MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Hole-Filling Method Using Depth Based In-Painting For View Synthesis in Free Viewpoint Television (FTV) and 3D Video

Kwan-Jung Oh, Sehoon Yea, Yo-Sung Ho

TR2009-025 June 2009

Abstract

Depth image-based rendering (DIBR) is generally used to synthesize a virtual view in free viewpoint television (FTV) and 3D video. One of the key techniques in DIBR is how to fill the holes caused by disocclusion regions and wrong depth values. In this paper, we propose a new hole-filling method using depth-based in-painting. The proposed method is designed by combining the depth-based-hole-filling and in-painting. From the experiments, we confirm that the proposed hole-filling method provides better rendering quality objectively and subjectively.

Picture Coding Symposium 2009

HOLE-FILLING METHOD USING DEPTH BASED IN-PAINTING FOR VIEW SYNTHESIS IN FREE VIEWPOINT TELEVISION (FTV) AND 3D VIDEO

Kwan-Jung Oh, Sehoon Yea, and Yo-Sung Ho*

School of Information and Mechatronics
Gwangju Institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712 Republic of Korea
Mitsubishi Electric Research Laboratories (MERL)*
201 Broadway, Cambridge, MA 02139, USA*
{kjoh81, hoyo}@gist.ac.kr, yea@merl.com*

ABSTRACT

Depth image-based rendering (DIBR) is generally used to synthesize a virtual view in free viewpoint television (FTV) and 3D video. One of the key techniques in DIBR is how to fill the holes caused by disocclusion regions and wrong depth values. In this paper, we propose a new hole-filling method using depth-based in-painting. The proposed method is designed by combining the depth-based hole-filling and the in-painting. From the experiments, we confirm that the proposed hole-filling method provides better rendering quality objectively and subjectively.

Index Terms— DIBR, View Synthesis, FTV, 3D Video, Hole-filling, In-painting

1. INTRODUCTION

Free viewpoint television (FTV) and 3D video provide both realistic 3D impression and free view navigation of the scene to users and are now considered key technologies that could spur a next wave of multimedia experiences such as 3D cinema, 3D broadcasting, 3D displays, and 3D mobile services etc [1]-[3].

Among the key technical building blocks of the whole processing chain that constitutes such systems are coding and rendering. The role of efficient coding becomes much more important in these 3D systems due to the drastic potential increase in the volume of data. Some of the past researches and standardization efforts to address this issue include MPEG-2 multiview video profile (MVP) [4], MPEG-4 multiple auxiliary component (MAC) [5], and MPEG/JVT multiview video coding (MVC) [6].

On the other hand, given the ever increasing diversity of forms of 3D services and displays, proper rendering of 3D views is indispensable that could meet the requirements that enable the key features of the application at hand. In other words, it becomes necessary to resample the views and resize each view depending on the number of views and resolutions required by the display, respectively. The case when there are more views to be rendered at the display than are actually coded such as FTV, the resampling means generation of virtual views based upon the actual views. The problem of generating an arbitrary view of a 3D scene has been heavily addressed in the area of computer graphics.

Among the techniques for rendering, image-based rendering (IBR) techniques [7] have drawn much attention lately for rendering real world scenes. These techniques use image rather than geometry as primitives for rendering virtual views and often are classified into three categories depending on how much geometric information is used: rendering without geometry, with explicit geometry, and with implicit geometry. The 3D warping and layered-depth-images (LDI) belongs to the second category whereas view morphing and view interpolation belong to the third category.

Recently, the moving picture experts group (MPEG) has initiated a new exploration experiment (EE) specifically targeted towards FTV applications. Whereas the previous MPEG/JVT standardization activities for MVC focused on improvement of compression efficiency for generic multi-view coding scenarios, this new EE activities currently focus on depth estimation and rendering and reference softwares [8] are released.

The reference software for rendering exploits an in-painting technique to fill the holes occurred during view synthesis. However, since the existing hole-filling technique does not consider the disocclusion property, it wrongly fills the holes located on the border between a foreground and a background from both a foreground and a background neighboring pixels even though it naturally fills the other holes. To complement this defect, we propose a new hole-filling method using depth-based in-painting. The proposed hole-filling technique fills the holes only from neighborhood located on background by distinguishing the foreground and background using depth value.

2. OVERVIEW OF VIEW SYNTHESIS

The schematic diagram of a typical depth-based view synthesis system is shown in Fig. 1. The goal of such a system is to synthesize a virtual view from its neighboring views using the camera parameters, texture images, and depth images.

