Overview of Current Development in Depth Map Coding of 3D Video and its Future

Yui-Lam Chan [1*], Chang-Hong Fu[2], Hao Chen [2], Sik-Ho Tsang [1]

[1] Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China
[2] School of Electronic and Optical Engineering, Nanjing University of Science and Technology, No. 200 Xiao Ling Wei Street, Nanjing, China
[*] enylchan@polyu.edu.hk

Abstract: 3D videos have attracted lots of attention from academia and industry after great success in the film industry. Multi-view Video plus Depth (MVD) is the most popular 3D video format to provide vivid 3D feeling and has been adopted as an international 3D video coding standard, namely 3D-HEVC. MVD includes a limited number of textures and depth maps to synthesize virtual views. In MVD, depth samples describe the distance between a camera and an actual object as a grey-level image. The characteristics of depth maps are quite different from texture images. Consequently, new coding tools are designed for depth maps in 3D-HEVC to improve the coding efficiency at the expense of high computational complexity, which faces great challenges in coding systems. Depth map coding is also an important technique in immersive media to support 3DoF+/6DoF applications such as Virtual Reality/Augmented Reality (VR/AR). The paper starts with an overview of what has been done over the last decade in 3D-HEVC, especially depth map coding, regarding theories, methodologies, current research and development of state-of-the-art fast approaches such as machine learning, etc. Following this, a comprehensive comparison of the reviewed techniques is presented, and an outlook on its future trends is provided.

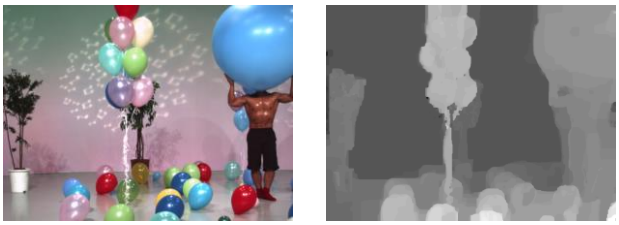
1. Introduction

3D videos provide an exciting viewing experience with the illusion of depth perception. It is not limited to a theatre nowadays. It also becomes increasingly interesting for home entertainment due to the proliferation of 3D television, Blu-ray 3D, VR headsets, etc. Pervasive applications include immersive video conference developed by Heinrich Hertz Institute (HHI) [1], 3D movie, Microsoft motion-sensing game (XBOX) [1], etc. In principle, 3D scene perception can be achieved by presenting two different videos to the viewer's left and right eyes simultaneously via 3D glasses [2].

Due to the user habit, the requirements of the 3D video in the home, however, are quite different from the theatre scenario, where an audience sits in a chair for a limited amount of time and paying full attention to the movie. In the theatre, stereo content is sufficient enough and the audience wears 3D glasses for watching. In contrast, there is a great variety of 3D displays designed for home users, starting from traditional 2-view stereo displays with glasses to more sophisticated auto-stereoscopic displays without glasses. In the auto-stereoscopic display, more than two views are displayed at a time. Recent auto-stereoscopic displays are capable of displaying *N* different views at the same time, of which only a stereo pair is visible from a specific location. We believe the auto-stereoscopic display will become dominant in a living home environment in the coming future. To support this, multiple views are either provided as input to the display, or these views are synthesized locally with the help of depth maps at the display. Owing to limitations in the production environment and bandwidth constraints [3], the latter approach is the current trend where only limited views and their corresponding depth maps are captured, compressed and transmitted for 3D video applications [4]-[5], and this 3D video format is referred to as Multi-view Video plus Depth (MVD) [6]-[7]. With the aid of depth information, it can synthesize an arbitrary number of views from these limited views via the Depth-Image-Based Rendering (DIBR) technology [8] at the decoder side.

In March 2011, MPEG issued a Call for Proposals (CfP) on 3D video technology [5] based on the state-of-the-art High Efficiency Video Coding (HEVC) standard [9]-[10]. The responses and the subjective tests demonstrated the superior of MVD, and it was then adopted as the 3D extension of HEVC (3D-HEVC). Consequently, the ISO/IEC MPEG and ITU-T Video Coding Experts Group (VCEG) standardization bodies established the Joint Collaborative Team on 3D Video Coding Extension (JCT-3V) to develop the next generation of an international 3D video coding standard in July 2012. Based on the HEVC [10], the first 3D-HEVC reference software was contributed by proponents that achieve the best performance to the CfP [5], [11]. In February 2015, 3D-HEVC was finally included in the 3rd version of HEVC which provides the joint coding tools of texture and depth map for various advanced 3D displays [12].

In MVD, a depth map is represented by a grey-level image, as shown in Fig. 1, which captures the distance between a camera and an actual object. By representing the depth information, a depth map plays a key role in the DIBR process and should be carefully coded. It is observed that the characteristics of a depth map are quite different from those of a texture image. First, a depth map has mostly smooth regions delimited by sharp edges [6], [13]. Second, the distortion of sharp edges induces ringing artifacts at object boundaries in synthesized views [14]. Different from 2D video, preserving the sharp edges rather than the visual quality becomes the most critical task for depth map coding in 3D-HEVC.



(a) (b)

Fig. 1. *Example of MVD video* ***(a)*** *texture video,* ***(b)*** *depth map*

Depth map coding in 3D-HEVC adopts a quadtree-based Coding Tree Unit (CTU) structure in HEVC. The CTU is a basic unit and is divided into several Coded Units (CUs) that can be represented by a recursive quadtree structure. An optimal CU partition within a CTU is a combination of different square sizes or Depth Level (*DL*) ranging from 64×64 (*DL=0*) to 8×8 pixels (*DL=3*). The CU is then split into several Prediction Units (PU), where the best prediction mode for each PU is selected from many candidates. In addition to the coding framework of HEVC, 3D-HEVC introduces many new coding tools especially for depth maps, which will be described in Section 2 and Section 3. These new coding techniques in 3D-HEVC yield significant coding efficiency and provide the outstanding perceptual quality of synthesized views. However, the coding complexity for depth maps is increased drastically, and fast depth map coding techniques are desirable for making 3D-HEVC practice.

The rest of this paper is organized as follows. The basic concept of the 3D-HEVC is given in Section 2. Section 3 then describes various coding techniques for depth maps in 3D-HEVC in detail. Section 4 introduces the recent fast algorithms for depth map coding. The emerging learning-based fast coding methods are then discussed in Section 5, together with our proposed decision tree-based depth coding algorithm. Finally, Section 6 depicts an outlook on the use of depth maps in immersive video coding.

2. System Structure of 3D-HEVC

In this Section, the MVD data format representing 3D video as well as the corresponding transmission system is described in Section 2.1. The 3D-HEVC coding system designed for MVD videos is then introduced in Section 2.2.

2.1. Transmission of 3D video

The Multi-View Video (MVV) data format is always used for 3D video representing, where each view requires one camera to capture the scene independently. Although the MVV data format can effectively represent one N-view video, the amount of data is extremely large to represent a remarkable number of views. In recent years, the DIBR technology is designed and introduced to synthesize virtual views. With DIBR, not all views must be transmitted since additional views can be synthesized by the limited textures and depth maps. As a result, MVD, which is composed of a small number of captured views with their associated depth maps, has replaced MVV due to its superior coding efficiency for 3D video representation. For one MVD video, a limited number of textures and their associated depth maps, as well as the camera parameters, are coded and multiplexed into one bitstream. After transmission. the additional views can then be synthesized at the decoder side.

The basic structure of the 3D video transmission system using the MVD data format is illustrated in Fig. 2. In general, one MVD video consists of multiple textures, the corresponding depth maps, and the camera parameters. These signals are all set as input signals of the 3D-HEVC encoder. After the coding process, the bitstream referring to the whole MVD is then sent to the transmission system. It is noted that each transmitted bitstream packet contains a header that signals the type information of the current packet. Therefore, based on header information, the 3D transmitted bitstream

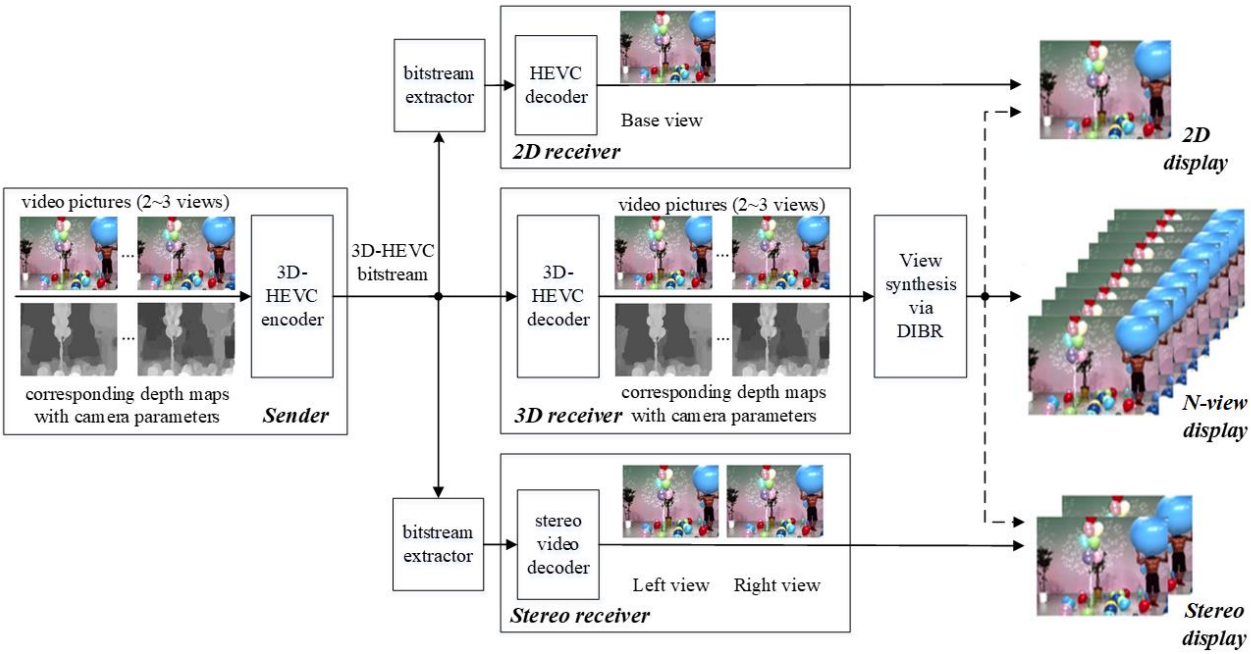


Fig. 2. *Overview of the system structure and the data format for the transmission of 3D video*

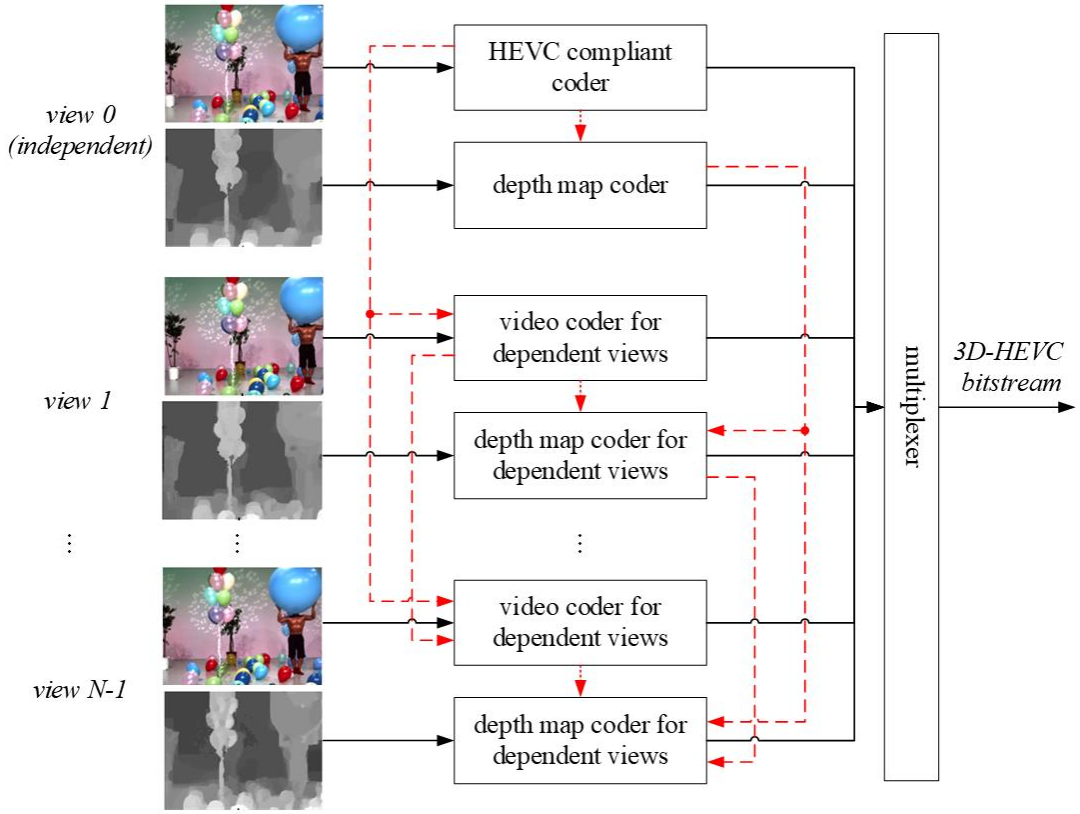


Fig. 3. *Basic codec structure of 3D-HEVC with inter-component prediction (red arrows)*

can be extracted and used for heterogeneous displays such as a 2D display, a stereo display, and an N-view display. This system structure facilitates the format scalability. For instance, if the bitstream is received by an N-view display, the input textures, depth maps, and camera parameters are reconstructed by the 3D-HEVC decoder. The DIBR technology using the reconstructed signals can then synthesize the virtual intermediate views for the N-view display. Here, the N-view display can be considered as an auto-stereoscopic display, where the two views for the stereo video can be selected by an audience. On the other hand, if the 3D receiver is replaced by a stereo or 2D receiver, the required bitstream can be extracted for the corresponding stereo or 2D display.

