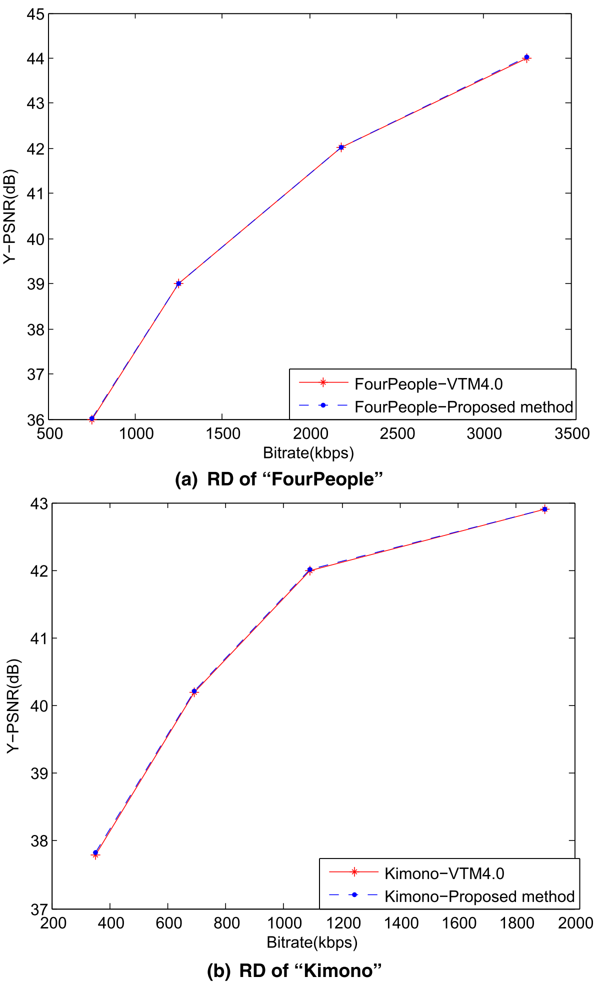

FIGURE 5. The RD performance of the proposed overall method.

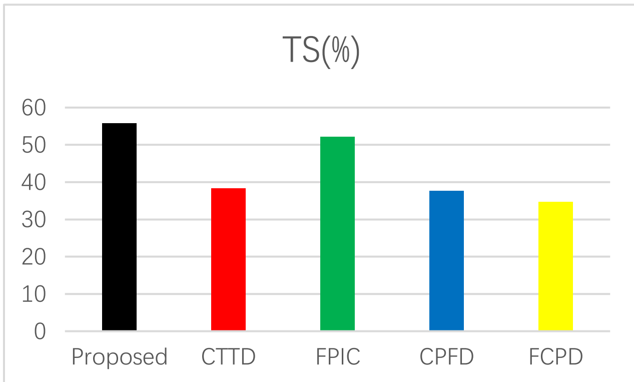

FIGURE 6. Time saving.

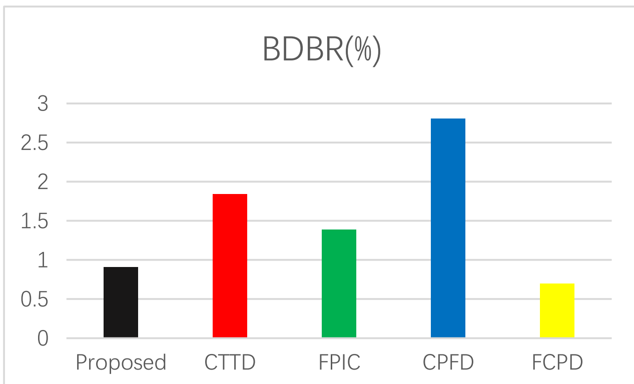

FIGURE 7. BDBR increase.

The proposed overall method includes the FPAC and FMOR method, where the FPAC method can avoid unnecessary RDO calculations and the FMOR scheme can reduce the calculation of many unnecessary prediction modes. Therefore, the proposed overall method can effectively save more coding time and has better compression performance compared with the latest fast algorithms.

Figs 6 and 7 demonstrate the coding property of the proposed overall method compared with the latest fast methods of H.266/VVC. It is observed that the proposed overall scheme can reduce the amount of calculation complexity and has good RD performance. Compared to CTTD, FPIC, CPFD and FCPD method, the proposed overall method saves about 3.62-21.02% of coding time. In addition, the proposed overall scheme reduces BDBR by 0.26-1.85% compared with the CTTD, FPIC and CPFD method. Experiment results demonstrate that the proposed overall scheme is effective for various videos, the proposed overall method is superior to the latest fast schemes in performance evaluation.

IV. CONCLUSION

In this paper, we propose a fast CU partition and intra mode decision method. This scheme includes two strategies, i.e. fast CU partition based on RFC model and fast intra prediction modes optimization based on texture region features. The main idea of this scheme is to achieve fast CU partition based on the RFC model and analyze the energy of the current CU in four directions based on the texture region characteristics to skip unnecessary intra prediction modes. Experimental results demonstrate that the proposed method can save 54.91% coding time, while the BDBR increment is only 0.93% compared with the anchor method. Moreover, the proposed scheme is better than well-known fast methods and has better performance to reduce complexity.

