VIEW SYNTHESIS OPTIMIZATION BASED ON TEXTURE SMOOTHNESS FOR 3D-HEVC

Huan Dou[1], Yui-Lam Chan[2], Ke-Bin Jia[1], Wan-Chi Siu[2]

[1] Digital Multimedia Information Processing and Communication Technique Lab
Department of Electronic Information and Control Engineering
Beijing University of Technology, Beijing, China
[2] Center for Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ABSTRACT

This paper presents view synthesis optimization for 3D-HEVC based on a new texture smoothness process. In the original method, all pixels are exhaustively rendered to get distortions from synthesized views. Since not all pixels from the distorted depth map may cause distortions in the synthesized view, it brings unnecessary coding complexity. In this paper, lines of pixels are skipped based on the analysis of pixel regularity from smooth texture regions. It is due to the fact that the distorted disparity may not have much effect on the synthesized view in smooth texture regions. The proposed method can reduce the coding complexity of view synthesis optimization without significant performance loss.

Index Terms— 3D-HEVC, depth coding, view synthesis optimization, VSO, texture smoothness

1. INTRODUCTION

Three-dimensional (3D) video, compared to the conventional 2D video, can provide visual experiences with depth perception by the special displays that re-projected 3D scene from slightly different directions for the left and right eye [1]. With the recent advances in multimedia technology, the resolution of 3D videos becomes higher. Besides, stereoscopic displays, which provide two views of videos, cannot satisfy the users' demand. Multi-view displays, which consist of a multitude of views, have become a new focus among multimedia information industry. However, a multitude of views will lead to a huge amount of data. To reduce the data rate of multi-view videos, an efficient coding scheme is desired.

The most recent 2D video coding standard is the High Efficiency Video Coding (HEVC) standard. It was officially approved in 2013 as ITU-T Recommendation H.265 and ISO/IEC 23008-2 (MPEG-H Part 2). Due to its superior performance for 2D videos, Moving Picture Experts Group (MPEG) and Joint Collaborative Team on 3D Video Coding (JCT-3V) have started the standardization for 3D video coding, namely 3D-HEVC [1-4].

Multi-view plus Depth (MVD) video format is supported in 3D-HEVC. It consists of 2 or 3 texture videos associated with their corresponding depth maps. By making use of depth maps, virtual views between base views can be generated using Depth Image Based Rendering (DIBR) technique [5, 6]. Coding of depth maps can save lots of bit rates and reduce much complexity compared to texture coding. Since depth maps are only used to generate virtual views and not directly viewed, it is not optimal to encode depth maps using the conventional video encoder. Hence, decisions for depth map coding need to be based on synthesized view distortion instead of depth map distortion.

Currently, many depth map coding algorithms focusing on estimating the synthesized view distortion have been suggested. They can be classified into two categories: rendering and non-rendering. In the first category, the view synthesis process is applied and the calculation of the exact synthesized view distortion change (SVDC) [7] replaces the original rate distortion (RD) calculation. This approach uses real distortions of synthesized views and is referred to as View Synthesis Optimization (VSO). It can achieve high coding efficiency but brings high coding complexity due to the view rendering process. Many research works [8-10] have focused to reduce the computing complexity. In [8], an algorithm makes an analysis to determine the condition in which pixels in the distorted depth map do not cause distortions in the synthesized view, but the analysis process is time consuming. In the second category, the approximated synthesized view distortion is estimated instead of direct calculation of SVDC [11-14]. This approach fully mimics the view rendering process in order to save time. Nevertheless, it may not estimate the accurate synthesized view distortion.

In this paper, we propose an algorithm to speed up the VSO process. The view synthesis process is analyzed and the texture smoothness is employed to judge if it is necessary to use the VSO process for some regions. This judging process is used to skip some unnecessary calculation of VSO in order to reduce the computational complexity. The proposed algorithm quickly determines the threshold using the statistical method in order to minimize the threshold computing process and further reduce the time for view synthesis optimization.

