Fast depth intra coding based on texture feature and spatio-temporal correlation in 3D-HEVC

Tiansong Li | Li Yu | Hongkui Wang | Yamei Chen

The School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China

Correspondence
Li Yu, The School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China.
Email: hustlyu@hust.edu.cn

Abstract
To alleviate the computation burden of the depth intra coding in 3D-HEVC, a complexity reduction scheme based on texture feature and spatio-temporal correlation is proposed. Firstly, a maximum splitting depth layer decision algorithm is proposed to reduce unnecessary splitting depth layer of the coding tree unit utilising the information of the previous encoded I frame in the same view. Secondly, a new texture complexity model is built by pixel-based statistical method combined with edge detection. Based on the proposed model, the coding unit block is divided into the smooth block, texture or edge block. On the coding unit level, an early termination of coding unit splitting algorithm for smooth blocks is proposed to filter out unnecessary coding blocks. Thirdly, on the predicting unit level, a fast candidate mode decision algorithm considering predicting unit’s types and spatial correlation is proposed to decide the candidate mode list directly. Experimental results describe that the proposed algorithm reduces 53.8% depth intra coding time on average, with 0.43% BD-rate loss on synthesised views.

1 | INTRODUCTION

With the rapid development of multimedia technology, simple two-dimensional video is not satisfying people’s visual perception of the real world. Thus, three-dimensional video (3DV) with the sense of interaction and immersion starts to emerge. Compared with two-dimensional video, 3DV adds depth perception on the basis of planar vision, forms stereo vision, and brings an immersive feeling to the audience. At the same time, 3DV also provides the choice of arbitrary views, and viewers can choose different perspectives to watch depending on their own preferences. The collection of real-world information has changed from passive acceptance to active acquisition. 3DV format is usually presented by the multi-view video plus depth (MVD) [1, 2], which consists of the limited number of texture maps and corresponding depth maps. Hence, the amount of 3DV data is much larger than two-dimensional video, which brings great pressure to the transmission of video data. In order to meet the existing storage capacity and network transmission conditions, how to effectively compress 3DV is a hot issue of current research.

Currently, 3D-HEVC is the latest 3D video coding standard [3–5], which can utilise inter-view dependency to compress texture video and depth video efficiently. Depth video is used to synthesise arbitrary-view video by using depth-image-based rendering (DIBR) technology [6, 7], so as to provide the audience more choices of perspective. Therefore, the coding quality of the depth video is very critical for 3D-HEVC. In 3D-HEVC standard, depth intra coding still follows intra coding technology in HEVC [8]. Since the depth map is composed of a large flat area separated by sharp edges, some additional coding tools have also been added specifically for this feature of depth map, such as depth modelling mode (DMM) [9], segment-wise depth coding (SDC) [10] and view synthesis optimization (VSO) [11] etc. However, these tools also bring great coding complexity for depth intra coding. As far as we know, the encoding time of depth intra coding is approximately six times longer than that of texture intra coding in 3D-HEVC. Therefore, in order to realise the real-time application of the 3D-HEVC standard, a fast and efficient coding method for the depth map is very necessary.

Recently, a large number of fast depth intra coding algorithms have been proposed for 3D-HEVC, which can be summarised as three types. The summary of these state-of-the-art algorithms...
TABLE 1 The summary of recent state-of-the-art algorithms

| Type | Fast depth intra coding algorithms |
|------|----------------------------------|
| I    | Chen [12], Fu [13], Saldanha [14], Chen [15] |
| II   | Shen [16], Hamout [17], Zhang [18], Lei [19], Guo [20] |
|      | Zhao [21], Sanchez [22] [23], Park [24], Zhang [25] |
| III  | Li [26], Peng [27], Jing [28] |

is listed in Table 1. The type I is to reduce the number of unnecessary traversal coding units (CUs) or terminate CU partitioning early, the type II is to reduce the prediction modes or optimise DMMs, and the type III is a comprehensive algorithm including the first two types. For the type I, Chen et al. [12] proposed an early termination of CU splitting when the rate-distortion cost (RDcost) of the current block with depth intra skip (DIS) mode is less than or equal to four times the RDcost of the first sub-block with DIS mode. Fu et al. [13] proposed to terminate CU splitting early if DIS-parent CU is detected at the higher depth levels. Saldanha et al. [14] utilised static CU splitting decisions trees based on data mining and machine learning to achieve early termination of CU splitting. In our previous work [15], we proposed an adaptive QP convolutional neural network (AQ-CNN) structure to terminate CU partition early. For the type II, Shen et al. [16] utilised intra coding information from spatiotemporal, inter-component, and inter-view neighbouring CUs to reduce the prediction modes. Hamout et al. [17] used tensor feature extraction and big data analysis to speed up the prediction mode decision process. Zhang et al. [18] reduced the prediction modes using the spatial distribution characteristics of the reference pixels. Lei et al. [19] proposed a simplified search algorithm for DMM1. Guo et al. [20] proposed a fast depth intra coding scheme using gray-level co-occurrence matrix (GLCM) to classify the CUs into different types, and avoided checking some unnecessary modes for certain CUs. Zhao et al. [21] optimised DMMs based on the depth of the scene represented by the depth value. Sanchez et al. [22] proposed a fast intra modes scheme by reducing candidate modes and optimise DMM1 [23]. Park et al. [24] used edge classification in the Hadamard transform domain to optimise DMM1. In [25], the optimal prediction mode with segment-wise depth coding in rough mode decision is decided when its RDcost was smaller than the certain threshold and the RDcost of the wedgelet pattern of DMM1 is calculated by the squared Euclidean distance of variances. For the type III, in our previous work [26], the frequency distribution characteristics of the RDcost of the CTU level in depth layer 0 is used to predict the maximum depth layer of CTU, the candidate mode list is decided early based on the frequency distribution characteristics of the RDcost of the optimal rough mode in rough mode decision and the spatial correlation, and a fast wedgelet pattern decision method based on K-Means is proposed to optimise DMM1. Peng et al. [27] set a certain threshold of the minimum RDcost to reduce candidate modes, and using the properties of the current CU to skip smaller CU size. Jing et al. [28] proposed a classification and regression tree to predict the depth layer and reduce intra prediction modes.

