

Received May 15, 2020, accepted May 27, 2020, date of publication June 8, 2020, date of current version June 19, 2020. *Digital Object Identifier* 10.1109/ACCESS.2020.3000565

A Fast QTMT Partition Decision Strategy for VVC Intra Prediction

YIBO FAN^{®1}, (Member, IEEE), JUN'AN CHEN^{®1}, (Graduate Student Member, IEEE), HEMING SUN^{®2,4}, (Member, IEEE), JIRO KATTO^{®2,3}, (Member, IEEE), AND MING'E JING¹, (Member, IEEE)

¹State Key Laboratory of ASIC and System, Fudan University, Shanghai 201203, China

²Waseda Research Institute for Science and Engineering, Waseda University, Tokyo 169-8555, Japan

³Faculty of Science and Engineering, Waseda University, Tokyo 169-8555, Japan

⁴JST, PRESTO, Saitama 332-0012, Japan

Corresponding author: Heming Sun (hemingsun@aoni.waseda.jp)

This work was supported in part by Fudan University Key Laboratory Senior Visiting Scholarship, in part by the National Natural Science Foundation of China under Grant 61674041, in part by Alibaba Innovative Research (AIR) Program, in part by IBM Faculty Award, in part by the Innovation Program of Shanghai Municipal Education Commission, in part by the Pioneering Project of Academy for Engineering and Technology, in part by the Fudan-CIOMP (Changchun Institute of Optics, Fine Mechanics and Physics) joint fund, and in part by JST, PRESTO under Grant JPMJPR19M5, Japan.

ABSTRACT Different from the traditional quaternary tree (QT) structure utilized in the previous generation video coding standard H.265/HEVC, a brand new partition structure named quadtree with nested multitype tree (QTMT) is applied in the latest codec H.266/VVC. The introduction of QTMT brings in superior encoding performance at the cost of great time-consuming. Therefore, a fast intra partition algorithm based on variance and Sobel operator is proposed in this paper. The proposed method settles the novel asymmetrical partition issue in VVC by well balancing the reduction of computational complexity and the loss of encoding quality. To be more concrete, we first terminate further splitting of a coding unit (CU) when the texture of it is judged as smooth. Then, we use Sobel operator to extract gradient features to decide whether to split this CU by QT, thus terminating further MT partitions. Finally, a completely novel method to choose only one partition from five QTMT partitions is applied. Obviously, homogeneous area tends to use a larger CU as a whole to do prediction while CUs with complicated texture are prone to be divided into small sub-CUs and these sub-CUs usually have different textures from each other. We calculate the variance of variance of each sub-CU to decide which partition will distinguish the sub-textures best. Our method is embedded into the latest VVC official reference software VTM-7.0. Comparing to anchor VTM-7.0, our method saves the encoding time by 49.27% on average at the cost of only 1.63% BDBR increase. As a traditional scheme based on variance and gradient to decrease the computational complexity in VVC intra coding, our method outperforms other relative existing state-of-the-art methods, including traditional machine learning and convolution neural network methods.

INDEX TERMS Asymmetric block size, fast partition decision, intra prediction, quadtree with multi-type tree, versatile video coding.

I. INTRODUCTION

With the fast development of the video market, there is a growing demand for videos of higher resolution and quality. Therefore, the adoption and application of video coding are faced with a huge challenge and there is an urgent need for the creation of future generation video coding standards that can support high-resolution well. To address this issue,

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Callico¹⁰.

Video Coding Experts Group (VCEG) and Moving Picture Experts Group (MPEG) collaboratively published Call for Proposals (CfP) on video compression and its extensions [1]. Joint Video Exploration Team (JVET) is composed of VCEG and MPEG. The Versatile Video Coding (VVC) standard was launched by JVET after evaluating the CfP responses. As the latest video coding standard, VVC can achieve an overall 43.81% bit-rate reduction than its predecessor High Efficiency Video Coding (HEVC) [2], but at the cost of a huge amount of computational complexity.

107900

Similar to its predecessor HEVC, VVC uses blockbased coding with each frame first divided into coding tree units (CTUs) and then CTUs are further partitioned into smaller coding units (CUs) of different sizes. Quadtree with nested multi-type tree (QTMT) is an obvious difference in intra prediction between VVC and HEVC. In HEVC, only quaternary tree (QT) partition is allowed which means CUs can only be square shapes. The width and height of a CU should be 64, 32, 16 or 8 at the same time. The width and height of a prediction unit (PU) can also be 4. However, in VVC, asymmetric partitions are permitted so the width and height of a CU no longer need to be the same. Theoretically, they can be any combination of 128, 64, 32, 16, 8 or 4. The length of 2 or 1 is even existed in PU owing to the newlyintroduced intra sub-partition (ISP) mode [3]. Meanwhile, it should be noted that there exist some restrictions by setting several parameters in the configuration file of the reference software in practice, so the actual permitted CU sizes may differ from theory. In the default setting, for example, Max-CUWidth and MaxCUHeight are both 64 so the 128×128 CTU must do QT partition as the first step. MinQtSize is set as 8 which means the permitted minimum size of a QT node is 8×8 and an 8×8 CU can only do MT partitions. MaxBtSize and MaxTtSize are both 32, which means the largest MT node is 32×32 . In the QTMT structure, five ways can be utilized to split a block, including QT, horizontal binary tree (BH), vertical binary tree (BV), horizontal ternary tree (TH) and vertical ternary tree (TV). BH and BV are called binary tree (BT) together. TH and TV are called ternary tree (TT) together. The union of BT and TT is called multitype tree (MT). The transition state of QT and QTMT is called quadtree plus binary tree (QTBT) which is an advanced coding technology put forward by JVET. Then QTBT evolves into QTMT after improvement.

By using MT partitions, rectangular sizes are derived. Fig.1 shows the five possible partition structures of a QT node. It is noted that a QT node can be partitioned by all the five modes but an MT node can no longer be partitioned by QT, only MT is allowed no matter the sub-block shape of the MT node is square or not. In actual operation, the partition is strictly limited by the size of CU, the shape of CU, current CU depth and partition of neighboring CUs. There exist redundant CU splits that are forbidden in reality. Fig.2 shows several restriction examples. Case (a) is when a QT node is split by TH, the area of this square is divided into three parts from top to bottom. The second sub-part cannot be further split by BH because the final partition is the same as first split the square by BH and then split the two sub-CUs by BH separately. Case (b) is when a QT node is split by BH and its upper sub-CU is split by BV, the lower sub-CU cannot be split by BV anymore. This is because the partition is totally the same as directly split the square by QT. Case (c) is an asymmetric version of case (a). When this MT node is partitioned by TV first, the second part of it cannot be split by BV.



FIGURE 1. Five partition structures of QTMT using QT node as an example.



