Peng et al. EURASIP Journal on Image and Video Processing (2016) 2016:24
DOI 10.1186/s13640-016-0128-3

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Joint processing and fast encoding algorithm for multi-view depth video

Zongju Peng*, Huimin Han, Fen Chen, Gangyi Jiang and Mei Yu

Abstract

The multi-view video plus depth format is the main representation of a three-dimensional (3D) scene. In the 3D extension of high-efficiency video coding (3D-HEVC), the main framework for depth video is similar to that of color video. However, because of the limitation of the mainstream capture technologies, depth video is inaccurate and inconsistent. In addition, the depth video coding method in the current 3D-HEVC software implementation is highly complex. In this paper, we introduce a joint processing and fast coding algorithm for depth video. The proposed algorithm utilizes the depth and color features to extract depth discontinuous regions, depth edge regions, and motion regions as masks for efficient processing and fast coding. The processing step includes spatial and temporal enhancement. The fast coding method mainly limits the traversal of the CU partition and mode decision. Experimental results demonstrate that the proposed algorithm reduces the coding time and the depth video coding bitrate; the proposed algorithm reduces overall coding time by 44.24 % and depth video coding time by 72.00 % on average. In addition, there is a 24.07 % depth video coding bitrate reduction with an average of 1.65 % Bjontegaard delta bitrate gains.

Keywords: Fast depth video coding, Depth video processing, Prediction mode decision

1 Introduction

As computing, communication, and multimedia technologies rapidly advance, users are increasingly interested in three-dimensional (3D) video system applications, such as 3D television, free-viewpoint television, and photorealistic rendering of 3D scenes [1, 2]. Multi-view video plus depth (MVD) is a mainstream format that represents 3D scenes. MVD includes multiple viewpoint color and depth video. The color video represents the visual information of the scene while the corresponding depth video represents the geometric information of the 3D scene. In this scene representation, virtual views are synthesized using depth image based rendering (DIBR) [3]. The MVD data format satisfies the 3D video system's requirements and supports the wide viewing angle of 3D displays and autostereoscopic displays [4]. However, MVD contains a large amount of data, which becomes a challenge for data storage and network transmission. Therefore, multi-view depth video as well as color video should be compressed efficiently [5].

Recently, depth video coding has become an active research area. High compression ratio, high virtual view quality, and low computational complexity are the targets of depth video coding. Cheung et al. [6] proposed depth map compression technique based on sparse representation. Lei et al. [7] improved the depth video coding performance by utilizing the depth-texture and motion similarities. Liu et al. [8] presented two depth compression techniques, the trilateral filter and sparse dyadic mode. Kang et al. [9] designed an adaptive geometry-based intra-prediction scheme for depth video coding. Oh et al. [10] proposed a depth boundary reconstruction filter to code the depth video. Zhang et al. [11] proposed regional bit allocation and rate distortion (RD) optimization algorithms for multi-view depth video coding using the imbalance bitrate allocation for different regions. Shao et al. [12] proposed a depth video coding algorithm based on distortion analysis. Standardization of MVD coding was also investigated by the MPEG and ITU-T/ISO/IEC Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) [13]. The standard for high-efficiency video coding (HEVC)

* Correspondence: pengzongju@126.com
Faculty of Information Science and Engineering, Ningbo University, Ningbo, China

has been completed by ISO and ITU. However, the 3D extension framework for HEVC (3D-HEVC) remains under development.

So far, depth video of natural scene can be obtained via Kinect sensors, depth camera systems, and depth estimation software [14–16]. The depth video is inaccurate and inconsistent because of the corresponding technique limitation. Depth images captured by Kinect sensors suffer from temporal flickering, noise, and holes because of the principle limitation of structured light technique. The depth video, obtained by depth camera system which is based on the principle of time-of-flight, may be inconsistent with the scene because of ambient light noise, motion artifacts, specular reflections, and so on. Depth video estimated by software usually contains discrete and rugged noises. Consequently, the ideal compression performance cannot be achieved, even if the state-of-the-art encoding methods are used. To improve encoding and virtual view rendering performance, many depth video processing algorithms [17–23] have been proposed. Hu et al. [17] proposed a depth video restoration algorithm which is effective for depth video corrupted by additive white Gaussian noise. Nguyen et al. [18] suppressed the coding artifacts over object boundaries by using a weighted mode filtering. Zhao et al. [19] proposed a depth no-synthesis-error (D-NOSE) model and presented a smoothing scheme for depth video coding. Lei et al. [20] proposed a depth sensation enhancement method for multiple virtual view rendering. Silva et al. [21] proposed a depth processing method based on a just noticeable depth difference (JNDD) model. In our previous works, the correlation of depth video is enhanced for high compression performance [22, 23]. The main contribution of these algorithms [17–23] is for better virtual view quality or high compression ratio under H.264/AVC framework. In 3D-HEVC, the size of CU and prediction modes are different from those of H.264/AVC; ideal compression ratio might not be achieved under 3D-HEVC.

The computational complexity of depth video coding is also a concern. The 3D-HEVC standard still uses the quadtree partition structure introduced in HEVC [24]. Both color and depth videos are split into a sequence of coding tree units (CTUs), and each CTU is recursively divided into four leaf coding units (CUs); the largest CU size is 64×64. Each CU contains different prediction units (PUs), and different prediction modes, i.e., SKIP, merge, inter modes (Inter2N×2N, Inter2N×N, InterN×2N, and InterN×N), asymmetric motion partitioning (AMP) modes (Inter 2N×nU, Inter 2N×nD, Inter nL×2N, Inter nR×2N), and intra-prediction modes. All prediction modes are probed among all temporal and inter-view frames to determine the optimal mode for the current CU that achieves the best RD performance. It is clear that adopting the full search scheme to obtain the best CU quadtree structure as well as the motion or disparity vector for each CU consumes considerable search time [25, 26]. So far, many fast algorithms have been proposed to optimize mode selection, reference frame selection, motion estimation, and disparity estimation in depth video coding [27–33]. These algorithms can be categorized into two classes, those that exploit the correlations between color and depth video [27–29] and those that use depth image features [30–33]. Zhang et al. [27] proposed a low complexity MVD coding algorithm that includes motion vector sharing based on the texture image similarity correlation and SKIP mode decision in depth video coding. Shen et al. [28] proposed a fast depth video coding algorithm that uses the correlations of the prediction modes, reference frames, and motion vectors from color videos and depth maps. Lee et al. [29] proposed a fast and efficient multi-view depth image coding algorithm based on the temporal and inter-view correlations between the previously encoded texture images. Park [30] proposed a fast depth video coding algorithm based on edge classification and depth-modeling mode omission. Mora et al. [31] presented quadtree limitation coding tool and the associated predictive coding part. The runtime of depth video coding is reduced by exploiting the texture-depth correlation. Tsang et al. [32] proposed an intra-prediction algorithm using a single prediction direction instead of multiple prediction directions. Wang et al. [33] proposed a fast depth video encoding algorithm based on depth video partitioning. However, these algorithms were proposed based on inconsistent and inaccurate depth video and still have room for improvement for processed depth video.