To reduce the computational complexity of the depth intra coding, we propose a fast depth intra coding scheme based on texture feature and spatiotemporal correlation. This paper is an extended version of our previous conference paper [29], in which early termination of CU splitting (ETCS) and fast candidate mode decision (FCMD) algorithms are presented. In this paper, a maximum splitting depth layer decision algorithm (MSDLD) is further developed to reduce the number of unnecessary traversal CUs. At the same time, a detailed description for ETCS and FCMD algorithms is presented, and a richer comparison of experimental results is given. The main contributions are listed as follows:

Due to the strong temporal correlation between two consecutive frames in the same view, the MSDLD algorithm is proposed to cut down the number of unnecessary traversal CUs using the maximum splitting depth layer and the RDcost of the co-located CU in the previous encoded frame.

A new texture complexity model is built by pixel-based statistical analysis combined with edge detection. Based on the texture complexity model, the CU block is divided into the smooth block, texture or edge block. Depth texture characteristics and spatial correlation are combined to early determine candidate modes and terminate unnecessary CU splitting.

The remainder of this paper is organised as follows: Section 2 shows Background and motivation. Section 3 describes details of the proposed algorithm. Experimental comparisons are given in Section 4. Finally, Section 5 presents the conclusions.

2 BACKGROUND AND MOTIVATION

The CTU quad-tree partition structure in depth intra coding has the same as that in HEVC [30]. Each CTU covers a square area, which can be split into multiple sub-CUs. As shown in Figure 1, the rate-distortion cost (RDcost) calculation order of CU in CTU is given, where the largest CU size is 64×64 and depth layer is equal to 0, and it can be divided into four 32×32 sub-CUs with depth layer 1. When the smallest CU with 8×8 reached, the partition process is stopped. To obtain the optimal quad-tree partition structure, each CU needs to calculate the RDcost of various CU sizes under the maximum splitting depth layer 3, about the total of 85 CUs (1 CU with 64×64, 4 CUs with 32×32, 16 CUs with 16×16 and 64 CUs with 8×8). Each CU also needs to calculate combinations of various PU sizes and prediction modes, resulting in high coding complexity.

In order to study the maximum depth layer \( D_{\text{max}} \) of the CTU in depth intra coding, the \( D_{\text{max}} \) of the five test sequences is counted under the experimental conditions in Table 4, and the experimental results are shown in Figure 2. It can be seen that, on average, 63.03\% of CU’s \( D_{\text{max}} \) is 0, 11.36\% of CU’s \( D_{\text{max}} \) is equal to 1, and only about 25\% of CU’s \( D_{\text{max}} \) is 2 and 3. Since the depth map is composed of large flat area and small portion of sharp edges, flat areas are more suitable for selecting larger CU size, and sharp edge areas usually choose smaller CU size. Therefore, if the \( D_{\text{max}} \) of the CTU can be determined early, the traversal of unnecessary depth layers will be reasonably terminated, and it will avoid traversing more unnecessary CUs.
In HEVC, intra prediction uses the reconstructed pixel values of neighbouring PUs to predict current PU. The selection of predict modes are the key issues to be solved in intra prediction. HEVC uses larger and more block sizes to adapt to the content characteristics of high-definition video, and supports more types of intra modes to accommodate richer textures. Intra prediction in HEVC supports five PU sizes: 4\times4, 8\times8, 16\times16, 32\times32, 64\times64, each of which corresponds to 35 prediction modes, including Planar mode, DC Mode and 33 angle modes, as shown in Figure 3 left. In summary, as shown in Figure 3 right, the PU and the CU have the same size for intra prediction. Only when the size of the CU is 8\times8, current CU is divided into four sub-PUs. To cut down the computational complexity of the encoder, a fast two-stage intra prediction algorithm is adopted in HEVC. In the first stage, rough mode decision process is conducted, that is, the RDcost function based on sum of absolute transformed difference is used to screen out the N most likely candidate modes from 35 modes (for a PU size with 8\times8 and 4\times4, eight best modes are selected; for PU sizes with 64\times64, 32\times32 and 16\times16, three best modes are selected). Due to the spatial correlation of neighbouring PUs, the two most probable modes of the spatial neighbouring PUs are added to the candidate mode list. In the second stage, the RDcost of all modes in the candidate mode list will be calculated to select the best prediction mode with the minimum RDcost.

