Fast CU Partition Decision Method Based on Texture Characteristics for H.266/VVC

QIUWEN ZHANG\textsuperscript{1}, (Member, IEEE), YONGBO ZHAO\textsuperscript{1}, BIN JIANG\textsuperscript{1}, LIXUN HUANG\textsuperscript{1}, AND TAO WEI\textsuperscript{2}

\textsuperscript{1}College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China
\textsuperscript{2}Computer College, Henan Institute of Engineering, Zhengzhou 450001, China

Corresponding authors: Qiuwen Zhang (zhangqwen@126.com) and Tao Wei (hnsqwt@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771432, Grant 61302118, and Grant 61702464; and in part by the Scientific and Technological Project of Henan Province under Grant 202102210179.

ABSTRACT The versatile video coding (VVC) is the latest video coding standard, which uses a Multi-Type Tree (MTT) coding structure. Compared with existing video coding standards, this structure can flexibly split coding blocks according to the complex texture features of the image. As MTT structure introduces binary tree (BT) and ternary tree (TT) splitting, it will lead to a sharp increase in computational complexity. In the paper, a fast Coding Unit (CU) size decision method for intra prediction of VVC is proposed, which can significantly reduce the calculation in the intra encoding of VVC. The proposed method consists of two steps: 1) determine whether CU is divided and 2) select the best CU splitting mode. In the intra prediction process, the CU texture complexity is firstly calculated, which judges whether the CU is divided into sub-CUs. Then, the unnecessary splitting mode candidates are discarded according to the relationship between the texture direction and the CU splitting mode. The experimental results show that our proposed fast CU partition method reduces about 48.58\% the computational complexity, while the BDBR only increases by 0.91\%.

INDEX TERMS VVC, MTT, intra prediction, fast CU splitting, texture characteristics.

I. INTRODUCTION

In recent years, videos such as 360 degrees and high dynamic range (HDR) have shown huge demand potential. The currently widely used High efficiency video coding (HEVC) is a video coding standard formulated by the ITU-T VCEG in 2013 [1]. However, it is difficult for HEVC to cope with the need for higher compression rates for videos such as 360 degrees and high dynamic range (HDR). In order to solve this problem, the Joint Video Exploration Group (JVET) defined the first draft of the VVC and VVC Test Model 1.0 (VTM 1.0) coding methods in 2018 [2]. A series of encoding tools with higher compression efficiency were gradually added in the subsequent meetings. For example, the MTT splitting structure is used in the coding tree unit (CTU) partitioning process [3]. In HEVC, a CU is divided into four sub-CUs. MTT splitting structure has no restriction that only uses the quad-tree (QT) splitting structure like HEVC, so that VVC can use QT, BT and TT structure in coding process. This allows a CU to be divided into two, three or four CUs in VVC, and the encoder can split CU more flexibly to reduce the video bitrate. This structure has a great effect on improving the coding rate. In addition, for I slices, the coding tree structure of luma and chroma CU is different [4]. To improve the prediction accuracy of the current CU, the number of directional intra modes of VVC has been increased to 65, which far exceeds the 33 of HEVC. In the intra prediction process, all sizes of luma and chroma CU can use these 65 modes [5].

The coding efficiency of VVC is higher than HEVC by about 50\% because these new coding tools are introduced. However, these encoding tools also bring huge complexity for the encoding process. Especially the flexibility of MTT structure results in high coding computational complexity [6]. Unlike HEVC, VVC needs to predict the CU depth and select the CU splitting modes during CU partition process, where the CU splitting modes include vertical binary split, horizontal binary split, vertical ternary split, or horizontal ternary split. Therefore, the computational complexity brought by the MTT structure limits the use of VVC. In the past few years, many methods based on the QT splitting structure...
have emerged to accelerate intra prediction coding [7]–[11]. But these strategies rarely reduce the complexity of VVC. Therefore, VVC coding standards still have great potential to speed up encoding. In this paper, a fast encoding method is proposed including two processes, which can effectively reduce the complexity of intra prediction. The first process is to predict whether the CU is split based on the texture complexity energy; the second process is to use the texture direction information to determine the use of the horizontal division mode, the vertical division mode and the QT division mode. These effective methods are used to reduce the complexity of intra prediction for VVC at the first time and accelerate the complex process of intra prediction based on the MTT architecture. In order to verify the effectiveness of the proposed method, the time of intra prediction process on VTM 7.0 [12] is obtained under the condition of intra prediction configuration.

This paper has four parts besides the above. Section II presents the intra CU prediction method and CU splitting mode. Section III first analyzes the correlation between the CU texture complexity and the CU partition decision, then analyzes the correlation between the CU texture direction and the CU splitting mode. According to these correlations, the encoding process can be effectively increased. Finally, a fast CU partition decision method based on texture complexity and texture direction is proposed. Section IV shows the experimental results and analysis. The conclusion of the paper is presented in Section V.