In conclusion, the studies on depth video processing and fast depth video coding are conducted separately. Actually, the optimization chain of depth video processing and depth video coding should be coupled instead of mutually independent. Hence, a joint depth video processing and fast encoding algorithm is proposed in this paper. The algorithm includes two aspects. One is depth video processing for 3D-HEVC coding. The other is the fast depth video coding based on the processed depth video. The fast encoding method, based on acceleration of CU partition and mode decision, is suitable for the processed depth video. The proposed algorithm not only improves the compression ratio but also speeds up the encoding process.

This paper is a follow-up work which the depth video is processed based on our previous works [22, 23]. The contribution of this paper is the joint algorithm which couples depth video processing and fast algorithms together.

The rest of this paper is organized as follows. Motivation and analysis of this paper are given in Section 2, and the proposed algorithm is described in Section 3. Section 4 presents the experimental results, and the conclusions are drawn in Section 5.

2 Motivation and analysis

Depth video contains a sequence of gray images that represent the distance between the objects and capturing device. In the MVD system, depth video is auxiliary information used for rendering virtual views. Figure 1a, b shows the tenth frames of the second view in the color video and corresponding depth video of the "Newspaper1" sequence. The depth video was obtained by depth estimation software. It is inaccurate and inconsistent. The object edge in the depth map does not exactly align with the real edge. The color map has more details than the depth map; most edges in the depth map are included in edges of the color map, except for some inaccurately estimated regions such as the pixels indicated by the green rectangular box. Depth map pixels in the green box should have the same pixel value as their surroundings, as the pixels in color video clearly have the same distance from camera. In this study, we focus on the problem of inaccurate and inconsistent depth maps and propose a processing method to enhance this kind of depth video.

In the 3D-HEVC, color video and depth video are successively coded. The inaccuracy and inconsistency of depth video not only deteriorates the compression ratio but also affects the design of fast encoding algorithm. Figure 2a, b shows the CU splitting and mode selection of the original and the smoothed depth video in the "Newspaper1" sequence, respectively. The green mask in the depth video represents whether the final prediction mode is SKIP or merge, and the cross line is the final splitting quadtree result. Most CTUs in flat regions contain no further splitting, and splitting always occurs around edge regions. By comparing the encoding results in the enlarged area, we conclude that the CU splitting and mode selection are more complicated for original depth video than they are for smoothed depth video. Most of the fast algorithms proposed so far are based on the statistical results of CU splitting and mode selection in original depth video. In other words, these algorithms do not consider the effect of depth video inaccuracy and inconsistency on the computational complexity of video coding. Hence, in this paper, we proposed a joint processing and fast encoding algorithm.

3 Joint depth video processing and fast encoding algorithm

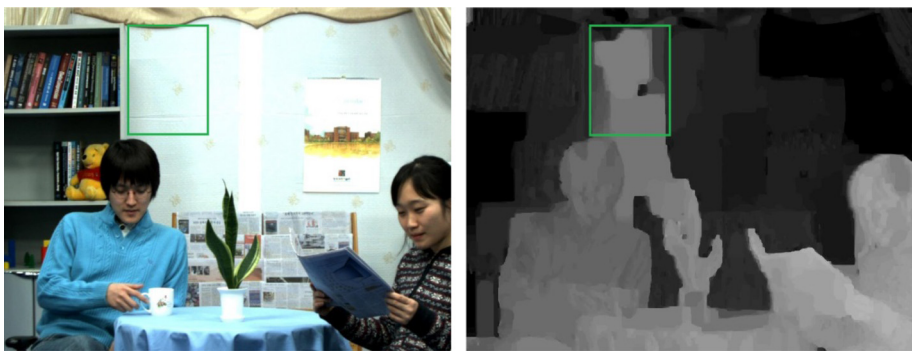
Figure 3 shows the block diagram of the proposed algorithm. It mainly consists of two parts, depth video processing and fast encoding. In depth video processing method, the discontinuous regions (DRs) and edge regions (ERs) of original depth video are firstly extracted and then the original depth video is spatially and temporally enhanced consecutively. We define DRs as the regions in which depth value abrupt varies. In fast depth video coding method, the ERs of processed depth video and motion regions (MRs) of color video are extracted and then different CU partition and mode decision strategies are utilized.

3.1 Mask extraction

In the proposed algorithm, DRs, ERs, and MRs are extracted as masks for the succeeding processing and encoding process. Their extraction methods are detailed in this subsection.

3.1.1 Discontinuous and edge regions extraction

In 3D video systems, most capturing camera systems are arranged in parallel. In parallel camera systems, depth distortion results only in horizontal geometric displacement according to the principle of DIBR [23]. Hence, abrupt depth variation along horizontal direction will lead to large holes during the rendering process and



(a) Image in color video (b) Image in depth video

Fig. 1 Comparison between color and depth images in "Newspaper1" sequence. **a** Image in color video. **b** Image in depth video

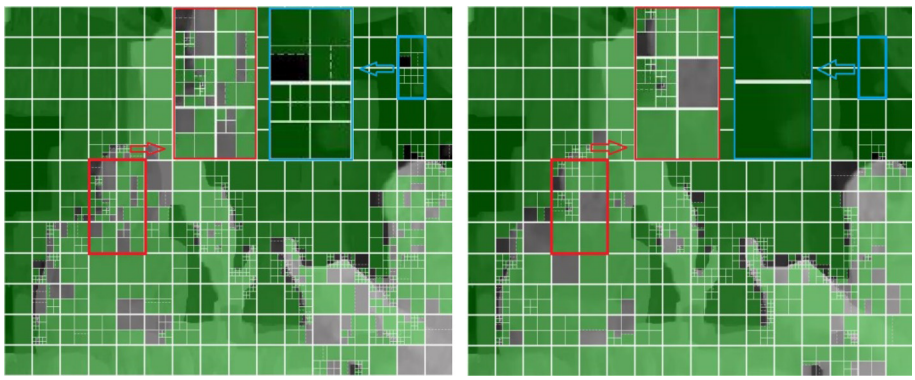


Fig. 2 CU splitting and mode selection comparison. **a** Original depth video. **b** Smoothed depth video

consequently deteriorates the virtual view quality. Let $\mathbf{R}_{\mathbf{DR}}$ be the DRs which are extracted by

$$\mathbf{R}_{\mathbf{DR}} = \{(i,j),(i-1,j) \mid 0 \le i \le W, 0 \le j \le H, \quad abs(dp(i-1,j)-dp(i,j)) > Th_0\}, \tag{1}$$

where $dp(i, j)$ is the depth value at (i, j), W and H are the width and height, respectively, of a frame in a depth video, and Th_0 is set to 10 as an empirical threshold, $abs(\cdot)$ represents obtaining an absolute value. Figure 4 shows DR extract result of the tenth frame of the second view in the "Newspaper1" depth sequence. Clearly, most of DRs are located at object boundaries, except for some noise that is caused by inaccurate depth estimation.

