

An efficient coding algorithm for 360-degree video based on improved adaptive QP Compensation and early CU partition termination

Mengmeng Zhang¹ · Jing Zhang¹ · Zhi Liu¹ · Changzhi An²

Received: 11 March 2018 / Revised: 4 May 2018 / Accepted: 15 June 2018 /
Published online: 10 July 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract Virtual reality technology enables people to experience the video content immersively. In order to provide realistic presence and dynamic view, the virtual reality video requires higher resolution (4 K or 8 K) image and more data to display relative to traditional video. Therefore, to improve the coding efficiency of 360-degree video becomes a key consideration. In the coding of 360-degree video, the spherical image is projected to a 2D image (such as ERP projection) and the standard video coding framework is utilized to accomplish the rest work. However, such a projection introduces much different levels of distortion according to coordinates, which degrades performance of rate distortion optimization in video encoding process. In this paper, we propose a compression optimization algorithm by using adaptive QP compensation based on coordinates to improve compression efficiency, and utilizing early termination of CU partition based on spatial correlation to reduce encoding time. To further reduce encoding complexity, the prewitt operator and the adaptive mode selection are adopted to reduce unnecessary intra prediction modes. Experimental results show that compared with HM-16.16, the proposed algorithm can reduce time by 42.4%, increase WSPSNR by 0.03 and decrease BD-rate by 0.3%.

✉ Mengmeng Zhang
muchmeng@126.com

Jing Zhang
381889962@qq.com

Zhi Liu
lzliu@ncut.edu.cn

Changzhi An
changzhi.an@ceict.com.cn

¹ North China University of Technology, Beijing, China

² Beijing China Electronic Intelligent, Communication Technology Co., Ltd., Beijing, China

Keywords 360-degree video encoding · CU partition · Mode selection · Adaptive QP compensation

1 Introduction

In VR system, multiple cameras capture 360-degree real-world scene and then use image rendering to restore the scene. After the collected spherical information has been rendered, people can enjoy the video immersively by using the head-mounted display. Therefore, the high frame rate and high resolution required by 360-degree video, which brings great challenges to storage capacity and transmission bandwidth. Therefore 360-degree video needs a more efficient compression algorithm. In order to handle and compress the contents of these spherical regions, we need to map the spherical content information into a 2D image before encoding. A variety of mapping methods have been proposed at JVET conferences, such as ERP projection, CMP projection [2], SSP projection, ECP projection and so on. Projected content can be handled by using a standard video coding framework such as HEVC / H.265 [18, 22]. However, projected image introduces much different levels of distortion according to coordinates [24], which makes the rate-distortion optimization performance in video coding much inefficiently.

In order to solve this problem, people use some solutions such as adaptive QP compensation to improve the efficiency of coding. Spherical images have uneven sampling characteristics after being projected onto a 2D image, such as ERP projection format. By analyzing the characteristics of ERP projection, it has denser sampling at pole areas, which results encoder spends more bits in those regions for constant QP configuration. This typically results in quality of viewports of the pole areas being higher than the quality of viewports in the equator area. Therefore, the adaptive QP encoding approach applies coarser QP to blocks that are closer to the pole areas as the pole areas have higher sampling density due to the projection transformation. The non-uniform sampling property of projection schemes for 360-degree videos can be exploited to improve compression efficiency of 360-degree videos by modulating the distortion according to the sampling density.

For reducing the coding time and complexity, people make use of CU partition and intra mode selection. In HEVC, three coding units are used in coding. These are Coding Unit (CU), Prediction Unit (PU) and Transform Unit (TU). Relative to H.264 fixed-size macroblocks, HEVC provides CU unit sizes from 64×64 to 8×8 , with depths corresponding from 0 to 3, respectively. As shown in Fig. 1, the splitting and pruning of quadtree structure of CTU is as follows.

Intra mode selection: The first step is to perform coarse selection, that is to calculate the low complexity rate distortion cost of 35 types (DC, PLANAR and 33 kinds of angular modes) and select a set of the most possible candidates to fill in the initial list, the number of candidates is depend on the PU's size as determined in Table 1. It then determines whether the most probable mode (MPM) of its neighboring blocks is in the initial list and adds it if not. Then, we calculate RDO for the modes in the list in turn, and finally get prediction modes.

2 Related work on fast hevc encoding

Currently concerning intra encoding compression algorithms are mainly composed of CU fast partition and PU mode selection. In CU fast partition, the optimization is mainly made from premature termination of CU partition based on prediction of depth range. In PU mode

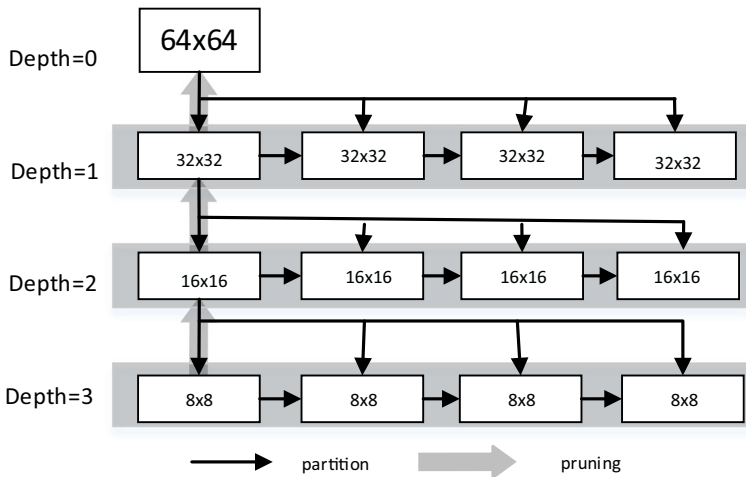


Fig. 1 the partition and pruning of CTU

selection, the optimization is mainly performed in 35 prediction modes and MPM modes. As for the early termination of CU partition and depth skip, Shen et al. [21] 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. [20] proposed a fast CU size decision algorithm to determine CU depth range (including the minimum depth level and the maximum depth level) and skip some specific depth levels rarely used in the previous frame and neighboring CUs. Kim et al. [11] proposed an efficient CU determination algorithm to use spatial and temporal information in which 13 neighboring coding tree units (CTUs) are defined. Lee et al. [13] 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. [16] proposed a novel fast algorithm for the CU size decision in intra coding. Both the global and local edge complexities in horizontal, vertical, 45 degrees diagonal, and 135 degrees diagonal directions are proposed and used to decide the partitioning of a CU. Cho et al. [6] 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. [28] proposed the Hadamard cost-based progressive rough mode search to selectively check the potential modes instead of traversing all candidates. Zhang et al. [29] 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. [32] proposed a novel intra prediction modes pruning method based on decision trees and a new three-step search algorithm, aiming

Table 1 num of PU candidate

Size of Pu	Num of candidates
64 × 64	3
32 × 32	3
16 × 16	3
8 × 8	8
4 × 4	8

at achieving higher encoding efficiency compared to the standard-HEVC. Liu et al. [15] proposed an adaptive mode decision algorithm based on texture complexity and direction for HEVC intra prediction. Zhang et al. [30] 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 were mainly used for natural sequence and didn't take into account the characteristic of 360-degree video, which has higher resolution and uneven sampling distribution after being projected onto a 2D image.

