

Fast Algorithm for 360-degree Videos Based on the Prediction of Cu Depth Range and Fast Mode Decision

Mengmeng Zhang¹, Jing Zhang¹, Zhi Liu¹, Fuqi Mao^{1,2}, Wen Yue²

¹ North china university of technology, Beijing 100144, China

² China university of geosciences, Beijing 100083, China

[e-mail: muchmeng@126.com, 381889962@qq.com, lzliu@ncut.edu.cn,
maofuqi@ncut.edu.cn, yw@cugb.edu.cn]

*Corresponding author: Mengmeng Zhang, Zhi Liu

*Received July 25, 2018; revised September 28, 2018; accepted December 26, 2018;
published June 30, 2019*

Abstract

Spherical videos, which are also called 360-degree videos, have become increasingly popular due to the rapid development of virtual reality technology. However, the large amount of data in such videos is a huge challenge for existing transmission system. To use the existing encode framework, it should be converted into a 2D image plane by using a specific projection format, e.g. the equi-rectangular projection (ERP) format. The existing high-efficiency video coding standard (HEVC) can effectively compress video content, but its enormous computational complexity makes the time spent on compressing high-frame-rate and high-resolution 360-degree videos disproportionate to the benefits of compression. Focusing on the ERP format characteristics of 360-degree videos, this work develops a fast decision algorithm for predicting the coding unit depth interval and adaptive mode decision for intra prediction mode. The algorithm makes full use of the video characteristics of the ERP format by dealing with pole and equatorial areas separately. It sets different reference blocks and determination conditions according to the degree of stretching, which can reduce the coding time while ensuring the quality. Compared with the original reference software HM-16.16, the proposed algorithm can reduce time consumption by 39.3% in the all-intra configuration, and the BD-rate increases by only 0.84%.

Keywords: 360-Degree Video, CU Partition, Fast Mode Selection, RMD Optimization

1. Introduction

In virtual reality video systems, multiple cameras are used to capture the frequently changing 360-degree real-world scenes. The captured scenes are subsequently displayed in a spherical format. People can freely view the scenes through a head-mounted display (HMD) and experience the real world immersively. In a typical 360-degree video compression and delivery framework, the stitched input 360-degree videos, represented in a native projection format, e.g., equirectangular (ERP), are converted into another projection format, e.g., cubemap (CMP), segmented sphere (SSP), equatorial Cylindrical(ECP) [1], etc.. These projection formats are important and would potentially improve the representation efficiency and coding performance. The existing HEVC coding standard can only process 2D planar video content. In order to be able to process 360-degree video, we need to project it to the 2D plane using these projection formats for further processing. Currently, most 360-degree panoramic videos are provided in ERP format, this paper only studies the characteristics of ERP projection format. ERP is a simple projection method that aims to map the warp into a vertical line with a constant interval and the weft as a horizontal line with constant spacing so that points can be mapped to a 2D plane. Most 360-degree videos have high resolution and high frame rate to provide a realistic visual experience, and these features create problems in storage capacity and transmission bandwidth. Therefore, efficient video coding tools are crucial for compressing VR video content.

High-efficiency video coding (HEVC) [2] is the latest video coding standard developed by the Joint Collaborative Team on Video Coding (JCT-VC). Compared with the H.264 codec, HEVC presents remarkable improvements in compression capability [3]. For example, an HEVC-compressed video is twice as efficient as that compressed by H.264. Videos with the same visual quality only occupy half of the space. Coding unit (CU) [4], prediction unit (PU), and transformation unit (TU) are used in HEVC coding. With regard to fixed-size macroblocks in H.264, HEVC provides CU sizes from 64×64 to 8×8 , and their corresponding depths are 0 to 3 respectively. Each CU can be further divided into PU and TU. A total of 35 prediction modes are defined based on PU, but the specific implementation of the intra prediction process is based on TU. The standard stipulates that PU can be divided into TUs in the form of a quadtree, and TUs share a prediction mode in the same PU. Rough traversal through Rough Mode Decision (RMD) in 35 modes produces a candidate list, which is compared with MPM to obtain the optimal intra prediction mode in each PU. However, these processing methods do not consider the characteristics of the ERP format video. Thus, a processing algorithm that can adapt to the ERP format content needs to be designed.

Current intra encoding compression algorithms mainly involve CU fast partitioning and PU mode selection. In CU fast partitioning, optimization is performed from premature termination of a CU partition by predicting the depth range. In PU mode selection, optimization is performed in 35 prediction modes. For early termination of a CU partition and depth skip, Shen et al., “[5] proposed a fast CU size decision algorithm for HEVC intra coding to speed up the process by reducing the number of candidate CU sizes required to be checked for each treeblock. Shen et al., “[6] proposed a fast CU size decision algorithm to determine CU depth range (including minimum and maximum depth levels) and skip specific depth levels that are rarely used in the previous frame and neighboring CUs. Kim et al., “[7] proposed an efficient CU determination algorithm for using spatial and temporal information, in which 13 neighboring coding tree units (CTUs) are defined. Lee et al., “[8] proposed a CU

size decision algorithm based on statistical analysis to speed up the intra coding process from three aspects: SKIP mode skipping, CU skipping ahead of time and CU premature termination. Min et al., “[9] proposed a novel fast algorithm for the CU size decision in intra coding. Global and local edge complexities in horizontal, vertical, 45° diagonal, and 135° diagonal directions are proposed and used to decide the partitioning of a CU. Cho et al., “[10] proposed a fast CU splitting and pruning method for HEVC intra coding. The proposed fast splitting and pruning method is performed in two complementary steps: (1) early CU split decision and (2) early CU pruning decision. In selection of PU fast mode: Zhang et al., “[11] proposed the Hadamard cost-based progressive rough mode search to selectively check the potential modes instead of traversing all candidates. Zhang et al., “[12] analyzed the relation between a block's texture characteristics and its best coding mode to further develop an adaptive strategy based on the analysis for fast mode decision in intra coding for HEVC. Zhu et al., “[13] proposed a novel intra prediction modes pruning method based on decision trees and a new three-step search algorithm, aiming at achieving higher encoding efficiency compared to the standard-HEVC. Liu et al., “[14] proposed an adaptive mode decision algorithm based on texture complexity and direction for HEVC intra prediction. Zhang et al., “[15] proposed a fast intra CU decision algorithm based on the texture characteristics of video and refined the partition results by considering the coding bits of each CU as auxiliary information. However, these algorithms are generally used for natural sequences and do not consider the characteristics of 360-degree video, which have higher resolution and uneven sampling distribution after being projected onto a 2D image.