### II. RELATED WORK

The H.266/VVC uses a hybrid video coding structure, where the coding block splitting structure is one of the most important parts of the coding process. In order to obtain higher video compression performance and more flexible coding, VVC introduces the MTT structure [13]. During VVC encoding, a video frame is divided into many CTUs, which is the same as HEVC. For a three-channel image frame, a CTU contains N×N luma blocks and two chroma blocks in the same area. In HEVC, a CTU is divided into sub-CUs through QT structure. Then, each sub-CU is further divided into one, two or four prediction units (PUs). After predicting PU and obtaining residual information, sub-CUs can be further divided into several transform units (TUs). It is the whole procedure of QT division in HEVC. However, there is no PU and TU in VVC, the concept of CU is uniformly adopted. In addition, CU division modes include horizontal splitting, vertical splitting and QT splitting, in which splitting structure is called a quad-tree structure with nested multi-type tree (QTMT). Fig. 1 shows the result of CU division in a CTU. The CUs are first split into sub-CUs through QT structure and each sub-CU can continue to divide smaller CUs according to the CU partition rule until reaching the predefined maximum depth of each splitting structure [14]. The QTMT partitioning structure can produce a better coding shape than HEVC, which can improve the compression rate.

| a | b | e |
|---|---|---|
| c | d | f |
| g | i | k |
| h | j | l |

(a) A split result of an MTT structure.

![MTT coding blocks splitting structure](image)

(b) The split process of MTT structure.

We know that an image is composed of many pixels, and a frame of 8-bits YUV video incorporates Y, U and V component. Each component has 256 levels. For areas with smooth textures, only a few bits can be used to describe them well; for areas with complex textures, more bits can be used to describe them. As shown in the Fig. 2, the texture of the area A is the smooth background region. The area B is the clothes fold area, and the texture is very complicated. So texture information of the area A is much less than that of area B.

Two stages are used to reduce the amount of calculation required by the encoder in VTM. Specifically, VVC has 67 candidate prediction modes for internal RMD processing. Each candidate prediction mode is calculated during the encoding process, and the 6 best candidate prediction modes are put into the merge table. In the rough mode decision (RMD) phase, the huge amount of calculation consumed is called JRMD [15]. In the second fine mode decision (FMD) phase, the best prediction mode is selected through the full
rate-distortion (RD) cost in the candidate mode list. $J_{RMD}$ and SATD are defined as follows.

$$J_{RMD} = SATD + \lambda_{pred}R_{pred}$$

$$SATD = \sum_{M} \sum_{M} |HXH|$$

where SATD is absolute sum of Hadamard transformed coefficients of residual signal, $\lambda_{pred}$ is the Lagrangian multiplier. Rate $R_{pred}$ represent the number of bits to encode the prediction mode information excluding residuals.

It is an important research direction to accelerate the depth decision of CU by studying the texture characteristics. Many researchers have proposed some effective methods, which can accelerate the intra prediction and reduce the encoding complexity. [16] and [17] both studied the relationship between SATD and CU depth, where some bad candidate patterns can be eliminated in advance according to the size of the SATD of the neighboring CU and the SATD of the current CU. To further accelerate intra coding, the CU depth information of neighboring CUs is used in [18] to speed up intra prediction, which can greatly reduce the calculation in the current CU partition process. Moreover, it is limited because it only considers the time correlation between neighboring CUs. Almost all of these algorithms perform a preliminary evaluation of RMD. Then, CU splitting mode with relatively small SATD are added to MPM. Finally, RD calculation is performed to select the optimal division mode. A fast TU size decision algorithm is proposed in [19] using Bayes decision method and the variance relationship between the residual factor and the transform size, which can reduce the number of candidates for transform size in HEVC. [20] proposed a HEVC-SCC effective intra mode decision method that effectively utilizes the texture complexity of the HE tree block. The method analyzes the texture complexity of the SC tree block according to the degree of change of the brightness value. This method may be used in the future to reduce the coding complexity of VVC. These methods above-mentioned can save a lot of coding time. However, since the CU splitting structure of VVC is significantly different from HEVC, these methods cannot be directly applied to VVC.

In addition, many fast intra prediction algorithms based on CNN have been proposed [21]–[23]. Most of these algorithms are shape-based CNN training schemes, using CNN training to accelerate the prediction of CU partitioning mode and CU depth. However, the rectangles with an aspect ratio of 1/8, 1/4, 1/2, 1, 2, 4, 8 will be generated in the encoding process of VTM. Therefore, using CNN in the VVC encoding process will significantly increase the parameters of CNN and lead to the significantly complexity of CNN. In other words, it is difficult to achieve the same efficiency as HEVC uses CNNs. Recently, many researchers have studied the characteristics of MTT structure to improve the coding complexity of VVC. A fast method based on variance and gradient is devised in [24] to speed up encoding of VVC. The algorithm first terminates the further division of the smooth area. Then, a better QT partition is further selected based on the gradient features calculated by the Sobel operator. Finally, by comparing the variances of the sub-CUs, a partition is directly selected from five possible partition modes by comparing the variances of the sub-CUs. [25] designed 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 mode optimization based on texture region features. The usage of the Intra Sub-Partitions (ISP) algorithm is introduced in [26]. The ISP subdivides the intra prediction block into 2 or 4 sub-partitions with a maximum of 16 samples, and then encodes each partition separately. This method can improve the code rate of intra prediction, but increases the calculation complexity. [27] proposed a fast CU partition decision algorithm based on an improved directed acyclic graph support vector machine (DAG-SVM) model, a fast CU partition decision algorithm is designed. This method proposes a new idea for reducing coding complexity. In [28], a novel separable transform algorithm based on the Karhunen-Loève transform (KLT) is proposed, where this method solves the disadvantages of traditional KLT. A forward-looking prediction-based CU size pruning algorithm is introduced in [29] to reduce redundant MTT partitions, which speeds up the intra prediction mode decision process by identifying unnecessary division directions in advance. Although these methods can speed up the intra prediction process of VVC, the correlation between MTT structure and texture features is not well utilized. Therefore, there is still a lot of potential to further reduce the complexity of VVC coding.

