Efficient Depth Map Recovery Using Concurrent Object Boundaries in Texture and Depth Images

Se-Ho Lee*, Tae-Young Chung*, Jae-Young Sim†, and Chang-Su Kim*
*School of Electrical Engineering, Korea University, Seoul, Korea
E-mails: {seholee, tychung, cskim}@mcl.korea.ac.kr
†Ulsan National Institute of Science and Technology, Ulsan, Korea
E-mail: jysim@unist.ac.kr

***Abstract*—An efficient depth map recovery algorithm, using concurrent object boundaries in texture and depth signals, is proposed in this work. We first analyze the effects of a distorted depth map on the qualities of synthesized views. Based on the analysis, we propose an object boundary detection scheme to restore sharp boundaries from a distorted depth map. Specifically, we initially estimate object boundaries from a depth map using the gradient magnitude at each pixel. We then multiply the gradient magnitudes of texture and depth pixels. Then, we suppress boundary pixels with non-maximum magnitudes and refine the object boundaries. Finally, we filter depth pixels along the gradient orientations using a median filter. Experimental results show that the proposed algorithm significantly improves the qualities of synthesized views, as compared with conventional algorithms.**

I. Introduction

Recently, 3DTV [1] and free-viewpoint TV (FTV) [2] technologies have been developed to render realistic scenes or multiple viewpoints. A 3DTV enables the depth perception of a scene by displaying two views captured from different viewpoints. An FTV allows a user to navigate any viewpoint of a scene. To provide more realistic 3D sensation or the freedom to choose more viewpoints, many views of the same scene are required. However, the transmission or storage of a multi-view video sequence requires a wide bandwidth or huge storage space, even though a lot of efforts have been made for the compression [3]. To overcome this drawback, the multi-view video plus depth (MV+D) format was proposed [4]. An MV+D video sequence consists of two or more texture videos and their corresponding depth videos. Using the depth information, the decoder can generate arbitrary views based on the depth image based rendering (DIBR) [5]. This MV+D format facilitates the rendering of arbitrary views, while occupying relatively little storage space. Thus, the MV+D format has become one of the major representation schemes for 3D visual information. However, distortions in the depth information may lead to wrong rendering results, degrading the qualities of synthesized views severely.

This work was supported partly by the National Research Foundation of Korea(NRF) grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (No. 2009-0083495), and partly by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government (MSIP) (2011-0031648).

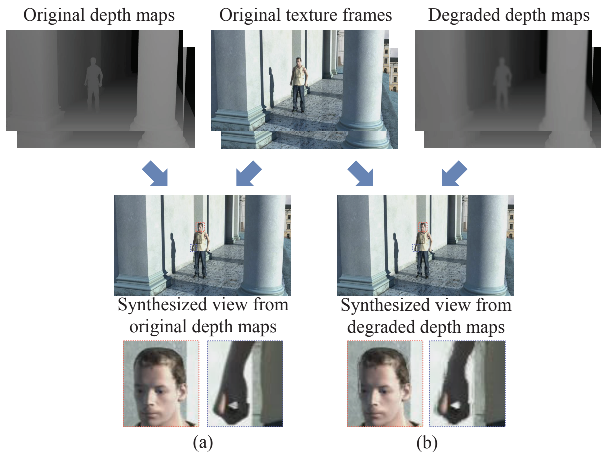
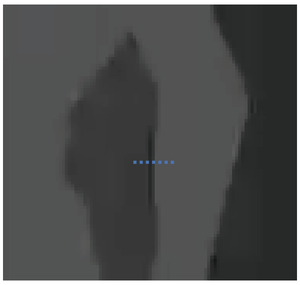


Fig. 1. View synthesis for the “Undo_Dancer” sequence with and without depth distortions: (a) synthesized view using original texture frames and original depth maps and (b) synthesized view using original texture frames and distorted depth maps.