This work presents a fast decision-making algorithm for CU partition and an adaptive mode decision for intra prediction modes in the ERP format of 360-degree videos. First, according to the characteristics of the ERP format(the pole region is severely stretched while the equatorial region is less stretched), that is, the compression coding needs to be processed separately according to the locations of CTUs. Therefore, the algorithm first determines the locations of CTUs and determines the different neighboring reference blocks based on the locations. Then different decision conditions are set to predict the depth interval of the current CTU. Second, in the prediction mode, the original 35 prediction modes are first optimized by reducing angle traversal modes based on PU depth [16, 17] to reduce complexity. Considering that the horizontal stretching of the pole area is severe, that is, the lateral prediction mode is more likely to become the optimal mode. Therefore, the traversal interval is further optimized. Furthermore, in order to obtain the optimal mode more accurately, the algorithm applies two RMD processes [18]. Experimental results show that computational complexity is further reduced in while maintaining the video quality. The rests of the paper are organized as follows: Section 2 presents the proposed algorithm. Section 3 provides the results of the experiment. Section 4 shows the conclusion.

2. Early termination of CU partition and fast intra mode decision

2. 1 Early termination of CU partition

The most widely used format for 360-degree videos at present is the ERP format because it has a simple mapping relationship. Most test sequences are stored in ERP format and compressed by video encoding. However, the points mapped onto a 2D plane through ERP projection are neither equal-area nor equi-angular. That is, ERP projection stretches the original sequence, especially in the pole area. Hence, it causes serious great redundancy in video content and reduces the efficiency of coding compression.

The video redundancy caused by the characteristics of the ERP projection format also affects the encoding information of the current CTU. Because the projection of the ERP format causes the pole area of the video content to be severely stretched, which will result in redundancy and distortion of the video content in the pole region, and makes it inaccurate to predict the depth information of the current CTU by referring to the adjacent three CTUs (Left, Above-Left, Above) in pole area. That is, it will produce wrong predictions that affect video quality. If the original framework is not modified, the accuracy of the prediction will be reduced. Hence, this work classifies the CTUs of the pole and equatorial regions in ERP format and handles these CTUs accordingly. The video content is severely stretched in the pole area, especially in the horizontal direction, which renders the spatial correlation of the adjacent CTU on the left side larger than that on the above and above-left adjacent CTUs [19]. That is, the left adjacent CTU has a larger weight in predicting the depth interval of current CTU. Furthermore, due to the horizontal stretching of the pole region, the correlation of the adjacent above CTU is less than that of the left CTU, and the CTU distortion caused by horizontal stretching makes it inaccurate to refer to the entire above CTU. Therefore, the algorithm uses only the 64×32 -size block above the current CTU for reference, as shown in Fig. 1. The spatial correlation of the CTU [20, 21] in the above-left direction is smaller than that of the above CTU. This notion indicates that, the weight of the above-left CTU is extremely small, and the distortion caused by stretching makes it unreasonable to refer the above-left CTU. Therefore, we only refer to the left CTU and the above 64×32 -size block and set the judgment conditions to predict the depth interval of the current CTU.

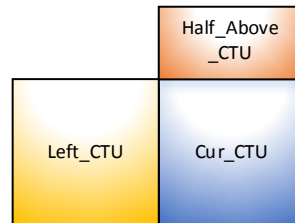


Fig. 1. Prediction depth range

To determine the depth interval of the current CTU quickly, we define the cost function of the absolute CU depth difference ($AbsCUDD_d$). The formula is as follow:

$$AbsCUDD_d = \frac{1}{N} \sum_{i=0}^{N-1} |depth_i - d|, \quad (1)$$

where $depth_i$ delegates the depth value corresponding to the i -th 4×4 block in the CTU, and its range is $[0, 3]$ and value is discretely distributed; the baseline depths of d are set to 0, 1, 2, and 3, respectively. Because the range of $depth_i$ is $[0, 3]$, in order to calculate all the absolute CU depth difference ($AbsCUDD_d$) of different blocks in CTU, the values of d need to be changed accordingly. Therefore we set four different standard depth values (define 0, 1, 2, 3 as baseline depths) specified in the HEVC Coding Standard that correspond to the sizes of CU are 64×64 , 32×32 , 16×16 , 8×8 depth values; N represents the number of 4×4 blocks in the CTU. The size of the CTU is 64×64 , and the smallest block size is 4×4 , that means it exists the 256 of 4×4 -size blocks.

This study distinguishes the equatorial region from the pole area by calculating weight w in WS-PSNR because this weight applies to the ERP projection format. The area decision algorithm is as follow: with CTU as a unit, the sum of the weights in each CTU in the first

column is calculated sequentially and averaged, as shown in Formulas (2) and (3), and we denote it as W_{CTU} . W_{CTU} represents the weight value of each row of CTU. The range of W_{CTU} is (0, 1), and when $W_{CTU} < 0.5$, the current areas are divided into the pole area; when $0.5 \leq W_{CTU} < 1$, the current areas are divided into the equatorial region.

$$W(i, j) = \cos\left(\left(j - \frac{height}{2} + \frac{1}{2}\right) \times \frac{\pi}{height}\right), \quad (2)$$

$$W_{CTU}(i, j) = \frac{1}{N} \sum_{index \times N}^{N-1+index \times N} w(i, j), \quad (3)$$

where N is the width of CTU, height is the height of image and j represents the y coordinate of a pixel in a image (j is from 0 to the height of image). The index is the serial number of a CTU in a column.

Pole area:

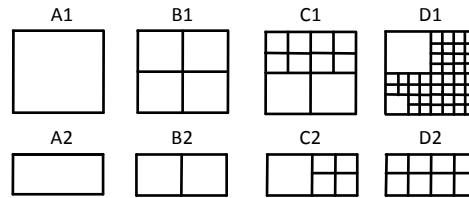


Fig. 2. Division of left CTU and above 64×32 -size blocks in the pole area.

