J Sign Process Syst (2017) 88:323–331
DOI 10.1007/s11265-016-1158-x

Depth Map Boundary Filter for Enhanced View Synthesis in 3D Video

Yunseok Song[1] · **Yo-Sung Ho**[1]

Received: 24 April 2016 / Accepted: 7 July 2016 / Published online: 23 August 2016

Abstract In 3D video systems, view synthesis is performed at the receiver end using decoded texture and depth videos. Due to errors and noise, boundary pixels at coded depth data exhibit inaccuracy, which affects the rendering quality. In this paper, we propose a boundary refinement filter for coded depth data. Initially, we estimate the boundary region based on gradient magnitudes, using its standard deviation as a threshold. Consecutively, we replace the depth value at the boundary region with a weighted average by means of the proposed filter. Three weights are calculated in this process: depth similarity, distance, and boundary direction. Experiment results demonstrate that the proposed filter increases the PSNR of synthesized images. The improvements are confirmed subjectively as well. Hence, the quality of synthesized images is enhanced, aided by the proposed depth map filter.

Keywords 3D video system · Depth map refinement · View synthesis

1 Introduction

3D video adds depth perception to 2D video, providing a realistic feel. The market for 3D video has grown extensively since the successes of numerous 3D commercial films in late 2000s. Now with more advanced technologies, content providers attract customers with marketable 3D content via 3D displays such as stereoscopic or auto-stereoscopic displays.

Generally, 3D video generation requires multiple views of texture and depth videos [1]. In a multi-view camera system, the number of cameras should not be too high due to cost and space restrictions. Due to this limitation, view synthesis is performed to generate images at virtual views using the existing data.

In the 3D video system, as shown in Fig. 1, depth data are acquired directly from depth cameras or from depth estimation using multi-view cameras [2, 3]. Yet, various factors cause inaccurate depth maps. Depth cameras may induce flickering and mixed pixels while depth estimation is prone to false estimation around object boundaries [4]. Furthermore, distortion occurs as a result of coding [5].

Furthermore, the 3D video system transmits compressed N views of texture and depth data. At the receiver end, M views are generated based on the N decoded views and synthesized views. Hence, the number of output views is always greater than that of input views. For view synthesis, decoded texture and depth data are used as input in depth image based rendering [6]. Thus, the quality of decoded data directly impacts the quality of rendered images.

Many studies have presented depth enhancement techniques. The bilateral filter is one of the most well-known filters for edge preservation and noise reduction [7]. This employs weights based on Euclidean distance and intensity difference of nearby pixels. Oh et al. have used occurrence, similarity, and distance to improve depth coding efficiency [8]. Dorea and de Queioriz have exploited color-based depth region merging [9]. Merkle et al. have introduced platelet-based depth coding [10].

In this paper, we propose a boundary filter to enhance depth data which affects the quality of view synthesis

✉ Yo-Sung Ho
hoyo@gist.ac.kr

Yunseok Song
ysong@gist.ac.kr

[1] Gwangju Institute of Science and Technology (GIST), 123 Cheomdangwagi-ro, Buk-gu, Gwangju 61005, Republic of Korea

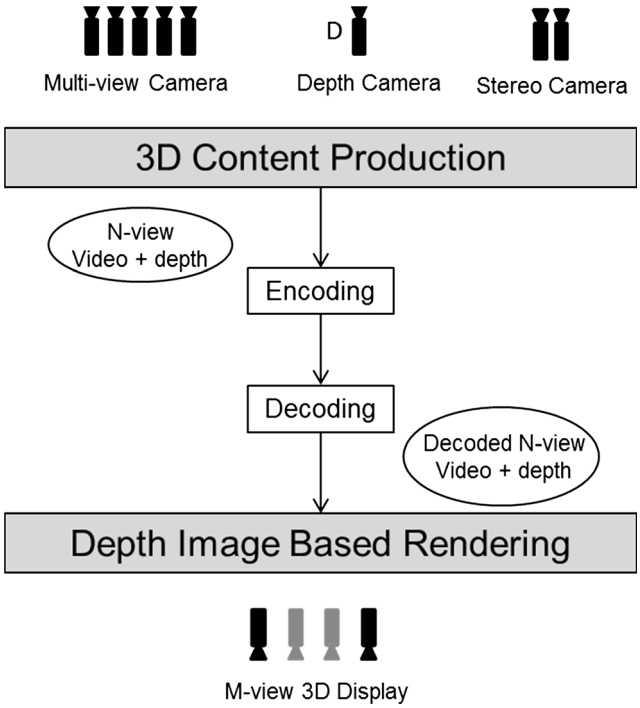


Figure 1 Framework of a 3D video system.