III. PROPOSED METHOD

Generally, an image usually contains some complex and smooth areas and there will be a certain transition area between the complex and smooth areas. Furthermore, the brightness value of the pixels in the transition area has a clear relationship with the distance from the texture, where the closer the pixel is to the texture, the higher the similarity is with the pixel in the texture area. When a region has high texture complexity, many pixels are located between multiple textures, and these pixels are affected by multiple
texture pixels at the same time, where these pixels between multiple textures will be very complex. For pixels between multiple textures in the intra coding process, the larger the QP value, the more difficult it is to be truly expressed. The pixels of these transition regions have very high residuals during the encoding process, which results in the RMD calculation cost of the complex regions being greater compared with the smooth regions [30]. Therefore, we can judge whether the CUs are split by calculating the texture complexity.

A. FAST CU PARTITION DECISION BASED ON TEXTURE ENERGY

In this paper, the angular second moment (ASM) is used to describe the texture complexity of the image. The gray-level co-occurrence matrix (GLCM) is a square and symmetric matrix that indicates the frequency of different combinations of grayscale in the image [31]. The size of the GLCM depends on the number of gray levels considered, not the size of the image itself. We use $P$ to represent the statistical value of GLCM. As we all know, the gray level of most images in the image itself. We use $P$ to represent the statistical value.

$$ASM = \left\{ \begin{array}{ll} 1 & \text{if } (i, j) \in \{1, 2\} \\ 0 & \text{otherwise} \end{array} \right.$$ (3)

$$P = \frac{1}{(M - \delta)N} \sum_{i,j} \delta$$ (4)

$$R_{ij} = \left\{ \begin{array}{ll} 1, & g_{ij} - g_{(i+\delta)j} = 0 \\ 0, & g_{ij} - g_{(i+\delta)j} \neq 0 \end{array} \right.$$ (5)

where $M$ and $N$ are the length and width of the CU, $g_{ij}$ is the gray value of the pixel at the position $(i, j)$. $R_{ij}$ represents the number of the same pixels in the 0 degree and 90 degree directions. The other angles are similar to these two angles and will not be listed.

As shown in Fig. 3, the “akiyo_cif” video sequence is divided into $8 \times 8$ squares and the ASM is calculated for each divided area. In Fig. 3, the angles used when calculating ASM are 0, 90, 180, 270, and every $8 \times 8$ pixel area is an ASM calculation unit. The ASM can well represent the texture complexity in a specific area of the image.

In order to study the influence of image texture complexity on the prediction mode decision in the video encoding process, we counted the JRMD and ASM of the CU generated during the encoding of many video sequences. According to the statistical results, we defined LowerLimit, HigherLimit, $P_1$ and $P_2$. Generally, LowerLimit is the smallest ASM among those CUs that continue to be split in CTU, and $P_2$ is the ratio of CUs with ASM less than LowerLimit. HigherLimit is the largest ASM among those CUs that have stopped splitting in CTU, and $P_1$ is the ratio of CUs with ASM greater than HigherLimit. In a CTU, it can almost be concluded that if the ASM of the current CU is greater than the HigherLimit, the current CU will be split into several sub-CUs. If the ASM of the CU is less than the LowerLimit, the current CU will stop splitting. This method can reduce a lot of calculations for JRMD. However, if the ASM of the current CU is between LowerLimit and HigherLimit, the original VTM algorithm is required to calculate JRMD to determine whether the current CU is split. According to the definitions of LowerLimit and HigherLimit, it can be seen that if the ASM of a CU is located between LowerLimit and HigherLimit, the encoding speed is slower, and vice versa. According to the definition of LowerLimit and HigherLimit, it can be seen that if there are a large number of CU ASMs between LowerLimit and HigherLimit, the encoding process will be slower. In order to make the algorithm more applicable, we add adaptive factors in LowerLimit and HigherLimit. The formula used in the iteration process is as follows,

$$H_c = \frac{1}{2} (H'_{c1} + H'_{c2})$$ (6)

$$L_c = \frac{1}{2} (L_{c1} + L_{c2})$$ (7)

$$H'_{c1} = \frac{1}{2} [1 + 2P_{c1} - (P_{11} - P_{22})] \times H_c$$ (8)

