An Adaptive Segment-Based View Synthesis Optimization Method for 3D-HEVC

Huan Dou*†, Yui-Lam Chan†, Ke-bin Jia*, Wan-Chi Siu†, Peng-yu Liu* and Qiang Wu*
* Beijing University of Technology, Beijing, China
† The Hong Kong Polytechnic University, Kowloon, Hong Kong
E-mail: douhuan88@gmail.com; enylchan@polyu.edu.hk; kebinj@bjut.edu.cn; enwcsiu@polyu.edu.hk

Abstract— **This paper presents an adaptive segment-based view synthesis optimization (VSO) for depth map coding in 3D-HEVC. The original method skips lines of pixels where there is no change in depth data, and exhaustively renders pixels from all the rest of lines to get distortions from synthesized views. It brings unnecessary coding operations and ignores smaller regions which can also be skipped in VSO. The proposed method is designed to skip proper segments of pixels for the non-skipped lines based on both of the texture smoothness and depth data unchangeableness. Experimental results show that the proposed method can achieve low coding complexity without significant performance loss for the synthesized views.**

Index Terms— **3D-HEVC, depth coding, view synthesis optimization, VSO, early skip**

I. Introduction

Nowadays, three-dimensional (3D) video applications become popular in both movie industry and consumer entertainment. As the current widely used 3D display, a stereoscopic display provides two views of videos and requires special glasses or screens to exhibit depth perception for audiences [1]. With the audiences' eager demand of more comfortable visual experience, a multi-view display such as auto-stereoscopic display attracts more attention from both academia and industry. This kind of display requires more than two views of videos and in turn brings a huge amount of data for transmission and storage.

Multi-view plus Depth (MVD) data format includes two or three texture videos and their corresponding depth maps [2]. Virtual intermediate views located between base views can be generated using a depth image based rendering (DIBR) technique [3]. Since depth maps are sets of grayscale images and use an 8-bit gray value to describe depth data, the coding efficiency of MVD data format is very valuable. The MVD data format is supported in 3D-HEVC which is the state-of-the-art 3D video coding standard specified by the Moving Picture Experts Group (MPEG) and Joint Collaborative Team on 3D Video Coding (JCT-3V) [1, 4].

Depth maps are employed in the virtual view rendering process because geometry information is necessary to be extracted in order to generate virtual views. However, lossy coding for depth maps will cause distortions in the synthesized views [5]. It is interesting to note that depth maps are not directly viewed by audiences. The distortion measure using the Lagrangian technique for the mode decision process of depth maps is then modified in consideration of both distortions in the synthesized intermediate views and the depth map in 3D-HEVC [6]. This technique is referred to as view synthesis optimization (VSO).

This VSO considers the distortion of synthesized views. Currently, there are two approaches to compute VSO. The first approach is called as synthesized view distortion change (SVDC) [7]. It is based on the real view synthesis process and has been the part of the HTM test model for the development of 3D video coding technology. It can achieve high coding efficiency but brings high coding complexity due to its pixel-by-pixel rendering operation. Many works have been proposed to reduce the computation complexity of SVDC [6-10]. The authors in [6-8] proposed an early skip (ES) method for SVDC, which has been included in the HTM test model. This ES method skips lines of pixels in a depth block if all these pixels are not distorted after encoding. This method can speed up the encoding process for depth maps with only negligible performance loss. The works in [9, 10] made an analysis to determine the condition in which pixels in the distorted depth map do not cause distortions in the synthesized view and these pixels are skipped to reduce encoding time. The second approach estimates the synthesized view distortion without direct real view synthesis process [11-14]. This well-known technique is called as view synthesis distortion (VSD) [11-12] which is also part of the HTM test model. The VSD method analyzes the relationship between depth quality and rendering quality, and fully avoids view rendering process in order to save time. Nevertheless, it may not estimate the accurate synthesized view distortion as compared with SVDC.

