https://doi.org/10.1016/j.jvcir.2016.11.012

Segment–based view synthesis optimization scheme in 3D-HEVC

Huan Dou[1, 2], *Yui-lam Chan*[2, *], *Ke-bin Jia*[1], *Wan-Chi Siu*[2]

[1] *College of Electronic Information and Control Engineering*
Beijing University of Technology, Beijing, China

[2] *Department of Electronic and Information Engineering*
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

E-mail: douhuan88@gmail.com; enylchan@polyu.edu.hk; kebinj@bjut.edu.cn; enwcsiu@polyu.edu.hk

Abstract

The 3D extension of high efficiency video coding (3D-HEVC) adopts a view synthesis optimization (VSO) technique to improve the quality of synthesized views for depth map coding. The exact synthesized view distortion change (SVDC) is calculated in VSO which in turn brings huge coding complexity to the 3D-HEVC encoder due to the real view synthesis process. This work presents a scheme aimed at reducing coding complexity of the SVDC calculating process in the 3D-HEVC encoder. It skips line segments of pixels with variable lengths based on information from both of the textures and depth maps in the SVDC calculation. Experimental results demonstrate that the proposed scheme can reduce the coding complexity without any significant loss in rate distortion performance for the synthesized views.

Keywords

3D-HEVC; 3D video coding; depth coding; view synthesis optimization (VSO); synthesized view distortion change (SVDC); early skip mode

1 Introduction

With the rapid development of multimedia information technology, 3D video applications have broad development prospects in both film and home entertainment industries. Using a special display which re-projects 3D scenes from slightly different directions to left and right eyes, audiences can realize depth perception by the brain. It results in greatly improving the immersive feelings in comparison to 2D video [1]. The most widely used 3D display adopts the stereoscopic video that provides two views of videos for audiences. However, a multi-view display such as the auto-stereoscopic display and the free-view television has attracted much more attentions recently due to its comfortable and free viewing experience [2]. Nevertheless, the multi-view display requires more than two views of videos referred to as multi-view video (MVV) [3]. However, the bitrate of MVV is linearly proportional to the number of transmitted views. Although the multi-view coding (MVC) [3] technology in H.264/AVC can significantly compress these videos by exploiting the correlation between different views, the achievable compression is not sufficient to overcome constraints on channel bandwidth and in turn places a heavy burden on transmission and storage because of the huge amount of video data [4, 5]. Therefore, an efficient 3D video coding scheme is desired to solve this problem.

Owing to limitations in the production environment and bandwidth constraints, only limited views (for example, two to three views) and their corresponding depth maps are captured, compressed and transmitted for 3D video applications [6, 7], and this 3D video format is called multi-view video plus depth (MVD) [8, 9]. By means of depth information, it can synthesize an arbitrary number of intermediate views from these limited views via the depth-image-based rendering (DIBR) technology [10, 11] at decoder side based on the camera parameters and the original views' positions. This new MVD format can decouple content creation from display so that it supports any type of displays and adapts to specific technical conditions and user preferences. Based on this property, the advanced MVD format can achieve efficient free viewpoint video services without transmitting all desired views, and requires less data burden compared to the MVV format. MVD has been adopted by the state-of-the-art High Efficiency Video Coding (HEVC) standard [12]. Recently, the Joint Collaborative Team on 3D Video Coding Extension (JCT-3V), formed by the Moving Picture Experts Group (MPEG) and the Joint Collaborative Team on 3D

Video Coding (JCT-3V), has been developing a new generation of an international 3D video coding standard, namely 3D-HEVC [1]. Similar to the MVC extension of H.264/AVC, 3D-HEVC takes advantage of the inter-view prediction [13] to exploit the correlations among multiple views. An extended technique to inter-view prediction is to design efficient coding tools for jointly coding texture videos and depth maps.

Depth maps have distinctive characteristics compared to texture videos. For instance, the depth maps have mostly smooth regions delimited by sharp edges [9], and the distortion of sharp edges induces ringing artifacts at object boundaries in synthesized views [14]. To account for these characteristics of depth maps, specific edge preserving coding tools such as depth modeling modes [15], segment-wise DC coding [16], etc. have been adopted in depth map coding of 3D-HEVC. In addition, depth maps are transmitted for view synthesis instead of directly viewed by audiences. If the conventional distortion metrics such as the sum of squared differences (SSD) or sum of absolute differences (SAD) is employed, the reconstructed depth map can be obtained in terms of its own image quality. However, the quality of the reconstructed depth map cannot guarantee the synthesized view quality. Consequently, 3D-HEVC replaces the original rate distortion optimization (RDO) for the mode decision process in depth map coding by the synthesized view distortion [15]. It is referred to as view synthesis optimization (VSO), which takes both of the synthesized view and depth map quality into consideration. The distortion measure [8, 17], D_{VSO}, is composed of the distortions in the synthesized view, D_s, and depth map, D_d, expressed as

$$D_{VSO} = \left(w_s \times D_s + w_d \times D_d\right) / \left(w_d + w_s\right) \tag{1}$$

where w_s and w_d are the associated weighting factors of D_s and D_d, respectively. D_d is calculated by the sum of squared difference (SSD) or sum of absolute Hadamard transform difference (SATD). The calculation of D_s includes the most complex view synthesis process in 3D-HEVC in order to measure the synthesized view distortion for determining the best modes and coding partition. Adopting VSO in 3D-HEVC can guarantee the best coding performance in synthesized views, but it imposes a heavy computational burden to the encoder.

