

Research Article

Low-Complexity Texture Video Coding Based on Motion Homogeneity for 3D-HEVC

Qiuwen Zhang , Shuaichao Wei, and Rijian Su 

College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

Correspondence should be addressed to Rijian Su; rijiansu@126.com

Received 16 July 2018; Revised 29 November 2018; Accepted 17 December 2018; Published 15 January 2019

Academic Editor: Basilio B. Fraguela

Copyright © 2019 Qiuwen Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Three-dimensional extension of the high efficiency video coding (3D-HEVC) is an emerging international video compression standard for multiview video system applications. Similar to HEVC, a computationally expensive mode decision is performed using all depth levels and prediction modes to select the least rate-distortion (RD) cost for each coding unit (CU). In addition, new tools and intercomponent prediction techniques have been introduced to 3D-HEVC for improving the compression efficiency of the multiview texture videos. These techniques, despite achieving the highest texture video coding efficiency, involve extremely high-complex procedures, thus limiting 3D-HEVC encoders in practical applications. In this paper, a fast texture video coding method based on motion homogeneity is proposed to reduce 3D-HEVC computational complexity. Because the multiview texture videos instantly represent the same scene at the same time (considering that the optimal CU depth level and prediction modes are highly multiview content dependent), it is not efficient to use all depth levels and prediction modes in 3D-HEVC. The motion homogeneity model of a CU is first studied according to the motion vectors and prediction modes from the corresponding CUs. Based on this model, we present three efficient texture video coding approaches, such as the fast depth level range determination, early SKIP/Merge mode decision, and adaptive motion search range adjustment. Experimental results demonstrate that the proposed overall method can save 56.6% encoding time with only trivial coding efficiency degradation.

1. Introduction

With the advancement of three-dimensional televisions (3DTV) and free viewpoint TV (FTV) technology, 3D video has received much attention in the consumer electronics market. Multiview video (MVV) has been developed for 3D video applications. MVV of the same scene is captured via multiple synchronized cameras, offering a better experience than previous 2D videos because users can choose different viewing angles interactively. However, an MVV system requires a huge amount of information associated with the multiview content and a very high transmission rate, which result in high computational complexity of the transmission coding. To address this issue, multiview video coding (MVC) is developed by the VCEG and MPEG [1]. MVC enables inter-view prediction to improve MVV system compression capability, as well as supporting ordinary spatial and

temporal prediction. By doing this, a reduction in MVV bitrate of the MVC can be achieved without loss of compression quality.

The recent high efficiency video coding (HEVC) achieves a higher rate-distortion (RD) efficiency than H.264/AVC (50% bitrate savings for equivalent perceptual video quality) by utilizing new interprediction and intraprediction modes [2]. Following this, the next generation 3D video coding based on HEVC (3D-HEVC) standard has more advanced compression capability and supports for synthesis of additional perspective views [3]. In 3D-HEVC, when coding the dependent texture video views and the depth maps, modified HEVC coders which include new tools and techniques that make use of coded data inside the same access unit are used [4]. For texture coding, disparity-compensated prediction, inter-view motion prediction, advanced residual prediction, illumination compensation,

and view synthesis prediction (VSP) are presented to code-dependent texture video [5]. Similar to HEVC, a computationally expensive mode decision in 3D-HEVC is executed to select the one with the least rate distortion cost from all potential prediction modes by employing Lagrange multiplier. Therefore, it is very desirable to develop a fast texture video coding algorithm that can reduce the complexity of 3D-HEVC with only a tiny loss of 3D video quality.

The rest of the paper is structured as follows: Section 2 reviews the related work and Section 3 presents the motion homogeneity analysis. The proposed low-complexity texture video method is detailed in Section 4. Simulation results and conclusions are provided in Sections 5 and 6, respectively.

2. Related Work

In this part, the many prior efforts on MVC encoder accelerations are mentioned. These papers can be divided into methods with mode reduction and methods with fast disparity estimation (DE)/motion estimation (ME).

Category 1: Mode reduction algorithms are proposed in literatures [6–9]. A content-aware prediction algorithm is designed in [6] to alleviate the coding burden of MVC. Because video contents are very relevant, unnecessary coding computation will be reduced. A fast intermode decision is presented in [7] to alleviate encoding burden by employing textural segmentation and correlations. However, the characteristics of the intermode selection precedence in 3D-HEVC are unlike those in MVC. An early termination model is proposed in [8] for MVC fast mode decision, which can adaptively adjust the RD cost threshold based on the video content and motion properties. It can reduce 79.57% to 89.21% computation burdens with tiny coding degradation. An algorithm based on optimal stop theory is studied in [9] to realize a good balance between accelerating time and decision quantity in MVC. Considering both predicted mode probability and estimated coding time of all coding modes, a hybrid model is employed relying on the multiview coding mode characteristics. To reduce the computational complexity of multiview depth map coding, a fast mode decision algorithm based on intercomponent and inter-view correlations is introduced in [10]. However, these mode reduction methods cannot be used in the 3D-HEVC mode decision, in which the number of prediction modes for each prediction unit (PU) is increased in 3D-HEVC texture video coding (for example, intramodes have been raised to 35 from 9 in MVC). Moreover, the key factors in these methods of MVC are not considered and cannot work well when encountering the increased number of PU prediction in 3D-HEVC.

Category 2: Fast DE and ME methods are in literatures [11–14]. A view-adaptive algorithm for DE and ME is introduced in [11], alleviating the computational burdens of MVC. In [12], a reduplicative search for motion and parallax estimation algorithm is presented to obtain motion and parallax vectors simultaneously by utilizing the stereo-motion consistency constraint. There exists an algorithm in [13], cutting down candidates for MVC using two methods:

fast ME/DE and the fast mode decision. Adaptive DE and mode size decision in [14] employ a depth map to alleviate the coding burden of multiview texture video. However, those algorithms cannot accelerate the coding process efficiently for the new HEVC-based 3D video.

Recently, more traditional algorithms to reduce computational complexities of mode decision [15–21] have been introduced for the HEVC encoder to speed up selecting dominant candidate prediction modes. Fast intramode decision method [15] employs brink information of the current PU to pick out candidate prediction directions. This method calculates the current PU pixels and defines a lessened set of intramodes to be tested predominant based on this orientation. A fast CU size and mode decision method investigated in [16] alleviates the computational burdens. By fully exploiting the RD cost and prediction mode connections in varying depth levels or spatially nearby CUs, this method can cut down some modes. A fast mode selection algorithm is proposed in [17] based on linear programming to allocate computational complexities among the frames, which can determine complexity factors for all frames and CUs. A novel zero block (ZB) detection algorithm presented in [18] explores fake ZBs while trying to alleviate the augmentations of the computational burdens for HEVC. A fast transform unit (TU) size decision method proposed in [19] cuts down the candidate transform sizes of HEVC. It reports 30%–46% computational complexity reduction with tiny degradation. An adaptive fractional-pixel ME-skipped scheme has been proposed in [20] for low-complexity HEVC, which is based on the characteristics of the variable-size PUs and the video content partition relationship among variable-size PUs. A fast CU size decision algorithm based on temporal and spatial correlation is proposed in our previous work [21] to reduce the computation complexity of the HEVC encoder. The aforementioned fast mode decision methods play an irreplaceable part in HEVC encoders accelerating coding process with tiny loss of quality degradation. However, there is still some room for further improvement in the mode decision process of 3D-HEVC because the prediction structures which involve inter-view motion prediction and disparity-compensated prediction are different from that of HEVC. They can be further adopted for complexity reduction and complex mode decisions.

