Signal Processing: *Image Communication* 24 (2009) 65–72

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

View generation with 3D warping using depth information for FTV

Yuji Mori *, Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, Masayuki Tanimoto

Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan

ARTICLE INFO

Article history:
Received 6 October 2008
Accepted 19 October 2008

Keywords:
Depth-image-based rendering
Free-viewpoint TV
3D warping
Post-filtering

ABSTRACT

In this paper, we propose a new method of depth-image-based rendering (DIBR) for free-viewpoint TV (FTV). In the conventional method, we estimated the depth of an object on the virtual image plane, which is called view-dependent depth estimation, and the virtual view images are rendered using the view-dependent depth map. In this method, virtual viewpoint images are rendered with 3D warping instead of estimating the view-dependent depth, since depth estimation is usually costly and it is desirable to eliminate it from the rendering process. However, 3D warping causes some problems that do not occur in the method with view-dependent depth estimation; for example, the appearance of holes on the rendered image, and the occurrence of depth discontinuity on the surface of the object at virtual image plane. Depth discontinuity causes artifacts on the rendered image. In this paper, these problems are solved by projecting depth map to the virtual image plane and performing post-filtering on the projected depth map. In the experiments, high-quality arbitrary viewpoint images were obtained by rendering images from relatively small number of cameras.

1. Introduction

In an era where communication is becoming a major component of human life, visual communication comes as the top among all other means of communication. Nowadays, with the great progress that computers have made, digital imaging has become a tremendously powerful and successful tool that eases production, transmission and display of video content. In this context, multi-view imaging (MVI) has attracted increasing attention. The availability of multiple views of the scene significantly broadens the field of applications, the field of applications. Free-viewpoint TV (FTV) [8] is one of the most important applications of MVI and is a new type of media that expand the use experience beyond what is offered by traditional media. In the FTV system, user can freely control the viewpoint position of any dynamic real-world scene.

The FTV system is based on the Ray-Space method [1], an image-based-rendering (IBR) approach that can render photorealistic images like actually captured images. A light ray in three-dimensional (3D) space can be defined using the Plenoptic Function with seven parameters: $f(x, y, z, \theta, \phi, \lambda{:}t)$. In the function, (x, y, z) is the point light ray is observed (recorded), (θ, ϕ) is the direction, λ is the wavelength and t is the time of a light. Three IBR techniques, namely Ray-Space, Light Field [5] and the Lumigraph [3], were proposed around the same time. These methods define rays in 3D space with four parameters assuming that the ray travels in 3D space straight without attenuation. For example, the Ray-Space method represents rays as a function $f(x, y, \theta, \phi)$. Multi-camera images are reset in new 4D space as rays. Virtual views are synthesized by loading ray information in 4D parameter space. Practically, since captured rays are limited, ray interpolation is needed when the requested ray is unavailable. The interpolation of novel views can be translated into the problem of estimating the ray depths, reflected by the object. This correspondence problem has been widely studied in stereo vision and multiple

* Corresponding author. Tel.: +81 52 789 3163; fax: +81 52 789 3628.
E-mail address: mori@tanimoto.nuee.nagoya-u.ac.jp (Y. Mori).

0923-5965/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2008.10.013

methods have already been tested. Fukusima et al. [2] estimated depth map of the view to generate, called as view-dependent depth map, in almost real-time. This method brought most of its mighty to real-time optimization. Meanwhile, high-quality depth estimation methods, for example, color segmentation based one was proposed [4]. These methods outperform real-time estimation with quality especially at edge and occluded area, but they require tremendous amount of time. In depth estimation process, estimation accuracy and computation cost is usually traded-off.

The realization of a high-quality FTV system relies heavily on the image quality and processing time, so these two traded-off factors must be satisfied in high level. Thus, this paper assumes that the depth maps are estimated as an off-line process, which enables us to use high-accuracy depth maps, without computation at rendering stage. In this paper, multiple viewpoint images and estimated depth maps at each real camera positions are treated as a dataset for image generation, and novel viewpoint images are rendered with 3D warping [6], using depth map of a image to warp. In this method, arbitrary viewpoint images can be rendered, only projecting pixels to the virtual image plane. However, some problems, which do not appear in view-centered method, occur in 3D warping. For example, holes may appear on the rendered image, and depth discontinuities may occur on the surface of the object at the virtual image plane. Depth discontinuity causes artifacts on the rendered images. Moreover, 3D warping is not compatible with the Ray-Space method since the Ray-Space method requires depth map at the virtual image plane. The difference of image quality between the rendered images with 3D warping and the Ray-Space method outstands especially at the occluded regions when the virtual camera moves away from the reference plane. For this reason, a Ray-Space-compatible method is strongly desired to achieve a FTV system with high interactivity. We dealt with these problems by introducing post-filtering on the projected depth map.

