Classified-edge guided depth resampling for multi-view coding*

LU Yu (陆宇)[1,2]**, **ZHOU Yang** (周洋)[1], and **CHEN Hua-hua** (陈华华)[1]

1. College of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China

2. Department of Electrical Engineering and Computer Science, Oregon State University, Corvallis 97331, U.S.A.

(Received 21 October 2015)

A new depth resampling for multi-view coding is proposed in this paper. At first, the depth video is downsampled by median filtering before encoding. After decoding, the classified edges, including credible edge and probable edge from the aligned texture image and the depth image, are interpolated by the selected diagonal pair, whose intensity difference is the minimum among four diagonal pairs around edge pixel. According to different category of edge, the intensity difference is measured by either real depth or percentage depth without any parameter setting. Finally, the resampled depth video and the decoded full-resolution texture video are synthesized into virtual views for the performance evaluation. Experiments on the platform of multi-view high efficiency video coding (HEVC) demonstrate that the proposed method is superior to the contrastive methods in terms of visual quality and rate distortion (RD) performance.

Document code: A **Article ID:** 1673-1905(2016)01-0077-4

DOI 10.1007/s11801-016-5207-2

The state-of-art standard of high efficiency video coding (HEVC) has been finalized by the joint collaborative team on video coding (JCT-VC)[1]. Targeting a large number of three dimensional (3D) video applications, both multi-view video plus depth (MVD) HEVC standard and multi-view HEVC standard have been released[2]. Depth information is important for 3D video coding to improve its coding performance. For example, depth image is combined with texture image for joint bit allocation to enhance the coding quality[3]. And it is applied to fast mode decision algorithm for multi-view coding[4]. In addition, it is also proved that depth resampling can achieve bit rate savings without degrading the overall peak signal-to-noise ratio (*PSNR*) performance of video compression[5]. Bilateral filtering consisting of range kernel and spatial kernel is a typical edge-preserving upsampling method. It is extended to joint bilateral upsampling (JBU) for high resolution depth reconstruction with corresponding texture image while preserving boundary[6]. Afterwards, JBU method is integrated with other methods, such as median filtering, to enhance the effect of depth upsampling[7]. The main disadvantage for bilateral filtering and its derivative methods is the high computational workload and the required selection of filter parameters. Recently, Kim et al[8] proposed an edge-preserving depth upsampling method. But they only estimated the horizontal and vertical directions of edge pixel, which are not enough to represent all edge directions. And only common edges from the original depth image and the texture image were considered, which will result in poor performance when it is applied to the practical video coding. In addition, the exponential function for the calculation of cost function needs to set up the smoothing parameter manually. Actually, feature correspondence plays a significant role for multi-view coding to attain higher coding efficiency[9]. Considering the edge correspondence between texture image and depth image, we propose a classified-edge guided depth resampling method for multi-view coding in this paper.

According to the proposed coding structure shown in Fig.1, the texture video is encoded and decoded at the full resolution, while the depth video is resampled during encoding and decoding. Firstly, the depth video is downsampled to low resolution by median filtering before encoding, whose function is edge-preserved. After decoding, the texture video is downsampled by median filtering, and then it generates the edge map aligned with the decoded depth video. Guided by classified edge map, the depth video is refined by diagonal interpolation. Finally, the refined depth video is upscaled to the original resolution by the nearest interpolation algorithm, and then synthesized into virtual views with decoded texture video.

* This work has been supported by the National Natural Science Foundation of China (Nos.61401132 and 61372157), and the Zhejiang Provincial Natural Science Foundation of China (No.LY12F01007),

** E-mail: prcshylu@gmail.com

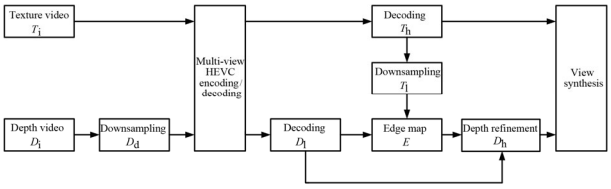


Fig.1 Schematic diagram of proposed coding structure

At first, the edge E_d of decoded depth video D_l and the edge E_t of downsampled texture video T_l are obtained by performing Canny operator, which is followed by morphological dilation operator with 3×3 square structuring element on E_d and E_t, respectively. The edge map is composed of E_d and E_t, i.e.,

$$E = E_d \bigcup E_t . \tag{1}$$

Then it is divided into three regions as

$$E = \begin{cases} E_c, & E_c \in E_d \bigcap E_t \\ E_p, & E_p \in (E_d \bigcup E_t) \text{ and } E_p \bigcap E_c = \varnothing \\ E_n, & \text{otherwise} \end{cases}, \tag{2}$$

where E_c is the credible edge which belongs to the common edge of E_d and E_t, E_p is the probable edge which belongs to either E_d or E_t but is not their common edge, and E_n is the remainder non-edge region in E. Aligned with classified edge map E, the corresponding regions in D_l are labeled as R_c, R_p and R_n, respectively, which will guide the depth refinement further.