To this end, a few approaches are published for 3D-HEVC encoders to accelerate texture video coding. The adaptive search range adjustment and early termination mode search are presented in [22] to reduce the 3D-HEVC encoder complexity by using the correlations between the dependent view and the base view. A fast mode decision method is proposed in [23] based on a Bayesian classifier to predict the current CU mode by utilizing the information of already encoded neighboring CUs. An algorithm has been designed in [24], which aims at lessening the memory access without hampering coding efficiency. It reports up to 21% texture video encoding time reduction with 0.1% bitrate loss. In [25], a fast texture video coding scheme is presented to cut down the encoding time of 3D-HEVC. A fast algorithm developed in [26] is based on the early SKIP mode detection

and the prediction size correlation-based mode decision to reduce 3D-HEVC encoding time for real-time applications, by jointly exploiting correlations among multiview texture video and depth map and the interlevel correlation in quadtree structure of HEVC. An early merge mode decision method is presented in [27], utilizing inter-view correlations to accelerate the dependent texture view coding process. It reports an average 47.1% complexity reduction with 0.1% bitrate loss for texture video coding. An efficient online learning-based complexity reduction scheme is employed in [28] for the coding process of the dependent texture views in 3D-HEVC encoders. In addition, this method can adaptively coordinate the motion search range and alleviate the burden of prediction mode search. However, this method needs to confirm threshold via training some texture video data online, emphasizing that threshold accuracy is important. A fast mode decision is proposed in [29] to reduce candidate modes of 3D-HEVC, where the inter-view and grayscale similarity correlation are jointly used. A motion and disparity vectors early determination algorithm is proposed in [30], which exploits the spatial and temporal motion vector of neighboring treeblocks to reduce 3D-HEVC computational complexity. Up to 33.0% complexity reduction is reported with insignificant RD performance loss for texture video coding. Fast coding algorithms are also developed in our previous work [31, 32] based on the coding information from depth map-texture video and inter-view correlations to alleviate the computational burden of 3D-HEVC encoders. The aforementioned methods play a good role in multiple texture video and achieve significant encoding time reduction. However, they only employ texture video-depth map, inter-view, and spatial-temporal correlations. As shown in the aforementioned research, the property of the texture video motion homogeneity is not efficiently utilized. It shows that coding time can no more be shortened. More time of each step in process of texture video coding should be cut down by jointly studying the property of the texture video motion homogeneity.

For conquering the aforementioned limitation, we propose a fast texture video relying on motion homogeneity classification to reduce 3D-HEVC computational complexity. Considering that the multiview texture videos' motion characteristics are highly content dependent, it is not efficient to use a fixed motion search range and depth level range for the whole encoding process. Therefore, we can skip some specific motion search range and depth levels rarely used in 3D-HEVC mode decision. Extensive experimental results demonstrate that the proposed low-complexity texture video algorithm can significantly reduce the encoding time of 3D-HEVC with tiny loss of RD performance.

3. Motion Homogeneity Analysis

In natural 2D video sequences, there exists high connection among the neighboring treeblocks in spatial and temporal domains [26]. Furthermore, different from single-view video coding HEVC [33, 34], inter-view correlation is employed to reduce multiview texture video redundancy. The 3D-HEVC uses the variable-size prediction techniques of HEVC to exploit the spatial, temporal and inter-view correlation within successive frames. Thus, there exists a strong motion correlation between the neighboring treeblocks in spatial, temporal, and inter-view in multiview texture videos in 3D-HEVC. If the current treeblock and corresponding treeblocks (spatial, temporal, and inter-view correlation) belong to the same video object, then they have similar motion activities, and hence these can be defined as statistically homogeneous treeblocks.

On the basis of these observations, we propose to analyze the current treeblock motion vector using the motion information from the spatial-temporal and the previously coded view corresponding treeblocks. A set of motion predictors (Ω) is defined in equation (1) including three types of motion predictors:

$$\Omega = \{M_S, M_T, M_V\}, \quad (1)$$

where M_S denotes spatial motion predictors in the current texture view (including M_1, M_2, M_3 and M_4 in Figure 1), M_T denotes temporal motion predictors in the previously coded frame (including M_5, M_6, M_7, M_8 , and M_9) as the current texture treeblock M_S , and M_V denotes the inter-view motion predictor in the previously coded view (including $M_{10}, M_{11}, M_{12}, M_{13}$, and M_{14}) located at the same position as the M_S in Figure 1.

To verify motion information correlation of the spatial-temporal and the previously coded view corresponding treeblocks, extensive simulations have been conducted on a set of video test sequences with various motion activities and spatial resolutions. Among these test sequences, “Kendo,” “Balloons,” and “Newspaper” are in 1024×768 resolution, while “Undo_Dancer,” “GT_Fly,” “Poznan_Street,” “Poznan_Hall2,” and “Shark” are in 1920×1088 resolution. Experimental conditions are set as follows: I-B-P view structure; QPs are 25, 30, 35, and 40, respectively; treeblock size is 64; depth level range is from 0 to 3; search range of motion estimation (ME) is 64; the sum of test frames is one hundred. Tables 1–3 show the correlations of the current treeblock among spatial-temporal and inter-view neighboring treeblocks. The coding information (Motion Vector (MV)) correlation degree between the current treeblock and spatial-temporal and inter-view neighboring treeblocks is defined as follows:

$$\text{Corr} = \frac{\sum_{i=0}^{i=D} (\text{Treeblock}_{\text{cur}_i} - \text{Treeblock}_{\text{inter}_i} (\text{Treeblock}_{\text{spa}_i} / \text{Treeblock}_{\text{tem}_i}))}{D}, \quad (2)$$

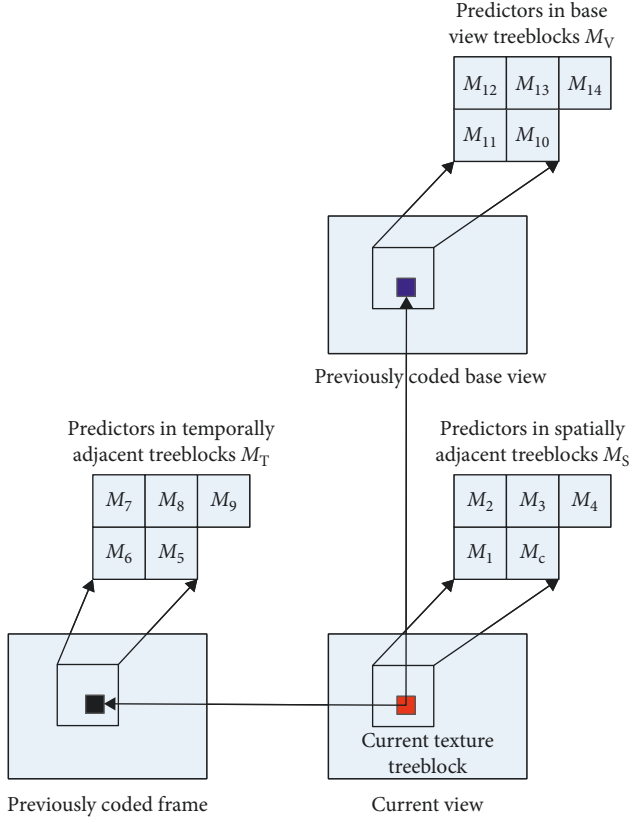


FIGURE 1: Predictors of the current treeblock and neighboring treeblocks.

TABLE 1: Motion information correlation of the current texture treeblock and its spatial treeblocks.

Index in Figure 1	M_1	M_2	M_3	M_4
Correlation	0.783	0.731	0.801	0.713

TABLE 2: Motion information correlation of the current texture treeblock and its temporal treeblocks.

Index in Figure 1	M_5	M_6	M_7	M_8	M_9
Correlation	0.623	0.604	0.572	0.592	0.557

TABLE 3: Motion information correlation of the current texture treeblock and its inter-view treeblocks.

Index in Figure 1	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}
Correlation	0.712	0.692	0.668	0.682	0.649

where Corr represents the coding information correlation degree, which ranges from 0 to 1, D denotes the max depth value of current CU, i indicates the depth level of current treeblock, $\text{Treeblock}_{\text{cur}_i}$ denotes the current treeblock at the i th depth level, $\text{Treeblock}_{\text{tem}_i}$ denotes the temporal adjacent treeblocks of the current treeblock ($\text{Treeblock}_{\text{cur}_i}$) at the same view, $\text{Treeblock}_{\text{spa}_i}$ denotes the spatially neighboring treeblocks, and $\text{Treeblock}_{\text{inter}_i}$ denotes the

colocated treeblocks in the previously coded base view frames. The previously coded view adjacent treeblock $\text{Treeblock}_{\text{inter}_i}$ (in predictors M_V), the spatially adjacent treeblock $\text{Treeblock}_{\text{spa}_i}$ (in predictors M_S), and temporally adjacent treeblock $\text{Treeblock}_{\text{tem}_i}$ (in predictors M_T) are described as in Figure 1, whose information are independent. There exist strong correlations in current texture video treeblock and the spatial-temporal or the previously coded view corresponding treeblocks.