This paper proposes three methods to improve the intra coding compression performance. First, we use adaptive QP compensation [23] based on the y coordinate of the pixel to improve compression efficiency of 360-degree video. Second, we apply premature termination of CU partition based on spatial correlation to reduce encoding time. Then we expand the pole areas of judgement condition based on the weight of WS-PSNR in ERP projection to further speed up encoding process. Third, in order to further decrease coding complexity, we make use of prewitt operator to predict mode directly and reduce unnecessary prediction modes through RDO cost analysis.

3 Proposed algorithm

In this paper, first of all, according to the uneven sampling characteristic of ERP projection, adaptive QP compensation is basically performed at pole areas with severe stretching, modulating the distortion and improving compression efficiency of 360-degree video. Then, in order to decrease encoding time, the premature termination of CU partition based on the prediction of depth range algorithm [9, 27] is proposed, which can skip unnecessary depth traversal. In addition, considering the weight near pole areas is very small in ERP projection, which means the distortion of pole areas has very little impact on the whole under the assessment of WS-PSNR. Therefore we can expand the judgement conditions of these regions to further decrease the coding time. Finally, to further reduce encoding complexity, we handle separately according to the size of PU block in 35 kinds of intra prediction modes [12]. For example, 4×4 PU block uses prewitt operator to predict mode directly and 8×8 , 16×16 , 32×32 and 64×64 PU block decrease encoding complexity by adaptive mode selection in RMD process.

3.1 Improved adaptive QP compensation

The projection of 360-degree video has a characteristic of uneven sampling in 2D image. For example, 360-degree video with equal rectangular (ERP) projection has more dense samples at pole areas, which spends more bits in those regions for constant QP configuration. Therefore, the adaptive QP encoding approach applies coarser QP to blocks that are closer to the pole areas as the pole areas have higher sampling density due to the projection transformation.

In addition, in order to control the video quality, adaptive QP can not be too coarse for optimization of the pole areas. In this paper, by analyzing WS-PSNR weight w , we find it is suitable for ERP projection scheme and we can make further change based on it. Therefore, we modify the QP value by updating the following formula (1):

$$QP_{new} = QP - 3 \times \log_2(w_{new}) \quad (1)$$

Compared with the existing methods [14], the QP in this paper will decrease in the equatorial region and increase in the pole areas instead of simply increasing the QP for reducing bits that may sacrifice the video quality. Because the scope of the newly defined variable w_{new} is from 0.09478 to 1.56698 that corresponds from the pole to the equator. That is to say, when in the pole areas, the variable $\log_2(w_{new})$ is less than 0 and makes the QP_{new} greater than the QP , when in the equatorial areas, the variable $\log_2(w_{new})$ is greater than 0 and makes the QP_{new} less than the QP . That is why this algorithm can divide the equator precisely and partition the pole areas roughly respectively.

The weight used by WS-PSNR is as follows:

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

where N is the height of a CTU and j represents the y coordinate of the pixel in a frame (j is from 0 to the frame height). This algorithm is based on the weight of WS-PSNR and is calculated in each CTU. The calculation process is as follows:

- (1) Define `Num_of_Ctu_Height` as the quantity of CTUs in a column and calculate each CTU the sum of its weights and take the average:

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

Among them, the index is the serial number of a CTU in a column and its range is $[0, \text{Num_of_Ctu_Height}]$ and N represents the height of a CTU.

- (2) Calculate the sum of the average of each CTU weights in a column in the first step and define it as:

$$Total_Ctu_Average_weight = \sum_0^{Num_of_Ctu_Height} w_{Index_of_Ctu_Average_Weight}(i, j) \quad (4)$$

- (3) This paper modifies the weight as follow:

$$w_{new} = \frac{w_{Index_of_Ctu_Average_Weight}}{Total_Ctu_Average_weight} \times Num_of_Ctu_Height \quad (5)$$

$$QP_{new} = QP - 3 \times \log_2(w_{new}) \quad (6)$$

3.2 Early termination of cu partition based on spatial correlation

There is a strong correlation between pixels of each frame in the video and between coding units of each CTU level. Therefore, the average depth of adjacent CTUs (upper left CTU, upper CTU, left CTU) can be used to predict the current CTU depth range based on spatial correlation. The flowchart is shown in Fig. 2. DR value corresponds to the depth interval according to Table 2.

First, we classify the CTU partitions [8, 17]. As shown in Fig. 3, the value above each CTU partition [4] is the average depth of each CTU and the range to the left of each row includes all the depths of CTU partition in that row. When the average depth of the CTU is less than or equal to 1, the prediction depth range of CTU partition is [0, 1]. When the average depth of the CTU is equal to 1.25, 1.5 or 1.75, the prediction depth range of CTU partition is [1, 2]. When the average depth of the CTU is greater or equal to 1.3125 and less than or equal to 2.4375 that removes two depths 1.5 and 1.75, the prediction depth range of CTU partition is [1, 3]. When the average depth of the CTU is greater to 2.0625, the prediction depth range of CTU partition is [2, 3]. For example, if the average depth of the CTU is greater to 2.0625, the CTU partition may only has depth 2 and 3 except for depth 0 and 1.

According to the Table 2, the prediction depth range of CTU partitions are classified into four categories, these are T1, T2, T3 and T4 respectively. Each categorie includes all the depths in corresponding depth range. For example, if the CTU prediction depth category is belong to T2, the CTU partition has both depth 1 and 2. When the prediction depth range of the CTU is [0, 1], the prediction depth category is belong to T1. When the prediction depth range of the CTU is [1, 2], the prediction depth category is belong to T2. When the prediction depth range of the CTU is [1, 3], the prediction depth category is belong to T3. When the prediction depth range of the CTU is [2, 3], the prediction depth category is belong to T4. Among them, when prediction depth categories are T3 and T4, their prediction depth ranges are overlapping. For splitting accurately, we define the prediction depth category to be T3 when the average depth of CTU range is [2.0625, 2.4375] and determine it to be T4 when the average depth of CTU range is greater than 2.4375, as shown in formula (7).

$$\left\{ \begin{array}{ll} \text{Prediction Depth Range} & \text{Average Depth of CTU} \\ \text{Depth : 0, 1} & \text{Depth} \leq 1 \\ \text{Depth : 1, 2} & \text{Depth} = 1.25/1.5/1.75 \\ \text{Depth : 1, 2, 3} & 1.3125 \leq \text{Depth} \leq 2.4375 (\text{Depth} \neq 1.5/1.75) \\ \text{Depth : 2, 3} & \text{Depth} > 2.4375 \end{array} \right. \quad (7)$$