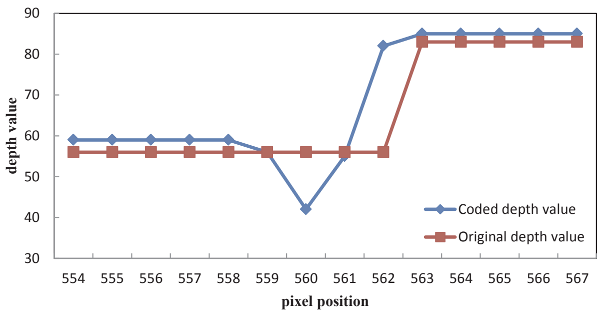
Merkel *et al*. [6] showed that compression artifacts on depth data affect the qualities of synthesized views adversely. Many researches have been carried out to restore depth maps from compression distortions and reduce the quality degradation of synthesized views. Liu *et al*. [7] proposed a joint trilateral filter which is applied as an in-loop filter in the video compression system. Their filter recovers a depth map by exploiting the structure similarity between a depth frame and the corresponding texture frame. However, it may fail to filter a depth map correctly if the texture structure is different from the depth structure. Oh *et al*. [8] proposed a depth boundary reconstruction filter, which enhances a depth map by considering three factors between a pixel and its neighboring pixels: closeness, depth difference, and occurrence frequency. After applying their filter, they further used a bilateral filter to remove remaining noise components.

In this work, we propose a novel depth map recovery algorithm to reduce compression artifacts and to improve the qualities of synthesized views. First, we estimate initial object boundaries based on the gradients of an input depth map. Then, we refine the object boundaries by applying the non-maximum suppression to the gradients of both texture and depth jointly. Finally, we recover the depth map by employing a median filter near the object boundaries. Simulation results demonstrate that

(a)



(b)



(c)

Fig. 2. Impacts of compression distortions on depth maps: (a) original depth map, (b) distorted depth map, and (c) depth values along the red line and the blue line in (a) and (b), respectively.

the proposed algorithm restores a higher quality depth map and thus synthesizes intermediate views more faithfully than the conventional algorithms [7], [8].

The rest of this paper is organized as follows. Section II analyzes depth map distortions. Section III describes the proposed algorithm. Section IV compares the proposed algorithm with the conventional algorithms experimentally. Finally, Section V concludes the paper.

II. Depth Map Distortions

Fig. 1 compares a synthesized view using original depth maps in the "Undo_Dancer" sequence with that using distorted depth maps. We see that the distortions in the depth maps degrade the synthesized view severely, even though the texture frames are intact. The quality degradation is especially noticeable near object boundaries.

Notice that a typical depth map is composed of homogeneous regions, which are separated by sharp object boundaries. However, due to compression distortions, such as blurring and ringing artifacts, sharp boundaries are distorted and blurred. We should recover sharp object boundaries from a distorted depth map. For example, Fig. 2 shows the impacts of compression distortions on a depth map. Fig. 2(a) is an original depth map, which is homogenous except for the boundaries. Fig. 2(b) is a distorted depth map due to the compression, in which boundaries are blurred. Fig. 2(c) compares the depth values along the red dotted line in Fig. 2(a) and those along the blue dotted line in Fig. 2(b). Our objective is to recover the original red signal, which is similar to a step function, from the distorted blue signal with blurring and ringing artifacts.

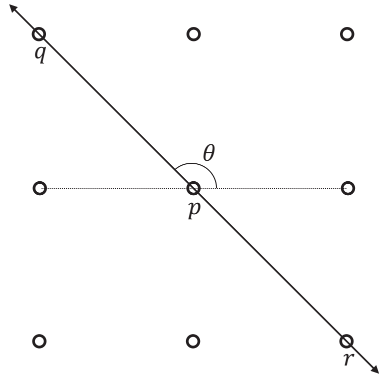


Fig. 3. Non-maximum suppression along a gradient orientation.