Many researchers have been devoted their efforts on reducing the complexity of VSO [18-27]. Since distortions in synthesized views are caused by the depth map distortions due to the encoding process, some algorithms were proposed by deriving relationships between distortions in the coded depth map and rendered virtual view [18-21]. The well-known technique is called as view synthesis distortion (VSD) [19], which is accepted as part of the 3D-HEVC reference software HTM-16.0 [17]. It analyzes the relationship between depth quality and rendering quality, and weights the depth distortion with the sum of absolute horizontal gradients of the co-located texture. VSD estimates synthesized views without actual view synthesis in order to save time. Based on the fact that different depth distortion values may be projected to the same geometrical distortion in the synthesized virtual view, the scheme in [22] proposed an allowable depth distortion model with piecewise function to establish a new rate distortion model for mode decision and motion/disparity estimation by minimizing the view synthesis distortion at given bitrates. The work in [23] considers video-coding-induced distortion, depth-quantization-induced distortion and inherent geometry distortion together to characterize the view synthesis quality. In order to provide a better trade-off between compression efficiency and coding complexity, the algorithm proposed in [24] classifies the pixels in depth map into occluded pixels, interpolated pixels, and disoccluded pixels by the view synthesis process. Various models are then applied to compute the distortions adaptively. However, these algorithms [18-24] estimate the distortion in synthesized views using mathematical models or metrics. Their results are approximate values and may not represent the accurate synthesized view distortion. In order to calculate the exact synthesized view distortion, the synthesized view distortion change (SVDC) method was proposed to measure the change of the distortion in a synthesized view based on the change of the depth data [15, 25] and is also adopted in the HTM-16.0. Since SVDC needs to bring the rendering functionalities into the encoder process, the complexity and encoding time increases significantly. To reduce the complexity, a number of fast algorithms were proposed to skip some pixels' rendering process for the SVDC method based on the measure of zero-synthesized view difference [26-27]. However, it still remains room for further encoding complexity reduction. This paper proposes an adaptive segment-based skipping (ASS) algorithm for VSO in order to deal with some small line segments of pixels which will not cause distortion of the synthesized view during the SVDC process. With the proposed ASS algorithm, the VSO processing time can be further reduced.

The remainder of this paper is organized as follows. Section 2 briefly reviews the SVDC calculation and the early skip mode. Section 3 particularly presents our proposed adaptive segment-based skipping method for VSO in 3D-HEVC. Experimental results and analyses are provided in Section 4, and conclusion is drawn in Section 5.

2 SVDC calculation and early skip mode

To acquire a precise distortion measure for VSO, SVDC, defined as the change in distortion between two synthesized textures, is computed at the encoder as depicted in Fig. 1.

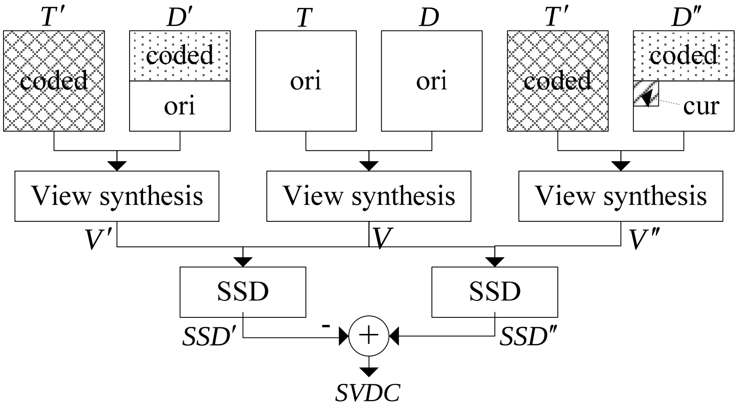

Fig.1 SVDC calculation.

In Fig. 1, V is the synthesized reference frame rendered by the original texture frame T and the original depth frame D. V' and V'' are two synthesized frames rendered from the reconstructed texture frame T' and the two partially encoded depth frames, D' and D'', respectively. The difference between the two depth frames is that D'' of the current block contains the distorted depth data due to the actual mode under test. SVDC is finally measured by the difference between the two sum of squared distortions (SSD) between V' and V'', denoted by SSD' and SSD'', and is computed as

$$SVDC = SSD'' - SSD' = \sum [V'' - V]^2 - \sum [V' - V]^2 \tag{2}$$

The detailed process of the SVDC calculation is described as follows. Prior to encoding the current frame, the original texture frame T and the original depth frame D are used to render the synthesized frame V as the reference. D' is formed by the reconstructed depth of the already encoded blocks, the original depth for the current depth block and the original depth for the remaining depth blocks, as shown in Fig. 1, and is used to render V' with T'. When a block is under evaluation, all possible coding modes use their corresponding distorted depth data in the block to get different D''. Each D'' is then used to render V'' with T'. Finally, SVDC can be calculated by V, V' and V'', according to Eq. (2). The best coding mode will be determined after the SVDC calculation. This view rendering process is performed recursively during the encoding process and in turn brings large computing complexity for the encoder.

For view synthesis, it is understood that distortions in depth maps may not always induce geometry changes in synthesized views. From this observation, the current implementation of the HTM-16.0 adopts block-based and line-based early skip (ES) decisions to reduce the processing time of the SVDC calculation. They are based on the principle of zero disparity distortion in which the original and distorted depth pixels are mapped to the same disparity vector. If this is the case, the distortion D_s of this pixel in Eq. (1) is also equal to zero and the SVDC calculation can be skipped. The block-based condition for ES mode decision is checked according to the disparity distortion for the entire block. If all the pixels in the block satisfy the zero disparity distortion, the SVDC calculation of the whole block is skipped. If the block is not skipped, each line of the block is checked whether all the pixels within the line satisfy the zero disparity distortion. In this situation, the SVDC calculation is skipped for those lines that fulfill the line-based early skip condition.