$$H'_{c2} = \frac{1}{2} [1 + 2P_{c2} - (P_{11} - P_{22})] \times L_c$$ (9)

where $H_c$ and $L_c$ are the HigherLimit and LowerLimit of the current CTU. $H'_{c1}$ and $L'_{c1}$ are the $H_c$ and $L_c$ added with the adaptive factor after the current CTU is encoded. $H'_{c2}$ and $L'_{c2}$ are the upper CTU, $H'_{c}$ and $L'_{c}$ of the left CTU. $P_{c1}$ and $P_{c2}$ are $P_1$ and $P_2$ of the current CTU, $P_{11}$ and $P_{12}$ are $P_1$ and $P_2$ of the upper CTU, and $P_{21}$ and $P_{22}$ are $P_1$ and $P_2$ of the upper CTU. We call the LowerLimit and HigherLimit of the first CTU in a frame of image $L_0$ and $H_0$. Fig. 4(a) shows the iteration of variables when the current CTU becomes the right CTU. Fig. 4(b) shows the iteration of variables when the current CTU becomes the left CTU.

To verify the proposed algorithm, we tested some video sequences. We took the “akiyo_cif” sequence as an example to explain the test results. Fig. 5 shows the verification of HigherLimit and LowerLimit by the statistical results of ASM and JRMD of a CTU when using the original VTM. Red dots indicate CUs that have stopped being split, and blue dots indicate CUs that have continued to be split. In the figure, the red dot above and the blue dot below indicate

**FIGURE 3.** The image and ASM image of “akiyo_cif” sequence.
CUs with incorrect depth prediction when the proposed algorithm is used. The dots between HigherLimit and LowerLimit indicate the CUs which are calculated by the original algorithm. The statistical results show that the proportion of CU depth prediction errors is 7.14%, the proportion of CU calculated using the original algorithm is 13.57%, and the proportion of correct CU depth prediction is 79.29%. We know that in a CTU, the wrong depth prediction will cause the BDBR to increase. If we reduce the wrong depth prediction, the number of CUs between the HigherLimit and LowerLimit will increase, which will reduce the encoding speed. Therefore, to balance performance and complexity, we should appropriately control $P_{c1}$ and $P_{c2}$. In order to reduce the complexity of the algorithm, we regard the CU located between HigherLimit and LowerLimit as a uniform distribution, and calculate $H_c$ and $L_c$. If it is regarded as an exponential distribution or logarithmic distribution to modify the formulas of $H_c$ and $L_c$, it is possible there will be better results. Since verifying this hypothesis will consume more work, we will verify it in future work.

**B. FAST CU SPLITTING MODE DECISION BASED ON TEXTURE DIRECTION**

The CU splitting mode in VVC is different from HEVC. Before CU is split in HEVC, it is only necessary to decide whether the CU further uses the QT structure splitting. In VVC, the CU not only needs to decide whether to split, but also selects an optimal partitioning mode by testing all candidate modes. Therefore, VTM consumes a lot of time and leads to extremely high complexity if CU is divided.

The image texture direction indicates that the pixel values along the image texture direction are very similar [15], and the ASM value in this direction is also very small. Intuitively, Fig. 6 shows that the pixels on the same texture are very similar.

In the intra encoding process, the optimal splitting mode means that the CU is split into sub-CUs, where the JRMD of each sub-CU is the smallest. According to the statistical data in Fig. 5, it is found that there is a correlation between ASM and JRMD and this correlation can eliminate some poor candidate patterns in advance. Specifically, if the best splitting direction is adjacent to the texture direction in determining the splitting mode process, those splitting modes that are perpendicular to the texture direction can be abandoned from the candidate set. Thus, the splitting mode adjacent to the texture direction is only selected. In order to simplify the algorithm, we only need to determine the horizontal texture complexity energy and vertical texture complexity energy of CU in the picture, which are calculated as,

$$SAD_{hor} = \frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} |Y(x,y) - \overline{Y}_x|$$

$$\overline{Y}_x = \frac{1}{M} \sum_{y=0}^{M-1} Y(x,y)$$

$$SAD_{ver} = \frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} |Y(x,y) - \overline{Y}_y|$$

$$\overline{Y}_y = \frac{1}{N} \sum_{x=0}^{N-1} Y(x,y)$$
where \( M \) and \( N \) represent the number of columns and rows, respectively. \( Y(x, y) \) represents the gray value of the pixel located at \((x, y)\), and \( \text{Th}_{\text{SAD}} \) is set as the splitting mode selection threshold. Generally, the CU texture direction predefines four sets of slope texture directions \((\pm 1/4, \pm 1/2, \pm 2, \pm 4)\), and then matches the calculated texture values with these four sets of texture directions [32]. We only need to judge whether the texture is horizontal or vertical, so we will use the texture directions with slopes of \( \pm 1/4 \) and \( \pm 4 \). If \( \text{Th}_{\text{SAD}} > 4 \), the current CU texture direction is vertical; if \( \text{Th}_{\text{SAD}} < 0.25 \), the current CU texture direction is horizontal; if \( \text{Th}_{\text{SAD}} \) is between \( 0.25 \) and \( 4 \), the current CU has no obvious texture direction.