Object edges in depth video are important for virtual view rendering, and pixel value variations in these regions also lead to virtual view distortions during the rendering process. We use classical Canny operator to extract the edges for preservation. The edge extraction process includes depth smoothing using a Gaussian filter that is designed to reduce the impact of noise during the process. Calculation of the amplitude and angle of depth gradients, non-maximum suppression of gradient amplitudes, and edge extraction using double-threshold method are conducted consecutively.

Canny operator uses a double threshold to extract ERs; we set at 120 and 40 based on experiments in this paper. Figure 5 shows the ERs of the second view in the "Newspaper1" depth video.

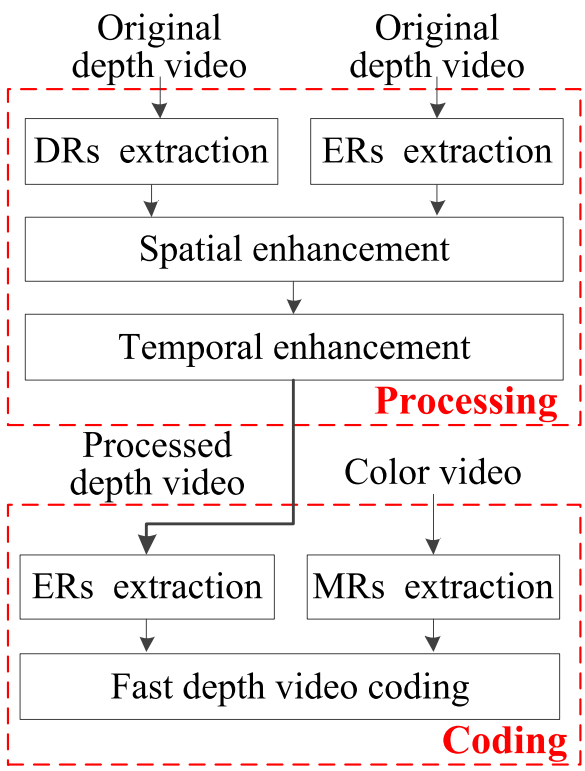


Fig. 3 Framework of the proposed algorithm

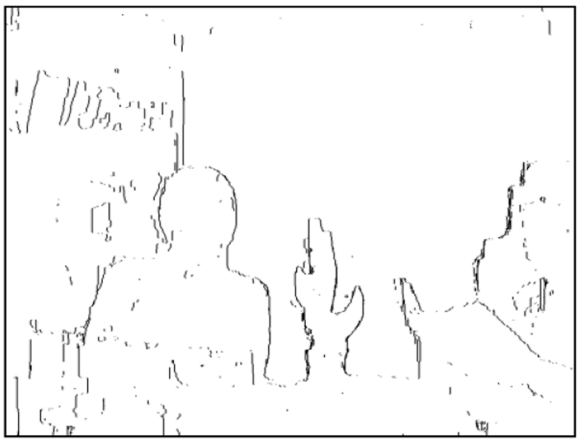


Fig. 4 DR extraction result

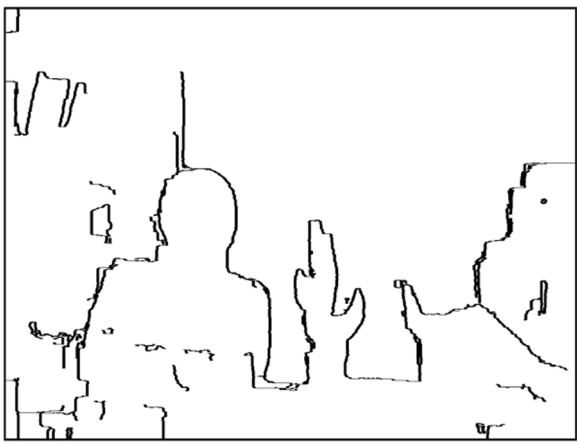


Fig. 5 ER extraction result

3.1.2 Motion regions extraction

Because depth video is inconsistent in the temporal direction, it is difficult to extract accurate motion regions. In this study, motion regions of the color video are used to predict the motion regions of depth video. We calculate the sum of square difference (SSD) between the current 4×4 block and the corresponding blocks in the reference frames. Let $\mathbf{R_{MR}}$ be the MRs which are obtained by

$$\mathbf{R_{MR}} = \left\{ x \,\middle|\, \min\left(SSD_f(x), SSD_b(x)\right) \le Th_1, x \in \mathbf{\Omega} \right\}, \tag{2}$$

where $SSD_f(x)$ and $SSD_b(x)$ be the SSDs of the co-located blocks of block x in the forward frame and backward reference frames, Ω is the universal set which represent all blocks in depth video, and Th_1 is a threshold and is set to 1000 based on experiments. Figure 6 shows the MRs of the second view in the "Newspaper1" where white block regions indicate MRs.

Fig. 6 The mask of motion region

3.2 Depth processing

In the proposed algorithm, the depth video is processed in a manner that enhances the spatial and temporal correlations.

3.2.1 Depth video spatial enhancement

The ER is preserved for the sake of rendering performance. For non-ERs, a Gaussian filter and adaptive window smoothing filter are used, respectively.

The Gaussian filter is selected by considering the tradeoff between encoding performance and virtual view quality. First, it is easy to achieve a high compression ratio for smooth depth video. Second, in the DIBR process, rendering holes are decreased while edge regions are appropriately smoothed. Some small holes may disappear when using the smoothing filter.

An adaptive window smoothing filter is defined as follows: if current depth pixel does not belong to DR and ER, we set this pixel to be the center of an adaptive window with a max search range n, where n represents the number of consecutive pixels, and n is set to 5 empirically. Four directions which include left, right, up, and down are simultaneously searched. If any pixel in DR or ER is detected, the search is stopped. A cross-shaped window is then formed, as shown in Fig. 7a.

Next, set each pixel in the vertical axis of the cross window as the centers, and n is the max search range. We check pixels in the left and right directions. If any pixel which does not belong to DR or belong to ER is detected, this process is stopped. After the left and right directions of each center are searched, the adaptive window is formed, as shown in Fig. 7b. The pixel of the adaptive window center is set to the mean of all pixels within this window, and the adaptive window smoothing filtering for the current pixel is completed.