3 The proposed Adaptive segment-based skipping (ASS) algorithm

When computing SVDC for the current block in the original VSO process, every pixel in this block is warped, interpolated and blended to render a synthesized view. By doing so, large amounts of pixels need to be rendered because of the high resolution of 3D video. This pixel by pixel rendering process brings huge computational complexity. The aforementioned ES methods can skip some unnecessary rendering process. Nevertheless, the coding complexity is still tremendous especially for calculating SSD'' in Eq. (2) since various modes under evaluation lead to different prediction and reconstruction signals. While the complexity reduction of ES is important, alone it is not sufficient to have an efficient 3D-HEVC implementation. Table 1 lists out the line skip rate of the ES mode for calculating SSD''. Noted that if the whole block is skipped, the skip rate in Table 1 considers that all lines in this block are skipped. 100 frames were tested for all sequences ("Balloons", "Kendo", "GTFly", "PoznanSteet", "PoznanHall2" and "Shark") with 4 depth quantization parameters (QPs) set to {34, 39, 42, 45}. From Table 1, it can be seen that as QP value increases, the line skip rates of all sequences drop down due to the decrease in the quality. For sequences with many unsmooth regions (such as "Balloons" and "Kendo") or fast motion contents (such as "PoznanStreet") in depth maps, the skip rates of ES are lower than those of other sequences.

Table 1 Line skip rates for calculating SSD''.

QP	Balloons	Kendo	GTFly	PoznanStreet	PoznanHall2	Shark
34	25.13%	28.26%	42.02%	17.44%	32.76%	31.23%
39	23.86%	26.72%	35.81%	16.29%	28.33%	29.27%
42	21.19%	23.30%	29.91%	13.29%	22.69%	25.96%
45	19.09%	20.78%	26.09%	12.15%	17.36%	21.98%
Average	22.32%	24.77%	33.46%	14.79%	25.29%	27.11%

In HEVC, it supports larger coding blocks to achieve better compression for high-resolution videos. However, larger coding blocks have lower possibility to meet the conditions of block-based and line-based ES modes than smaller coding blocks since the zero disparity distortion for all pixels in the line/block is less likely in the large coding block. It is expected that ES modes for large coding blocks may not as efficient as small coding blocks. To verify this observation, we fixed the CU size to 64×64, 32×32 and 16×16, and measured the line skip rates for CUs with different sizes, as shown in Table 2. It can be found that for all the tested sequences using different QPs, the line skip rate for CU with the size of 16×16 is larger than that for CU with the size of 32×32, and the line skip rate for CU with the size of 32×32 is also larger than that for CU size of 64×64. It can be concluded that the ES utilization deceases as CU size increases. This is reasonable since less number of entire lines inside a small CU are likely to have zero disparity distortion. It indicates that ES is not very useful for large coding blocks.

Table 2 Line skip rates with different CU size for calculating SSD''.

Sequence	Width of CU	QP				Average
		34	39	42	45	
Balloons	64	14.31%	13.81%	12.32%	11.38%	12.96%
	32	27.83%	27.05%	25.15%	23.74%	25.94%
	16	37.46%	32.89%	30.47%	23.72%	31.14%
Kendo	64	17.15%	17.11%	15.35%	14.41%	16.01%
	32	31.83%	30.65%	28.04%	26.90%	29.36%
	16	39.46%	34.87%	32.85%	32.46%	34.91%
GTFly	64	34.19%	30.66%	22.47%	18.86%	26.55%
	32	40.98%	35.11%	30.94%	28.08%	33.78%
	16	47.00%	40.97%	37.43%	35.67%	40.27%
PoznanStreet	64	7.53%	7.67%	5.88%	4.59%	6.42%
	32	16.08%	14.36%	13.25%	11.63%	13.83%
	16	24.33%	20.96%	19.53%	19.41%	21.06%
PoznanHall2	64	20.03%	16.65%	14.04%	12.46%	15.80%
	32	31.55%	27.17%	24.25%	20.69%	25.91%
	16	34.31%	29.44%	25.93%	23.20%	28.22%
Shark	64	18.81%	17.61%	15.07%	13.97%	16.37%
	32	33.99%	30.64%	28.02%	24.93%	29.39%
	16	43.19%	36.35%	32.97%	30.16%	35.67%

To tackle the low ES utilization for large CUs, it is not necessary to restrict the zero disparity distortion to an entire line. To make the skipping more flexible, our proposed adaptive segment-based skipping (ASS) algorithm is to study the undistorted depth data for a line segment. If a short line segment that may not cause distortion in a synthesized view can be identified and skipped, the computational complexity of the SVDC calculation can be further reduced. Based on this idea, the proposed ASS picks out the segment of pixels based on the distortions due to texture coding and depth map coding in which both of them give impact on the synthesized view quality. In ASS, except the depth distortion considered by early skip mode, the criterion used to determine the line segments also takes texture distortion into consideration.