In this paper, we propose an algorithm to speed up the SVDC process for VSO. For the line that does not satisfy ES method, this line of pixels is checked to determine the segment within the line that does not cause distortion in the synthesized view. It is based on both of the texture smoothness and depth data unchangeableness. By skipping

This paper was partially supported by CSP, EIE, PolyU, a grant from the Central Research Grant of POLYU, China (Grant No. PolyU 5119/13E), the National Natural Science Foundation of China (Grant No. 61100131), the Key Project of Beijing Municipal Education Commission (Grant No. KZ201310005004), and the project supported by Beijing Postdoctoral Research Foundation (2015ZZ-23).

some segments, the proposed algorithm can further reduce the VSO processing time for the non-skipped lines in a block.

The rest of this paper is organized as follows. Section II generally presents the view synthesis optimization. Section III introduces the proposed segment-based VSO in detail. Section IV and Section V give the experimental results and the conclusions.

II. VIEW SYNTHESIS OPTIMIZATION

In the ordinary rate-distortion optimization (RDO) scheme, Lagrangian technique is used for mode decision and motion estimation. Each candidate mode and motion parameter are associated with a cost, and the candidate with the smallest cost is selected as the optimal decision. The Lagrangian formulation is formulated as

$$J = D + \lambda \times R \tag{1}$$

where J is the rate-distortion cost, D is the distortion, λ is the Lagrangian multiplier, and R is the number of bits used. D is usually measured by calculating the sum of squared differences (SSD) or the sum of absolute differences (SAD).

After lossy coding of depth maps, distortions from coded depth maps will lead to distortions in synthesized views. Therefore, the distortion measure should consider the synthesized view distortion. In 3D-HEVC, the distortion measure for depth maps calculates a weighted average of the synthesized view distortion and the depth map distortion, as below:

$$D = w_1 \cdot D_{depth} + w_2 \cdot D_{synth} \tag{2}$$

where D is the distortion obtained in VSO, w_1 and w_2 are the weights for the distortions from the depth map, D_{depth}, and the synthesized view, D_{synth}. D_{depth} uses SSD or SAD while D_{synth} adopts either SVDC or VSD. This paper mainly focuses on the SVDC process.

A. Synthesized View Distortion Change

Synthesized view distortion change (SVDC) defines the change of the overall distortion in the synthesized view due to the change of the depth data. It is defined as the distortion difference ΔD between two synthesized textures S'_T and $\tilde{S}_T$, as follows,

$$\begin{aligned} \Delta D &= \tilde{D} - D \\ &= \sum_{(x,y)\in I} \left[\tilde{S}_T(x,y) - S'_{T,\mathrm{Ref}}(x,y) \right]^2 \\ &\quad - \sum_{(x,y)\in I} \left[S'_T(x,y) - S'_{T,\mathrm{Ref}}(x,y) \right]^2 \end{aligned} \tag{3}$$

where $S'_{T,\mathrm{Ref}}$ denotes a reference texture rendered by the original texture and the original depth map. S'_T denotes a texture rendered from the reconstructed texture and the depth map S_D, which is composed of the encoded depth map in encoded blocks and the original depth map in other blocks. $\tilde{S}_T$ denotes a texture rendered from the reconstructed texture

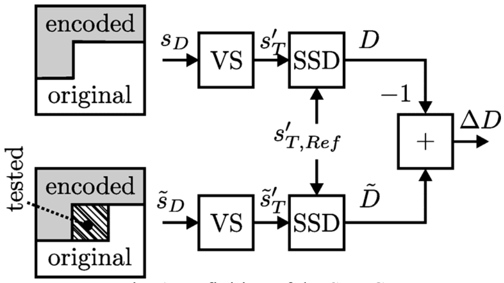

Fig. 1. Definition of the SVDC

and a depth map $\tilde{S}_D$ which differs from S_D where it contains the distorted depth map for the current block. The definition of SVDC is depicted in Fig. 1.