Table 1. $AbsCUDD_d$ for A1, B1, C1, and D1

$AbsCUDD_d$	A1	B1	C1	D1
$AbsCUDD_0$	0	1	1.5	2.4375
$AbsCUDD_1$	1	0	0.5	1.4375
$AbsCUDD_2$	2	1	0.5	0.5625
$AbsCUDD_3$	3	2	1.5	1.5625

Table 2. $AbsCUDD_d$ for A2, B2, C2, and D2

$AbsCUDD_d$	A2	B2	C2	D2
$AbsCUDD_0$	0	1	1.5	2
$AbsCUDD_1$	1	0	0.5	1
$AbsCUDD_2$	2	1	0.5	0
$AbsCUDD_3$	3	2	1.5	1

Table 1 and **Table 2** display the $AbsCUDD_d$ values corresponding to the division of the CTUs of A1, B1, C1, and D1 and A2, B2, C2, and D2 in **Fig. 2**. A1, B1, C1 and D1 are the divisions of the left CTU, and A2, B2, C2, and D2 are the partitions of the above 64×32 size blocks. This study's evaluation is in accordance with the above-mentioned CTU division situations and the corresponding $AbsCUDD_d$ values. $CUDepthTypePole_0$ is defined as the depth category value of the left CTU, and $CUDepthTypePole_1$ is defined as the depth type value of the above 64×32 size blocks. $CurCUType$ is defined as the depth type value of the current CTU prediction depth interval. $CUDepthTypePole_0$ and $CUDepthTypePole_1$ are used to determine $CurCUType$ to obtain the depth prediction range of the current CTU according to the correspondence relationship, as shown in **Table 3**. $CUDepthTypePole_0(AbsCUDD_d$ for A1, B1, C1, and D1) is calculated using Formula (4):

$$CUDepthTypePole_0 = \begin{cases} 0 & AbsCUDD_0 + AbsCUDD_1 = 1 \\ 2 & AbsCUDD_2 + AbsCUDD_1 = 1 \text{ \& } AbsCUDD_1 < 0.5 \\ 3 & \begin{cases} AbsCUDD_2 + AbsCUDD_1 = 1 \text{ \& } AbsCUDD_1 \geq 0.5 \\ || AbsCUDD_2 + AbsCUDD_1 > 1 \text{ \& } AbsCUDD_1 \leq 1.4375 \end{cases} \\ 4 & AbsCUDD_1 > 1.4375 \end{cases} \quad (4)$$

$CUDepthTypePole_1(AbsCUDD_d)$ for A2, B2, C2, and D2) is calculated as Formula (5):

$$CUDepthTypePole_1 = \begin{cases} 0 & AbsCUDD_0 + AbsCUDD_1 = 1 \\ 1 & \begin{cases} AbsCUDD_0 < 1.5 \text{ \& } AbsCUDD_1 + AbsCUDD_2 = 1 \\ || AbsCUDD_0 = 0 \end{cases} \\ 2 & AbsCUDD_0 > 1.5 \text{ \& } AbsCUDD_0 \leq 2 \\ 4 & AbsCUDD_0 > 2 \end{cases} \quad (5)$$

Table 3. Current CTU depth type values

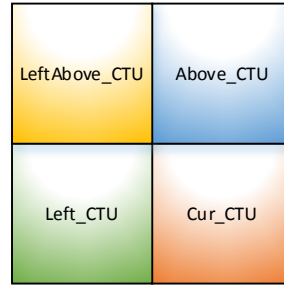
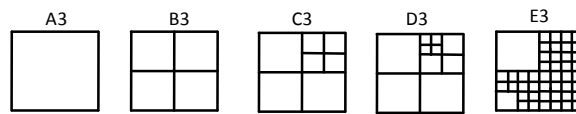
CurCUType	CTUDepthRange
0	[0,1]
1	[0,2]
2	[1,2]
3	[1,3]
4	[2,3]
5	[0,3]

Next, this study obtains $CurCUType$ by calculating the values of $CUDepthTypePole_0$ and $CUDepthTypePole_1$. We consider the following expressions.

$$CurCUType = \begin{cases} 0 & \begin{cases} CUDepthTypePole_0 + CUDepthTypePole_1 = 0 \\ || CUDepthTypePole_0 = 0 \text{ \& } CUDepthTypePole_1 = 1 \end{cases} \\ 1 & CUDepthTypePole_0 = 0 \text{ \& } CUDepthTypePole_1 = 2 \\ 2 & CUDepthTypePole_0 = 2 \text{ \& } CUDepthTypePole_1 = 1 \\ 3 & \begin{cases} CUDepthTypePole_0 = 3 || CUDepthTypePole_0 = 4 \\ \text{\& } CUDepthTypePole_1 \neq 4 \end{cases} \\ 4 & CUDepthTypePole_0 = 4 \text{ \& } CUDepthTypePole_1 = 4 \\ 5 & \text{none of the above conditions are satisfied} \end{cases} \quad (6)$$

Equatorial region:

The spherical content is mapped onto the 2D plane after ERP projection and the original video content is stretched. However, the equatorial region is less stretched than the pole region. That is, the video content in the equatorial region is less affected than that in the pole region. Therefore, to predict the depth interval of the current CTU [22], we need to refer to the left, above-left and above CTUs, as shown in Fig. 3.

**Fig. 3.** Equatorial region reference CTU**Fig. 4.** Classification of CTUs in the equatorial region**Table 4.** $AbsCUDD_d$ for A3, B3, C3, D3, and E3

$AbsCUDD_d$	A3	B3	C3	D3	E3
$AbsCUDD_0$	0	1	1.25	1.3125	2.4375
$AbsCUDD_1$	1	0	0.25	0.3125	1.4375
$AbsCUDD_2$	2	1	0.75	0.6875	0.5625
$AbsCUDD_3$	3	2	1.75	1.6875	1.5625

Table 4 shows the $AbsCUDD_d$ values corresponding to the division of A3, B3, C3, D3 and E3 in **Fig. 4**. A3, B3, C3, D3, and E3 are the CTU partitions in the equatorial region, as defined in this paper. $CUDepthType_i$ ($i = 1$ represents the left CTU, $i = 2$ represents the above CTU, and $i = 3$ represents the above-left CTU) is defined as the depth type values of adjacent CTUs. We obtain it by calculating $AbsCUDD_d$, as shown in Formula(7).

$$CUDepthType_i = \begin{cases} 0 & AbsCUDD_0 + AbsCUDD_1 = 1 \\ 2 & \begin{cases} AbsCUDD_2 + AbsCUDD_1 = 1 \\ \& AbsCUDD_1 \leq 0.25 \end{cases} \\ 3 & \begin{cases} AbsCUDD_2 + AbsCUDD_1 = 1 \\ \& AbsCUDD_1 \geq 0.3125 \parallel AbsCUDD_2 + AbsCUDD_1 > 1 \\ \& AbsCUDD_1 \leq 1.4375 \end{cases} \\ 4 & AbsCUDD_1 > 1.4375 \end{cases} \quad (7)$$

The value of $CurCUDepth$ is obtained by calculating of $CUDepthType_i$, as shown in Formula (8).