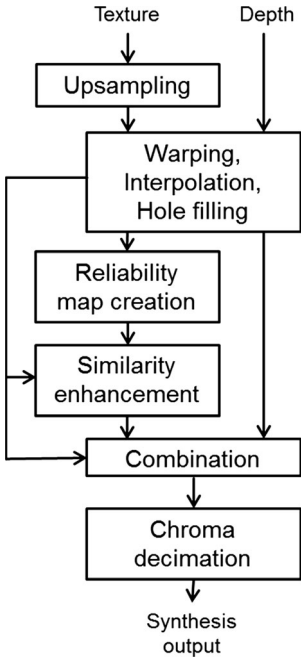


Figure 2 Flowchart of view synthesis.

results. We apply the proposed method to decoded depth videos at the receiver end and evaluate the quality of synthesized images.

2 Depth Map Processing in View Synthesis

This section briefly describes how depth data are used in view synthesis. Several view synthesis methods have been proposed; in this paper, we employ the method currently used in Joint Collaborative Team on 3D Video Coding Extension Development (JCT3V) standardization activity [11]. Figure 2 illustrates its flowchart. View synthesis procedure can be roughly divided to warping stage and combining stage for depth data.

Specifically, warping, interpolation, and hole filling are carried out line-wise with a processing direction of left-to-right. Due to this nature, this rendering method is known as Fast 1D View Synthesis. Depending on calculated warped intervals, interpolation or hole filling is performed. An interval is defined by warped positions of sample positions of the two neighboring input views. If the warped interval is greater than twice the sampling distance, i.e., disocclusion is assumed, hole filling is carried out based on left and right interval boundaries. Otherwise, interpolation is executed at full sample positions between the interval.

In the combination stage, two texture views extrapolated from the virtual view are combined. When blending the sample value for the synthesized view, depth difference of the left and right view is considered to determine whether to use the front sample or the back sample. Hence, the quality of depth map affects the synthesized output image; errors in the input image may propagate in the processing steps. Generally, the 2D video system does not require depth information when generating video content. However, the 3D video system uses depth information to support reality and immersion. Depth images can be acquired by depth cameras such as time-of-flight (ToF) cameras. Moreover, they can be estimated via stereo matching using captured color images from multi-view or stereoscopic camera systems.

3 Proposed Method

The objective of the proposed method is to refine the boundary region of depth maps so that the quality of synthesized images can be improved. Initially, the boundary region is estimated using the given depth data. Afterwards, the proposed boundary filter is applied to such a region. Figure 3 represents the steps of the proposed approach.

3.1 Boudnary Region Estimation

Due to the characteristics of depth data, sudden changes occur along the boundaries. The proposed method targets to modify only the boundary region, not the entire image. We use gradient information of the depth map to estimate the boundary region. For each sample in the image, x-direction and y-direction gradients are calculated using the Sobel operator. From

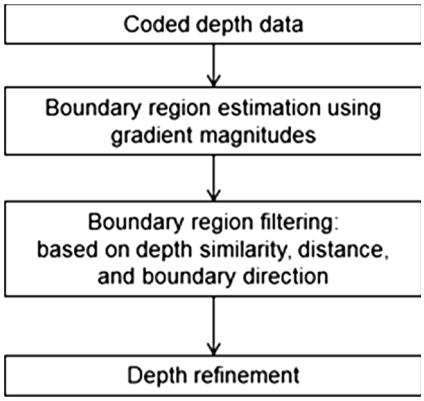


Figure 3 Steps of the proposed method.

the gradient values, we obtain the gradient magnitude. This is processed by (1), (2), and (3)

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} *A \tag{1}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} *A \tag{2}$$

$$G = \sqrt{Gx^2 + Gy^2} \tag{3}$$

where G_x, G_y, and G denote x-direction gradient, y-direction gradient, and gradient magnitude, respectively. A represents the source image.

In the first frame, we calculate the standard deviation of gradient magnitudes, which is set as a threshold throughout the successive frames. Pixels possessing gradient magnitudes higher than the threshold are determined as boundary samples. Figure 4 shows the estimated boundary region of "Dancer" sequence. Boundary region samples are represented as white in the binary map.

Consecutively, we assign direction to the estimated boundary samples. This direction labeling is necessary for the proposed filtering process. Four directions are defined: vertical, horizontal, 45° diagonal, and 135° diagonal. As shown in Fig. 5, directions are partitioned to have equal probabilities of being selected. In order to determine the direction, we use the gradient direction obtained by (4). Depending on the value of gradient direction d, one of the four directions is assigned.

Figure 4 Estimated boundary region of "Dancer".