As shown in Figure 4, based on above intra coding in HEVC, 3D-HEVC depth map intra prediction is summarised as the following five steps. The first step is the traditional HEVC intra-mode selection process, including rough mode decision and most probable mode process. In the second step, the DMM1 and DMM4 (DMMs) are added to the candidate mode list.
In the third step, all candidates in the candidate mode list are encoded utilising traditional residual coding to calculate their RDcost. As seen, for example, in (4), the view synthesis optimization technique is used to calculate the RDcost ($D_{VSO}$) of each candidates. In conventional video coding system, one commonly used distortion function is the sum of squared differences (SSD), which is defined between original and encoded depth block as

$$D_{depth} = \sum_{(x,y) \in B} [s_D(x,y) - \tilde{s}_D(x,y)]^2 \quad (1)$$

where $s_D(x,y)$ and $\tilde{s}_D(x,y)$ indicate the original and reconstructed depth map, respectively, and $(x,y)$ means the sample position in PU. However, the conventional SSD metric is not an good estimate of the synthesised view distortion. Instead, the following view synthesis distortion metric provides an better estimate by that weighting the depth distortion $D_{depth}$ with the sum of absolute horizontal texture gradients, which is defined by

$$D_{syn} = \sum_{(x,y) \in B} \left( \frac{1}{2} \times \alpha \times |s_T(x,y) - \tilde{s}_T(x,y)| \right. \\
\times \left[ |\tilde{s}_T(x,y) - \tilde{s}_T(x-1,y)| + |\tilde{s}_T(x,y) - \tilde{s}_T(x+1,y)| \right] \quad (2)$$

where $\tilde{s}_T$ indicates the reconstructed texture, and $\alpha$ is proportional coefficient.

$$D_{VSO} = (w_{syn} \times D_{syn} + w_{dep} \times D_{dep})/(w_{syn} + w_{dep}) \quad (3)$$
where $w_{\text{syn}}$ and $w_{\text{dep}}$ denotes the weights for the synthesised view distortion and the depth map distortion.

$$J_{\text{VSO}} = D_{\text{VSO}} + \lambda \times R \tag{4}$$

where $J_{\text{VSO}}$ is the RDcost based on view synthesis optimization, $\lambda$ is the Lagrangian multiplier. The fourth step is to use a fast intra segment-wise depth coding (SDC) technique for all candidates in the candidate mode list. The fifth step is to determine the optimal prediction mode and whether it uses the traditional residual coding or intra SDC.

The time distribution of 3D-HEVC depth intra coding is given in Figure 5. In the total intra encoding time, the encoding time consumption of texture video only accounts for 14%, and the encoding time of depth video accounts for 86%. In depth intra mode decision process, the largest encoding time is step 2 to step 4, which account for 25.18%, 27.50% and 23.39%, respectively. As described in depth intra prediction process above, about 7 to 12 candidate modes are added to the candidate mode list. By calculating the $J_{\text{VSO}}$ of each combination of candidate mode and transform coding method, the optimal combination is selected. Since most candidates in the candidate mode list are not selected, resulting in intolerable time costs in step 3 and step 4. If the candidate modes can be decided early, the encoding complexity will be greatly reduced.

3 | THE PROPOSED FAST ALGORITHM

3D-HEVC achieves efficient encoding for MVD format videos, and the coding technique for depth video plays an important role in 3D-HEVC. The depth video consists of large areas with slowly changing pixels and sharp boundaries, which is mainly used to synthesise virtual views. The boundary information contained in depth map is used to represent the difference between foreground and background, which has a direct impact on the quality of synthesised views. In encoding process, border areas tend to choose more fine divisions, and smooth areas tend to choose larger CU sizes [31]. The CTU based on the quad-tree partition structure needs to traverse the depth layer 0–3 in turn. At each depth layer, a large number of the RDcost calculations are required to select the best prediction mode, which brings very large computational complexity. Through in-depth research on the correlation between two adjacent depth maps, the optimal $D_{\text{max}}$ of the current CTU is determined early. At the same time, by extracting the texture information of the current CU, the partition of the flat CU block can be terminated early, resulting in more encoding time reduction.

In depth intra coding, the Z-Scan order is used to encode each CTU. Hence, when encoding the current CU, the optimal prediction mode of the above CU and the left CU has been determined. Due to the strong spatial correlation in depth map, the coding information of the neighbouring CUs can effectively guide the prediction mode selection of the current CU. There are two PU types in the depth map. One is composed of near-constant or slowly changing depth values, while the other contains sharp boundaries. DC and Planar modes are very suitable for encoding the first PU type. If the texture information of each PU is combined with spatial correlation to more accurately predict the candidate mode set, a large number of redundant prediction modes will be reduced. The DMMs is mainly used to encode PUs with sharp edges. If each PU is pre-processed before DMMs process, by obtaining its edge information and combining spatial correlation, testing DMMs process can be skipped. In this section, we will introduce the proposed fast algorithm scheme in detail.

3.1 | Maximum splitting depth layer decision

In depth video, for two consecutive I frames in the same view, there are only differences in the position of the object due to the
TABLE 2: The \( D_{\text{max}} \) Distribution of The Current CTU When \( f_{\text{curCU}}^\text{max} = 0 (P(D_{\text{max}}^\text{curCU}, f_{\text{curCU}}^\text{max} = 0)) \)

| Sequences     | Depth 0       | Depth 1       | Depth 2       | Depth 3       | \( P_m \) |
|---------------|---------------|---------------|---------------|---------------|-----------|
| Balloons      | 90.91%        | 6.57%         | 1.84%         | 0.68%         | 47.81%    |
| Newspaper     | 88.68%        | 7.35%         | 2.59%         | 1.38%         | 35.75%    |
| GhostTownFly  | 95.45%        | 3.56%         | 0.79%         | 0.20%         | 77.09%    |
| PoznanHall2   | 94.52%        | 3.61%         | 1.19%         | 0.67%         | 82.25%    |
| PoznanStreet  | 88.39%        | 8.22%         | 2.4%          | 0.98%         | 60.16%    |
| AVG           | 91.59%        | 5.87%         | 1.76%         | 0.78%         | 60.61%    |