We are proposing this framework, which renders novel images from captured images and computed depth maps, to MPEG. In the MPEG meeting, depth-image-based rendering (DIBR) is drawing attention and well discussed as the candidate for the next generation FTV format.

This paper is organized as follows: Section 2 explains pinhole camera model that is used for image generation in this work. Section 3 presents an overview of our algorithm. Experimental results and discussion are described in Section 4. Section 5 concludes the paper.

2. Pinhole camera model

Let $\tilde{M} = [X,\ Y,\ Z, 1]^T$ be a world point and $\tilde{m} = [u, v, 1]^T$ be its projection in a camera with projection matrix P in homogeneous coordinates. A camera is modeled as the usual pinhole (Fig. 1); the relationship between $\tilde{M}$ and $\tilde{m}$ is given in Eq. (1), where $\boldsymbol{s}$ means non-zero scalar and 3×4 matrix $\boldsymbol{P}$ is called the camera matrix. Camera matrix P can be decomposed as in Eq. (2).

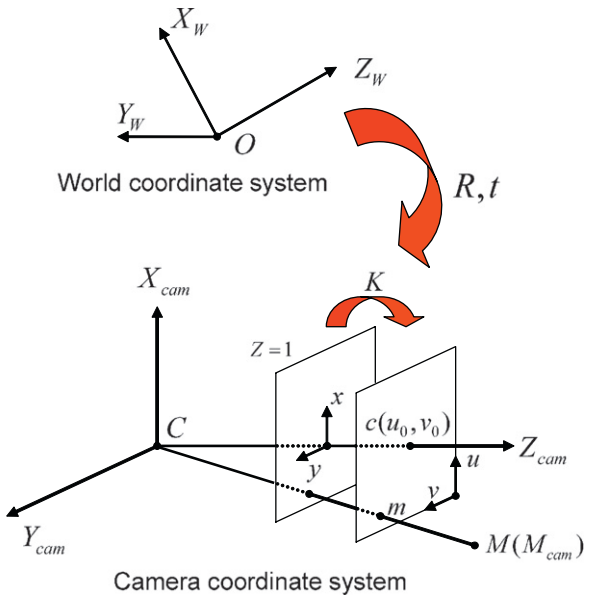

Fig. 1. Pinhole camera model.

In Eq. (2), 3×3 orthogonal matrix $\boldsymbol{R}$ represents the orientation and 3-vector $\boldsymbol{t}$ represents the position. $\boldsymbol{K}$ is 3×3 upper triangular matrix given by Eq. (3), with focal length $\boldsymbol{f}$, skew parameter γ, and principal point (u_0, v_0). The matrix $\boldsymbol{K}$ is called the intrinsic matrix and represents the inner structure of the camera. The matrix $[R;-Rt]$ is called the extrinsic matrix and it indicates the relationship between the world coordinates and the camera coordinates. Note that this model ignores non-linear distortion.

$$P\tilde{M} = s\tilde{m} \tag{1}$$

$$P = K[R;-Rt] \tag{2}$$

$$K = \begin{bmatrix} f_u & \gamma & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

3. The proposed algorithm

The proposed algorithm is shown in Fig. 2 and it consists of four steps. These steps are explained in detail in Sections 3.1–3.3, respectively.

3.1. Depth map projection with 3D warping

3D warping projects an image to another image plane. As described in Eq. (1), the 3D point $\tilde{M}$ can be reconstructed from the image point $\tilde{m}$ using projection matrix $\boldsymbol{P}$ and depth Z. Then, the reconstructed 3D point is projected to the virtual image plane with the projection matrix of the virtual camera. In this step, depth maps of two nearest cameras are projected to virtual image plane. Several points can be projected to the same pixel on the virtual view image; in this case, the nearest point is

adopted, as described in Eq. (4)

$$Z(u,v) = \underset{\tilde{M}_{u,v}}{arg\min}\ Z \tag{4}$$

In Eq. (4), $Z(u, v)$ is the depth value at (u, v) on the virtual image plane, $\tilde{M}_{u,v}$ is the 3D point whose projection is (u, v). In this method, two nearest cameras are used to render a virtual view image and the virtual view image is generated between them. In this paper, the camera that is placed at the left of the virtual camera is called as the left camera and the other one is called as the right camera. The projected depth maps from each camera for one arbitrary scene are shown in Fig. 3(a) and (b). Here, we assumed that the depth should change smoothly inside the same object. However, depth discontinuity was observed in the projected depth maps and in addition, there are many blank points appeared. To solve this, we need to smooth the projected depth map, which is achieved by performing a post-filtering process.