In the encoding process, we can easily determine the texture direction based on \( \text{Th}_{\text{SAD}} \), and then quickly decide the CU splitting mode. After analyzing the “Campfire”, “Kimono”, “FourPeople” and “RaceHorses” sequences, we found that the shape ratio of the CU has an effect on \( \text{Th}_{\text{SAD}} \). Thus, the adaptive factor \( W_{hv} \) is introduced to eliminate this effect. The improved \( \text{Th}_{\text{SAD}} \) is expressed as,

\[
\text{Th}_{\text{SAD}} = W_{hv} \cdot \frac{\text{SAD}_{\text{hor}}}{\text{SAD}_{\text{ver}}}
\]

\[
W_{hv} = \frac{N}{M}
\]

where \( M \) and \( N \) represent the number of columns and rows, respectively. \( Y(x, y) \) represents the gray value of the pixel located at \((x, y)\), and \( \text{Th}_{\text{SAD}} \) is set as the splitting mode selection threshold. Generally, the CU texture direction predefines four sets of slope texture directions \((\pm 1/4, \pm 1/2, \pm 2, \pm 4)\), and then matches the calculated texture values with these four sets of texture directions [32]. We only need to judge whether the texture is horizontal or vertical, so we will use the texture directions with slopes of \( \pm 1/4 \) and \( \pm 4 \). If \( \text{Th}_{\text{SAD}} > 4 \), the current CU texture direction is vertical; if \( \text{Th}_{\text{SAD}} < 0.25 \), the current CU texture direction is horizontal; if \( \text{Th}_{\text{SAD}} \) is between \( 0.25 \) and \( 4 \), the current CU has no obvious texture direction.

In the encoding process, we can easily determine the texture direction based on \( \text{Th}_{\text{SAD}} \), and then quickly decide the CU splitting mode. After analyzing the “Campfire”, “Kimono”, “FourPeople” and “RaceHorses” sequences, we found that the shape ratio of the CU has an effect on \( \text{Th}_{\text{SAD}} \). Thus, the adaptive factor \( W_{hv} \) is introduced to eliminate this effect. The improved \( \text{Th}_{\text{SAD}} \) is expressed as,

\[
\text{Th}_{\text{SAD}} = W_{hv} \cdot \frac{\text{SAD}_{\text{hor}}}{\text{SAD}_{\text{ver}}}
\]

\[
W_{hv} = \frac{N}{M}
\]

where \( M \) and \( N \) represent the number of columns and rows, respectively. \( Y(x, y) \) represents the gray value of the pixel located at \((x, y)\), and \( \text{Th}_{\text{SAD}} \) is set as the splitting mode selection threshold. Generally, the CU texture direction predefines four sets of slope texture directions \((\pm 1/4, \pm 1/2, \pm 2, \pm 4)\), and then matches the calculated texture values with these four sets of texture directions [32]. We only need to judge whether the texture is horizontal or vertical, so we will use the texture directions with slopes of \( \pm 1/4 \) and \( \pm 4 \). If \( \text{Th}_{\text{SAD}} > 4 \), the current CU texture direction is vertical; if \( \text{Th}_{\text{SAD}} < 0.25 \), the current CU texture direction is horizontal; if \( \text{Th}_{\text{SAD}} \) is between \( 0.25 \) and \( 4 \), the current CU has no obvious texture direction.

In the encoding process, we can easily determine the texture direction based on \( \text{Th}_{\text{SAD}} \), and then quickly decide the CU splitting mode. After analyzing the “Campfire”, “Kimono”, “FourPeople” and “RaceHorses” sequences, we found that the shape ratio of the CU has an effect on \( \text{Th}_{\text{SAD}} \). Thus, the adaptive factor \( W_{hv} \) is introduced to eliminate this effect. The improved \( \text{Th}_{\text{SAD}} \) is expressed as,

\[
\text{Th}_{\text{SAD}} = W_{hv} \cdot \frac{\text{SAD}_{\text{hor}}}{\text{SAD}_{\text{ver}}}
\]

\[
W_{hv} = \frac{N}{M}
\]

Fig. 7 shows the framework of the proposed method, which combines the proposed Algorithm I and Algorithm II. The figure shows the complete process of CU splitting in intra prediction, in which the gray background module is the improved part of the paper. The proposed method is used to solve the high computational complexity during the CU partition process in the original VTM software. As shown, in the second stage, LowerLimit and HigherLimit of the current CU are calculated. In the third stage, the ASM of the current CU is calculated. It determines whether the current CU stops splitting by comparing the ASM with LowerLimit and HigherLimit. Then if the ASM of the current CU is between LowerLimit and HigherLimit, the original VTM algorithm is required to predict CU depth to determine whether the current CU is split. In the last step, a horizontal division mode (horizontal BT or TT splitting mode) or a vertical division mode (vertical BT or TT splitting mode) is selected after calculating the \( \text{Th}_{\text{SAD}} \) of the current CU.