Based on the motion predictor Ω in equation (1), the motion information from the spatial-temporal and inter-view adjacent predictors is extracted to analyze ME characteristics of the current texture treeblock in 3D-HEVC. We build a novel criterion to identify the motion homogeneity degree of the current texture treeblock between the adjacent treeblocks. It is explained as follows: The motion vectors from 4×4 block level covered by the current texture treeblock and its corresponding treeblocks (all spatial-temporal and inter-view adjacent treeblocks: M_1, M_2, \dots, M_{14} as in Figure 1) are employed to represent the motion homogeneity. Assuming that a texture treeblock located at the r th row and w th column is expressed as TB_{rw} , the motion vectors of its covered 4×4 blocks are denoted as $\text{MV}_{ij} = (\text{MV}x_{ij}, \text{MV}y_{ij})$, $i \in [4r, 4r + 3]$, $j \in [4w, 4w + 3]$. The motion homogeneities of the current texture treeblock in x and y directions are described separately:

$$\text{MH}x = \frac{1}{N} \sum_{(i,j) \in \Omega} \left| \text{MV}x_{ij} - \alpha_{ij} \cdot \sum_{(i,j) \in \Omega} \text{MV}x_{ij} \right|, \quad (3)$$

$$\text{MH}y = \frac{1}{N} \sum_{(i,j) \in \Omega} \left| \text{MV}y_{ij} - \alpha_{ij} \cdot \sum_{(i,j) \in \Omega} \text{MV}y_{ij} \right|, \quad (4)$$

where $\text{MH}x$ and $\text{MH}y$ are the motion homogeneities of the x and y components and N is the total number (4×4 block) of blocks covered by the current texture treeblock and its adjacent treeblocks (all spatial-temporal and inter-view adjacent treeblocks $M_S, M_T, M_V: M_1, M_2, \dots, M_{14}$ as in Figure 1). For our experiments we have used $N = (14 + 1) \times 4 \times 4 = 240$. α_{ij} is the motion-weight factor. $\sum \alpha_{ij} = 1$. α_{ij} is ruled relying on the effect of adjacent treeblocks on the current texture treeblock. According to motion information connection (the current texture treeblock and its adjacent treeblocks M_S, M_T, M_V) in Tables 1–3, the motion-weight factor α_{ij} is summarized in Table 4. These values α_{ij} are set based on a wide range of experiments, and these values achieve good coding results on a variety of 3D sequences with different motion properties. Equations (3) and (4) are not only used to calculate the correlation between the current treeblock and the reference treeblocks. In fact, the motion correlation between the reference treeblock and the current treeblock is also calculated in the formula. Thus, the motion homogeneity category parameter is as follows:

$$\text{MH} = \text{MH}x + \text{MH}y. \quad (5)$$

Relying on motion homogeneity parameter MH , each treeblock is classified into two styles, statistically

TABLE 4: The motion-weight factors in predictor Ω .

Index in Figure 1	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}
α_{ij}	0.10	0.10	0.10	0.10	0.05	0.05	0.05	0.05	0.05	0.07	0.07	0.07	0.07	0.07

homogeneous treeblocks and complex motion treeblocks. If T_c is the current texture treeblock, the criterion is defined as follows:

$$T_c \subset \begin{cases} \text{Statistically homogeneous region,} & \text{if } MH < Thr, \\ \text{Complex motion region,} & \text{otherwise,} \end{cases} \quad (6)$$

where Thr is the threshold factor. Thr is employed to identify whether a treeblock is the part with statistically homogeneous or complex motion. When the motion homogeneity parameter MH is less than Thr , the current texture treeblock is considered in a statistically homogeneous region; otherwise, the current texture treeblock is with complex motion region. Thus, the probability of static and slow motion is higher than that of complex motion in real-world 3D video. Hence, we focus on static and slow motion separately to save as much encoding time as possible. On the basis of the above discussion, we use the average motion homogeneity of each treeblock to determine threshold Thr . The average motion homogeneity of each treeblock MH_{average} is calculated as follows:

$$MH_{\text{average}} = \frac{1}{R \times C} \cdot \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} (MHx_{r,c} + MHy_{r,c}), \quad (7)$$

where R and C represent the total number of treeblocks in a row and in a column, respectively. Based on extensive experiments, Thr is set to $0.7 \cdot MH_{\text{average}}$, which achieves a good and consistent performance on a variety of test sequences with different texture characteristics and motion activities and fixed for each treeblock QP level in 3D-HEVC encoders.

4. Proposed Low-Complexity Texture Video Coding Algorithm

4.1. Fast Depth Level Range Determination. 3D-HEVC usually allows a maximum CU size equal to 64, and the depth level range is from 0 to 3. The CU depth has a fixed range for a whole video sequence in the 3D-HEVC reference software. In fact, the depth value of “0” occurs very frequently for static or homogeneous region coding. On the other hand, the depth value of “0” is rarely chosen for treeblocks with complex motion [26]. These results show that depth level range should be adaptively determined based on the motion homogeneity property of texture video treeblocks.

To achieve great savings in 3D-HEVC encoding time while minimizing the loss in coding efficiency, the depth level of a texture video treeblock having limited contribution to coding efficiency should be skipped. Hence, we use motion homogeneity property to filter out the unsuitable depth level to speed up the 3D-HEVC encoding process.

According to the motion homogeneity of the current texture video treeblock (based on equation (6)), each treeblock can be classified into two types, statistically homogeneous treeblocks and complex motion treeblocks.

Table 5 shows the depth level distribution for two types of treeblocks, where “Level 0,” “Level 1,” “Level 2,” and “Level 3” are the depth levels of the texture video treeblock. It can be seen that for treeblocks with statistically homogeneous region, about 63.6% of texture video treeblocks choose the optimal depth level with “0” and about 33.3% treeblocks choose the optimal depth value with “1.” In other words, if the maximum depth level is set to be “1,” it will most likely cover about 96.9% of the texture video treeblocks. The mode prediction on depth levels of “2” and “3” (CU sizes 16×16 and 8×8) can be skipped. For treeblocks with motion region, about 98.4% of treeblocks choose depth levels with “1,” “2,” and “3” (CU sizes 32×32 , 16×16 and 8×8). If the minimum depth level is set to be “1” and the maximum depth level is set to be “3,” it will be the most likely cover about 98.4% of the texture video treeblocks. On the other side, the probability of choosing the depth level with “0” is very low, less than 1.6%, and thus mode prediction on depth level of “0” (CU size 64×64) can be skipped. Based on the above analysis, the candidate depth levels that will be tested using RD optimization (RDO) for each texture video treeblock are summarized in Table 6. With the proposed depth level range determination algorithm, most texture video treeblocks can skip one to two tested depth levels. As a result, the candidate depth levels are limited to a small subset, and the computational complexity of 3D-HEVC could be highly reduced.

4.2. Early SKIP/Merge Mode Decision. SKIP/Merge mode provides good coding performance and requires a lower computational complexity in 3D-HEVC encoder. Thus, once SKIP/Merge mode can be predecided, variable-size ME and DE computation for a treeblock can be entirely saved in a 3D-HEVC mode decision procedure. In fact, the decision to use SKIP/Merge mode is delayed until the RD costs of all other prediction modes have been determined, and it is found that SKIP mode costs less. Usually, after computing the RD costs of all prediction modes, many texture video treeblocks finally end up with being decided as SKIP mode in 3D-HEVC because they belong to homogeneous region or a motionless object [35, 36]. Based on this analysis, we propose a novel early SKIP/Merge mode decision algorithm for statistically homogeneous treeblock to avoid the whole variable-size ME and DE procedure.

To sustain the rationality of our early SKIP/Merge mode decision algorithm, we conducted simulation experiments, as shown in Table 7. By studying the thorough mode decision, we study the mode distribution for texture video treeblocks with statistically homogeneous region.

TABLE 5: Texture video depth level distribution for two types of treeblocks.