3.1 depth distortion criterion

During the view synthesis process, data from depth maps are converted into disparity vectors with the help of camera parameters. These disparity vectors are used to warp pixels from the original view to the synthesized view. The disparity vector *d* can be derived from the real depth value, z, as follows:

$$d = \frac{f \cdot B}{z} \tag{3}$$

where *f* and *B* denote the focal length of the camera, and the baseline distance between two views, respectively. The actual distance of the object in the 3D space, *z*, can be expressed as

$$z = 1 \Big/ \left[\frac{I}{255} \cdot \left(\frac{1}{z_{near}} - \frac{1}{z_{far}} \right) + \frac{1}{z_{far}} \right] \tag{4}$$

where z_{near} and z_{far} are the nearest and the farthest depth values, respectively, which are recorded in the camera configure files of the test sequences recommended by the ISO/IEC and ITU-T JCT-3V group. *I* is a depth value quantized with 8-bit per sample, which gives a range between 0 and 255.

With Eq. (3) and Eq. (4), the disparity vector can be obtained from the depth value as follows:

$$d = \frac{f \cdot B \cdot I}{255} \cdot \left(\frac{1}{z_{near}} - \frac{1}{z_{far}} \right) + \frac{f \cdot B}{z_{far}} \tag{5}$$

Since a depth value I is only used to calculate a disparity vector d, distortion in I only leads to a position error in synthesized views. d is rounded to ¼ sub-pixel position due to the adoption of quarter-accuracy interpolation in the HTM-16.0. In Eq. (5), several I values may map to the same sub-pixel disparity vector in the view synthesis process such that no position error exists in the synthesized view. If a segment of distorted and rounded disparity vectors derived by the encoded depth values remain the same as the disparity vectors derived by the original depth values, the distortion of this depth segment can be neglected and the SVDC calculation process can be skipped. We then define S_D to be the depth segment with zero disparity distortion in which all distorted disparity vectors, d_i', are equal to the original disparity vector, d_i, as

$$d_i = d_i', i \in S_D \tag{6}$$

where i denotes a pixel position in S_D.

3.2 Texture smoothness criterion

Even though distortion in depth value will induce to a pixel being warped to an incorrect position in the synthesized view resulting in position or geometry errors, it is not the necessary condition for the appearance of distortion in the synthesized view. Fig. 2 illustrates two examples of a position error due to depth distortion. In this figure, u and v are two pixels from the base view. The solid arrows are the correct disparity vectors calculated from the original depth value and the dotted arrows are the distorted disparity vectors calculated from the distorted depth value after depth map coding. u_1 and v_1 are correct rendering positions, while u_2 and v_2 are distorted rendering positions. Since u in the example of Fig. 2 is located at a smooth area, the distorted pixel position u_2 in the synthesized view has the similar pixel value as u_1. On the contrary, since v is located at a complex texture area, the distorted disparity will cause enormous rendering error. That is to say, if a texture segment belongs to a texture smooth area, the position error may not cause distortion in the synthesized view, and the SVDC calculation process can be avoided.

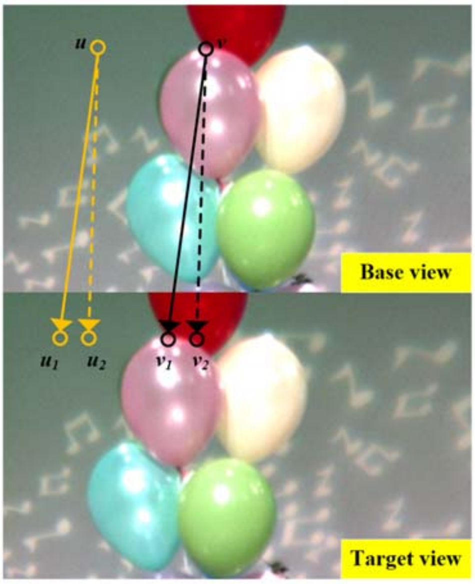

Fig. 2 Warping process during rendering.

Based on this observation, let us define the texture smoothness criterion of a pixel pair along the horizontal line, p_i and $p_{i+1.}$ It checks the pixel luminance difference between two neighboring pixels from the corresponding reconstructed texture frame as

$$\left|p_i - p_{i+1}\right| \leq Th \tag{7}$$

In Eq. (7), the threshold, *Th*, is used to judge the texture smoothness. If the pixel pair in a texture region satisfies the above criterion, the two pixels in the pixel pair belong to a texture smooth region. A segment of pixels is then defined as the texture smooth segment, S_T, in which all pixel pairs fulfill the criterion in Eq. (7).

In order to precisely find the texture smooth segment, *Th* should be determined carefully. In 3D-HEVC, it utilizes an intra prediction tool to exploit the spatial correlation within each frame. Therefore, intra prediction can accurately characterize the texture regularity of a frame by using prediction modes. An Intra DC mode uses an average value of reference samples for intra prediction [28]. In other words, if a block chooses an Intra DC mode as its best mode, it is reasonable to consider it as a smooth block. In addition, it is noted that the texture video in 3D-HEVC is encoded before the depth video. In the determination of the texture smoothness, the threshold *Th* is thus set according to the image characteristics of these already coded smooth blocks in the texture frame. Therefore, the threshold *Th* of the texture smoothness criterion can be derived after the coding of the texture frame and before the coding of the depth frame, and the process is described in the following steps:

Step 1: All blocks of $N_w \times N_h$ size in the texture frame using the Intra DC mode as their best mode are collected.