2.2. 3D-HEVC Codec Structure

The 3D-HEVC encoder in Fig. 2 is designed to encode MVD data. It deals with several views and associated depth maps at the same time instant rather than one single video picture in HEVC. Fig. 3 describes the basic codec structure of 3D-HEVC. In principle, each texture or depth signal is coded by an encoder inherited from the technology of HEVC. The bitstream packets referring to each signal component are then multiplexed together to form the 3D-HEVC bitstream. There is a base/independent view among all views, which is first coded by a fully HEVC compliant codec in order to support format scalability. Other views are dependent views and coded by a modified HEVC codec. As shown in Fig. 3, the independent view can be used as the reference for the dependent views, which is called the inter-view prediction technique. Besides, at each view, the texture picture has an associated depth map capturing the same scene from the same camera position. Thus, 3D-HEVC designs several inter-component techniques for reducing the redundancy between the textures and associated depth maps. Both inter-view techniques and inter-component techniques are marked as the red lines in Fig. 3. Due to the special characteristics of depth maps, several additional coding tools are designed for depth maps in 3D-HEVC. Detailed descriptions of the depth map coding techniques in 3D-HEVC will be discussed in the next section.

3. Depth Map Coding Techniques

Depth map coding is the key factor of 3D-HEVC, and its framework is mostly inherited from the texture video coding in HEVC or 3D-HEVC, such as quadtree coding structure, inter-view coding tools, etc. The inter-view coding tools, including Disparity-Compensated Prediction (DCP), inter-view motion prediction, are designed for reducing the redundancy between different views. These inherited techniques will not be described in this paper and the interested reader is referred to [15].

In the following, we will focus on the additional coding tools for depth maps in 3D-HEVC. Instead of traditional Rate-Distortion Optimization (RDO) used in HEVC, a new optimization method considering the synthesized view quality for depth coding in 3D-HEVC will be described in Section 3.1. The inter-component techniques that reduce the redundancy between depth maps and their associated texture videos will be discussed in Section 3.2. In Section 3.3, we will describe the new depth intra coding tools for better representing the characteristics of depth maps.

3.1. View Synthesis Optimization

Depth maps are transmitted for view synthesis instead of directly viewed by audiences. In this situation, the reconstructed depth map quality cannot guarantee the synthesized view quality from the traditional RDO. Therefore, the View Synthesis Optimization (VSO) scheme [16]-[19] was designed in 3D-HEVC to look for the optimal mode in depth map coding. Different from RDO, the VSO scheme considers both of the synthesized view and depth map quality. The VSO cost $J_{VSO}(m_{DL})$ is computed by the distortion $D_{VSO}(m_{DL})$ plus the Langrangian multiplier λ times the coding rate $R(m_{DL})$ as follows:

$$J_{VSO}(m_{DL}) = D_{VSO}(m_{DL}) + \lambda \times R(m_{DL}) \quad (1)$$

where m_{DL} is one of the possible candidate modes at the depth level of DL. In VSO, $D_{VSO}(m_{DL})$ takes both the distortion of the synthesized view, $D_{syn}(m_{DL})$, and the distortion of the depth map, $D_{dep}(m_{DL})$, into account which is formulated as

$$D_{VSO}(m_{DL}) = \frac{w_{syn} \times D_{syn}(m_{DL}) + w_{dep} \times D_{dep}(m_{DL})}{w_{syn} + w_{dep}} \quad (2)$$

where w_{syn} and w_{dep} are the weighting factors of $D_{syn}(m_{DL})$ and $D_{dep}(m_{DL})$, respectively. $D_{dep}(m_{DL})$ is calculated by the Sum of Squared Error (SSE) or sum of Absolute Hadamard Transform Difference (SATD) between the original and reconstructed depth maps with the chosen mode. The way to compute $D_{syn}(m_{DL})$ can either be computed by the rendering approach or the non-rendering approach.

The Synthesized View Distortion Change (SVDC)-based VSO defined in [20] is considered as the rendering approach, which directly performs view synthesis using the encoded data. As shown in Fig. 4, $D_{syn}(m_{DL})$ is represented by the change of overall distortion of a synthesized view depending on the change of the depth data within a coding block being tested:

$$D_{syn}(m_{DL}) = \sum_{(x,y)\in V_T}\left[\tilde{V}_{T(x,y)}(m_{DL}) - V_{T(x,y)}\right]^2 - \sum_{(x,y)\in V_T}\left[V'_{T(x,y)} - V_{T(x,y)}\right]^2 \quad (3)$$

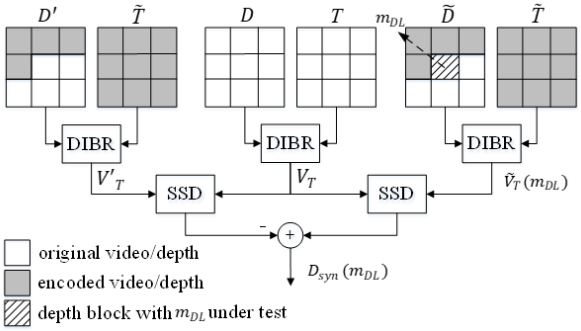


Fig. 4. *Definition of the SVDC related to the distorted depth data of the block depicted by the hatched area which is the candidate mode under test; DIBR is used for view synthesis and SSD stands for sum of squared differences*

where V_T refers to the whole original synthesized view, and (x, y) means the sample position in V_T. V_T is synthesized from the original video, D, and depth data, T. $V'_{T(x,y)}$ is the synthesized view constructed by the reconstructed video and depth values for already encoded block before the current block is determined, $\tilde{T}$ and D', respectively in Fig. 4. $\tilde{V}_{T(x,y)}(m_{DL})$ is similar except that the reconstructed depth values $\tilde{D}$ from the mode, m_{DL}, under test is used for view synthesis. Since the view synthesis is included in SVDC, it induces high computational complexity.

The non-rendering approach, on the other hand, is the model-based view synthesis distortion (VSD) [15], which weights the depth distortion with the sum of absolute horizontal gradients of the co-located texture as

$$D_{syn}(m_{DL}) = \sum_{(i,j)\in PU}\left(\frac{1}{2}\cdot\alpha\cdot\left|PU_{D(i,j)} - \widetilde{PU}_{D(i,j)}(m_{DL})\right|\cdot\left[\left|\widetilde{PU}_{T(i,yj)} - \widetilde{PU}_{T(i-1,j)}\right| + \left|\widetilde{PU}_{T(i,j)} - \widetilde{PU}_{T(i+1,j)}\right|\right]^2\right) \quad (4)$$

where (i, j) means the sample position in a PU. $PU_{D(i,j)}$, and $\widetilde{PU}_{D(i,j)}(m_{DL})$ indicate the original and reconstructed depth data in a PU, respectively. $\widetilde{PU}_{T(i,j)}$ indicates the reconstructed texture data. And α is the proportional coefficient determined by the depth distance from the camera. In comparison with the SVDC-based VSO, the VSD-based VSO requires less computational complexity, but only gives lower accuracy because the calculation of $D_{syn}(m_{DL})$ does not actually carry out view synthesis.

3.2. Inter-Component Techniques

In 3D-HEVC, the inter-component techniques allow the depth coding to use the already coded data from the associated texture video from the same view. The Motion Parameter Inheritance (MPI) [21]-[22], and the Depth Quadtree Prediction (DQP) [23] can be classified in this category.

3.2.1 MPI: The motion characteristics of the depth map and their associated texture video are very similar because they project the same scene from the same viewpoint at the

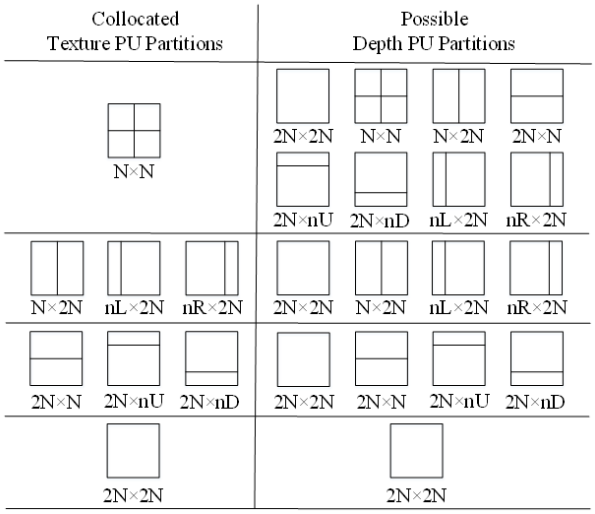


Fig. 5. *Texture partitions and corresponding possible candidates of depth partitions*

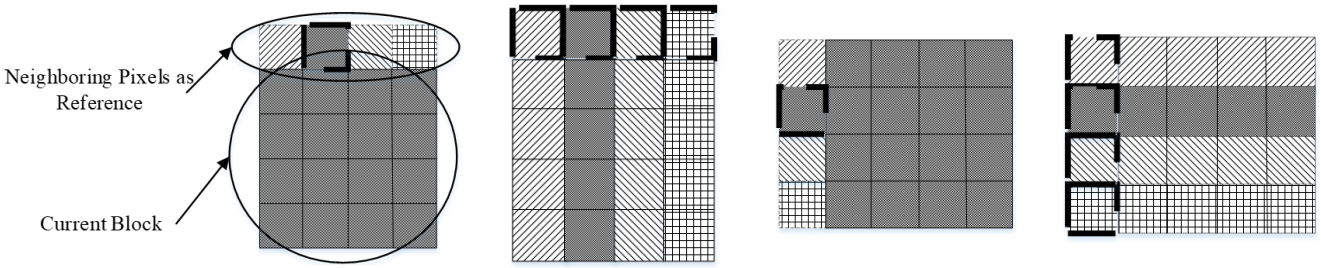


Fig. 6. *Examples of CUs predicted by DIS* ***(a)*** *Vertical mode (Type 0),* ***(b)*** *Vertical mode (Type 1),* ***(c)*** *Horizontal mode (Type 2),* ***(d)*** *Horizontal mode (Type 3)*

same time. MPI is therefore introduced in 3D-HEVC, where the motion parameters from the texture video are allowed to be inherited as one candidate of merge mode in the corresponding depth block. For each depth block, it is adaptively determined whether the co-located block of the associated video is added into the candidate list of Merge mode or not. In other words, MPI mode is used to extend the candidate list of merge mode in which the first candidate refers to merging the co-located block from the associated video. The advantage of using the merge mode syntax is to allow efficient signaling in the case where MPI is adopted without coding a residual signal. It is noted that quarter-pixel accuracy is used in the motion vectors of texture videos, while depth map only utilizes full-pixel accuracy. Therefore, the inherited MPI candidates are quantized to their nearest full-sample position in the depth map coding.

3.2.2 DQP: In spite of representing the same scene from the same camera, the texture video always contains more details than its associated depth map. Consequently, the corresponding quadtree structure of texture video tends to be more complicated than that of the depth map. DQP performs a prediction of the depth quadtree from the corresponding texture quadtree. In general, the partition level of each CU in a depth map cannot be larger than that of the collocated texture blocks. Due to this observation, the possible depth PU partitions are limited to a small range compared with texture PU partitions. Without DQP, there are totally eight possible partitions for each depth PU: 2N×2N, N×N, N×2N, 2N×N, 2N×nU, 2N×nD, nL×2N, nR×2N. As shown in Fig. 5, the limited possible depth PU partitions are listed in the right according to the collocated texture PU partitions that have been determined in the left. For example, if the collocated texture PU is selected as 2N×2N, the depth PU is decided as 2N×2N without checking all other PU partitions. With DQP, the coding complexity from PU or CU partition can be remarkably reduced. At the same time, the optimal partition may be skipped in some cases resulting in an inaccurate prediction. In order to avoid quality degradation in key reference frames, DQP is only applied in inter slices that do not belong to random access pictures.