Taking into account the non-uniform sampling characteristics of ERP projection, the weights near pole areas are close to 0. Therefore we can appropriate to modify the judgment conditions to further decrease coding time. After the data analysis and experiment, it is found that in the projected region with weight $w < 0.4$ (weight w is WS-PSNR’s weight), we can modify judgement condition, as is shown in formula (8). That is to say, the category of CTU average depth range [2, 2.4375] is changed from T3 to T4. The purpose is to reduce the traversal of depth 1. Then the judgement condition of depth=2 is summarized in the

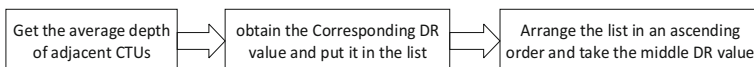


Fig. 2 CTU premature termination algorithm

Table 2 DR value and corresponding depth range

Prediction Depth Category	DR value	Prediction Depth Range [min, max]
T1	0	[0, 1]
–	1	[0, 2]
T2	2	[1, 2]
T3	3	[1, 3]
T4	4	[2, 3]
–	5	[0, 3]

prediction depth range [1, 2]. The purpose is to reduce the traversal of depth 3 but also introduces prediction errors.

$$\left\{ \begin{array}{ll} \text{Prediction Depth Range} & \text{Average Depth of CTU} \\ \text{Depth : 0, 1} & \text{Depth} \leq 1 \\ \text{Depth : 1, 2} & \text{Depth} = 1.25/1.5/1.75/2 \\ \text{Depth : 1, 2, 3} & 1.3125 < \text{Depth} < 2(\text{Depth} \neq 1.5/1.75/2) \\ \text{Depth : 2, 3} & \text{Depth} > 2 \end{array} \right. \quad (8)$$

However, considering that pole areas have lower weight coefficient, which means these areas have little impact on quality on the whole under the assessment of WS-PSNR, and only the regions with weight $w < 0.4$ of judgement conditions are changed. After the data analysis, we can ensure that this solution changes rate distortion cost little and further accelerates the process of CU partition.

In this paper, the early termination of CU partition algorithm is as follow: According to Fig. 3 and Table 2, the three adjacent CTUs are classified based on the average depth of CTU to obtain the prediction depth range. Then we obtain the corresponding depth categorie based on the prediction depth range and achieve the corresponding DR value according to Table 2. Next

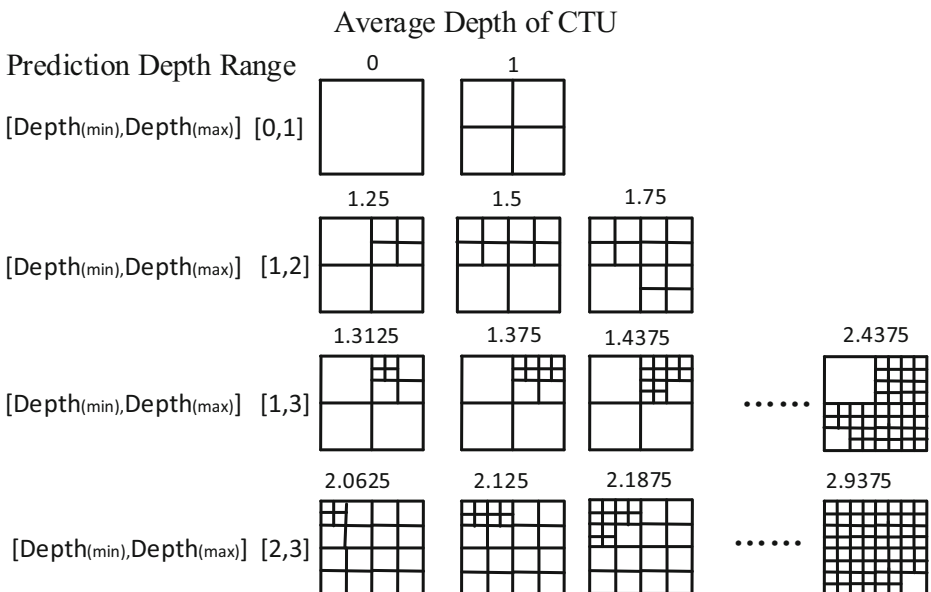


Fig. 3 types of CTU partition

three DR values are put into the list and arranged in an ascending order. We take the middle DR value of the list and get its corresponding prediction depth range. Then we set this prediction depth range as the current CU prediction depth range.

But there is a problem with this algorithm: it may be misjudged because of the narrow interval like [0, 1], [1, 2] and [2, 3]. For example, after obtaining the depth categorie of adjacent CTUs, we get the DR values according to the Table 2 and put them in a list in an ascending order. The list is obtained as [0, 0, 2]. In this case, the middle DR value of the list is 0 and corresponds to category T1([0, 1]) according to Table 2. However, after full traversing the current CTU, the DR category is actually T2([1, 2]). This is a misjudgment caused by the narrow interval. Therefore, the algorithm needs to add two DR values (1 and 5) additional, as shown in Table 2. At this moment, the list is expanded to be [0, 0, 1, 2, 5] and we take the middle DR value 1, which means the current CTU prediction depth range is [0, 2] according to Table 2. It is easy to see that DR 1([0, 2]) covers the DR 0([1, 2]), which reduces the misjudgemnet. When three values in the list are 0, it indicates that the current CTU is mostly in a smooth area, therefore categories T1 is highly likely. Similarly, adding DR = 5 is to cope with region with higher complexity and reduces the misjudgment of category T4. Therefore, this paper initializes the list to [1, 5] and modifies it by obtaining the prediction depth range categories of adjacent CTUs and then take the middle DR value as the current CTU prediction depth categorie.

3.3 Fast intra prediction mode decision

The intra prediction angular mode [19, 31] is strongly related to the pixel changes in one direction of the current coding unit. It is possible to know the approximate range of the angular prediction in advance if we get the direction in which the pixel variation of the encoding unit is minimized. By this way, we can predict the range of the angular mode roughly, which reduces the encoding complexity. Because it is not only skipping the rough selection process directly, but also greatly reducing the number of intra prediction modes involved in the RDO calculation process. This paper proposes to use the prewitt operator to extract the pixel change direction in a 4×4 size coding unit. Figure 4 shows the prewitt operator template:

prewitt operator is a 3×3 pixel matrix, the center point is $f(x, y)$. First, the template operator is used to calculate the gradient of the corresponding direction in four directions, as shown in (9) ~ (12). G_x and G_y are obtained by combining the gradient values in four directions as shown in Eqs. (13) to (14). After obtaining G_x and G_y and taking their average, the maximum direction of pixel variation can be obtained using Eq. (15), and finally the predicted angular mode is obtained by (16) and (17).