TABLE 3: The \( D_{\text{max}} \) Distribution of CTU in \( E_0 \) and \( E_1 \)

| Sequences     | \( P_A \) | \( P_b \) | \( P_B \) | \( P_d \) |
|---------------|-----------|-----------|-----------|-----------|
| Balloons      | 98.69%    | 28.26%    | 94.37%    | 3.07%     |
| Newspaper     | 96.14%    | 17.90%    | 90.44%    | 2.34%     |
| GhostTownFly  | 98.74%    | 42.62%    | 95.92%    | 2.25%     |
| PoznanHall2   | 98.73%    | 47.94%    | 95.70%    | 2.46%     |
| PoznanStreet  | 95.14%    | 29.56%    | 94.64%    | 3.59%     |
| AVG           | 97.37%    | 33.26%    | 94.22%    | 2.74%     |

motion of the object or the camera, but they have similar backgrounds. The texture features of the backgrounds or the same object tend to be smooth, and larger CU size are suitable for coding these regions. For these areas containing sharp edges, a smaller CU size can be chosen. Since the current CTU has similar texture feature as the co-located CTU in the previous encoded I frame, their coding information is also basically similar [32]. Consequently, it is feasible to explore encoding information of the co-located CTU in the previous encoded I frame to determine the \( D_{\text{max}} \) of the current CTU in the same view.

Based on the above analysis, due to the strong correlation between co-located positions of two consecutive depth frames in the same view [33], the statistical experiments of the \( D_{\text{max}} \) correlation in two consecutive I frames is investigated under the experimental conditions in Table 4. As is shown in Table 2, \( D_{\text{max}}^\text{colCU} \) denotes the \( D_{\text{max}} \) of the co-located CTU in the previous encoded depth map. The conditional probability \( P(D_{\text{max}}^\text{curCU}, D_{\text{max}}^\text{colCU} = 0) \) represents the distribution of the current CTU when the \( D_{\text{max}} \) of the co-located CTU in the previous encoded depth map is equal to 0. \( P_m \) indicates the percentage of the current CTU in the whole encoded CTUs when \( D_{\text{max}}^\text{colCU} \) is equal to 0. From Table 2, \( P(D_{\text{max}}^\text{curCU}, D_{\text{max}}^\text{colCU} \leq 2 | D_{\text{colCU}} = 0) \) is up to 99.22% and \( P_m \) accounts for 60.61% on average. This means that when \( D_{\text{max}}^\text{colCU} \) is equal to 0, the probability that the \( D_{\text{max}} \) of the current CTU does not exceed 2 reaches 99.22%. Therefore, \( D_{\text{max}}^\text{curCU} \) is set to 2 when \( D_{\text{max}}^\text{colCU} \) is equal to 0.

\[
E_0 = \{ D_{\text{max}}^\text{colCU} = 0 \& f_{\text{curCU}}^0 \leq f_{\text{colCU}}^0 \} \quad (5)
\]

\[
P_A = P\left(D_{\text{max}}^\text{curCU} = 0 | E_0 \right) \quad (6)
\]

\[
E_1 = \{ D_{\text{max}}^\text{colCU} = 1 \& f_{\text{curCU}}^0 \leq f_{\text{colCU}}^0 \} \quad (7)
\]

\[
P_B = P\left(D_{\text{max}}^\text{curCU} = 1 | E_1 \right) \quad (8)
\]

The \( D_{\text{max}} \) and the prediction modes for the CTU are determined by minimising \( J_{\text{VSO}} \). Hence, \( J_{\text{VSO}} \) is an important parameter to predict the \( D_{\text{max}} \). The event \( E_0 \) and the event \( E_1 \) is defined, for example, by (7) and (8), respectively, where \( f_{\text{curCU}}^0 \) and \( f_{\text{colCU}}^0 \) represents \( J_{\text{VSO}} \) of the current CTU and the co-located CTU when the splitting depth layer is equal to 0, \( f_{\text{curCU}}^1 \) represents \( J_{\text{VSO}} \) of the co-located CTU when the maximum depth layer is equal to 1. The percentage of the conditional probabilities \( P_A \) and \( P_B \) are exhibited in Table 3 under the conditions of the event \( E_0 \) and the event \( E_1 \). As is demonstrated in Table 3, 97.37% of CTUs on average choose that the \( D_{\text{max}} \) is equal to 0 when the condition \( E_0 \) is satisfied, the number of CTUs with \( D_{\text{max}} = 1 \) account for 94.22% on average under the condition \( E_1 \). \( P_A \) and \( P_B \) denotes the percentage of the current CTU when the condition \( E_0 \) and \( E_1 \) is satisfied in the whole encoded CTUs, respectively. It can be seen that \( P_A \) appears from 17.90% to 47.94%, 33.26% on average, \( P_B \) accounts for 2.74% on average. Based on the above analysis, the \( D_{\text{max}} \) of the current CTU is defined by

\[
D_{\text{max}} = \begin{cases} 
0, & \text{when } E_0 \text{ is satisfied.} \\
1, & \text{when } E_1 \text{ is satisfied.} \\
3, & \text{otherwise.}
\end{cases} \quad (9)
\]