FIGURE 2. Examples of partition restrictions for redundant CU splits. (a) and (b) represent QT node. (c) represents MT node.

Over the years, plenty of contributions have been made to accelerate the intra partition decision process in HEVC, its extensions, and VVC. Different methods adopted by former researchers will be illustrated in Section II. Although various fast algorithms in HEVC have contributed to excellent encoding time-saving in previous works, they cannot be transplanted to VVC directly since the partition structure of the two standards are totally different. In order to make an improvement in VVC intra partition to reduce computational complexity, this paper proposes a fast algorithm with three steps to solve the emerging asymmetric partition problem. The main contributions of our work are as follows:

1. Apart from the conventionally-used early termination method to directly skip all the further partitions of CU, we also use Sobel operator to compute gradient features so that QT partition can be decided and asymmetric rectangular partitions are terminated. Then a completely novel method to choose only one partition from five QTMT partitions by calculating the variance of variance of each sub-CU is applied, which is totally different from the traditional simple computation of the CU variance to judge homogeneity.

2. With our proposed fast algorithm, the encoding time is reduced by 49.27% compared to anchor VTM-7.0 with a relatively small increase, 1.63%, in BDBR. As a traditional scheme based on variance and gradient to decrease the computational complexity in VVC intra coding, our method outperforms other relative existing state-of-the-art methods, including machine learning and neural network methods.

The organization of this paper is as follows. Section II illustrates state-of-the-art acceleration methods adopted in HEVC and VVC intra partition. Section III is a brief explanation of our motivation to propose such an algorithm. The main body of our method is shown in detail in Section IV, where the detailed judgment procedures will be presented. Section V shows the experimental results of our proposed algorithm and comparisons with other works are given. Section VI is a conclusion of this paper.

II. RELATED WORKS

Over the years, plenty of researches have been conducted on the issue of intra partition no matter based on the predecessor of H.266/VVC, H/264/AVC, and H.265/HEVC, or coding standards proposed by other organizations. Since VVC is proposed recently, there are still not many related studies. As a predecessor and closest version to VVC, HEVC has a lot of research results that are worth to learn from. The following is a brief generalization of methods used on HEVC to reduce computational complexity based on different methods. The methods can be generally classified into three groups. The first is traditional correlation-based method, which mainly uses variance features or gradient features to detect textures [4]-[6], [20], [23]. The second is traditional machine learning method, where decision tree (DT) and support vector machine (SVM) are frequently used [7]–[12], [19], [21], [22]. The other is the recently emerging end-to-end neural network method, which uses convolution layers, pooling layers and fully connected layers to form a network and then use it to do classification. The output of network represents a different choice which is whether to further partition the CU or not in the intra partition problem [12]-[18]. In part II.A, approaches applied to the complexity reduction of HEVC QT structure are presented [4]-[18]. In part II.B, existing state-of-the-art QTBT and QTMT intra partition methods dealing with the novel asymmetric partition issue are concluded [19]-[23].

A. METHODS FOR QT

Convolution neural network (CNN) is an emerging method in recent years, which only requires building an optimized network without any extra consideration of what features to use. The latest progress shows that the whole partition structure of a CTU can be decided by a CNN. Several representative approaches are concluded here. Reference [4] first establishes a large database and randomly divides them into training, validation and test sets. Then it forms the partition of an entire CTU as a hierarchical CU partition map (HCPM). CNN is used to predict the HCPM. In [5], a deep CNN model is constructed to predict the partition of CTU. It first establishes a large-scale database called CPIH database and then models the partition as a three-level classification problem. The partition decisions of three levels are realized in one CNN framework with a different convolution filter size on the first layer for each level. An asymmetric-kernel CNN structure is proposed in [6] which has three branches in the first convolution layer. The first is a conventional square shape kernel and the other two are asymmetric kernels used to detect near-horizontal and near-vertical textures. The output of the three kernels are the same, so they can be concatenated. Although these CNN methods are efficient enough to reduce the computational complexity of intra coding in HEVC, it is not that easy in VVC. On one side, we need to consider not only the depth of each block but also whether it is able to be partitioned into this structure in VVC due to the rules introduced by QTMT. If a CNN model is adopted to generate

the entire partition of CTU in VVC, we cannot ensure that this is a feasible partition. On the other side, since the side lengths of CUs in VVC are variable and the width and height of a CU can be combinations of different values, there are great difficulties in setting the parameters of the network.

Traditional machine learning methods are also used in a wide range. They usually extract useful features in the first step and then find ways to minimize an error function or use Bayesian decision rule, DT or SVM to make judgments. Reference [7] puts forward a level filtering strategy to reduce the number of prediction unit levels from five to two. It uses the partition decision of HEVC reference software HM as the ground truth to minimize error rate and receive an optimal threshold by training. Reference [8] adopts features like neighboring CU depth, rate-distortion cost, and coding flags to construct the proposed SVM classifier. An online-learning approach based on Bayesian decision rule is shown in [9] to reduce the computational complexity of screen content coding (SCC). Variance and gradient features also play an important role in machine learning methods. Apart from the aforementioned strategies, there are some others that make use of variance and gradient as features to product SVM or DT, like [10] and [11]. In [10], variance and gradient kurtosis were two features used for distinguishing different types of blocks. The features can classify a block as natural image or screen content, "partitioned" or "non-partitioned", "directional" or "non-directional". Reference [11] illustrates an adaptive fast CU size decision algorithm, which extracts certain image features by neighboring mean squared error (NMSE) and angular Sobel operator, and then employs SVM to analyze and construct the classification model.

Traditional correlation-based methods are adopted frequently in decades. Reference [12] adopts both correlationbased and SVM methods. It first uses average gradients in the horizontal direction and vertical direction computed by Sobel operator to early terminate homogeneous CUs. Then two linear SVM employing the depth difference and Hadamard transform-based (HAD) costs ratio as features to do early split and early termination of CUs. Work [13]–[18] are typical correlation-based approaches to detect the texture of CUs and make early termination judgments. Reference [13] computes the variance of Sum of Absolute Differences (SAD) among pixels in a CU in four directions. Then the number of evaluated CU sizes is reduced by comparing the smoothness parameter to an experimentally derived threshold. Reference [14] presents a fast CU size decision method by exploiting the depth information of neighboring CUs. A concept of edge pixel density produced by Canny operator is introduced to decide whether a CU is a texture CU and make early termination accordingly. Global and local edge complexities in horizontal, vertical, 45° and 135° diagonal directions are used in [15] to determine the partition of CU by comparing the edge complexity of a CU and its sub-CUs. The global edge is calculated by the complexity difference between the two halves in a CU and local edge is obtained by four local filters. Based on the texture homogeneity, [16] develops a method for

the early determination of CU size with adaptive thresholds by calculating mean absolute deviation (MAD). Meanwhile, a novel bypass scheme based on a weighted average depth of the neighboring coded CUs is also proposed. Reference [17] computes the average luminance and the variance of the subblocks to decide whether the sub-blocks are in a smooth area. If so, they are likely to be a whole part. A texture analysis method based on local range is introduced in [18], which is the variation of a pixel relative to its local neighborhood. It computes the mean and variance of local range (LR) and finds that CUs with high LR values tend to split further, and vice versa.