Step 2: The luminance difference of the reconstructed texture frame at pixel location (i, j) in the block between two horizontal neighboring pixels, $p_{i,j}$ and $p_{i+1,j}$, is calculated. For j^{th} line, the sum of absolute line difference ($SALD_j$) is computed as follows

$$SALD_j = \frac{\sum_{i=0}^{N_w-1} \left| p_{i,j} - p_{i+1,j} \right|}{N_w - 1} \tag{8}$$

Step 3: By taking the average of ($SALD_j$) for all lines in the block coded using the Intra DC mode, the average horizontal luminance difference of the block, $SALD_{DC}$, is calculated as

$$SALD_{DC} = \frac{\sum_{j=0}^{N_h} SALD_j}{N_h} \tag{9}$$

Step 4: The threshold *Th* for the texture smoothness is then computed by averaging all $SALD_{DC}$ in the texture frame as

$$Th = \frac{\sum DALD_{DC}}{IntraDCBlockNum} \tag{10}$$

where *IntraDCBlockNum* denotes the number of blocks coded using the Intra DC mode in the texture frame. Noted that *Th* is updated for each I-frame in the texture video. Therefore, depth frames from the same intra period share the same *Th* as their threshold for the texture smoothness criterion.

3.3 Final Segment determination

Based on the above texture smoothness and depth distortion criteria, the proposed ASS skips eligible segments which may not cause distortions in synthesized views to reduce the computational burden for SVDC in VSO. The procedure of eligible segment determination in ASS is shown in Fig. 3. For a line in the current block being coded, the texture smoothness criterion is used to locate S_T which has small variation of pixel values, and the length of S_T is recorded as L_T, as shown in Fig. 3. Simultaneously, the depth distortion criterion is employed to find another line segment S_D where all the encoded depth data will not cause a position error in the synthesized view, and the length of S_D is L_D, as depicted in Fig. 3. Finally, the line segment that has the longer length is selected as the final skipped segment in ASS, and its length is defined as L_F in Fig. 3. After then, it moves to the pixel next to the skipped segment to recur the above procedure until the current processing pixel is not eligible in the proposed ASS algorithm.

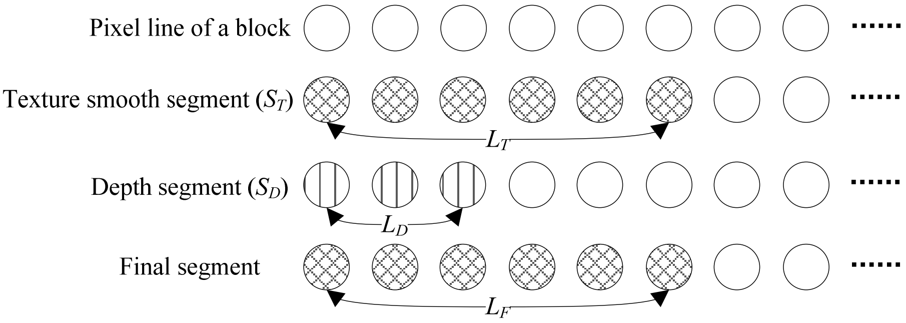

Fig. 3 Segment determination in ASS.

3.4 Flowchart of the proposed ASS

The flow chart of the proposed ASS algorithm is illustrated in Fig. 4. Our proposed ASS is integrated into both of the block-based and line-based ES techniques. Given a PU with the size of $N \times N$, and a tested mode, the block-based and line-based ES decisions are checked first. If the PU in this mode is satisfied with the condition of block-based ES, the SVDC calculation for whole block is skipped. Otherwise, the line-based ES decides whether the SVDC calculation for each line is necessary or not. If a line of pixels is not early skipped, the two criteria described in Sub-sections 3.1 and 3.3 are applied to determine the line segments, S_T and S_D, with the length of L_T and L_D which may not cause distortions in the synthesized view. L_F is then determined by maximizing L_T and L_D. The rendering processes of those line segments are skipped to reduce the SVDC calculation time during depth map coding. In order to keep the efficiency of the proposed ASS, the line segments are skipped only if the length of the segment is larger or equal to 4. For those ineligible pixels that cannot satisfy with both block-based, line-based, and segment-based ES conditions, the conventional SVDC is used to calculate the distortions.