3.3. Depth Intra Coding Tools

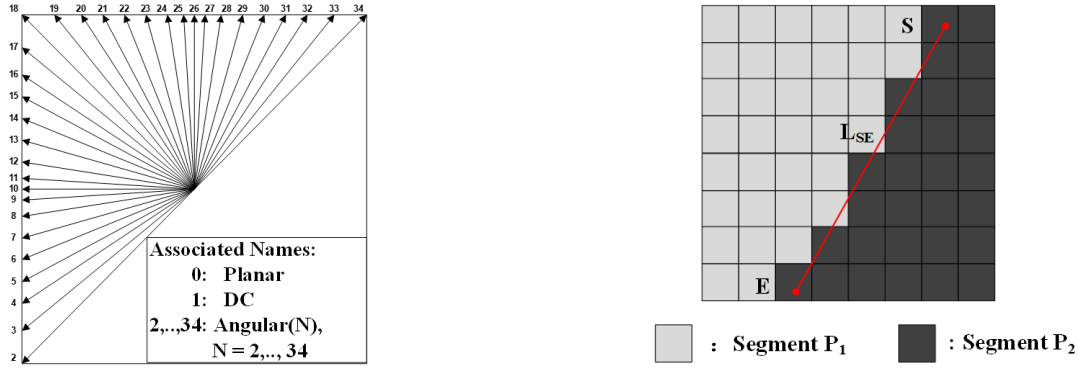
As mentioned in Section 1, depth maps have large flat areas delimited by sharp edges. Preserving sharp edges in depth maps is the most critical task. Consequently, the investigations into alternative depth intra coding tools were carried out. The new tools include Depth Intra Skip (DIS) mode [24], Depth Modelling Mode (DMM) [17], and Segment-wise Depth Coding (SDC) [25]. DIS is designed for flat areas, while DMM and SDC help to preserve the sharp edges in depth maps.

3.3.1 DIS: DIS mode is a new intra mode especially for depth map coding, which is quite useful in coding flat regions. As shown in Fig. 6, DIS directly uses the reconstructed value(s) of some designated spatial neighboring pixel(s) to represent the current CU. In DIS, no prediction residual is encoded, and this is the major difference between DIS and other intra modes.

There are two vertical modes and two horizontal modes of DIS, which is signaled by an index (DIS type 0-3). The vertical DIS mode indexed by type 0 and type 1 is shown in Fig. 6(a) and (b), where the neighboring reference pixels are above the current block. Type 0 in Fig. 6(a) constructs a predicted CU with one single depth value from the mid-above pixel, while Type 1 in Fig. 6(b) is the same as the vertical angular mode in terms of block prediction. Besides, the two horizontal DIS types with left-neighboring reference pixels are also described in Fig. 6(c) and (d). Type 2 in Fig. 6(c) constructs a predicted CU with one single depth value from the mid-left pixel, while Type 3 in Fig. 6(d) is the same as the horizontal angular mode in terms of block prediction. For each DIS mode type, the VSO cost is separately calculated without residual coding. Finally, the DIS type with the minimum VSO cost using (3) is determined as the best DIS type for the current CU.

3.3.2 DMM: Different from the Conventional HEVC Intra Modes (CHIMs) consisting of planar, DC and 33 angular modes as illustrated in Fig. 7(a), DMM does not use the reconstructed neighboring pixels. DMM in Fig.7(b) is a new intra mode to better describe the sharp edges in a depth map. Two regions (P_1 and P_2) are generated from flexible non-rectangle partition, by varying the wedgelet separation line L_{SE}. Each segment is predicted by a constant pixel value (CPV). As shown in Fig.7(b), for each partition of DMM, S and E denote the start and end points of L_{SE}, respectively. As S and E move around the block boundary, the total number of possible divisions increases significantly, which results in a huge number of wedgelet patterns in DMM. Since it is necessary to find the best wedgelet pattern as a prediction mode, the intra mode decision obviously becomes very time-consuming.

3.2.3 SDC: After prediction, residuals are encoded by subtracting the prediction from the original pixel values. In



(a) (b)

Fig. 7. *Depth intra modes in 3D-HEVC,* ***(a)*** *35 HEVC traditional intra modes,* ***(b)*** *DMM*

residual coding, SDC is raised as an alternative coding method in 3D-HEVC. For natural videos, discrete cosine transformation and quantization (QDCT) are applied to residual blocks after inter or intra prediction. In depth map coding, some non-zero residuals always distribute along the sharp edge, and the traditional QDCT may degrade the sharp edge due to the loss of high-frequency QDCT coefficients. SDC allows using only a CPV to represent each segment instead of QDCT. It is performed in the pixel-domain and can be applied to CHIMs and DMM. The residual block is regarded as one segment for CHIMs, while it is composed of two segments for DMM. In SDC, the residual signals of each segment are calculated by the delta CPV, CPV_{delta}, which is defined as

$$CPV_{delta} = CPV_{ori} - CPV_{pre} \tag{5}$$

where CPV_{ori} denotes the original CPV (i.e., DC value of original block) and CPV_{pre} is the predicted CPV. For CHIMs, CPV_{pre} is the mean value of four corner pixels of the predicted block, while it is predicted from the neighboring pixels for DMM. An offset around CPV_{ori} (five candidates with offsets of 0, -1, 1, 2, -2) is searched based on rendering-based VSO to provide the accurate description. After that, the rectified delta CPVs are signaled in the bitstream. Moreover, SDC uses the depth lookup table (DLT) to encode each CPV and its predicted CPV for reducing bitrate further [25]. Due to the characteristics of depth maps, only a small amount of depth values is utilized from the full depth range of 0~255. The DLT only uses a restricted set of the valid depth values, and they are mapped into an index table.

4. Fast Algorithms for Depth Maps

With the additional coding tools in 3D-HEVC, depth map coding induces huge computational complexity. Therefore, the development of fast algorithms become crucial in 3D-HEVC. In this section, two main categories of the current fast algorithms are introduced. They are fast PU mode decision and fast CU size decision.

4.1. Fast Algorithms for PU Mode Decision

For PU inter mode decision, the coding techniques for depth maps do not have much difference from those for texture videos in 3D-HEVC. We thus do not introduce the techniques related to PU inter mode decision in this paper. Interested readers may refer to [15]. In contrast, a lot of the current research works pay more attention to the PU intra mode decision for depth maps, where some additional coding tools are added. In Section 4.1.1, the PU intra mode decision process for depth maps in 3D-HEVC, as well as a rough complexity analysis based on the reference software HTM-16.0 [26], are described. Then, the recent fast algorithms designed for depth intra mode decision are introduced in Section 4.1.2.

4.1.1 PU Intra mode decision: The implementation of PU intra mode decision includes three major parts: DIS, Intra 2N×2N and Intra N×N, as shown in Fig. 8. In DIS, the DIS type with the minimum $J_{VSO}(m_{DL})$ using SVDC-based VSO using (3) is determined as the best DIS type for the current 2N×2N PU. In Intra 2N×2N or Intra N×N, the best mode from CHIMs and DMM mentioned above are selected, in which the detailed process can be summarized in the following five steps:

Step 1. Rough mode decision (RMD) [15] is used to choose a number of effective candidates from CHIMs by minimizing $J_{VSO}(m_{DL})$ in (1), where $D_{syn}(m_{DL})$ is calculated by the model-based VSD in (4) and $R(m_{DL})$ includes only the bits for each prediction mode. These effective candidates (3 modes for 64×64, 32×32 and 16×16 PUs, and 8 modes for 8×8 and 4×4 PUs) are then put into the candidate pool for further evaluation. In addition, the three most probable modes (MPMs) from the left and up neighboring PUs are appended as candidates in the pool.

Step 2. The optimal wedgelet partition among all the DMM candidates is sought by minimizing $J_{VSO}(m_{DL})$ using model-based VSO in (3). Then the optimal DMM wedgelet partition is added into the candidate pool.

Step 3. $J_{VSO}(m_{DL})$ using SVDC-based VSO in (3) is computed for each candidate in the candidate pool from Step 1-2. At this stage, $R(m_{DL})$ includes the bits for both the prediction mode and residuals. It is noted that the residuals are coded with the traditional QDCT with limited Residual Quad-Tree (RQT) [15].

Step 4. $J_{VSO}(m_{DL})$ using SVDC-based VSO in (3) is calculated again for each candidate in the pool with SDC as the residual coding approach.

Step 5. $J_{VSO}(m_{DL})$ from Steps 3-4 are compared to select the final optimal intra mode. With the optimal intra mode, the full RQT is searched to determine an optimal transform kernel size.

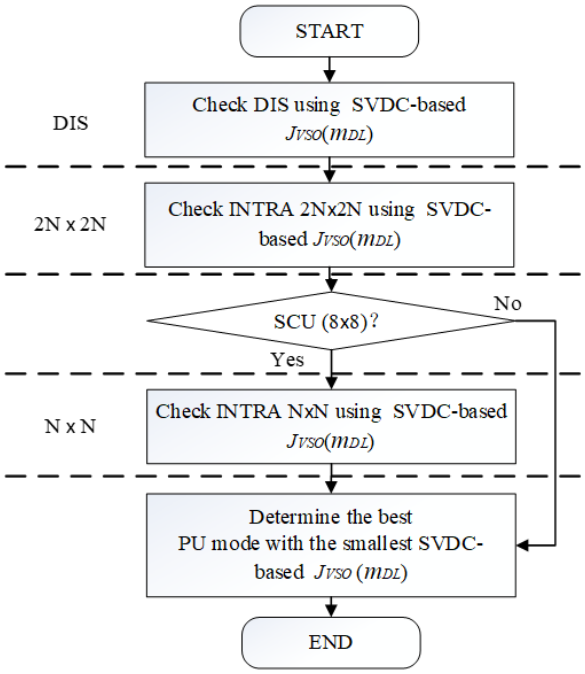


Fig. 8. *PU intra mode decision for depth map in 3D-HEVC*

As shown in Fig. 8, the corresponding optimal cost in Step 5 is further compared with the VSO cost of DIS to obtain the final optimal intra mode.

From Step 1 to Step 5, the main idea is to construct a candidate pool and then select the optimal intra mode from the pool that is inherited from the texture intra coding. However, due to the VSO calculation and DMM or SDC tools designed for depth maps, depth intra coding is far more complicated than texture intra coding.

To study the coding complexity of depth intra coding in detail, we did some statistical analysis of the encoding time occupation for each step of depth intra decision in Fig. 9. It is noted that all the experiments were conducted with HTM-16.0 under All-Intra configuration. The test sequences include all 8 sequences recommended in the Common Test Condition (CTC) [27]. As shown in Fig. 9, DIS calculation only occupies about 4.48% of the total depth intra coding time, while the RMD and MPM search also takes up a small proportion of 6.02%. DIS only has four types and does not need residual coding. RMD has 35 intra modes to consider but employs the non-rendering method for calculating $J_{VSO}(m_{DL})$. And the MPM search does not include the calculation for $J_{VSO}(m_{DL})$. These coding steps do not need very complex calculation and are usually not the major issue for depth intra coding.

From Fig. 9, it can be observed that the three most time-consuming processes are the optimal DMM wedgelet pattern decision in Step 2 (20.25%), the residual coding with limited RQT in Step 3 (29.04%), and the residual coding with SDC in Step 4 (24.74%). Although the calculation of $J_{VSO}(m_{DL})$ using the non-rendering method is adopted in Step 2, the enormous number of DMM wedgelet pattern candidates involves huge computational complexity. Step 3 and Step 4 employ the time-consuming $J_{VSO}(m_{DL})$ calculation using SVDC-based $D_{syn}(m_{DL})$ for 5 to 13 candidates, resulting in intolerable time costs. Besides, Step 4 checks all the combinations of offsets for CPVs, which again causes significantly high complexity.