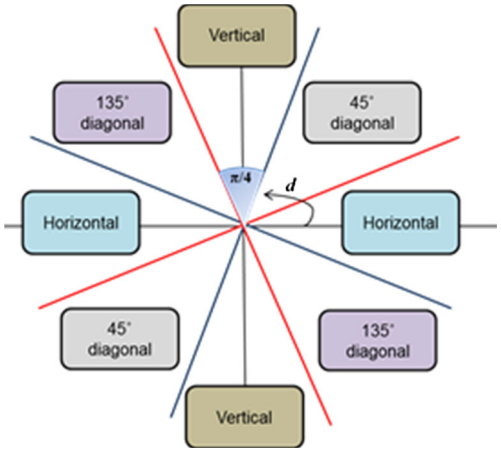


Figure 5 Partitions with respect to boundary directions.

$$d = \arctan\left(\frac{G_y}{G_x}\right) \tag{4}$$

3.2 Depth Refinement Based on Depth Similarity, Distance, and Boundary Direction

The proposed method applies window-based filtering on boundary samples. We assign new values considering three factors: depth similarity, distance, and boundary direction. In [8], occurrence, similarity, and distance weights are employed. The proposed method uses direction term, disregarding occurrence. The main drawback of occurrence term is that the weight calculation can be processed only after the histogram of the window is completed. Moreover, in the proposed method, distance weight is higher for pixels farther from the center pixel; this is the opposite case for both [8] and the bilateral filter.

For depth similarity, distance, and boundary direction, each factor is represented by a weight possessing a value between 0 and 1. The replacement depth is generated by (5) where the overall weight is the multiple of all weights, represented by (6)

$$D_{p,new} = \frac{\sum_q D_q w_{p,q}}{\sum_q w_{p,q}} \tag{5}$$

$$w_{p,q} = w\_sim_{p,q} \times w\_distance_{p,q} \times w\_direction_{p,q} \tag{6}$$

where D_p, D_q denote depth values of center pixel p and the neighboring pixel q, respectively. $D_{p,new}$ represents the replacement value for p. $w_{p,q}$ is the overall weight estimated by the relation of p and q. $w\_sim_{p,q}$, $w\_distance_{p,q}$, and

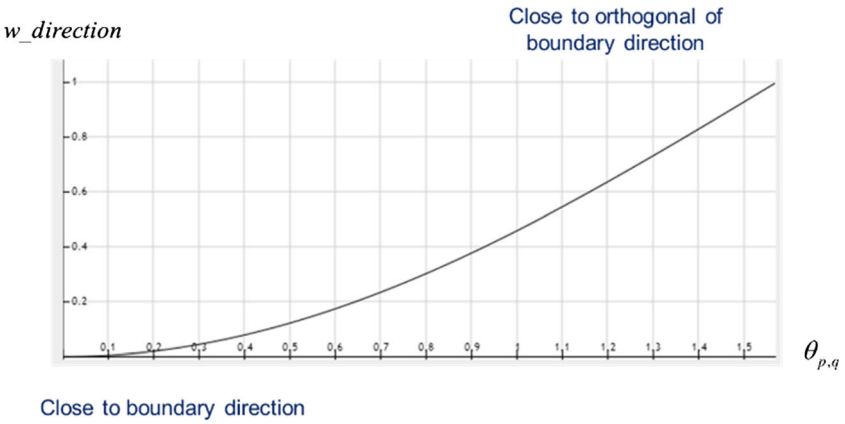


Figure 6 Direction weight distribution.

$w\_direction_{p,q}$ are weights of depth similarity, distance, and boundary direction, respectively.

First, similarity represents the difference depth values of p and q. Small difference means little deviation, which is more desirable. The similarity weight is defined by (7)

$$w\_sim_{p,q} = \exp\left(\frac{-|D_p - D_q|^2}{2\sigma_{sim}{}^2}\right) \tag{7}$$

where σ_{sim} represents the sigma in the Gaussian function.

Second, we consider the Euclidean distance between p and q, defined by (8). In the bilateral filter, shorter distance means higher weight. However, through experiments, we determined pixels farther from the center pixel are more reliable. This is due to the difference of target area; the proposed filter is only applied to the boundary region while the bilateral filter is used for the entire image. In coded depth maps the boundary region possesses inaccurate depth values.

$$w\_distance_{p,q} = 1 - \exp\left(\frac{-\left[(p_x - q_x)^2 + \left(p_y - q_y\right)^2\right]}{2\sigma_{distance}{}^2}\right) \tag{8}$$

where $\sigma_{distance}$ represents the sigma of the Gaussian function. p_x and p_y denote x- and y-coordinates of p, this applies to q as well.

Finally, we examine the location of q with taking the boundary direction of p into account. The boundary direction is acquired in the boundary region estimation process. Since noise and errors exist along the coded boundaries, we assign small weights to boundary direction samples. On the contrary, pixels located closer to the orthogonal direction of the boundary direction contain less error, and thus more reliable. The direction weight is represented by (9).

$$w\_direction_{p,q} = 1 - \cos\left(\theta_{p,q}\right), \quad 0 \le \theta_{p,q} \le \frac{\pi}{2} \tag{9}$$

where $\theta_{p,q}$ means the angle between p and q. A cosine function is applied to control the weight in between 0 and 1.

