Zhang et al. EURASIP Journal on Image and Video Processing (2019) 2019:53
https://doi.org/10.1186/s13640-019-0446-3

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Fast intra algorithm based on texture characteristics for 360 videos

Mengmeng Zhang[1*], Xiaosha Dong[1], Zhi Liu[1*], Fuqi Mao[1,2] and Wen Yue[2]

Abstract

With the rapid progress of virtual reality technology, 360 videos have become increasingly popular. Given that the resolution of a 360 video is ultra-high (generally 4K to 8K), the encoding time for this type of video is considerably high. To reduce encoding complexity, this study proposed a fast intra algorithm that is based on image texture characteristics. On the one hand, the proposed algorithm determines whether to terminate the coding unit partition early on the basis of texture complexity. On the other hand, the proposed algorithm reduces the number of candidate modes in mode decision according to texture directivity. Experimental results showed that the proposed algorithm can obtain an average time reduction rate of 53% and a Bjontegaard delta rate increase of only 1.3%, which is acceptable for rate distortion performance.

Keywords: HEVC, 360 video, Texture characteristics, Intra prediction, Mode decision

1 Introduction

With a growing commercial interest in virtual reality (VR) fields in recent years, ITU-T's Video Coding Experts Group (VCEG) and ISO/IEC's Moving Picture Exports Group (MPEG) jointly established the Joint Video Exploration Team (JVET) for future video coding research and proposed the VR 360° video (referred to as 360 video) scheme [1]. A 360 video is usually obtained from a multi-camera array, such as a GoPro Omni camera. Images from multiple cameras are assembled to achieve a scene with a spherical projection in the horizontal direction at 360° and in the vertical direction at 180°.

With the spherical feature of VR 360 videos, traditional video coding methods are difficult to apply directly. Therefore, JVET proposed 11 different projection formats for spherical videos to address coding issues. A 360 video is projected onto a two-dimensional plane and converted into a 2D projection format in a certain ratio [2], such as equirectangular projection (ERP), octahedron projection (OHP), truncated square pyramid projection (TSP), rotated sphere projection (RSP), cubemap projection (CMP), and segmented sphere projection (SSP) (Fig. 1). The video is then encoded as a traditional video.

The 360 video coding framework based on High Efficiency Video Coding (HEVC) is shown in Fig. 2 [3]. On the basis of traditional video coding, 360 video coding increases the down-sampling process before encoding and the up-sampling process after decoding and the conversion between formats in the codec process, and some new quality evaluation standards proposed for 360 videos such as peak to signal noise ratio weighted by sample area (WSPSNR) and spherical peak to signal noise ratio (PSNR) without interpolation (S_PSNR_NN) [4]. HEVC is the latest international coding standard and uses the traditional block-based hybrid coding framework. In HEVC, blocks are divided into different sizes. The coded image is divided frame by frame into a series of segments called a coding unit (CU), prediction unit (PU), and transform unit (TU). Compared with H.264, HEVC has 35 different prediction modes for luminance information, including 33 angle predictions, a planner, and a DC prediction. The mode decision is divided into two processes, namely, rough mode decision (RMD) and most probable mode (MPM).

The remainder of this paper is organized as follows. Section 2 provides the related works. Section 3 explains in detail the specific steps of the proposed algorithm. Section 4 verifies the effectiveness of the proposed

* Correspondence: muchmeng@126.com; lzliu@ncut.edu.cn
[1]North China University of Technology, Beijing 100144, China
Full list of author information is available at the end of the article

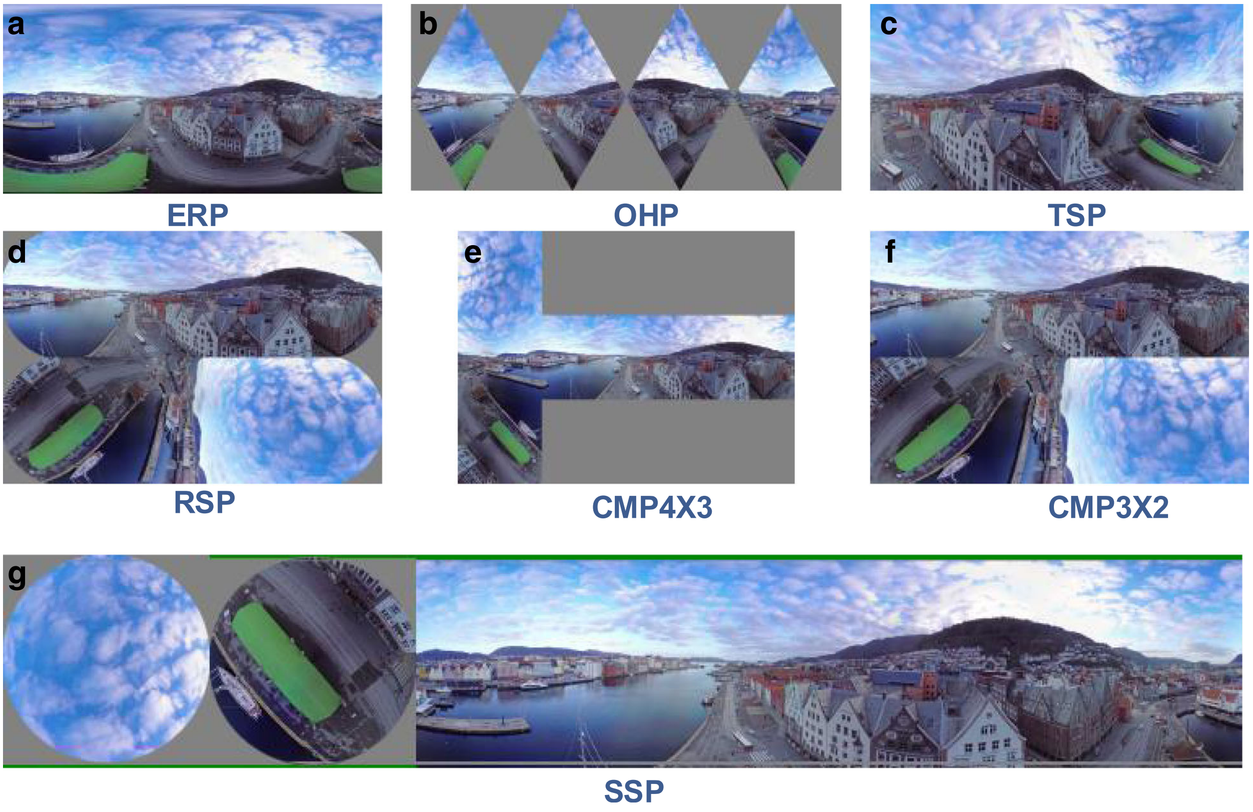

Fig. 1 Projection formats of 360 video. **a** ERP. **b** OHP. **c** TSP. **d** RSP. **e** CMP4X3. **f** CMP3X2. **g** SSP

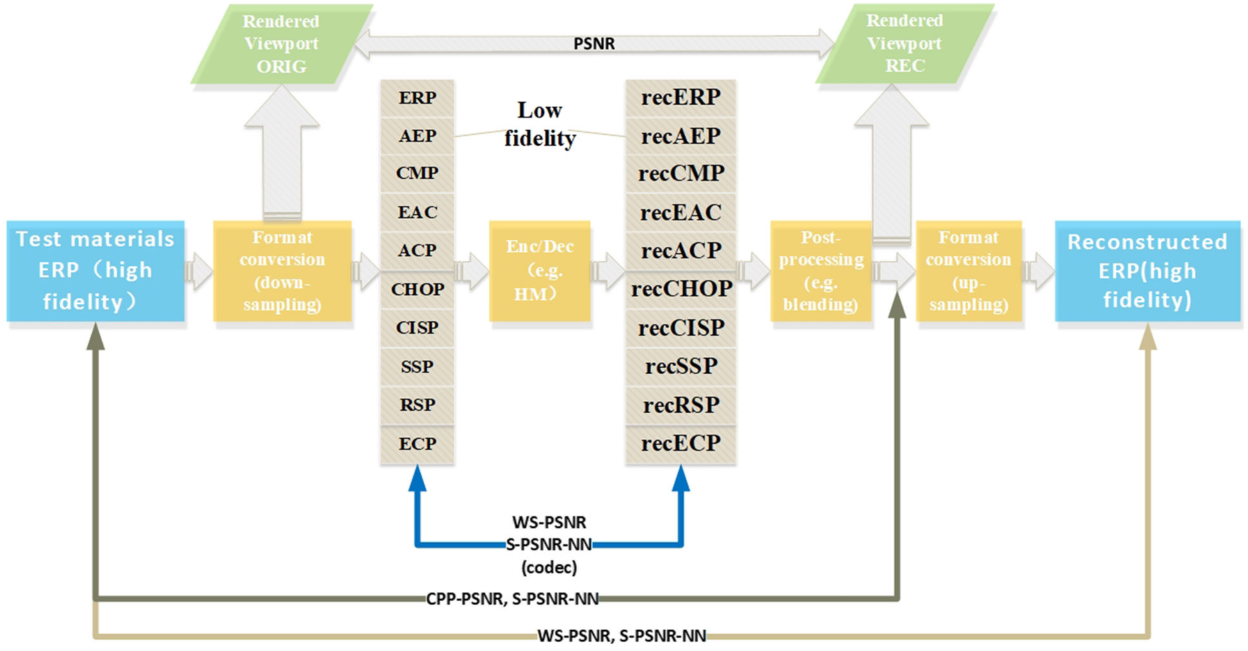

Fig. 2 Encoding process for 360 video testing

algorithm through the results of the experiment. Section 5 provides the conclusions.