3.2.2 Depth video temporal enhancement

To further enhance the depth video correlation and reduce the coding bitrate, we use a depth video temporal processing method after the spatial enhancement. The temporal enhancement method is similar to part of our previous works [22, 23]. For the convenience of narration, the coordinates of the pixels in depth video are 3D. Besides common horizontal and vertical coordinates, temporal direction is appended as a coordinate because temporal consecutive frames are used. Let $dp(i, j, t)$ be the pixel at (i, j) in the t-th frame of original depth video, $dp\ \ (i, j, t)$ be the corresponding filtered depth value, the temporal filtering is represented as

$$dp'(i,j,t) = \frac{\sum_{t'=t-t_0}^{t+t_0} w(i,j,t') \times dp(i,j,t')}{\sum_{t'=t-t_0}^{t+t_0} w(i,j,t')}, \tag{3}$$

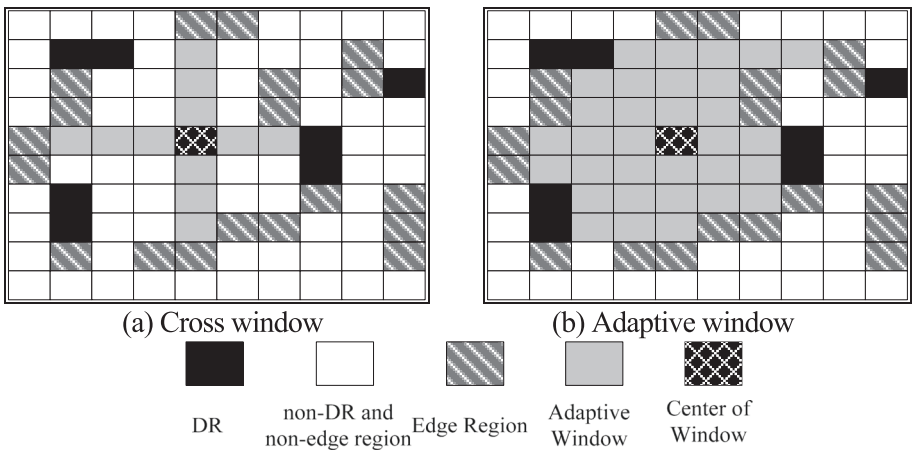


Fig. 7 Illustration of forming an adaptive window. **a** Cross window. **b** Adaptive window

where t_0 is the size of the temporal filtering window and t_0 is empirically set to 4. $w(i,j,t')$ is the filtering weight at (i,j,t') which combines the temporal interval and depth difference and can be expressed as

$$w(i,j,t') = w_1(t') \times w_2(i,j,t'). \tag{4}$$

In general, as the temporal interval increases, the correlation of the pixels in the depth video along the temporal direction decreases. Hence, we use the exponential function to simulate a relationship between the correlation weight and temporal interval.

$$w_1(t') = e^{1-|t'-t|} \tag{5}$$

Pixel values in depth video represent the distance between the camera and captured objects. The probability that two depth pixels belong to the same object decreases as the pixel difference increases. We model this relationship as $w_2(i,j,t')$ and calculated by

$$w_2(i,j,t') = 1 - \frac{1}{1+e^{5-\frac{dif(i,j,t')}{4}}}, \tag{6}$$

where $dif(i,j,t')$ is the difference between corresponding pixels of the temporal consecutively frames,

Table 1 Test conditions

Full resolution of color and depth video	
Color QP values	25, 30, 35, 40
Depth QP values	34, 39, 42, 45
VSO	ON
Texture SAO : ON	Texture SAO : OFF
RDOQ	ON
View synthesis s/w	1D-fast VSRS

$$dif(i,j,t') = |dp(i,j,t) - dp(i,j,t')|. \tag{7}$$

Depth map is a gray image with DRs, ERs, and large areas of constant depth value. DRs or ERs in an accurate depth map should be exactly aligned with the objects' boundaries in the color video. However, the depth videos are always inaccurate, and some of the depth edges do not exactly correspond to the color edge positions. After depth video processing, the inconsistency and discontinuity in the depth video decrease in the spatial domain, as inconsistent pixels in the time domain have been filtered out. Eventually, depth video consistency is enhanced. Many DRs in the processed depth video are reduced and become flat regions. As the proposed processing method protects sensitive regions for rendering, the quality degradation of the rendered view is restricted to a minimal degree. During the coding process, stationary and non-ERs in successive frames are very similar to the collocated and neighbor regions. Hence, a predetermination of prediction modes will reduce the coding complexity without significant fluctuations in the rendered view quality.

3.3 Fast depth video coding

Depth video coding in 3D-HEVC inherits the most effective technologies in the color video coding process and adds some tools that are designed for depth video. In the coding process, the optimal prediction modes and

Table 2 Test sequence information and view numbers used for encoding

Sequence	Resolution	Views	Virtual views
Balloons	1024 × 768	3–1	2
Kendo	1024 × 768	5–3	4
Newspaper1	1024 × 768	4–2	3
Poznan_Street	1920 × 1088	3–5	4

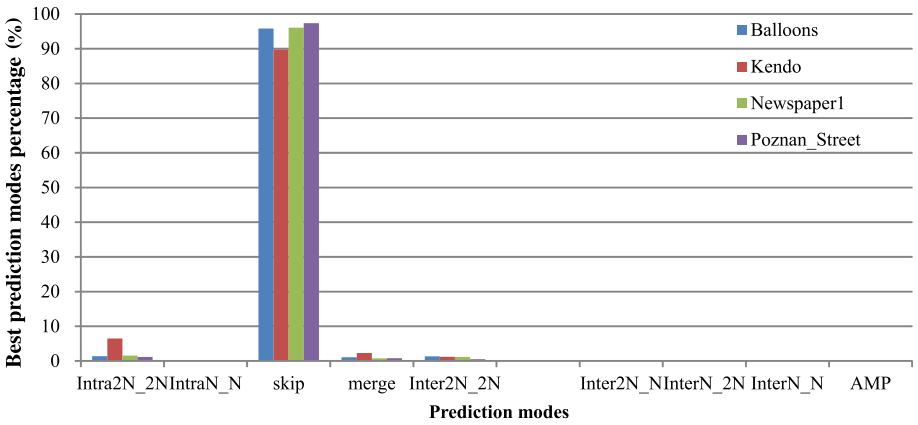


Fig. 8 Statistical distribution of prediction modes in Non-ER and non-MR in depth video

the CU splitting depth are decided by an RD optimization process, and all the decision process is time consuming.

Compared with other prediction modes, the SKIP modes require less computational complexity. Each inter mode coded CU will determine the best motion parameters, including the motion vector, reference picture index, and reference picture list flag, while a CU coded in SKIP mode only contains one PU without a significant transform coefficient or motion vectors, and the reference picture list flag is inherited from the merge mode. The merge mode can be applied to the SKIP mode and any inter mode.

The proposed fast encoding method focuses on improving the prediction mode decision process and the CU splitting process. Since most of the CU splitting depth of depth video is less than that the corresponding CU splitting depth in color video [34], we use the corresponding CU splitting depth in color video as upper bound and narrow the CU depth range. We divide pixels in the depth video into two classes, $\mathbf{R_{MR}}\cap\mathbf{R_{ER}}$ and others. The regions $\mathbf{R_{MR}}\cap\mathbf{R_{ER}}$ contain motion of different objects or different classes of pixels with respect to edge properties. It is suitable to use a fine prediction mode test to determine the best one. Other regions in the depth map contain large areas of flat regions. They are highly likely to have the same pixel values as the neighboring inter coded unit, which means that the best prediction mode is likely to be SKIP or merge. The proposed method pre-decides the prediction modes, which can simplify the RD optimization process and reduce computational complexity.