B. Early Skip (ES) Method of SVDC

In order to speed up the SVDC process, an ES method was proposed to eliminate unnecessary rendering and distortion calculation steps for lines of pixels from the current block. All the early skipped lines satisfy a particular condition in which the distortions of disparities for all pixels in the line is zero. In other words, if the distorted depth and the original depth are mapped to the same disparity vector for all pixels in a line of the depth block, rendering process is skipped for the line [8]. The disparity vector is calculated using the depth data according to the following equation:

$$disparity = (s \cdot v + o) >> n \tag{4}$$

$$s = \frac{f_c \cdot B}{255}\left(\frac{1}{z_{near}} - \frac{1}{z_{far}}\right) \tag{5}$$

$$o = \frac{f_c \cdot B}{z_{far}} - CS_T + CS_S \tag{6}$$

where s is the transmitted scale factor calculated by (5), v is the depth sample value, o is the transmitted offset calculated by (6), and n is a shift parameter depending on the required accuracy of the disparity vectors. Generally, n is set to 2, f_c is the focal length, B is the baseline distance between cameras, z_{near} and z_{far} are the nearest and farthest depth values, CS_T and CS_S are camera shifts of the target and source cameras.

The ES method proceeds in two steps: (1), the coded disparity vectors of all pixels in the block are determined whether they are all equal to the original disparity vectors. If this is the case, the rendering process of the whole block will be skipped and the SVDC of this block is set to 0. (2), the coded disparity vectors of all pixels in each line of the block are checked when the whole block cannot be skipped in step (1). If the coded disparity vectors are all equal to the original disparity vectors, this line can be skipped and the SVDC of this line is set to 0.

III. PROPOSED ADAPTIVE SEGMENT-BASED VSO (ASVSO) METHOD

In the original SVDC process, every pixel in the current block will be rendered to calculate SVDC in (1). This kind of one by one pixel rendering process will bring large computational complexity. The ES method can skip lines of pixels by judging if their coded disparity vectors are not distorted. However, there are still some segments in the non-skipped lines which may not cause distortions in the synthesized view. If these particular regions are identified and skipped, the SVDC process can further be speeded up. Based on this idea, the proposed adaptive segment-based VSO (ASVSO) method identifies these segments of pixels using both texture and depth information, and skips these segments in order to reduce the SVDC processing time.

A. Undistorted Conditions

This subsection describes two conditions for skipping segments of pixels in the current block that may not cause distortion in the synthesized view.

Condition 1: Texture smoothness

In the view rendering process, depth data is used to compute a disparity vector based on (4). It then finds the pixel position in the synthesized view from the base view. As shown in Fig. 2, dot A and dot B represent two pixel positions in the original view. The solid arrows in Fig. 2 denote disparity vectors calculated from the original depth values and the dotted arrows denote the distorted disparity vectors calculated from the distorted depth values due to the coding process. Dot A' and dot B' represent pixel positions in the synthesized view, while dot A" and dot B" denote the distorted pixel positions in the synthesized view. From Fig. 2 it can be observed that position A of the original view is located in a smooth background region. Even the depth value is distorted, the distorted position A" still has the same texture information as position A'. That is to say if the pixel is going to be warped to a smooth region in the synthesized view, the distorted depth data may not cause distortions in the synthesized view. Consequently, the rendering process of pixels in the segment can be skipped and SVDC of this segment can be considered as zero if the corresponding texture of the current depth segment is smooth. In the proposed algorithm, we will make use of this texture smoothness condition to measure the consistent luminance component value. In other words, the segment of pixels to compute SVDC can be skipped if its corresponding texture has consistent luminance component value.

Condition 2: Undistorted depth data

Since depth data are only used to obtain disparity vectors, change in the depth data only affects the pixel positions in the synthesized view. Instead of considering disparity vectors of all pixels in the whole line, our proposed algorithm is to study the undistorted depth data for a line segment. If all the depth pixels in the segment do not change after encoding, the view rendering process of this segment can be skipped and the SVDC can be set to zero.