Further, as shown in Fig. 6, the weight function is designed to be nonlinear; neighboring pixels forming an angle less than $\pi/4$ are likely to be disregarded. Since (9) has a constraint on the range of angle, adjustment is necessary as shown in Fig. 7. Wherever the neighboring pixel is located, the angle generated from the boundary line can always be adjusted to be in between 0 and $\pi/2$.

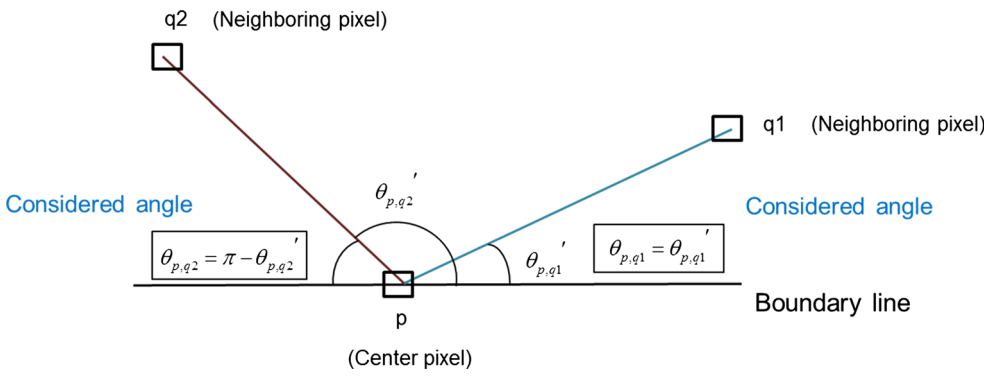
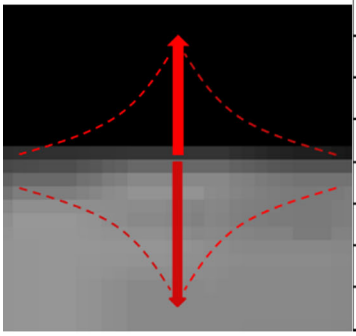


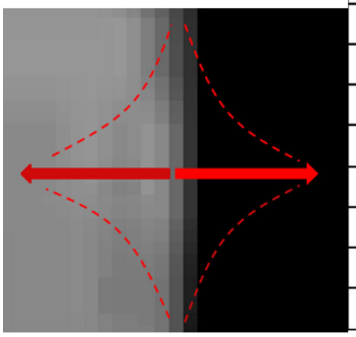
Figure 7 Angle adjustment for the direction term.

Figure 8 Direction weights according to boundary directions.



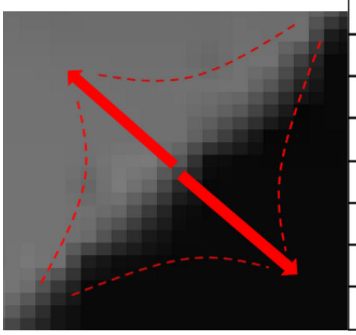
0.29	0.4	0.55	0.76	1	0.76	0.55	0.4	0.29
0.2	0.29	0.45	0.68	1	0.68	0.45	0.29	0.2
0.11	0.17	0.29	0.55	1	0.55	0.29	0.17	0.11
0.03	0.05	0.11	0.29	1	0.29	0.11	0.05	0.03
0	0	0	0	0	0	0	0	0
0.03	0.05	0.11	0.29	1	0.29	0.11	0.05	0.03
0.11	0.17	0.29	0.55	1	0.55	0.29	0.17	0.11
0.2	0.29	0.45	0.68	1	0.68	0.45	0.29	0.2
0.29	0.4	0.55	0.76	1	0.76	0.55	0.4	0.29

(a) Horizontal boundary and the direction weights.



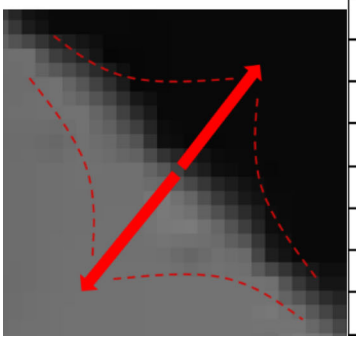
0.29	0.2	0.11	0.03	0	0.03	0.11	0.2	0.29
0.4	0.29	0.17	0.05	0	0.05	0.17	0.29	0.4
0.55	0.45	0.29	0.11	0	0.11	0.29	0.45	0.55
0.76	0.68	0.55	0.29	0	0.29	0.55	0.68	0.76
1	1	1	1	1	1	1	1	1
0.76	0.68	0.55	0.29	0	0.29	0.55	0.68	0.76
0.55	0.45	0.29	0.11	0	0.11	0.29	0.45	0.55
0.4	0.29	0.17	0.05	0	0.05	0.17	0.29	0.4
0.29	0.2	0.11	0.03	0	0.03	0.11	0.2	0.29

(b) Vertical boundary and the direction weights.