Sequences	Statistically homogeneous region				Complex motion region			
	Level 0 (%)	Level 1 (%)	Level 2 (%)	Level 3 (%)	Level 0 (%)	Level 1 (%)	Level 2 (%)	Level 3 (%)
Kendo	65.2	31.7	2.1	1.0	1.9	27.2	31.7	39.2
Balloons	67.1	29.5	2.6	0.8	1.4	28.7	32.3	37.6
Newspaper	59.4	35.7	3.2	1.7	2.2	33.9	35.7	28.2
Shark	56.7	40.6	1.8	0.9	0.6	22.7	28.6	48.1
Undo_Dancer	57.8	38.6	2.5	1.1	0.5	22.4	30.2	46.9
GT_Fly	64.1	33.2	1.9	0.8	1.5	29.2	32.5	36.8
Poznan_Street	66.2	31.3	1.9	0.6	1.7	25.6	34.5	38.2
Poznan_Hall2	72.5	26.1	1.2	0.2	3.3	38.4	34.2	24.1
Average	63.6	33.3	2.2	0.9	1.6	28.5	32.5	37.4

TABLE 6: Candidate depth levels of texture video coding for two types of treeblocks.

Texture video treeblock type	Candidate depth levels	Depth range [Depth _{min} , Depth _{max}]
Statistically homogeneous region	0, 1	[0, 1]
Complex motion region	1, 2, 3	[1, 2, 3]

TABLE 7: Prediction mode distribution for treeblocks with statistically homogeneous region.

Sequences	SKIP mode (%)	Merge mode (%)	Inter $2N \times 2N$ (%)	Intra $2N \times 2N$ (%)	Inter $2N \times N$ (%)	Inter $N \times 2N$ (%)	Intra $N \times N$ (%)	Small intermodes (%)
Kendo	94.7	3.5	0.6	0.3	0.2	0.2	0.1	0.4
Balloons	95.4	3.2	0.4	0.2	0.3	0.1	0.1	0.3
Newspaper	92.9	4.1	0.8	0.4	0.4	0.3	0.2	0.9
Shark	89.1	4.8	1.8	1.4	0.8	0.6	0.3	1.2
Undo_Dancer	88.9	5.2	1.5	1.1	0.8	0.8	0.4	1.3
GT_Fly	94.2	3.2	0.9	0.3	0.4	0.3	0.1	0.6
Poznan_Street	95.1	2.6	0.6	0.2	0.3	0.3	0.1	0.8
Poznan_Hall2	97.8	1.5	0.2	0.1	0.1	0.1	0.0	0.2
Average	93.5	3.5	0.9	0.5	0.4	0.3	0.2	0.7

Inter $N \times N$, inter $2N \times nU$, inter $2N \times nD$, inter $nL \times 2N$, and inter $nR \times 2N$ are the small intermodes.

The percentages of choosing SKIP/Merge mode are 93.5% and 3.5%. The percentages of choosing other mode sizes are lower than 0.9%. In a word, our early SKIP/Merge mode decision method can significantly reduce useless ME and DE on small CU size.

4.3. Adaptive Motion Search Range Adjustment. Motion search plays an important role in defining the computational complexity of 3D-HEVC encoder, which is defined as the search for the best matched treeblock in temporally preceding frames. A suitable motion search window can reduce the computational complexity of 3D-HEVC and also maintain the good RD performance in 3D-HEVC [37, 38]. In the joint model of 3D-HEVC, motion search has a large fixed range for a whole texture video coding. The large fixed motion search range achieves the highest texture video coding efficiency but requires a significant computational complexity. Because the optimal motion search range is highly dependent on treeblock motion homogeneity characteristic, it is not advisable to use a large fixed motion search window in 3D-HEVC. Because the predicted vector is not always accurate and sometimes totally wrong, a large search range window is necessary for reliable motion

estimation. On the other side, search range window could be reduced without loss of coding efficiency when the predicted vector is accurate. Because of the wide variation in motion fields across different frames in a 3D video sequence, it is necessary to adjust the search range window on a frame-by-frame basis for 3D-HEVC motion estimation. Furthermore, a 3D video frame may contain various regions with different levels of motion homogeneity, and the search range of ME needs to be adjusted on the motion complexity. Based on the above analysis, we can determine motion search range and skip some specific range levels rarely used in statistically homogeneous region.

It can be seen from Table 8 that for texture video treeblocks in static homogeneous region, more than 97.0% of all texture motion vectors lie in $[SR/4 \times SR/4]$ window. In other words, if the maximum search range is set to $SR/4$, it will most likely cover about 97.0% of all texture motion vectors. For the texture video treeblocks in complex motion region, the percentage of all texture motion vector lying in $[SR/16 \times SR/16]$, $[SR/4 \times SR/4]$, and $[SR/2 \times SR/2]$ windows are about 48.5%, 73.7%, and 82.7%, respectively, and thus 3D-HEVC motion search range cannot be reduced. From Table 8, the dynamically motion adjust search range that will be employed in 3D-HEVC texture coding is defined as follows:

TABLE 8: Motion search distributions for two types of treeblocks.

Sequences	Treeblocks in statistically homogeneous region			Treeblocks in complex motion region		
	SR1 (%)	SR2 (%)	SR3 (%)	SR1 (%)	SR2 (%)	SR3 (%)
Kendo	82.2	97.1	97.5	50.2	77.3	84.1
Balloons	80.4	96.3	97.2	48.3	74.8	82.7
Newspaper	83.7	97.8	98.3	51.7	79.2	86.1
Shark	79.6	95.2	96.5	35.8	58.6	72.4
Undo_Dancer	78.9	96.1	97.1	34.7	56.3	71.6
GT_Fly	84.1	97.5	98.2	54.2	79.8	87.2
Poznan_Street	82.4	96.6	97.3	52.7	78.6	85.3
Poznan_Hall2	88.6	99.2	99.4	60.2	84.9	91.8
Average	82.5	97.0	97.7	48.5	73.7	82.7

“SR1,” “SR2,” and “SR3” represent the windows of $[SR/16 \times SR/16]$, $[SR/4 \times SR/4]$, and $[SR/2 \times SR/2]$, respectively.

search range

$$= \begin{cases} SR/4 \times SR/4, & \text{treeblock} \in \text{static homogeneous region,} \\ SR \times SR, & \text{treeblock} \in \text{motion complex region.} \end{cases} \quad (8)$$

With the proposed motion search range adjustment method, most texture video treeblocks of 3D-HEVC are capable of omitting useless candidate range in statistically areas. Based on the above three methods, the depth level range is selectively enabled, the sum of candidate modes in texture video is cut down, and the search range of ME and DE is adaptively determined. Finally, the proposed low-complexity coding algorithm selects the best mode among all texture video candidate modes and the encoding time in 3D-HEVC encoders can be reduced dramatically.

4.4. Overall Algorithm. According to the aforementioned analysis, the basic idea of the proposed low-complexity texture video coding algorithm for 3D-HEVC is to adjust various steps of ME and DE based on motion homogeneity of texture motion field. The proposed overall algorithm incorporates fast depth level range determination, early SKIP/Merge mode decision, and adaptive motion search range adjustment based on motion homogeneity. The flowchart of the proposed overall algorithm is given in Figure 2. The overall algorithm is shown below.

Step 1: start mode decision for a texture video treeblock.

Step 2: obtain the coding information from predictors Ω in the spatial-temporal/the previously coded view (shown in Figure 1).

Step 3: compute motion homogeneity parameter MH based on (5) and Thr based on (6) and categorize the current texture video into statistically homogeneous part and complex motion region.

Step 4: execute fast depth level range determination. When the current texture video treeblock is a statistically homogeneous part, the optimal depth level is set to be “0” to “1”; when the current texture video treeblock is a complex motion region, the optimal depth level is set to be “1” to “3.”

Step 5: perform adaptive motion search range adjustment. When the current texture video treeblock is a statistically homogeneous region, the optimal search window is reconfigured with $[SR/4 \times SR/4]$.

Step 6: perform early SKIP/Merge mode decision. If the texture video treeblock in a statistically homogeneous region, only SKIP and Merge mode are used for the best mode and skip variable-size ME and DE and go to Step 7.

Step 7: determine the best mode. Go to step 1 and proceed with next treeblock.