REFERENCES

[1] CR Formatted B. Bross, K. Andersson, M. Blaser, V. Drugeon, S.-H. Kim, J. Lainema, J. Li, S. Liu, J.-R. Ohm, G. J. Sullivan, and R. Yu, "General video coding technology in responses to the joint call for proposals on video compression with capability beyond HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1226–1240, May 2020, doi: 10.1109/TCSVT.2019.2949619.

[2] J. Chen, M. Karczewicz, Y.-W. Huang, K. Choi, J.-R. Ohm, and G. J. Sullivan, "The joint exploration model (JEM) for video compression with capability beyond HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1208–1225, May 2020, doi: 10.1109/TCSVT.2019.2945830.

[3] F. Bossen, X. Li, and K. Sühring, *AHG Report: Test Model Software Development (AHG3)*, document JVET-Q0003, Joint Video Experts Team (JVET) of ITU-T and ISO/IEC, Jan. 2020.

[4] A. Wieckowski, T. Hinz, V. George, J. Brandenburg, J. Ma, B. Bross, H. Schwarz, D. Marpe, and T. Wiegand, *NextSoftware: An alternative implementation of the Joint Exploration Model (JEM)*, document JVET-H0084, Joint Video Experts Team (JVET) of ITU-T and ISO/IEC, Oct. 2017.

[5] Z. Wang, X. Meng, C. Jia, J. Cui, S. Wang, S. Wang, and S. Ma, *Description of SDR Video Coding Technology Proposal by DJI and Peking University*, document JVET-J0011, Apr. 2018.

[6] V. Baroncini, J.-R. Ohm, and G. J. Sullivan, *Report of Results From the Call for Proposals on Video Compression With Capability Beyond HEVC*, document JVET-J003, Joint Video Experts Team (JVET) of ITU-T and ISO/IEC, Apr. 2018.

[7] B. Du, W.-C. Siu, and X. Yang, "Fast CU partition strategy for HEVC intra-frame coding using learning approach via random forests," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Hong Kong, Dec. 2015, pp. 1085–1090.

[8] D. Ruiz, G. Fernández-Escribano, V. Adzic, H. Kalva, J. L. Martínez, and P. Cuenca, "Fast CU partitioning algorithm for HEVC intra coding using data mining," *Multimedia Tools Appl.*, vol. 76, no. 1, pp. 861–894, Jan. 2017.

[9] S. Ryu and J. Kang, "Machine learning-based fast angular prediction mode decision technique in video coding," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5525–5538, Nov. 2018.

[10] Y. Zhang, S. Kwong, X. Wang, H. Yuan, Z. Pan, and L. Xu, "Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding," *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2225–2238, Jul. 2015.

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

[12] L. Zhu, Y. Zhang, Z. Pan, R. Wang, S. Kwong, and Z. Peng, "Binary and multi-class learning based low complexity optimization for HEVC encoding," *IEEE Trans. Broadcasting*, vol. 63, no. 3, pp. 547–561, Sep. 2017.

[13] Y. Zhang, Z. Pan, N. Li, X. Wang, G. Jiang, and S. Kwong, "Effective data driven coding unit size decision approaches for HEVC INTRA coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3208–3222, Nov. 2018.

[14] X. Liu, Y. Li, D. Liu, P. Wang, and L. T. Yang, "An adaptive CU size decision algorithm for HEVC intra prediction based on complexity classification using machine learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 144–155, Jan. 2019.

[15] D. Lee and J. Jeong, "Fast intra coding unit decision for high efficiency video coding based on statistical information," *Signal Process., Image Commun.*, vol. 55, pp. 121–129, Jul. 2017.

[16] X. Sun, X. Chen, Y. Xu, Y. Wang, and D. Yu, "Fast CU partition strategy for HEVC based on Haar wavelet," *IET Image Process.*, vol. 11, no. 9, pp. 717–723, Sep. 2017.

[17] Q. Zhang, Y. Zhao, W. Zhang, L. Sun, and R. Su, "Efficient intra mode decision for low complexity HEVC screen content compression," *PLoS ONE*, vol. 14, no. 12, pp. 1–16, Dec. 2019.

[18] Y. Zhang, G. Wang, R. Tian, M. Xu, and C. C. J. Kuo, "Texture-classification accelerated CNN scheme for fast intra CU partition in HEVC," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, Utah, USA, Mar. 2019, pp. 241–249.

[19] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, "Probabilistic decision based block partitioning for future video coding," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1475–1486, Mar. 2018.

[20] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, "Effective quadtree plus binary tree block partition decision for future video coding," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Apr. 2017, pp. 23–32.

[21] T. Lin, H. Jiang, J. Huang, and P. Chang, "Fast binary tree partition decision in H.266/FVC intra Coding," in *Proc. IEEE Int. Conf. Consum. Electron.-Taiwan (ICCE-TW)*, Taichung, Taiwan, May 2018, pp. 1–2.