B. METHODS FOR QTBT AND QTMT

The emerging QTBT and QTMT intra partition structure is a breakthrough in improving the coding performance of video sequences. However, the unavoidable increase in computational complexity leaves researchers a puzzle. There are several publications recently dealing with the issue of QTBT and QTMT. References [19] and [20] are proposals on QTBT, where [19] uses a traditional machine learning method and the other uses an end-to-end CNN method. In [19], a dynamic partition parameter derivation method (DPPD) at the CTU level is proposed to reduce partition in homogeneous areas. Meanwhile, a four-output decision tree structure is designed at the CU level to further remove unnecessary splitting iterations and control the risk of false prediction. Reference [20] is another proposal on QTBT. The QTBT partition range is formulated as a multi-class classification problem in [20]. CNN is adopted to predict the partition depth range of 32×32 CU based on the inherent texture richness of the block.

References [21]–[23] are the state-of-the-art techniques dealing with the QTMT issue. Reference [21] adopts the Bayesian decision rule to eliminate the redundant selection of QTMT. The split types and intra prediction modes of sub-CUs are adopted as the input features. Early skip for vertical split including BV and TV is first conducted, then is the early skip for TH. A novel fast QTMT decision framework using decision tree is developed in [22] to determine the partition based on texture features like gradient and local difference which is evaluated by texture variance. Reference [23] creates a CNN model with changeable kernel sizes to deal with the flexible side lengths in QTMT.

The differences between the proposed fast partition decision algorithm and the previous methods are described as follows. Variance and gradient features are explored to make decisions on QTMT partition. Instead of the conventional computation of the CU variance to judge homogeneity, the variance of variance of each sub-CU in a CU under different partition conditions is computed.

III. MOTIVATION

Intra prediction is very time-consuming due to the complicated Rate-Distortion Optimization (RDO) process. Although the number of modes that need to go through the RDO process is reduced compared to HEVC, the complicated QTMT partition structure still gives rise to significant computational complexity and time increase in VVC encoder. Therefore, if we can reduce the number of CUs that need to do the RDO process, the time spent on RDO will be decreased. Thus, we focus on cutting down the partitions that need to be done within a CU.

In HEVC, intra partition is simply deciding whether to split the CU by QT, which is a yes-or-no question because there is only one partition in HEVC intra prediction. So, the partition decision in HEVC intra coding is mainly an early termination or early split problem. However, the QTMT in VVC is not as easy as QT in HEVC. We need to decide not only whether the CU should be further partitioned, but also how the CU should be partitioned. There are five partitions in total, and traversing them will cost a great amount of time while a random choice will lead to great encoding quality loss according to our test. Consequently, in designing VVC fast intra algorithm, it is unavoidable to consider these two issues together, which will be completely different from previous research in HEVC.

VVC is now an imperfect version and is constantly being revised. Since the first decisive JVET conference on VVC in 2018, few adaptive algorithms have been proposed to speed up the QTMT intra prediction. As aforementioned, there is a Bayesian method in [21], a decision tree method in [22] and a CNN method in [23] to our knowledge up to now. Although machine learning and neural network methods are becoming increasingly popular in recent years in solving problems in image processing and video coding field, we cannot deny that traditional methods such as variance-based or gradient-based are also preferred.

IV. PROPOSED ALGORITHM

The basis of our work is the official document [24] published by JVET and the relevant reference software VTM-7.0. The VTM supports both symmetric and asymmetric partitions, which means the width and the height of a CU can be either the same or different. According to VTM default settings, the maximum size of a CU is 128×128 and it must first be divided by QT as mentioned in Section I. Then the four 64×64 sub-CUs can be divided by QT or not since MaxBt-Size and MaxTtSize are both 32. As a result, for 64×64 CUs, it is only a problem of partition or not. As for how to choose among the five QTMT partitions, we only need to deal with CUs which are no larger than 32×32 . Considering that directly deciding the partition of 64×64 CU may lead to a huge loss in encoding performance, we do not take the partition decision of 64×64 into our account in this work. Also, small CUs do not occupy much encoding time, so we finally take a compromise. We thus decide to determine the partition of 32×32 CU by using fast algorithm. Therefore, the encoding quality and efficiency can achieve a nice balance.

In order to design a feasible algorithm, we need to consider the correlation between pixels and the changing tendency in a row or in a column. Since different textures within a CU are likely to be divided into different sub-parts, catering to such division is apparently a good way. However, before conducting this step, we first consider two cases that have the tendency to directly skip further split or choose a certain partition. Our fast algorithm generally includes three steps. First, we compute the variance of the 32×32 CU to decide whether to early terminate further splitting of it. By terminating further splitting of the CU, all the five partitions will be skipped and the time will be saved greatly. This method is widely used in partition because homogeneous areas usually stay as a whole. Second, gradient features are extracted by Sobel operator. These features are used to decide whether to choose QT as the partition of CU. If so, the other four partitions will be skipped. This step early terminates all the MT partitions. Third, one partition structure from the five QTMT structures is chosen based on the variance of variance of each sub-CU. Part IV.A shows the details of the three steps in our method.

A. THREE-STEP FAST ALGORITHM

1) STEP 1. EARLY TERMINATION BASED ON VARIANCE

First, we need to determine whether the block is homogeneous. If so, there is no need to split this block any further. This method has been proposed in HEVC acceleration scheme in [18], so here we borrow the core idea of it as a preprocessing step. We use equation (1) to compute the variance of the original pixels in the 32×32 CU. If the variance is lower than a threshold TH_1 , we can deem this CU as a flat texture.

$$var = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} (X(i,j) - \mu)^2$$
(1)

where *var* on the left side of the equation is the calculated variance and μ on the right side is the mean value of all the pixels in the CU. Here, W and H are literally the width and height of the CU, which are both 32 in our proposed case. The judgment condition of step 1 is: if *var* < *TH*₁ then skip all further split.