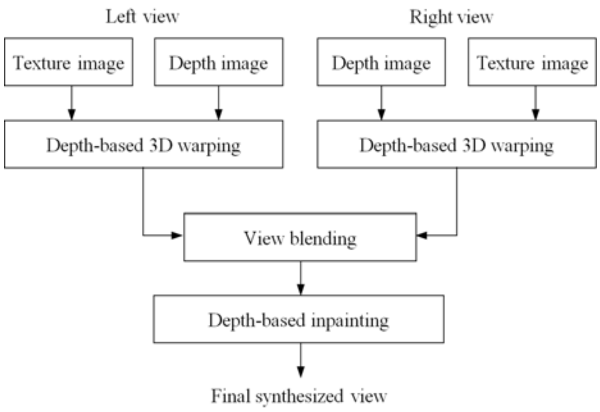


Fig. 1. Depth image-based virtual view synthesis

The three-dimensional image warping (3D warping) is a key technique in depth-based view synthesis. In 3D warping, pixels in reference image are back-projected to 3D spaces, and re-projected onto the target viewpoint as shown in Fig. 2.

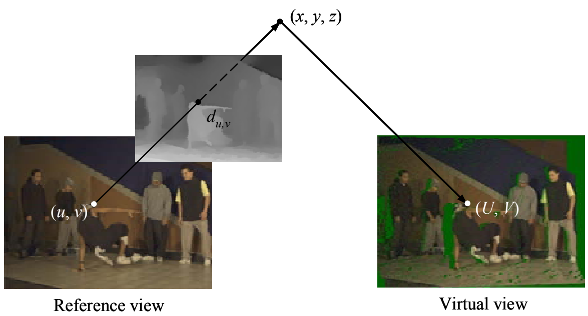


Fig. 2. General concept of 3D warping

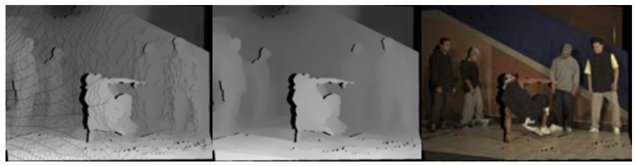
Eq. (1) and (2) represent the back-projection and the re-projection processes, respectively.

$$(x,y,z)^T = R_{ref} A_{ref}^{-1} (u,v,1)^T d_{u,v} + t_{ref} \tag{1}$$

$$(l,m,n)^T = A_{ver} R_{ver}^{-1} \{(x,y,z)^T - t_{ver}\} \tag{2}$$

where A, R, and t are camera parameters and d represents the depth value of a point in the 3D space. The coordinate (u, v) located on the reference view is 3D warped to (U, V) on the virtual view. The coordinate (l, m, n) in (2) is normalized to (l/n, m/n, 1) and then represented as an integer-coordinate (U, V) in the virtual view.

To avoid the errors like black-contours appeared in the 3D warped image as shown in Fig. 3 (a), the depth image is firstly warped and then median filtered to remove the block-contour errors. Finally, the texture image is synthesized by calling the matched texture pixels from the reference view as shown in Fig. 3 (c).



(a) (b) (c)

Fig. 3. 3D warping without erroneous blanks: (a) 3D warped depth image, (b) median filtered depth, (c) 3D warped texture image

The boundary trace-errors around the big holes are generally caused by inaccuracy of the camera parameters and inaccurate boundary matching between texture images and depth images. To remove these visible errors we extend the holes boundaries by using image dilation as shown in Fig. 4. These extended holes can be filled by the other 3D warped view and we can expect more natural synthesized view by removing this kind of errors.

(a) (b)

Fig. 4. Hole extension: (a) before extension (b) after extension

The next step is view blending to combine 3D warped views with the virtual view and the simplest way would be taking a weighted sum of two images as below:

$$I_V(u,v) = \alpha \; I_L(u,v) + (1-\alpha) I_R(u,v) \tag{3}$$

where I_L and I_R are the 3D warped reference texture images and I_V is an image to be blended. Generally, the weighting factor α is calculated based on the baseline distance as in (4).

$$\alpha = \frac{|t_V - t_L|}{|t_V - t_L| + |t_V - t_R|} \tag{4}$$

where t is a translation vector for each view. The final step is the hole-filling for the remaining holes.