$$CurCUType = \begin{cases} 0 & \begin{cases} CUDepthType_0 + CUDepthType_1 + CUDepthType_2 = 0 \\ \parallel CUDepthType_0 + CUDepthType_1 + CUDepthType_2 = 2 \end{cases} \\ 1 & \begin{cases} CUDepthType_0 + CUDepthType_1 + CUDepthType_2 = 4 \& \\ CUDepthType_0 \neq 4 \& CUDepthType_1 \neq 4 \& CUDepthType_2 \neq 4 \end{cases} \\ 2 & \begin{cases} CUDepthType_0 + CUDepthType_1 + CUDepthType_2 = 6 \\ \& CUDepthType_0 \neq 3 \& CUDepthType_1 \neq 3 \& CUDepthType_2 \neq 3 \\ \& CUDepthType_0 \neq 4 \& CUDepthType_1 \neq 4 \& CUDepthType_2 \neq 4 \end{cases} \\ 3 & CUDepthType_0 + CUDepthType_1 + CUDepthType_2 = 9 \\ 4 & CUDepthType_0 + CUDepthType_1 + CUDepthType_2 = 12 \\ 5 & \text{none of the above conditions are satisfied} \end{cases} \quad (8)$$

We then obtain the prediction depth interval [23] of the current CTU by using *CurCUType* corresponding to Table 3.

The prediction of the CU depth range in this algorithm depends on spatial correlation. In the pole area, when the values of depth type (*CUDepthTypePole₀* and *CUDepthTypePole₁*) of the neighboring blocks are low, the complexity around the current CU is also low. The depth interval of the current CU is predicted to be [0,1]. If the left adjacent values of depth type (*CUDepthTypePole₀*) are not low and not high, we cannot directly predict the depth interval of the current CU. In this case, the depth type value (*CUDepthTypePole₁*) of the 64×32 block on the above side should be further determined for joint prediction. According to the different situations, the depth range of the current CU is determined to be [0,2] or [1,2] or [1,3]. When the depth type values (*CUDepthTypePole₀* and *CUDepthTypePole₁*) of the neighboring blocks are high, the complexity around CU is large. Therefore, the depth range of the current CU can be predicted as [2,3]. For the same reason, in the equatorial region, the complexity of the surrounding CU is also determined according to the depth type values (*CUDepthType_d*) of the neighboring blocks to predict the depth interval of the current CU.

Supplementary explanation: The weights of the reference neighboring blocks in the equatorial region show little difference because of reduced stretching after ERP projection. That is, we should consider the values of three adjacent *CUDepthType_d* simultaneously to predict the depth interval of current CU. In the pole area, the weight of the neighboring block on the left side is much greater than that on the other side due to severe stretching. The prediction of the depth interval of the current CU depends largely on the *CUDepthTypePole₀* of the left CU. Auxiliary judgment is performed on *CUDepthTypePole₁* of the above 64×32 block to jointly predict the depth interval of the current CU.

2. 2 Intra mode fast decision

The conventional RMD process needs to traverse 35 modes to obtain few candidate modes for full RD cost computation. All intra prediction modes are sorted by the calculated SATD values and we select the number of modes in the candidate list according to the PU size. For example, when the PU sizes are 64×64 , 32×32 , 16×16 , 8×8 , and 4×4 , the numbers of the candidate modes are 3, 3, 3, 8, and 8, respectively. This detailed calculation results in a large amount of complexity, especially for 360-degree videos with high frame rate and high resolution. To reduce computational complexity, this study optimizes the pole and equatorial regions by adaptively reducing the number of primary and candidate modes.

The video content in the pole area is severely stretched because of the characteristics of the ERP projection format. Hence, considerable data redundancy exists in the horizontal direction, and the optimal angle mode is likely to belong to the horizontal mode [corresponding to angle modes 2 to 18 (these numbers correspond to the index of the angle prediction mode)]. In addition, the optimal mode of a large-sized PU [24] is typically fixed at the predict mode (such as modes 0, 1, 10, and 26), but the optimal mode of a small-sized PU changes considerably.

By analyzing the characteristics of the ERP format, this study first reduces the number of primary modes required for the first RMD process [25, 26] and obtains a candidate list. Then, we judge the types of the first and second modes in the candidate list and deal with them separately. Next, we place four angle modes adjacent to the first or second mode with reduced primary modes that belong to the first RMD [27] process to perform the second RMD process, which further reduces the number of modes that required to implement **Rate Distortion Optimization (RDO)**. The algorithm can effectively reduce coding complexity. The detailed process is as follow:

Pole area: First, the interval of the primary mode is given according to the depth of the PU. By definition, when the depth of the PU is 0 and 1, the interval is 8. When the depth of the PU is 2 and 3, the interval is 8. When the depth of the PU is 4, the interval is 4, as shown in [Table 5](#).

Table 5. Original reduced primary modes in pole area

First RMD	
reduced primary modes	PU Depth
$\{0, 1, 2, 10, 18, 26, 34\}$	$[0, 1]$
$\{0, 1, 2, 10, 18, 26, 34\}$	$[2, 3]$
$\{0, 1, 2, 6, 10, 14, 18, 22, 26, 30, 34\}$	$[4]$

We consider the serious horizontal stretching in the pole region, that is, the probability that the horizontal mode becomes the optimal mode is high. To more accurately predict the angle model accurately, we further amend to the above interval, as shown in [Table 6](#) and [Fig. 5](#):

Table 6. Improved reduced primary modes in pole area

First RMD	
reduced primary modes	PU Depth
$\{0, 1, 2, 6, 10, 14, 18, 26, 34\}$	$[0, 1]$
$\{0, 1, 2, 6, 10, 14, 18, 26, 34\}$	$[2, 3]$
$\{0, 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 22, 26, 30, 34\}$	$[4]$

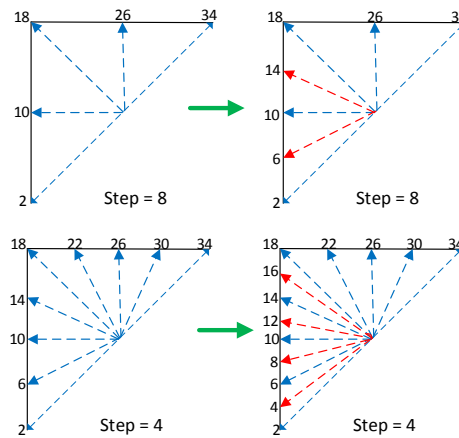


Fig. 5. Improved primary mode list

That is, in the interval specified by the depth of PU, we add several modes to the interval that belongs to modes 2 to 18 (modes 2 to 18 are horizontal modes). For example, when the current depth of PU is 0 and 1, the original traversal mode list is $\{0, 1, 2, 10, 18, 26, 34\}$. Considering that the current CTU is in the pole area, the horizontal stretching is severe, that is, the possibility that the horizontal prediction mode becomes the optimal prediction mode is very high. Therefore, in prediction modes 2 to 18, interval 8 is changed to 4, and in prediction modes 19 to 34, interval 8 remains unchanged. The changed traversal mode list is $\{0, 1, 2, 6, 10, 14, 18, 26, 34\}$. Similarly, when the depth of PU is 2 or 3, we change the traversal mode list to be $\{0, 1, 2, 6, 10, 14, 18, 26, 34\}$. When the current depth of PU is 4, the original traversal mode list is modified as $\{0, 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 22, 26, 30, 34\}$.