Given these considerations, the statistical distribution of the prediction modes in these regions was analyzed using 3D-HEVC reference software HTM 10.0 [34], with the configuration listed in Table 1 and using test sequences listed in Table 2. The test sequences, "Kendo" and "Balloons," have many MRs, "Newspaper1" has many edges, and "Poznan_Street" is relatively flat. The proposed fast coding method mainly decreases the coding complexity in non-ERs and non-MRs in depth video. Figure 8 shows the optimal mode distribution in these regions, where all AMP modes are chosen in 1 % of cases, and, for all sequences, the percentage of SKIP mode is about 90 %. Other prediction modes were chosen at a rate of less than 10 %, especially for sequences with many flat regions, such as "Poznan_Street." If we limit the prediction mode to SKIP and merge mode, less distortion will occur.

Figure 9 shows the flowchart of the proposed fast encoding method which is described as follows.

Step 1 If the collocated color CU depth is larger than current CU depth, further splitting is taken.

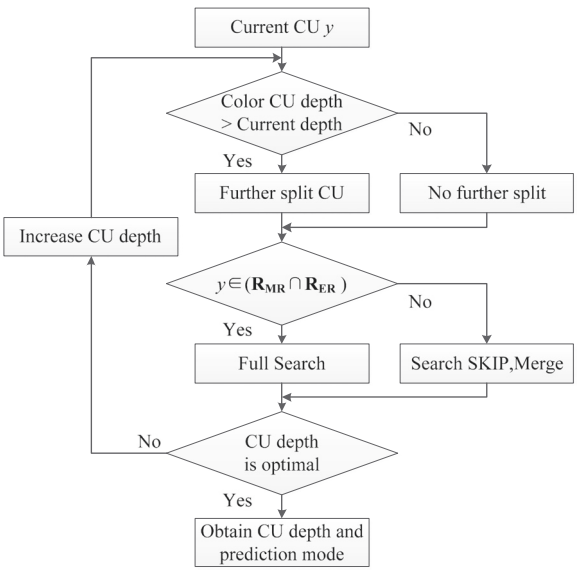


Fig. 9 Flowchart of the proposed fast encoding method

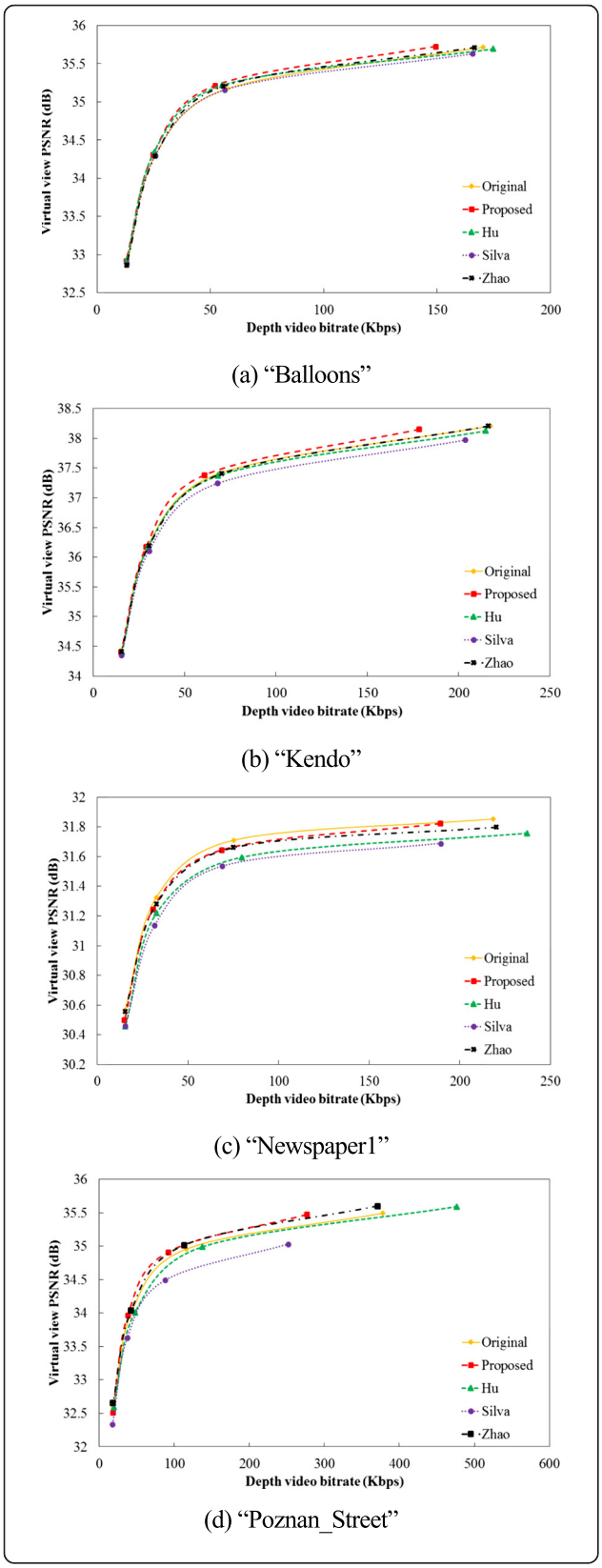


Fig. 10 RD curves of the original depth video, the proposed processing depth video, Hu's method, Silva's method, and Zhao's method: **a** "Balloons," **b** "Kendo," **c** "Newspaper1," and **d** "Poznan_Street"

Otherwise, further splitting is stopped, and the optimal CU depth is determined.

Step 2 If current CU belongs to $R_{MR} \cap R_{ER}$, all the prediction modes are searched to select the optimal one. Otherwise, only search SKIP and Merge modes.

Step 3 If current CU depth is the optimal, the final optimal CU depth and prediction mode is determined. Otherwise, increase the current CU depth, go to step 1 until the current CU depth is optimal.

4 Experimental results

In this section, we evaluate the performance of the proposed algorithm under the common test conditions required by JCT-3V which are shown in Table 1 [35]. We tested several sequences provided in the 3DV core experiments in two view configurations (right-left): "Poznan_Street," provided by Poznan University of Technology; "Kendo" and "Balloons," provided by Nagoya University, and "Newspaper1" provided by Gwangju Institute of Science and Technology. As the proposed algorithm is specially designed for depth videos that are estimated using stereo matching, the computer-generated sequences, "Undo_Dancer," "GT_Fly," and "Shark" were not tested. Table 2 lists the details of the test sequences used for coding.

4.1 Performance evaluation of depth video processing

Depth videos are geometric information of 3D scene and used for virtual view rendering on the client side. Hence, the performance of depth video processing method is assessed by the virtual view quality and bitrate of depth video. Virtual views are listed in Table 2, which were rendered from the decoded video using the 1D-fast VSRS method [35]. We evaluate the depth video processing method of this paper by comparing experimental results with Hu's [17], Silva's [21], and Zhao's [19] methods. It is noted that, for Hu's method, the white Gaussian corrupted depth video is used for testing. We test Hu's method with the condition of standard deriv-ation σ at 10.