3. PROPOSED HOLE-FILLING METHOD

Even though the view blending efficiently fills up most disocclusion regions, some holes still remain. In general, these kinds of remaining holes are caused by still remaining disocclusion regions and wrong depth values. Disocclusion regions are defined as areas that cannot be seen in the reference image, but exist in the synthesized one.

Most of the existing hole-filling methods use image interpolation or in-painting techniques and fill up the remaining holes using neighboring pixels solely based upon geometrical distance. However, observe that it makes more sense to fill up the holes using the background pixels rather than the foreground ones as the disoccluded area usually belongs to the background by definition. Therefore, we propose a depth based hole-filling algorithm which prefers the background pixels over the foreground ones in addition to considering the existing in-painting technique.

The general in-painting problem is as follow [9]: the region to be in-painted Ω and its boundary ∂Ω are defined and the pixel *p* belong to Ω would be in-painted by its neighboring region $B_\varepsilon(p)$ as shown in Fig. 5.

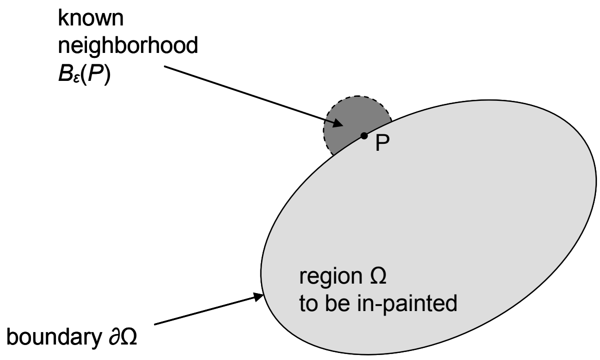


Fig. 5 General in-painting circumstance

This concept is quite reasonable for common image in-painting but it should be changed to be applied to hole-filling in view synthesis because ∂Ω of a certain hole can be located on the both foreground and background. In this case, we replace the boundary region bordering on foreground to the background region located on the opposite side as depicted in (5). That is, we intentionally manipulate the hole to have neighborhood only come from background as shown in Fig. 6. Now, since the holes only have the background pixels the in-painted image is more natural compared to the previous result.

$$\begin{aligned} p_{fg} \in \partial\Omega_{fg} &\rightarrow p_{bg} \in \partial\Omega_{bg} \\ B_\varepsilon(p_{fg}) &\rightarrow B_\varepsilon(p_{bg}) \end{aligned} \quad (5)$$

where *fg* and *bg* mean the foreground and background.

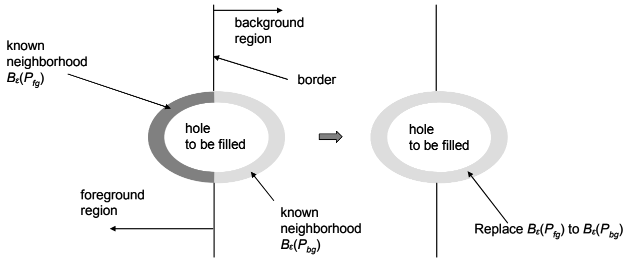


Fig. 6 Manipulation of hole to have neighborhood only come from background

To distinguish the foreground with background we use a corresponding depth data for blended image. For the horizontally opposite two depth pixels on hole, we regard the pixel having larger depth value as a foreground. Fig. 7 shows the procedure of the proposed hole-filling technique using depth based in-painting.

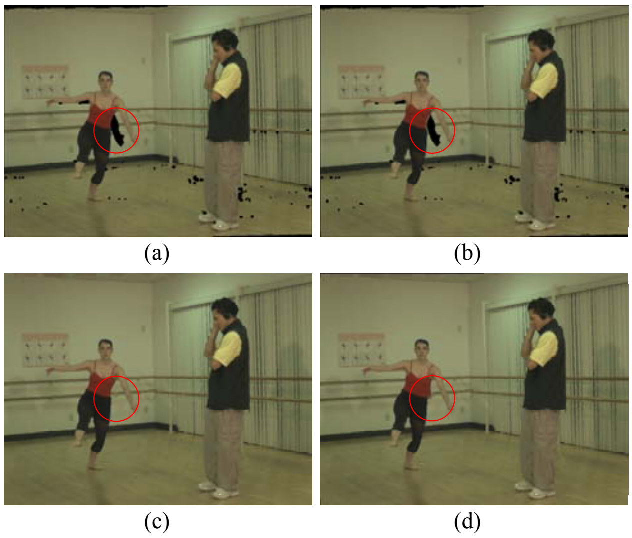


Fig. 7 In-painting procedure: (a) image with holes, (b) boundary region copy from background, (c) proposed depth based in-painting, (d) previous in-painting

4. EXPERIMENTAL RESULTS AND ANALYSIS

We have tested the proposed algorithm on test sequences, “Breakdancers” and “Ballet” [10]. Among the 8 views, view 3 and view 5 are selected as reference views and view 4 is set as the virtual view. The proposed method is evaluated by existing objective evaluation measures, such as PSNR, SSIM [11], and VQM [12]. Whereas the larger value means the better quality in PSNR, the smaller value means the better quality for VQM. In case of SSIM, the closer value to 1 means the better quality.