Equatorial region: The equatorial region exhibits reduced stretching after ERP projection format mapping. Therefore, this article posits that when the depth of PU is 0 and 1, the traversal range of the initial selection mode is 8. When the depth of PU is 2 and 3, the traversal interval of the primary selection mode is 4. When the depth of PU is 4, the traversal region of the original selection mode is 2, as shown in **Table 7** and **Fig. 6**:

Table 7. Improved reduced primary modes in equatorial area

First RMD	
reduced primary modes	PU Depth
$\{0, 1, 2, 10, 18, 26, 34\}$	$[0, 1]$
$\{0, 1, 2, 6, 10, 14, 18, 22, 26, 30, 34\}$	$[2, 3]$
$\{0, 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34\}$	$[4]$

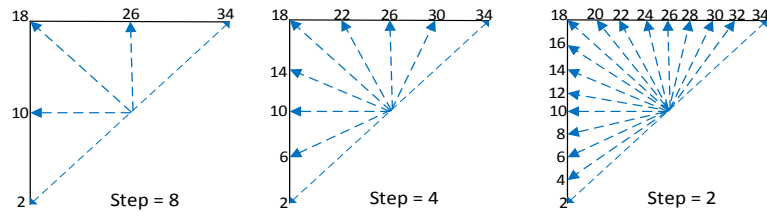


Fig. 6. Prediction mode traversal range in the equatorial region

We obtain the candidate list after the first RMD process [28]. We perform the second RMD process on the basis of the modes within the obtained candidate list, thereby reducing the number of modes that are required implement RDO. We define the obtained primary candidate list as *MyMode*. The first mode in *MyMode* is called FM, and the second mode is called SM. The detailed algorithms are as follows:

(1) PU sizes of 16×16 , 32×32 , and 64×64 : This study first distinguishes FM in *MyMode*. When FM is DC, PLANAR, or VDM (VDM is mode 26), we retain the three modes in *MyMode*. If FM is an angular mode, the four modes adjacent to FM (FM-1, FM-2, FM+1, and FM+2) are added to a list with reduced primary modes in the first RMD process for performing the second RMD process. Additionally, the four angle modes mentioned above are adjacent to FM in 35 angle prediction modes. For example, if FM is angle mode 6, then FM-1 corresponds to angle mode 5, FM-2 corresponds to angle mode 4, and then FM+1 and FM+2 can be obtained by analogy. The first two modes are used as the best prediction modes [29] in the final list that is obtained after the second RMD process. Compared with the original HEVC framework, this current framework has one less prediction mode, that is, from the original three candidate modes to two candidate modes. For example, when the FM in *MyMode* is angular mode 6 after the first RMD process, the four modes adjacent to 6 are 4, 5, 7, and 8. Then, these modes are added to a list with reduced primary modes to perform the second RMD process. Lastly, we use the first two modes in the final list as the prediction modes of the PU.

(2) PU sizes of 8×8 and 4×4 : After the first RMD process, we determine whether FM and SM are DC or PLANAR in *MyMode*. In this case, this algorithm choose the first two modes of *MyMode* as the best prediction modes. When FM or SM in *MyMode* is an angular mode, adding four modes adjacent to the angular mode (FM or SM) with reduced primary modes in a list for the second RMD process. We obtain the final list after the second RMD process and take the first two modes as the prediction modes, that is, from the original eight candidate modes to two candidate modes. Notably, when the angular mode numbers are 2 or 34, only two modes are adjacent to them. **Fig. 7** illustrates the algorithm flow chart.

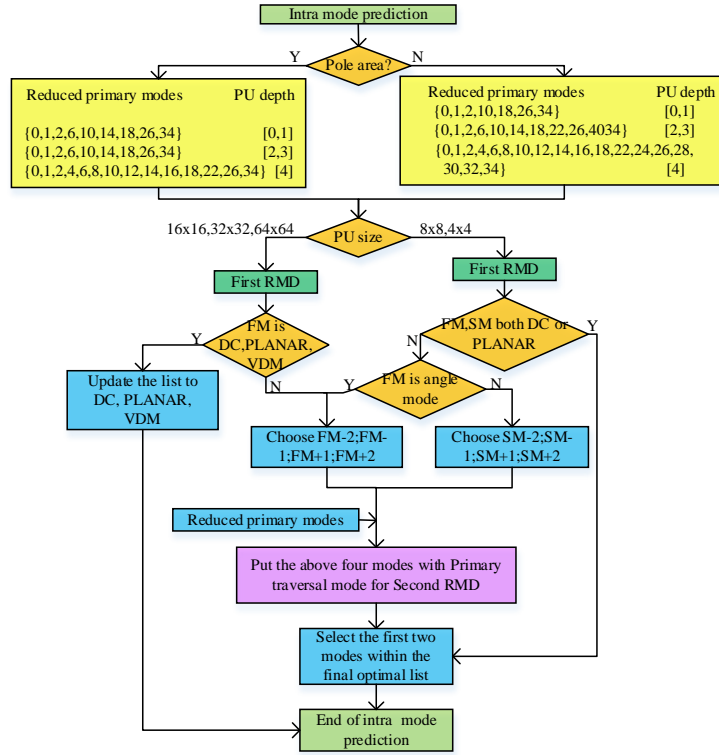


Fig. 7. Intra mode prediction fast decision flow chart

3 Analysis of experiment result

To verify the feasibility of the proposed compression algorithm, we use the algorithm to test the rate distortion performance and coding complexity in HM-16.16-360Lib-4.0. The experimental platform hardware is an Intel Core i7-7700 CPU with 3.60GHz processing speed and 8 GB of RAM. The coding parameter of the experiment is encoder_intra_main10. The number of coding frames is 100, and the QPs are 22, 27, 32 and 37 respectively. To measure the rate distortion performance of the algorithm, we use BD-rate to represent the bit rate variation under the same image quality condition. The symbol ΔT is adopted to measure time saved, and WS-PSNR evaluates the distortion by weights given offline. The calculation processes of WS-PSNR are as follows:

the weights of distortion in ERP is shown in Formula (9):

$$W(i,j) = \frac{w(i,j)}{\sum_{i=0}^{width-1} \sum_{j=0}^{height-1} w(i,j)}, \quad (9)$$

where $width$ and $height$ are the size of images. $w(i,j)$ is the scaling factor of area from equirectangular to sphere which can be represented as:

$$w(i,j) = \cos\left(\left(j - \frac{height}{2} + \frac{1}{2}\right) \cdot \frac{\pi}{height}\right). \quad (10)$$