2 Related works

Considering that HEVC is proposed for traditional videos, a 360 video coding framework based on HEVC cannot efficiently encode 360 videos. To improve the performance of 360 video coding, the authors in [5] proposed two algorithms of adaptive encoding techniques for omnidirectional videos (OVs) to reduce the bitrate of OVs after compression, as well as two other algorithms to reduce the bitrate. In [6], a real-time 360 video stitching framework was proposed to render the entire scene at different levels of detail. A motion-estimation algorithm that can improve the accuracy of motion prediction in 360 videos was proposed in [7]. In [8], considering distortion in the spherical domain, Li derived the optimal rate distortion relationship in the spherical domain and presented its optimal solution that can achieve bit savings of up to 11.5%. In [9, 10], the video encoding is optimized from wavelet image features and human visual system features.

A 360 video has a relatively high resolution and thus requires a large number of coding tree units (CTUs) in encoding. The process is time consuming and can affect real-time video encoding and transmission. We can improve coding efficiency by enhancing the HEVC intra prediction algorithm. In HEVC intra prediction, the rate distortion (RD) cost is used to determine the CU partition and mode decision and thus leads to high computational complexity. Therefore, a large number of improved algorithms have been proposed. These algorithms include fast CU/PU size decision such as that in [11, 12], fast intra prediction mode decision such as that in [13, 14], and fast mode decision such as that in [15, 16]. In [16], it exploits the depth information of neighboring CUs to make an early CU split decision or CU pruning decision, which can save 37.91% computational complexity on average as compared with the current HM (HEVC test model) with only a 0.66% increase. In [17], an algorithm that combines CU coding bits with the reduction of unnecessary intra prediction modes was proposed to decrease computational complexity; it provides an average time reduction rate of 53% with only a 1.7% Bjontegaard delta rate (BD rate) increase. Optimized algorithms based on image texture, such as the gradient-based algorithm in [18], edge density algorithm in [19], and texture-based algorithm in [20] are used in traditional videos. In [18], it proposes a fast hardware-friendly intra block size selection algorithm with simple gradient calculations and the bottom-up structure which can save 57.3% encoding time on average for all-intra main case with 2.2% BD rate increases. In [19], the algorithm includes a pre-partitioning for CUs based on the edge density of the textures to simplify the partitioning, it can save about 33% encoding time for all intra configurations with a slight loss of PSNR. In [20], it reduces the average encoding time by 47.21% at a cost of a 2.55% bit rate increase.

These proposed intra prediction algorithms are based on traditional videos. Traditional videos and 360 video projection formats show different image textures. Therefore, the texture thresholds selected in traditional videos are not applicable to 360 videos. In the present work, we propose a fast intra algorithm that is based on video texture characteristics that can be applied to 360 videos. We experimentally re-selected the texture complexity thresholds in the CU partition process to render the thresholds relatively suitable for 360 videos and added texture directivity to the mode decision process. After the experiments, we found that the proposed algorithm can efficiently reduce computational complexity and encoding time.

The proposed algorithm is divided into two parts. First, according to the thresholds of image texture complexity, the level of complexity is classified to determine whether the current CU is to be skipped or further divided. Second, the candidate prediction mode is further reduced on the basis of texture directionality.

3 Proposed algorithm

In VR 360 video coding, a spherical video is projected into a two-dimensional rectangular format. The ERP format is the most representative format in the VR 360 video format. In many encoding processes, other formats are first converted into an EPR format for encoding. For the most popular ERP projection format used in VR 360 video coding, the contents located near two poles are stretched substantially, resulting in changes in the texture feature of the area. We performed a comprehensive statistical analysis of 360 video sequences (4K–8K) in ERP format, all sequences used in the experiments are the test sequences provided by JVET. Figure 3 shows some sequences of different resolutions and encoding bits. As shown in Fig. 3, the stretch of the upper and lower parts of these sequences is usually the largest, and the texture is usually homogeneous. This means that these parts can use the smaller CU partition depth for intra prediction during the encoding process. Thus, these parts can be encoded using large blocks. The block information in the middle part of these sequences is relatively complex and requires small blocks to be encoded.

The proposed algorithm reduces the encoding complexity of a 360 video in two aspects. First, as shown in Fig. 4, the red part is located at the two poles of the