5. Experimental Results

For evaluating the low-complexity texture video coding algorithm, we execute it on the 3D-HEVC test model (HTM 16.0 [4, 39]). We have tested eight sequences in two resolutions ($1024 \times 768/1920 \times 1088$) recommended by JCT-3V Group [40]. In these sequences, the “Shark” and “Undo_Dancer” possess a large global motion region or rich texture region; the “Kendo,” “Balloons,” “Newspaper,” “GT_Fly,” and “Poznan_Street” have a middle local motion or a glabrous texture region; and there exists a small global motion or homogeneous texture in “Poznan_Hall2.” Coding conditions are set as follows: 3-view case: center-left-right (in coding order); P-I-P interview prediction; texture video QP values for independent view: 40, 35, 30, and 25; depth map QP values: 45, 42, 39, and 34; the number of test coding frames in each sequence is 150. The “VSRS-1D-Fast” software is employed for the view synthesis. In this section, we compare the proposed algorithm in Tables 9–15 with the 3D-HEVC encoder using exhaustive mode decision and the state-of-the-art fast methods [25, 28], where coding efficiency is measured by texture video and rendered view peak signal-to-noise ratio (PSNR), and the computation complexity is measured by the consumed run time. The Bjontegaard Delta PSNR (BDPSNR) [41] represents the PSNR gain, Bitrate (BDBR) represents the improvement of total bitrates, and “Dtime (%)” represents the encoding time change in percentage:

$$Dtime = \frac{Time_{proposed} - Time_{original}}{Time_{original}} \times 100\%, \quad (9)$$

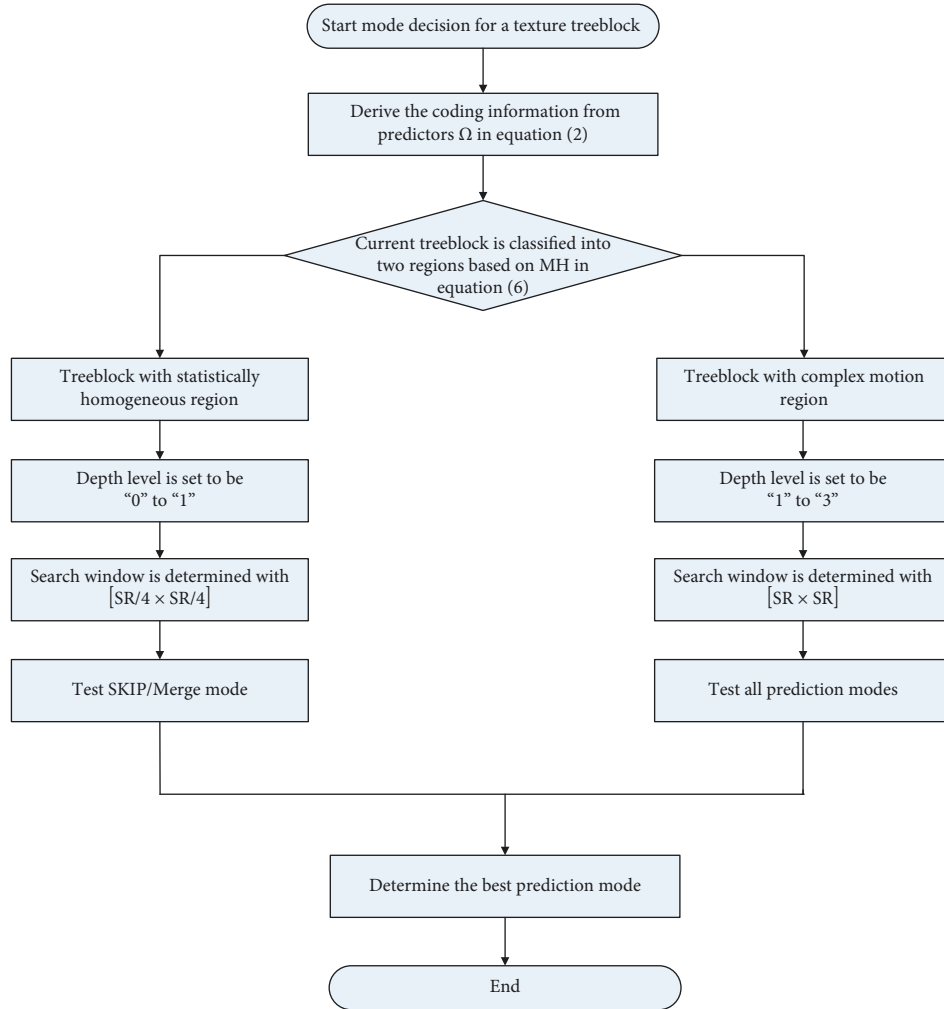


FIGURE 2: Flowchart of the proposed overall algorithm.

TABLE 9: Coding results of FDLRD compared to original 3D-HEVC.

Sequences	Texture video		Rendered view		Dtime (%)
	BDBR (%)	BDPSNR (dB)	BDBR (%)	BDPSNR (dB)	
Kendo	0.68	-0.02	0.22	-0.01	-29.4
Balloons	0.64	-0.02	0.19	-0.01	-28.9
Newspaper	0.89	-0.03	0.39	-0.02	-30.8
Shark	1.23	-0.04	0.58	-0.02	-37.2
Undo_Dancer	1.16	-0.04	0.52	-0.02	-36.5
GT_Fly	0.71	-0.02	0.26	-0.01	-31.2
Poznan_Street	0.27	-0.01	0.11	-0.01	-29.8
Poznan_Hall2	0.18	-0.01	0.04	-0.00	-23.7
Average	0.72	-0.02	0.29	-0.01	-30.9

where $\text{Time}_{\text{proposed}}$ and $\text{Time}_{\text{original}}$ are the run time of the proposed method and the original 3D-HEVC encoders, respectively. Two kinds of situations (“Texture video” and “Rendered view”) are used for experiments. We can measure the “Rendered view” via comparing the encoded render view with images rendered using uncompressed texture video and depth maps. Simulations were run on two Intel Xeon E5-2640 v2 2.0 GHz processors with 32 GB

DDR3 random access memory. The operating system was Windows 7 SP1.

5.1. Performance Evaluation of the Individual Algorithms. Tables 9–11 provide the results of our algorithms compared with the original 3D-HEVC encoders, i.e., fast depth level range determination (FDLRD), early

TABLE 10: Coding results of ESMMD compared to original 3D-HEVC.

Sequences	Texture video		Rendered view		Dtime (%)
	BDBR (%)	BDPSNR (dB)	BDBR (%)	BDPSNR (dB)	
Kendo	0.29	-0.01	0.13	-0.01	-17.6
Balloons	0.25	-0.01	0.11	-0.01	-17.0
Newspaper	0.42	-0.01	0.18	-0.01	-15.4
Shark	0.83	-0.03	0.28	-0.01	-12.9
Undo_Dancer	0.75	-0.03	0.25	-0.01	-12.4
GT_Fly	0.24	-0.01	0.10	-0.01	-18.3
Poznan_Street	0.21	-0.01	0.06	-0.00	-19.1
Poznan_Hall2	0.09	-0.00	0.02	-0.00	-25.3
Average	0.39	-0.01	0.14	-0.01	-17.2

TABLE 11: Coding results of AMSRA compared to original 3D-HEVC.

Sequences	Texture video		Rendered view		Dtime (%)
	BDBR (%)	BDPSNR (dB)	BDBR (%)	BDPSNR (dB)	
Kendo	0.59	-0.02	0.18	-0.01	-28.3
Balloons	0.53	-0.02	0.13	-0.01	-27.8
Newspaper	0.91	-0.03	0.43	-0.02	-31.2
Shark	1.58	-0.05	0.92	-0.03	-39.6
Undo_Dancer	1.47	-0.05	0.85	-0.03	-38.7
GT_Fly	0.88	-0.03	0.23	-0.01	-29.8
Poznan_Street	0.48	-0.02	0.11	-0.01	-30.3
Poznan_Hall2	0.24	-0.01	0.06	-0.00	-24.5
Average	0.84	-0.03	0.36	-0.02	-31.3

TABLE 12: Coding results of the overall algorithm compared to original 3D-HEVC.