3.2 Early termination of CU splitting

The selection of the CU size strongly related to the complexity of its texture. Chen et al. [34] proposed a fast CU size selection algorithm for HEVC intra coding using the angular second moment (ASM) of the gray-level co-occurrence matrix (GLCM) to measure image texture, which accelerates the encoding speed of texture video. Considering the difference between the texture features of the depth map and the texture map, the texture feature and edge information of the CU in depth map are extracted by using GLCM [35] and Sobel operator respectively. There are \( n \) grey levels for an image, GLCM computes in the way a pixel with intensity \( i \) occur horizontally, vertically or diagonally to a pixel with intensity \( j \). An element value in the GLCM matrix is defined by accumulating how many times a pixel pair \((i, j)\) occurs, which represents a \( n \times n \) dependence matrix square. The mathematic expression of the grey level co-occurrence matrix is defined as

\[
f_{i,j}(x, y) = \delta(x + d, y + d) \sum_{x} \sum_{y} f(x, y) = i, f(x + d, y + d) = j
\]

(10)
where pixel coordinates are \((x, y)\) \(x, y = 0, 1, 2, \ldots, n - 1\), GLCM coordinates are \((i, j)\) \(i, j = 0, 1, 2, \ldots, n - 1\), and \(d\) represents step length. Hence, GLCM need to be normalised as

\[
\hat{p}_{d, \theta}(i, j) = \frac{p_{d, \theta}(i, j)}{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p_{d, \theta}(i, j)}
\]

where the second-order joint probability \(\hat{p}_{d, \theta}(i, j) \in [0, 1]\) of two pixels is separated by a distance of \(d\) along direction \(\theta\) with grey level \(i\) and \(j\) \((0 \leq i, j \leq n)\), respectively.

Haralick et al. [36] defined 14 kinds of feature parameter values of GLCM to represent the texture characteristic. Ulaby et al. [37] found that only four characteristic values were not related, which could be easily calculated and achieve a higher classification accuracy of texture complexity. In order to more accurately describe the characteristic of CU, angular second moment (ASM), contrast (CON) and correlation (COR) are applied in this paper. ASM is the sum of the squares of each the element value of GLCM, which is used to measure the stability of the grayscale distribution of the image and reflect the uniformity of the grayscale distribution of the image. CON is used to measure intensity contrast between a pixel and its neighbour over the whole image. COR measures the similarity of the grey image value in the row or column direction. Formally, they are defined as

\[
\text{CON} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i - \mu_x)(j - \mu_y)\hat{p}_{d, \theta}(i, j)
\]

\[
\text{ASM} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \hat{p}_{d, \theta}(i, j)^2
\]

\[
\text{COR} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \frac{(i - \mu_x)(j - \mu_y)\hat{p}_{d, \theta}(i, j)}{\sigma_x \sigma_y}
\]

\[
\mu_x = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} i\hat{p}_{d, \theta}(i, j)
\]

\[
\mu_y = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} j\hat{p}_{d, \theta}(i, j)
\]

\[
\sigma_x = \sqrt{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i - \mu_x)^2 \hat{p}_{d, \theta}(i, j)}
\]

\[
\sigma_y = \sqrt{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (j - \mu_y)^2 \hat{p}_{d, \theta}(i, j)}
\]

Since the maximum grey level \(n\) of the depth map reaches 255, it is very time-consuming for computing GLCM.

Therefore, to reduce the computational burden, the grey-level of the depth image is compressed to 16 levels and step length \(d\) is set to 2 in this paper. However, it would bring some distortion on the weak edges. The texture feature of the depth map can be represented by combining GLCM and Sobel. In Sobel operator [38], \(G_x\) and \(G_y\) represent the approximate horizontal and vertical gradients of each pixel. The approximate gradient value of each pixel is calculated by \(|G| = |G_x| + |G_y|\). If \(|G|\) is greater than a threshold (it is set to 5 empirically), the pixel is considered as an edge point.

For the homogenous CUs, the pixels usually have similar values along four directions \((0^\circ, 45^\circ, 90^\circ, 135^\circ)\). GLCM feature vector (GFV) of CU or PU is defined by (19). As depicted in Figure 6, for smooth region, GFV\(_{90}\) is always equal to \((0, 0, 0)\) for different \(\theta\) \((\theta=0^\circ, 45^\circ, 90^\circ, 135^\circ)\). Hence, in order to correctly characterise the texture features of the depth map, a new texture complexity model is built by using GLCM and Sobel operator as is shown, for example, in (23). CU’s texture feature is divided into two categories: smooth, edge or complex. The homogeneous regions are suitable for coding with large CU size, and small CU size is used for coding the edge or complex regions. If the condition \(E_3\) is satisfied \((T^*\) is set to 5 empirically), current CU is identified as the flat block and CU splitting is early terminated.

\[
\text{GFV}_{90} = \text{CON, ASM, COR}
\]

\[
\overline{\text{GFV}} = \frac{\text{GFV}_{0^\circ} + \text{GFV}_{45^\circ} + \text{GFV}_{90^\circ} + \text{GFV}_{135^\circ}}{4}
\]

\[
E_2 = \overline{\text{GFV}} = (0, 1, 0) \& \& N_{\text{edgepoint}} < T^*
\]

\[
E_3 = \overline{\text{GFV}} \neq (0, 1, 0) \& \& N_{\text{edgepoint}} > T^*
\]

\[
CU \text{ texture} = \begin{cases} 
\text{smooth, when } E_2 \text{ is satisfied.} \\
\text{edge or complex, when } E_3 \text{ is satisfied.}
\end{cases}
\]
3.3 Fast candidate mode decision