Here gives the reason why we use variance to determine if the partition of CU should be early terminated or not. We find that flat areas tend to have small variances after we analyze variance, horizontal gradient and vertical gradient features of CU original pixels. Meanwhile, the gradient features are not similar under different circumstances. On one hand, small gradients and similar horizontal and vertical gradients only occur under the case of flat textures when pixel values are similar to each other, but on the other hand, flat textures not necessarily have small gradients. That is to say, there exist special cases where variance can explicitly reflect the smooth nature of textures while gradient cannot. Such situations usually occur when an area is smooth visually but variable microscopically. For instance, a monotonous area that has been blurred by camera lens focus or post-processing gives the human eye a feeling that it is like a single background color, especially the area that is already smooth enough, such as grass or animal fur. However, we cannot ensure that these natural sceneries do not have any stains. If there are several sporadic dark spots that are negligible to the human eye, but can lead to a great increase in microscopical gradient because the total gradient is an accumulation of absolute gradient values of each pixel, we cannot arbitrarily make the conclusion that this area is not flat. Thus, we choose variance instead of gradient as the feature to early terminate further partition of 32×32 CU.

2) STEP 2. CHOOSING QT BASED ON GRADIENT

Second, we compute the sum of the absolute gradients of each pixel. Equation (2) and (3) are the calculation of the total gradient in horizontal and vertical direction respectively. D_X is obtained by D_x of each point and D_Y is obtained by D_y of each point. D_X and D_Y are calculated as

$$D_X = \sum_{i=1}^{W} \sum_{j=1}^{H} abs(D_x(i,j))$$
(2)

$$D_Y = \sum_{i=1}^{W} \sum_{j=1}^{H} abs(D_y(i,j))$$
(3)

where D_x and D_y are extracted by using the Sobel operator. D_x and D_y are calculated as equation (4) and (5).

$$D_x(i,j) = M_{i,j} \times \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$
(4)

$$D_{y}(i,j) = M_{i,j} \times \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$
(5)

M in equation (4) and (5) is the 3×3 original pixel matrix centered with the point currently being calculated. *i* and *j* represent the position of the current center pixel in a row and in a column respectively. For pixels in the top row, bottom row, leftmost column, and rightmost column, we pad the pixels outside the CU with its nearest pixel value within the CU, as shown in Fig.3. In the schematic diagram, each small square block represents a pixel. The shaded part represents the current CU, and the white is the padding part.



FIGURE 3. Schematic diagram of padding when computing gradients.

We use the ratio of D_X and D_Y to signify a tendency that the area is a horizontal texture or a vertical texture. We divide the bigger one of D_X and D_Y by the other and if the quotient is smaller than a threshold TH_2 , it means the overall gradients in the horizontal and the vertical direction are similar and we can view the texture of this block as monotonous. It is noted that monotonous not necessarily means homogeneous, because the CU may still have complex textures that appear in a repeated pattern or a horizontal and vertical symmetrical mode, like tile walls and chessboard. Apparently, partitions of such textures cannot be skipped and QT is the best choice because the most important feature of this style is horizontal and vertical symmetry. Among the five partition structures, only QT meets this condition. Also, in this case, D_X and D_Y should not be small, so we add a condition that if D_X and D_Y are both larger than a threshold TH_3 , then the CU is directly partitioned by QT without processing any MT partitions. If step 1 is said to be a termination of all partition structures, then step 2 can be viewed as a termination of all the MT partition structures, where rectangular partitions can be directly excluded.

The judgment condition of step 2 is: if the following three conditions are met $(1 < D_X/D_Y < TH_2 \text{ or } 1 < D_Y/D_X < TH_2 (2) D_X > TH_3 (3) D_Y > TH_3$, then select QT and skip MT.

3) STEP 3. CHOOSING ONE PARTITION FROM FIVE CANDIDATES BASED ON VARIANCE OF VARIANCE

If the 32×32 CU does not meet the conditions in step 1 and step 2, variance of the variance of each of its sub-CU under all the five partition conditions will be computed separately. For each QTMT partition, the variance of original pixels of every sub-CU is computed first to get a set of variances. Then the variances of the variance sets are computed to derive five values. Each value corresponds to one partition. The maximum of the variances is chosen and the corresponding partition is selected as the only partition of current CU.

The theoretical basis of this step is that blocks are partitioned to sub-blocks with relatively different textures from each other, which means different textures are likely to be split into different sub-blocks to achieve a better prediction performance. Therefore, the differences among the variance of each sub-block tend to be large. To get a more quantitative theoretical support, we use 100 images from DIV2K data set to get the amount of five candidates with different variance of variance hitting final correct partition, which can be called ground truth, without step 1 and step 2. The statistical results using four quantization parameters (QPs) are shown separately in Table 1. The meaning of each column from top to bottom is as follows: QP, the partition with the largest variance of variance being the ground truth, the partition with the second largest variance of variance being the ground truth, the partition with the third largest variance of variance being the ground truth, the partition with the fourth largest variance of variance being the ground truth, the partition with the smallest variance of variance being the ground truth.

Intuitively, the results in Table 1 show that the partition corresponding to the second maximum variance of variance seems to be the most probable selection. However, these statistics are derived without step 1 and step 2 and meanwhile, if QT is directly chosen in step 2, there is no need to further choose one partition based on variance of variance. Therefore, on one hand, we cannot ignore the influence caused by the decision of step 2. On the other hand, it is not feasible to count the ground truth partitions after step 2, since the thresholds used in step 1 and step 2 are fixed after our whole method is determined. If we first use arbitrary values of three thresholds to do this test and then decide a final scheme to set the thresholds, it will fall into an endless loop. As a solution, we further test the number of CUs which select QT as ground truth in each variance of variance group to exclude the impact of step 2 as much as possible. The results are shown in Table 2. By analyzing the data in Table 1 and Table 2 together, we can find that although the proportion of the partitions corresponding to the second maximum variance of variance is the largest in Table 1, the number of CUs partitioned by QT that belong to the second group also occupies the largest proportion in Table 2. Therefore, we can reasonably speculate that the *tendency* of selecting one partition from five partitions is choosing the one with a relatively large variance of variance. The second largest value being the best result in Table 1 may be related to the selection of QT partition in the previous step.

The calculation expressions of the five partitions are shown in equation (6). In the equation, each denominator of the first fraction to the right of the equation represents the number of sub-CUs when using the corresponding partition structure. *k* represents the index of the *k*-th sub-CU. For instance, the value of *k* of the left part and the right part in a BH partition are 1 and 2 respectively. w_k , h_k and μ_k are the width, height and the mean value of pixels of the *k*-th sub-CU. μ_{QT} , μ_{BT} , μ_{BV} , μ_{TH} and μ_{TV} are the mean values of the variances of all the sub-CUs under corresponding partition conditions.