B. Segment Determination

Based on the two conditions described in Section III.A, the adaptive segment-based VSO (ASVSO) method divides each line of the current block into segments adaptively. The procedure is shown in Fig. 3 and can be described as follows:

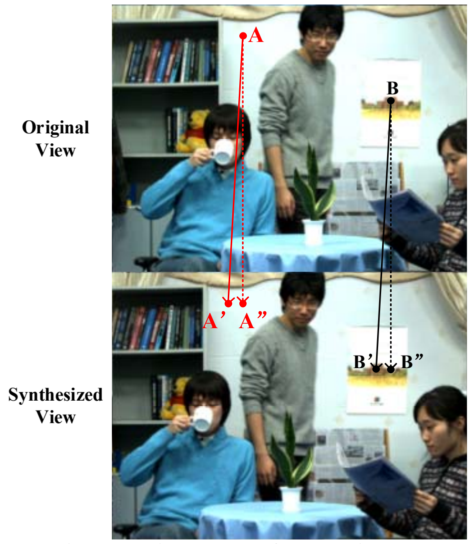

Fig. 2 Warping process of texture picture.

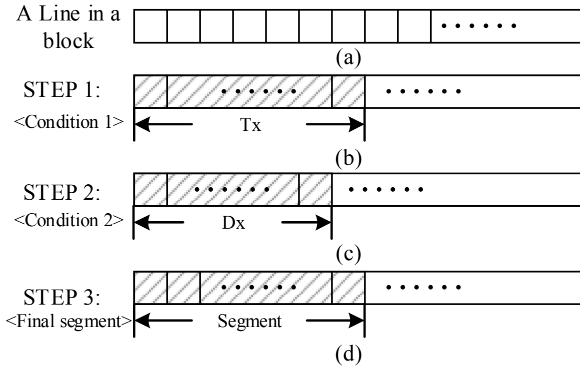

Fig. 3 Segment determination (a) A line in a block, (b) using Condition 1 to set a segment, (c) using Condition 2 to set a segment, (d) the final segment setting.

1) For a line in the current block being coded, firstly use Condition 1 (Texture smoothness) to find a segment with the consistent texture luminance component value, and the length of the segment is recorded as T_x, as shown in Fig. 3(b).

2) Use Condition 2 (Undistorted depth data) to locate a segment where all the encoded depth data are not distorted, and the length of the segment is recorded as D_x, as shown in Fig. 3(c).

3) Compare T_x and D_x and choose the larger one as the final segment length, as shown in Fig. 3(d).

C. ASVSO Algorithm

In this subsection, the process of the proposed ASVSO method is shown as follows. If a line of pixels is not early skipped, the two undistorted conditions described in Section III.A are applied to determine if there are segments of pixels which may not cause distortions in the synthesized view. If the segments are located through the segment determination described in Section III.B, their rendering process are skipped. The proposed ASVSO is worked with the early skip (ES) method. The detailed procedure of the ASVSO is described in Fig. 4. In order to keep the efficiency of the ASVSO, we skip