As described in Section 2, through the selection of the prediction modes in rough mode decision process, most probable modes process and DMMs process, the candidate mode list containing 3–12 candidate modes are finally generated. In the depth intra coding, Planar mode or DC mode is suitable for predicting large areas with smooth texture. DMMs are specifically used to encode sharp edges. Angular mode is used to predict some complex texture regions. Guo et al. [20] utilised the ASM of the GLCM to identify smooth CUs; DMMs are skipped for these smooth CUs. The directional features of the CU were measured by the COR of the GLCM to prune angular modes. However, Guo’s [20] did not achieve a significant complexity reduction of the depth intra coding. The main reason is that the texture features of the depth map are not fully mined, which is composed of large smooth areas divided by very few sharp edges, but only a few of the edge regions have directional texture. In this paper, we are more concerned about identifying most of the smooth areas and how to quickly determine the optimal prediction mode of the PUs in the smooth areas. Due to the stronger spatial correlation of the depth map, the pixels of neighbouring CUs are very similar, and the same prediction mode can be applied to neighbouring regions. Neighbouring PUs are defined as left PU and above PU of the current PU in this paper. Combining the texture features of the PU with strong spatial correlation, the candidate mode list can be determined in advance, without requiring complicated the RDcost calculation.

Based on the above analysis, a fast candidate mode decision (FCMD) algorithm for depth intra coding is explored. The condition $E_4$ is defined, for example, by (24) and the candidate mode list (CML) is determined, for example, by (25); the number of the mode candidates is reduced from 6 or 11 to only 3. Moreover, the most probable modes process is skipped.

$$E_4 = \neg GFV = (0, 1, 0) \& \& M_{left PU} \leq 1 \& \& M_{top PU} \leq 1 \quad (24)$$

$$CML = \begin{cases} -0, 1, 26, & \text{when } E_4 \text{ is satisfied} \\ \text{generated by traditional method, otherwise} \end{cases} \quad (25)$$

Due to the DMMs is designed to represent sharp edges, it has no effect on the encoding of flat blocks. Depth map contains a very small part of the edge area, only edge or complex blocks need to evaluate DMMs. If one of neighbouring PUs has selected DMMs or the current PU contains edge points, DMMs are tested, and most unnecessary the RDcost evaluations for DMMs are avoided. Hence, the condition $E_5$ is defined by e.g. (26) ($T$ is set to 0 empirically in $E_5$).

$$E_5 = \neg E_3 \| M_{left PU} \text{ or } M_{top PU} \in (DMM1, DMM4) \quad (26)$$

Tests have been conducted to verify the reliability of the conditions $E_4$ and $E_5$ by following the common test conditions (CTC) [39]. Statistical data are extracted from PU mode decision process as shown in Figures 7 and 8. It can be found from Figure 7, in the case of $E_4$, $P_t$ represents the percentage of the counts that $E_4$ is satisfied in PU mode decision process, $P_t$ appears from 72% to 94%. $P(E_4)$ means that the optimal prediction mode belonged to {0, 1, 26} when $E_4$ is satisfied, and it reaches more than 99%. As depicted in Figure 8, in case of $E_5$, $P(E_5)$ reaches about 82–94% that the optimal prediction mode belongs to DMMs when $E_5$ is satisfied, $P_d$ represents the percentage of that DMMs traversal counts are reduced in PU mode decision process, it appears from 28% to 72%. Taking all the analyses above into count, an efficient fast candidate mode decision algorithm can be developed by utilising the texture feature and spatial correlation.

3.4 The proposed overall algorithm

The flowchart of the proposed maximum splitting depth layer decision, early termination of CU splitting algorithm and the modified depth intra coding process are illustrated in Figure 9, whose placement within the overall scheme is indicated by gray shading.

4 EXPERIMENTAL RESULTS

In order to test the effectiveness of the proposed algorithm, eight test sequences with two resolutions (1024×768/1920×1088) are evaluated under common test
The three views of each test sequence are sequentially encoded, the texture video of each view is encoded first, and then the corresponding depth map video is encoded. To evaluate encoding performance of the depth map, from \( V_{0.25} \) to \( V_{1.75} \), six intermediate views are synthesised using the decoded texture and decoded depth videos at the decoder side, and the coding efficiency is evaluated by Bjontegaard Delta bitrate (BD-rate) [40] of synthesised views. BDBR represents the bitrate savings of the two methods under the same objective quality. \( \Delta T (\%) \) denotes the percentage of the coding time reduction compared with the author algorithm, which is defined by

\[
\Delta T = \frac{T_{fast} - T_{HTM-16.0}}{T_{HTM-16.0}}
\]  

(27)

### 4.1 Performance of the individual algorithm

As shown in Table 5, experiment results of the maximum splitting depth layer decision (MSDLD) algorithm based on encoding information of previously I frame compared with Fu's

**TABLE 5** Results of state-of-the-art algorithm and the proposed MSDLD algorithm

| Test sequence  | BDBR(%) | \( \Delta T (\%) \) | BDBR(%) | \( \Delta T (\%) \) |
|----------------|---------|---------------------|---------|---------------------|
| Balloons       | 0.06    | -19.7               | 0.08    | -19.3               |
| Kendo          | 0.04    | -25.5               | 0.12    | -25.4               |
| Newspaper      | 0.08    | -15.4               | 0.07    | -15.7               |
| GT_Fly         | 0.01    | -42.7               | 0.04    | -44.0               |
| Poznan_Hall2   | 0.25    | -48.4               | 0.15    | -42.4               |
| Poznan_Street  | 0.02    | -26.5               | 0.05    | -34.0               |
| Undo_Dancer    | 0.15    | -36.0               | 0.05    | -32.2               |
| Shark          | -0.02   | -33.9               | 0.05    | -28.8               |
| AVG            | 0.07    | -31.0               | 0.10    | -30.3               |