The rest of this paper is organized as follows. Section 2 generally presents the SVDC process. Section 3 introduces the proposed VSO based on texture smoothness in detail. Section 4 and Section 5 give the experimental results and the conclusions.

2. SYNTHESIZED VIEW DISTORTION CHANGE

Synthesized view distortion change (SVDC) relates the distortion of a depth map to the distortion of the synthesized view in order to improve the rate distortion optimization in depth map coding [7] and has become part of the test model for the development of 3D video coding technology [15].

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 shown in (1):

$$\begin{aligned}\Delta D &= \widetilde{D} - D \\ &= \sum_{(x,y)\in I}\left[\tilde{s}_T(x,y) - s'_{T,Ref}(x,y)\right]^2 \\ &- \sum_{(x,y)\in I}\left[s_T'(x,y) - s'_{T,Ref}(x,y)\right]^2\end{aligned} \quad (1)$$

where $S_{T,Ref}'$ denotes a reference texture rendered by the original texture and 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. $\tilde{S}_T$ is a texture rendered from the reconstructed texture 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.

2.1. Computation of SVDC

In order to avoid the complete rendering process resulting in huge complexity, the current test model uses a method called a renderer model to enable fast computation of SVDC. The renderer model has three core processes: Initialization, partial re-rendering and SVDC calculation.

The initialization process is carried out before the depth map coding. In this process, the complete synthesized texture is rendered by the original input depth and the original input textures.

After the final coded depth data of the block is known, a partial re-rendering process takes place. In this process, the rendering model changes the block in S_D from the original to coded data and re-renders only parts of the synthesized view S_T'in Eq. (1) that are affected by the change of the depth map data.

The computation of the SVDC follows Eq. (1). In this process, though the re-rendering is carried out, no variables of the rendering model will be modified. It makes several SVDC computations from multiple depth candidates without the need of re-rendering the original data.

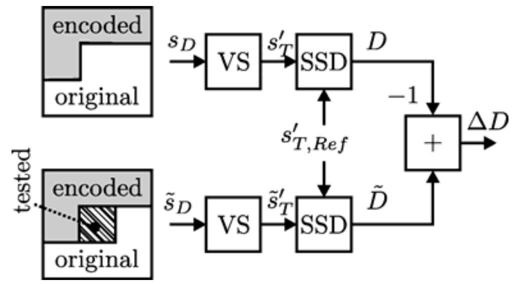

Fig. 1. Definition of the SVDC

2.2. Early Skip Mode of SVDC

The test model also introduces an Early Skip mode for SVDC in order to speed up the computation process if the distortion of the disparity is zero [15]. The Early Skip mode proceeds in two steps. Firstly, the coded depth values of all the pixels in the block will be determined whether they are all equal to the original depth values. If it is the case, the whole block will skip the rendering process and SVDC is set to 0. Secondly, if the whole block cannot be skipped, each line of the block will be check if the coded depth values are all equal to the original. If it is true, this line can be skipped and SVDC of this line is set to 0.

3. PROPOSED VSO BASED ON TEXTURE SMOOTHNESS

When computing SVDC for the current block in the original VSO process, every pixel in this block will be warped, interpolated and blended to render a virtual view. Then the encoder will calculate SVDC in Eq. (1). By doing so, a large amount of pixels need to be rendered because of the high resolution of 3D video. This kind of one by one pixel rendering process will bring large computing complexity.

In fact, it is not necessary to render every pixel. The Early Skip mode is a kind of solution and can skip lines or blocks of pixels by judging if their depth values are not distorted. However, there is another scenario due to the local characteristics of the reference texture video that may not cause distortions of the synthesized view in spite of the distorted depth map. The proposed algorithm is then based on the texture smoothness to speed up the VSO process.