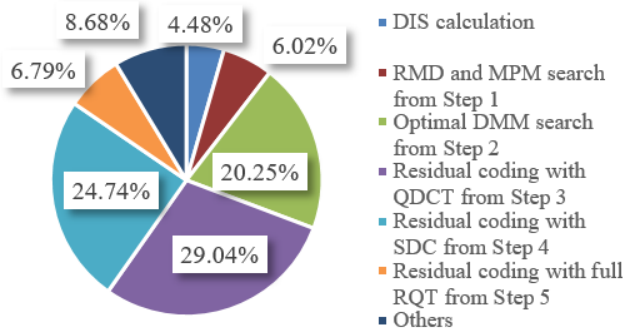


Fig. 9. *Encoding time distribution of different intra mode decision steps*

4.1.2 Fast PU Mode Decision Algorithms: Since the inter coding techniques for depth maps mainly inherit from that of texture videos, most of the fast algorithms for PU inter mode decision in texture videos can also be used in depth maps. One of our previous works has tried to apply the algorithm for texture videos into inter mode decision for depth maps [28], and achieve a similar performance in both texture videos and depth maps. In the following, we only focus on the intra mode decision rather than the inter mode decision for depth maps.

The fast PU mode decision algorithms for depth intra coding can be divided into four categories: skipping the whole DMM process in Step 2 for unnecessary PUs [29]-[34]; limiting the number of wedgelet patterns for DMM in Step 2 [35]-[40]; simplifying $J_{VSO}(m_{DL})$ in Step 3 to Step 5 [41]-[43]; and reducing the number of candidates and limiting the search range for SDC in Step 4 [44]-[45].

The DMM skipping algorithms in [29]-[32] find out smooth regions to skip the whole DMM decision where the PU has already predicted well by the best candidate in CHIMs such as the Planar or DC mode. In this case, DMM becomes unnecessary and can be skipped. In [29], PU is identified as a smooth region when the Planar mode is selected in RMD. As the extension of [29], Silva *et al.* [30] suggested a threshold-based DMM skipping method based on the rough mode cost from RMD. Similarly, the work in [31] excludes the DMM mode decision process in a smooth region when the Planar mode is one of the most probable mode. The algorithm in [32] utilizes the coding information from the spatial neighboring block and its co-located texture to skip the whole DMM calculation. To further extend this concept, the investigations in [33]-[34] were studied to early terminate unnecessary DMM based on not only smooth regions but also simple edge regions in which they have already well predicted by one of the angular modes in CHIMs. A Hadamard transform domain edge classifier [33] and a simple edge classifier by comparing the difference of four corner pixels with a pre-defined threshold [34] were proposed to detect the PUs with a horizontal or vertical edge such that more redundant PUs can be skipped in DMM. It is noted that all the wedgelet patterns are evaluated in [29]-[34] if the current PU is not identified as the PU that can be well predicted by CHIMs.

In contrast, the algorithms in [35]-[40] do not skip the whole DMM decision, but they only reduce the number of wedgelet patterns to be searched. Our prior works divide all DMM candidates into six subsets. The variance is then considered as the feature to select which subset should be

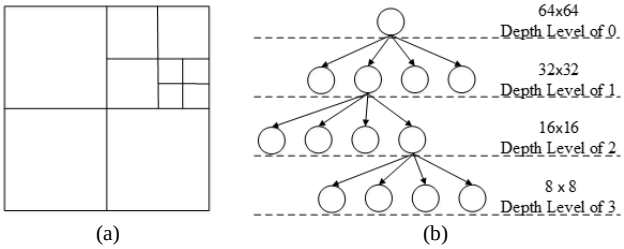


Fig. 10. *An example of coding structure in 3D-HEVC* ***(a)*** *flexible CU block sizes,* ***(b)*** *quadtree structure*

checked [35]-[36]. The work in [37] developed a two-step wedgelet pattern decision algorithm, one for coarse wedgelet search in the double pixel-domain and the other for refinement. In addition, Xu *et al*. [38] utilizes the directional mode with the minimal cost in RMD and searches in its corresponding wedgelet pattern subset. The algorithm in [39] looks for the wedgelet patterns around the border samples with the largest gradient. A flexible block partitioning scheme in [40] was proposed to work with a constrained wedgelet pattern subset based on the partition blocks.

There are also a number of algorithms designed for simplifying the calculation of the VSO cost [41]-[43]. An adaptive distortion model based on different pixel intervals was suggested in [41] to estimate the VSO. The VSO cost in [42] proposed an area-based scheme by using co-located texture information for VSO calculation. In [43], we designed a very simple but efficient VSO metric by using only the variance of the two wedgelet regions to entirely replace the computational demanding SVDC-based VSO in (3) for DMM decision.

The work in [44] expedites the process of SDC by comparing the VSO cost of prior checked modes such as the mode information of RQT. Lee *et al*. [45] limited the search range of an offset for CPVs in the SDC decision process in which the offsets of 2 and -2 can be skipped if the offset of 0 has the lowest VSO among the offsets of -1, 0, and 1.

In addition to these algorithms, some of our previous works [46]-[48] jointly reduce the number of CHIMs and DMM for residual coding with traditional RQT or SDC in Step 3 and Step 4. The candidate pool is pruned according to the costs from RMD [43], [46], the reference pixels classification [47], or the hierarchical larger CUs [48].

4.2. Fast Algorithms for CU size Decision

The CU size decision for depth maps in 3D-HEVC is quite different from that of texture videos in HEVC. Depth maps have many large flat areas, which are usually coded as large CU size and can be accelerated significantly. However, the sharp edges in depth maps need more careful protection due to their importance in the synthesis process. These sharp edges tend to be destroyed when they are coded by large CU size. Thus, some works have been proposed to design an early CU size decision in flat areas without blurring sharp edges. In Section 4.2.1, CU size decision in 3D-HEVC is reviewed, and the CU size distribution in the final depth map bitstream is then studied. Besides, the fast algorithms designed for CU size decision are introduced in Section 4.2.2.

4.2.1 CU size decision: In 3D-HEVC, a depth map is divided into non-overlapping CTUs. Each CTU can be a maximum of 64×64 pixels. Starting from CTU, the CU partitioning allows the CTU flexibly splitting into four equally sized sub-CUs, as shown in Fig. 10(a). The corresponding CU partitioning structure in Fig. 10(b) is also called a quadtree structure. For each partitioning, the Depth Level (DL) is increased by 1. Obviously, there are numerous possible quadtree structures for the codec to select the optimal one.

In order to obtain the optimal quadtree structure with the best coding efficiency, all possible CU sizes are tested during the CU partitioning process as the order shown in Fig. 11. For each CU, the PU mode decision will be conducted to obtain the best $J_{VSO}(m_{DL})$. The splitting of CU in the final optimal quadtree is determined by comparing the VSO cost of the current CU at the depth level of DL, $J_{VSO}(m_{DL})$, and the sum of VSO costs of its four smaller sub-CUs at the higher depth level of DL+1, $\sum_{i=0}^{3} J_{VSO}(m_{DL+1}^{i})$, according to

$$Flag_{Split} = \begin{cases} 0, & if\ J_{VSO}(m_{DL}) \le \sum_{i=0}^{3} J_{VSO}(m_{DL+1}^{i}) \\ 1, & if\ J_{VSO}(m_{DL}) > \sum_{i=0}^{3} J_{VSO}(m_{DL+1}^{i}) \end{cases} \quad (6)$$

where $Flag_{Split}$ of 0 or 1 is the splitting flag to indicate Non-Split and Split, respectively, for the current CU. m_{DL+1}^{i} is the optimal mode of the i^{th} CU at depth level of $DL+1$, determined by the PU mode decision mentioned in Section 4.1.

Taking CU 3 in Fig. 11 as an example, if the total VSO cost of its sub-CUs (CU 4-7) is larger than the VSO cost of CU 3, CU 3 will not be split into sub-CUs in the final structure as the decision in (6). Otherwise, CU 4-7 will replace the CU 3 as the optimal partition modes of the current CU. The same decision is employed for all CUs recursively from high to low depth levels. Finally, the quadtree with the minimum VSO cost is selected as the optimal structure and coded in the bitstream.

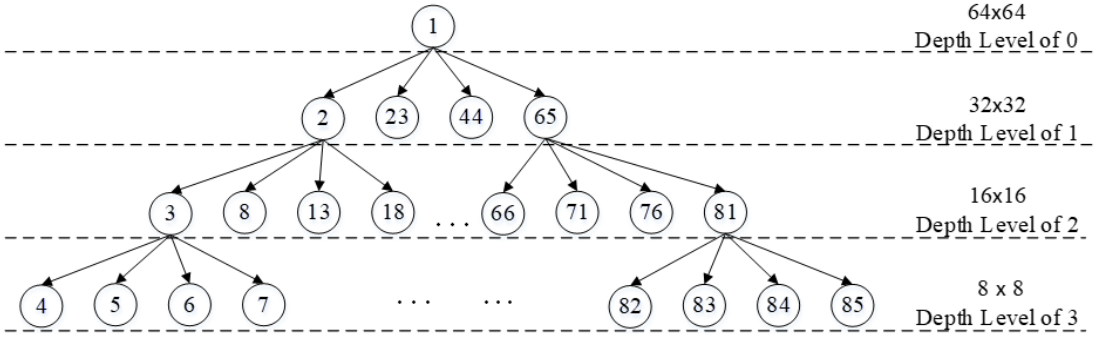


Fig. 11. *CU partitioning among different depth levels (the number in the circle is the coding order of the CUs)*

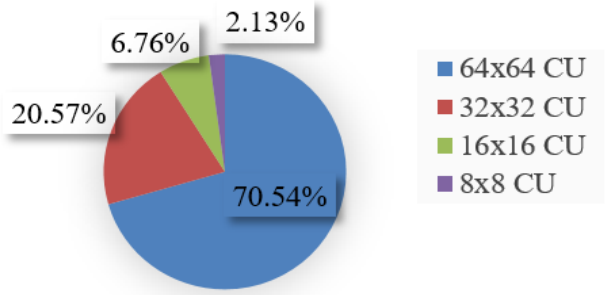


Fig. 12. *CU size distribution for depth maps*

It is noted that no matter what the final quadtree is, there must be 85 times recursive computation for all possible CUs within one CTU, as shown in Fig. 11. However, not all CUs are necessary to be checked, especially when depth maps have extremely biased CU size distribution.

Using HTM-16.0 under All-Intra configuration, Fig. 12 shows the CU size distribution for depth maps. It is noted that the test sequences include all 8 sequences recommended in CTC [27]. As we can see, the most blocks are coded with large CU size (70.54% for 64×64 CU size, and 20.57% for 32×32 CU size), while few blocks select 16×16 CU size or 8×8 CU size. In other words, many blocks can be early decided as large CU sizes and do not employ further quadtree partitioning.

4.2.2 Fast CU Size Decision Algorithms: The CU quadtree structure for depth maps in 3D-HEVC is inherited form HEVC. Theoretically, the fast algorithms [49]-[53] for texture videos in HEVC can also be used for depth maps directly. In [49], the quadtree depth range is determined by the coded quadtree information from previous frames and neighboring CUs. The quadtree information of neighboring CUs was also exploited in [50]-[51] to make an early CU split decision or CU pruning decision. Using the variance value of the input image, the method in [52] early determines the CU sizes for texture intra coding. Min *et al.* [53] proposed a fast CU size decision method for intra coding in HEVC, which is based on the global and local edge complexity of the current CU and its four sub-CUs.

However, the characteristics of depth maps are quite different from that of texture videos as mentioned in Section 3, i.e. large flat areas with sharp edges. It contains much more large CUs than texture video in the optimal quadtree. Besides, the depth map coding in 3D-HEVC contains additional tools compared with texture coding. To consider these new characteristics of depth maps, a number of CU size decision algorithms [54]-[57] were designed specifically for depth maps.

Mora *et al.* [54] exploited texture-depth redundancies and developed an inter-component coding tool, in which the depth quadtree is restricted by the coded texture quadtree. It is noted that the quadtree limitation of [54] is switched off for intra slices since it damages the synthesized view quality significantly and causes decoder decoupling. Our previous work in [55] discovered that the multi-directional edges or corner points are highly related to the optimal CU sizes within block areas, and then limits the range of depth level based on the existence of corner points. In addition to the information from pixel-domain or coded texture blocks, the intermediate value from the depth coding process for the current CU is also exploited in [56]. An CU quadtree pruning algorithm was then designed by considering the variance of block areas and the distortion of DIS. In our previous work [57], we found that the VSO cost of DIS, as the optimal PU mode, is always much smaller than that of other intra modes. We then proposed an early partition termination decision for intra depth blocks where their first sub-CUs are coded with DIS.

5. Fast Algorithm using Machine Learning Tools

In recent years, machine learning tools have achieved great success in many computer vision tasks. At the same time, learning-based approaches are considered more and more in the aspect of video coding. Some learning-based methods [58]-[61] have been designed for texture videos in HEVC. In [58], the determination of CU splitting is inferred by an online classification model using the Pegasos algorithm. Based on off-line training, Du et al. [59] proposed a random forest classifier to skip or terminate the current CU depth level in HEVC. An adaptive fast CU size decision was proposed in [60], where the Support Vector Machine (SVM) is employed to analyse and construct the classification model according to the CU complexity. Zhu et al. [61] developed a binary and multi-class SVM based fast HEVC encoding algorithm, in which a learning-based method is used to take the advantages of both off-line and on-line learning modes for classifiers.