1	0.86	0.68	0.49	0.29	0.14	0.05	0.01	0
0.86	1	0.8	0.55	0.29	0.11	0.02	0	0.01
0.68	0.8	1	0.68	0.29	0.05	0	0.02	0.05
0.49	0.55	0.68	1	0.29	0	0.05	0.11	0.14
0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29
0.14	0.11	0.05	0	0.29	1	0.68	0.55	0.49
0.05	0.02	0	0.05	0.29	0.68	1	0.8	0.68
0.01	0	0.02	0.11	0.29	0.55	0.8	1	0.86
0	0.01	0.05	0.14	0.29	0.49	0.68	0.86	1

(c) Diagonal 45° boundary and the direction weights.



0	0.01	0.05	0.14	0.29	0.49	0.68	0.86	1
0.01	0	0.02	0.11	0.29	0.55	0.8	1	0.86
0.05	0.02	0	0.05	0.29	0.68	1	0.8	0.68
0.14	0.11	0.05	0	0.29	1	0.68	0.55	0.49
0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29
0.49	0.55	0.68	1	0.29	0	0.05	0.11	0.14
0.68	0.8	1	0.68	0.29	0.05	0	0.02	0.05
0.86	1	0.8	0.55	0.29	0.11	0.02	0	0.01
1	0.86	0.68	0.49	0.29	0.14	0.05	0.01	0

(d) Diagonal 135° boundary and the direction weights.

In Fig. 8, the broken lines indicate how direction weights are distributed for each direction. The arrows represent directions orthogonal to the boundary. Moreover, the direction weights are predetermined according to the window size. Figure 8 contains the direction weight tables. Weights at the boundary directions are zero, which means these values are totally disregarded. On the other hand, weights close to the orthogonal directions are high, greatly influencing the final weighted average.

4 Experiment Results

We conducted experiments on coded depth data to evaluate the performance of the proposed filter in comparison to the bilateral filter, one of the most widely used techniques. The proposed method was specifically applied to estimated boundary region while the bilateral filter was used in the entire image. In addition, the combination of them was tested as well. In the combined version, we applied the bilateral filter at the non-boundary region and the proposed filter at the boundary region.

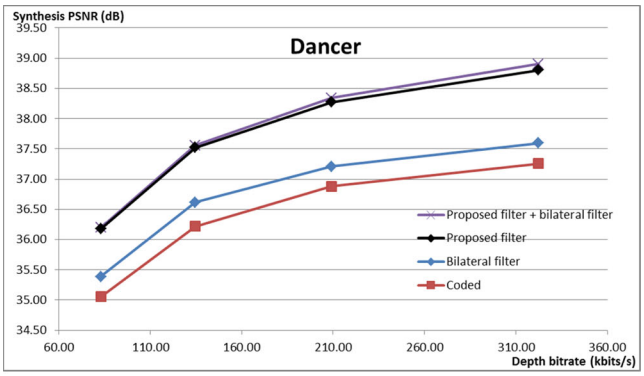
Filtering was performed on depth videos decoded from 3D extension of advanced video coding (3D-AVC). We used reference software developed by the joint collaborative team on 3D video coding extension development (JCT3V) [9, 10]. The reference data for synthesized data were created from the results of view synthesis using original texture and depth data. Furthermore, original depth data were used as reference for depth quality evaluation.

Tests were performed on 100 frames of "Dancer", "Poznan_Street", "Newspaper" and "Balloons" sequences. We used a window size of 9 × 9, and set the sigma of Gaussian as σ_{sim} = 10, and $\sigma_{distance}$ = 10; such values were empirically selected. Performance difference according to varying parameters was not significant. The rest of simulation conditions follows the settings used by JCT3V [11]. Table 1 shows the results of view synthesis and Fig. 9 displays the PSNR curves.

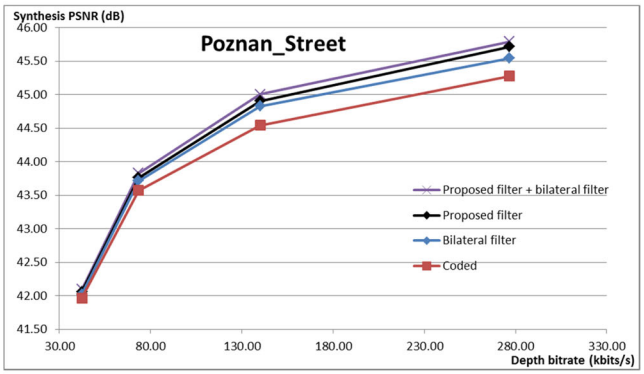
In terms of synthesized image quality, the combination of the bilateral filter and the proposed filter provides the highest performance, increased PSNR of 0.66 dB on average. The proposed filter itself shows 0.61 dB increase, very close to the results by the combined version. Hence, when the combination was used, most of the gain resulted from the proposed filter. This improvement was much higher than 0.28 dB increase by the bilateral filter case. Specifically, the proposed filter performs better at higher QPs. Naturally, depth data coded at higher QPs exhibit less noise and errors in general. Since distance and direction weights are fixed according to the window size, the accuracy of samples at highly weighted positions affect the overall quality.