Sequences	Texture video		Rendered view		Dtime (%)
	BDBR (%)	BDPSNR (dB)	BDBR (%)	BDPSNR (dB)	
Kendo	0.98	-0.03	0.39	-0.02	-55.3
Balloons	0.91	-0.03	0.36	-0.02	-53.9
Newspaper	1.45	-0.04	0.67	-0.02	-56.2
Shark	2.43	-0.06	1.16	-0.03	-60.8
Undo_Dancer	2.29	-0.06	1.08	-0.03	-59.9
GT_Fly	1.16	-0.03	0.24	-0.01	-57.1
Poznan_Street	0.57	-0.02	0.17	-0.01	-58.1
Poznan_Hall2	0.38	-0.02	0.11	-0.01	-51.6
Average	1.27	-0.04	0.52	-0.02	-56.6

TABLE 13: Runtime ratio of the proposed overall algorithm for only encoding texture video.

Sequences	Dtime (tex) (%)	Dtime (base) (%)	Dtime (dep) (%)
Kendo	-76.6	-60.8	-82.7
Balloons	-75.1	-59.1	-81.8
Newspaper	-79.1	-61.8	-84.2
Shark	-83.3	-65.3	-88.2
Undo_Dancer	-82.1	-64.4	-86.7
GT_Fly	-80.2	-63.9	-84.8
Poznan_Street	-80.1	-65.4	-83.2
Poznan_Hall2	-72.6	-57.1	-78.1
Average	-78.6	-62.2	-83.7

SKIP/Merge mode decision (ESMMD), and adaptive motion search range adjustment (AMSRA). It will be known that three algorithms are capable of accelerating coding time with tiny coding loss. For FDLRD algorithm, about 30.9% encoding time has been reduced. This coding time reduction

is very high for “Shark” (37.2%) and “Undo_Dancer” (36.5%), but still evident for small motion sequences such as “Poznan_Hall2” (23.7%). We receive that the average bitrate increase is 0.72% (or 0.02 dB PSNR drop) for texture video and 0.29% bitrate augment (or 0.01 dB PSNR drop) for

TABLE 14: Coding results of the proposed overall method compared with the FEDTC [25] method.

Sequences	Texture video		Rendered view		Dtime (%)
	BDBR (%)	BDPSNR (dB)	BDBR (%)	BDPSNR (dB)	
Kendo	0.21	-0.01	0.11	-0.01	-3.2
Balloons	0.27	-0.01	0.19	-0.01	-2.7
Newspaper	0.56	-0.02	0.22	-0.01	-7.1
Shark	1.62	-0.05	0.49	-0.02	-9.5
Undo_Dancer	1.51	-0.05	0.56	-0.02	-9.0
GT_Fly	1.07	-0.03	0.29	-0.01	-2.4
Poznan_Street	0.25	-0.01	0.37	-0.02	-11.3
Poznan_Hall2	0.04	-0.00	0.22	-0.01	-8.7
Average	0.69	-0.02	0.31	-0.01	-6.7

rendered views in Table 9. As for the ESMMD algorithm, about 17.2% encoding time has been reduced with the highest gain of 25.3% in “Poznan_Hall2” and the lowest gain of 12.4% in “Undo_Dancer.” The RD loss is very negligible with increase or 0.01 dB PSNR drop for texture video and 0.14% bitrate increase or 0.01 dB PSNR drop for rendered views. This result indicates that ESMMD can accurately reduce unnecessary ME and DE on small CU size. For the proposed AMSRA algorithm, it also shows fine coding efficiency for vary motion activities and texture characteristics sequences, about 31.3% encoding time has been subtracted with the best gain of 39.6% in “Shark” and the lowest gain of 24.5% in “Poznan_Hall2.” Meanwhile, the coding efficiency almost has no loss, with 0.24–1.58% bitrate increase for texture videos and 0.06–0.92% bitrate increase for rendered views. Therefore, the AMSRS algorithm is capable of alleviating computational burden with tiny coding quantity loss.

5.2. Performance Evaluation of the Overall Method. The coding performance of FDLRD, ESMMD, and AMSRA approaches is illustrated in Table 12. In Table 12, our algorithm plays an important role in shorting time for entire sequences which is different with 3D-HEVC encoders. Our algorithm cuts 56.6% time with the minimum of 51.6% for “Poznan_Hall2” and the maximum of 60.8% in for “Shark.” For sequences “Shark,” and “Undo_Dancer,” our algorithm saves more than 59.9% computational complexity. The computation decreasing is large owing to the exhaustive depth level and mode decision process of a bit of texture video treeblocks which are rationally omitted in 3D-HEVC. Table 12 shows a truth that the efficiency decline is tiny, where the bitrate increase is 1.27% (or 0.04 dB PSNR drop) for texture video and 0.52% bitrate increase (or 0.02 dB PSNR drop) for rendered views. Since the proposed overall algorithm is only applicable to the texture video, the encoding time ratio over original 3D-HEVC encoder for only encoding texture video is shown in Table 13. “Dtime (tex),” “Dtime (base),” and “Dtime (dep)” represent runtime savings of all the texture views (including base and dependent views), the base texture views, and the dependent texture views ratio, respectively. As can be seen from Table 13, the proposed overall algorithm can reduce the encoding time to 78.6% for all texture video coding. At the same time, the encoding time ratio over HTM16.0 for encoding the base view and dependent view of texture video

is 62.2% and 83.7%, respectively. It can be observed that the proposed overall algorithm can achieve consistent runtime saving in all base and dependent views coding. Therefore, the proposed overall algorithm keeps nearly the same RD performance of the original 3D-HEVC encoder, reducing the computational complexity of the encoding process considerably.

Figure 3 shows detailed information of our algorithm compared to 3D-HEVC for the two sequences “Newspaper” (1024 × 768), and “Shark” (1920 × 1088) in “Rendered view.” In Figure 3, our algorithm is capable of decreasing the time over a large bitrate range with tiny loss in PSNR and augment in bitrate. Moreover, the encoder runtime will be saving increasingly with decreasing the coding bitrate. Because of the QP increasing, both the potential of only testing coding level 0-1 for texture video treeblocks due to FDLRD and the potential of only deciding SKIP/Merge mode and small motion search range for texture video treeblocks due to ESMMD and AMSRA are all increased.

5.3. Performance Comparison with the State-of-the-Art Methods. In addition to the 3D-HEVC encoder, the results of our algorithm are also compared with the fast 3D-HEVC methods. Tables 14 and 15 compare the proposed overall algorithm with two state-of-the-art fast methods, fast encoder decision for texture coding (FEDTC) [25], and OLCRS [28] based on the “CTC” condition. The QP values used for the texture and depth are chosen as follows: (25, 34), (30, 39), (35, 42), and (40, 45). All algorithms are executed on a computer for comparison. Our algorithm is superior to FEDTC, saving over 6.7% of the coding time with 11.3% in “Poznan_Street” and 2.4% in “GT_Fly.” Due to making full use of the property of texture motion uniformity to predict the current tree block, the poor prediction process of many texture video tree blocks has not been processed by 3D-HEVC encoders, so the calculation is greatly reduced. The mean loss of RD performance is tiny in Table 14. Our algorithm has only 0.69% bitrate augment (or 0.02 dB PSNR drop) for texture video and 0.31% bitrate increase (or 0.01 dB PSNR drop) for rendered views. These mean that our algorithm is able to alleviate lots of computational burdens while keeping nearly the same RD performance as the FEDTC method. Compared with OLCRS, 27.8%–39.7% encoding time has been reduced in Table 15. Meanwhile, the average increase of bitrate is 0.87% (or 0.03 dB PSNR drop)