$$G_H = f(x-1, y+1) + f(x, y+1) + f(x+1, y+1) - f(x-1, y-1) - f(x, y-1) - f(x+1, y-1) \quad (9)$$

$$G_V = f(x+1, y-1) + f(x+1, y) + f(x+1, y+1) - f(x-1, y-1) - f(x-1, y) - f(x-1, y+1) \quad (10)$$

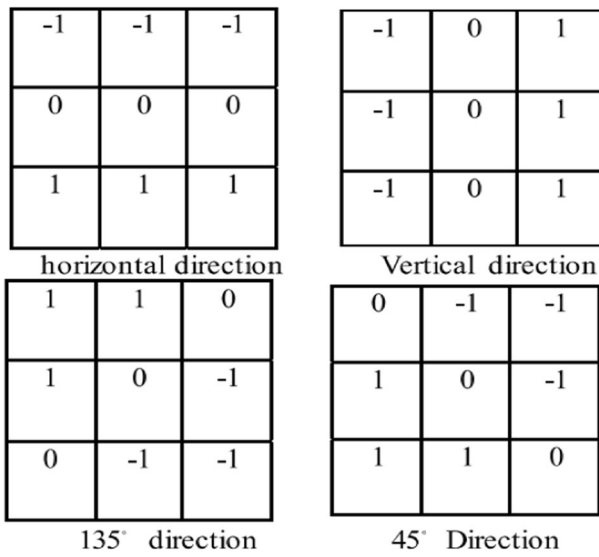


Fig. 4 prewitt operator template

$$G_{45} = f(x-1, y) + f(x-1, y + 1) + f(x, y + 1) - f(x, y-1) - f(x + 1, y-1) - f(x + 1, y) \quad (11)$$

$$G_{135} = f(x-1, y-1) + f(x, y-1) + f(x-1, y) - f(x + 1, y + 1) - f(x, y + 1) - f(x + 1, y) \quad (12)$$

$$G_x = G_v + \frac{G_{45}}{2} + \frac{G_{135}}{2} \quad (13)$$

$$G_y = G_H + \frac{G_{45}}{2} - \frac{G_{135}}{2} \quad (14)$$

$$\theta = \arctan\left(\frac{\overline{G}_y}{\overline{G}_x}\right), \theta \in (-0.5\pi, 0.5\pi) \quad (15)$$

$$\theta^{new} = \begin{cases} \theta + 0.5\pi & \theta \in (-0.25\pi, 0.5\pi) \\ \theta + 1.5\pi & \theta \in (-0.5\pi, -0.25\pi) \end{cases} \quad (16)$$

$$mode = -32 \times \frac{\theta^{new}}{\pi} + 42 \quad (17)$$

$$|\theta_i - \theta_j| < 0.4, \quad i, j = 1, 2, 3, 4 \quad i \neq j \quad (18)$$

For a 4×4 size PU block, it contains four 3×3 pixel matrices. The paper sets angles of pixel matrices in four directions to $\theta_1, \theta_2, \theta_3$ and θ_4 respectively. Then texture correlation of four blocks are used to judge the significance of the texture information in the current PU. $\theta_1, \theta_2, \theta_3$ and θ_4 have six different combinations. It shows that the current two sub blocks have large texture correlation [10] if the following formula (18) holds. In the meantime, if the formula (18) holds, then N(N is initialized to 0) increases by 1. The texture correlation of the current PU is not critical if the number of N is less than 3, which means that the texture information of the current PU block is in disorder and the approximate prediction mode based on texture information is probably inaccurate. Thus, in the case that the texture direction is not significant, the prediction modes with the lowest cost value selected by RMD will be set to candidate prediction modes. If the number of the set N is over 2, the texture direction of the current PU is significant and we calculate the prediction angular modes by formula (15, 16, 17).

For a 8×8 size PU block, it has 36 pixel matrixs and requires a large amount of computation. When PU size is $16 \times 16, 32 \times 32, 64 \times 64$, the computational complexity of pixel matrixs is even huge, therefore this paper doesn't continue to use prewitt operator. Through many statistical analysis and tests, we find it can reduce complexity by reducing unnecessary prediction modes. This paper first analyzes the best mode and the cost calculated by the rough selection process [5], then proposes an efficient mode decision algorithm. This paper performs RMD in two steps to reduce computational complexity. During the first RMD process, 35 original modes are reduced to 11 modes, these are 0, 1, 2, 6, 10, 14, 18, 22, 26, 30 and 34 respectively. The details are shown in Fig. 5. After the first RMD, distinguishing the first mode(FM is short for the first mode) in the list that obtained after the first RMD and then putting modes adjacent to FM like FM-2, FM-1, FM + 1, FM + 2 and modes 0, 1, 2, 6, 10, 14, 18, 22, 26, 30, 34 into a new list for performing the second RMD. After these two phases, getting the first two modes in the final list obtained after the second RMD process for Rate Distortion Operation (RDO). Compared to the original HEVC, it reduces the unnecessary candidate modes in RDO [26]. The following is the detailed algorithm.

First, we simplify 33 angular modes into 9 angular modes [25] with equal intervals, as shown by the red lines in Fig. 5. The first RMD traverses 11 modes, these are nine angular

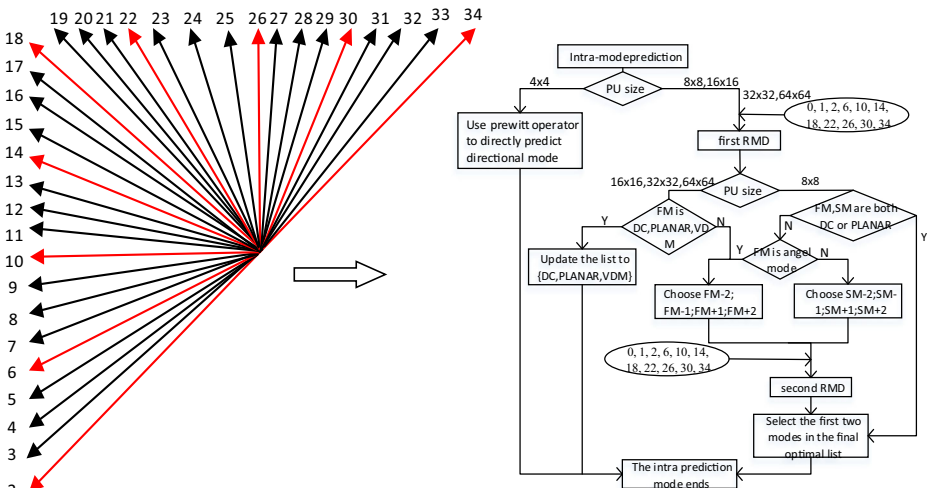


Fig. 5 11 modes for the first RMD

modes, DC and PLANAR respectively. The list that obtained after the first RMD process is defined as MyMode. Second, we distinguish whether the FM in MyMode is mode 0 or 1. If the FM is mode 0 or 1, the PU best mode is likely to be mode 0 or 1; if the FM is an angular mode, it can be speculated that the PU is very likely to be a mode that adjacent to the FM. Then we add these modes that adjacent to the FM with 11 modes that belongs to the first RMD in a new list for performing the second RMD.