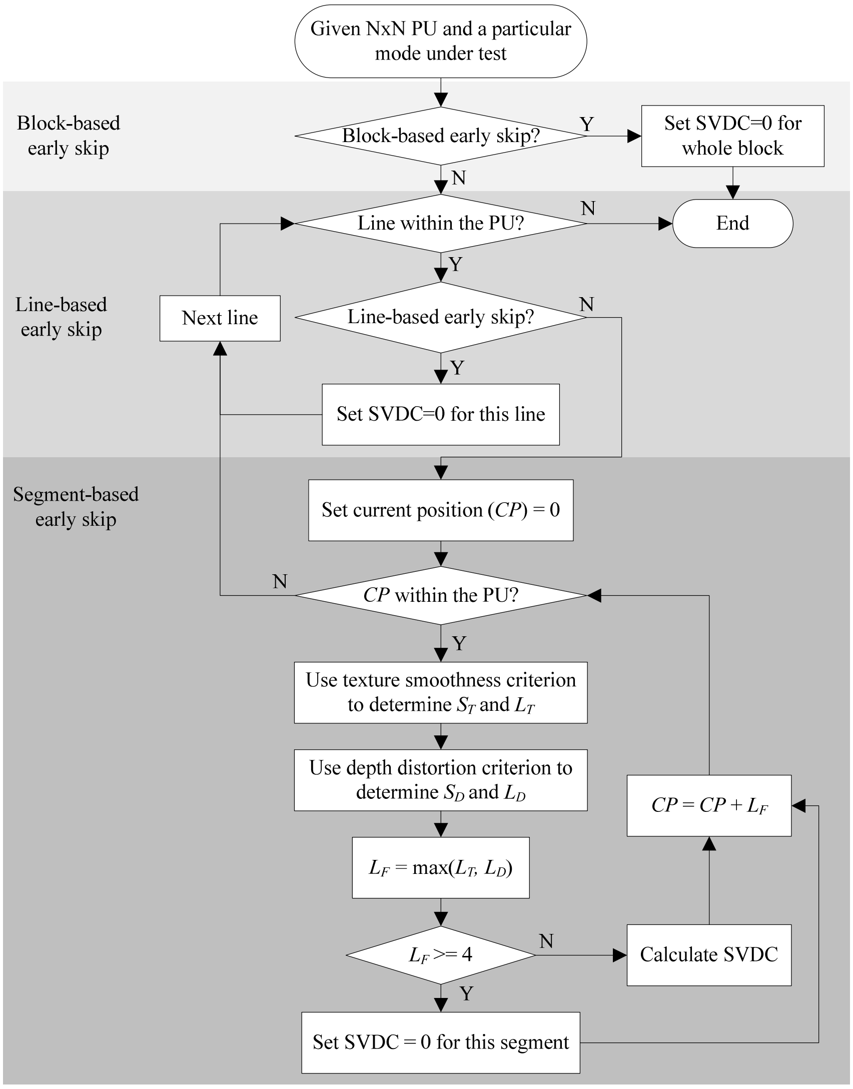

Fig. 4 Flowchart of the ASS algorithm.

4 Experimental results

To evaluate the performance, the proposed ASS has been built on the reference software HTM-16.0 of the 3D-HEVC standard. The proposed algorithm was compared with the HTM-16.0 which adopts the techniques in [25] and [26]. The format of three views plus depth was used in this experiment, and several sequences with two resolutions of 1920×1088 and 1024×768 recommended by the JCT-3V group were encoded. They include "Balloons" (1024×768), "Kendo" (1024×768), "GTFly" (1920×1088), "PoznanSteet" (1920×1088), "PoznanHall2" (1920×1088) and "Shark" (1920×1088). The quantization parameters (QPs) were set to {(25, 34), (30, 39), (35, 42) and (40, 45)} for coding texture videos and depth maps, respectively. Full-length frames were tested for each sequence under the common test conditions (CTC) specified in [29]. The GOP length was set to 8 and the intra period was set to 24. Three views, V_0, V_1, and V_2, for each sequence were coded and the step size of the synthesized view was set to 0.25 resulting in generating six synthesized views, $V_{0.25}$, $V_{0.5}$, $V_{0.75}$, $V_{1.25}$, $V_{1.5}$ and $V_{1.75}$, rendered at decoder side. The objective evaluation of synthesized view quality is calculated by ΔBDBR, which comes from Bjontegaard metrics [30] measuring the rate distortion performance of the synthesis view quality and the total bitrate including texture and depth videos. ΔT (%) represents depth coding time saving in percentage in comparison with the algorithm in HTM-16.0+[25]+[26]. Positive and negative values denote increments and decrements, respectively. The simulations were conducted on a 64-bit MS Windows 10 OS running on an Intel(R) Core(TM) i7-4790 CPU of 3.60 GHz and 32.0GB RAM.

Table 3 tabulates the skip rate of pixels of ASS for all QPs. In measuring this skip rate, the lines that fulfill block-based and line-based ES are excluded. It can be seen that the average skip rate of all test sequences is 73.22%, which significantly increases ES utilization. Table 3 also shows percentage of the skip rate that is affected by the depth distortion criterion ($L_D > L_T$), the texture smoothness criterion ($L_T > L_D$) and both criteria ($L_D = L_T$). For all sequences, the texture smoothness criterion shows greater impact than the depth distortion criterion. As QP increases, the influence on skip rate of the depth distortion criterion decreases and that of the texture smoothness criterion increases for all the testing sequences. This is because for lower QP, the depth coding result is more accurate and causes less depth distortion. Thus the influence of the depth distortion criterion in lower QP is more than that in higher QP situation. Moreover, since the reconstructed textures are smoother in larger QP, the influence of the texture smoothness criterion increases as the QP value increases.

Table 3 Skip rate of pixels in the proposed ASS.

Sequence	QP	Skip rate of pixels	$L_D > L_T$	$L_D < L_T$	$L_D = L_T$
Balloons	34	69.39%	15.67%	82.60%	1.73%
	39	70.81%	13.04%	85.78%	1.18%
	42	71.50%	10.36%	88.78%	0.86%
	45	70.45%	8.50%	90.82%	0.68%
Kendo	34	72.70%	10.82%	87.49%	1.69%
	39	74.31%	8.80%	90.12%	1.08%
	42	76.31%	6.82%	92.47%	0.71%
	45	76.46%	5.77%	93.70%	0.53%
GT_Fly	34	61.52%	16.87%	81.31%	1.83%
	39	74.26%	8.32%	90.53%	1.15%
	42	82.59%	4.63%	94.66%	0.72%
	45	83.22%	3.31%	96.23%	0.45%
Poznan_Hall2	34	61.04%	27.76%	71.19%	1.05%
	39	65.46%	20.99%	78.05%	0.96%
	42	70.11%	14.89%	84.31%	0.80%
	45	71.16%	10.70%	88.50%	0.80%
Poznan_Street	34	68.85%	19.58%	78.84%	1.57%
	39	67.62%	17.25%	81.66%	1.09%
	42	79.23%	8.89%	90.48%	0.63%
	45	76.67%	7.71%	91.77%	0.52%
Shark	34	73.96%	13.02%	84.84%	2.15%
	39	77.98%	8.87%	89.60%	1.54%
	42	79.26%	6.62%	92.36%	1.03%
	45	82.51%	4.27%	95.05%	0.67%
average		73.22%	11.39%	87.55%	1.06%