Figure 10 shows the depth video coding rate distortion curves (RD curves) of the original, the proposed, Hu's, Silva's, and Zhao's methods in different test sequences. The results show that the proposed method outperforms the others. We take the sequence "Newspaper1" as an example to explain the comparison result of rate distortion performance. Figure 11 shows the processed results and the corresponding local enlargements of the

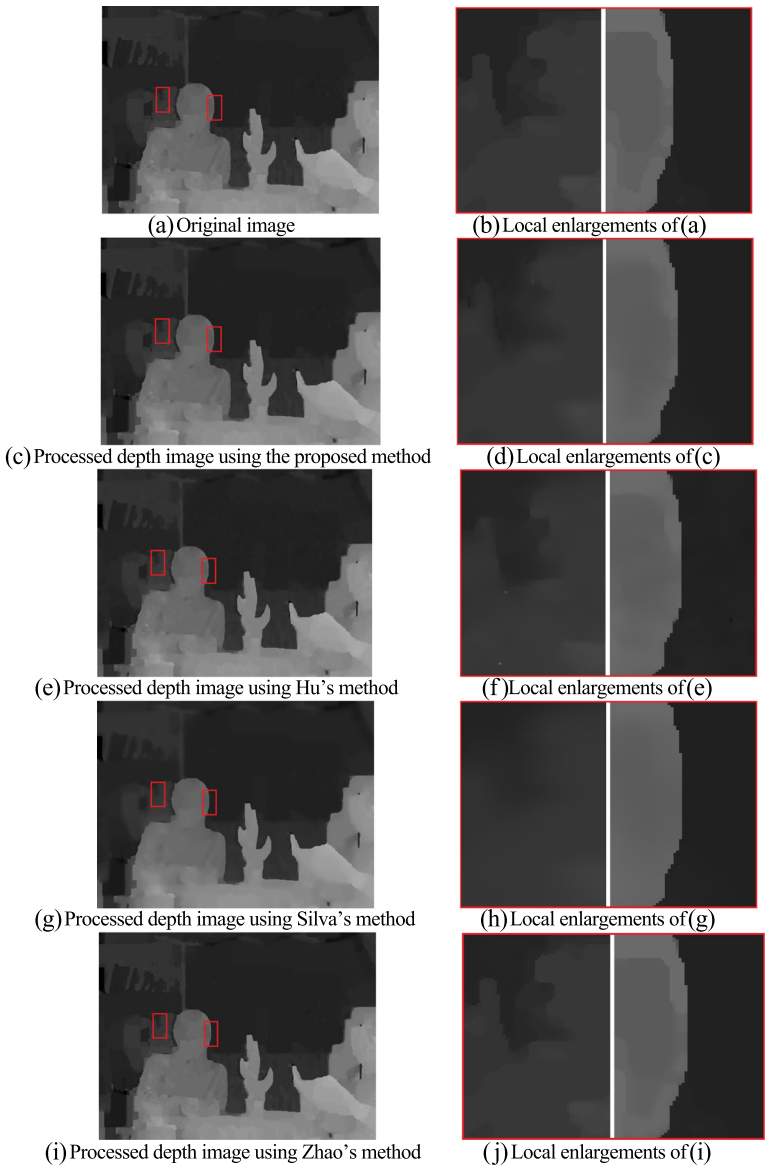


Fig. 11 Depth video and local enlargement in "Newspaper1" sequence: **a** original image, **b** local enlargements of **a**, **c** processed depth image using the proposed method, **d** local enlargements of **c**, **e** processed depth image using Hu's method, **f** local enlargements of **e**, **g** processed depth image using Silva's method, **h** local enlargements of **g**, **i** processed depth image using Zhao's method, and **j** local enlargements of **i**

original, the proposed, Hu's, Silva's, and Zhao's methods. For the proposed method, as shown in Fig. 11c, d, the depth video is spatially and temporally smoothed, and the edge regions are preserved to maintain the virtual view quality. Since white Gaussian corrupted depth video is used in Hu's method, many noises still exist in the processed depth video. The bitrate of Hu's method is higher than those of the other methods, because it is suitable for white Gaussian corrupted depth videos but not good for general ones. In Silva's and Zhao's methods, depth video is spatially processed with JNDD and D-NOSE model, respectively. As the constraint of D-NOSE model is more rigorous than JNDD model, the processed depth video of Silva's method is smoother than that of Zhao's method. Hence, the bitrate of Silva's method is lower than Zhao's method. Although the processed depth video of Silva's method is also smoother than that of the proposed method, the edges of the depth video of Silva's method are not well preserved. Consequently, the edge distortion will deteriorate the virtual view quality.

4.2 Performance evaluation of fast coding

After depth processing, fast coding is performed on the processed depth video. We conducted coding complexity analysis using percentage of encoding time reduction.

Table 3 Reduction of overall and depth video encoding times for Mora's scheme and the proposed algorithm (%)

Sequence	QP = 25		QP = 30		QP = 35		QP = 40		Average	
	$\Delta T_{MORA}/\Delta t_{MORA}$	$\Delta T_{proposed}/\Delta t_{proposed}$	$\Delta T_{MORA}/\Delta t_{MORA}$	$\Delta T_{proposed}/\Delta t_{proposed}$	$\Delta T_{MORA}/\Delta t_{MORA}$	$\Delta T_{proposed}/\Delta t_{proposed}$	$\Delta T_{MORA}/\Delta t_{MORA}$	$\Delta T_{proposed}/\Delta t_{proposed}$	$\Delta T_{MORA}/\Delta t_{MORA}$	$\Delta T_{proposed}/\Delta t_{proposed}$
Balloons	−32.63/−58.13	−36.51/−65.42	−37.64/−64.96	−40.23/−69.74	−41.49/−68.27	−43.44/−71.32	−44.29/−70.44	−46.00/−72.34	−35.10/−60.39	−41.55/−69.71
Kendo	−33.47/−57.78	−38.25/−65.81	−36.37/−63.07	−40.23/−69.24	−39.40/−66.14	−42.83/−71.35	−41.66/−68.56	−44.51/−72.31	−39.52/−66.07	−41.46/−69.68
Newspaper	−37.32/−62.32	−41.82/−69.41	−41.50/−67.14	−44.74/−72.30	−44.70/−69.66	−47.10/−73.89	−47.08/−71.57	−49.43/−74.92	−43.01/−68.94	−45.77/−72.63
Poznan_Street	−36.98/−63.32	−43.24/−72.88	−42.56/−69.13	−46.95/−75.76	−46.45/−71.70	−50.15/−76.92	−49.21/−73.56	−52.37/−78.38	−45.56/−71.03	−48.18/−75.99
Overall									−40.80/−66.61	−44.24/−72.00

Table 4 Bitrate variation of depth video for Mora's scheme and the proposed algorithm (%)

	QP = 25		QP = 30		QP = 35		QP = 40		Average	
Sequence	ΔBR_{MORA}	$\Delta BR_{\text{proposed}}$	ΔBR_{MORA}	$\Delta BR_{\text{proposed}}$	ΔBR_{MORA}	$\Delta BR_{\text{proposed}}$	ΔBR_{MORA}	$\Delta BR_{\text{proposed}}$	ΔBR_{MORA}	$\Delta BR_{\text{proposed}}$
Balloons	−23.83	−31.78	−21.04	−28.97	−14.00	−17.08	−11.20	−8.23	−24.10	−34.44
Kendo	−11.05	−25.99	−9.30	−18.33	−7.94	−11.80	−7.45	−6.65	−17.47	−31.85
Newspaper	−35.28	−43.19	−26.11	−33.71	−17.78	−24.44	−9.31	−9.00	−11.45	−21.10
Poznan_Street	−35.40	−54.55	−21.77	−38.72	−14.55	−24.56	−9.16	−8.16	−15.80	−9.30
Overall									−17.20	−24.07