3.1. Description of the Proposed Algorithm

In view rendering process, depth value is used to get a disparity vector to find the pixel position in the synthesized view from the base view. As shown in Fig. 2, the arrows denotes the disparity, the black dots represent the pixel positions in the original view, and the white dots indicate the pixel positions in the synthesized view. Dotted arrow denotes the distorted disparity vector that is calculated from the coded depth value, and the grey dot denotes the distorted pixel position. Pixel c from the base view is located in a smooth region (a, b, c, d) and the distorted warped position c” is different from the original warped position c’. But the synthesized view may not change because the pixel is warped to the same smooth region. That is to say if the pixel is going to be warped in the smooth region of the synthesized view, the depth change may not have much effects on the synthesized view.

Based on this observation, we propose an algorithm to speed up the VSO process using the texture smoothness. If a line of pixels is not skipped by the Early Skip mode, and it is in a smooth region of the texture picture, the rendering process for this line can be skipped. The procedure of the proposed method can be described in Fig. 3.

3.2. Texture Smoothness Judgement

The key issue of the proposed method is to judge the texture smoothness. Generally, the luminance (Y) and chrominance (UV) values can be used to measure the texture smoothness. If a region of pixels is smooth, the YUV values of each pixel must be very regular. Here we give a condition to judge the regularity for pixels in a smooth region:

$$\left|Y_p - Y_q\right| \leq T \tag{2}$$

where Y denotes the luminance value for the pixel, p and q are neighboring pixels, and T is the threshold which reflects the regularity of the smoothness.

An intra DC prediction mode uses an average value of reference samples for intra prediction. In other words, if a block chooses an intra DC mode as its best mode, it can be seen as a smooth block. Hence, we recorded CU with the size of $n \times n$ that uses intra DC mode and calculated the average difference between each pixel value and the average pixel value:

$$\bar{Y} = \frac{1}{n \times n} \sum_{i}^{n} \sum_{j}^{n} Y_{i,j} \tag{3}$$

$$A = \frac{1}{n \times n} \sum_{i}^{n} \sum_{j}^{n} \left(Y_{i,j} - \bar{Y}\right) \tag{4}$$

where $\bar{Y}$ denotes the average pixel value, i and j denote the pixel coordinates, and A denotes the average difference.

For each sequence, we compute the value of A for each CU according to (3) and (4). Then the most representative values of A for different sequences can be obtained as shown in Table 1. The values of A in the table are fractional because all values are derived based on the average values of A from 4 QPs and 3 base views. Based on the statistical results, we make the threshold equals to 3. That is to say if all the pixels in a line satisfy the condition where the difference between neighboring pixels are all less than 3, this line can be seen as a smooth line. According to this threshold, we made statistics of the percentage of smooth lines. The results are shown in Table 2 in which the Early Skip mode is excluded. From Table 2, we can find except the lines skipped by the Early Skip mode, half of the rest lines can be judged as smooth lines using the proposed threshold.

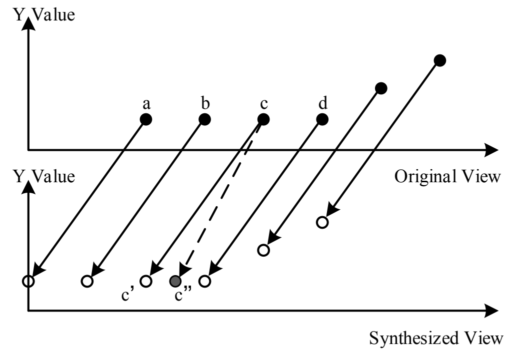

Fig. 2 Warping process of texture picture

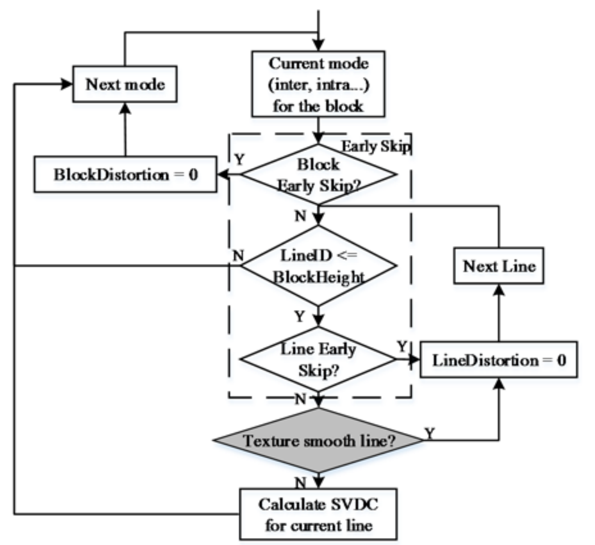

Fig. 3 Flowchart of the proposed algorithm in VSO

Table 1. Distribution of A in Eq. (5)