Table 4 then gives the comparison of depth coding time and ΔT of the proposed ASS. From Table 4, it can be found that ASS can achieve about 28.87% complexity reduction in average, which gets the benefit of the flexible segment in the SVDC calculation. It means the proposed ASS can successfully reduce the computational complexity of HTM-16.0+[25]+[26]. Basically, the depth coding time saving increases as the QP value becomes larger. It can be explained by using Table 3. In this Table 3, it can be seen that the skip rate affected by the texture smoothness criterion is generally greater than that affected by the depth distortion criterion. Since the texture smoothness criterion dominates and its influence on the skip rate becomes greater for larger QP, time saving is more remarkable as QP value increases.

Table 4 Depth coding time comparison of the proposed ASS compared with HTM-16.0+[25]+[26].

Sequence	QP	Depth coding time (s)		ΔT
		HTM-16.0+[25]+[26]	ASS	
Balloons	34	7310.141	5471.476	25.15%
	39	6066.451	4216.774	30.49%
	42	5414.575	3791.106	29.98%
	45	5147.194	3871.25	24.79%
Kendo	34	8539.806	5975.63	30.03%
	39	6636.737	4635.884	30.15%
	42	5919.974	4028.396	31.95%
	45	5430.704	3620.909	33.33%
GTFly	34	16895.93	14356.27	15.03%
	39	13102.01	10063.24	23.19%
	42	12107.7	8089.824	33.18%
	45	10991.04	6829.133	37.87%
PoznanHall2	34	8426.171	7047.016	16.37%
	39	7302.423	5519.947	24.41%
	42	7214.7	4997.733	30.73%
	45	7120.829	4807.676	32.48%
PoznanStreet	34	13805.97	10393.54	24.72%
	39	10908.72	8480.488	22.26%
	42	10620.13	7362.732	30.67%
	45	10537.48	7029.297	33.29%
Shark	34	21023.55	15752.19	25.07%
	39	18535.16	11489.53	38.01%
	42	13454.02	9350.175	30.50%
	45	12931.27	7863.787	39.19%
average				28.87%

Table 5 presents the BDBR performance of the proposed ASS compared to HTM-16.0+[25]+[26]. It shows the ΔBDBR of the base view (Base), the synthesized view (Syn.), and both of the base and synthesized view (All) for the proposed ASS. From Table 5, it can be seen that 2.2% ΔBDBR increase of the synthesized views and 1.5% ΔBDBR increase of both of the base and synthesized views for all the testing sequences in average are resulted from the proposed ASS compared to the HTM-16.0+[25]+[26]. The proposed ASS can achieve the better ΔBDBR for "Balloons" and "PoznanStreet" with 1.5% ΔBDBR increase in the synthesized views. Therefore, for all the testing sequences, the proposed ASS shows no significant ΔBDBR loss.

Table 5 ΔBDBR comparison of the proposed ASS compared with HTM-16.0+[25]+[26].

Sequence	ASS compared to HTM-16.0+[25]+[26] (%)		
	Base	Syn.	All
Balloons	0.0	1.5	0.9
Kendo	0.0	2.5	1.6
GTFly	0.0	2.0	1.4
PoznanHall2	0.0	3.0	2.2
PoznanStreet	0.0	1.5	1.0
Shark	0.0	2.5	1.7
Average	0.0	2.2	1.5

5 Conclusion

In this paper, an adaptive segment-based skipping algorithm for view synthesis optimization in 3D-HEVC has been proposed to reduce the computational complexity of depth map coding. The proposed algorithm releases the constraints of the zero disparity and texture smoothness in the line-based ES by identifying small line segments of depth pixels which may not cause distortions in synthesized views. Those skipped small segments are obtained based on both texture and depth information. The experimental results show that the proposed algorithm can further reduce the SVDC processing time while keeping the comparable ΔBDBR performance.

Acknowledgments

This work was supported by Central Research Grant of POLYU, China (Grant PolyU 5119/13E), Project for the National Natural Science Foundation of China (Grant No. 61672064), the Project for the key Project of Beijing Municipal Education Commission (Grant KZ201610005007), Beijing Postdoctoral Research Foundation (Grant 2015ZZ-23), China Postdoctoral Research Foundation (Grant 2015M580029, 2016T90022), and Computational Intelligence and Intelligent System of Beijing Key Laboratory Research Foundation (Grant 002000546615004).

References

[1] K. Müller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. H. Rhee, G. Tech, M. Winken, T. Wiegand, 3D high-efficiency video coding for multi-view video and depth data, IEEE Trans. Image Process. 22 (9) (2013) 3366–3378.

[2] L. Zhang, Y. Chen, M. Karczewicz. Disparity vector based advanced inter-view prediction in 3D-HEVC, in: IEEE Int. Symposium on Circuits and Syst. (ISCAS), 2013, pp. 1632-1635.