For different sizes of PU, we take two different algorithms to get the best prediction mode. (1) PU size is 16×16 , 32×32 , 64×64 : This paper first distinguishes the FM in MyMode, if the FM is a DC, a PLANAR or a VM (VM is short for mode 26), we retain three modes in MyMode. If the FM is an angular mode, the four modes that adjacent to the FM (FM-1, FM-2, FM + 1, FM + 2) are added to a list with 11 modes for performing the second RMD. It takes the first two modes to be best prediction modes in the final list that obtained after the second RMD. Comparing with the original HEVC framework, it decreases one prediction mode. For example, after the first RMD, if the FM in MyMode is angular mode 6, the four modes adjacent to 6 are 4, 5, 7 and 8. Then these are added to a list with 11 modes for performing the second RMD. At last, we take the first two modes in the final list obtained after the second RMD process as prediction modes of the PU; (2) PU size is 8×8 : after the first RMD, determining FM and SM (SM is short for the second mode) in MyMode whether both are DC or PLANAR. In this case, this algorithm will choose the first two of MyMode as the best prediction modes; if FM or SM in MyMode is an angular mode, adding four modes adjacent to the angular mode (FM or SM) in a list with 11 modes for the second RMD. Finally, getting the final list obtained after the second RMD and taking the first two modes as the prediction modes. By the way, when angular mode numbers are 2 or 34, these are only two modes adjacent them.

4 Algorithm process

- (1) Calculate the newly defined w_{new} and obtain newly QP_{new} according to the formula (18). Then we set it as the current Slice parameter;
- (2) Analyzing categories of CTU partition, we get the prediction depth range based on the average CTU depth. Then we obtain the DR value according to Table 2. Next, we put these DR values in a list and arrange the list in an ascending order. At last, we take the middle DR value of the list to predict the current CTU depth range based on spatial correlation. At the same time, considering the characteristic of ERP projection, we expand the judgement conditions for pole areas with weight $w < 0.4$, which can further decrease the encoding time;
- (3) In intra prediction mode, we use prewitt operator to analyze 4×4 PU texture. If texture correlation is strong enough, we can directly predict the angular mode. At the same time, we cope with 8×8 and 16×16 , 32×32 , 64×64 PU respectively. The first RMD traverses only 11 modes with equal space and obtains the FM in MyMode. During the second RMD, getting four modes that are adjacent to the FM, then we traverse them with 11 modes (in the first RMD) for the second RMD and get the final list at last. We take the first two of the final list as prediction modes.

Figure 6 shows the architecture of proposed algorithm:

Intra Encoding Optimization Algorithm Pseudo-code:

Improved Adaptive QP Compensation:

Calculate W_{new} based on the formula(3) and (4) in the formula(5) for QP_{new}

Then get QP_{new} according to W_{new} in the formula(6)

Ctu Early Termination Partition:

w is the weight in WS_PSNR

If($w < 0.4$)

then use formula(8) to get the prediction depth range based on the average depth of

CTU partition

Else

use formula(7) to get the prediction depth range according to the average depth of

CTU partition

Obtain DR value according to the table 2

Init DR list=[5, 5, 5, 1, 5]

Put DR values obtained from three adjacent CTUs in a list and arrange in an ascending order

Take the middle DR value of the list as the current CTU prediction depth category

Fast Mode Selection:

4x4 PU:

Use prewitt operator to get four θ_n according to the formula(9)–(14)

Then use formula(18) to judge the texture correlation

If these four θ_n hold the formula(18)

Then we use formula(15)–(17) to get the prediction mode

64x64, 32x32, 16x16, 8x8 PU:

First RMD:

Put DC, Planar and 9 modes shown in Figure.5 to perform first RMD and then get MyMode list

Second RMD:

64x64, 32x32, 16x16:

If (FM= = DC || Planar || VM)

Retain the first three modes in MyMode for RDO

Else

Put four modes adjacent to FM(FM-2, FM-1, FM+1, FM+2) with 11 modes to perform the second RMD

Get the final candidate list

Retain the first two modes in the final candidate list for RDO

8x8:

If ((FM= = DC && SM= =Planar) || (FM= = Planar && SM= =DC))

Retain the first two modes in MyMode for RDO

Else

If (FM= = AngularMode)