Sequence	*Most representative values of A*	Average values of A
Newspaper	3.16	4.08
Balloons	2.75	3.42
Kendo	2.81	3.00
GTFly	1.88	2.25
UndoDancer	3.00	3.00
PoznanStreet	5.67	6.08
Average	3.21	3.64

4. EXPERIMENTAL RESULTS

To evaluate the performance, the proposed algorithm was built on the 3D-HTM reference software 9.2 [16] and was performed complying with the common test conditions [17]. The format of three views is used in this experiment. The

Table 2. Statistics of the percentage of smooth lines

Sequence	Percentage of smooth lines (%)			
	QP=25	QP=30	QP=35	QP=40
Newspaper	38.81	47.32	54.25	62.17
Balloons	50.03	57.73	64.53	70.49
Kendo	32.73	43.00	50.25	55.47
GTFly	40.06	52.55	67.72	74.57
UndoDancer	37.51	50.70	61.86	66.79
PoznanStreet	33.67	44.19	53.57	63.86

Table 3 BD-rate comparison

Sequence	Proposed (%)			Method in [8] (%)		
	Base	Syn.	All	Base	Syn.	All
Balloons	0.0	0.2	0.1	0.0	-0.2	-0.1
Kendo	0.0	0.6	0.3	-0.1	0.1	0.0
Newspaper	0.0	1.6	1.0	0.0	-0.1	0.0
GTFly	0.0	0.1	0.1	0.0	0.0	0.0
UndoDancer	0.0	0.9	0.7	0.0	0.2	0.1
Average	0.0	0.7	0.4	0.0	0.0	0.0

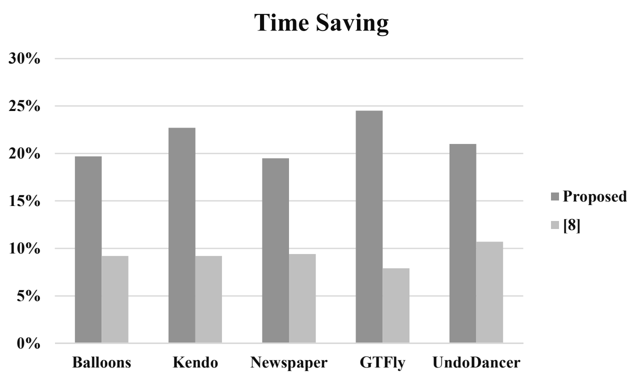

Fig. 4 Time saving of the proposed algorithm compared with the method in [8]

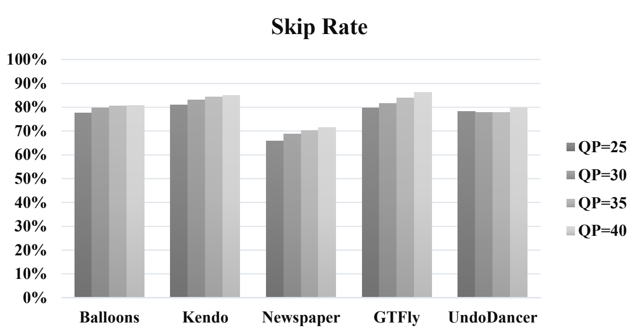

Fig. 5 Skip rate of the proposed algorithm

GOP size is set to 8 and the intra period is set to 24. The virtual view step size is set to 0.25. The texture QPs are set to 25, 30, 35 and 40. The depth QPs are set to 34, 39, 42 and 45. Balloons (1024x768), Kendo (1024x768), Newspaper (1024x768), GTFly (1920x1088), and UndoDancer (1920x1088) are the testing sequences.