Generally, there are eight neighbor pixels around central pixel. In order to decrease the complexity, we merge them into four diagonal pairs whose directions are represented as 0°, 45°, 90° and 135° as shown in Fig.2. The four directions are used for the direction estimation of edge pixel. The steps for depth refinement are described as follows.

(1) Find pixel pair with minimum intensity difference. For region R_c, the intensity difference for each pair is calculated as

$$P_k = ||G_{k1} - G_{k2}||, \ k = 0°, 45°, 90° \text{ and } 135°, \tag{3}$$

where G_{k1} and G_{k2} are the real depth intensity for each pixel in one pair, respectively, and $||\cdot||$ is the Euclidean distance operator.

For region R_p, the intensity of each pixel in the eight neighbors around edge pixel is denoted as

$$I_j = G^p_{j,0} + F^p_{j,0}, \ j = 1,2,\cdots,8 , \tag{4}$$

where $G^p_{j,0}$ is the percentage depth intensity, which is defined as

$$G^p_{j,0} = \frac{G_{j,0} - \min\_ G_{j,0}}{\max\_ G_{j,0} - \min\_ G_{j,0}}, \ j = 1,2,\cdots,8 , \tag{5}$$

where

$$G_{j,0} = \left\| G_j - G_0 \right\|, \quad j = 1,2,\cdots,8 , \tag{6}$$

where G_j is the depth intensity of neighbor pixel j, G_0 is the depth intensity of central edge pixel, and $\min\_G_{j,0}$ and $\max\_G_{j,0}$ are the minimum and the maximum among $G_{j,0}$, respectively. $F^p_{j,0}$ is the co-located percentage texture intensity, which has the similar definition as

$$F^p_{j,0} = \frac{F_{j,0} - \min\_ F_{j,0}}{\max\_ F_{j,0} - \min\_ F_{j,0}}, \ j = 1,2,\cdots,8 , \tag{7}$$

where

$$F_{j,0} = \left\| F_j - F_0 \right\|, \quad j = 1,2,\cdots,8 , \tag{8}$$

where F_j is the texture intensity of neighbor pixel j, F_0 is the texture intensity of central edge pixel, and $\min\_F_{j,0}$ and $\max\_F_{j,0}$ are the minimum and the maximum among $F_{j,0}$, respectively.

Then the intensity difference for each diagonal pair is calculated by

$$P_k = ||I_{k1} - I_{k2}||, \ k = 0°, 45°, 90° \text{ and } 135°, \tag{9}$$

where I_{k1} and I_{k2} are the intensity for each pixel in one pair, and are computed by Eq.(4), respectively.

(2) The candidate pair which has the minimum P_k is selected to calculate the new depth intensity G_m of central edge pixel as

$$G_m = (G_{m1} + G_{m2})/2, \tag{10}$$

where G_{m1} and G_{m2} are the depth intensity of each pixel in the candidate pair, respectively.

(3) For region R_r, the depth intensity is maintained without any change.

Finally, the refined depth video is upscaled by the nearest interpolation algorithm to the same resolution as that of the decoded texture video T_h for view synthesis.

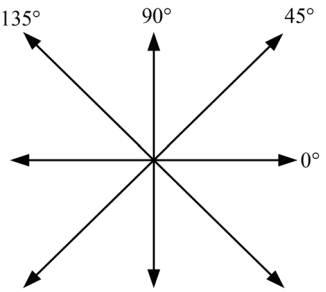


Fig.2 Diagonal pairs around central edge pixel

All experiments are performed on the multi-view HEVC reference software HTM-12.0. The coding configuration conforms to the common test conditions (CTC) defined by JCT-3V[10]. View 2 and view 4 for sequence Newspaper as well as view 1 and view 3 for sequence Balloons are synthesized into five virtual views, respectively. Quantization parameters (QPs) adopted in the experiments are 25 for texture video and 34, 39, 42 and 45 for depth video, respectively. The downsampled factor is 2, and 3×3 window is used for median filtering.

The percentage threshold for Canny edge detector is [0.12 0.4]. Other parameters setting for JBU method and Kim's method are referred in Ref.[8]. Depth-image-based-rendering (DIBR) algorithm is used for view synthesis[11]. The results for 27th frame of Newspaper are shown in Fig.3, while the results for 30th frame of Balloons are given in Fig.4. Fig.3(a) and Fig.4(a) are the depth images of view 2 and view 1, respectively, and Fig.3(b) and Fig.4(b) are the texture images of view 2.5 and view 1.5, respectively. The cropped images are shown in Fig.3(c)—(h) and Fig.4(c)—(h). Both in Fig.3(c) and (d) and Fig.4(c) and (d), there are some disperse particles along the edge of object. And jaggy artifacts are also seen in Fig.3(e) and (f). In addition, there are small holes in Fig.4(e) and some noticeable distortions on the border of pink ball in Fig.4(f). Oppositely, the results for the proposed method appear integrated and smooth on the edge of object both in Fig.3(g) and (h) and Fig.4(g) and (h).