Table 1 Results of view synthesis.

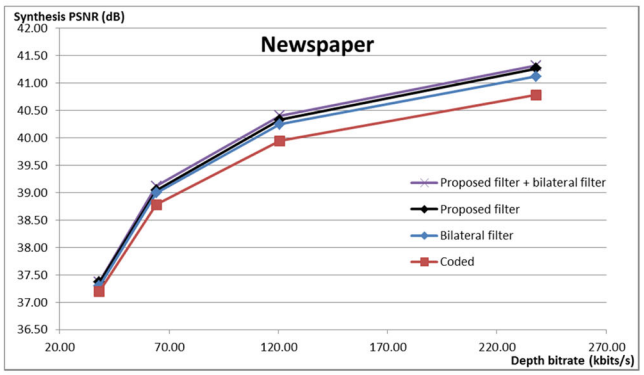
Sequence	QP	Depth bitrate (kbits/s)	Synthesis PSNR (dB)			
			Coded	Bilateral filter	Proposed filter	Proposed filter + bilateral filter
Dancer (1920 × 1088)	26	322.30	37.26	37.60	38.80	38.91
	31	209.13	36.88	37.21	38.27	38.34
	36	134.51	36.22	36.62	37.52	37.56
	41	82.90	35.06	35.39	36.18	36.20
Poznan_Street (1920 × 1088)	26	276.58	45.28	45.54	45.72	45.80
	31	140.05	44.55	44.83	44.91	45.01
	36	73.37	43.57	43.71	43.76	43.83
	41	42.39	41.96	42.02	42.06	42.10
Newspaper (1024 × 768)	26	238.11	40.78	41.12	41.27	41.32
	31	120.69	39.95	40.25	40.33	40.41
	36	64.24	38.78	39.00	39.04	39.12
	41	38.02	37.20	37.30	37.37	37.37
Balloons (1024 × 768)	26	303.98	43.54	44.03	44.34	44.32
	31	151.46	42.82	43.24	43.49	43.52
	36	77.45	41.91	42.20	42.33	42.36
	41	43.75	40.57	40.74	40.76	40.79
Average ΔPSNR				0.28	0.61	0.66



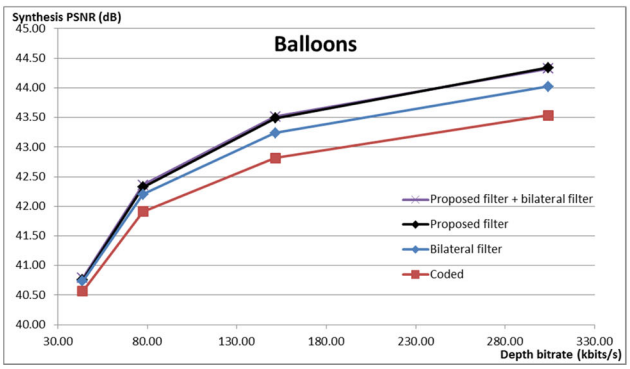
(a) Results of "Dancer" test



(b) Results of "Poznan_Street" test



(c) Results of "Newspaper" test



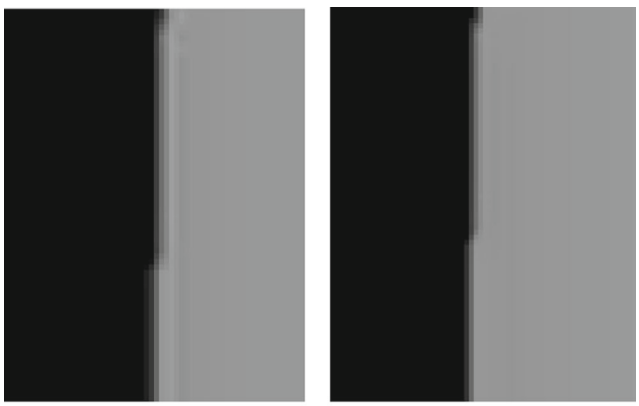
(d) Results of "Balloons" test

Figure 9 PSNR curves of synthesis results.