III. Proposed Algorithm

To recover object boundaries in a depth map, we use the information in the texture frame as well as the depth map. We first estimate the gradients of depth map and extract initial object boundaries. Then, we refine the initial object boundaries by applying the non-maximum suppression to depth and texture gradients jointly. Finally, we recover the depth map by applying a median filter to pixels near the object boundaries.

A. Estimation of Initial Object Boundaries

We first apply a Gaussian filtering to an input depth map to eliminate noise components. Then, using the Sobel operator [9], we calculate the first derivatives in the horizontal direction, $G_{D_x}(p)$, and the vertical direction, $G_{D_y}(p)$, at each pixel p in the depth map. Then, the magnitude and the orientation of the depth gradient are given by

$$|G_D(p)| = \sqrt{G^2_{D_x}(p) + G^2_{D_y}(p)}, \tag{1}$$

$$\theta(p) = \arctan\left(\frac{G_{D_y}(p)}{G_{D_x(p)}}\right). \tag{2}$$

Pixel p is declared to belong to an initial object boundary, if the magnitude $|G_D(p)|$ of the depth gradient is greater than a threshold τ_D, which is fixed to 20 in this work. The orientation $\theta(p)$ is quantized into one of the four angles, $0°$, $45°$, $90°$ or $135°$, to facilitate the remaining steps. Also, similarly to (1), we compute the magnitude $|G_T(p)|$ of the texture gradient at each pixel p in the texture frame.

B. Refinement of Object Boundaries

To refine object boundaries, we adopt a non-maximum suppression approach and find local maximums of gradient magnitude. If a pixel belongs to a proper object boundary, its gradient magnitude should be a local maximum along the gradient orientation. For example, in Fig. 3, let q and r denote the neighboring pixels of p along the gradient orientation $\theta(p)$. Suppose that p is an object boundary pixel. Then, we expect that $|G_D(p)| \geq |G_D(q)|$ and $|G_D(p)| \geq |G_D(r)|$ in the depth map, and that $|G_T(p)| \geq |G_T(q)|$ and $|G_T(p)| \geq |G_T(r)|$ in the texture frame.

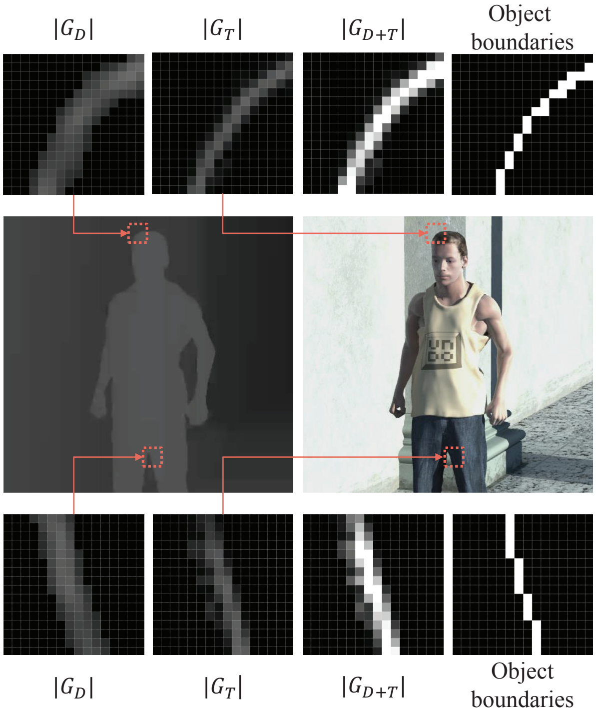


Fig. 4. Examples of the object boundary refinement based on the non-maximum suppression.