Restrict-PRO [13] is given. Fu et al. [13] proposed early termination of CU splitting method based on the consistency of depth intra skip mode between different depth layers. According to Table 5, on average, Fu’s Restrict-PRO achieves a 31.0% encoding time reduction and a 0.07% BD-rate increase. The proposed MSDLD algorithm achieves comparable performance compared to Fu’s Restrict-PRO, and 30.3% encoding time has been reduced, while bringing 0.10% increase in BDBR. As far as we know, this is the first method that the correlation between two consecutive I-frames is used to decide the \( D_{\max} \) for the subsequent I-frame in the recursive partitioning process. It is clear from this result, the MSDLD algorithm can predict the \( D_{\max} \) of the current CTU and skip unnecessary CU size efficiently.
The experiment results of ETCS algorithm and FCMD algorithm compared with Sanchez’s [22] are shown in Table 6. Sanchez et al. [22] utilised the information of neighbouring PUs and the border of current PU to speed up DMM1 and reduced the size of candidate mode list based on experimental statistical analysis. Both the proposed FCMD algorithm and Sanchez’s [22] reduce the length of the candidate pattern list. On average, 37.4% the encoding time has been reduced, bringing 0.27% BD-rate loss. It can be seen clearly that the proposed ETCS and FCMD algorithms also reduce the encoding time greatly and bring about negligible loss of coding performance compared with Sanchez’s [22]. On average, about 37.6% encoding time has been reduced with 0.31% BD-rate loss.

### 4.2 Detail performance analysis of the proposed overall algorithm

The experiment result of the proposed overall algorithm comparing with state-of-the-art algorithms is shown in Table 7. On average, 53.8% encoding time has been reduced, resulting in 0.43% BD-rate loss on synthesised views. The largest BDBR increase only reaches 0.81%, it shows that the proposed overall method is more efficient in encoding all sequences. In terms of time saving, the smallest time saving is 37.3% for “Newspaper,” the largest time saving is 68.8% for “Poznan_Hall2” and the time reduction of most sequences is concentrated between 40% and 65%. It reveals that our proposed method can achieve unobvious fluctuations in coding performance and time reduction. Zhang’s [25] reduced 39.3% encoding time on average, bringing 0.53% BD-rate increase. Fu’s [13] integrates the Restrict PRO method into Zhang’s [25], further reducing the complexity of CU partitioning. Fu’s [13] achieved 50.1% encoding time saving with 0.64% BD-rate loss. Compared with the [13, 22, 25], the proposed overall algorithm has reduced large the encoding time, while maintaining the same visual quality as the encoder. Therefore, it can be observed that the proposed algorithm achieves better coding performance than the other three algorithms in terms of the coding efficiency and the encoding complexity.

### 5 CONCLUSION

To cut down coding complexity of depth maps, a fast depth intra coding algorithm based on texture feature and spatio-temporal correlation was proposed. According to the $D_{max}$ and RDcost information of the previously encoded I frame in the same view, the $D_{max}$ of the current CTU is determined effectively. Then, a new texture complexity model was proposed by utilising GLCM and Sobel edge detection. Based on the proposed texture complexity model, the CU block was divided into two categories: smooth block, texture or edge block. On CU level, the ETCS algorithm for smooth CU was proposed to filter out unnecessary coding blocks. On PU level, taking PU’s types and the prediction mode of neighbour PUs into consideration, the FCMD algorithm was developed to cut short the redundant candidate modes. Experimental results shows that the proposed overall algorithm achieves 53.8% encoding time saving
ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 61871437 and in part by the Natural Science Foundation of Hubei Province of China under Grant 2019CFA022.