[3] A. Vetro, T. Wiegand, G. J. Sullivan, Overview of the stereo and multiview video coding extensions of the H. 264/MPEG-4 AVC standard, Proc. IEEE 99 (4) (2011) 626-642.

[4] Y. Chen, M. M. Hannuksela, T. Suzuk, S. Hattori, Overview of the MVC+ D 3D video coding standard, J. Vis. Commun. Image Represent. 25 (4) (2014) 679-688.

[5] Z. Peng, F. Chen, G. Jiang, M. Yu, F. Shao, Y. S. Ho, Depth video spatial and temporal correlation enhancement algorithm based on just noticeable rendering distortion model, J. Vis. Commun. Image Represent. 33 (2015) 309-322.

[6] A. Smolic, K. Müller, P. Merkle, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, R. Tanger, P. Kauff, T. Wiegand, T. Balogh, Z. Megyesi, and A. Barsi, Multi-view video plus depth (MVD) format for advanced 3D video systems, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q, Document JVT-W100, Apr. 2007.

[7] Call for proposals on 3D video coding technology, ISO/IEC JTC1/SC29/WG11, Document MPEG2011/N12036, Mar. 2011.

[8] P. Merkle, A. Smolic, K. Muller, T. Wiegand, Multi-view video plus depth representation and coding, in: IEEE Int. Conference on Image Process. (ICIP), 2007, pp. I-201-I-204.

[9] K. Muller, P. Merkle, T. Wiegand, 3-D video representation using depth maps, Proc. IEEE 99 (4) (2011) 643-656.

[10] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, T. Wiegand, Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems, in: IEEE Int. Conference on Image Process. (ICIP), 2008, pp. 2448-2451.

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

[12] G. J. Sullivan, J. R. Ohm, W. J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, IEEE Trans. Circ. Syst. Video Technol. 22 (12) (2012) 1649–1668.

[13] H. Schwarz, T. Wiegand, Inter-view prediction of motion data in multiview video coding, in: Proc. Picture Coding Symp. (PCS), 2012, pp. 101-104.

[14] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P. H. N. de With, T. Wiegand, The effects of multiview depth video compression on multiview rendering, Signal Process.: Image Commun. 24 (1) (2009) 73-88.

[15] K. Müller, P. Merkle, G. Tech, T. Wiegand, 3D video coding with depth modeling modes and view synthesis optimization, in: Asia-Pacific IEEE Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012, pp. 1-4.

[16] F. Jäger, Simplified depth map intra coding with an optional depth lookup table, in: Int. Conference on 3D Imaging (IC3D), 2012, pp. 1-4.

[17] https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-16.0 (accessed 2015).

[18] W. S. Kim, A. Ortega, P. L. Lai, D. Tian, C. Gomila, Depth map coding with distortion estimation of rendered view, in: Proc. SPIE7543, Visual Information Process. and Commun., 2010, pp. 75430B-75430B-10.

[19] B. T. Oh, K, J. Oh, View synthesis distortion estimation for AVC-and HEVC-compatible 3-D video coding, IEEE Trans. Circ. Syst. Video Technol. 24 (6) (2014) 1006-1015.

[20] R. Ma, N. M. Cheung, O. C. Au, D. Tian, Novel distortion metric for depth coding of 3D video, in: IEEE Int. Conference on Image Process. (ICIP), 2013, pp. 1714-1718.

[21] T. Zhang, X. Fan, D. Zhao, W. Gao, New distortion model for depth coding in 3DVC, in: IEEE Visual Communications and Image Processing (VCIP), 2012, pp. 1-6.

[22] Y. Zhang, S. Kwong, S. Hu, C. C. J. Kuo, Efficient multiview depth coding optimization based on allowable depth distortion in view synthesis, IEEE Trans. Image Process. 23 (11) (2014) 4879-4892.

[23] Y. Liu, Q. Huang, S. Ma, D Zhao, W. Gao, Joint video/depth rate allocation for 3D video coding based on view synthesis distortion model, Signal Process.: Image Commun. 24 (8) (2009) 666-681.

[24] C. Li, X. Jin, Q. Dai, A novel distortion model for depth coding in 3D-HEVC, in: IEEE Int. Conference on Image Process. (ICIP), 2014, pp. 3228-3232.

[25] G. Tech, H. Schwarz, K, Müller, T. Wiegand, 3D video coding using the synthesized view distortion change, in: Proc. Picture Coding Symp. (PCS), 2012, pp. 25-28.

[26] S. Ma, S. Wang, W. Gao, Low complexity adaptive view synthesis optimization in HEVC based 3D video coding, IEEE Trans. Multimedia 16 (1) (2014) 266-271.

[27] B. T. Oh, J. Lee, D. S. Park, G. Tech, K. Müller, T. Wiegand, S. Wang, S. Ma, H. Liu, J. Jia, J. Jung, 3D-CE8.h results on view synthesis optimization by Samsung, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Document JCT3V-A0093, July 2012.

[28] J. Lainema, F. Bossen, W. J. Han, J. Min, K. Ugur, Intra coding of the HEVC standard, IEEE Trans. Circ. Syst. Video Technol. 22(12) (2012) 1792-1801.

[29] K. Müller, A. Vetro, Common Test Conditions of 3DV Core Experiments, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Document JCT3V-G1100, Jan. 2014

[30] G. Bjontegaard, Calculation of Average PSNR Differences between RD-curves, ITU-T Q.6/SG16 VCEG, Document VCEG-M33, Apr. 2001.