IV. EXPERIMENTS AND RESULTS

In the experiment, we merge the proposed algorithm into VTM 7.0. Experiments with VTM 7.0 can eliminate the influence of the fast algorithm that added in the updated version on the comparison results with other methods. It uses the “encoder_intra_vtm” configuration file, the data in Table 1 is the average value when QP are 22, 27, 32, 37. Set GOP (Group of Pictures) to 1. Test the 13 standard dynamic range (SDR) video sequences [33] in Table 1. The RD performance is measured using Bjontegaard incremental bit rate (BDBR) [34]. Finally, the original VTM is compared with the proposed method. Table 1 lists the experimental results and clearly shows the comparison results. The time-saving \( TS_i \) is defined as:

\[
TS_i = \frac{\text{Tim}_{\text{original}} - \text{Tim}_i}{\text{Tim}_{\text{original}}} \times 100\%
\]

where \( \text{Tim}_{\text{original}} \) and \( \text{Tim}_i \) indicate the encoding time of the original VTM and the \( i \)-th algorithm. “Fast CU partition decision based on texture energy algorithm” is the algorithm proposed in Section III (A). We simplified the name of this algorithm to Algorithm I. Similarly, “Fast CU splitting mode decision based on texture direction algorithm” is the algorithm proposed in Section III (B), the name of this algorithm is simplified to Algorithm II.

| Class | Test sequences | Algorithm I | Algorithm II |
|-------|----------------|-------------|--------------|
|       | BDBR (%) | TS (%)      | BDBR (%) | TS (%)      |
| Class A | Campfire     | 0.65        | 41.39      | 0.48        | 35.29      |
|        | Kimono       | 0.77        | 37.12      | 0.55        | 33.92      |
| Class B | ParkScene    | 0.76        | 38.90      | 0.68        | 34.12      |
|        | BQTerrace    | 0.49        | 34.55      | 0.34        | 33.69      |
|        | RaceHorsesC  | 0.48        | 32.96      | 0.32        | 28.23      |
| Class C | BasketballDrill | 0.55       | 30.50      | 0.48        | 31.32      |
|        | PartyScene   | 0.43        | 33.22      | 0.28        | 26.85      |
|        | RaceHorses   | 0.43        | 28.02      | 0.28        | 29.02      |
| Class D | BBSquare     | 0.22        | 32.65      | 0.23        | 21.71      |
|        | BlowingBubbles | 0.41       | 32.76      | 0.28        | 26.77      |
|        | FourPeople   | 0.67        | 35.13      | 0.58        | 35.62      |
| Class E | Vidyo1       | 0.75        | 37.58      | 0.66        | 33.85      |
|        | KristenAndSara | 0.66    | 38.87      | 0.57        | 32.78      |
| Average |             | 0.57        | 34.89      | 0.45        | 31.01      |
Table 1 shows the performance comparison between the anchoring algorithm and the proposed single algorithm. The overall method proposed in Section III includes Algorithm I and Algorithm II. The data in Table 1 are the experimental results of the two algorithms. From Table 1, it can be seen that Algorithm I saves an average time of 34.89%, and BDBR increases by 0.57%. The average time saving of Algorithm II algorithm is 30.13%, and BDBR increases by 0.45%. Therefore, both Algorithm I and Algorithm II methods can effectively save coding time, and the loss of RD performance is negligible.

Table 2 shows the results of the proposed algorithm compared with two state-of-the-art fast methods including PV-CNN [23], FQPD-VG [24] and RFCTR [25] method. It can be found that the proposed overall algorithm saves 48.58% coding running time on average, the maximum is 55.93% (ParkScene), and the minimum is 37.14% (BQSquare). “ParkScene” contains fewer motion scenes and has clearer textures, so it saves more time than other test videos. Since many unnecessary splitting modes are reasonably skipped in areas with the fewest motion scenes and sharp textures, which can increase the reduction of coding time and only increase the BDBR by 0.91% on average. Therefore, the proposed algorithm can reduce a lot of intra prediction time and has good RD performance.

Figs. 8 and 9 show more details of comparing the proposed method with VTM 7.0. The proposed method can achieve coding time saving from low bitrate to high bitrate, and has almost similar RD performance. In addition, as the QP value increases, the proposed algorithm can continue to save more coding time. The reason is that as the QP value increases, the ASM value of most coding blocks calculated in Algorithm I will be too small, causing some CUs to stop splitting in advance. The Algorithm II will be easier to identify the texture direction, thereby speeding up the splitting mode decision.

As shown in Fig. 11, these four methods can save coding time more than 23%, where FQPD-VG method can save coding time more than 60% under some conditions. However, we should not just focus on the coding time saving and ignore BDBR. Compared with the PV-CNN and FQPD-VG, the proposed algorithm achieves a better balance between coding speed and BDBR. It can be seen that the proposed algorithm can achieve a total coding time reduction of 37.14%-55.93% and appropriate coding performance.
The proposed algorithm can reduce a lot of encoding time for the performance. The above simulation results show that the proposed algorithm saves more coding time or has better BDBR performance. In Fig. 10, while saving the coding time similar to RFCTRF method, the proposed algorithm has better RD performance by 51.34%-59.65%. Compared with the RFCTRF method includes two algorithms: a fast CU partition decision algorithm based on texture energy and a CU splitting mode decision algorithm based on texture direction. The texture characteristics are used to predict the CU splitting mode, and discard unnecessary candidate modes in the encoding process of VVC. The proposed algorithm is implemented on the latest VVC reference software VTM 7.0. Experiment results show that the average coding time of the proposed method is reduced by 48.58% compared with VTM 7.0, and the coding loss is negligible. We believe that our method will promote further research and improvement of the coding standard.