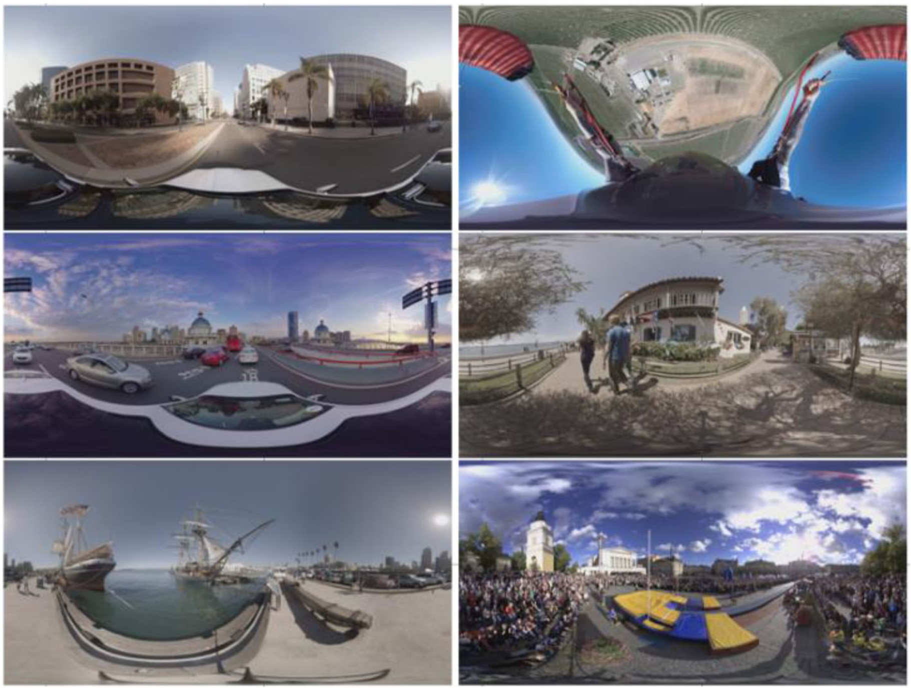

Fig. 3 Features of the upper and lower parts of 360 ERP video sequences

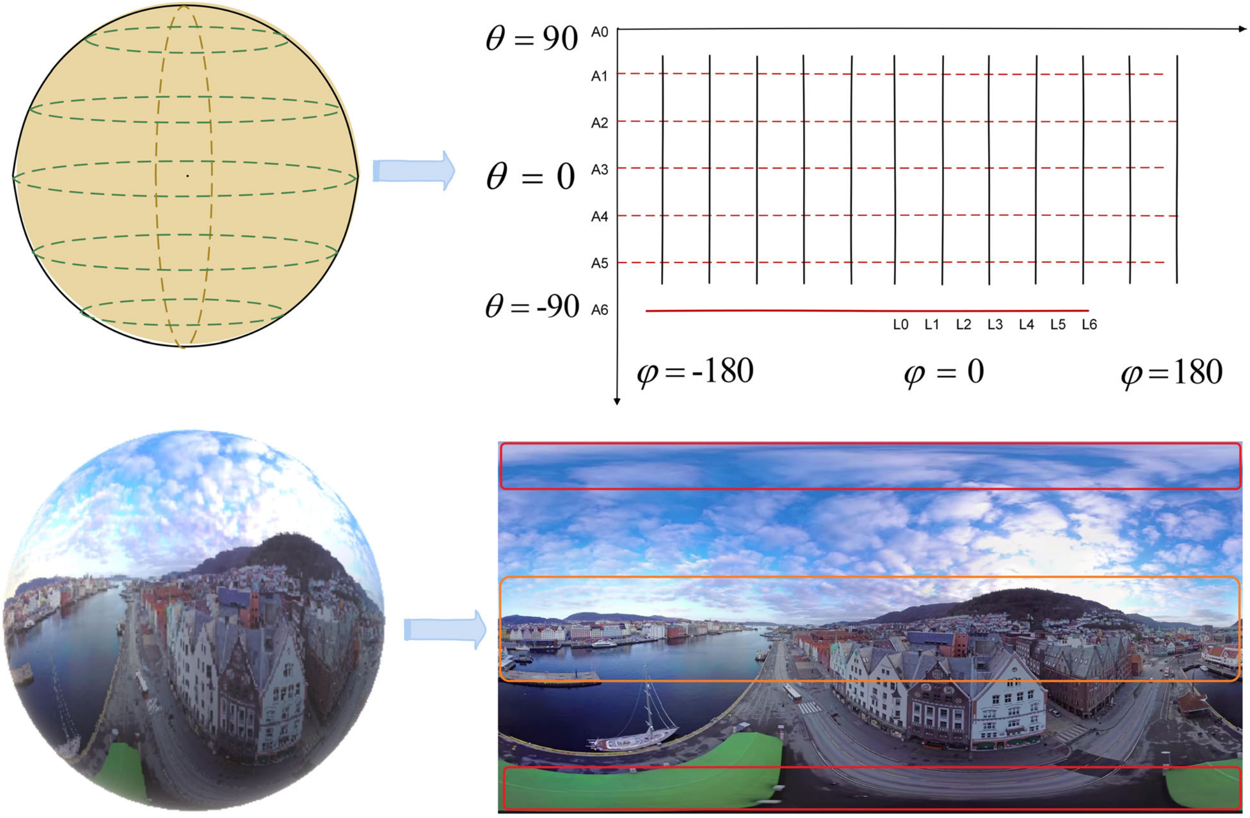

Fig. 4 ERP format: expands from spherical to rectangular

sphere; its stretching is the largest, and the stretching of the yellow sphere in the middle of the frame is relatively minimal. The proposed algorithm based on texture complexity can effectively skip the rate distortion optimization (RDO) calculation of these parts with minimal loss of viewing experience. The texture of the stretched part, such as the sky and water in Fig. 4, is relatively homogeneous. We can encode the CTUs with a large block and low depth to establish a balance between encoding bits and image quality. CTU blocks with complex textures, such as the houses and ships in Fig. 4, contain additional information and usually need to be divided into high depths to keep the video image information intact during the compression process. Second, the horizontal stretch of a 360 ERP video is much larger than the vertical stretch, thereby causing the directionality of the prediction mode to change relative to the case of traditional videos. We can optimize the mode decision by determining the texture directionality.

The following sections elaborate the proposed algorithm in three aspects. First, the ideas and preparations used in the algorithm are introduced. Second, the CU size decision algorithm based on texture complexity is presented. Third, the mode decision algorithm based on texture direction is discussed.

3.1 Proposed method

In the HEVC intra prediction for 360 videos, CU size and mode decision use RD cost to determine the best partition depth and mode. The RD costs of the current depth block and four sub-blocks are calculated and compared to determine whether to divide the blocks further.

The stretch and fill characteristics of a 360 video horizontally represent information about a 360° circle enclosing a person. The vertical direction shows a 180° angle of view information from the head of a person to the sole. In HEVC, the CTU size or number of pixels can be 16 × 16, 32 × 32, or 64 × 64. CTU can be decomposed into several CUs in a quad-tree structure. The CUs in the same level must be four square blocks that have uniform sizes, are close to each other, and are not overlapping. Each CU has up to four layers of decomposition, namely, 64 × 64, 32 × 32, 16 × 16, and 8 × 8, with corresponding depths of 0, 1, 2, and 3, respectively. Figure 5 shows the quad-tree structure of a CU in HEVC.

The proposed algorithm uses texture characteristics to improve the intra prediction of a 360 ERP video. On the basis of the complexity of the texture characteristics, we can predict the depth of a CU block in advance in the CU partition process. The directionality of the texture characteristics can also be used to determine the direction of candidate modes in the mode decision process.

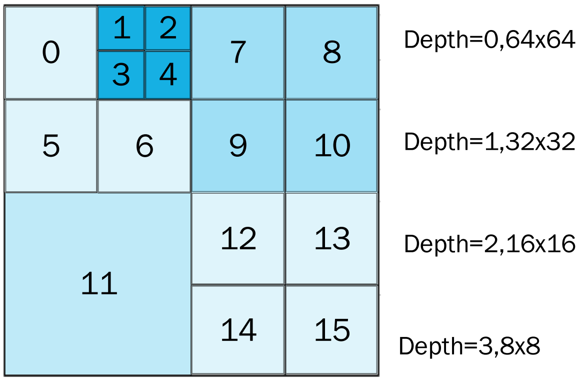

Fig. 5 Partition of a CU structure (64 × 64): quad-tree-based coding structure

The main purpose of the CU size decision algorithm based on texture complexity is to classify the horizontal and vertical texture complexities before RDO. Texture complexity is calculated in advance to determine whether the current block is to be further divided. If the current block texture has low complexity, then we skip the calculation of RDO and directly decide that no further division is needed; if the current block has high texture complexity, then we should divide it into small ones to balance the compression and image quality. In this case, we can skip the calculation of RDO and directly proceed to the partition process. For the blocks with uncertain texture complexities, further division is determined by calculating the RD cost. In this way, many unnecessary calculations and comparisons of RD cost can be skipped, thereby reducing the computational complexity and saving time.

In the prediction mode decision algorithm, we add a number of new decisions in the RMD and MPM processes before RDO (Fig. 6) according to the vertical and horizontal texture directions calculated in the previous step. The angle candidate mode is reduced from 33 to 17, and then the 17 modes are divided into five groups to further determine which group is likely to be the optimal mode. Finally, we add the planner and DC modes, determine the candidate mode with small RD cost and two optimal candidate modes, and finally compare and add them to the MPM. Compared with the original 35 to 8/3 algorithms, the original HM algorithm must traverse 35 modes to determine the optimal algorithm. The proposed algorithm directly reduces the candidate mode by half by determining the texture direction and then further filtering the remaining half to determine the optimal mode.

The test conditions of the proposed algorithms show good agreement with those in [21]and are based on the

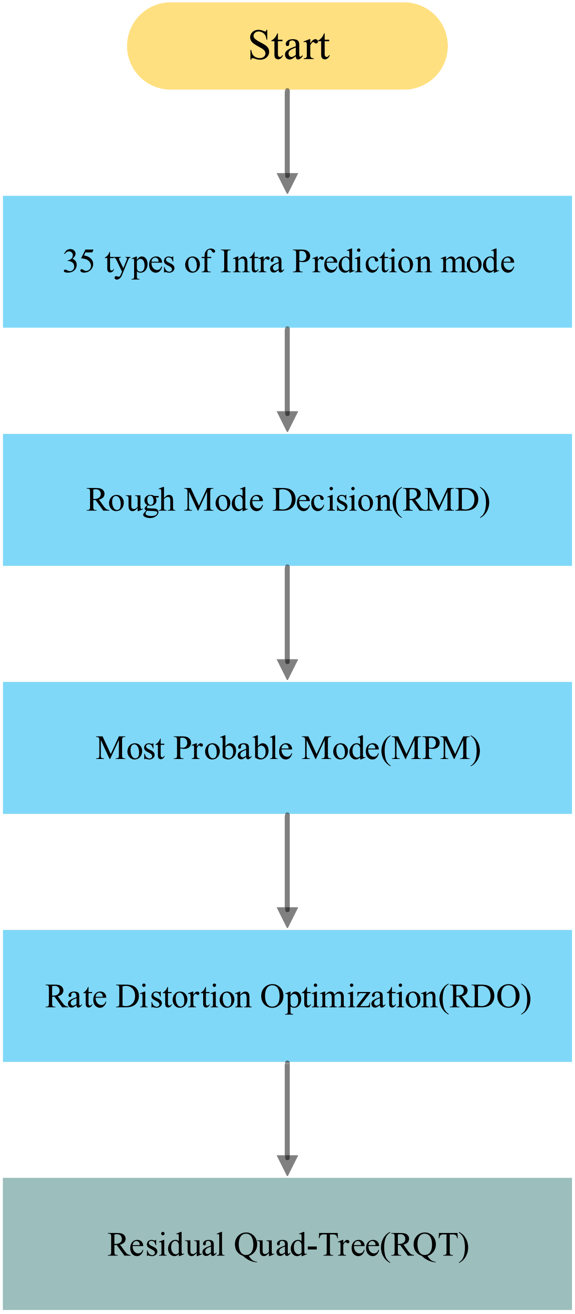

Fig. 6 CU size and mode decision in HEVC intra mode

latest HM16.16 for research. The following section will discuss the two proposed algorithms.