WS-PSNR is obtained using:

$$WS-PSNR = 10 \log\left(\frac{MAX^2}{WMSE}\right), \quad (11)$$

$$WMSE = \sum_{i=0}^{width-1} \sum_{j=0}^{height-1} (y(i,j) - y'(i,j))^2 \cdot W(i,j), \quad (12)$$

where $W(i,j)$ is calculated in Formula (9), $y(i,j)$, $y'(i,j)$ are the original pixel value and reconstruct pixel value and MAX is the max pixel value in the image. Time reduction is calculated with Formula (13), where $T_{HM16.16}$ is the coding time of HM-16.16, $T_{proposed}$ is the coding time of the proposed algorithm, and ΔT is the time reduction. The change in $WS-PSNR_Y$ is calculated with Formula (14).

$$\Delta T = \frac{T_{HM16.16} - T_{proposed}}{T_{HM16.16}} \times 100\% \quad (13)$$

$$\Delta WS-PSNR_Y = WS-PSNR_{HM16.16} - WS-PSNR_{proposed} \quad (14)$$

In this work, 12 standard test sequences are proposed using the proposals of JVET-D0026, JVET-D0039, JVET-D0053, JVET-G0147, JVET-D0143 and JVET-D0179. Prior to encoding, the test sequences are converted to low-resolution ERP for encoding (for accuracy quality assessment). For 8K and 6K ERP videos, the encoding size is set to 4096×2048 , and for 4K ERP videos, the encoding size is set to 3328×1664 .

In general, the test sequences are reduced at varying percentages of time because of different textures and contents. The experimental results in [Table 8](#) show that compared with the standard algorithm, the algorithm in this work reduces the average time by 39.3%, and the BD-rate only increases by 0.84%. WS-PSNR decreases by an average of 0.044. We obtained these results because our algorithm divides the regions into pole and equatorial according to the distribution location of CTUs. In the CU division stage, when the current CTU is in the pole area, only the left neighboring CTU and the above 64×32 size adjacent block are referenced. The depth interval of the current CTU is predicted according to $AbsCUDD_d$ and $CUDepthTypePole_i$, as defined in this work; When the current CTU is in the equatorial region, we should refer to the left, above-left, and above three neighboring CTUs to determine the depth range of the current CTU based on $AbsCUDD_d$ and $CUDepthType_e$ defined in this work. In intra mode fast decision, we first determine whether the current CTU is located in the pole region or equatorial region. When it is in the pole region, the possibility that the optimal prediction mode belongs to horizontal modes is high. Therefore, in the reduced primary modes defined by PU depth, the number of traversal horizontal modes increases, that is, the interval of traversal horizontal mode is narrowed down. When the current CTU is in the equatorial region, we define the prediction mode interval according to PU depth. Then we perform the first RMD and obtain the candidate list. We place the four modes that are adjacent to FM or SM in the candidate list with the reduced primary modes for the second RMD process. Finally, the first two modes are selected as the best modes in the final candidate list, which can reduce 1 or 5 modes for full RD cost computation. This algorithm can guarantee video quality and further reduce the complexity.

The algorithm reduces much time in the Landing2, Balboa, Trolley, Harbor, and SkateboardInLot sequences mainly because the textures of these backgrounds are relatively simple and may skip the coding of unnecessarily small CUs. However, KiteFlite, BranCastle2 and ChairliftRide sequences have less time reduction because the textures of these test sequences are complex and likely to be divided into small CUs, that is, their average depth is larger than that of the other sequences. Therefore, the early termination algorithm limits their

improvement in reducing the encoding time. **Fig. 8** shows a comparison of BD-rate for four sequences, namely, AerialCity, Landing2, Gaslamp, and Harbor. The BD-rate performance of the test sequences remains at almost the same level. **Fig. 9** presents a comparison of CU partitions for the four sequences. We conclude that the algorithm partition error is small. **Table 8** shows the experimental data of this algorithm. **Fig. 10** shows the accuracy of the algorithm's CTU division. Overall, the proposed algorithm significantly improves coding efficiency while maintaining the performance of the BD-rate at an almost similar level.