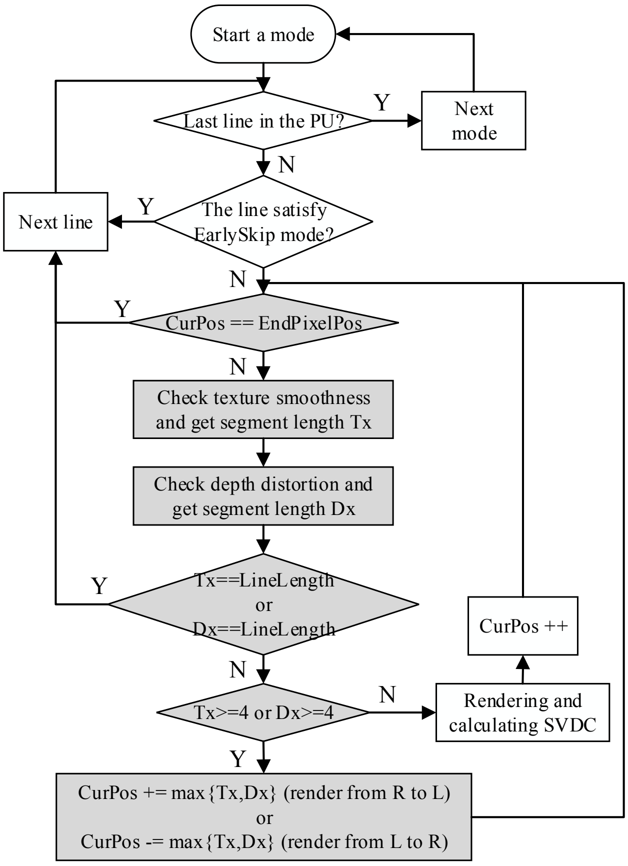

Fig. 4 Flowchart of the ASVSO.

segments only if the length of the segment is larger or equal to 4.

IV. EXPERIMENTAL RESULTS

Simulations were conducted to evaluate the performance of the proposed ASVSO with the ES method (ASVSO+ES). The ES algorithm [8] and Ref. [9] are reference methods. All tested algorithms were built on the 3D-HTM reference software 9.2 [15] and were evaluated based on the common test conditions specified in [16]. The GOP size and the intra period were set to 8 and 24, respectively. The virtual view step size was set to 0.25. The texture QPs were set to 25, 30, 35 and 40. The depth QPs were set to 34, 39, 42 and 45. “Balloons”, “Kendo”, “Newspaper”, “GTFly”, “PoznanStreet” and “UndoDancer” are the testing sequences. The parameters of these testing sequence are shown in Table I.

In order to verify the RD performance, we compared our ASVSO+ES to the ES proposed in [8] and algorithm proposed in [9]. Bjontegaard metric [17] was used to measure the coding performance between different algorithms. The bitrate reduction of base views and synthesized views for ASVSO+ES to [8] and [9] is shown in Table II. “Base”, “Syn.” and “All” in Table II denote the bitrate reduction of the base view, synthesized view and both of them, respectively.

From Table II, it can be found that Bjontegaard delta bitrate of ASVSO+ES compared to ES only increases 1.92% for the synthesized view and 1.32% for both of the synthesized and base views. For “Balloons” and “PoznanStreet” sequences, the proposed ASVSO+ES can achieve the highest performance with less than 0.6% Bjontegaard delta bitrate increase. That is to say the proposed ASVSO+ES do not bring significant performance loss for the synthesized views. In comparison with [9], Bjontegaard delta bitrate of ASVSO+ES increases 1.97% for the synthesized view and 1.25% for both of the synthesized and base views. For “PoznanStreet” sequences, ASVSO+ES increases 0.3% Bjontegaard delta bitrate. Since ASVSO+ES uses two conditions to determine suitable segments of pixels, the coding performance can be maintained.

TABLE I
TEST SEQUENCES’ PARAMETERS

Sequence	Resolution	Number of frames	Frame rate	3-view input
Balloons	1024*768	300	30	1-3-5
Kendo	1024*768	300	30	1-3-5
Newspaper	1024*768	300	30	2-4-6
GTFly	1920*1088	250	25	9-5-1
PoznanStreet	1920*1088	250	25	5-4-3
UndoDancer	1920*1088	250	25	1-5-9

TABLE II
BD-RATE COMPARISON

Sequence	ASVSO+ES to [8] (%)			ASVSO+ES to [9] (%)		
	Base	Syn.	All	Base	Syn.	All
Balloons	0.0	0.6	0.4	0.0	0.9	0.6
Kendo	0.0	1.4	0.9	0.1	1.4	1.0
Newspaper	0.0	2.6	1.7	0.0	2.7	1.8
GTFly	0.0	1.2	0.9	0.0	1.2	0.9
PoznanStreet	0.0	0.5	0.3	0.0	0.5	0.3
UndoDancer	0.0	5.3	3.7	0.0	5.1	2.9
Average	0.0	1.93	1.32	0.02	1.97	1.25