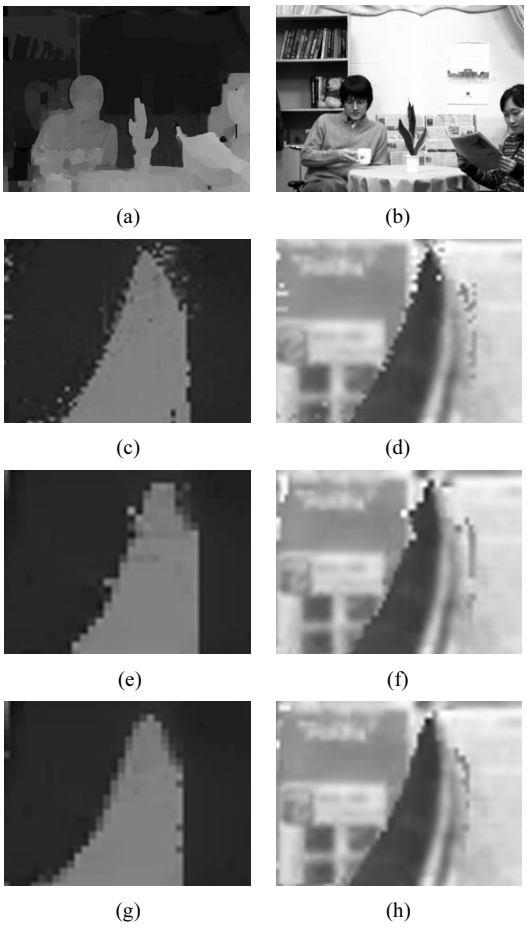


Fig.3 Experimental results for Newspaper: (a) original depth image; (b) synthesized texture image using (a); (c) depth image upsampled by JBU method; (d) synthesized texture image using (c); (e) depth image upsampled by Kim's method; (f) synthesized texture image using (e); (g) depth image upsampled by the proposed method; (h) synthesized texture image using (g)

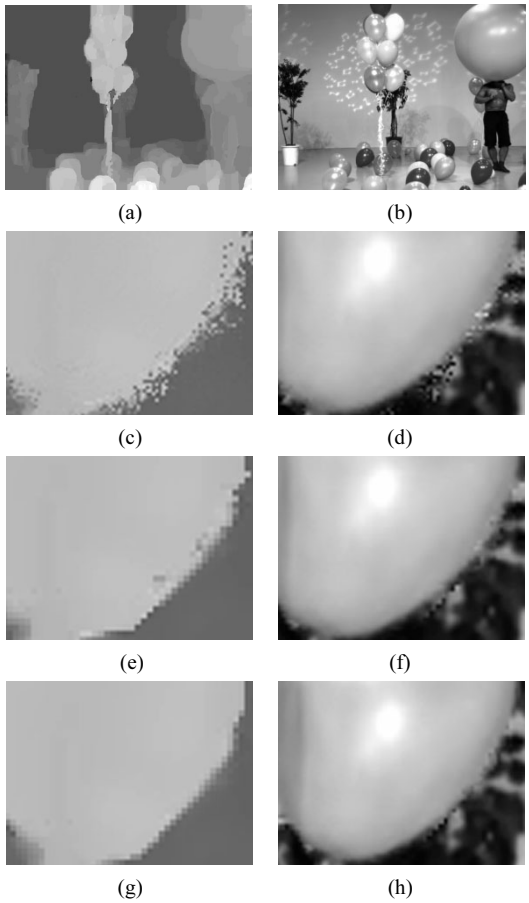
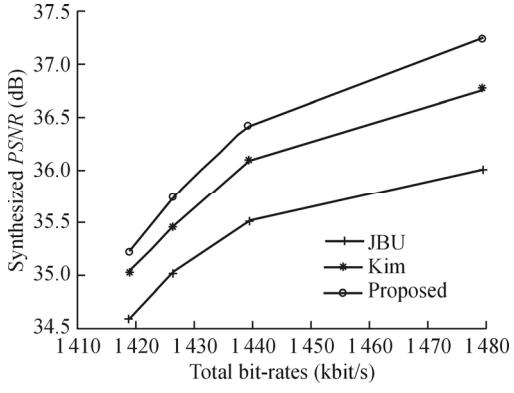