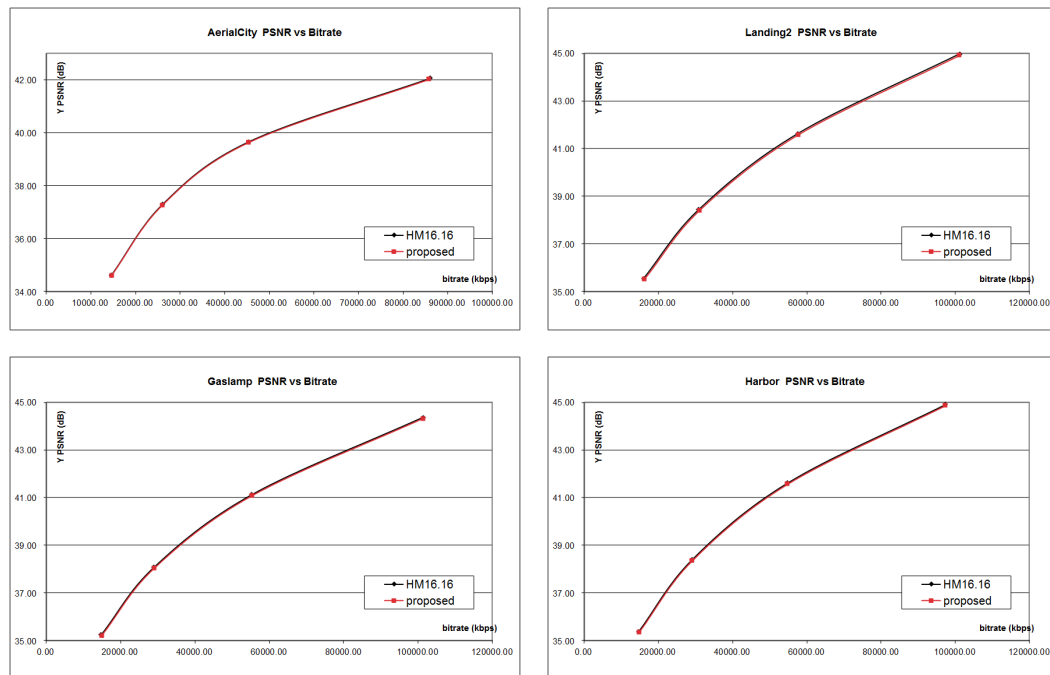


Fig. 8. Comparison of the BD-rate of test sequences

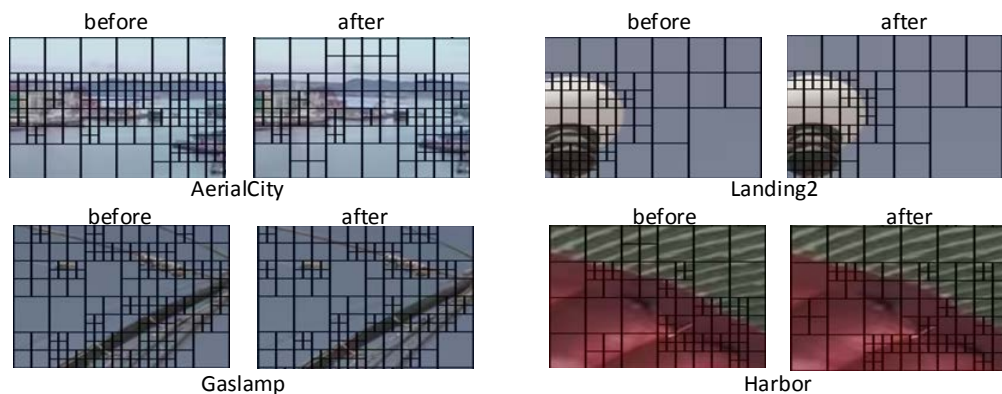
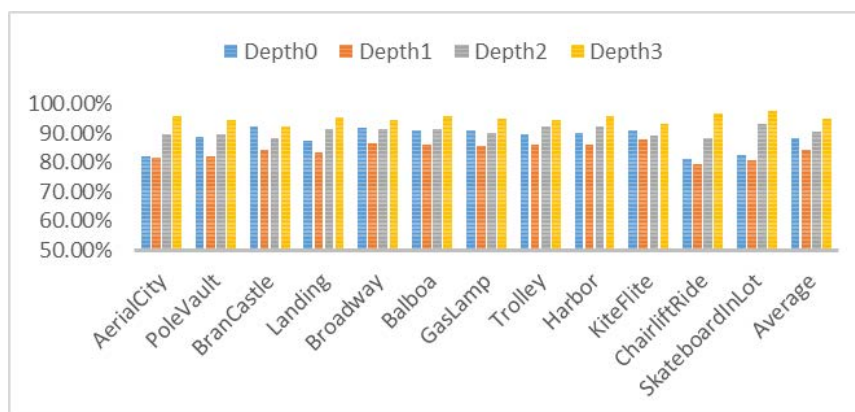


Fig. 9. Comparison of CU partitions (algorithm and HM-16.16)

Table 8. Experimental data

Class	Sequence	BD-rate	$\Delta T\%$	$\Delta WS-PSNR_V$
4K	AerialCity	0.70%	40.18%	-0.026
	PoleVault	0.70%	38.56%	-0.054
6K	BranCastle2	0.60%	35.89%	-0.064
	Landing2	1.10%	42.67%	-0.04
	Broadway	1.00%	39.20%	-0.038
	Balboa	0.80%	42.77%	-0.037
8K	Gaslamp	1.40%	41.17%	-0.043
	Trolley	0.50%	41.78%	-0.047
	Harbor	1.10%	41.65%	-0.045
	KiteFlite	0.70%	34.12%	-0.059
	ChairliftRide	0.70%	33.76%	-0.026
	SkateboardInLot	0.80%	39.81%	-0.044
	Average	0.84%	39.30%	-0.044

**Fig. 10.** Accuracy of CU division (compare with HM-16.16)

4. Conclusion

This paper proposes a novel intra fast algorithm for fast CU partition and mode decision based on the characteristics of ERP format of 360-degree videos. For fast CU partition, the video content is divided into pole and the equatorial areas, and different neighboring blocks and determination conditions are set for each CTU based on the area it located. For fast intra mode decision, reduced primary modes instead of 35 modes are used in the proposed algorithm to reduce the number of traversal modes. Considering the effect of lateral stretching, the traversal interval is further optimized by adding horizontal modes in prediction modes 2 to 18. Furthermore, two RMD processes are used to improve the accuracy of obtaining the optimal mode. Compared with the original HM-16.16, the proposed algorithm can improve the efficiency of intra coding, The total encoding time is reduced by 39.3%, and the BD-rate increases by only 0.84%. Therefore, this algorithm is applicable to virtual reality video coding.

5. Acknowledge

This work is supported by the National Natural Science Foundation of China (No.61370111), Beijing Municipal Natural Science Foundation (No.4172020), Great Wall Scholar Project of Beijing Municipal Education Commission (CIT&TCD20180304), Beijing Youth Talent Project (CIT&TCD 201504001), and Beijing Municipal Education Commission General Program (KM201610009003).