Put FM-2, FM-1, FM+1, FM+2 modes with 11 modes for the second

RMD

Get the final candidate list

Maintain the first two modes in the final candidate list for RDO

Else

Put SM-2, SM-1, SM+1, SM+2 modes with 11 modes for the second

RMD

Get the final candidate list

Maintain the first two modes in the final candidate list for RDO

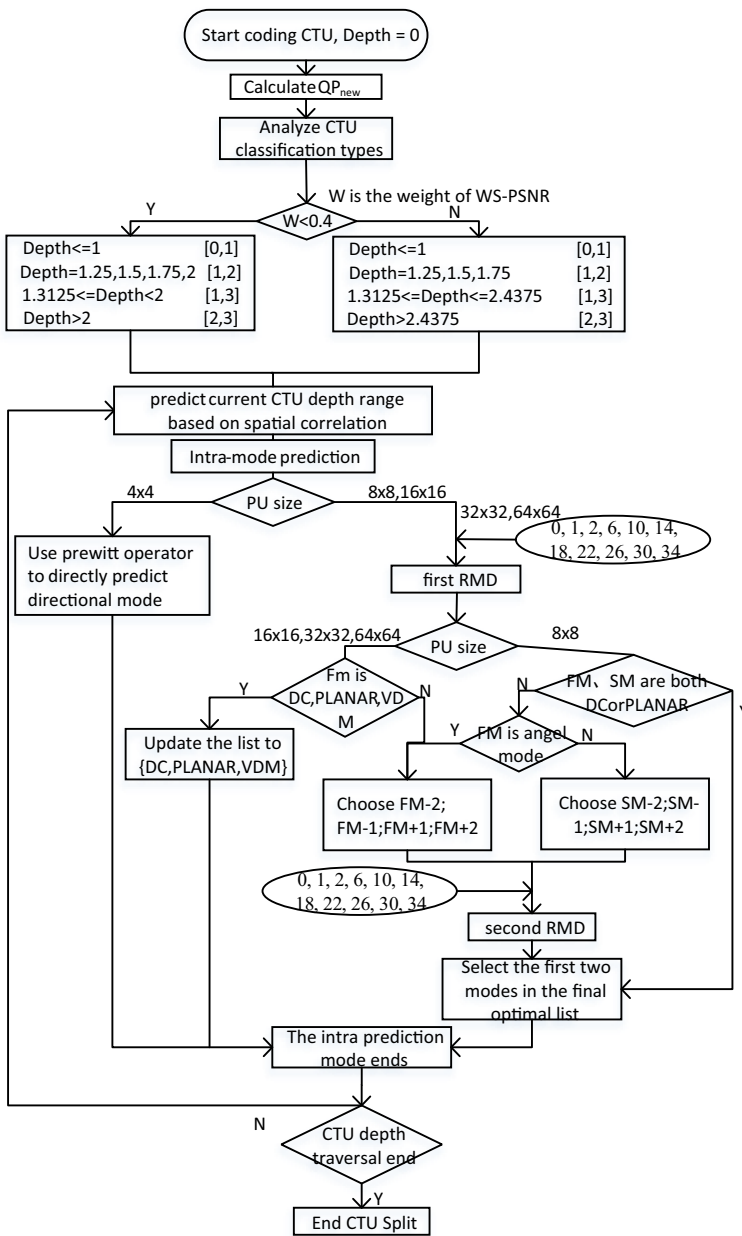


Fig. 6 Architecture of proposed algorithm

5 The experiment result analysis

In order to verify the feasibility of the proposed compression algorithm, we integrate the algorithm to test the rate distortion performance and coding complexity in HM-16.16-360Lib-4.0. Experimental platform hardware is set to Intel Core i7–7700 CPU @ 3. 60GHz and RAM is 8. 0GB. The coding parameter of the experiment is All Intra Main10 (AI-Main10). The

Table 3 Experimental data

Type	QP	Sequences	Bitrates		WS-PSNR				Time			
			HM	Proposed	Δ BR(%)	HM	Proposed	Δ W/SPSNR	HM	Proposed	Δ T(%)	
3840 × 1920	22	AerialCity	85,939.5	85,933.93	0.20%	42,069	42,129	0.06	3433.745	2069.023	39.74%	
	27		45,365.35	45,910.48		39,672	39,727	0.055	2856.984	1624.507	43.14%	
	32		26,009.66	26,588.78		37.3	37,369	0.069	2575.613	1380.009	46.42%	
	37	PoleVault_le	14,574.11	14,907.75	-1.70%	34,629	34,659	0.03	2404.203	1160.734	51.72%	
	22		148,029.4	146,417.8		43,481	43,553	0.072	3724.37	2501.894	32.82%	
	27		84,984.37	84,018.35		39,702	39,748	0.046	3223.346	2074.558	35.64%	
	32		48,095.45	47,746.03		36,222	36,267	0.045	2826.193	1738.821	38.47%	
6144 × 3072	37	BranCastle2	26,721.05	26,440.35	-1.20%	32,995	32,998	0.003	2557.195	1413.582	44.72%	
	22		279,834.7	277,816.8		42,689	42,734	0.045	6111.028	4395.394	28.07%	
	27		172,627.6	171,012.3		38,506	38,523	0.017	5344.813	3730.952	30.19%	
	32	98,833.07	97,895.25		34,69	34,703	0.013	4630.548	3079.191	33.50%		
	37	52,159.1	51,184.58	0.50%	31,284	31,246	-0.038	4120.219	2225.45	45.99%		
	22	98,601.07	98,634.1		44,979	45,034	0.055	4556.858	2674.955	41.30%		
	27	55,704.05	55,953.77		41,655	41,689	0.034	4087.944	2213.772	45.85%		
8192 × 4096	32	Landing2	29,671.31	30,085.7	1.50%	38,497	38,516	0.019	3773.1	1856.971	50.78%	
	37		15,286.61	15,559.95		35,641	35,605	-0.036	3547.801	1561.947	55.97%	
	22		210,940.2	213,227.3		44,933	45,02	0.087	4621.089	3010.77	34.85%	
	27	123,681.1	125,933.2		41,423	41,487	0.064	4169.642	2579.537	38.14%		
	32	69,906.07	71,980.27		38,092	38,138	0.046	3855.016	2176.022	43.55%		
	37	37,958.36	39,165.24	1.30%	34,941	34,923	-0.018	3606.977	1764.306	51.09%		
	22	184,396.3	188,438.6		45,285	45,442	0.157	4426.458	2696.57	39.08%		
8192 × 4096	27	Balboa	108,340	111,600.1	0.90%	41,781	41,914	0.133	4036.841	2311.36	42.74%	
	32		60,346.19	62,780.83		38,442	38,559	0.117	3792.855	1970.061	48.06%	
	37		31,888.71	33,225.98		35,331	35,374	0.043	3516.671	1633.556	53.55%	
	22	101,304.4	101,734		44,365	44,42	0.055	4813.029	3198.355	33.55%		
	27	55,236.76	55,865.61		41,125	41,173	0.048	4255.502	2713.391	36.24%		
	32	28,870.07	29,574.74		38,073	38,113	0.04	3879.339	2304.002	40.61%		
	37	14,706.84	15,171.79	-1.40%	35,244	35,234	-0.01	3678.767	1842.245	49.92%		
8192 × 4096	22	Trolley	234,429.1	232,080.1		42,982	43,014	0.032	5804.426	3626.068	37.53%	
	27		144,917.2	142,877.5		38,852	38,859	0.007	5046.348	3082.977	38.91%	
	32	80,646.49	79,396.16		35,03	35,031	0.001	4470.08	2592.48	42.00%		
	37	40,020.31	39,253.67		31,699	31,662	-0.037	3993.759	2034.316	49.06%		

Table 3 (continued)

		Bitrates	WS-PSNR	Time	
22	Harbor	97,291.79	44.914	4638.971	2846.147
27		54,745.94	41.62	4164.31	2471.061
32		29,055.82	38.402	3845.415	2139.729
37		14,634.02	35.376	3585.434	1816.155
22	KiteF1ite	215,071.3	43.21	5679.648	3917.995
27		133,425.4	39.284	5021.385	3394.096
32		76,789.59	35.573	4465.089	2909.569
37		40,216.15	32.237	4020.239	2320.204
22	Chairf1fRide	117,227.3	43.821	5024.001	3033.952
27		60,217.89	40.613	4350.517	2281.632
32		28,752.41	37.765	3919.722	1879.08
37		13,293.76	35.316	3698.362	1621.015
22	Skateboardf1mLot	87,940.4	45.399	4531.508	2865.716
27		48,271.25	42.035	4096.827	2323.765
32		23,996.96	38.785	3795.731	1943.923
37		11,130.97	35.935	3551.233	1640.735
AVERAGE					42.39%

number of coding frames is 120 and the QP are 22, 27, 32, and 37 respectively. In order to measure the rate distortion performance of the algorithm, BD-rate is used to represent the bit rate variation under the same image quality condition. The symbol ΔT is used to measure time saving and the WS-PSNR is used to measure the image quality change. Time reduction is calculated by the following formula(19), where $T_{HM16.16}$ is the coding time of HM16.16, $T_{proposed}$ is the coding time of the proposed algorithm, and ΔT is the time reduction. The decrease in WSPSNR-Y is calculated using formula(20).

$$\Delta T = \frac{T_{HM16.16} - T_{proposed}}{T_{HM16.16}} \times 100\% \tag{19}$$

$$\Delta WSPSNR_Y = WSPSNR_{HM16.16} - WSPSNR_{proposed} \tag{20}$$

In this paper, twelve standard test sequences proposed by the proposals of JVET-D0026 [1], JVET-D0039 [3], JVET-D0053 [7], JVET-G0147, JVET-D0143 and JVET-D0179. Prior to encoding, test sequences are converted to a low-resolution ERP for encoding (for accuracy quality assessment). For 8 K and 6 K ERP video, the encoding size is set to 4096×2048 , and for 4 K ERP video, the encoding size is set to 3328×1664 .