These learning-based methods were designed for HEVC instead of depth maps in 3D-HEVC, where the additional coding tools for depth coding, as well as the special features of depth maps, have not been specially exploited. In this section, by taking the characteristics of depth maps and the new coding tools into consideration, we introduce an efficient fast algorithm for depth intra coding using a classical

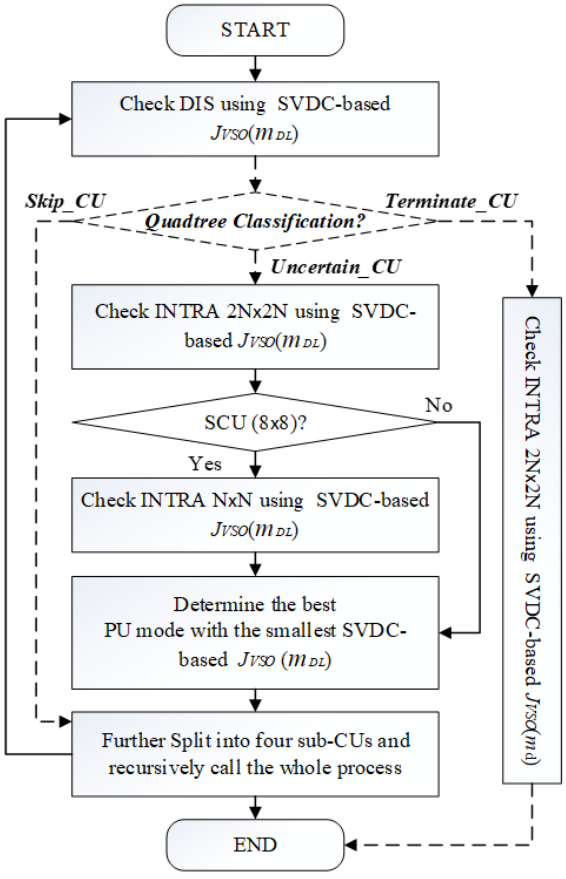


Fig. 13. *Proposed early CU size decision*

Table 1 The description of features and samples

		Descriptions
Selected Features		DIS_cost after DIS calculation DL, ranging from {0,1,2,3} QP, ranging from {35,39,42,45}
Collected Samples	$Terminate\_CU$	CU samples: terminated at that current DL in the final quadtree
	$Skip\_CU$	CU samples: skipped and split into smaller sub-CUs in the final quadtree

classification method: decision tree. Section 5.1 proposes an early decision model for depth intra coding. In Section 5.2, the features and samples selected for model training will be discussed. The way of using a decision tree as the required classification model and its related learning process are described in Section 5.3. Finally, simulation results for this fast algorithm are shown in Section 5.4.

5.1. Proposed DIS based Quadtree Classification

In our previous work [57], it is found that DIS mode is dominant compared to other intra modes in depth map coding. The evaluation of the DIS mode is the first step of PU intra mode decision as shown in Fig.8. Moreover, the computational time of DIS itself is insignificant as shown in Fig.9. These observations make it possible to perform the DIS mode first and then exploit the VSO cost of DIS as one of the features to train a classifier for the acceleration of the following coding process.

Based on the VSO cost of DIS, an early CU size decision can be conducted not only for early terminating some branches of quadtree partitioning but also for skipping some quadtree node that directly splits the CU being considered into four smaller sub-CUs. The flowchart of the proposed algorithm is shown in Fig.13. A quadtree classification is set after the DIS evaluation, where the temporary coding information of DIS can be used as the classifier input. The quadtree classification has three outputs, and they are labeled as $Terminate\_CU$, $Skip\_CU$, and $Uncertain\_CU$. The CUs labeled as $Terminate\_CU$ do not carry out further CU split, while the CUs with the label of $Skip\_CU$ are decided to skip the evaluation of the current quadtree node and split it into smaller CU size directly. In addition, the CUs with the label of $Uncertain\_CU$ refers to the CUs which cannot be easily put into the two labels of $Terminate\_CU$ and $Skip\_CU$, where the original depth intra coding flow is performed.

5.2. Feature Selection and Sample Collection

Table 1 describes the selected features. Since the quadtree classifier in Fig. 13 is after the DIS evaluation, the VSO cost of DIS, DIS_cost, can be used as a feature in the proposed learning-based algorithm. Since the VSO cost varies with different block sizes, the quadtree Depth Level (DL) of each CU is added as another feature. Besides, the Quantization Parameter (QP) is used as the third feature, with which the codec tolerates different degrees of distortion.

As shown in Fig.12, the distribution of the final selected CU size is biased to a large size such as 64×64. In the collection of samples, instead of using all the nodes in Fig. 11, only the nodes remained in the final optimal quadtree, such as Fig.10(b), are labeled as samples. The descriptions of the collected samples are also listed in Table 1. The terminal nodes in the final quadtree are labeled as $Terminate\_CU$, while the nodes which are split into smaller sub-CUs are labeled as $Skip\_CU$. Taking the final quadtree in Fig. 10(b) as an example, we show the labels of the CU samples in Fig. 14. The dark CUs are labeled as $Skip\_CU$ since they are further split in the quadtree. The white CUs with the label of $Terminate\_CU$ are those terminal nodes in the final quadtree, which are terminated at that DL. It is noted that although three outputs are described in the flow chart of Fig.13, a binary classifier is considered in the collection of samples and the training process. Later when the classifier is used for acceleration, one more parameter, Gini value, is used to evaluate the accuracy of classification. The CU node with a high Gini value is believed to have low accuracy in classification. Those nodes are then classified as the third label of $Uncertain\_CU$.

Table 2 Test Sequences and Properties

Resolution	Test Sequence	Frame Rate	All Frames
1024x768	Balloons[E]	30	300
	Kendo[E]	30	300
	Newspaper[T, E]	30	300
1920x1088	GT_Fly[E]	25	250
	Poznan_Hall[E]	25	200
	Poznan_Street[E]	25	250
	Shark[E]	30	300
	Undo_Dancer[E]	25	250

T: Sequence for training;
E: Sequence for evaluation;

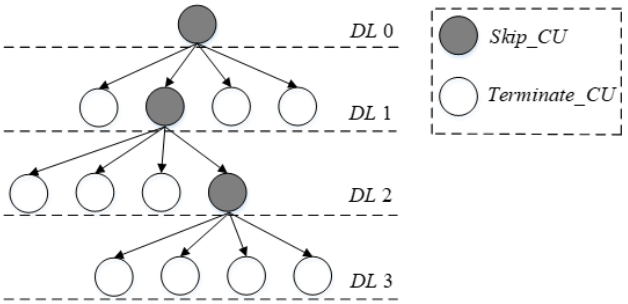


Fig. 14. *An example of Sample Collection*

5.3. Training Strategy for Decision Tree Model

According to the CTC specified in [27], there are 8 sequences, as shown in Table 2, for the evaluation of various depth map coding algorithms. The sequence *Newspaper* is selected for sample collection during the model training. To avoid redundant samples, samples were collected by extracting the frames of Newspaper every 30 frames, i.e., 1st, 31th, 61th, …, 271th frames. The coding model HTM-16.0 [26] with All-Intra configuration is used to conduct the original depth intra coding in 3D-HEVC for the ground truth labels.

The Scikit-learn [62], a popular machine learning package, was adopted in this paper for offline training.

Table 3 Performances of the proposed algorithm, the algorithms in [43], [54] compared with HTM-16.0

Test Sequence	Zhang's [43]		Mora's [54]		Proposed	
	ΔBDBR (%)	ΔT (%)	ΔBDBR (%)	ΔT (%)	ΔBDBR (%)	ΔT (%)
Balloons	+0.40	-38.53	+5.47	-45.13	+0.04	-52.14
Kendo	+0.46	-40.24	+4.19	-54.73	+0.17	-55.56
Newspaper	+0.87	-35.51	+4.40	-45.64	+0.13	-47.40
GT_Fly	+3.01	-41.99	+6.06	-49.08	+0.26	-64.34
Poznan_Hall2	+0.89	-49.45	+2.77	-63.59	+0.12	-72.22
Poznan_Street	+0.27	-43.83	+1.52	-50.44	+0.06	-60.95
Shark	+0.91	-41.94	+3.87	-52.21	+0.01	-62.11
Undo_Dancer	+0.31	-43.40	+0.79	-46.59	+0.15	-63.88
Average	**+0.89**	**-41.86**	**+3.63**	**-50.93**	**+0.12**	**-59.83**

Written as a python package in Scikit-learn, the well-known classification and regression trees algorithm [63] was used to construct the decision tree in the proposed algorithm. It is noted that each leaf node of the final tree has a Gini value representing for the impurity of the final classification decision. The lower the Gini value is, the more accurate the leaf decision is. The Gini value for each node is calculated as follows:

$$Gini = 1 - \sum_{k=1}^{2} p_k^2 = 1 - \sum_{k=1}^{2} \left(\frac{N_k}{N}\right)^2 \quad (7)$$

where k of 0 or 1 refers to the label of $Terminate\_CU$ or $Skip\_CU$ in sample collection, p_k is the proportion of the samples with k label at the current leaf node, which is calculated by the number of samples with k labels, N_k, and the total number of samples, N, at the current leaf node.

Fig. 15 shows an example of the final decision tree for the proposed early quadtree decision. As we can see, each leaf node with the label of $Terminate\_CU$ or $Skip\_CU$ has a Gini value. As mentioned before, the Gini value is used as the criteria to assess the accuracy of classification. In this paper, we set a threshold $TH = 0.1$ for the Gini value of the leaf node. If the Gini of the leaf node is less than TH, we then adopt the original leaf node label, $Terminate\_CU$ or $Skip\_CU$, into the proposed flowchart of Fig. 13. Otherwise, the CUs are classified as $Uncertain\_CU$ and the original depth intra coding process is used, e.g. the leaf nodes bounded with the dash lines in Fig.15. The overall proposed early CU size decision is also shown as follows:

$$Decision = \begin{cases} Terminate\_CU, & T\_leaf\ with\ Gini \le TH \\ Skip\_CU, & S\_leaf\ with\ Gini \le TH \\ Uncertain\_CU, & Otherwise \end{cases} \quad (8)$$

where T_leaf and S_leaf refers to the leaf nodes which are labeled as $Terminate\_CU$ and $Skip\_CU$, respectively, from the trained decision tree model.

5.4. Simulation Results

The proposed early CU size decision algorithm of depth intra coding was implemented in HTM-16.0 [26]. The original depth intra mode decision in HTM-16.0 was an anchor for comparison with the algorithms in [43], [54] and the proposed algorithm. The quantization parameters (QP) were set as 25, 30, 35, 40 for texture views and 34, 39, 42, 45 for the corresponding depth views. Sequences for evaluation are all eight sequences listed in Table 2, which were tested with All-Intra configuration under CTC [27]. The experimental work was implemented on the platform with a 64-bit Microsoft Windows 10 OS running on an Intel Core i7-4790 CPU of 3.60 GHz and 16.0GB RAM.

To study the performance of the proposed algorithm compared with the state-of-the-art algorithms, coding results including complexity reduction and coding efficiency are taken into account. The average of encoding time saving ΔT for four different QPs is used to evaluate the complexity reduction. And the coding efficiency is evaluated by the Bjøntegaard delta bitrate (BDBR) [64], which is calculated by the PSNR of synthesized views and the total bitrate of depth and texture videos.

Table 3 shows the results of the proposed algorithm with the Gini threshold (*TH*) of 0.1. It can be seen that the proposed algorithm has a remarkable time reduction of 59.83% with negligible BDBR increase. Besides, the result of the algorithm in [43] only obtains the time reduction of 41.86% with 0.89% BDBR increase, while the algorithm in [54] saves 50.93% of the depth intra coding time with 3.63%

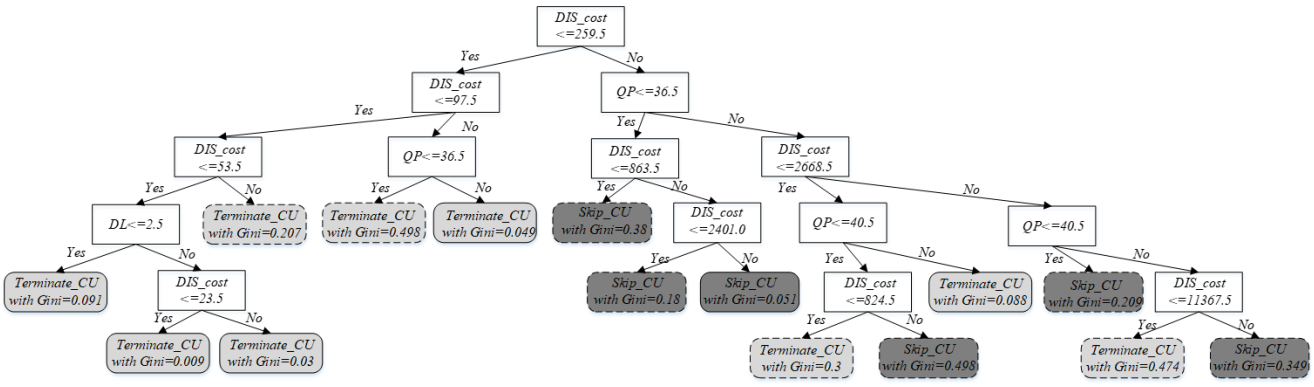


Fig. 15. *Final decision tree for the proposed early CU size decision algorithm*

BDBR increase. These two algorithms are uncompetitive to our proposed algorithm, especially in BDBR performance.