3.2 Texture complexity-based CU size decision algorithm

3.2.1 Measure of image texture complexity

The proposed CU size algorithm for 360 videos achieves improved performance by calculating image texture complexity. An image texture complexity is an important concept in image processing. Several metrics have been proposed to describe image texture, and they include local binary patterns, Markov random field, and gray-level co-occurrence matrix. The above methods are widely used because of their accuracy and precision. We tried to use them in HM 16.16, but we found that only one frame of video has taken several times longer than the original HM 16.16. The long computation time of these methods leads to difficulties in encoding real-time videos. Through experimental comparison, we chose another more appropriate metric. The mean of the absolute difference (MAD) measures image texture, which can balance the calculation time cost and the accuracy of describing the texture complexity. The MAD of an image is calculated as follows:

$$\mathrm{MAD}=\frac{1}{n^2}\sum_{y=0}^{n-1}\sum_{x=0}^{n-1}\left|p(x,y)-\overline{m}\right| \tag{1}$$

For the CTU of a 360 video, the MAD of the horizontal texture of the image pixels is smoother than that of the vertical texture. The horizontal stretch of the 360 video in ERP format is greater than the vertical stretch. Directly calculating the MAD of the entire CTU cannot accurately represent texture complexity. A CTU may have the characteristics found in Fig. 7 and thus shows a simple texture. However, when we calculate the MAD of the entire block, the result is relatively large, leading to an inaccurate texture complexity of the CTU. Therefore, to represent the texture complexity of the CTU, we adjust the MAD formula and calculate the horizontal MAD (HMAD) and vertical MAD (VMAD) for each row separately.

$$\mathrm{VMAD}_i=\sum_{y=0}^{n-1}\left|p(i,y)-\overline{m}_i\right| \quad (0\le i\le n) \tag{2}$$

$$\mathrm{HMAD}_j=\sum_{x=0}^{m-1}\left|p(x,j)-\overline{m}_j\right| \quad (0\le j\le m) \tag{3}$$

Then, we calculate the average VMAD and HMAD as follows:

$$\mathrm{meanVMAD}=\frac{1}{n}\sum_{0}^{n-1}\mathrm{VMAD}_i \quad (0\le i\le n) \tag{4}$$

$$\mathrm{meanHMAD}=\frac{1}{m}\sum_{0}^{m-1}\mathrm{HMAD}_j \quad (0\le j\le m) \tag{5}$$

The decision of the thresholds is further described below.

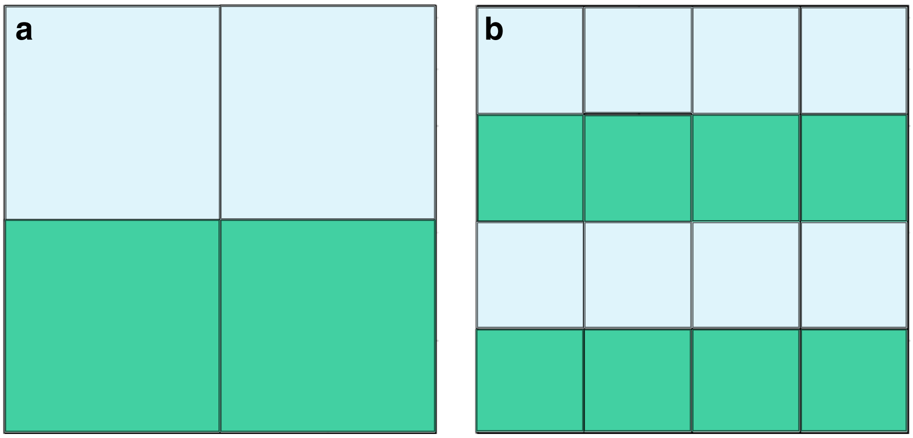

Fig. 7 CTUs with smooth texture features **a**. image with a simple horizontal texture **b**. image with multiple lines of simple horizontal orientation texture

3.2.2 Flow of the proposed algorithm

The accuracy of the CU partition of the proposed algorithm largely depends on the threshold setting. Therefore, selecting an appropriate threshold is key for the proposed algorithm. Two thresholds, namely, α and β, are defined here. α is used to represent the minimum threshold of the complex image texture, and β represents the maximum homogeneous threshold of the image texture. We render the following improvements on the basis of the original HM 16.16. Figure 8 shows the process of this algorithm.

Through the preset thresholds, we can divide the CU into three parts.

(a) When $\mathrm{meanHMAD} < \alpha$, that is, the texture information of the CTU block is homogeneous, further calculation of the RD cost comparison between the CTU block and its four sub-blocks is skipped, and the current depth is directly determined as the optimal depth of the CTU.

(b) When $\mathrm{meanHMAD} > \beta$, that is, the texture information of the CTU block is complex, further calculation of the RD cost comparison between the CTU block and its four sub-blocks is skipped, and the current depth is directly determined as the non-optimal depth and thus requires further division.

(c) When $\alpha < \mathrm{meanHMAD} < \beta$, the RD cost should be calculated to determine whether to proceed with further division.

3.2.3 VMAD/HMAD threshold parameters

To ensure that the thresholds are appropriately set, we select a large number of frames of several sequences to perform a statistical analysis and analyze the thresholds. In the appropriate range, a large number of statistical tests are conducted. In order to avoid yield overfitting, we did not select all the test sequences, we selected a sequence for the 4K, 6K, and 8K sequences for statistics respectively. We judge the texture complexity of the CU, the selected statistic set already contains enough samples (64×64, 32×32, 16×16, and 8×8 CU samples are 14,600, 58,400, 233,600, and 934,400 respectively). We find that the range of texture complexity thresholds obtained based on different videos is similar, and the statistical results are shown in Fig. 9. We then use the obtained thresholds to test other test sequences and find that the threshold can measure not only texture complexity for statistical video sequences, but also other unstated video sequences. As shown in Fig. 9, using the CU partition with depths of 0, 1, and 2 and quantization parameters (QP) of 27 and 37 of the first frame of all 360 video sequences as examples, we separately calculate the HMAD of the CU blocks that need to be further divided and the HMAD of the CU blocks that can be skipped. Through the statistical analysis, we find that the HMAD value of the CU that does not need to be divided is generally always smaller than the HAMD value of the CU that needs to be divided. As the depth increases, the number of CUs that need to be divided decreases, and the number of CUs that do not need to be divided increases. When the HMAD is greater than a certain value, the CU must be divided. Similarly, when the HMAD is less than a certain value, this CU does not need to be divided. By accurately setting the thresholds, the proposed algorithm could decide in advance whether to proceed with the division, thereby improving the coding efficiency.

We find that for different QP and CU depths, different thresholds should be selected (Fig. 10). For different CU depths and QPs, α maintains a relatively small fluctuation range. For β, different CU depths have a remarkable impact on the threshold decision.

We select one frame in these test sequences and separately calculate the CU partition of the original HM 16.16 and the CU partition of the proposed algorithm. As

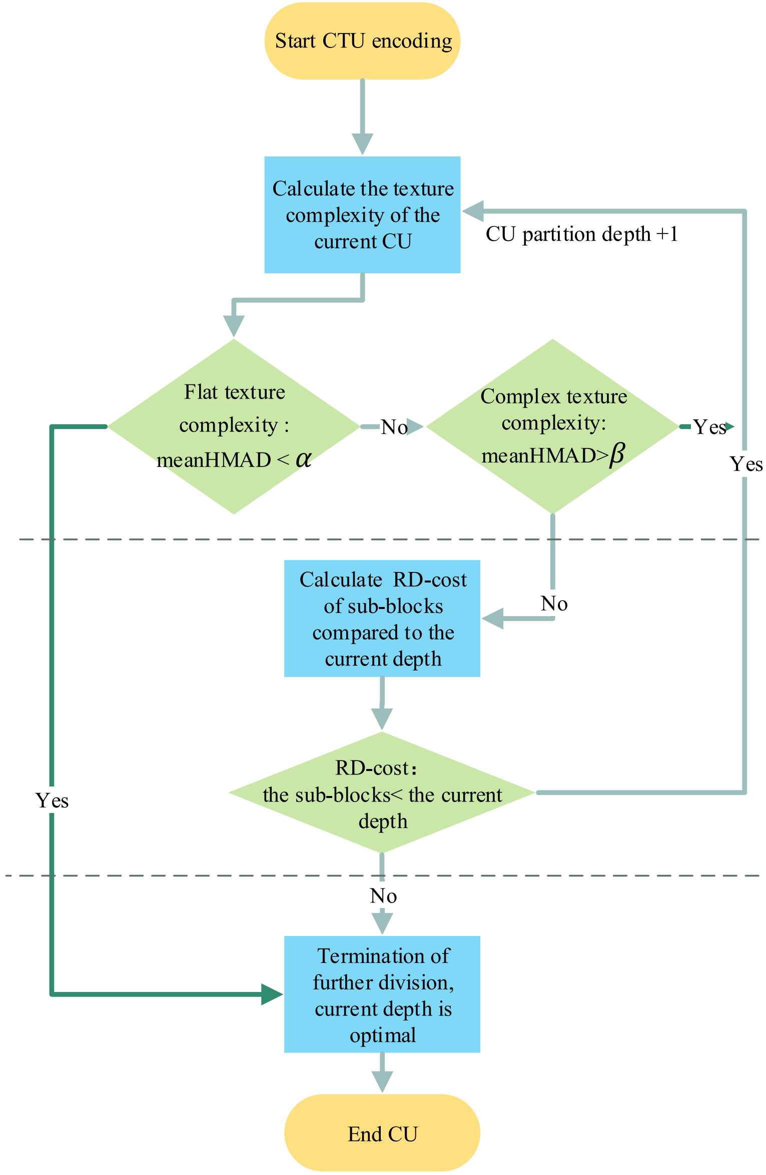
Start CTU encoding
Calculate the texture complexity of the current CU
CU partition depth +1
Flat texture complexity : meanHMAD < α
No
Complex texture complexity: meanHMAD>β
Yes
Yes
Calculate RD-cost of sub-blocks compared to the current depth
No
RD-cost: the sub-blocks< the current depth
Yes
No
Termination of further division, current depth is optimal
End CU

Fig. 8 Schematic of the proposed CU size decision algorithm based on texture complexity

shown in Tables 1 and 2, the similarity rate of the proposed CU size decision algorithm is above 90%.