Three schemes, the HTM10.0, Mora's scheme [31], and the proposed joint depth video processing and fast encoding algorithm, were tested. Mora's scheme has been adopted to the current 3D-HEVC. We use the encoding time of HTM10.0 as a benchmark to compute the time reduction of Mora's scheme and the proposed algorithm. The computational time of the proposed algorithm includes both depth video processing and fast depth video encoding. Let T_{ori}, T_{MORA}, and T_{proposed} be the overall encoding time of MVD signal using HTM10.0, Mora's scheme and the proposed algorithm, t_{ori}, t_{MORA}, and t_{proposed} be the encoding time of corresponding depth video using HTM10.0, Mora's scheme and the proposed algorithm, the overall time reduction ΔT_i and depth video coding time reduction Δt_i (i = {MORA, proposed}) were evaluated by

$$\Delta T_i = \frac{T_i - T_{\text{ori}}}{T_{\text{ori}}} \times 100\% \tag{8}$$

$$\Delta t_i = \frac{t_i - t_{\text{ori}}}{t_{\text{ori}}} \times 100\% \tag{9}$$

Table 3 lists the time reduction of the proposed method and Mora's scheme. The proposed method saves overall time by 44.24 % on average and depth video encoding time by 72.00 % on average. For Mora's scheme, the corresponding average time reduction is about 40.8 and 66.61 %, respectively. Hence, the proposed method is superior to Mora's scheme in term of the performance of time reduction.

Besides the depth video processing, the fast encoding methods influence the encoding bitrate. Let BR_{ori}, BR_{MORA}, and BR_{proposed} be the bitrate of depth video using HTM10.0, Mora's scheme and the proposed algorithm, the bitrate variation of depth video ΔBR_i (i = {MORA, proposed}) is computed by

$$\Delta BR_i = \frac{BR_i - BR_{\text{ori}}}{BR_{\text{ori}}} \times 100\%. \tag{10}$$

Table 5 PSNR and MS-SSIM results of the virtual view

Sequences	QP	PSNR (dB)			MS-SSIM		
		Original	MORA	Proposed	Original	MORA	Proposed
Balloons	25	35.75	35.72	35.73	0.9876	0.9876	0.9876
	30	35.24	35.17	35.23	0.9846	0.9846	0.9846
	35	34.34	34.29	34.32	0.9786	0.9785	0.9786
	40	32.89	32.86	32.90	0.9661	0.9661	0.9661
Kendo	25	38.23	38.21	38.14	0.9900	0.9900	0.9899
	30	37.43	37.41	37.36	0.9875	0.9875	0.9874
	35	36.22	36.20	36.17	0.9831	0.983	0.983
	40	34.43	34.44	34.40	0.9748	0.9748	0.9748
Newspaper	25	31.86	31.85	31.84	0.9771	0.977	0.977
	30	31.71	31.71	31.65	0.9744	0.9743	0.9744
	35	31.32	31.33	31.23	0.9688	0.9689	0.9688
	40	30.59	30.56	30.52	0.9583	0.9581	0.9581
Poznan_Street	25	35.47	35.49	35.49	0.9798	0.9798	0.9794
	30	34.94	34.96	34.92	0.9726	0.9726	0.9722
	35	34.03	34.03	33.96	0.9611	0.9611	0.9606
	40	32.64	32.61	32.57	0.9441	0.9439	0.9435

Table 4 lists the bitrate variation of depth video for Mora's scheme and the proposed algorithm which negative sign means bitrate saving. The average bitrate reduction of Mora's scheme is 17.2 % while the proposed method is 24.07 %. If we count up the bitrate of color video, the overall bitrate reduction of Mora's scheme and the proposed method is about 3.05 and 4.29 %, respectively.

In Mora's scheme, CU depth limitation and prediction mode pre-decision are utilized. The depth video bitrate reduction in Mora et al.'s method mainly caused by the cost of transmitting split flags and partition sizes. The bitrate reduction of the proposed method is mainly contributed by the depth video processing. The depth video processing enhances the correlation and makes the depth video more smooth and consistent. In addition, the fast coding method of the proposed method mainly reduces the coding complexity in smoothing regions. Hence, both the encoding compression ratio and speed are improved.

In the experiment, the virtual views were rendered with the reconstructed color and depth videos. We evaluated the objective quality of the rendered virtual views using PSNR and the multi-scale structural similarity index (MS-SSIM), which is calculated using rendered views and real views [36]. Table 5 presents the PSNR and MS-SSIM of the original HTM10.0, Mora et al.'s scheme, and the proposed algorithm. Because the proposed method includes both processing and encoding, the PSNR and MS-SSIM are slightly affected, and both methods have almost the same quality as the original view. MS-SSIM is an approximation of the human-perceived image quality. Thus, we use the Bjontegaard delta MS-SSIM (BD-MS-SSIM) and Bjontegaard delta bitrate (BDBR) to estimate the coding performance [37], where BDBR is calculated using the overall coding bitrate and MS-SSIM. The results are listed in Table 6. Clearly, the BD-MS-SSIM of the proposed algorithm in most test sequences is higher than Mora et al.'s scheme, and its BDBR is lower than Mora et al.'s.

5 Conclusions

In this paper, we proposed a joint processing and fast depth video coding algorithm that takes into account the depth processing and depth video feature information which include DRs, ERs, and MRs. Because the depth videos captured by mainstream technologies are inaccurate and inconsistent, the proposed processing method improves the consistency of depth video in the spatial and temporal domains. The fast coding method is based on processed depth video and statistical analysis of prediction modes. Experimental results show that the proposed algorithm reduces the coding time and depth video coding bitrate while it maintains the quality of the rendered virtual view.

Table 6 BD-MS-SSIM and BDBR comparison of MORA's scheme and the proposed algorithm

Sequences	BD-MS-SSIM		BDBR	
	MORA	Proposed	MORA (%)	Proposed (%)
Balloons	0.0001	0.0002	−1.22	−1.96
Kendo	0.0003	0.0004	−0.69	−0.79
Newspaper	0.0001	0.0001	−3.47	−3.76
Poznan_Street	0.0002	0.0000	−1.77	−0.09
average	0.0002	0.0002	−1.79	−1.65

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 61620106012, Grant 61271270 and Grant U1301257, National High-tech R&D Program of China (863 Program, 2015AA015901), Natural Science Foundation of Zhejiang Province (LY16F010002, LY15F010005), and Natural Science Foundation of Ningbo (2015A610127, 2015A610124). It is also sponsored by K.C. Wong Magna Fund in Ningbo University.