$$\begin{cases} var_{QT} = \frac{1}{4} \left(\sum_{k=1}^{4} \left(\frac{1}{w_{k} \times h_{k}} \sum_{i=1}^{w_{k}} \sum_{j=1}^{h_{k}} (X(i,j) - \mu_{k})^{2} - \mu_{QT} \right) \right)^{2} \\ var_{BH} = \frac{1}{2} \left(\sum_{k=1}^{2} \left(\frac{1}{w_{k} \times h_{k}} \sum_{i=1}^{w_{k}} \sum_{j=1}^{h_{k}} (X(i,j) - \mu_{k})^{2} - \mu_{BT} \right) \right)^{2} \\ var_{BV} = \frac{1}{2} \left(\sum_{k=1}^{2} \left(\frac{1}{w_{k} \times h_{k}} \sum_{i=1}^{w_{k}} \sum_{j=1}^{h_{k}} (X(i,j) - \mu_{k})^{2} - \mu_{BV} \right) \right)^{2} \\ var_{TH} = \frac{1}{3} \left(\sum_{k=1}^{3} \left(\frac{1}{w_{k} \times h_{k}} \sum_{i=1}^{w_{k}} \sum_{j=1}^{h_{k}} (X(i,j) - \mu_{k})^{2} - \mu_{TH} \right) \right)^{2} \\ var_{TV} = \frac{1}{3} \left(\sum_{k=1}^{3} \left(\frac{1}{w_{k} \times h_{k}} \sum_{i=1}^{w_{k}} \sum_{j=1}^{h_{k}} (X(i,j) - \mu_{k})^{2} - \mu_{TV} \right) \right)^{2} \end{cases}$$

$$(6)$$

The judgment condition of step 3 is: if $max(var_{QT}, var_{BH}, var_{BV}, var_{TH}, var_{TV}) = var_n$, then select n(n = QT, BH, BV, TH, TV). For example, if the maximum of the five variances is var_{BH} , then select BH as the final partition.

QP	22	27	32	37
Max var	22900	21384	19058	15857
2 nd max	26524	23877	20829	17153
3 rd max	18231	17408	16265	14949
4 th max	9536	9884	10738	11460
5 th max	8665	9279	10383	12012

TABLE 1. Statistical results of the association between the value of variance of variance and ground truth partition.

TABLE 2. Statistical results of the distribution of QT partition in relation to the order of variance of variance.

QP	22	27	32	37
Max var	16935	14308	10718	6444
2 nd max	20427	16852	12547	7370
3 rd max	11661	9673	7167	4234
4 th max	3676	3017	2195	1254
5 th max	2185	1786	1303	800

B. DERIVATION OF THRESHOLDS

In step 1 and step 2 of our proposed algorithm, the three thresholds we use, namely TH_1 , TH_2 and TH_3 , are all derived by concatenating ten 1024 × 1024 images from DIV2K data set, including natural scenery, buildings, animals and people, into a sequence as a training sequence. Before training, we first analyze the features of the three thresholds and find that under the fixed condition that block size is 32×32 , TH_1 is positively related to quantization parameter (QP) while TH_2 and TH_3 are irrelevant to QP. Thus, the thresholds can be expressed by equation (7), (8) and (9) separately.

$$TH_1 = \alpha \times QP \tag{7}$$

$$TH_2 = \beta \tag{8}$$

$$TH_3 = \gamma \tag{9}$$

where α , β and γ are all adjustable parameters.

The initial value of the parameters come from our previous work [25]. We first set β and γ as constants and run the training sequence with different values of α s. The selection procedure is shown in detail in Fig.4. The blue line in each chart represents BDBR and the orange line represents time saving. The red points are the final selected values. It is noted that, although the selection of the three parameters should be based on both encoding performance and efficiency, the values of time reduction derived in all cases fluctuate around 46%, so time does not contribute to an important factor during this judgment and BDBR is the priority. The line chart of BDBR and time saving as a function of α is shown in Fig.4(a). Apparently, the smaller α is, the better the encoding performance is because fewer 32×32 CUs are skipped compared to conditions with larger α s. As a result, there is no need to choose α with the best encoding quality, so we choose the value at the turning point of BDBR. The performance of the encoder remains almost the same when α is less than 9, but begins to turn worse with the growth of α , so α is set as 9.

Now that α is fixed, we vary β to decide the best TH_2 . The line chart of BDBR and time saving as a function of β is shown in Fig.4(b). In our algorithm, TH_2 cannot be too large since the probability of a CU to be partitioned by QT cannot be too high. If TH_2 is set as a large value, it means almost all the CUs will be partitioned by QT and the prediction program will not proceed into step 3. As a result, similar to the selection process of α , the turning point 2.7, the slope before and after which changes apparently, is chosen to be the value of β to ensure a good performance. The line chart of BDBR and time saving as a function of γ is shown in Fig.4(c). In the figure, we can see that BDBR fluctuates with the changing of γ , but its changes are not drastic when γ is in a certain range. So we select γ corresponding to the smallest BDBR, which is 30000. So finally, we set (α, β, γ) as (9, 2.7, 30000).

The realization of our method is implemented into VTM-7.0 and pseudo codes of the proposed algorithm are shown in Algorithm 1. A judgment of channel is first made before all the computation since our method is currently applied to luminance component only. The RDO of the 32×32 CU itself is computed before the following three proposed steps.

Algorithm 1 Fast Partition Decision
FastPartition(currentCU)
{
if isLuma and SIZE == 32×32 then
RDO(currentCU)
varCurrCU = var(currentCU);
for i from 1 to 32 do
for j from 1 to 32 do
$D_x(i, j) = \text{Sobel}(M(i, j))$
$D_y(i, j) = Sobel(M(i, j))$
end for
end for
$D_X = sum(abs(D_x(i, j)))$
$D_{\rm Y} = {\rm sum}({\rm abs}(D_{\rm y}({\rm i},{\rm j})))$
if varCurrCU $<$ TH ₁ then
skip SPLIT;
else if $((1 < D_X/D_Y < TH_2)$ or $(1 < D_Y/D_X < TH_2))$
and $((D_X > TH_3)$ and $(D_Y > TH_3))$ then RDO(QT);
else
$varQT = var(var(subCU_1), var(subCU_2),$
var(subCU ₃), var(subCU ₄));
$varBH = var(var(subCU_1), var(subCU_2));$
$varBV = var(var(subCU_1), var(subCU_2));$
$varTH = var(var(subCU_1), var(subCU_2),$
var(subCU ₃));
$varTV = var(var(subCU_1), var(subCU_2),$
var(subCU ₃)); RDO(PartitionOfMax(varQT,
varBH, varBV, varTH, varTV));
end if
end if
}







FIGURE 4. Training results of each parameter concerning both BDBR and time saving. The blue line in each chart represents BDBR and the orange line represents time saving. The red points are the final selected values. The values of time saving in all cases fluctuate around 46%, so when selecting the best point, BDBR is the priority. (a) Line chart of BDBR and time saving as a function of α . Apparently, when α is less than 9, BDBR remains almost the same. When α is greater than 9, BDBR is proportional to α . 9 is a turning point, the slope before and after which changes apparently. (b) Line chart of BDBR and time saving as a function of β . β cannot be too large since the probability of a CU to be partitioned by QT cannot be too high. As a result, turning point 2.7, the slope before and after which changes apparently, is chosen to ensure a good performance. (c) Line chart of BDBR and time saving as a function of γ . Because the value fluctuates, γ corresponding to the smallest value of BDBR in a certain range is directly selected.