Figure 11 shows the CU segmentation result of the original HM 16.16 and of the proposed algorithm. As shown in the figure, the CU partition of the proposed algorithm is basically the same as the HM 16.16 and has high accuracy. Furthermore, the proposed CU size decision algorithm performs better than the original HM 16.16 in some homogeneous blocks that not need to be divided and in complex blocks that need further partitioning. For example, in Fig. 11d, the roof texture in the lower left corner is complex; the proposed algorithm makes a more detailed division of these CU blocks. The sky texture in the upper right corner is simple; compared with the original HM 16.16, the proposed algorithm for these CU blocks division is larger (i.e., the depth of CU is smaller). It can also be seen in Fig. 11h that the CU blocks of the cable that need to be subdivided is more finely divided, and the partitioning of CU blocks with simple texture is larger.

3.3 Texture direction-based prediction mode decision algorithm

VMAD and HMAD determine the texture complexity and texture directionality of the image. The smaller the MAD in the horizontal or vertical direction, the lower the texture complexity in this direction. This condition indicates that the prediction mode in this direction involves a small RD cost. To resolve the problem of the

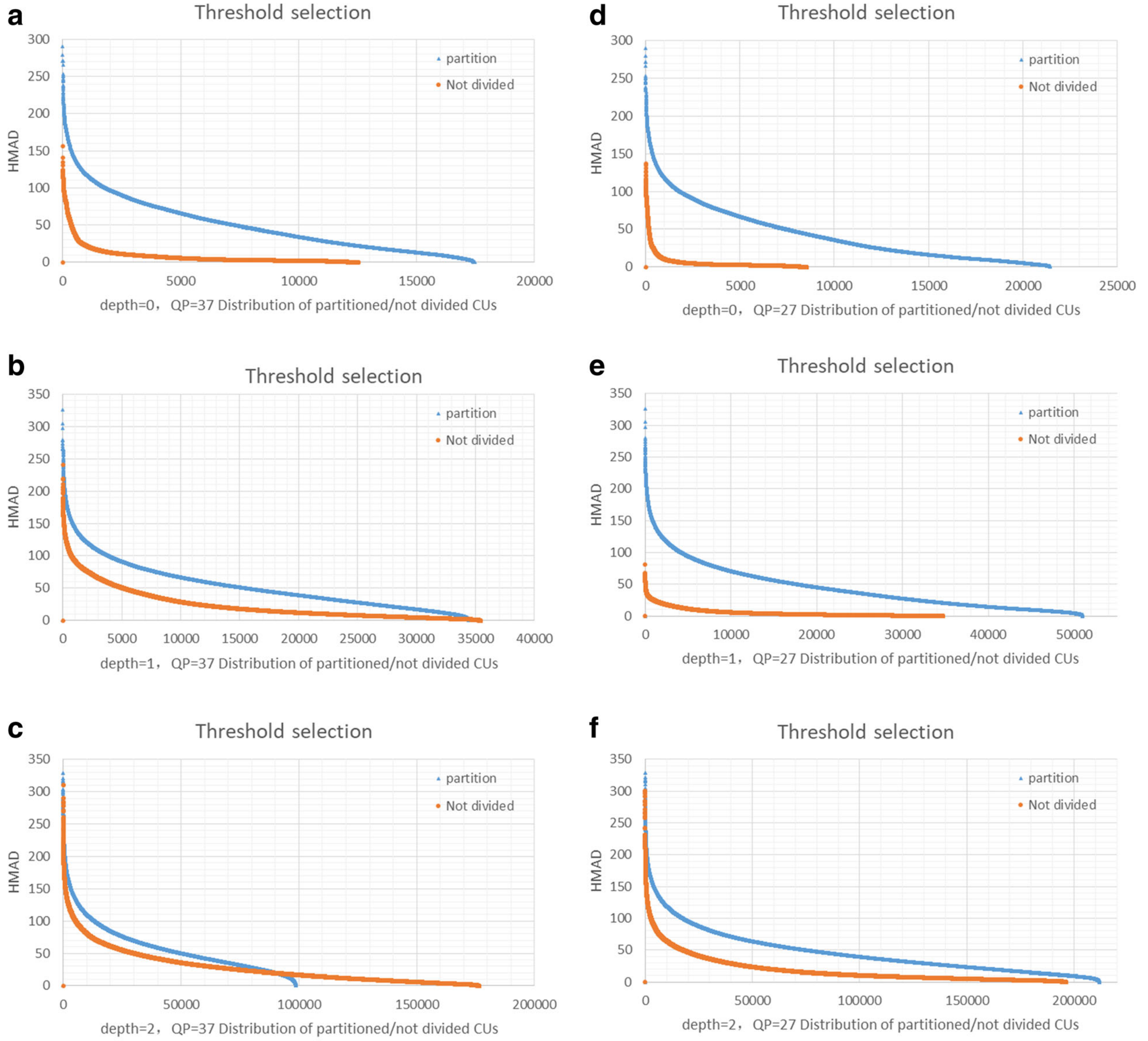

Fig. 9 HMAD maps of CUs that can be skipped and further divided. **a** CU division of depth = 0, QP = 37. **b** CU division of depth = 1, QP = 37. **c** CU division of depth = 2, QP = 37. **d** CU division of depth = 0, QP = 27. **e** CU division of depth = 1, QP = 27. **f** CU division of depth = 2, QP = 27

large computation in the RDO process for a 360 video and to reduce the large number of candidate modes in RMD, we propose to use texture directionality in RMD and MPM. By reducing the candidate modes, the proposed algorithm can further reduce the computational complexity in RDO.

The original HM 16.16 for 360 videos traverses 35 types of prediction modes, calculates the RD cost to sort the optimal mode, and then selects three (when PU or TU size is 32 × 32 or 64 × 64) or eight (when PU or TU size is 4 × 4, 8 × 8, or 16 × 16) modes. We then compare these modes with MPM to determine the best prediction model. Many candidate modes in RMD can be excluded in advance. Therefore, we can use the VMAD and HMAD calculated in the proposed CU mode decision algorithm to divide 33 angular predictions into horizontal (2–18) and vertical (18–34) modes according to the texture directionality (Fig. 12). We label the yellow part of the figure as the horizontal mode and the red part as the vertical mode.

$$\text{a) If VMAD} > \text{HMAD}, C1 = \text{horizontal mode} \tag{6}$$

$$\text{b) If VMAD} \leq \text{HMAD}, C1 = \text{vertical mode} \tag{7}$$

When classifying the vertical and horizontal textures, we first reduce the 35 modes to 19 (including 0, 1 mode), further reduce the 19 modes to obtain the two

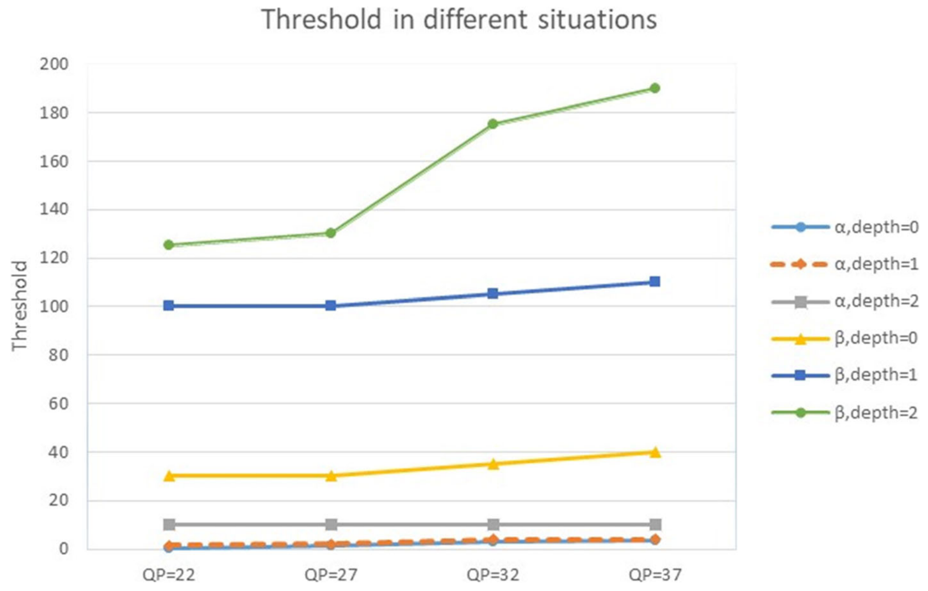

Fig. 10 Threshold range of *α* and *β* of HMAD

best candidate modes, and finally add the candidates to MPM. Candidate modes can be divided into three layers and are calculated layer by layer. In Fig. 12, the mode marked with a yellow line is the horizontal mode C2, and the red line is the vertical mode C2 (Table 3).