Authors' contributions

ZP designed the proposed algorithm and drafted the manuscipt. HH tested the proposed algorithm. FC carried out the depth video processing studies. GJ participated in the algorithm design. MY performed the statistical analysis. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Received: 10 October 2015 Accepted: 16 August 2016
Published online: 01 September 2016

References

1. M Tanimoto, FTV: free-viewpoint television. Signal Process. Image Commun. **27**(6), 555–570 (2012)
2. A Aggoun, E Tsekleves, MR Swash, Immersive 3D holoscopic video system. IEEE Trans. Multimedia **20**(1), 28–37 (2013)
3. C Fehn, *Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV* (SPIE Conference on Stereoscopic Display and Virtual Reality System XI, San Jose, 2004), pp. 93–104
4. K Müller, P Merkle, G Tech et al., *3D video formats and coding methods* (IEEE International Conference on Image Processing, Hong Kong, 2010), pp. 2389–2392
5. A De Abreu, P Frossard, F Pereira et al., Optimizing multiview video plus depth prediction structures for interactive multiview video streaming. IEEE J. Sel. Top. Sign. Proces. **9**(3), 487–500 (2015)
6. G Cheung, A Kubota, A Ortega, *Sparse representation of depth maps for efficient transform coding* (Picture Coding Symposium, Nagoya, Japan, 2010), pp. 298–301
7. J Lei, S Li, C Zhu et al., Depth coding based on depth-texture motion and structure similarities. IEEE Trans. Circuits Syst. Video Technol. **25**(2), 275–286 (2015)
8. S Liu, P Lai, D Tian et al., New depth coding techniques with utilization of corresponding video. IEEE Trans. Broadcast. **57**(2), 551–561 (2011)
9. M-K Kang, Y-S Ho, Depth video coding using adaptive geometry based intra prediction for 3D video systems. IEEE Trans. Multimedia **14**(1), 121–128 (2011)
10. K-J Oh, A Vetro, Y-S Ho, Depth coding using a boundary reconstruction filter for 3-d video systems. IEEE Trans. Circuits Syst. Video Technol. **21**(3), 350–359 (2011)
11. Y Zhang, S Kwong, L Xu et al., Regional bit allocation and rate distortion optimization for multiview depth video coding with view synthesis distortion model. IEEE Trans. Image Process. **22**(9), 3497–3511 (2013)
12. F Shao, W Lin, G Jiang, M Yu et al., Depth map coding for view synthesis based on distortion analyses. IEEE J. Emerging Sel. Top. Circuits Syst. **4**(1), 106–117 (2014)

13. K Muller, H Schwarz, D Marpe et al., 3D High-efficiency video coding for multi-view video and depth data. IEEE Trans. Image Process. **22**(9), 3366–3378 (2013)
14. J Smisek, M Jancosek, T Pajdla, *3D with Kinect* (IEEE International Conference on Computer Vision Workshops, Barcelona, 2011), pp. 1154–1160
15. S Foix, G Alenyà, C Torras, Lock-in time-of-flight (ToF) cameras: a survey. IEEE Sensors J. **11**(9), 1917–1926 (2011)
16. ISO/IEC JTC1/SC29/WG11, M16923, *Depth estimation reference software (DERS) 5.0* (Xian, China, 2009)
17. W Hu, X Li, G Cheung et al., *Depth map denoising using graph-based transform and group sparsity, IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, 2013, pp. 1–6
18. V-A Nguyen, D Min, MN Do, Efficient techniques for depth video compression using weighted mode filtering. IEEE Trans. Circuits Syst. Video Technol. **23**(2), 189–202 (2013)
19. Y Zhao, C Zhu, Z Chen et al., Depth no-synthesis-error model for view synthesis in 3D video. IEEE Trans. Image Process. **20**(8), 2221–2228 (2011)
20. J Lei, C Zhang, Y Fang et al., Depth sensation enhancement for multiple virtual view rendering. IEEE Trans. Multimedia **17**(4), 457–469 (2015)
21. DVSX De Silva, E Ekmekcioglu, WAC Fernando et al., Display dependent preprocessing of depth maps based on just noticeable depth difference modeling. IEEE J. Sel. Top. Sign. Proces. **5**(2), 335–351 (2011)
22. Z Peng, G Jiang, M Yu et al., Temporal pixel classification and smoothing for higher depth video compression performance. IEEE Trans. Consum. Electron. **57**(4), 1815–1822 (2011)
23. Z Peng, F Chen, G Jiang et al., Depth video spatial and temporal correlation enhancement algorithm based on just noticeable rendering distortion model. J. Vis. Commun. Image Represent. **33**(11), 309–322 (2015)
24. GJ Sullivan, JR Ohm, WJ Han et al., Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)
25. Lee, J.-H., Goswami, K., Kim, B.-G., et al. Fast encoding algorithm for high-efficiency video coding (HEVC) system based on spatio-temporal correlation. J. Real-Time Image Proc. **12**(2), 407–418 (2016).
26. Ahn, Y.-J., Sim, D. Square-type-first inter-CU tree search algorithm for acceleration of HEVC encoder. J. Real-Time Image Proc. **12**(2), 419–432 (2016).
27. Q Zhang, P An, Y Zhang et al., Low complexity multi-view video plus depth coding. IEEE Trans. Consum. Electron. **57**(4), 1857–1865 (2011)
28. L Shen, Z Zhang, Z Liu, Inter mode selection for depth map coding in 3D video. IEEE Trans. Consum. Electron. **58**(3), 926–931 (2012)
29. JY Lee, HC Wey, DS Park, A fast and efficient multi-view depth image coding method based on temporal and inter-view correlations of texture images. IEEE Transactions on Circuits and Systems for Video Technology **21**(12), 1859–1868 (2011)
30. C-S Park, Edge-based intra mode selection for depth-map coding in 3D-HEVC. IEEE Trans. Image Process. **24**(1), 155–162 (2015)
31. EG MORA, J Jung, M Cagnazzo et al., Initialization, limitation and predictive coding of the depth and texture quardtree in 3D-HEVC. IEEE Trans. Circuits Syst. Video Technol. **24**(9), 1554–1565 (2014)
32. SH Tsang, YL Chan, WC Siu, Efficient intra prediction algorithm for smooth regions in depth coding. Electron. Lett. **48**(18), 1117–1119 (2012)
33. Y Wang, Z Peng, G Jiang et al., Fast mode decision for depth video coding based on depth segmentation. KSII Trans. Internet Inf. Syst. **6**(4), 1128–1139 (2012)
34. JCT-3V, H1003, *Test model 8 of 3D-HEVC and MV-HEVC* (Geneva, 2014)
35. JCT-3V, F1100, *Common test conditions of 3dv core experiments* (Valencia, ES, 2013)
36. P Hanhart, E Bosc, P Le Callet et al., *Free-viewpoint video sequences: a new challenge for objective quality metrics, IEEE 16th International Workshop on Multimedia Signal Processing (MMSP)*, 2014, pp. 22–24
37. G Bjontegaard, *Calculation of average PSNR differences between RD-curves* (Video Coding Experts Group, Austin, 2001)