V. CONCLUSION
In this paper, we propose a fast CU partition based on texture features to speed up coding of VVC. The proposed overall method includes two algorithms: a fast CU partition decision algorithm based on texture energy and a CU splitting mode decision algorithm based on texture direction. The texture characteristics are used to predict the CU splitting mode, and discard unnecessary candidate modes in the encoding process of VVC. The proposed algorithm is implemented on the latest VVC reference software VTM 7.0. Experiment results show that the average coding time of the proposed method is reduced by 48.58% compared with VTM 7.0, and the coding loss is negligible. We believe that our method will promote further research and improvement of the coding standard.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] E. Aishina and J. Chen, Algorithm Description for Versatile Video Coding and Test Model 1 (VTM 1), document JVET-H002 ITU-T SG 16 WP 3 ISO/IEC JTC 1/SC 29/WG, Apr. 2018.

[3] J. Chen, Y. Ye, and S. H. Kim, Algorithm Description for Versatile Video Coding and Test Model 3 (VTM 3), document JVET-L1002 ITU-T SG 16 WP 3 ISO/IEC JTC 1/SC 29/WG, Oct. 2018.

[4] C. Chuang and T. Cheng, “Separate Tree Partitioning at 64×64-luma/32×32-Chroma Unit Level,” document JVET-K0230-v3, ITU-T SG 16 WP 3 ISO/IEC JTC 1/SC 29/WG, Jul. 2018.

[5] S.-H. Park and J.-W. Kang, “Fast affine motion estimation for versatile video coding (VVC) encoding,” IEEE Access, vol. 7, pp. 150875–150884, 2019.

[6] 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.

[7] K. Lim, J. Lee, S. Kim, and S. Lee, “Fast PU skip and split termination algorithm for HEVC intra prediction,” IEEE Trans. Circuits Syst. Video Technol., vol. 25, no. 8, pp. 1335–1346, Aug. 2015.

[8] A. S. Motra, A. Gupta, M. Shukla, P. Bansal, and V. Bansal, “Fast intra mode decision for HEVC video encoder,” in Proc. 26th Int. Conf. Softw., Telecommun. Comput. Netw., Split, India, 2012, pp. 1–5.

[9] A. Heindel and A. Kaup, “Fast intra mode decision in HEVC using early distortion estimation,” in Proc. IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP), Chengdu, China, Jul. 2015, pp. 559–565.

[10] L. Shen, Z. Zhang, and Z. Liu, “Effective CU size decision for HEVC intra coding,” IEEE Trans. Image Process., vol. 23, no. 10, pp. 4232–4241, Oct. 2014, doi: 10.1109/TIP.2014.2341927.

[11] H. Zhang and Z. Ma, “Fast intra mode decision for high efficiency video coding (HEVC),” IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 4, pp. 660–668, Apr. 2014.

[12] J. Chen, Y. Ye, and S. Kim, “Algorithm Description for Versatile Video Coding and Test Model 7 (VTM 7),” document JVET-P2002-v1 ITU-T SG 16 WP 3 ISO/IEC JTC 1/SC 29/WG, Oct. 2019.

[13] J. Chen, Y. Ye, and S. H. Kim, “Algorithm Description for Versatile Video Coding and Test Model 3 (VTM 3),” document JVET-L1002 ITU-T SG 16 WP 3 ISO/IEC JTC 1/SC 29/WG, Oct. 2018.

[14] 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.

[15] D. Zhang, Y. Chen, and E. Izquierdo, “Fast intra mode decision for HEVC based on texture characteristic from RMD and MPM,” in Proc. IEEE Vis. Commun. Image Process. Conf., Valletta, Malta, Dec. 2014, pp. 510–513, doi: 10.1109/VCIP.2014.7051618.
L. Shen, Z. Zhang, X. Zhang, P. An, and Z. Liu, “Two-stage fast multiple transform selection algorithm for VVC intra coding,” in Proc. IEEE Int. Conf. Multimedia Expo. (ICME), Shanghai, China, Jul. 2019, pp. 61–66.

M. Jamali, S. Coulombe, and F. Caron, “Fast HEVC intra mode decision based on edge detection and SATD costs classification,” in Proc. Data Compress. Conf., Snowbird, UT, USA, Apr. 2015, pp. 43–52.

H. S. Lee, K. Y. Kim, and G. H. Park, “Fast encoding algorithm based on depth of coding unit for high efficiency video coding,” J. Opt. Eng., vol. 51, no. 6, Jun. 2012, Art. no. 067402.