Take for example the horizontal direction. Before the RMD, we first determine whether the candidate mode is horizontal or vertical through the first level C1. Then, we consider the five adjacent modes as a whole, calculate the RD cost of the five representative prediction modes (2, 6, 10, 14, 18) in the second level C2, and finally obtain the smallest and most suitable C3 (for example, the RD cost of mode 6 in C2 is the smallest, and the candidate modes of C3 are 4, 5, 7, and 8). We then traverse C3 to obtain the optimal mode. With this layering, the number of candidate modes to be calculated is reduced. Notably, in the actual calculation, each layer needs the addition of modes 0 and 1. The specific process is shown in Fig. 13.

Through this proposed algorithm, we effectively reduce the number of candidate modes. Table 4 lists the number of candidate modes of the proposed algorithm and the original HM 16.16 in RMD and RDO.

The proposed algorithm first determines the horizontal or vertical mode and reduces the candidate mode to seven (five types of C2 + modes 0 and 1). If modes 0 and 1 are the two best modes, then we add these two modes to RDO. If modes 0 and 1 are not the two best modes, then the mode in C3 is further calculated, and the two best candidate modes are finally added to RDO.

In the process of achieving real-time transmission, the encoding time must be controlled within a certain range. Through experiments, we find that the combination of the two proposed image texture-based algorithms can effectively reduce coding time with a minimal loss of the BD rate.

4 Experimental results and discussion

The effectiveness of the proposed algorithm is determined on the basis of the latest HM16.16. The experimental hardware used is Intel Core i7-7700 CPU @ 3.60 GHz with 8.0 GB RAM. The test sequence includes 4K, 6K, 8K, and 8K 10-bit sequences, which are provided by JVET [22]. The test sequences used in this study are from GoPro [23], InterDigital [24], LetinVR [25], and Nokia [26], and they have been recommended by CTC

Table 1 Comparison of the division of threshold α and HM 16.16

	QP = 22	QP = 27	QP = 32	QP = 37
Depth = 0	95%	92%	92%	96%
Depth = 1	92%	99%	95%	97%
Depth = 2	98%	98%	98%	99%
Average similarity rate				96%

Table 2 Comparison of the division of threshold *β* and HM 16.16

	QP = 22	QP = 27	QP = 32	QP = 37
Depth = 0	98%	94%	96%	95%
Depth = 1	96%	99%	90%	90%
Depth = 2	93%	90%	90%	91%
Average similarity rate				94%

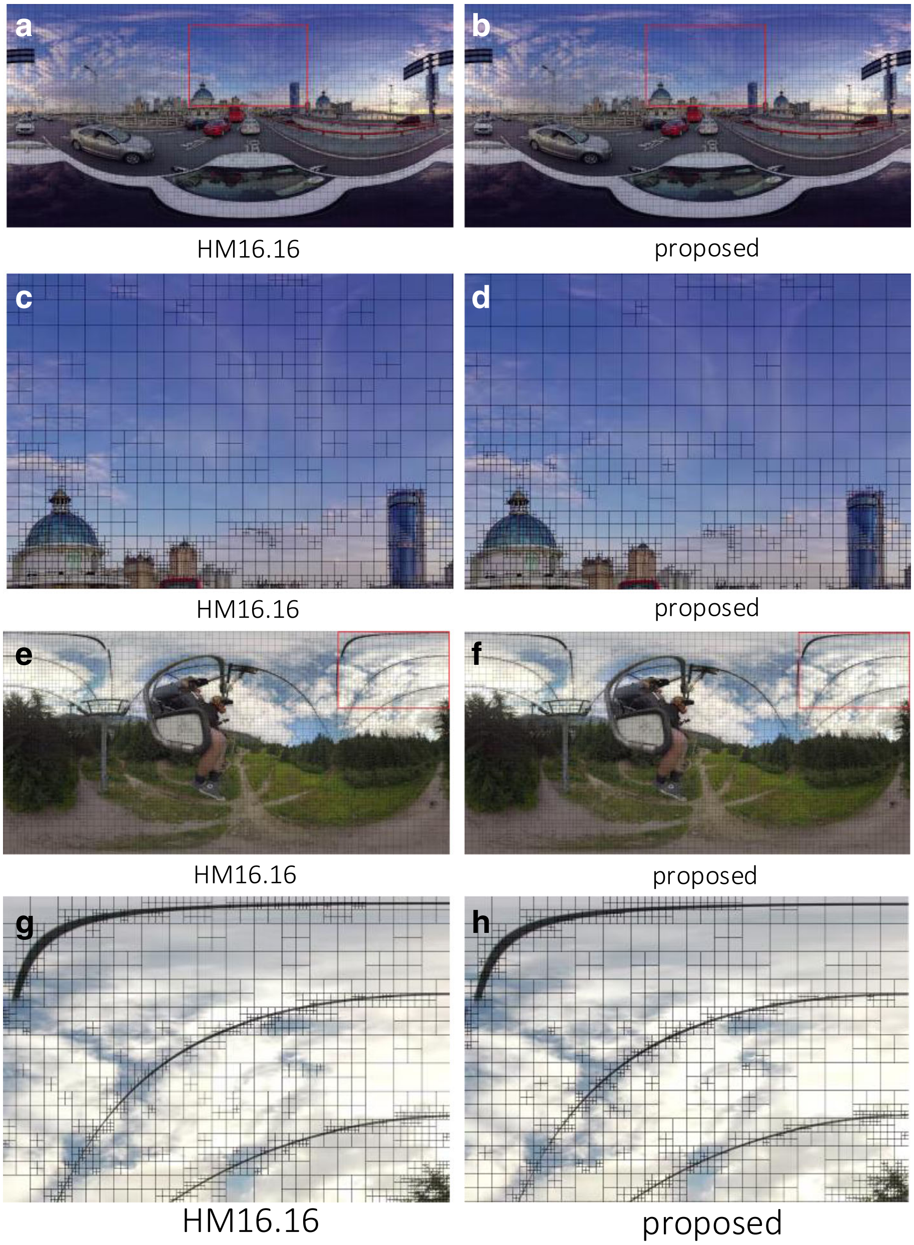

Fig. 11 Comparison of CU partition line images of the proposed algorithm and HM 16.16. **a**, **e** CU partition line image of HM 16.16. **b**, **f** CU partition line image of the proposed algorithm. **c**, **g** Partially enlarged CU partition line image of HM 16.16. **d**, **h** Partially enlarged CU partition line image of the proposed algorithm

as test sequences for the 360° video. All 360 videos with different resolutions are tested to validate the proposed algorithm. For each video sequence, four quantization parameter values are used: 22, 27, 32, and 37. We use the encoder configuration All Intra and BD rate to measure the quality of the algorithm. The PSNR uses three parameters: WSPSNR, S_PSNR_NN, and PSNR. WSPSNR and S_PSNR_NN are the two new quality evaluation standards proposed for 360 videos [4]. According to the characteristics of a 360 video, its quality can be accurately measured using WSPSNR and S_PSNR_NN. Time reduction is calculated by:

$$\Delta T = \frac{T_{\mathrm{HM16.16}} - T_{\mathrm{proposed}}}{T_{\mathrm{HM16.16}}} \times 100\% \qquad (8)$$

The experimental results are shown in Tables 5, 6, and 7. The BD rate_Y1–Y3 in the table represents the values calculated using WSPSNR Y, S_PSNR_NN Y, and PSNR Y.

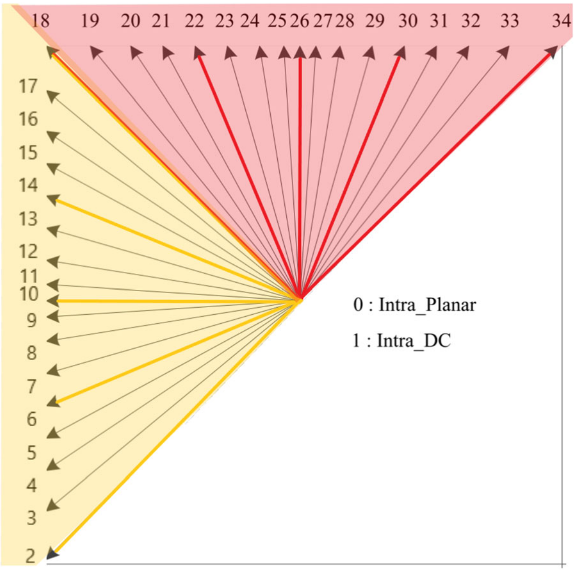

Fig. 12 Vertical and horizontal mode decisions

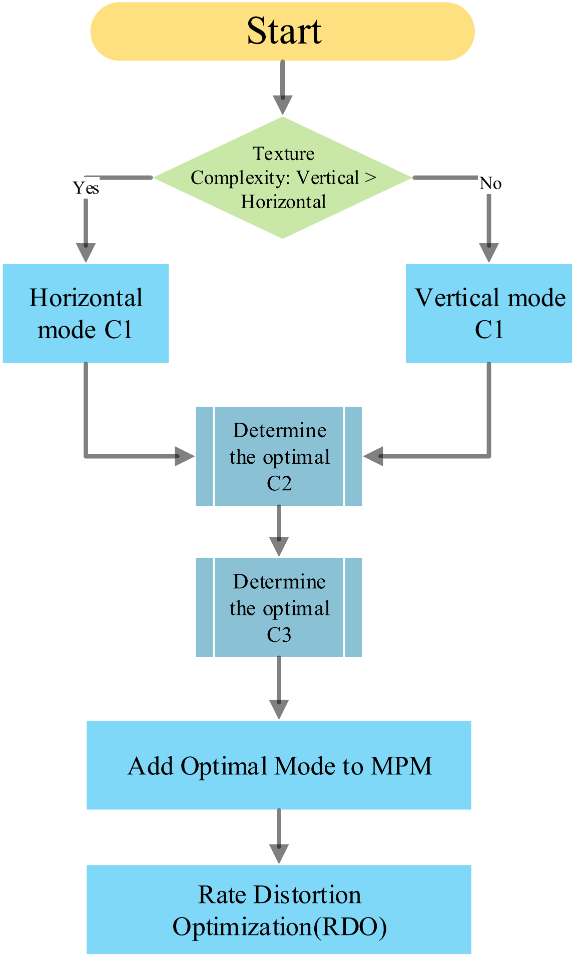

Fig. 13 Schematic of the proposed prediction mode decision algorithm based on texture direction