To evaluate the computational efficiency of different methods, we recorded the skip rate of ASVSO+ES, and then compared the depth encoding time of each sequence using the proposed ASVSO+ES, ES [8] and the method proposed in [9]. The skip rate of the pixels in non-skipped lines using the proposed ASVSO+ES is shown in Table III. The “Total” shows the skip rate of the whole ASVSO+ES algorithm. “Affected by Cond.1”, “Affected by Cond.2” and “Affected by Cond.1&2” in Table III show each proportion of skipped pixels affected by the two undistorted conditions (described in Section III.A) respectively and the two conditions simultaneously. “Cond.1 only” and “Cond.2 only” show the skip rates of pixels separately using only one of the two conditions respectively with the ES. It can be observed that the average skip rate of all test sequences is 47.43%, and the average skip rate of higher resolution sequences is larger than that of lower resolution sequences. For each sequences, Condition 1 shows greater impact than Condition 2 in average, while for lower QP situation Condition 2 makes more contribution to skip pixels. In larger QP situation, video contents is smoother so that Condition 1 plays major role in ASVSO. If the QP value is small, the encoding accuracy can be guaranteed and the depth distortion can be reduced so that Condition 2 has an obvious effect. Nevertheless, Condition 1 and Condition 2 both take important role in the proposed

TABLE III
SKIP RATE OF THE PROPOSED ASVSO ALGORITHM.

Sequence	Depth QP	Skip rate of pixels in non-skipped lines (%)				Cond.1 only (%)	Cond.2 only (%)
		Total	Proportion for each skipped pixels (%)				
			Affected by Cond.1	Affected by Cond.2	Affected by Cond.1&2		
Balloons	34	43.78	42.10	55.83	2.07	25.84	25.34
	39	41.74	50.82	47.61	1.57	27.48	20.72
	42	39.91	57.77	40.91	1.33	28.44	16.71
	45	39.44	68.04	30.56	1.40	31.42	12.12
Kendo	34	44.14	52.95	43.98	3.07	30.76	20.25
	39	44.13	63.48	34.26	2.26	33.83	16.07
	42	44.85	70.65	27.72	1.64	36.82	13.06
	45	46.63	78.00	20.54	1.46	40.90	9.92
Newspaper	34	37.12	32.56	65.83	1.61	16.97	25.44
	39	37.14	46.74	51.72	1.53	22.07	19.77
	42	37.31	59.76	38.80	1.45	26.46	14.99
	45	37.75	69.85	28.70	1.45	30.07	11.09
GTFly	34	52.31	55.77	41.42	2.81	38.28	23.53
	39	54.02	68.93	28.84	2.24	44.85	17.15
	42	60.67	80.70	17.82	1.48	54.91	13.03
	45	63.76	85.52	13.21	1.26	60.25	9.95
PoznanStreet	34	42.11	29.24	69.64	1.12	17.59	29.91
	39	44.70	46.67	51.92	1.41	27.30	23.83
	42	51.40	66.24	32.33	1.43	40.86	18.10
	45	53.64	75.16	23.34	1.49	45.72	13.72
UndoDancer	34	60.04	28.53	70.24	1.23	27.20	47.40
	39	55.53	44.14	54.33	1.53	32.81	34.57
	42	53.16	58.25	40.22	1.53	37.87	24.94
	45	53.11	70.88	27.61	1.51	43.72	16.52
Average 1024x768		41.16	57.73	40.54	1.74	29.25	17.12
Average 1920x1088		53.70	59.17	39.24	1.59	39.28	22.72
Average		47.43	58.45	39.89	1.66	34.27	19.92

TABLE IV
DEPTH ENCODING TIME COMPARISON OF DIFFERENT ALGORITHMS.