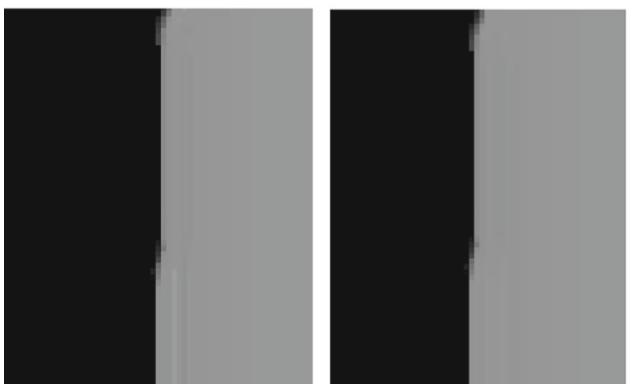
The performance of the boundary filter can be confirmed subjectively as well. Figures 10 and 11 represent boundary portion data of "Dancer's" depth view and synthesized view, respectively. Results are shown for unmodified coded depth, bilateral filter, proposed filter and the combination of two methods. In the depth data, the proposed filter removes errors around the depth boundary more effectively than the bilateral filter.

Consequently, in the synthesized images, boundary samples are more smoothly aligned presenting less distortion. Specifically, in the synthesized image of "Dancer", the proposed filter was able to connect the discontinuation observed from coded and bilateral filter cases. This explains why the PSNR increase for "Dancer" is much higher than those of other sequences. Overall, the synthesis results from the proposed filter and the combination are very similar, i.e., most of the benefits are achieved by the proposed filter itself.

Further, the performance of the proposed filter was compared with that of Oh et al.'s filter in [6]. Unfiltered data means the decoded depth data that has



(a) Coded (b) Bilateral filter



(c) Proposed filter (d) Combined (proposed + bilateral)

Figure 10 Results of depth filtering, "Dancer", QP26, view 5.

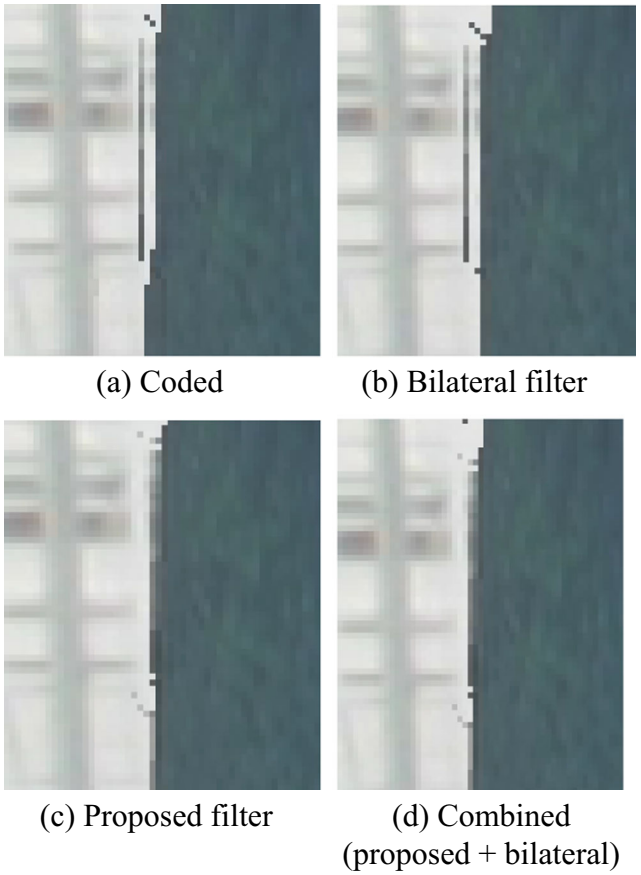


Figure 11 Results of view synthesis, "Dancer", QP26, view 4.

not been filtered at all. For view synthesis, original texture data were used for all cases since we only focus on depth data in this paper. Table 2 represents the results. Compared with unfiltered data, the proposed filter showed improvement of 0.58 dB and 0.41 dB for synthesized and depth data, respectively. Although Oh et al.'s method also presented enhancement, the effect was not as beneficial as the proposed method.

5 Conclusion

In this paper, we proposed a depth boundary filter to improve the quality of synthesized images. First, we estimate the boundary region using the standard deviation of gradient magnitudes. Then, we performe window-based filtering on boundary samples using weights based on depth similarity, distance, and boundary direction. Experiment results reported that the proposed filter increased the PSNR of synthesized images by 0.61 dB on average. When the proposed method was combined with non-boundary region bilateral filtering, 0.66 dB increase of average PSNR was achieved. The improvement was confirmed subjectively as well. We also showed that the proposed filter performed more efficient than the state-of-the-art filter.

Table 2 Performance evaluation in terms of comparison with [6].