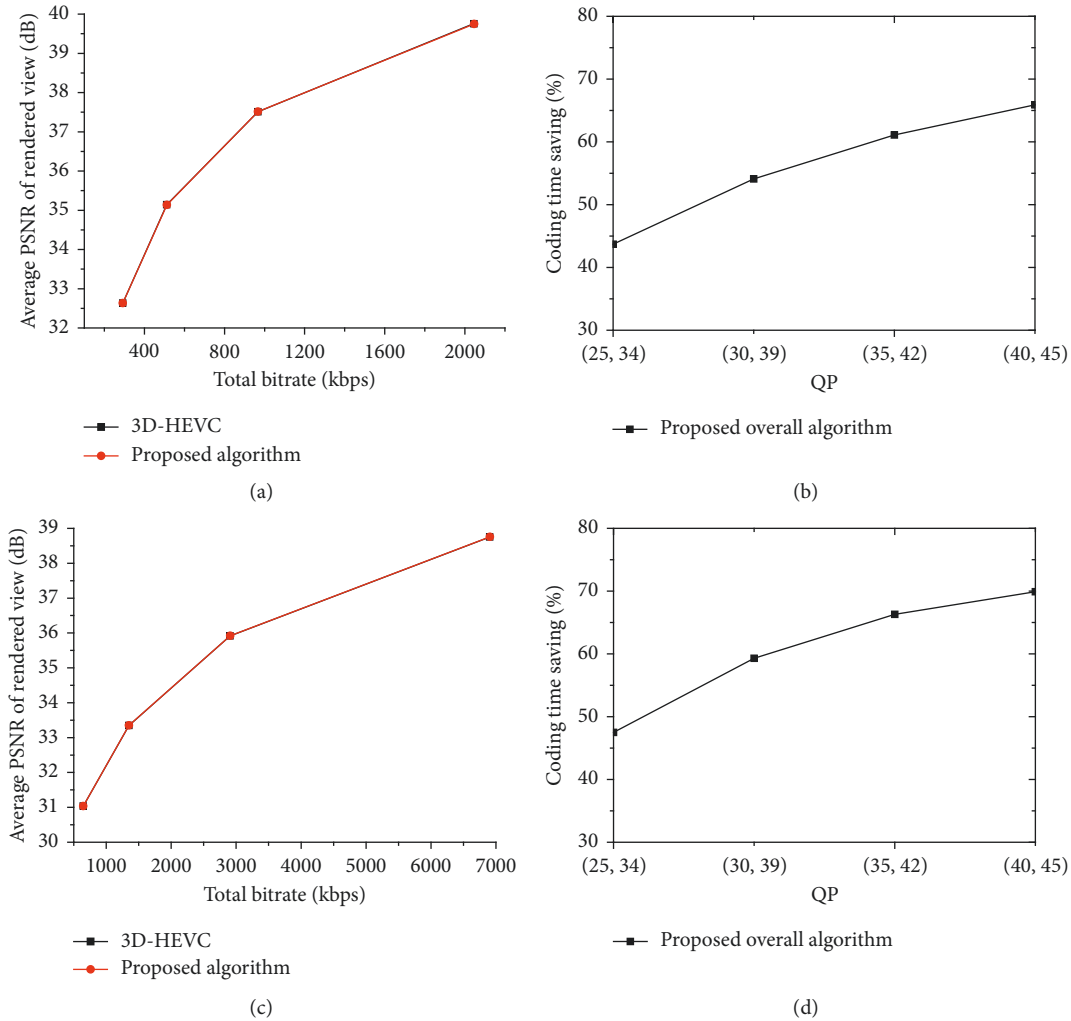


FIGURE 3: Experimental results of “Newspaper” and “Shark.” (a) RD curve of “Newspaper.” (b) Time saving curves of “Newspaper.” (c) RD curve of “Shark.” (d) Time saving curves of “Shark.”

TABLE 15: Coding results of the proposed overall method compared with the OLCRS [28] method.

Sequences	Texture video		Rendered view		Dtime (%)
	BDBR (%)	BDPSNR (dB)	BDBR (%)	BDPSNR (dB)	
Kendo	0.45	-0.02	0.12	-0.01	-31.5
Balloons	0.52	-0.02	0.09	-0.01	-30.1
Newspaper	1.01	-0.03	0.39	-0.02	-31.2
Shark	1.91	-0.05	0.56	-0.02	-39.7
Undo_Dancer	1.84	-0.05	0.95	-0.03	-38.5
GT_Fly	0.92	-0.03	0.11	-0.01	-31.6
Poznan_Street	0.21	-0.01	0.04	-0.00	-32.2
Poznan_Hall2	0.12	-0.01	0.02	-0.00	-27.8
Average	0.87	-0.03	0.29	-0.01	-32.8

for texture video and 0.29% (or 0.01 dB PSNR drop) for rendered views. We can observe from Tables 14 and 15 that our algorithm achieves a higher encoding time saving for each test 3D sequence than FEDTC and OLCRS methods, while both methods do not introduce a noticeable RD efficiency loss. The encoding time saving is particularly high in our algorithm because the full-mode decision process of a significant number of texture video CUs is reasonably

omitted, which demonstrates that the proposed texture video coding based on motion homogeneity are more suitable for selecting candidate modes for 3D-HEVC coding.

Figure 4 proves the coding speed of our algorithm and 3D-HEVC, FEDTC, and OLCRS methods. Our algorithm is supreme for sequences such as “Shark” and “Undo_Dancer” with more runtime saving. Compared with FEDTC and OLCRS, our algorithm receives the best coding

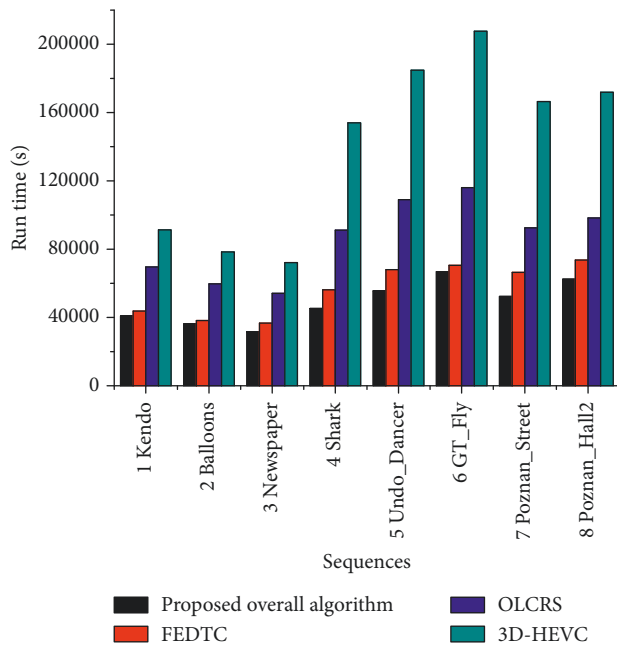


FIGURE 4: Coding speed of the proposed overall method and FEDTC and OLCRS methods compared to 3D-HEVC.

efficiency and accelerates the coding time by 56.6%. The above simulation results demonstrate that the proposed low-complexity texture video coding based on motion homogeneity is efficient for all test sequences and outperforms the state-of-the-art methods with a better coding speed performance.

6. Conclusion

In this paper, we have proposed a fast texture video coding method to accelerate the coding time of 3D-HEVC encoder, which include the three approaches, i.e., fast depth level range determination, early SKIP/Merge mode decision, and adaptive motion search range adjustment. The experimental results show that our algorithm is able to decrease about 56.6% encoding time on average as compared with the HTM 16.0 encoder with only tiny loss of RD performance. Furthermore, it uniformly outperforms two most advanced fast 3D-HEVC methods with an additional 2.4–39.7% coding time saving.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

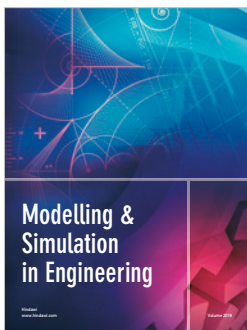
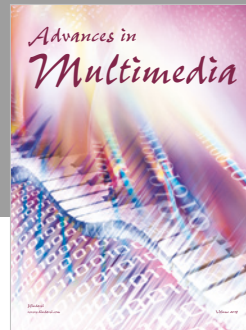
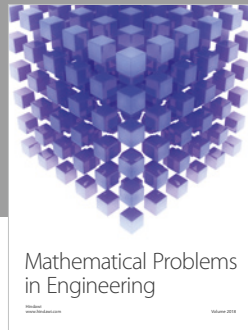
This work was supported in part by the National Natural Science Foundation of China (Nos. 61771432, 61773018, 61302118, 61702464, 61374014, and 61803346), Scientific

Project of Henan Province (Nos. 182102210156 and 182102210610), Innovation Talents of Henan Province (No. 17HASTIT022), Young Key Teacher of Henan Province (No. 2016GGJS-087), and Education Department Project of Henan Province (Nos. 18B510019 and 17B510011).