Sequence	QP	ES/Non-ES	[9]/Non-ES	(ASVSO+ES)/Non-ES	(ASVSO+ES)/ES	(ASVSO+ES)/[9]
Balloons	34	64.48%	82.12%	56.24%	87.23%	68.49%
	39	64.17%	82.49%	56.99%	88.82%	69.09%
	42	65.24%	83.30%	57.93%	88.80%	69.55%
	45	67.68%	85.30%	59.99%	88.64%	70.33%
Kendo	34	65.50%	82.93%	56.98%	86.99%	68.70%
	39	65.97%	82.91%	57.67%	87.41%	69.55%
	42	66.66%	83.31%	57.87%	86.82%	69.47%
	45	68.32%	88.04%	58.50%	85.62%	66.44%
Newspaper	34	70.81%	85.79%	63.45%	89.60%	73.96%
	39	71.14%	86.20%	63.36%	89.05%	73.50%
	42	71.25%	86.74%	63.22%	88.74%	72.89%
	45	72.02%	87.53%	63.68%	88.41%	72.75%
GTFly	34	60.34%	82.11%	51.37%	85.13%	62.56%
	39	63.10%	86.47%	52.80%	83.67%	61.06%
	42	67.82%	91.27%	53.44%	78.80%	58.55%
	45	68.78%	93.24%	53.08%	77.18%	56.93%
PoznanStreet	34	68.02%	81.09%	59.68%	87.75%	73.61%
	39	68.71%	83.00%	59.47%	86.55%	71.65%
	42	70.46%	86.87%	57.79%	82.01%	66.52%
	45	71.22%	88.90%	57.65%	80.95%	64.85%
UndoDancer	34	61.62%	75.78%	51.56%	83.68%	68.04%
	39	63.88%	80.27%	53.52%	83.78%	66.68%
	42	66.78%	84.57%	55.50%	83.10%	65.62%
	45	67.94%	86.14%	55.65%	81.92%	64.61%
Average 1024x768		67.77%	84.72%	59.66% (save 40.34%)	88.01% (save 11.99%)	70.39% (save 29.61%)
Average 1920x1088		66.56%	84.98%	55.13% (save 44.87%)	82.88% (save 17.12%)	65.06% (save 34.94%)
Average		67.16%	84.85%	57.39% (save 42.61%)	85.44% (save 14.56%)	67.72% (save 32.28%)

ASVSO and jointly devote to skipping pixels for VSO. For higher resolution sequences, Condition 1 is more relevant because the contents in a certain size of a block is more consistent than lower resolution sequences. Therefore, ASVSO can skip more pixels in a block of higher resolution sequences.

Table IV gives the results of the depth coding time comparison of the proposed ASVSO and other reference algorithms. “ES” and “Non-ES” represent the original SVDC with and without ES, respectively. From Table IV, it can be observed that the proposed ASVSO together with ES can save 42.61% of the depth map encoding time in average compared to the original SVDC without ES and can further reduce 14.56% depth encoding time in average compared to ES. Compared to the method proposed in [9], the proposed ASVSO can save 32.28% depth encoding time. With the same reason described above, ASVSO can save more depth encoding time for high resolution sequences.

V. CONCLUSIONS

In this paper, a segment-based view synthesis optimization algorithm is proposed to skip unnecessary pixel rendering process for depth maps in 3D-HEVC. The proposed algorithm skips segments of pixels for the SVDC calculating process in non-skipped lines of a block by taking both texture and depth information into account. Using the proposed algorithm, the SVDC processing time from the non-skipped lines can further be reduced without significant performance loss. This new VSO method can enhance the early skip method and can achieve better time performance for high resolution sequences.