[22] S. De-Luxan-Hernandez, H. Schwarz, D. Marpe, and T. Wiegand, "Fast line-based intra prediction for video coding," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Taichung, China, Dec. 2018, pp. 135–138.

[23] T. Fu, H. Zhang, F. Mu, and H. Chen, "Fast CU partitioning algorithm for H.266/VVC intra-frame coding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Shanghai, China, Jul. 2019, pp. 55–60.

[24] J.-J. Chen, Y.-Y. Chiu, C.-H. Lee, and Y.-H. Tsai, "Utilize neighboring LCU depth information to speedup FVC/H.266 intra coding," in *Proc. Int. Conf. Syst. Sci. Eng. (ICSSE)*, Dong Hoi, Vietnam, Jul. 2019, pp. 308–312.

[25] M. Lei, F. Luo, X. Zhang, S. Wang, and S. Ma, "Look-ahead prediction based coding unit size pruning for VVC intra coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 4120–4124.

[26] S.-H. Park and J.-W. Kang, "Context-based ternary tree decision method in versatile video coding for fast intra coding," *IEEE Access*, vol. 7, pp. 172597–172605, 2019.

[27] J. Chen, H. Sun, J. Katto, X. Zeng, and Y. Fan, "Fast QTMT partition decision algorithm in VVC intra coding based on variance and gradient," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Sydney, NSW, Australia, Dec. 2019, pp. 1–4.

[28] H. Yang, L. Shen, X. Dong, Q. Ding, P. An, and G. Jiang, "Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1668–1682, Jun. 2020, doi: 10.1109/TCSVT.2019.2904198.

[29] Z. Jin, P. An, L. Shen, and C. Yang, "CNN oriented fast QTBT partition algorithm for JVET intra coding," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Petersburg, FL, USA, Dec. 2017, pp. 1–4.

[30] Z. Jin, P. An, C. Yang, and L. Shen, "Fast QTBT partition algorithm for intra frame coding through convolutional neural network," *IEEE Access*, vol. 6, pp. 54660–54673, 2018.

[31] G. Tang, M. Jing, X. Zeng, and Y. Fan, "Adaptive CU split decision with pooling-variable CNN for VVC intra encoding," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2019, pp. 1–4.

[32] T. Amestoy, A. Mercat, W. Hamidouche, D. Menard, and C. Bergeron, "Tunable VVC frame partitioning based on lightweight machine learning," *IEEE Trans. Image Process.*, vol. 29, no. 5, pp. 1313–1328, Jan. 2020.

[33] T. L. Lin, H. Y. Jiang, J. Y. Huang, and P. C. Chang, "Fast intra coding unit partition decision in H. 266/FVC based on spatial features," *J. Real-Time Image Process.*, pp. 1–18, 2018.

[34] N. Tang, J. Cao, F. Liang, J. Wang, H. Liu, X. Wang, and X. Du, "Fast CTU partition decision algorithm for VVC intra and inter coding," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Nov. 2019, pp. 361–364.

[35] X. Liu, Y. Liu, P. Wang, C.-F. Lai, and H.-C. Chao, "An adaptive mode decision algorithm based on video texture characteristics for HEVC intra prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 8, pp. 1737–1748, Aug. 2017, doi: 10.1109/TCSVT.2016.2556278.

[36] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD Curves*, document ITU-T SG16 Q.6, VCEG Meeting, VCEG-M33, Apr. 2001.

QIUWEN ZHANG (Member, IEEE) received the Ph.D. degree in communication and information systems from Shanghai University, Shanghai, China, in 2012. Since 2012, he has been with the Faculty of the College of Computer and Communication Engineering, Zhengzhou University of Light Industry, where he is currently an Associate Professor. He has published over 30 technical articles in the fields of pattern recognition and image processing. His major research interests include 3D signal processing, machine learning, pattern recognition, video codec optimization, and multimedia communication.

YIHAN WANG received the B.S. degree in information engineering from the Zhengzhou University of Light Industry, Zhengzhou, China, in 2018, where she is currently pursuing the master's degree in information and communication systems with the School of Computer and Communication Engineering. Her current research interests include image processing, 3D video coding, and the extensions of the high-efficiency video coding.

LIXUN HUANG received the B.S. degree in electronic information science and technology and the M.S. degree in computer application technology from the Henan University of Technology, China, in 2006 and 2009, respectively, and the Ph.D. degree in communication and information systems from Shanghai University, China, in 2013. He has been a Lecturer with the Zhengzhou University of Light Industry, since July 2013. His research interests include iterative learning control and pattern recognition.

BIN JIANG received the M.S. degree in computer application technology from Henan University, Kaifeng, China, in 2009, and the Ph.D. degree in electronic circuit and system from the Beijing University of Technology, Beijing, China, in 2014. He is currently a Lecturer with the Zhengzhou University of Light Industry, China. His current research interests include 2D and 3D video processing, image processing, and multimedia systems.