We compared our algorithm to the original method with early skip disabled. The algorithm in [8] was used as a reference algorithm. Bjontegaard metric [18] was used to measure the coding performance between different algorithms. The bitrate reduction of base views and synthesized views for the proposed algorithm and [8] are separately shown in Table 3. “Base”, “Syn.” and “All” in Table 3 denote the bitrate reduction of the base view, synthesized view and both of them, respectively.

It can be seen from Table 3 that compared to the original algorithm, the proposed algorithm basically kept the same coding performance, with 0.6% bitrate increase for the synthesized view and 0.4% bitrate increase for the synthesized and base view in average. The proposed algorithm can achieve the highest performance for “GhostTownFly” because the sequence has more smooth background scenes so that the proposed method can speed up the VSO process with less performance loss.

The time saving of the proposed algorithm and [8] compared to the original algorithm are shown in Fig. 4. The total skip rates are also shown in Fig. 5. From Fig. 4 and Fig. 5, it can be observed that the proposed algorithm can save above 21% time and skip above 70% lines of pixels in average compared to the original method with early skip mode disabled. The proposed method also reduces computational complexity as compared with the method in [8]. This is because the looping process in [8] for computation of the threshold takes time but the proposed method does not need this looping process. The results show that the proposed algorithm based on texture smoothness is very effective for VSO, and can even achieve good performance for high resolution videos. It is because for high resolution videos, more smooth regions can be skipped.

5. CONCLUSIONS

In this paper, a new texture smoothness measure applied to view synthesis optimization is proposed to skip the unnecessary process for 3D-HEVC. The proposed algorithm analyzes the regularity of pixels in the smooth texture region, and utilizes the regularity to skip lines of pixels which are in smooth regions. This proposed algorithm can reduce the heavy computational burden of the exhaustive pixel rendering process without significant performance loss. This new VSO algorithm shows improvement compared to other methods proposed in the literature.

6. ACKNOWLEDGEMENT

This paper was supported by the National Natural Science Foundation of China (Grant No. 61100131) and the Key Project of Beijing Municipal Education Commission (Grant No. KZ201310005004). This paper was also partially supported by CSP, EIE, PolyU, and a grant from the Central Research Grant of POLYU, China (Grant No. PolyU 5119/12E).

12. REFERENCES

[1] Muller. K., Schwarz. H., Marpe. D., Bartnik. C., Bosse. S., Brust. H., Hinz. T., Lakshman. H., Merkle. P., Rhee. F.H., Tech. G., Winken. M., Wiegand. T., "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] *Text of ISO/IEC DIS 23008-2 High Efficiency Video Coding*, Standard ISO/IEC JTC1/SC29/WG11, July 2012.

[3] Sullivan. G. J., Ohm. J., Han. W. J., Wiegand. T., "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.22, no.12, pp.1649-1668, 2012.

[4] Tohidypour, Hamid Reza, Mahsa T. Pourazad, Panos Nasiopoulos, "A low complexity mode decision approach for HEVC-based 3D video coding using a Bayesian method," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.895-899, 2014.

[5] Tehrani. M. P., Ishikawa. A., Okui. M., Inoue. N., Takahashi. K., Fujii, T., "Synthesis Error COmpeNsateD Multiview Video plus Depth for representation of multiview video," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 890-894, May 2014.

[6] Smolic. A, Muller. K., Dix. K., Merkle. P., Kauff. P., Wiegand. T., "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.

[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] 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.

[9] 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.

[10] Shiqi Wang, Siwei Ma, Hongbin Liu, Jie Jia, "CE8.h: Results of Simplification of View Synthesis Optimization by Detection of Zero Distortion Change in Synthesized View," *JCT2-A0083, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, 1st JCT3V Meeting, Stockholm, SE, 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] 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.

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

[17] 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.

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