References

- [1] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC Standard," in *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, 2011.
- [2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [3] G. Tech, Y. Chen, K. Muller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of high efficiency video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 35–49, 2016.
- [4] Y. Chen, G. Tech, K. Wegner, and S. Yea, "Test model 11 of 3D-HEVC and MV-HEVC," in *Proceedings of Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-K1003, 11th Meeting*, Geneva, Switzerland, February 2015.
- [5] K. Muller, H. Schwarz, D. Marpe et al., "3D high-efficiency video coding for multi-view video and depth data," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3366–3378, 2013.
- [6] L. Ding, P. Tsung, S. Chien, W. Chen, and L. Chen, "Content-aware prediction algorithm with inter-view mode decision for multiview video coding," *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1553–1563, 2008.
- [7] W. Zhu, X. Tian, F. Zhou, and Y. Chen, "Fast inter mode decision based on textural segmentation and correlations for multiview video coding," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1696–1704, 2010.
- [8] Y. Zhang, S. Kwong, G. Jiang, X. Wang, and M. Yu, "Statistical early termination model for fast mode decision and reference frame selection in multiview video coding," *IEEE Transactions on Broadcasting*, vol. 58, no. 1, pp. 10–23, 2012.
- [9] T. Zhao, S. Kwong, H. Wang, Z. Wang, Z. Pan, and C.-C. J. Kuo, "Multiview coding mode decision with hybrid optimal stopping model," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1598–1609, 2013.
- [10] J. Lei, J. Sun, Z. Pan, S. Kwong, J. Duan, and C. Hou, "Fast mode decision using inter-view and inter-component correlations for multiview depth video coding," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 978–986, 2015.
- [11] L. Shen, Z. Liu, T. Yan, Z. Zhang, and P. An, "View-adaptive motion estimation and disparity estimation for low complexity multiview video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 6, pp. 925–930, 2010.
- [12] Z.-P. Deng, Y.-L. Chan, K.-B. Jia, C.-H. Fu, and W.-C. Siu, "Fast motion and disparity estimation with adaptive search range adjustment in stereoscopic video coding," *IEEE Transactions on Broadcasting*, vol. 58, no. 1, pp. 24–33, 2012.
- [13] W. Zhu, Y. Zheng, P. Chen, and J. Feng, "Fast macroblock encoding algorithm based on rate-distortion activity for multiview video coding," *Signal Processing: Image Communication*, vol. 29, no. 10, pp. 1063–1078, 2014.

- [14] Q. Zhang, P. An, Y. Zhang, L. Shen, and Z. Zhang, "Low complexity multiview video plus depth coding," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1857–1865, 2011.
- [15] T. L. da Silva, L. V. Agostini, and L. A. da Silva Cruz, "Fast HEVC intra prediction mode decision based on EDGE direction information," in *Proceedings of 20th European Signal Processing Conference*, pp. 1214–1218, Bucharest, Romania, August 2012.
- [16] L. Shen, Z. Zhang, and P. An, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 207–213, 2013.
- [17] T. Zhao, Z. Wang, and S. Kwong, "Flexible mode selection and complexity allocation in high efficiency video coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1135–1144, 2013.
- [18] K. Lee, H.-J. Lee, J. Kim, and Y. Choi, "A novel algorithm for zero block detection in high efficiency video coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1124–1134, 2013.
- [19] L. Shen, Z. Zhang, X. Zhang, P. An, and Z. Liu, "Fast TU size decision algorithm for HEVC encoders using Bayesian theorem detection," *Signal Process.: Image Communication*, vol. 32, no. 10, pp. 121–128, 2015.
- [20] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 14, no. 1, pp. 1–19, 2018.
- [21] Q. Zhang, J. Zhao, X. Huang, and Y. Gan, "A fast and efficient coding unit size decision algorithm based on temporal and spatial correlation," *Optik*, vol. 126, no. 21, pp. 2793–2798, 2015.
- [22] H. R. Tohidypour, M. T. Pourazad, P. Nasiopoulos, and V. Leung, "A content adaptive complexity reduction scheme for HEVC-Based 3d video coding," in *Proceedings of 18th International Conference on Digital Signal Processing (DSP 2013)*, pp. 1–5, Santorini, Greece, July 2013.
- [23] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "A low complexity mode decision approach for HEVC-based 3D video coding using a Bayesian method," in *Proceedings of 2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pp. 895–899, Florence, Italy, May 2014.
- [24] L. Zhang, Y. Chen, X. Li, and S. Xue, "Low-complexity advanced residual prediction design in 3D-HEVC," in *Proceedings of 2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 13–16, Melbourne, Australia, June 2014.
- [25] N. Zhang, D. Zhao, Y.-W. Chen, J.-L. Lin, and W. Gao, "Fast encoder decision for texture coding in 3D-HEVC," *Signal Processing: Image Communication*, vol. 29, no. 9, pp. 951–961, 2014.
- [26] L. Shen, P. An, Z. Zhang, Q. Hu, and Z. Chen, "A 3D-HEVC fast mode decision algorithm for real-time applications," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 11, no. 3, p. 34, 2015.
- [27] Y.-X. Song and K.-B. Jia, "Early merge mode decision for texture coding in 3D-HEVC," *Journal of Visual Communication and Image Representation*, vol. 33, pp. 60–68, 2015.
- [28] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "Online-Learning-Based complexity reduction scheme for 3D-HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 10, pp. 1870–1883, 2016.
- [29] J. Lei, J. Duan, F. Wu, N. Ling, and C. Hou, "Fast mode decision based on grayscale similarity and inter-view correlation for depth map coding in 3D-HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 3, pp. 706–718, 2018.
- [30] Z. Pan, X. Yi, and L. Chen, "Motion and disparity vectors early determination for texture video in 3D-HEVC," *Multimedia Tools and Applications*, pp. 1–18, 2018.
- [31] Q. Zhang, M. Chen, X. Huang, N. Li, and Y. Gan, "Low-complexity depth map compression in HEVC-based 3D video coding," *EURASIP Journal on Image Video Process*, vol. 2015, no. 1, pp. 1–14, 2015.
- [32] Q. Zhang, H. Chang, Q. Wu, and Y. Gan, "Fast motion and disparity estimation for HEVC based 3D video coding," *Multidimensional Systems and Signal Processing*, vol. 27, no. 3, pp. 743–761, 2014.
- [33] C. Yan, Y. Zhang, J. Xu et al., "A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors," *IEEE Signal Processing Letters*, vol. 21, no. 5, pp. 573–576, 2014.
- [34] C. Yan, Y. Zhang, J. Xu et al., "Efficient parallel framework for HEVC motion estimation on many-core processors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 12, pp. 2077–2089, 2014.
- [35] Q. Zhang, Q. Wu, X. Wang, and Y. Gan, "Early SKIP mode decision for three-dimensional high efficiency video coding using spatial and interview correlations," *Journal of Electronic Imaging*, vol. 23, no. 5, pp. 053017-1–053017-8, 2014.
- [36] C. Yan, H. Xie, D. Yang, J. Yin, Y. Zhang, and Q. Dai, "Supervised hash coding with deep neural network for environment perception of intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 284–295, 2017.
- [37] Q. Zhang, N. Zhang, T. Wei, K. Huang, X. Qian, and Y. Gan, "Fast depth map mode decision based on depth-texture correlation and edge classification for 3D-HEVC," *Journal of Visual Communication and Image Representation*, vol. 45, no. 1, pp. 170–180, 2017.
- [38] Q. Zhang, K. Huang, X. Wang, B. Jiang, and Y. Gan, "Efficient multiview video plus depth coding for 3D-HEVC based on complexity classification of the treeblock," *Journal of Real-Time Image Processing*, 2017.
- [39] https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSsoftware/tags/HTM-16.0/cfg/3D-HEVC/.
- [40] K. Mueller and A. Vetro, "Common test conditions of 3DV core experiments," in *Proceedings of Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-G1100, 7th Meeting*, San Jose, CA, USA, January 2014.
- [41] G. Bjontegaard, "Calculation of average PSNR difference between RD-curves," in *Proceedings of 13th VCEG-M33 Meeting*, Austin, TX, USA, April 2001.



Hindawi

Submit your manuscripts at
www.hindawi.com