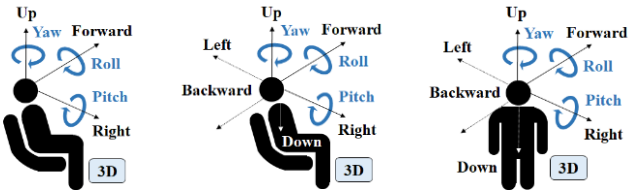
6. Role of Depth Coding in Future Immersive Media

The advancements of sensors, Head-Mounted Displays (HMDs) and 5G networks for Virtual Reality (VR), Augmented Reality (AR), Mixed Reality (MR), and 360-degree videos create new prospective applications and services in markets of education, entertainment, professional training, etc. To ensure interoperability, ISO/IEC MPEG drafted a technical report for the digital representations of immersive media applications, that includes various 3D representation and coding formats [65], and conducted a survey to seek industry opinions on VR [66] in early 2016. Later on, MPEG planned a 5-year standardization roadmap for addressing the needs of standardization from the industry to support the future VR applications and services [67]-[68], and launched MPEG-I project for the coding representation of immersive media [69]. Two mainstreams for the standardization activities of MPEG-I associated with VR - Point Cloud Compression (PCC) and image-based coding are concurrently under development. PCC mainly adopts non-image-based coding approaches such as octree coding, or voxelization with the coding of blocks and vertices [70]. It is noted that PCC is beyond the scope of this paper, and the interested reader is referred to [70]-[71]. For image-based coding, it is divided into three phases to support three degrees of freedom (3DoF), three degrees of freedom plus (3DoF+), and six degrees of freedom (6DoF) experience for viewing the immersive media, as shown in Fig. 16, and they are named as Phase 1a, Phase 1b, and Phase 2, respectively.

The goal of Phase 1a is to provide users with 3DoF experience of yaw, pitch, and roll, as in Fig. 16(a), for watching 360-degree video content. Unlike traditional video coding, 360-degree video includes additional stitching and projection of video content on both capture side and render side, as shown in Fig. 17 [72]-[75]. A 360-degree video, captured by a 360-degree video capture device or generated by multiple stitched videos, is a projection mapped [76] as a rectangular format (e.g. equirectangular projection (ERP)) followed by encoding. After transmission and decoding, the video is rendered on the sphere, and a user can view the sphere to experience the 360-degree environment through VR devices such as HMD. Phase 1a was completed in late 2017. However, the lack of the degree of freedom, i.e. parallax effect, to match the head movement makes the experience unnatural.

To overcome this, 3DoF+ in Phase 1b [77] and 6DoF in Phase 2 [78] of the MPEG-I work allow limited and significant translational movements of the user viewpoint within the 360-degree video, respectively. For instance of 3DoF+ in Fig. 16(b), a user sitting in a rotating chair or standing but without taking steps during watching VR [79]. The work of 3DoF+ is planned to be ready in 2020. In Fig. 16(c), it shows the example of 6DoF where a user can even walk freely in an environment to view the scene at diverse viewing positions and angles [80]. The specification is expected to be ready in 2021 or 2022.

To support 3DoF+/6DoF, the HMD must provide a large number of video viewpoints that the user watches. An



(a) (b) (c)

Fig. 16. *Degree of freedom* ***(a)*** *3DoF,* ***(b)*** *3DoF+,* ***(c)*** *6DoF*

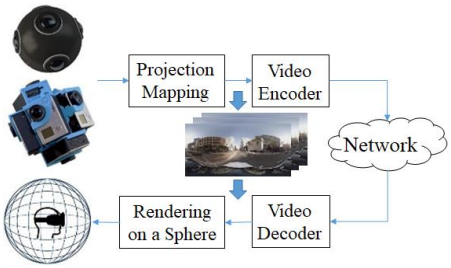
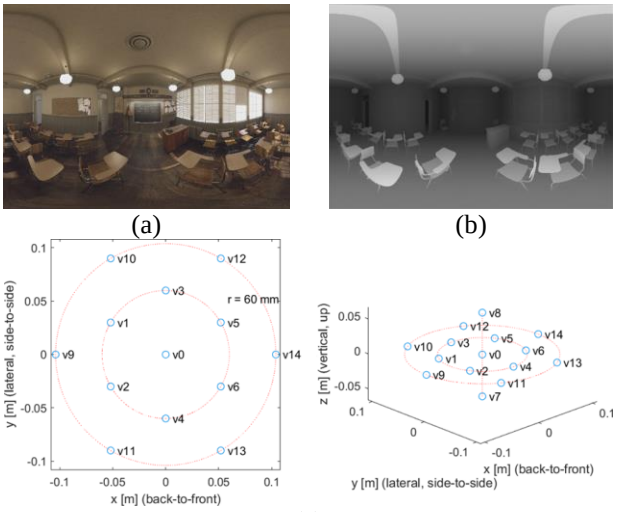


Fig. 17. *360-degree video coding*



(a) (b) (c)

Fig. 18. *Example of 3DoF+ sequence "ClassroomVideo"* ***(a)*** *texture,* ***(b)*** *depth map,* ***(c)*** *camera positions*

example of the 3DoF+ testing sequence "ClassroomVideo" is illustrated in Fig. 18. The source views are positioned as a hexagonally-packed circular disc with an additional top and bottom views, depicted in Fig. 18(c). Fig. 19 then depicts the 3DoF+ software platform framework [81]. As the number of views is large, source view pruning is performed to limit the transmitted views. First, the central view is synthesized. Second, sparse source views are generated in which pixels overlap with the central view are discarded. Partitioning and packing are then performed to have a more compact representation for compression. Third, these textures and depth maps are encoded and transmitted. Resolution is also suggested to be reduced for further bitrate reduction. Since both 3DoF+ and 6DoF videos response to the user's movement, a more sophisticated reference view synthesizer (RVS) [82]-[83] was designed to synthesize the immediate views using existing views. By using RVS, virtual views can be synthesized by more than two source views, which is much more flexible than the DIBR in 3D-HEVC which just picks the nearest left and right source views for synthesis. Due to the multiple source views for

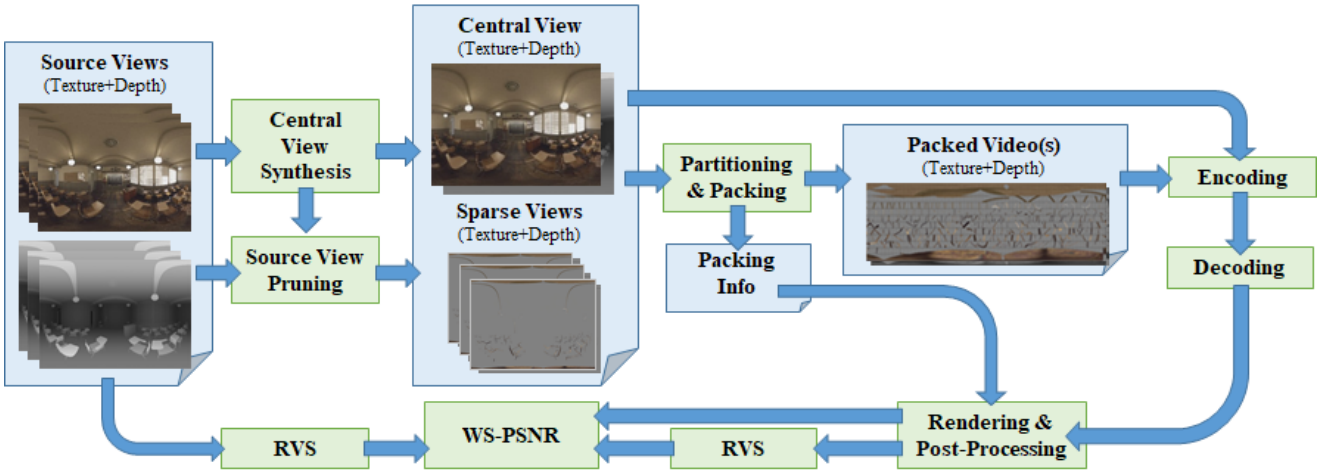


Fig. 19. *Framework of 3DoF+ software platform*

view synthesis, RVS can even allow a larger baseline between source views. It is crucial for 3DoF+ and 6DoF that allow translational movements of the user viewpoint. Particularly, in 360-degree videos, as ERP is used, WS-PSNR [84]-[85] is employed for objective quality measurement for the consideration of reconstructed video in the spherical domain.

As an extremely large volume of data is generated by both high-resolution texture and depth map in 3DoF+/6DoF videos, high coding efficiency is desired. Recently, the MPEG-I group has CfP [86] and Exploration Experiments (EE) [87] for testing 3D-HEVC and the future Versatile Video Coding (VVC) [88], which will be finalized in 2020. For unchanged MVD coding approach based on 3D-HEVC, a study on [89]-[90] indicates that about 0.04 bits per pixel including depth maps are required for a 3DoF+ video with 16 to 25 cameras with the resolution of 3840×2160 each, and a total of 150 to 240 Mbps for frame rate of 30 fps. In HMD applications, it needs a higher frame rate of at least 90 to 120 fps. The overall bitrate will then rise to three to four times. Since this scenario requires a large amount of data to be transmitted, further enhancement of MVD coding becomes a challenging task in the coming future.

The depth map plays a vital role for sophisticated view synthesis in immersive media standardization development. The characteristics of 3DoF+/6DoF videos differ substantially from the conventional depth map in 3D-HEVC. However, the state-of-the-art video codecs, that are designed for depth maps with straight edges and translational motions, are not well suited for a depth map in ERP format, as shown in Fig. 18(b). A promising approach that meets these specific requirements is necessary. Various depth map coding related research works have been proposed for depth estimation [91], view synthesis [92], bitrate reduction [93]-[94] and others [89], [95]-[96] in 3DoF+/6DoF realization, but they are still at the very early stage of technological development. MPEG is still developing the standard for immersive media. As mentioned, it is still during CfP for 3DoF+ and EEs for 6DoF, all methods in the literature are still very primitive, and there is plenty of room for further development and improvement. Research works related to robust depth estimation, enhanced view synthesis and efficient depth coding are crucial in 3DoF+ and 6DoF realizations. With such large number of views, fast approaches are also definitely essential.

For instance, several research works have been investigated on fast approaches for 360-degree video coding [97]-[103] but without any depth information, i.e. 3DoF. The algorithms in [97]-[99] proposed to have fast intra prediction, while the works in [100]-[101] and [102] suggested having fast CU approaches and fast PU approach respectively. Besides, the algorithm in [103] developed adaptive MV resolution to achieve a low-complexity encoder. On the other hand, some research works focus on adaptive encoding or streaming for VR according to the viewer's viewport [104]-[106]. But they aim at reducing the required bandwidth for VR application by the feedback channel of the viewer's viewport or head movement. To the best of our knowledge, there are no fast approaches proposed for 3DoF+/6DoF yet since the CTC [107] and the test model for immersive videos (TMIV) [108] in August 2019 are still developing, which are still very primitive. Thus, there are plenty of rooms for improvements to reduce the encoding complexity of depth maps in 3DoF+/6DoF, and we believe the machine learning approach is one of the fruitful directions for future research in this area.

7. Conclusions

Experts of JCT-3V have developed the 3D extension of HEVC (3D-HEVC). It offers a joint coding solution for texture videos and depth maps for different 3D displays, and this new 3D video format is referred to as MVD. The depth map, that records the distance of objects from the camera, is used to help view the synthesis process, but its characteristics differ substantially from video data. To improve the coding performance of depth maps, 3D-HEVC includes several new depth intra coding tools at the expense of increased complexity due to a flexible quadtree CU/PU partitioning structure and a huge number of intra mode candidates. In this paper, we reviewed the technological advances in 3D-HEVC and the current research works of its fast algorithms. We also presented a machine learning-based approach to expedite the depth map coding, and simulation results show that this state-of-the-art approach can achieve significant improvement over the other approaches.