V. EXPERIMENTAL RESULTS

A. TEST CONDITIONS

We use the recently-released version of official reference software VTM-7.0 to test our fast algorithm. All our tests are under All-Intra (AI) configuration and common test conditions (CTC) [26]. Test sequences with up to 100 frames each are used and the QP is set as 22, 27, 32 and 37 according to the standard. Bjøntegaard Delta Bit Rate (BDBR) [27] and time saving (TS) are used to measure the overall performance of our method. Fast algorithm usually leads to increase in BDBR and the BDBR of all the sequences are averaged to reflect an integral encoding quality. The larger the value of increase in BDBR is, the worse the encoding quality is. Average time saving (ATS) of all the test sequences are used to measure the complexity reduction of the encoder. A larger value of ATS means more time has been saved and the fast algorithm is efficient enough to reduce the computational complexity. To have an intuitive evaluation of the performances of different methods, we use the metric mentioned in [23], which is comparing ATS/ABDBR (average BDBR).

TS of each test sequence is computed by (10).

$$TS(\%) = \frac{Time_{ORI} - Time_{PROPOSED}}{Time_{ORI}}$$
(10)

B. TEST RESULTS OF PROPOSED ALGORITHM

The experimental results of BDBR, TS and ATS/ABDBR of all the test sequences are shown in Table 3 and Table 4. The two tables also present the test results of several relevant works. It should be noted that we turn off ISP, MIP and LFNST in the configuration file of VTM-7.0 which was not included in VTM-2.0 for fair comparison. Compared to anchor VTM-7.0, our method achieves 49.27% ATS with only 1.63% BDBR increase. It is noted that all the thresholds in our method are obtained by training ten 1024 \times 1024 images from DIV2K data set. So the derivation of thresholds has no connection with the test sequences, which means the test results are convincing enough to demonstrate the effectiveness of our method.

Table 5 shows the performance of each step in our method. In this test, still 100 frames are employed to obtain a qualitive

TABLE 3. Coding performance of the proposed algorithm compared to traditional ML methods.

Class	Sequence	JCDT[19] (QTBT/VTM-2.0)		Cascade DT[22] (QTMT/VTM-2.0)			Proposed (QTMT/VTM-7.0)			
	*	BDBR(%)	TS(%)	TS/BDBR	BDBR(%)	TS(%)	TS/BDBR	BDBR(%)	TS(%)	TS/BDBR
	BasketballDrive	4.22	67.98	16.11	2.70	70.39	26.07	3.28	59.35	18.09
D	BQTerrace	3.47	48.57	14.00	2.44	66.89	27.41	1.08	45.30	41.94
(1020×1080)	Cactus	4.14	49.20	11.88	2.45	66.86	27.29	1.84	52.44	28.50
(1920×1080)	Kimono	1.97	65.40	33.20	2.24	69.08	30.84	1.93	59.51	30.83
	ParkScene	5.08	60.80	11.97	1.98	66.60	33.64	1.26	51.84	41.14
	BasketballDrill	6.64	44.61	6.72	2.39	58.47	24.46	1.82	48.48	26.64
С	BQMall	4.87	30.19	6.20	2.54	65.41	25.75	1.87	52.47	28.06
(832×480)	PartyScene	1.90	40.25	21.18	1.00	60.45	60.45	0.26	38.62	148.54
	RaceHorsesC	2.42	36.24	14.98	1.59	61.29	38.55	0.88	49.05	55.74
	BasketballPass	9.21	49.33	5.36	2.59	58.40	22.55	1.95	47.70	24.46
D	BlowingBubbles	2.36	27.65	11.72	1.31	57.16	43.63	0.47	40.35	85.85
(416×240)	BQSquare	4.53	37.47	8.27	1.21	62.05	51.28	0.19	31.95	168.16
	RaceHorses	2.74	29.33	10.70	1.44	57.79	40.13	0.54	41.69	77.20
Б	FourPeople	5.08	43.10	8.48	3.39	66.49	19.61	2.70	57.57	21.32
E (1280×720)	Johnny	7.33	52.27	7.13	3.71	66.25	17.86	3.22	56.88	17.66
(1280×720)	KristenAndSara	5.15	43.34	8.42	3.09	67.13	21.72	2.78	55.11	19.82
Average		4.44	45.36	10.21	2.25	63.79	28.30	1.63	49.27	30.23

TABLE 4. Coding performance of the proposed algorithm compared to CNN method.

Class	Sequence	(Q	CNN[23] TMT/VTM-:	5.0)	Proposed (QTMT/VTM-7.0)		
	1	BDBR(%)	TS(%)	TS/BDBR	BDBR(%)	TS(%)	TS/BDBR
P	BQTerrace	0.95	34.50	36.32	1.08	45.30	41.94
B (1020×1080)	Kimono	0.87	33.32	38.30	1.93	59.51	30.83
(1920×1080)	ParkScene	0.83	35.41	42.66	1.26	51.84	41.14
G	BasketballDrill	1.30	33.39	25.68	1.82	48.48	26.64
(822) (820)	PartyScene	0.55	31.10	56.55	0.26	38.62	148.54
(832×480)	RaceHorsesC	0.37	23.63	63.86	0.88	49.05	55.74
P	BlowingBubbles	0.95	33.90	35.68	0.47	40.35	85.85
D (41(×240)	BQSquare	0.68	30.73	45.19	0.19	31.95	168.16
(416×240)	RaceHorses	0.71	31.79	44.77	0.54	41.69	77.20
E	FourPeople	1.38	38.01	27.54	2.70	57.57	21.32
(1280×720)	KristenAndSara	1.61	34.84	21.64	2.78	55.11	19.82
Average		0.93	32.78	35.25	1.26	47.22	37.35

TABLE 5. Coding performance of step 1 and 2 and step 3 in the proposed algorithm.

Class		BDBR(%)	TS(%)		
Class	Step 1	Step 2	Step 3	Step 1 & 2	Step 3
В	+0.07	+1.40	+0.48	+44.70	+8.99
С	+0.03	+0.88	+0.33	+40.79	+6.36
D	+0.02	+0.50	+0.29	+32.05	+8.37
Е	+0.16	+1.54	+1.36	+33.02	+23.50
Average	+0.06	+1.01	+0.56	+38.37	+10.90

value. When testing TS, we take step 1 and step 2 as a whole because we need to extract and store the data into matrices and calculate the variance in the first step. This process will take some time that almost compensate the time saved by step 1, so the changes in performance contributed by step 1 only is not obvious. We can see from the table that step 1 and step 2 contribute to 38.37% ATS together with 0.06% and 1.01% increase in ABDBR each. Step 3 further saves 8.90% encoding time at the cost of 0.56% ABDBR increase. It is noted that the average values in this table are all obtained by averaging the data of each *sequence*, so the results may differ from the average of each *class* after rounding.