Based on this property, we first obtain a local maximum $|G_D|_{\max}$ of gradient magnitude along the gradient orientation of each pixel p. Then, we normalize the magnitudes of the depth gradients of neighboring pixels by $|G_D|_{\max}$. Similarly, we normalize the magnitudes of the texture gradients of neighboring pixels by $|G_T|_{\max}$. Then, we calculate the product of the normalized depth and texture gradients as follows.

$$|G_{D+T}(s)| = \frac{|G_D(s)|}{|G_D|_{\max}} \times \frac{|G_T(s)|}{|G_T|_{\max}}, \quad s \in \mathcal{N}_{\theta(p)} \tag{3}$$

where $\mathcal{N}_{\theta(p)}$ is the set of neighboring pixels of p along the orientation $\theta(p)$. If $|G_{D+T}(p)| \geq |G_{D+T}(q)|$ and $|G_{D+T}(p)| \geq |G_{D+T}(r)|$, then p is selected as an object boundary pixel. Otherwise, it is suppressed and eliminated from the set of boundary pixels. In this way, each refined boundary pixel tends to achieve local maximums of gradient magnitude in both the depth map and the texture frame simultaneously.

Fig. 4 illustrates this procedure of object boundary refinement based on the non-maximum suppression. We see that, by combining the depth and the texture information, the proposed algorithm identifies the object boundaries accurately.

C. Depth Map Recovery

After obtaining the object boundary information, we apply a median filter to neighboring pixels of each boundary pixel p^*. More specifically, we select M neighboring pixels along the gradient orientation $\theta(p^*)$ and the opposite orientation $-\theta(p^*)$ separately, as depicted by the red and the blue circles in Fig. 5, respectively. In this work, we set $M = 4$. Then, to each selected pixel, we apply the 7-tap median filter along the gradient orientation to recover the depth value.

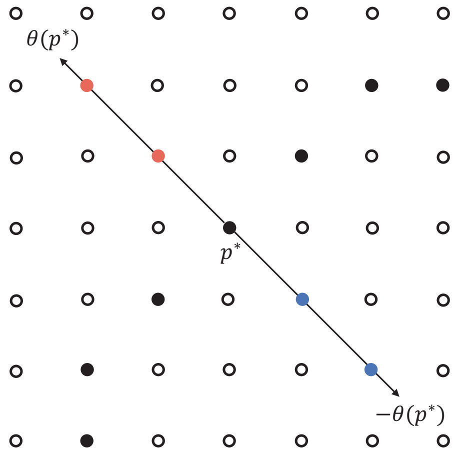


Fig. 5. Depth map recovery: black circles depict object boundary pixels, and red and blue circles are the pixels to be filtered with the 7-tap median filter along the gradient orientation.

IV. Experimental Results

We evaluate the performance of the proposed depth map recovery algorithm on the "Undo_Dancer" (1920 $\times$ 1088) sequence and the "Mobile" (720 $\times$ 528) sequence. The depth maps are encoded by the 3DV-ATM reference software [10] with three quantization parameters (QP's) 41, 46 and 50. Then, we recover the depth maps from the compression distortions, by employing the trilateral filter [11], the Oh *et al.*'s algorithm [8], and the proposed algorithm. Then, using the recovered depth maps, we synthesize intermediate views with the view synthesis reference software (VSRS) 3.0 [12]. Each synthesized view is compared with the original intermediate view, and the PSNR is computed.

Table I compares the PSNR performances of the proposed algorithm with those of the trilateral filter [11] and the Oh *et al.*'s algorithm [8]. We see that the proposed algorithm provides higher PSNR's than the conventional algorithms in most cases. For example, Fig. 6 shows the reconstructed depth frames and the synthesized intermediate views of the "Mobile" sequence. Figs. 6(a)∼(e) and (f)∼(j) show the depth maps of view 3 and view 5, respectively. Fig. 6(k) is the original frame at the intermediate view 4. Fig. 6(l) shows the synthesized frame for view 4 using the decoded depth maps, which contain the compression artifacts. Figs. 6(m)∼(o) show the synthesized frames using the recovered depth maps, which are obtained by the trilateral filter, the Oh *et al.*'s algorithm, and the proposed algorithm, respectively. Note that, although the conventional algorithms reduce the effects of the compression distortions in the depth maps and the synthesized views, they still yield visually annoying artifacts near the object boundaries. In contrast, the proposed algorithm recovers the object boundaries more accurately and thus synthesizes the intermediate view more faithfully without noticeable artifacts.