In general, the different test sequences achieve different time-saving percentages mainly because of different texture and contents. The experimental results in Table 3 show that the proposed algorithm reduces the average BD-rate by 0.3%, decreases coding time by 42.4% and improves the WS-PSNR by 0.03 compared with HM16.16. This is because the proposed algorithm uses spatial correlation to perform the CU early partition based on the prediction of depth range. By using the texture features of PU and reducing unnecessary prediction modes, the proposed algorithm reduces the coding

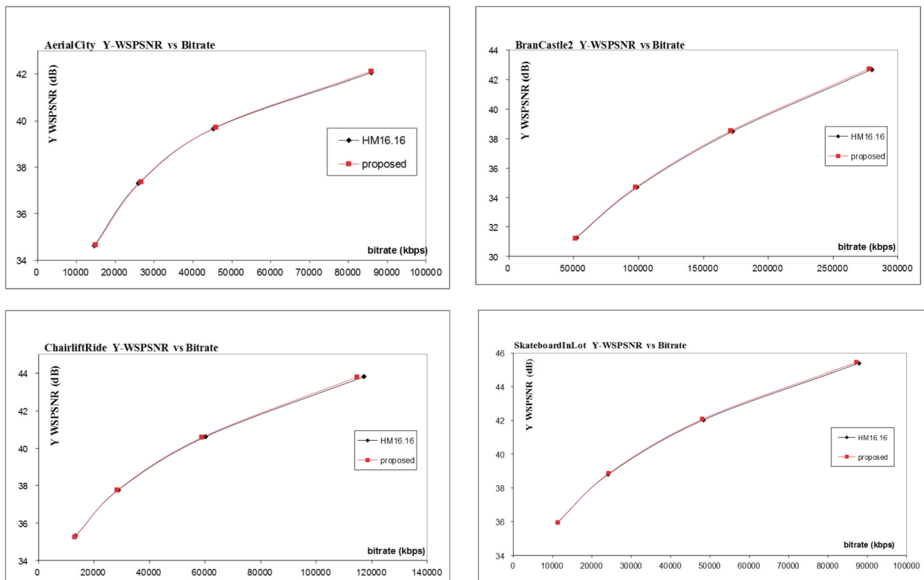


Fig. 7 the comparison between this paper and HM16.16 in rate distortion performance

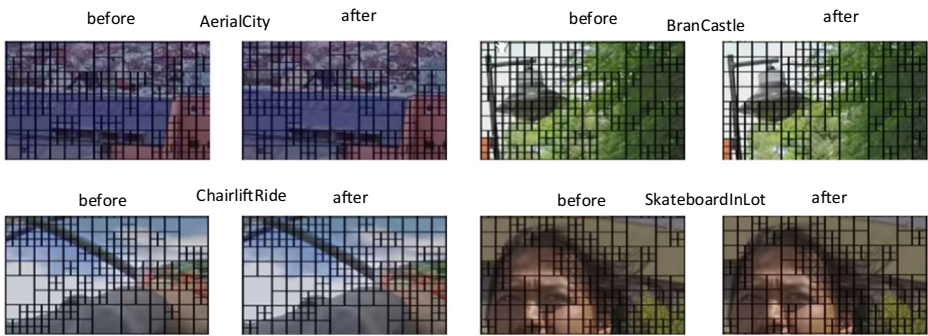


Fig. 8 the comparison between the proposed algorithm and HM16.16 in CU partition

complexity. In adaptive QP compensation, QP parameters are adaptively compensated based on the coordinate of pixel in ERP projection that can make encoding more efficiently and reasonably. This proposed algorithm reduces more time in the sequence of ChairliftRide, Balboa and SkateboardInLot mainly because the textures of these background are relatively simple that may skip coding unnecessary smaller size CU; but less time reduction for sequences such as KiteFlite and BranCastle2, because the textures of these test sequences are more complexity and mostly likely to be partitioned into smaller CUs, which reveals that their corresponding depths are larger than other sequences, and hence, the premature termination algorithm has a small effect for them on decreasing encoding time. By analyzing the uneven sampling characteristic of ERP projection, the weight is very small at pole areas, which means the distortion impact of these areas is very little on the whole under the assessment of WS-PSNR. Therefore we can expand the judgment conditions for these areas to further decrease the encoding time. Figure 7 compares the rate-distortion performance of HM16.16 with proposed algorithm in sequences like AerialCity, BranCastle2, ChairliftRide and SkateboardInLot. Figure 8 shows the differences between the proposed algorithm and HM16.16 in CU partition. Table 4 shows the accuracy between the proposed algorithm and HM16.16 in premature termination of CU partition.

Table 4 the accuracy of CTU classification in this paper

Sequence	Depth0	Depth1	Depth2	Depth3
AerialCity	81.58%	81.22%	86.46%	93.65%
PoleVault	88.28%	84.74%	87.42%	91.85%
BranCastle	92.59%	85.46%	86.01%	88.70%
Landing	84.90%	83.14%	89.69%	93.21%
Broadway	89.09%	82.98%	88.93%	91.96%
Balboa	81.31%	83.31%	89.80%	93.91%
GasLamp	89.84%	83.90%	88.22%	92.68%
Trolley	87.87%	86.83%	90.22%	92.80%
Harbor	88.01%	85.30%	90.15%	94.05%
KiteFlite	89.99%	82.01%	87.00%	90.73%
ChairliftRide	84.31%	85.28%	85.40%	93.15%
SkateboardInLot	83.01%	82.51%	90.22%	95.63%
Average	86.73%	83.89%	88.29%	92.69%

6 Conclusion

This paper presents an adaptive QP compensation, CU early partition and adaptive mode selection algorithm for reducing computational complexity of the 360 video coding. The proposed algorithm is performed on HEVC reference software HM16.16. The QP compensation is adaptive modified according to the coordinate of pixel in ERP projection. The CU early partition is based on the prediction of depth range and spatial correlation. The adaptive mode selection decreases the number of modes in RMD process and if the texture correlation of block is strong, it can directly predict the mode. The experimental results show that the proposed algorithm reduces the average BD-rate by 0.3%, decreases 42.4% of the average coding time and improves the WS-PSNR by 0.03 compared with HM16.16. This proves that the algorithm has strong practical value.