L. Shen, Z. Zhang, X. Zhang, P. An, and Z. Liu, “Fast TU size decision algorithm for HEVC encoders using Bayesian theorem detection,” Signal Process., Image Commun., vol. 32, pp. 121–128, Mar. 2015.

Y. Zhao, W. Zhang, H. Sun, and D. Zhang, “Efficient intra mode decision for low complexity HEVC screen content compression,” Proc. SPIE, vol. 9602, Mar. 2015.

Q. Hu, X. Zhang, Z. Shi, and Z. Gao, “Neyman-Pearson-Based early mode selection algorithm for HEVC encoding,” IEEE Trans. Multimedia, vol. 18, no. 3, pp. 379–391, Mar. 2016.

H. Gao, H. Shi, and A. Munteanu, “Deep learning based angular intra-prediction for lossless HEVC video coding,” in Proc. Data Compress. Conf. (DCC), Snowbird, UT, USA, Mar. 2019, p. 31.

G. Tang, M. Jin, 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), Sydney, NSW, Australia, Dec. 2019, pp. 1–4.

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.

Q. Zhang, Y. Wang, L. Huang, and B. Jiang, “Fast CU partition and intra mode decision method for H.266/VVC,” Proc. IEEE Access, vol. 8, pp. 117539–117550, 2020.

S. De-Luxan-Hernandez, V. George, J. Ma, T. Nguyen, H. Schwarz, D. Marpe, and T. Wiegand, “An intra subpartition coding mode for VVC,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Taipei, Taiwan, Sep. 2019, pp. 1203–1207.

Q. Zhang, Y. Wang, L. B. Huang, and X. Wang, “Fast CU partition decision for H.266/VVC based on the improved DAG-SVM classifier model,” Multimedia Systems, vol. 27, pp. 1–4, Aug. 2020.

K. Fan, R. Wang, W. Lin, J.-U. Hou, L. Duan, G. Li, and W. Gao, “Separable KLT for intra coding in versatile video coding (VVC),” in Proc. Data Compress. Conf. (DCC), Snowbird, UT, USA, Mar. 2019, p. 571.

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), Taipei, Taiwan, Sep. 2019, pp. 4120–4124.

Y. Liu, X. Liu, and P. Wang, “A texture complexity based fast prediction unit size selection algorithm for HEVC intra-coding,” in Proc. IEEE 17th Int. Conf. Comput. Sci. Eng., Chengdu, China, Dec. 2014, pp. 1585–1588.

J. Yu, “Texture segmentation based on FCM algorithm combined with GLCM and space information,” in Proc. Int. Conf. Electric Inf. Control Eng., Wuhan, China, Apr. 2011, pp. 4569–4572.

D. Ruiz, G. Fernández-Escríbano, J. L. Martínez, and P. Cuenca, “Fast intra mode decision algorithm based on texture orientation detection in HEVC,” Signal Process., Image Commun., vol. 44, pp. 12–28, May 2016.

F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, JVET Common test conditions and software reference configurations for SDR video, document JVET-L1010, ITU-T SG 16 WP3 ISO/IEC JTC 1/SC 29/WG, Joint Video Experts Team (JVET), Oct. 2018.

G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” in Proc. the 13th VCEG-M33 Meeting, Austin, TX, USA, Apr. 2001, pp. 1–5.

QIUPENG 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 field of pattern recognition and image processing. His major research interests include 3D signal processing, machine learning, pattern recognition, video codec optimization, and multimedia communication.

QIUPENG ZHANG (Member, IEEE) received the Ph.D. degree in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2003, the M.S. degree in computer science and technology from the Nanjing University of Science and Technology, China, in 2010, and the Ph.D. degree in software engineering from the China National Digital Switching System Engineering and Technological Research and Development Center, China, in 2019. He is currently an Associate Professor with the Computer College, Henan Institute of Engineering, China. His research interests include image processing, video codec optimization, and multimedia communication.

YONGBO ZHAO received the B.S. degree in electronic information science and technology from the Zhengzhou University of Light Industry, Zhengzhou, China, in 2018, where he is currently pursuing the master’s degree in computer science and technology with the School of Computer and Communication Engineering. His current research interests include image processing, high efficiency video coding, and extensions of the versatile video coding.

BIN JIANG received the Ph.D. degree in electronic circuit and system from the Beijing University of Technology, Beijing, China, in 2014, and the M.S. degree in computer application technology from Henan University, Kaifeng, China, in 2009. 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.

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 system from Shanghai University, China, in 2013. Since July 2013, he has been a Lecturer with the Zhengzhou University of Light Industry. His research interests include iterative learning control and pattern recognition.

TAO WEI received the B.S. degree in computer science and technology from PLA Information Engineering University, China, in 2003, the M.S. degree in computer science and technology from the Nanjing University of Science and Technology, China, in 2010, and the Ph.D. degree in software engineering from the China National Digital Switching System Engineering and Technological Research and Development Center, China, in 2019. He is currently an Associate Professor with the Computer College, Henan Institute of Engineering, China. His research interests include image processing, video codec optimization, and multimedia communication.