Table 5 shows the experimental results using the proposed CU size decision algorithm based on texture complexity. Table 6 shows the experimental results using the proposed prediction mode decision algorithm based on texture direction. Table 7 shows the experimental results using the two proposed texture characteristic algorithms.

As shown in Table 5, the original HM 16.16 and the proposed CU size decision algorithm show different performances for different sequences, particularly with regard to improving encoding time. For the sequences with relatively simple textures, the proposed CU size decision algorithm performs excellently in shortening the encoding time, with the highest improvement reaching 44%. For several sequences with relatively complex textures, the improvement of encoding complexity is limited.

Table 6 shows the performance comparison between the proposed prediction mode decision algorithm and the original algorithm for different video sequences under different QPs. According to the simulation results, the proposed intra prediction mode decision algorithm can reduce computational complexity by 33% on average, with the BD rate slightly increasing. Therefore, the proposed intra prediction mode decision algorithm reasonably and effectively reduces the number of candidate modes for RMD and RDO processing.

Compared with HM16.16 (Table 7), the proposed algorithm can save time at an average of 53% (up to 59%), with bitrate loss being 1.3%. For the two proposed algorithms, we use the same parameters of texture complexity mentioned in the CU size decision algorithm. As shown in Fig. 14, we select 360 video sequences: 4K-8 bit (DrivingInCountry), 6K-8 bit (Balboa), 8K-8 bit (KiteFlite), and 8K-10 bit (ChairliftRide), and compare the

Table 3 Horizontal and vertical candidate modes for each level

	First level mode set C1	Second level mode set C2	Third level mode set C3
Horizontal mode	2–18	2, 6, 10, 14, 18	C2−2, C2−1, C2+1, C2+2
Vertical mode	18–34	18, 22, 26, 30, 34	C2−2, C2−1, C2+1, C2+2

Table 4 Number of candidate modes of the proposed algorithm and original HM 16.16

	RMD	RDO
HM	35	3+MPM/8+MPM
Proposed	7/11	2+MPM

Table 5 Performance comparison between the original HM 16.16 and the proposed algorithms

Resolution	Sequences	BD-Rate_Y1	BD-Rate_Y2	BD-Rate_Y3	△Time
4K 8bit 3840 × 1920	AerialCity	0.1%	0.1%	0.1%	34%
	DrivingInCity	0.2%	0.2%	0.2%	44%
	DrivingInCountry	0.0%	0.0%	0.1%	31%
	PoleVault	0.1%	0.1%	0.1%	29%
6K 8bit 6144 × 3072	Balboa	0.2%	0.2%	0.3%	43%
	BranCastle	0.2%	0.2%	0.2%	28%
	Broadway	0.4%	0.4%	0.4%	44%
	Landing	0.1%	0.1%	0.2%	40%
8K 8bit 8192 × 4096	GasLamp	0.4%	0.4%	0.4%	35%
	Harbor	0.2%	0.2%	0.2%	19%
	KiteFlite	0.1%	0.1%	0.1%	20%
	SkateboardTrick	0.1%	0.1%	0.1%	32%
	Train	0.1%	0.0%	0.1%	37%
	Trolley	0.2%	0.2%	0.1%	21%
8K 10bit	SkateboardInLot	0.1%	0.1%	0.2%	41%
	ChairliftRide	0.1%	0.1%	0.1%	24%
Average		0.2%	0.2%	0.2%	33%

bitrates using the proposed algorithm and the original HM 16.16. The RD curves of the sequences are almost the same as those of the original encoder. Thus, the proposed algorithm offers advantages in terms of complexity and coding efficiency. The computational complexity is reduced while the BD rate increases insignificantly at an average of 1.3% (WSPSNR Y, S_PSNR_NN Y, and PSNR Y).

In the latest research into 360 video coding, no similar intra algorithm has been proposed. To verify the performance of the proposed algorithm, we experimented with all 360 sequences to test the performance of the algorithm. The proposed algorithm could tremendously reduce the amount of calculation and time complexity, with the highest reduction reaching 59% and with the BD rate only increasing by 1.3% and is thus negligible.

Table 6 Performance comparison between the original HM 16.16 and the proposed mode algorithm

Resolution	Sequences	BD-Rate_Y1	BD-Rate_Y2	BD-Rate_Y3	△Time
4K 8bit 3840 × 1920	AerialCity	1.1%	1.1%	1.1%	37%
	DrivingInCity	1.3%	1.3%	1.2%	37%
	DrivingInCountry	0.8%	0.8%	0.7%	40%
	PoleVault	1.1%	1.1%	1.0%	37%
6K 8bit 6144 × 3072	Balboa	1.1%	1.1%	1.0%	33%
	BranCastle	0.9%	0.9%	0.8%	34%
	Broadway	1.4%	1.4%	1.3%	31%
	Landing	1.5%	1.5%	1.4%	28%
8K 8bit 8192 × 4096	GasLamp	2.1%	2.1%	1.9%	27%
	Harbor	1.9%	1.9%	1.7%	32%
	KiteFlite	1.1%	1.1%	1.0%	31%
	SkateboardTrick	1.0%	1.0%	1.0%	35%
	Train	0.4%	0.4%	0.4%	33%
	Trolley	0.8%	0.8%	0.7%	32%
8K 10bit	SkateboardInLot	1.3%	1.3%	1.1%	34%
	ChairliftRide	0.9%	0.8%	0.7%	34%
Average		1.2%	1.2%	1.1%	34%

Table 7 Performance comparison between the original HM 16.16 and the proposed overall algorithm

Resolution	Sequences	BD-Rate_Y1	BD-Rate_Y2	BD-Rate_Y3	▵Time
4K 8bit 3840 × 1920	AerialCity	1.2%	1.2%	1.1%	54%
	DrivingInCity	1.5%	1.5%	1.4%	59%
	DrivingInCountry	0.8%	0.8%	0.7%	53%
	PoleVault	1.2%	1.3%	1.1%	51%
6K 8bit 6144 × 3072	Balboa	1.3%	1.3%	1.3%	57%
	BranCastle	1.1%	1.2%	1.1%	46%
	Broadway	1.8%	1.8%	1.8%	53%
	Landing	1.7%	1.6%	1.6%	52%
8K 8bit 8192 × 4096	GasLamp	2.6%	2.6%	2.4%	51%
	Harbor	2.1%	2.1%	1.9%	52%
	KiteFlite	1.2%	1.3%	1.1%	44%
	SkateboardTrick	1.1%	1.1%	1.1%	59%
	Train	0.5%	0.4%	0.5%	56%
	Trolley	1.0%	1.0%	0.9%	46%
8K 10bit	SkateboardInLot	1.4%	1.4%	1.2%	54%
	ChairliftRide	1.0%	1.0%	0.9%	47%
Average		1.3%	1.3%	1.3%	53%

5 Conclusion

A fast intra prediction algorithm based on texture characteristics for 360 videos was proposed. Two metrics, namely, VMAD and HMAD, were used to measure the texture characteristics of a CU in vertical and horizontal directions, respectively. A fast CU size decision algorithm based on texture complexity was proposed to reduce the computation complexity of RDO. According to the two metrics, a fast mode decision algorithm was designed. This algorithm dramatically reduced the number of candidate modes from 35 to 7/11 in the RMD process and the number of candidate modes from 8 to 2 in the RDO process. The experimental results showed that the proposed algorithm could relatively reduce encoding time to achieve savings of up to 53% on average while only incurring negligible loss of the BD rate and video quality.