Acknowledgements 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).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

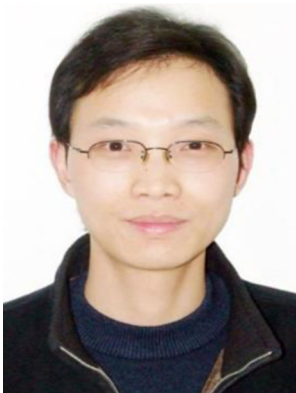
References

1. A Abbas, B Adsumilli (2016) (GoPro), "New GoPro Test Sequences for Virtual Reality Video Coding", Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 JVET-D0026 Chengdu
2. E Alshina, J Boyce, A Abbas, Y Ye (2017) (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
3. E Asbun, Y He, Y He, Y Ye (2016) (InterDigital), AHG8: InterDigital Test Sequences for Virtual Reality Video Coding. Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 JVET-D0039 Chengdu
4. Bai C, Yuan C (2014) Fast coding tree unit decision for HEVC intra coding[C]// *Icec-China Workshop*. IEEE 28–31
5. Chen ZY, Chang PC (2016) Rough mode cost-based fast intra coding for high-efficiency video coding[J]. *J Vis Commun Image Represent* 43:77–88
6. Cho S, Kim M (2013) Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding[J]. *IEEE transactions on Circuits & Systems for video. Technology* 23(9):1555–1564
7. K Choi, E Alshina, C Kim (2016) (Samsung), "AHG5: Fast encoding setting for JEM", Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 JVET-D0053 Chengdu
8. Goswami K, Kim BG, Jun D et al (2014) Early coding unit-splitting termination algorithm for high efficiency video coding (HEVC)[J]. *ETRI J* 36(3):407–417
9. Gu J, Tang M, Wen J, et al. (2017) Adaptive Intra Candidate Selection with Early Depth Decision for Fast Intra Prediction In HEVC[J]. *IEEE Sign Proc Lett* (99): 1–1
10. Hu J, He G, Li Y (2016) Fast algorithm based on the sole- and multi-depth texture measurements for HEVC intra coding[J]. *J Vis Commun Image Represent* 40:671–681
11. Kim BG (2017) Fast coding unit (CU) determination algorithm for high-efficiency video coding (HEVC) in smart surveillance application[M]. Kluwer Academic Publishers
12. Lee D, Jeong J (2017) Fast intra coding unit decision for high efficiency video coding based on statistical information[J]. *Signal processing image. Communication* 55:121–129

13. Lee J, Kim S, Lim K et al (2015) A fast CU size decision algorithm for HEVC[J]. *Circuit Syst Video Technol IEEE Trans* 25(3):411–421
14. Y Li, J Xu, Z Chen (2017) Spherical domain rate-distortion optimization for 360 video coding. *IEEE Int Conf Multimed Expo (ICME)*, Hong Kong 709–714
15. Liu X, Liu Y, Wang P, et al. (2016) An Adaptive Mode Decision Algorithm Based on Video Texture Characteristics for HEVC Intra Prediction[J]. *IEEE Trans Circuit Syst Video Technol* (99): 1–1
16. Min B, Cheung RCCA (2015) Fast CU size decision algorithm for the HEVC intra encoder[J]. *IEEE transactions on Circuits & Systems for video. Technology* 25(5):892–896
17. Nishikori T, Nakamura T, Yoshitome T, et al. (2014) A fast CU decision using image variance in HEVC intra coding[C]// *Industrial Electronics and Applications*. IEEE 52–56
18. Ohm JR, Sullivan GJ, Tan TK et al (2013) Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC)[J]. *IEEE transactions on Circuits & Systems for video. Technology* 22(12):1669–1684
19. Ruiz D, Fernández-Escribano G, Martínez JL et al (2016) Fast intra mode decision algorithm based on texture orientation detection in HEVC[J]. *Signal Process Image Commun* 44(C):12–28
20. Shen L, Liu Z, Zhang X et al (2013) An effective CU size decision method for HEVC encoders[J]. *IEEE Trans Multimed* 15(2):465–470
21. Shen L, Zhang Z, Liu Z (2014) Effective CU size decision for HEVC intracoding[J]. *IEEE transactions on image processing a publication of the IEEE signal processing Society* 23(10):4232
22. Sullivan GJ, Ohm JR, Han WJ et al (2012) Overview of the high efficiency video coding (HEVC) standard[J]. *IEEE transactions on Circuits & Systems for video Technology* 22(12):1649–1668
23. Y Sun, A Lu, L Yu (2016) AHG8: WS-PSNR for 360 video objective quality evaluation. *Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 JVET-D0040* Chengdu
24. HTT Tran, NP Ngoc, CM Bui, MH Pham, TC Thang (2017) An evaluation of quality metrics for 360 videos. *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, Milan, , pp. 7–11
25. Tseng CF, Lai YT (2016) Fast coding unit decision and mode selection for intra-frame coding in high-efficiency video coding[J]. *IET Image Process* 10(3):215–221
26. Wang T, Men Y, Zhang Y, et al. (2017) A fast intra-prediction decision algorithm in inter-frame based on a novel feature of HEVC[C]// *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE 1532–1536
27. Yang M, Grecos C (2014) Fast intra encoding decisions for high efficiency video coding standard[J]. *J Real-Time Image Proc* 1–10
28. Zhang H, Ma Z (2014) Fast intra mode decision for high efficiency video coding (HEVC)[J]. *IEEE transactions on Circuits & Systems for video Technology* 24(4):660–668
29. Zhang M, Zhao C, Xu J (2012) An adaptive fast intra mode decision in HEVC[C]// *IEEE International Conference on Image Processing*. IEEE 221–224
30. Zhang M, Bai H, Lin C, et al. (2015) Texture Characteristics Based Fast Coding Unit Partition in HEVC Intra Coding[C]// *Data Compression Conference*. IEEE 477–477
31. Zhang T, Sun MT, Zhao D et al. (2016) Fast Intra Mode and CU Size Decision for HEVC[J]. *IEEE Trans Circuits Syst Video Technol* (99):1–1
32. Zhu S, Zhang C (2016) A fast algorithm of intra prediction modes pruning for HEVC based on decision trees and a new three-step search[J]. *Multimed Tools Appl* 76(20):1–22



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.



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.



Changzhi An received the B.S. degree in electronic information technology and M.S. degree in communication and information system from Beijing Jiaotong University, China in 2000 and 2003 respectively. Currently, he is the general manager of Beijing China Electronic Intelligent Communication Technology Co., Ltd.