ACKNOWLEDGMENT

This paper was partially supported by CSP, EIE, PolyU, a grant from the Central Research Grant of POLYU, China (Grant No. PolyU 5119/13E), the National Natural Science Foundation of China (Grant No. 61100131), the Key Project of Beijing Municipal Education Commission (Grant No. KZ201310005004), and the project supported by Beijing Postdoctoral Research Foundation (2015ZZ-23).

REFERENCES

[1] Muller. K., Schwarz. H., Marpe. D., Bartnik. C., Bosse. S., 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.

[2] Merkle, P.; Smolic, A.; Muller, K.; Wiegand, T., "Multi-View Video plus Depth Representation and Coding," *2007 IEEE International Conference on Image Processing (ICIP 2007)*, vol.1, pp.I201-I204, Sept. 16 2007-Oct. 19 2007.

[3] Smolic. A, Muller. K., Dix. K., Merkle. P., Kauff. P., et al. "Intermediate View Interpolation based on Multiview Video plus Depth for Advanced 3D Video Systems," *15th IEEE International Conference on Image Processing*, San Diego, USA, pp.2448-2451, Oct. 2008.

[4] Text of ISO/IEC DIS 23008-2 High Efficiency Video Coding, *Standard ISO/IEC JTC1/SC29/WG11*, July 2012.

[5] Merkle, P., Morvan, Y., Smolic, A., Farin, D., Muller, K., et al. "The Effect of Depth Compression on Multiview Rendering Quality." *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video*, 2008. IEEE, 2008.

[6] Ying Chen, Gerhard Tech, Krzysztof Wegner, Sehoon Yea, “Test Model 8 of 3D-HEVC and MV-HEVC,” JCT3V-H1003, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 8th JCT3V Meeting, Valencia, ES, 2014.

[7] Tech. G., Schwarz. H., Muller. K., Wiegand. T., "3D Video Coding Using the Synthesized View Distortion Change," *2012 Picture Coding Symposium (PCS)*, pp.25-28, May 2012.

[8] Byung Tae Oh, Jaejoon Lee, Du Sik Park, Gerhard Tech, Karsten Müller, et al. “3D-CE8.h Results on View Synthesis Optimization by Samsung, HHI and LG-PKU,” JCT3V-A0093, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 1st Meeting: Stockholm, SE, 16–20 July 2012.

[9] Siwei Ma, Shiqi Wang, Wen Gao, "Low Complexity Adaptive View Synthesis Optimization in HEVC Based 3D Video Coding," *IEEE Transactions on Multimedia*, vol.16, no.1, pp.266-271, Jan. 2014.

[10] Siwei Ma, Shiqi Wang, Wen Gao, "Zero-synthesis View Difference Aware View Synthesis Optimization for HEVC based 3D Video Compression," *2012 IEEE Visual Communications and Image Processing (VCIP)*, pp.1-6, Nov. 2012.

[11] Byung Tae Oh, Jaejoon Lee, Du-Sik Park, "Depth Map Coding Based on Synthesized View Distortion Function," *IEEE Journal of Selected Topics in Signal Processing*, vol.5, no.7, pp.1344-1352, Nov. 2011.

[12] Byung Tae Oh, Kwan-Jung Oh, "View Synthesis Distortion Estimation for AVC- and HEVC-Compatible 3-D Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.24, no.6, pp.1006-1015, June 2014.

[13] Ma R, Cheung N M, Au O C, Tian D. “Novel Distortion Metric for Depth Coding of 3D Video,” *2013 IEEE International Conference on Image Processing (ICIP)*, pp.1714-1718, Sept. 2013.

[14] Zhang. Q., An. P., Zhang. Y., Zhang. Z., “Efficient Rendering Distortion Estimation for Depth Map Compression,” *2011 18th IEEE International Conference on Image Processing (ICIP)*, pp.1105-1108, Sept. 2011.

[15] https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-9.2/

[16] Karsten Muller, Anthony Vetro, “Common Test Conditions of 3DV Core Experiments”, JCT3V-G1100, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 7th JCT3V Meeting, San Jose, US, 2014.

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