C. COMPARISON WITH OTHER WORKS

We first compare our results with traditional machine learning methods, which are shown in Table 3. One of them uses a joint multi-class decision tree (JCDT) and the other uses a cascade decision tree (cascade DT). Wang's work [19] is originally a QTBT-oriented proposal based on HEVC reference software HM-13.0, but the method is reimplemented into VTM-2.0 by [22], so we use the test result of this method on VTM-2.0 here as a comparison. Compared to [19], our method reduces 3.91% encoding complexity and saves 2.81% BDBR, which is a considerable breakthrough. Yang's work [22] is QTMT-oriented and as we can see, the overall algorithm of [22]

TABLE 6. Coding performance measured by MS-SSIM and VMAF scores.

Class Saguanaa			MS-SSIM	(1 frame)		Aggregate VMAF (100 frames)
Class	Sequence	QP=22	QP=27	QP=32	QP=37	QP=32
	BasketballDrive	0.9968	0.9957	0.9926	0.9876	99.2887
	BQTerrace	0.9973	0.9955	0.9919	0.9839	96.0853
в	Cactus	0.9956	0.9944	0.9894	0.9809	91.3113
	Kimono	0.9981	0.9962	0.9915	0.9822	93.9501
	ParkScene	0.9964	0.9937	0.9879	0.9764	88.5855
	BasketballDrill	0.9962	0.9923	0.9860	0.9777	92.3974
C	BQMall	0.9983	0.9966	0.9929	0.9847	95.4246
C	PartyScene	0.9979	0.9965	0.9908	0.9758	91.4694
	RaceHorsesC	0.9977	0.9954	0.9904	0.9776	99.8332
	BasketballPass	0.9968	0.9933	0.9872	0.9815	91.6196
р	BlowingBubbles	0.9977	0.9942	0.9886	0.9769	89.8974
D	BQSquare	0.9980	0.9959	0.9948	0.9926	94.9279
	RaceHorses	0.9982	0.9955	0.9894	0.9810	99.7492
	FourPeople	0.9987	0.9975	0.9948	0.9892	91.1023
Е	Johnny	0.9982	0.9971	0.9947	0.9920	92.2218
	KristenAndSara	0.9984	0.9973	0.9950	0.9912	92.0872
	Average	0.9975	0.9954	0.9911	0.9832	93.7469





IEEEAccess

FIGURE 5. CU partition results of different algorithms. (a) Partition of BasketballPass by VTM-7.0. (b) Partition of BasketballPass by our method.



FIGURE 6. Decoded frames of different algorithms. (a) One decoded frame of BQMall by VTM-7.0. (b) One decoded frame of BQMall by our method.

contributes to 63.79% ATS with 2.25% BDBR increase. Our method does better in encoding quality at the expense of less ATS, which can be considered as a trade-off. By using the ATS/ABDBR metric, the value 30.23 in our method is higher than 10.21 in [19] and 28.30 in [22], showing that our scheme is competitive to the previous algorithms.

Then we compare our work with a CNN method, which is presented in Table 4. Since the test results given by [23] is not complete according to CTC, we only compare the existing data in it with ours. From Table 4, we can see the ATS/ABDBR of [23] is 35.25 while ours is 37.35, which is higher than the CNN method.

D. RESULTS UNDER MS-SSIM AND VMAF METRICS

Despite the frequently adopted BDBR metric, MS-SSIM (Multiscale Structural Similarity) [28] and VMAF (Visual Multimethod Assessment Fusion) [29] are also used to further verify the performance of our method. We test one frame for

MS-SSIM (QP = 22, 27, 32, 37) and 100 frames for VMAF (QP = 32) of each sequence from class B to class E and the scores are shown in Table 6. All the results retain 4 decimal places. From the table, we can see by using MS-SSIM metric, the performance of our method can achieve scores of more than 0.98 on average under all four tested QP conditions. From the perspective of VMAF, which is a comprehensive measure that combines multiple factors, our method still performs well. The score is 93.7469 on average when QP is 32.

E. PARTITION ANALYSIS

The two figures in Fig.5 show the different partitions of the first frame from BasketballPass by using the default algorithm in VTM-7.0 and our algorithm separately. To see the partitions of each CU more clearly, we analyze the situation when QP is 37. The red square in Fig.5(a) represents a 32×32 CU. Since five partitions are permitted for a 32×32 CU, we cannot simply compare whether the CUs in Fig.5(b) are split by the same way as in Fig.5(a). The part outlined by the blue frame in each figure is an example of the same partition at 32×32 level. In other parts, some areas use the same partition, and some are different. The orange part is an example of where our partition is not detailed enough. This kind of difference leads to the loss in performance but considering that the figures are derived by setting QP as 37, the skip of partition is tolerable especially by subjective evaluation.

F. SUBJECTIVE QUALITY EVALUATION

When QP is large, the differences between VTM-7.0 and our method by subjective judgment are probably subtle, so we use the decoded frames of BQMall from class C when QP is set as 22. From the two figures in Fig.6, we can see that the differences are also barely visible to the naked eye. The people, the reflection in the glass, the text on the board and other parts are almost the same.

VI. CONCLUSION

In this paper, we present a fast QTMT partition algorithm based on variance and gradient to reduce the computational complexity brought in by the novel MT partitions in VVC. We solve the asymmetric partition problem caused by QTMT and achieve a quite good performance on reference software VTM-7.0. Based on the similarity of pixel values in smooth areas, variance of original pixels is computed to early terminate further partition of 32×32 CU. If the horizontal texture and vertical texture are judged as similar by using Sobel operator, QT partition is directly chosen, thus MT partitions are early terminated. Meanwhile, by utilizing the feature that sub-parts of a split CU are prone to have different textures from each other, the partition types that need to be reversely traversed of 32×32 CU is reduced from five to one. An extra ten-frame sequence composed of ten 1024×1024 images from DIV2K data set is utilized to derive three thresholds in our method, therefore test results of standard video sequences are convincing enough to reveal the effectiveness of our method. Our algorithm outperforms other state-of-theart intra coding algorithms and achieves a pretty good tradeoff between complexity reduction and coding efficiency.