References

- [1] E. Alshina, J. Boyce, A. Abbas, Y. Ye (editors), "JVET common test conditions and evaluation procedures for 360° video," *Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 JVET-H1030 Macau*, 2017.
- [2] Sullivan G J, Ohm J R, Han W J, et al., "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits & Systems for Video Technology*, 22(12), 1649-1668, 2012. [Article \(CrossRef Link\)](#).
- [3] Ohm J R, Sullivan G J, Tan T K, et al., "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits & Systems for Video Technology*, 22(12), 1669-1684, 2013. [Article \(CrossRef Link\)](#).
- [4] Shen L, Zhang Z, Liu Z, "Effective CU size decision for HEVC intracoding," *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 23(10), 4232-4241, 2014. [Article \(CrossRef Link\)](#).
- [5] Shen L, Zhang Z, An P, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Transactions on Consumer Electronics*, 59(1), 207-213, 2013. [Article \(CrossRef Link\)](#).
- [6] Shen L, Liu Z, Zhang X, et al., "An Effective CU Size Decision Method for HEVC Encoders," *IEEE Transactions on Multimedia*, 15(2), 465-470, 2013. [Article \(CrossRef Link\)](#).
- [7] Kim B G, "Fast coding unit (CU) determination algorithm for high-efficiency video coding (HEVC) in smart surveillance application," *Kluwer Academic Publishers*, 2017. [Article \(CrossRef Link\)](#).
- [8] Lee J, Kim S, Lim K, et al., "A Fast CU Size Decision Algorithm for HEVC," *Circuits & Systems for Video Technology IEEE Transactions on*, 25(3), 411-421, 2015. [Article \(CrossRef Link\)](#).
- [9] Min B, Cheung R C C, "A Fast CU Size Decision Algorithm for the HEVC Intra Encoder," *IEEE Transactions on Circuits & Systems for Video Technology*, 25(5), 892-896, 2015. [Article \(CrossRef Link\)](#).
- [10] Cho S, Kim M, "Fast CU Splitting and Pruning for Suboptimal CU Partitioning in HEVC Intra Coding," *IEEE Transactions on Circuits & Systems for Video Technology*, 23(9), 1555-1564, 2013. [Article \(CrossRef Link\)](#).
- [11] Zhang H, Ma Z, "Fast Intra Mode Decision for High Efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits & Systems for Video Technology*, 24(4), 660-668, 2014. [Article \(CrossRef Link\)](#).
- [12] Zhang M, Zhao C, Xu J, "An adaptive fast intra mode decision in HEVC," in *Proc. of IEEE International Conference on Image Processing. IEEE*, 221-224, 2012. [Article \(CrossRef Link\)](#).
- [13] Zhu S, Zhang C, "A fast algorithm of intra prediction modes pruning for HEVC based on decision trees and a new three-step search," *Multimedia Tools & Applications*, 76(20), 21707–21728, 2017. [Article \(CrossRef Link\)](#).
- [14] Liu X, Liu Y, Wang P, et al., "An Adaptive Mode Decision Algorithm Based on Video Texture Characteristics for HEVC Intra Prediction," *IEEE Transactions on Circuits & Systems for Video Technology*, 27(8), 1737-1748, 2017. [Article \(CrossRef Link\)](#).
- [15] Zhang M, Bai H, Lin C, et al., "Texture Characteristics Based Fast Coding Unit Partition in HEVC Intra Coding," in *Proc. of Data Compression Conference. IEEE*, 477-477, 2015. [Article \(CrossRef Link\)](#).

- [16] Zhang M, Zhai X, Liu Z, "Fast and adaptive mode decision and CU partition early termination algorithm for intra-prediction in HEVC," *Eurasip Journal on Image & Video Processing*, 2017(1), 86, 2017. [Article \(CrossRef Link\)](#).
- [17] Gu J, Tang M, Wen J, et al., "Adaptive Intra Candidate Selection with Early Depth Decision for Fast Intra Prediction In HEVC," *IEEE Signal Processing Letters*, 25(2), 159-163, 2018. [Article \(CrossRef Link\)](#).
- [18] Yang M, Grecos C, "Fast intra encoding decisions for high efficiency video coding standard," *Journal of Real-Time Image Processing*, 13(4), 797-806, 2017. [Article \(CrossRef Link\)](#).
- [19] Lee D, Jeong J, "Fast intra coding unit decision for high efficiency video coding based on statistical information," *Signal Processing Image Communication*, 55, 121-129, 2017. [Article \(CrossRef Link\)](#).
- [20] Kim J, Choe Y, Kim Y G, "Fast Coding Unit size decision algorithm for intra coding in HEVC," in *Proc. of IEEE International Conference on Consumer Electronics. IEEE*, 637-638, 2013. [Article \(CrossRef Link\)](#).
- [21] Nishikori T, Nakamura T, Yoshitome T, et al., "A fast CU decision using image variance in HEVC intra coding," *Industrial Electronics and Applications. IEEE*, 52-56, 2013. [Article \(CrossRef Link\)](#).
- [22] Goswami K, Kim B G, Jun D, et al., "Early Coding Unit-Splitting Termination Algorithm for High Efficiency Video Coding (HEVC)," *Etri Journal*, 36(3), 407-417, 2014. [Article \(CrossRef Link\)](#).
- [23] Bai C, Yuan C, "Fast coding tree unit decision for HEVC intra coding," in *Proc. of Icce-China Workshop. IEEE*, 28-31, 2013. [Article \(CrossRef Link\)](#).
- [24] Ruiz D, Fernández-Escribano G, Martínez J L, et al., "Fast intra mode decision algorithm based on texture orientation detection in HEVC," *Signal Processing Image Communication*, 44(C), 12-28, 2016. [Article \(CrossRef Link\)](#).
- [25] Zhang T, Sun M T, Zhao D, et al., "Fast Intra Mode and CU Size Decision for HEVC," *IEEE Transactions on Circuits & Systems for Video Technology*, 27(8), 1714-1726, 2017. [Article \(CrossRef Link\)](#).
- [26] Silva T L D, Agostini L V, Cruz L A D S, "Fast HEVC intra prediction mode decision based on EDGE direction information," in *Proc. of Signal Processing Conference. IEEE*, 1214-1218, 2012. [Article \(CrossRef Link\)](#).
- [27] Chen Z Y, Chang P C, "Rough mode cost-based fast intra coding for high-efficiency video coding," *Journal of Visual Communication & Image Representation*, 43, 77-88, 2016. [Article \(CrossRef Link\)](#).
- [28] Wang T, Men Y, Zhang Y, et al., "A fast intra-prediction decision algorithm in inter-frame based on a novel feature of HEVC," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE*, 1532-1536, 2017. [Article \(CrossRef Link\)](#).
- [29] Tseng C F, Lai Y T, "Fast coding unit decision and mode selection for intra-frame coding in high-efficiency video coding," *Iet Image Processing*, 10(3), 215-221, 2016. [Article \(CrossRef Link\)](#).



Mengmeng Zhang received the B.S. degree in electronic information technology and M.S. degree in signal and information processing from Beijing Jiaotong University, China in 2000 and 2003 respectively, and the Ph.D. in communication and information systems from University of Science & Technology Beijing, China in 2012. Currently, he is a professor in North China University of Technology.



Zhi Liu received the B.S. degree in electronic information technology and the Ph.D. in signal and information processing from Beijing Jiaotong University, China in 2001 and 2011 respectively. Currently, he is a lecturer in North China University of Technology.



Wen Yu received the B.S. degree in mechanical design and manufacturing and automation and the M.S. the Ph.D. in geological engineering from China university of geosciences, Beijing in 2006 and 2009 respectively. he is a professor in China university of geosciences, Beijing.



Jing Zhang received a bachelor degree in communications engineering from Huaiyin Institute of Technology in 2016 and studies in North China University of Technology for postgraduate in 2016. Currently, he is a postgraduate in North China University of Technology.



Fuqi Mao received his B.S. degree in computer science and technology and M.S. degree in computer application technology from North China University of Technology. Currently, he is a research assistant at North China University of Technology