Fig.4 Experimental results for Balloons: (a) original depth image; (b) synthesized texture image using (a); (c) depth image upsampled by JBU method; (d) synthesized texture image using (c); (e) depth image upsampled by Kim's method; (f) synthesized texture image using (e); (g) depth image upsampled by the proposed method; (h) synthesized texture image using (g)

For objective evaluation, the performance comparison of rate distortion (RD) is illustrated in Fig.5, where horizontal axis represents the total bit rate including two texture videos and two depth videos for two original views, and vertical axis denotes the *PSNR* of synthesized views. It is seen that the *PSNR* for proposed method is bigger than those of other two methods[6,8], especially at



(a)

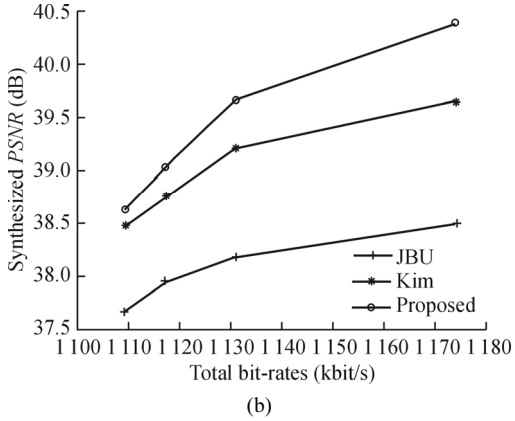


Fig.5 RD curves for (a) view 2.5 of Newspaper and (b) view 1.5 of Balloons using different methods

high bit rate. In order to compare *PSNR* for different methods given at the same bit rate, Bjontegaard Delta (BD) *PSNR* is used[12]. It can be observed in Tab.1 that BD-*PSNR*s of the proposed method are 0.32 dB, 0.57 dB, 0.52 dB and 0.43 dB bigger than those of Kim's method[8] for different views of two sequences, respectively.

Tab.1 BD-*PSNR* (dB) comparison of Kim's method and the proposed method

Sequence	View	Kim's method	Proposed method
Newspaper	2.5	0.68	1.00
	3.5	0.54	1.11
Balloons	1.5	1.17	1.69
	2.5	1.15	1.58

Targeting the application of multi-view coding, a novel classified-edge guided depth resampling method is proposed in this paper. Our contribution has three aspects. Firstly, compared with contrastive methods, four diagonal pairs can estimate the edge direction more accurately. Secondly, classified edges including credible edge and probable edge are interpolated by the selected diagonal pair, which can improve coding *PSNR*. Last but not least, the intensity measurement for each diagonal pair in the proposed method is parameterless.

References

[1] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan and T. Wiegand, IEEE Transactions on Circuits & Systems Video Technology **22**, 1669 (2012).

[2] L. Zhang, G. Tech, K. Wegner and S. Yea, Test Model 7 of 3D-HEVC and MV-HEVC, JCT-VC Document: JCT3V-G1005, Joint Collaborative Team on Video Coding, (2014).

[3] Zhao Zhen-jun, Shen Li-quan, Hu Qian-qian, Li Fei-fei and Zhang Zhao-yang, Journal of Optoelecrtonics·Laser **26**, 149 (2015). (in Chinese)

[4] He Wan-wen, An Ping, Wang Jian-xin, Zuo Yi-fan and Shen Li-quan, Journal of Optoelecrtonics·Laser **25**, 1565 (2014). (in Chinese)

[5] E. Ekmekcioglu, S. T. Worrall and A. M. Kondoz, Bit-Rate Adaptive Downsampling for the Coding of Multi-view Video with Depth Information, 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 137 (2008).

[6] J. Kopf, M. F. Cohen, D. Lischinsk and M. Uyttendaele, ACM Transactions on Graphics **26**, 673 (2007).

[7] Qingxiong Yang, Naremdra Ahuja, Ruigang Yang, Kar-Han Tan, Davis James, Culbertson Bruce, Apostolopoulos John and Gang Wang, IEEE Transactions on Image Processing **22**, 4841 (2013).

[8] S.-Y. Kim and Y.-S. Ho, IEEE Transactions on Consumer Electronics **58**, 971 (2012).

[9] Shao Feng, Jiang Gang-yi and Yu Mei, Optoelectronics Letters **5**, 232 (2009).

[10] D. Rusanovsky, K. Muller and A. Vetro, Common Test Conditions of 3DV Core Experiments, JCTVC Document: JCT3V-D1100, Joint Collaborative Team on Video Coding, (2013).

[11] C. Fehn, Proceedings of SPIE **5291**, 93 (2004).

[12] G. Bjøntegaard, Calculation of Average PSNR Differences between RD-Curves, ITU-T SG16 Q6 Document: 13th VCEG-M33 Meeting, (2001).