V. Conclusions

In this work, we proposed a depth map recovery algorithm, which enhances the quality of depth maps degraded by com-

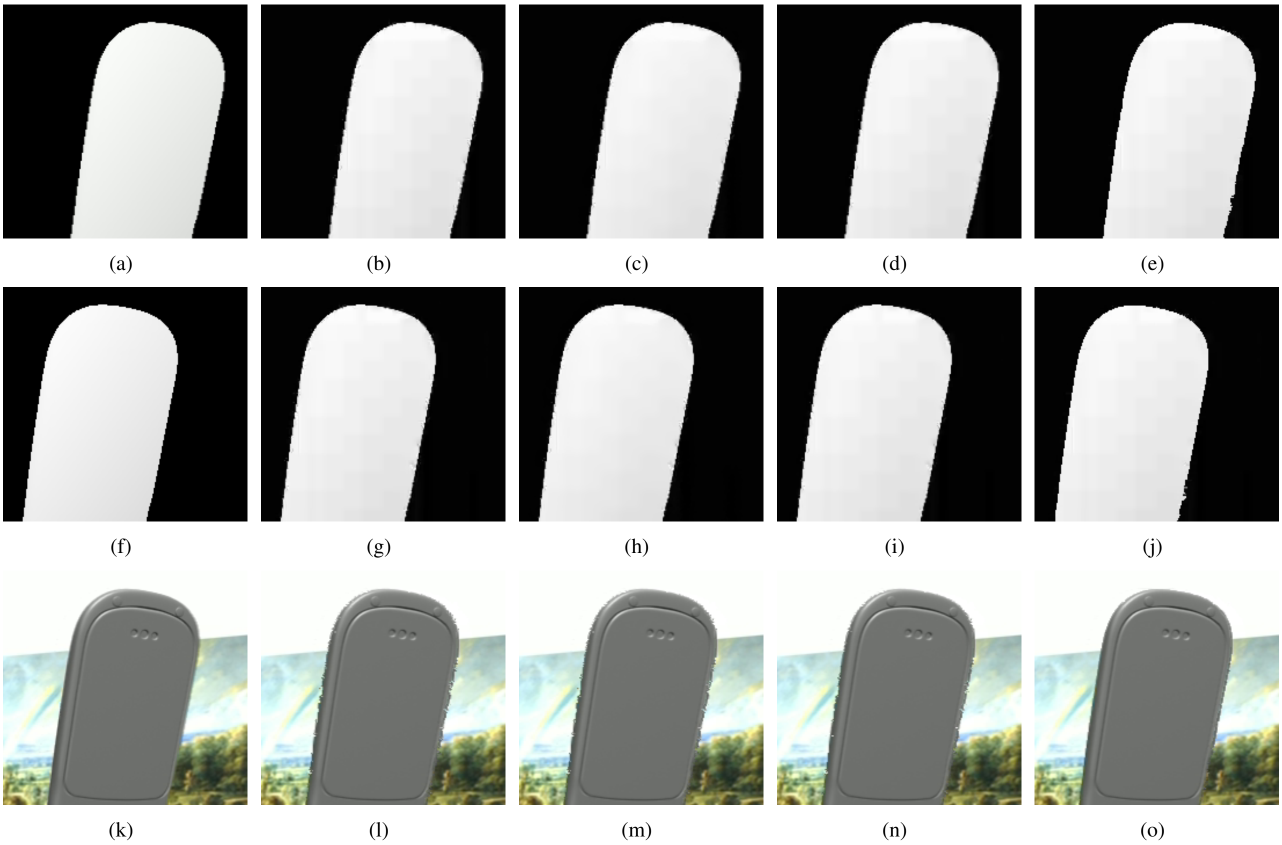


Fig. 6. Reconstructed depth maps and synthesized frames for the "Mobile" sequence. From top to bottom: depth maps for view 3, depth maps for view 5, and synthesized texture frames for view 4. Subfigures (a), (f), (k) are original depth maps and texture frames. Subfigures (b), (g), (l) are obtained using the encoded depth maps with QP 41. Subfigures (c), (h), (m) are obtained by the trilateral filter [11], subfigures (d), (i), (n) by the Oh *et al.*'s algorithm [8], and subfigures (e), (j), (o) by the proposed algorithm, respectively.