As the VR/AR/MR market is booming recently, ISO/IEC MPEG is currently working on the new era of future immersive media by MPEG-I. The MVD coding technologies for MPEG-I are also under exploration in the MPEG's working group. Therefore, it is expected that MVD including

depth map coding will have a remarkable impact on the advancement of future VR/AR/MR video technology.

8. Acknowledgments

This work was supported by the Hong Kong Research Grants Council under Research Grant PolyU 152112/17E.

References

[1] Schreer, O., Feldmann, I., Atzpadin, N., Eisert, P., Kau, P., and Belt, H.J.W.: '3D presence - a system concept for multi-user and multi-party immersive 3D video conferencing', Proc. European Conf. Vis. Media Produ. (CVMP), London, U.K., November 2008, pp. 1-8

[2] 'Microsoft Company', http://www.xbox.com/, accessed 27 March 2019

[3] Vetro, A., Wiegand, T., and Sullivan, G.J.: 'Overview of the stereo and multi-view video coding extensions of the H.264/MPEG-4 AVC standard', Proc. IEEE, 2011, 99, (4), pp. 626-642

[4] Chen, Y., Hannuksela, M.M., Suzuki, T., and Hattori, S.: 'Overview of the MVC+D 3D video coding standard,' J. Vis. Commun. Image Represent, 2014, 25, (4), pp. 679-688

[5] Smolic et al.: 'Multi-view video plus depth (MVD) format for advanced 3D video systems', ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JVT-W100, San Jose, April 2007

[6] ISO/IEC JTC1/SC29/WG11, 'Call for proposals on 3D video coding technology', Motion Picture Experts Group (MPEG), document N12036, March 2011

[7] Muller, K., Merkle, P., and Wiegand, T.: '3D video representation using depth maps', Proc. of IEEE Special Issue 3D Media Displays, 2011, 99, (4), pp. 643-656

[8] Lee, T.K., Chan, Y.-L., and Siu, W.-C.: 'Adaptive search range for HEVC motion estimation based on depth Information', IEEE Trans. Circuits Syst. Video Technol., 2017, 27, (10), pp. 2216-2230

[9] Kau, P., Atzpadin, N., Fehn, C., Muller, M., Schreer, O., Smolic, A., and Tanger, R.: 'Depth map creation and image based rendering for advanced 3DTV services providing interoperability and scalability', Signal Process. Image Commun. Special Issue 3DTV, 2007, 22, (2), pp. 217-234

[10] Sullivan, G.J., Ohm, J.-R., Han, W.-J., and Wiegand, T.: 'Overview of the high efficiency video coding (HEVC) standard', IEEE Trans. Circuits Syst. Video Technol., 2012, 22, (12), pp. 1649-1668

[11] High Efficiency Video Coding, document Rec. ITU-T H.265, October 2014

[12] Muller K., et al.: '3D High-efficiency video coding for multi-view video and depth data', IEEE Trans. Image Process., 2013, 22, (9), pp. 3366-3378

[13] Tech, G., Wegner, K., Chen, Y., and Yea, S., 3D-HEVC Draft Text 7, document JCT3V-K1001, Geneva, Switzerland, February 2015

[14] Tsang, S.-H., Chan Y.-L., and Siu, W.-C.: 'Efficient intra prediction algorithm for smooth regions in depth coding', Electron. Lett., 2012, 48, (18), pp. 1117-1119

[15] Merkle P., et al.: 'The effects of multiview depth video compression on multiview rendering', J. Signal Process. Image Commun., 2009, 24, (1), pp. 73-88

[16] Chen, Y., Tech, G., Wegner, K., and Yea, S.: 'Test Model 11 of 3D-HEVC and MV-HEVC', ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-K1003, Geneva, Switzerland, February 2015

[17] Kim, W.-S., Ortega, A., Lai, P., Tian, D., and Gomila, C.: 'Depth map coding with distortion estimation of rendered view', SPIE Conference Series, vol. 7543, pp. 7540B-75430B-10, January 2010

[18] Muller, K., Merkle, P., Tech, G., and Wiegand, T.: '3D video coding with depth modeling modes and view synthesis optimization', Proc. Asia-Pacific Signal Info. Process. Association Annual Summit Conf. (APSIPA ASC), Hollywood, U.S.A., December 2012, pp. 1-4

[19] Dou, H., Chan, Y.-L., Jia, K.B., Siu, W.-C., Liu, P.-Y., and Wu, Q.: 'An adaptive segment-based view synthesis optimization method for 3D-HEVC', Proc. Asia-Pacific Signal Info. Process. Association Annual Summit Conf. (APSIPA ASC), Hong Kong, China, December 2015, pp. 297-302

[20] Dou, H., Chan, Y.-L., Jia, K.B., and Siu, W.-C.: 'View Synthesis optimization based on texture smoothness for 3D-HEVC', Proc. Int. Conf. on Acoustics, Speech and Signal Process., Brisbane, Queensland, Australia, April 2015, pp. 1443-1447

[21] Tech G., et al.: '3D video coding using the synthesized view distortion change', Proc. Picture Coding Symposium, Krakow, Poland, May 2012, pp. 25-28

[22] Chen, Y., Liu, H.B., Zhang, L.: 'CE2: Sub-PU based MPI', ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-G0119, San José, U.S.A., January 2014

[23] Winken, M., Schwarz, H., and Wiegand, T.: 'Motion vector inheritance for high efficiency 3D video plus depth coding', Proc. PCS 2012, Picture Coding Symposium, Krakow, Poland, May 2012

[24] Jung, J., Mora, E.: 'Incorporated depth quadtree prediction', ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-B0068, Shanghai, China, October 2012

[25] Lee, J.Y., Park, M.W., and Kim, C.: '3D-CE1: Depth Intra Skip (DIS) Mode', ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-K0033, Geneva, Switzerland, February 2015

[26] Jager, F.: 'Simplified depth map intra coding with an optional depth lookup table', Proc. Int. Conf. 3D Imaging, Liège, Belgium, December 2012, pp. 1-4

[27] '3D-HEVC Reference Software: HTM-16.0', https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-16.0/, accessed 27 March 2019

[28] Muller, K., and Vetro, A.: 'Common test conditions of 3DV core experiments', ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-G1100, San Jose, January 2014

[29] Chen, H., Fu, C.-H., Zhang, Y., Chan, Y.-L., and Siu W.-C.: 'Early merge mode decision for depth maps in 3D-HEVC', Proc. IEEE Int. Conf. Digital Signal Process. (DSP), London, U.K., August 2017, pp. 1–5

[30] Gu, Z.Y., Zheng, J.H., Ling, N., and Zhang, P.: 'Fast depth modeling mode selection for 3D HEVC depth intra coding', Proc. IEEE Int. Conf. Multimedia Expo Workshop (ICMEW), July 2013, pp. 1-4

[31] Gu, Z.Y., Zheng, J.H., Nam, L., and Zhang, P.: 'Fast bi-partition mode selection for 3D HEVC depth intra

coding', Proc. IEEE Int. Conf. Multimedia Expo Workshop (ICME), Chengdu, China, July 2014, pp. 1-6

[32] Silva, T.D., Agostini, L., and Cruz, L.D.S.: 'Complexity reduction of depth intra coding for 3D video extension of HEVC', Proc. IEEE Int. Conf. Vis. Commu. and Image Process (VCIP), Valletta, Malta, December 2014, pp. 229-232

[33] Zhang, Q.W., Li, N.N., Xun L.X., and Gan, Y.: 'Effective early terminate algorithm for depth map intra coding in 3D-HEVC', Electron. Lett., 2014, 50, (14), pp. 994-996

[34] Park, C.S.: 'Edge-based intra mode selection for depth-map coding in 3D-HEVC', IEEE Trans. Image Process., 2015, 24, (1), pp. 155-162

[35] Sanchez, G., Saldanha, M., Balota, G., Zatt, B., Porto, M., and Agostini, L.: 'Complexity reduction for 3D-HEVC depth maps intra-frame prediction using simplified edge detector algorithm', Proc. IEEE Int. Conf. Image Process. (ICIP), Paris, France, October 2014, pp. 3209-3213

[36] Fu, C.-H., Zhang, H.-B, Su, W.-M, Tsang, S.-H., and Chan, Y.-L.: 'Fast wedgelet pattern decision for DMM in 3D-HEVC', Proc. IEEE Int. Conf. Digital Signal Process. (DSP), Singapore, July 2015, pp. 477-481

[37] Zhang, H.-B., Fu, C.-H., Chan, Y.-L., Tsang, S.-H., Siu, W.-C., and Su W.-M.: 'Efficient wedgelet pattern decision for depth modeling modes in three-dimensional high-efficiency video coding', Journal of Electronic Imaging, 2016, 25(3), pp. 033023

[38] Merkle, P., Müller, K., Zhao, X., Chen, Y., and Zhang, L.: 'Simplified wedgelet search for DMM modes 1 and 3', ITU-T SG 16WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-B0039, Shanghai, China, October 2012

[39] Zhang, M.M., Zhao, C., Xu, J.Z., and Bai, H.H.: 'A fast depth-map wedgelet partitioning scheme for intra prediction in 3D video coding', Proc. IEEE Int. Sympo. Circuits and Syst., Beijing, China, May 2013, pp. 2852-2855

[40] Sanchez, G., Saldanha, M., Balota, G., Zatt, B., Porto, M., and Agostini, L.: 'A complexity reduction algorithm for depth maps intra prediction on the 3D-HEVC', Proc. IEEE Int. Conf. Vis. Commu. and Image Process., Valletta, Malta, December 2014, pp. 137-140

[41] Lucas, L.F.R., Wegner, K., Rodrigues, N.M.M., Pagliari, C.L., Silva E.A.B., and Faria, S.M.M.: 'Intra predictive depth map coding using flexible block partitioning', IEEE Trans. Image Process., 2015, 24, (11), pp. 4055-4068

[42] Li, C.Y., Jin, X. and Dai, Q.H.:'A novel distortion model for depth coding in 3D-HEVC', Proc. IEEE Int. Conf. Image Process., Paris, France, October 2014, pp. 3228-3232

[43] Byung T.O., and Kwan, J.O.: 'View synthesis distortion estimation for AVC- and HEVC-compatible 3-D video coding', IEEE Trans. Circuits Syst. Video Technol., 2014, 24, (6), pp.1006-1015

[44] Zhang, H.-B., Fu, C.-H., Chan, Y.-L., Tsang, S.-H., Siu, W.-C.: 'Probability-based depth intra mode skipping strategy and novel VSO metric for DMM decision in 3D-HEVC', IEEE Trans. Circuits Syst. Video Technol., 2018, 28, (2), pp. 513-527

[45] Gu, Z.Y., Zheng, J.H., and Ling, N.: 'Fast intra SDC coding for 3D-HEVC intra coding', ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-I0123, Sappor, Japan, July 2014

[46] Lee, J.Y., Park, M.W., and Jin, Y.: '3D-CE2 related: Fast SDC DC offset decision', ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCT3V-I0084, Sapporo Japan, July 201

[47] Zhang, H.-B., Fu, C.-H., Su, W.-M., Tsang, S.-H., Chan, Y.-L: 'Adaptive fast intra mode decision of depth map coding by low complexity RD-Cost in 3D-HEVC', Proc. IEEE Int. Conf. Digital Signal Process. (DSP), Singapore, July 2015, pp. 487-491

[48] Zhang, H.-B, Fu, C.-H., Chan, Y.-L., Tsang, S.-H, Siu, W.-C.: 'Efficient depth intra mode decision by reference pixels classification in 3D-HEVC', Proc. IEEE Int. Conf. on Image Processing (ICIP), Quebec City, QC, Canada, September 2015, pp. 961-965

[49] Zhang, H.-B, Tsang, S.-H., Chan, Y.-L., Fu, C.-H, Su, W.-M.: 'Early determination of intra mode and segment-wise DC coding for depth map based on hierarchical coding structure in 3D-HEVC', Proc. Asia-Pacific Signal Info. Process. Association Annual Summit Conf. (APSIPA ASC), Hong Kong, China, December 2015, pp. 374-378

[50] Shen, L., Liu, Z., Zhang, X., Zhao, W., Zhang, Z.: 'An effective CU size decision method for HEVC encoders', IEEE Trans. on Multimedia, 2013, 15, (2), pp. 465-470

[51] Shang, X., Wang, G., Fan, T., Li, Y.: 'Fast CU size decision and PU mode decision algorithm in HEVC intra coding', Proc. IEEE Int. Conf. on Image Processing (ICIP), Quebec City, QC, Canada, September 2015, pp. 1593-1597