REFERENCES

- A. Segall, V. Baroncini, J. Boyce, J. Chen, and T. Suzuki, *Joint Call for Proposals on Video Compression With Capability Beyond HEVC*, document JVET-H1002, Joint Video Exploration Team (JVET), Oct. 2017.
- [2] C. Hsu, C. Chen, T. Chuang, H. Huang, S. Hsiang, C. Chen, M. Chiang, C. Lai, C. Tsai, Y. Su, Z. Lin, Y. Hsiao, J. Klopp, I. Wang, Y. Huang, and S. Lei, *Description of SDR Video Coding Technology Proposal by MediaTek*, document JVET-J0018, Joint Video Exploration Team (JVET), Apr. 2018.
- [3] 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.
- [4] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, and Z. Guan, "Reducing complexity of HEVC: A deep learning approach," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5044–5059, Oct. 2018.
- [5] T. Li, M. Xu, and X. Deng, "A deep convolutional neural network approach for complexity reduction on intra-mode HEVC," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Hong Kong, Jul. 2017, pp. 1255–1260.
- [6] J. Shi, C. Gao, and Z. Chen, "Asymmetric-kernel CNN based fast CTU partition for HEVC intra coding," in *Proc. IEEE Int. Symp. Circuits Syst.* (*ISCAS*), Sapporo, Hokkaido, May 2019, pp. 1–5.
- [7] H. Sun, D. Zhou, and S. Goto, "A low-complexity HEVC intra prediction algorithm based on level and mode filtering," in *Proc. IEEE Int. Conf. Multimedia Expo*, Melbourne, VIC, Australia, Jul. 2012, pp. 1085–1090.
- [8] M. Grellert, B. Zatt, S. Bampi, and L. A. da Silva Cruz, "Fast coding unit partition decision for HEVC using support vector machines," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 6, pp. 1741–1753, Jun. 2019.
- [9] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Online-learningbased Bayesian decision rule for fast intra mode and CU partitioning algorithm in HEVC screen content coding," *IEEE Trans. Image Process.*, vol. 29, pp. 170–185, 2020.
- [10] F. Duanmu, Z. Ma, and Y. Wang, "Fast mode and partition decision using machine learning for intra-frame coding in HEVC screen content coding extension," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 517–531, Dec. 2016.
- [11] 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.
- [12] T. Zhang, M.-T. Sun, D. Zhao, and W. Gao, "Fast intra-mode and CU size decision for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 8, pp. 1714–1726, Aug. 2017.
- [13] M. Ramezanpour and F. Zargari, "Early termination algorithm for CU size decision in HEVC intra coding," in *Proc. 9th Iranian Conf. Mach. Vis. Image Process. (MVIP)*, Tehran, Iran, Nov. 2015, pp. 45–48.
- [14] X. Shang, G. Wang, T. Fan, and Y. Li, "Fast CU size decision and PU mode decision algorithm in HEVC intra coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Quebec City, QC, Canada, Sep. 2015, pp. 1593–1597.
- [15] B. Min and R. C. C. Cheung, "A fast CU size decision algorithm for the HEVC intra encoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 5, pp. 892–896, May 2015.
- [16] L. Shen, Z. Zhang, and Z. Liu, "Effective CU size decision for HEVC intracoding," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4232–4241, Oct. 2014.
- [17] M. U. K. Khan, M. Shafique, and J. Henkel, "An adaptive complexity reduction scheme with fast prediction unit decision for HEVC intra encoding," in *Proc. IEEE Int. Conf. Image Process.*, Melbourne, VIC, Australia, Sep. 2013, pp. 1578–1582.
- [18] T. Mallikarachchi, A. Fernando, and H. K. Arachchi, "Efficient coding unit size selection based on texture analysis for HEVC intra prediction," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Chengdu, China, Jul. 2014, pp. 1–6.
- [19] 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.

- [20] 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, Sep. 2018.
- [21] 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.
- [22] 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.
- [23] 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)*, Sydney, NSW, Australia, Dec. 2019, pp. 1–4.
- [24] J. Chen, Y. Ye, and S. Kim, Alogorithm Description for Versatile Video Coding and Test Model 7 (VTM 7), document JVET-P2002, Joint Video Exploration Team (JVET), Oct. 2019.
- [25] 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.
- [26] K. Suehring and X. Li, JVET Common Test Conditions and Software Reference Configurations, document JVET-B1010, Joint Video Exploration Team (JVET), Feb. 2016.
- [27] G. Bjontegaard, Calculation of Average PSNR Differences Between RD Curves, document ITU-T SG16 Q6, VCEG-M33, Austin, TX, USA, Apr. 2001.
- [28] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, vol. 2, 2003, pp. 1398–1402.
- [29] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara. (Sep. 2019). Toward a Practical Perceptual Video Quality Metric. [Online]. Available: https://medium.com/netflix-techblog/toward-apractical-perceptual-video-quality-metric-653f208b9652



JUN'AN CHEN (Graduate Student Member, IEEE) received the B.E. degree in microelectronics from Fudan University, Shanghai, China, in 2018. She is currently pursuing the M.S. degree in circuits and systems with Fudan University. She joined the State Key Laboratory of ASIC and System, Fudan University, in 2016, as a Student Researcher. Her research interest includes algorithms and VLSI design for video codec.



HEMING SUN (Member, IEEE) received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2011, the M.E. degrees from Waseda University and Shanghai Jiao Tong University, in 2012 and 2014, respectively, through a double-degree program, and the Ph.D. degree in engineering from Waseda University, in 2017. He was a Researcher with the NEC Central Research Laboratories, from 2017 to 2018. He is currently an Assistant Professor with

Waseda University. He is selected as a JST PRESTO Researcher, for the period of 2019–2023. His research interest includes algorithms and VLSI architectures for image/video processing and neural networks.



JIRO KATTO (Member, IEEE) received the B.S., M.E., and Ph.D. degrees from The University of Tokyo, in 1987, 1989, and 1992, respectively. He worked at NEC Corporation, from 1992 to 1999. He was a Visiting Scholar with Princeton University, Princeton, NJ, USA, from 1996 to 1997. He then joined Waseda University, in 1999. His research interests include multimedia signal processing and multimedia communication systems. He is a Fellow of IEICE and a member of ACM.



YIBO FAN (Member, IEEE) received the B.E. degree in electronics and engineering from Zhejiang University, Hangzhou, China, in 2003, the M.S degree in microelectronics from Fudan University, Shanghai, China, in 2006, and the Ph.D. degree in engineering from Waseda University, Tokyo, Japan, in 2009. He was an Assistant Professor with Shanghai Jiao Tong University and Fudan University, from 2009 to 2014, and an Associate Professor with Fudan University, from

2014 to 2019. He is currently a Full Professor with the College of Microelectronics, Fudan University. His research interests include image processing, video coding, machine learning, and associated VLSI architecture.



MING'E JING (Member, IEEE) received the B.E., M.S., and Ph.D. degrees from Xidian University, Xi'an, China, in 1999, 2002, and 2005, respectively. She is currently an Associate Research Fellow with Fudan University, Shanghai, China. Her research interests include image processing, video coding, machine learning, and algorithm design.

...