The experimental results for depth based in-painting are given in Table 1 and Table 2 and synthesized sample images in Fig. 8. The proposed depth-based in-painting shows the better results by filling up the remaining holes using pixels located in background when the holes border on both foreground and background.

Table 1. Experimental results for "Breakdancers"

Evaluation Measures	Previous	Proposed
PSNR	31.7300	31.7484
SSIM	0.8381	0.8384
VQM	3.9973	3.9852

Table 2. Experimental results for "Ballet"

Evaluation Measures	Previous	Proposed
PSNR	31.7773	32.4967
SSIM	0.8736	0.8740
VQM	2.6134	2.5131

Fig. 8 Results of hole-filling: (top) previous in-painting, (bottom) proposed depth based in-painting

As shown in experimental results and examples for synthesized images, the proposed depth-based in-painting shows subjectively and objectively better results for hole-filling compared with the previous in-painting.

5. CONCLUSIONS

In this paper, we have proposed a hole-filling method using depth-based in-painting technique. The proposed method intentionally manipulates the holes to have neighborhood only come from background by considering depth value. And then, the existing in-painting technique is applied to fill up the holes. The effectiveness of the proposed method was confirmed by evaluating the quality of the synthesized image using various quality measures such as PSNR, SSIM, and VQM. We observed that the proposed method produced both subjectively and objectively better results compared with those by the current reference software being used in the MPEG FTV/3D video standardization activities.

ACKNOWLEDGMENT

This work was supported in part by ITRC through RBRC at GIST (IITA-2008-C1090-0801-0017).

REFERENCES

[1] M. Tanimoto, "Overview of free viewpoint television," *Proc. of Signal Processing: Image Communication*, vol. 21, pp. 454–461, July 2006.
[2] A. Smolic and P. Kauff, "Interactive 3D video representation and coding technologies," *Proc. of IEEE, Special Issue on Advances in Video Coding and Delivery*. vol. 93, pp. 99–110, 2005.
[3] F. Isgro, E. Trucco, P. Kauff, and O. Schreer, "3D image processing in the future of immersive media," *Proc. of IEEE Transactions on Circuits and Systems for Video Technology*, vol.14, no. 3, pp. 288–303, 2004.
[4] X. Chen and A. Luthra, "MPEG-2 multiview profile and its application in 3D TV," *Proc. of SPIE-Multimedia Hardware Architectures*, vol. 3021, pp. 212–223, 1997.
[5] H.A. Karim, S. Worrall, A. H. Sadka, and A. M. Kondoz, "3D video compression using MPEG-4-multiple auxiliary component (MPEG4-MAC)," *Proc. of IEE 2nd International Conference on Visual Information Engineering (VIE2005)*, Apr. 2005.
[6] ISO/IEC JTC1/SC29/WG11, "Survey of algorithms used for multi-view video coding (MVC)," Doc. N6909, Jan. 2005.
[7] S. C. Chan, H.Y. Shum, and K.T. Ng, "Image-based rendering and synthesis," *Proc. of IEEE Signal Processing Magazines*, pp. 22–33, Nov. 2007.
[8] ISO/IEC JTC1/SC29/WG11, "Reference Softwares for Depth Estimation and View Synthesis," Doc. M15377, Apr. 2008.
[9] A. Telea, "An image inpainting technique based on the fast marching method," *Proc. of Journal of Graphics Tools*, vol.9, no.1, pp.25–36, 2004.
[10] MSR 3D video Sequences. Available at: http://www.research.microsoft.com/vision/ImageBasedRealities/3DVideoDownload/.
[11] Z. Wang, A. C. Bovik, H. R. Sheik, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *Proc. of IEEE Transaction on Image Processing* vol. 13, no. 4, pp. 600–612, 2004.
[12] F. Xiao, "DCT-based video quality evaluation," *Final Project for EE392J Stanford Univ*. 2000.