[52] Kim, D.H., Kim, Y.H., Park, W.C.: 'Selective CU depth range decision algorithm for HEVC encoder', Proc. IEEE Int. Symposium on Consumer Electronics (ISCE), JeJu Island, South Korea, August 2014, pp. 1-2

[53] Nishikori, T., Nakamura, T., Yoshitome, T., Mishiba, K.: 'A fast CU decision using image variance in HEVC intra coding', in Proc. IEEE Symposium on Industrial Electronics & Applications, September 2013, pp. 52-56

[54] Min, B., Cheung, R.C.C.: 'A Fast CU size decision algorithm for the HEVC intra encoder', IEEE Trans. Circuits Syst. Video Technol., 2015, 25, (5), pp. 892-896

[55] Mora, E.G., Jung, J., Cagnazzo, M., and Pesquet, B.: 'Initialization, limitation, and predictive coding of the depth and texture quadtree in 3D-HEVC', IEEE Trans. Circuits Syst. Video Technol., 2014, 24, (9), pp. 1554-156

[56] Zhang, H.-B., Chan, Y.-L., Fu, C.-H, Tsang, S.-H, Siu, W.-C: 'Quadtree decision for depth intra coding in 3D-HEVC by good feature', Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, March 2016, pp. 1481-1485

[57] Kim, M., Lim, N., and Song, L.: 'Fast single depth intra mode decision for depth map coding in 3D-HEVC', Proc. IEEE Int. Conf. on Multimedia Expo (ICME), Turin, Italian, June 2015, pp. 1-6

[58] Chen, H., Fu, C.H., Chan, Y.-L., and Zhu, X.: 'Early intra block partition decision for depth maps in 3D-HEVC', Proc. IEEE Int. Conf. on Image Processing (ICIP), Athens, Greece, October 2018, pp. 2381-8549

[59] Oliveira, J.F.D., and Alencar, M.S.: 'Online learning early skip decision method for the HEVC inter process using the SVM-based Pegasos algorithm', Electronics Letters, 2016, 52, (14), pp. 1227-1229
[60] Du, B., Siu, W.C., and Yang, X.: 'Fast CU partition strategy for HEVC intra-frame coding using learning approach via random forests', Proc. Asia-Pacific Signal Info. Process. Association Annual Summit Conf. (APSIPA ASC), December 2016, pp. 1085-1090
[61] Liu, X., Li, Y., Liu, D., Wang, P., Yang, L.T.: 'An adaptive CU size decision algorithm for HEVC intra prediction based on complexity classification using machine learning', IEEE Trans. Circuits Syst. Video Technol., 2019, 29, (1), pp. 144-155
[62] Zhu, L., Zhang, Y., Pan, Z., Wang, R., Kwong, S., and Peng, Z.: 'Binary and multi-class learning based low complexity optimization for HEVC Encoding', IEEE Trans. Broadcasting, 2017, 63, (3), pp. 547-561
[63] 'Decision Trees scikit-learn 0.19.1 documentation', http://scikit-learn.org/stable/modules/tree.html, accessed 27 March 2019
[64] Safavian S.R., and Landgrebe, D.A.: 'A survey of decision tree classifier methodology', IEEE Trans. Syst., Man. Cybern., 1991, 21, (3), pp. 660-674
[65] Bjontegaard, G: 'Calculation of average PSNR differences between RD curves', 2001: ITU-T Video Coding Experts Group (VCEG)
[66] 'Technical Report of the Joint Ad Hoc Group for Digital Representations of Light/Sound Fields for Immersive Media Applications', ISO/IEC JTC1/SC29/WG11 MPEG, document N16352, Geneva, Switzerland, June 2016
[67] 'Summary of Survey on Virtual Reality', ISO/IEC JTC1/SC29/WG11 MPEG, document N16542, Chengdu, China, October 2016
[68] 'MPEG Strategic Standardisation Roadmap', ISO/IEC JTC1/SC29/WG11 MPEG, document N16316, Geneva, Switzerland, June 2016
[69] 'MPEG121 Version of MPEG Standardisation Roadmap', ISO/IEC JTC1/SC29/WG11 MPEG, document N17332, Gwangju, South Korea, January 2018.
[70] 'MPEG-I: Coded Representation of Immersive Media', https://mpeg.chiariglione.org/standards/mpeg-i, accessed 28 March 2019
[71] Schwarz S., Preda M., Baroncini V., et al.: 'Emerging MPEG Standards for Point Cloud Compression', IEEE J. Emerg. Sel. Topics Circuits Syst., 2019, 9, (1), pp. 133-148
[72] 'G-PCC Codec Description v2', ISO/IEC JTC1/SC29/WG11 MPEG, document N18189, Marrakech, Morocco, January 2019
[73] 'WD on ISO/IEC 23000-20 Omnidirectional Media Application Format', ISO/IEC JTC1/SC29/WG11 MPEG, document N16189, Geneva, Switzerland, June 2016
[74] 'How OMAF Fulfils MPEG-I Phase 1a Requirements', ISO/IEC JTC 1/SC 29/WG 11 MPEG, document N17372, Gwangju, South Korea, January 2018
[75] Skupin R., Sanchez Y., Wang Y.-K, et al.: 'Standardization status of 360 degree video coding and delivery', Proc. IEEE Int. Conf. Vis. Commun. Image Process. (VCIP), Saint Petersburg, Florida, U.S.A., December 2017, pp. 1-4
[76] Chen Z., Li Y., and Zhang Y.: 'Recent advances in omnidirectional video coding for virtual reality: projection and evaluation', J. Signal Process., 2018, 146, pp. 66-78
[77] 'Algorithm Descriptions of Projection Format Conversion and Video Quality Metrics in 360Lib Version 5', ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC29/WG11 JVET, document JVET-H1004, Macau, China, October 2017
[78] 'Requirements MPEG-I Phase 1b', ISO/IEC JTC1/SC29/WG11 MPEG, document N17331, Gwangju, South Korea, January 2018
[79] 'Requirements MPEG-I Phase 2', ISO/IEC JTC1/SC29/WG11 MPEG, document N18127, Marrakech, Morocco, January 2019
[80] 'MPEG-I Phase 1 Use Cases (v1.5)', ISO/IEC JTC1/SC29/WG11 MPEG, document N17886, Ljubljana, Slovenia, July 2018
[81] 'MPEG-I Phase 2 Use Cases', ISO/IEC JTC1/SC29/WG11 MPEG, document N17932, Macau, China, October 2018
[82] '3DoF+ Software Platform Description', ISO/IEC JTC1/SC29/WG11 MPEG, document N18070, Macau, China, October 2018
[83] 'Reference View Synthesizer (RVS) Manual', ISO/IEC JTC1/SC29/WG11 MPEG, document N18068, Macau, China, October 2018
[84] 'Reference View Synthesizer (RVS)', http://mpegx.int-evry.fr/software/MPEG/Explorations/3DoFplus/RVS, accessed 28 March 2019
[85] Sun Y., Lu A., and Yu L.: 'Weighted-to-spherically-uniform quality evaluation for omnidirectional video.' IEEE Signal Process. Lett., 2017, 24, (9), pp.1408-1412
[86] 'WS-PSNR Calculation Software', http://mpegx.int-evry.fr/software/MPEG/Explorations/3DoFplus/WS-PSNR, accessed 28 March 2019
[87] 'Call for Proposals on 3DoF+ Visual', ISO/IEC JTC1/SC29/WG11 MPEG, document N18145, Marrakech, Morocco, January 2019
[88] 'Exploration Experiments for MPEG-I: 6DoF', ISO/IEC JTC 1/SC 29/WG 11 MPEG, document N18170, Marrakech, Morocco, January 2019
[89] 'Working Draft of Versatile Video Coding', ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC29/WG11 JVET, document JVET-L1001, Macao, China, October 2018
[90] Hinds A. T., Doyen D., and Carballeira P.: 'Toward the realization of six degree-of-freedom with compressed light fields', Proc. IEEE Int. Conf. Multimedia Expo (ICME), Hong Kong, China, August 2017, pp. 1171-1176
[91] Lafruit G., Schenkel A., Tulvan C., et al.: 'MPEG-I Coding performance in immersive VR/AR applications' Proc. IBC Conf. (IBC), Amsterdam, Netherlands, September 2018, pp 1-9
[92] Wegner K., Stankiewicz O., Grajek T., et al.: 'Depth estimation from stereoscopic 360-Degree video', Proc. IEEE Int. Conf. Image Process. (ICIP), Beijing, China, October 2017, pp. 2945-2948
[93] Wegner K., Losiewicz D., Grajek T., et al.: 'Omnidirectional view synthesis and test images', Proc.

IEEE Int. Conf. Signals Electron. Syst. (ICSES), Krakow, Poland, September 2018, pp. 130-133
[94] Jeong J., Jang D., Son J., et al.: '3DoF+ 360 video view synthesis for immersive VR video streaming', Proc. Int. Conf. Inform. (ICTC), Jeju, South Korea, October 2018, pp. 581-586
[95] Jeong J., Jang D., and Son J.: '3DoF+ 360 video location-based asymmetric down-sampling for view synthesis to immersive VR video streaming', Sensors, 2018, 18, (9), 3148, pp. 1-20
[96] Ray B., Jung J, and Larabi C.: 'On the possibility to achieve 6-DoF for 360 video using divergent multi-view content', Proc. European Signal Process. Conf. (EUSIPCO), Rome, Italy, September 2018, pp. 211-215
[97] Huang J., Chen Z., Ceylan D., et al.: '6-DOF VR videos with a single 360-camera', Proc. IEEE Virtual Reality (VR), Los Angeles, California, U.S.A., March 2017, pp. 37-44
[98] Wang Y., Li Y., Yang D., and Chen Z.: 'A fast intra prediction algorithm for 360-degree equirectangular panoramic video,' Proc. IEEE Int. Conf. Vis. Comm. Image Process. (VCIP), Saint Petersburg, Florida, U.S.A., December 2017, pp. 1-4
[99] Liu Z., Xu C., Zhang M., and Guan X.: 'Fast Intra Prediction Algorithm for Virtual Reality 360 Degree Video Based on Improved RMD,' Proc. Data Compression Conf. (DCC), Snowbird, Utah, U.S.A., March 2019, pp. 593
[100] Storch I., Zatt B., Agostini L., da Silva Cruz L. A., and Palomino D.: 'FastIntra360: A Fast Intra-Prediction Technique for 360-Degrees Video Coding,' Proc. Data Compression Conf. (DCC), Snowbird, Utah, U.S.A., March 2019, pp. 605
[101] Liu Z., Song P., and Zhang M.: 'A CU Split Early Termination Algorithm Based KNN for 360-Degree Video,' Proc. Data Compression Conf. (DCC), Snowbird, Utah, U.S.A., March 2019, pp. 594
[102] Guan X., Dong X., Zhang M., and Liu Z.: 'Fast Early Termination of CU Partition and Mode Selection Algorithm for Virtual Reality Video in HEVC,' Proc. Data Compression Conference (DCC), Snowbird, Utah, U.S.A., March 2019, pp. 576
[103] Zhang M., Su R., Liu Z., Mao F., and Yue W.: 'Fast PU Early Termination Algorithm Based on WMSE for ERP Video Intra Prediction,' Proc. Data Compression Conf. (DCC), Snowbird, Utah, U.S.A., March 2019, pp. 614
[104] Ray B., Jung J. and Larabi M.: 'A Low-Complexity Video Encoder for Equirectangular Projected 360 Video Content,' Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process. (ICASSP), Calgary, Alberta, Canada, April 2018, pp. 1723-1727
[105] Ozcinar C., Cabrera J., and Smolic A.: 'Visual Attention-Aware Omnidirectional Video Streaming Using Optimal Tiles for Virtual Reality,' IEEE J. Emerg. Sel. Topics Circuits Syst., 2019, 9, (1), pp. 217-230
[106] Xu A., Chen X., Liu Y., and Wang Y.: 'A Flexible Viewport-Adaptive Processing Mechanism for Real-Time VR Video Transmission,' Proc. IEEE Int. Conf. Multimedia Expo Workshop (ICMEW), Shanghai, China, July 2019, pp. 336-341
[107] Xu Z., Zhang X., Zhang K., and Guo Z., 'Probabilistic Viewport Adaptive Streaming for 360-degree Videos,' Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Florence, Italy, May 2018, pp. 1-5
[108] 'Common Test Conditions for Immersive Video', ISO/IEC JTC1/SC29/WG11 MPEG, document N18563, Gotenburg, Sweden, August 2019
[109] 'Test Model 2 for Immersive Video', ISO/IEC JTC1/SC29/WG11 MPEG, document N18577, Gotenburg, Sweden, August 2019