REFERENCES

1. Harmske[s, M.M., et al.: Multiview-video-plus-depth coding based on the advanced video coding standard. IEEE Trans. Image Process. 22(9), 3449–3458 (2013)
2. Domazik[i, M., et al.: High efficiency 3D video coding using new tools based on view synthesis. IEEE Trans. Image Process. 22(9), 3517–3527 (2013)
3. Tech, G., et al.: Overview of the multiview and 3D extensions of high efficiency video coding. IEEE Trans. Circuits Syst. Video Technol. 26(1), 35–49 (2016)
4. Ying, C., et al.: Test model 11 of 3D-HEVC and MV-HEVC. ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc JCT3V-K1003, Geneva, Switzerland, February 2015
5. Müller, K., et al.: 3D high-efficiency video coding for multi-view video and depth data. IEEE Trans. Image Process. 22(9), 3366–3378 (2013)
6. Kauff, P., et al.: Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability. Signal Process., Image Commun. 22(2), 217–234 (2007)
7. Yuan, Y., et al.: Object shape approximation & contour adaptive depth image coding for virtual view synthesis. IEEE Trans. Circuits Syst. Video Technol. PP(99), 1–1 (2018)
8. Sullivan, G.J., et al.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. 22(12), 1649–1668 (2012)
9. Muller, K.R., et al.: 3D video coding with depth modeling modes and view synthesis optimization. In: Proceedings of the IEEE International Conference APSIPA ASC, pp. 1–4. IEEE, Piscataway, NJ (2012)
10. Gu, Z., et al.: Fast segment-wide DC coding for 3D video compression. In: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2780–2783. IEEE, Piscataway, NJ (2015)
11. Tech, G., et al.: 3D video coding using the synthesized view distortion change. In: 2012 Picture Coding Symposium (PCS), 2012, pp. 25–28. IEEE, Piscataway, NJ (2012)
12. Chen, H., et al.: Early intra block partition decision for depth maps in 3D-HEVC. In: 2018 25th Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 1777–1781. IEEE, Piscataway, NJ (2018)
13. Fu, C.H., et al.: Early termination for fast intra mode decision in depth map coding using DIS-inheritance. Signal Process., Image Commun. 80, 115644 (2020)
14. Saldanha, M., et al.: Fast 3D-HEVC depth map encoding using machine learning. IEEE Trans. Circuits Syst. Video Technol. 30, 850–861 (2019)
15. Chen, Y., et al.: Fast CU size decision based on AQ-CNN for depth intra coding in 3D-HEVC. In: 2019 Data Compression Conference (DCC), pp. 561–561. IEEE, Piscataway, NJ (2019)
16. Shen, L., et al.: Efficient intra mode selection for depth-map coding utilizing spatiotemporal, inter-component and inter-view correlations in 3D-HEVC. IEEE Trans. Image Process. 27(9), 4195–4206 (2018)
17. Hamout, H., Elyounfi, A.: Fast depth map intra coding based structure tensor data analysis. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 1802–1806. IEEE, Piscataway, NJ (2018)
18. Zhang, H.B., et al.: Efficient depth intra mode decision by reference pixels classification in 3D-HEVC. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 961–965. IEEE, Piscataway, NJ (2015)
19. Lei, J., et al.: A novel depth intra mode decision algorithm for 3D-HEVC. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 1126–1130. IEEE, Piscataway, NJ (2015)
20. Guo, L., et al.: Simplified depth intra coding for 3D-HEVC based on gray-level co-occurrence matrix. In: 2016 IEEE International Conference on Signal and Image Processing (ICSIP), pp. 328–332. IEEE, Piscataway, NJ (2016)
21. Zhao, J., et al.: An efficient depth modeling mode decision algorithm for 3D-HEVC depth map coding. Optik-Int. J. Light. Electron Opt. 127(24), 12048–12055 (2016)
22. Sanchez, G., et al.: A reduced computational effort mode-level scheme for 3D-HEVC depth maps intra-frame prediction. J. Vis. Commun. Image Represent. 54, 193–203 (2018)
23. Sanchez, G., et al.: Complexity reduction by modes reduction in rd-list for intra-frame prediction in 3D-HEVC depth maps. In: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–4. IEEE, Piscataway, NJ (2017)
24. Park, C.S.: Edge-based intramode selection for depth-map coding in 3D-HEVC. IEEE Trans. Image Process. 24(1), 155–162 (2015)
25. Zhang, H.B., et al.: Probability-based depth intra mode skipping strategy and novel VSO metric for DMM decision in 3D-HEVC. IEEE Trans. Circuits Syst. Video Technol. (99), 1–1 (2016)
26. Li, T., et al.: Fast depth intra coding based on spatial correlation and rate distortion cost in 3D-HEVC. Signal Process., Image Commun. 80, 115668 (2020)
27. Peng, K.K., et al.: Low complexity depth intra coding combining fast intra mode and fast cu size decision in 3D-HEVC. In: Proceedings of the IEEE International Conference Image Processing (ICIP), pp. 1126–1130. IEEE, Piscataway, NJ (2016)
28. Jing, R., et al.: CART-based fast cu size decision and mode decision algorithm for 3D-HEVC. Signal Image Video Process. 13(2), 209–216 (2019)
29. Li, T., et al.: Simplified depth intra coding based on texture feature and spatial correlation in 3D-HEVC. In: 2018 Data Compression Conference, Snowbird, UT, 2018, pp. 421–421
30. Mora, E.G., et al.: Initialization, limitation, and predictive coding of the depth and texture quadtree in 3D-HEVC. IEEE Trans. Circuits Syst. Video Technol. 24(9), 1554–1565 (2014)
31. Zhang, T., et al.: Fast intra-mode and CU size decision for HEVC. IEEE Trans. Circuits Syst. Video Technol. 27(8), 1714–1726 (2017)
32. Lei, J., et al.: Fast mode decision based on grayscale similarity and inter-view correlation for depth map coding in 3D-hevc. IEEE Trans. Circuits Syst. Video Technol. PP(99), 1–1 (2018)
33. Jung, S.H., Park, H.W.: A fast mode decision method in HEVC using adaptive ordering of modes. IEEE Trans. Circuits Syst. Video Technol. 26(10), 1846–1858 (2016)
34. Shen, Y., et al.: Image texture based fast CU size selection algorithm for HEVC intra coding. In: 2014 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pp. 363–367. IEEE, Piscataway, NJ (2014)
35. Baraldi, A., Parmiggiani, F.: Investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters. IEEE Trans. Geosci. Remote Sens. 33(2), 293–304 (1995)
36. Haralick, R.M., et al.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. 3(6), 610–621 (1973)
37. Ulaby, F.T., et al.: Textural information in SAR images. IEEE Trans. Geosci. Remote Sens. 24(2), 235–245 (1986)

38. Kanopoulos, N., et al.: Design of an image edge detection filter using the sobel operator. IEEE J. Solid-State Circuits 23(2), 358–367 (2002)

39. Müller, K., Vetro, A.: Common test conditions of 3DV core experiments. ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. JCT3V-G1100, January 2014

40. Bjontegaard, G.: Calculation of average PSNR differences between RD-curves. ITU-T Q.6/SG16 VCEG, Doc. VCEG-M33, 13th Meeting, Austin, TX, April 2001

How to cite this article: Li T, Yu L, Wang H, Chen Y. Fast depth intra coding based on texture feature and spatio-temporal correlation in 3D-HEVC. IET Image Process. 2021;15:206–217. https://doi.org/10.1049/ipr2.12021