TABLE I
COMPARISON OF THE PSNR'S OF THE SYNTHESIZED VIEWS, RECONSTRUCTED BY THE TRILATERAL FILTER [11], THE OH *et al.*'S ALGORITHM [8], AND THE PROPOSED ALGORITHM.

Video sequence	QP	Degraded	Trilateral	Oh *et al.*	Proposed
Undo_Dancer	41	34.08	34.27	34.12	**34.57**
	46	32.55	32.72	32.59	**33.10**
	50	31.72	31.81	31.87	**32.07**
Mobile	41	40.30	40.46	**40.77**	40.61
	46	39.51	39.59	39.79	**40.24**
	50	39.06	39.15	39.22	**39.90**

pression distortions. The proposed algorithm first estimates initial object boundaries from the distorted depth map and refines them by considering the corresponding boundaries in the texture frame. It employs the non-maximum suppression technique to determine the concurrent object boundaries. Finally, the proposed algorithm reconstructs the depth values near the estimated object boundaries using the directional median filter. Experimental results demonstrated that the proposed algorithm provides higher quality synthesized views than the conventional algorithms [8], [11].

REFERENCES

[1] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "Multiview imaging and 3DTV," *IEEE Signal Process. Mag.,* vol. 24, no. 6, pp. 10-21, Nov. 2007.

[2] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint TV," *IEEE Signal Process. Mag.,* vol. 28, no. 1, pp. 67-76, Jan. 2011.

[3] M. Flierl and B. Girod, "Multi-view video compression," *IEEE Signal Process. Mag.,* vol. 24, no. 6, pp. 66-76, Nov. 2007.

[4] Joint Video Team JVT of ISO/IEC MPEG ITU-T VCEG, *Multi-View Video Plus Depth (MVD) Format for Advanced 3D Video Systems,* JVT-W100, San Jose, USA, Apr. 2007.

[5] P. Kauff, N. Atzpadin, C. Fehn, M. Müller O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Processing: Image Commun.,* vol. 22, no. 2, pp. 217-234, Feb. 2007.

[6] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Multi-view video plus depth representation and coding," *in Proc. IEEE ICIP,* Sept. 2007, vol. 1, pp. 201-204.

[7] S. J. Liu, P. L. Lai, D. Tian, and C. W. Chen, "New depth coding techniques with utilization of corresponding video," *Broadcasting, IEEE Transactions on,* vol. 57, no. 2, pp. 551-561, June 2011.

[8] K. J. Oh, A. Vetro, and Y. S. Ho, "Depth coding using a boundary reconstruction filter for 3-D video systems," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 21, no. 3, pp. 350-359, Mar. 2011.

[9] I. Sobel, *An Isotropic* 3×3 *Gradient Operator, Machine Vision for Three Dimensional Scenes,* H. Freeman, Academic Pres, NY, 1990.

[10] ISO/IEC JTC1/SC29/WG11, *Test Model for AVC Based 3D Video Coding, MPEG2012/N12558,* Feb. 2012.

[11] D. Min, J. Lu, and M. N. Do, "Depth video enhancement based on weighted mode filtering," *IEEE Trans. Image Process.,* vol. 21, no. 3, pp. 1176-1190, Mar. 2012.

[12] ISO/IEC JTC1/SC29/WG11, *Reference Software for Depth Estimation and View Synthesis,* Doc. N15377, Apr. 2008.