Sequence	QP	Synthesis PSNR (dB)			Depth PSNR (dB)		
		Unfiltered	Oh et al. [6]	Proposed	Unfiltered	Oh et al. [6]	Proposed
Dancer (1920 × 1088)	26	37.26	37.59	38.64	42.29	42.58	44.74
	31	36.88	37.26	38.16	41.24	41.46	42.86
	36	36.22	36.65	37.43	39.14	39.26	39.96
	41	35.06	35.41	36.10	35.89	35.93	36.17
Poznan_Street (1920 × 1088)	26	45.28	45.50	45.72	43.10	43.14	43.15
	31	44.55	44.71	44.90	41.13	41.12	41.15
	36	43.57	43.64	43.75	38.60	38.59	38.65
	41	41.96	41.94	42.06	35.42	35.37	35.47
Newspaper (1024 × 768)	26	40.78	41.02	41.27	36.87	36.89	37.35
	31	39.95	40.13	40.33	34.90	34.88	35.10
	36	38.78	38.87	39.04	32.36	32.31	32.39
	41	37.20	37.23	37.36	29.50	29.46	29.49
Balloons (1024 × 768)	26	43.54	44.07	44.38	37.98	37.96	38.35
	31	42.82	43.22	43.50	35.89	35.85	36.04
	36	41.91	42.16	42.33	33.33	33.28	33.35
	41	40.57	40.68	40.76	30.60	30.54	30.58
Average ΔPSNR (comparison with unfiltered data)		–	0.23	0.58	–	0.02	0.41

Acknowledgments This research was supported by the'Cross-Ministry Giga KOREA Project' of the Ministry of Science, ICT and Future Planning, Republic of Korea(ROK). [GK16C0100, Development of Interactive and Realistic Massive Giga-Content Technology].

References

1. Smolic, A., Muller, K., Merkle, P., Fehn, C., Kauff, P., Eisert, P., & Wiegand, T. (2006). 3D video and free viewpoint video - technologies, applications and MPEG standards. In Proc. IEEE International Conference on Multimedia and Expo (pp. 2161–2164).
2. Fehn, C. (2004). Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV. In Proc. Electronic Imaging (pp. 93–104).
3. Mori, Y., Fukushima, N., Yendo, T., Fujii, T., & Tanimoto, M. (2009). View generation with 3D warping using depth information for FTV. *Signal Processing: Image Communication, 24*, 65–72.
4. Kim, W. S., Ortega, A., Lai, P., Tian, D., & Gomila, C. (2009). Depth map distortion analysis for view rendering and depth coding. In Proc. IEEE International Conference on Image Processing (pp. 721–724).
5. Tomasi, C., & Manduchi, R. (1998). Bilateral filtering for gray and color images. In Proc. of IEEE International Conference on Computer Vision (pp. 839–846).
6. Oh, K. J., Vetro, A., & Ho, Y. S. (2011). Depth coding using a boundary reconstruction filter for 3-D video systems. *IEEE Transactions on Circuits and Systems for Video Technology., 21*(3), 350–359.
7. Dorea, C., & de Queioriz, R. L. (2011). Depth map reconstruction using color-based region merging. In Proc. IEEE International Conference on Image Processing (pp. 1977–1980).
8. Merkle, P., Morvan, Y., Smolic, A., Muller, K., de With, P. H. N., & Wiegand, T. (2008). The effect of depth compression on multiveiew rendering quality. In Proc. of 3DTV Conference (pp. 245–248).
9. 3DV-ATM 13.0, http://mpeg3dv.research.nokia.com/svn/mpeg3dv/tags/3DV-ATMv13.0.
10. VSRS-1D-Fast, https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-13.0.
11. Rusanovskyy, D., Muller, K., & Vetro, A. (2013). Common test conditions of 3DV core experiments. ITU-T/ISO/IEC JCT3V, C1100.

Yunseok Song received his B.S. degree in electrical engineering from Illinois Institute of Technology in 2008, M.S. degree in electrical engineering from University of Southern California in 2009. He is pursuing a Ph.D. degree in the School of Electrical Engineering and Computer Science at Gwangju Institute of Science and Technology (GIST) in Korea. His research interests include digital video coding, 3D broadcasting, and multi-view image processing.

Yo-Sung Ho received both B.S. and M.S. degrees in electronic engineering from Seoul National University, Korea, in 1981 and 1983, respectively, and Ph.D. degree in Electrical and Computer Engineering from the University of California, Santa Barbara, in 1990. He joined the Electronics and Telecommunications Research Institute (ETRI), Korea, in 1983. From 1990 to 1993, he was with Philips Laboratories, Briarcliff Manor, New York, where he was involved in development of the advanced digital high-definition television (AD-HDTV) system. In 1993, he rejoined the technical staff of ETRI and was involved in development of the Korea direct broadcast satellite (DBS) digital television and high-definition television systems. Since 1995, he has been with Gwangju Institute of Science and Technology (GIST), where he is currently a professor in the School of Electrical Engineering and Computer Science. His research interests include digital image and video coding, image analysis and image restoration, advanced coding techniques, digital video and audio broadcasting, 3D television, and realistic broadcasting.