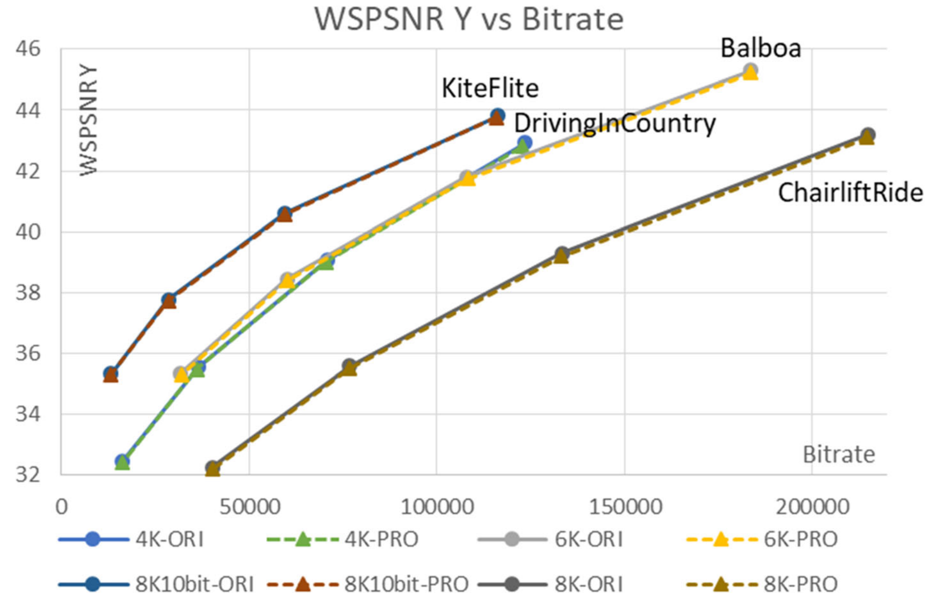

Fig. 14 Comparison of RD curves of different resolution sequences under different QPs (22, 27, 32, 37) between the original HM 16.16 and the proposed overall algorithm

Abbreviations

BD rate: Bjontegaard delta rate; CMP: Cubemap projection; CTU: Coding tree unit; CU: Coding unit; ERP: Equirectangular projection; HEVC: High Efficiency Video Coding; HM: HEVC test model; HMAD: The horizontal absolute difference; JVET: Joint Video Exploration Team; MAD: The absolute difference; MPEG: Moving Picture Exports Group; MPMs: Most probable modes; OHP: Octahedron projection; PSNR: Peak to signal noise ratio; PU: Prediction unit; QP: Quantization parameters; RD cost: Rate distortion cost; RD: Rate distortion; RDO: Rate distortion optimization; RMD: Rough mode decision; RQT: Residual quad-tree; RSP: Rotated sphere projection; S_PSNR_NN: Spherical PSNR without interpolation; SSP: Segmented sphere projection; TSP: Truncated square pyramid projection; TU: Transform unit; VCEG: Video Coding Experts Group; VMAD: Vertical absolute difference; VR: Virtual reality; WSPSNR: PSNR weighted by sample area

Acknowledgements

Not applicable.

Funding

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).

Availability of data and materials

The conclusion and comparison data of this article are included within the article.

Authors' contributions

MZ proposed the framework of this work, and XD carried out the whole experiments and drafted the manuscript. ZL offered useful suggestions and helped to modify the manuscript. All authors read and approved the final manuscript.

Authors' information

MMZ: Doctor of Engineering, professor, master instructor, master of Communication and Information Systems. His major research interests include the video codec, embedded systems, image processing, and pattern recognition. He has authored or co-authored more than 40 refereed technical papers in international journals and conferences in the field of video coding, image processing, and pattern recognition. He holds 21 national patents and 2 monographs in the areas of image/video coding and communications.

XSD: Studying master of North China University of Technology. Her major research is HEVC.

ZL: Doctor of Engineering, master instructor. He 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. His major research interests include the video codec, pattern recognition, and self-organizing network.

FQM: Master of engineering, He 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.

WY: Doctor of Engineering, professor, doctoral supervisor. His major research interests include mechanical tribology and surface technology, exploration technology and geological drilling, diamond and other superhard materials.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1]North China University of Technology, Beijing 100144, China. [2]China University of Geosciences, Beijing 100083, China.

Received: 9 October 2018 Accepted: 20 March 2019
Published online: 02 April 2019

References

1. Yuwen He, Xiaoyu Xiu, Yan Ye et al. "360Lib Software Manual", Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET 360Lib Software Manual, 2017
2. Y. Lu, *Jisheng Li, Ziyu Wen, Xianyu Meng, "AHG8: Padding Method for Segmented Sphere Projection", Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 7th Meeting: Torino, IT, 13 July-21* (2017)
3. G.J. Sullivan, J.R. Ohm, W.J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)
4. Y. Ye, E. Alshina, J. Boyce, *"Algorithm Descriptions of Projection Format Conversion and Video Quality Metrics in 360Lib", Joint Video Exploration Team of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-G1003, 7th Meeting* (2017)
5. M. Tang, Z. Yu, J. Wen, S. Yang, *"Optimized Video Coding for Omnidirectional Videos", IEEE International Conference on Multimedia and Expo (ICME)* (2017), pp. 799–804
6. W.-T. Lee, H.-I. Chen, M.-S. Chen, et al., High-resolution 360 video Foveated stitching for real-time VR. Computer Graphics Forum **36**(7), 115–123 (2017)
7. N. Kim, J.-W. Kan, Bi-directional deformable block-based motion estimation for frame rate-up conversion of 360-degree videos. Electron. Lett. **53**(17), 1192–1194 (2017)
8. Y. Li, J. Xu, Z. Chen, *"Spherical Domain Rate-Distortion Optimization for 360-Degree Video Coding", IEEE International Conference on Multimedia and Expo (ICME)* (2017), pp. 709–714
9. H. Bai, C. Zhu, Y. Zhao, Optimized multiple description lattice vector quantization for wavelet image coding. IEEE Trans. Circuits Syst. Video Technol **17**(7), 912–917 (2017)
10. H. Bai, W. Lin, M. Zhang, A. Wang, Y. Zhao, Multiple description video coding based on human visual system characteristics. IEEE Trans. Circuits Syst. Video Technol **24**(8), 1390–1394 (2014)
11. D.G. Fernández, A.A. Del Barrio, G. Botella, C. García, *"Fast CU Size Decision Based on Temporal Homogeneity Detection", Conference on Design of Circuits and Integrated Systems (DCIS)* (2016), pp. 1–6
12. M. Zhang, S. Dou, *Zhi liu, "Early CU size Determination Based on Image Complexity in HEVC", 2017 Data Compression Conference (DCC)* (2017), pp. 474–474
13. Y. Yao, Y.L. Xiaojuan Li, Fast intra mode decision algorithm for HEVC based on dominant edge assent distribution. Multimed. Tools Appl. **75**, 1963–1981 (2016)
14. Y. Huan, *Huabiao Qin, The Optimization of HEVC Intra Prediction Mode Selection, 2017 4th International Conference on Information Science and Control Engineering (ICISCE)* (2017), pp. 1743–1748
15. L. Shen, Z. Zhang, P. An, Fast CU size decision and mode decision algorithm for HEVC intra coding. IEEE Trans. Consum. Electron. **59**, 207–213 (2013)
16. X. Shang, G. Wang, T. Fan, Y. Li, *"Fast CU Size Decision and PU Mode Decision Algorithm in HEVC Intra Coding", IEEE International Conference on Image Processing (ICIP)* (2015), pp. 1593–1597
17. M. Zhang, X. Zhai, Z. Liu, Fast and adaptive mode decision and CU partition early termination algorithm for intra-prediction in HEVC. EURASIP Journal on Image and Video Processing **1**, 86–97 (2017)
18. Y.C. Ting, T.S. Chang, *"Gradient-Based PU Size Selection for HEVC Intra Prediction", IEEE International Symposium on Circuits and Systems (ISCAS)* (2014), pp. 1929–1932
19. H. Huang, F. Wei, *"Fast Algorithm Based on Edge Density and Gradient Angle for Intra Encoding in HEVC", IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)* (2016), pp. 347–351
20. J.M. Ha, J.H. Bae, M.H. Sunwoo, *"Texture-Based Fast CU Size Decision Algorithm for HEVC Intra Coding", 2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)* (2016), pp. 702–705
21. J. Boyce, E. Alshina, A. Abbas, *"JVET Common Test Conditions and Evaluation Procedures for 360° Video", Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-H1030, 8th Meeting* (2017)
22. Test sequences are available on ftp://jvet@ftp.ient.rwth-aachen.de and ftp://jvet@ftp.hhi.fraunhofer.de in the /testsequences/testset360 directory. Accredited members of VCEG and MPEG may contact the JVET chairs for login information
23. A. Abbas, B. Adsumilli, *"New GoPro Test Sequences for Virtual Reality Video Coding", Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-D0026, 4th Meeting* (2016)

24. E. Asbun, Y. He, Y. He, Y. Ye, *"AHG8: InterDigital Test Sequences for Virtual Reality Video Coding", Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-D0039, 4th Meeting* (2016)
25. R. Guo, W. Sun, *"Test Sequences for Virtual Reality Video Coding from LetinVR", Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-G0053, 7th Meeting* (2017)
26. S. Schwarz, A. Aminlou, I.D.D. Curcio, M.M. Hannuksela, *"Tampere Pole Vaulting Sequence for Virtual Reality Video Coding", Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-D0143, 4th Meeting* (Oct. 2016)