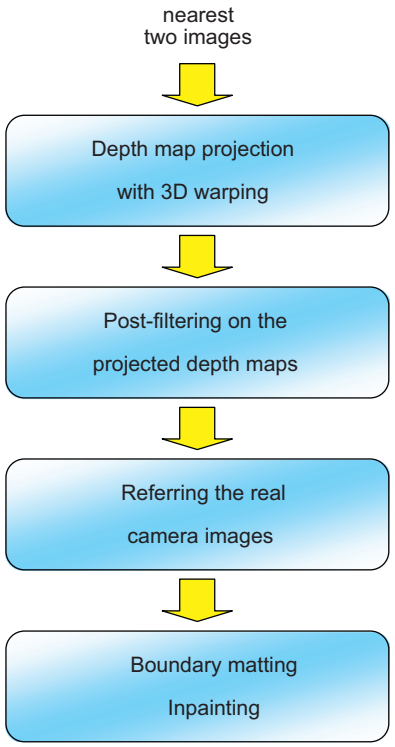

Fig. 2. Block diagram of the proposed algorithm.

3.2. Post-filtering on depth map

In some cases, smoothing the depth map is necessary to gain a natural-looking rendered image, because the noise on the resulted image can be reduced by smoothing. Here, if the depth map is smoothed, the resultant image will not be blurry. But, if the resultant image is smoothed it gets blurred. First, we consider blank points that appeared on the projected depth map. The reasons for the appearance of these blank points are round off errors of the image coordinate calculation and depth discontinuity. In Eq. (1), image coordinates (u, v) are calculated decimally, but rounded off to the nearest integer value. It can cause one pixel wide blank region to appear. This blank region can be filled by median filter. Depth discontinuity causes lump blank in the depth map. These blank regions cannot be filled by the median filter. However, most of the discontinuities in depth map is caused by occlusion; so, these areas can be filled using the image from another camera.

3D warping causes not only blank areas appear in an object, but irregular depth changes may also occur in the same object. These irregularities cause unnatural-looking pixel in the rendered image, so it is also desirable to smooth them away. They can be smoothed away by using a low-pass filter. However, edge regions should be preserved. Sometimes low-pass filtered edge in the depth map blurs the edges of objects in the rendered image. Considering that the low-pass filter cannot smoothen the desired edge areas perfectly, we adopted bilateral filtering [10] for this purpose, which is defined in Eq. (5)

$$h(x) = k^{-1} \iint_D f(\xi)c(\xi - x)s(f(\xi) - f(x))\,\mathrm{d}\xi \tag{5}$$

In Eq. (5), k and D mean the normalization constant and the filtering domain, respectively. This is shift-invariant Gaussian filtering. Here, both the closeness function c and similarity function s are Gaussian functions. More specifically, $\boldsymbol{c}$ is

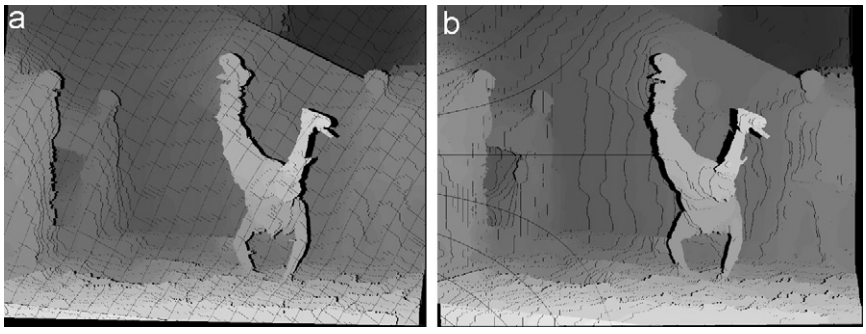

Fig. 3. Projected depth maps from the two nearest cameras. (a) Projection from the left side and (b) projection from the right side.

radically symmetric and it is shown in Eq. (6).

$$c(\xi - x) = \exp\left(-\frac{1}{2}\left(\frac{|\xi - x|}{\sigma_d}\right)^2\right) \tag{6}$$

Here, σ_d is the variance of Euclidean distance. The similarity function s is perfectly analogous to $\boldsymbol{c}$ and it is shown in Eq. (7)

$$s(\xi - x) = \exp\left(-\frac{1}{2}\left(\frac{|f(\xi) - f(x)|}{\sigma_r}\right)^2\right) \tag{7}$$

In Eq. (7), σ_r means the variance of color space. By introducing this factor, far point in color space would be less weighted, which preserves edge region from smoothing. Fig. 4(a) shows an original status and Fig. 4(b) is after adding Gaussian noise on it. Fig. 4(c) and (d) show the filtered images of Fig. 4(b) by using bilateral filtering and Gaussian filtering, respectively. Fig. 4(a) is well restored with bilateral filtering, especially at the edge regions. The images in Fig. 3(a) and (b) can be smoothed by using median filtering and bilateral filtering to obtain images in Fig. 5(a) and (b), respectively. Section 3.3 describes the

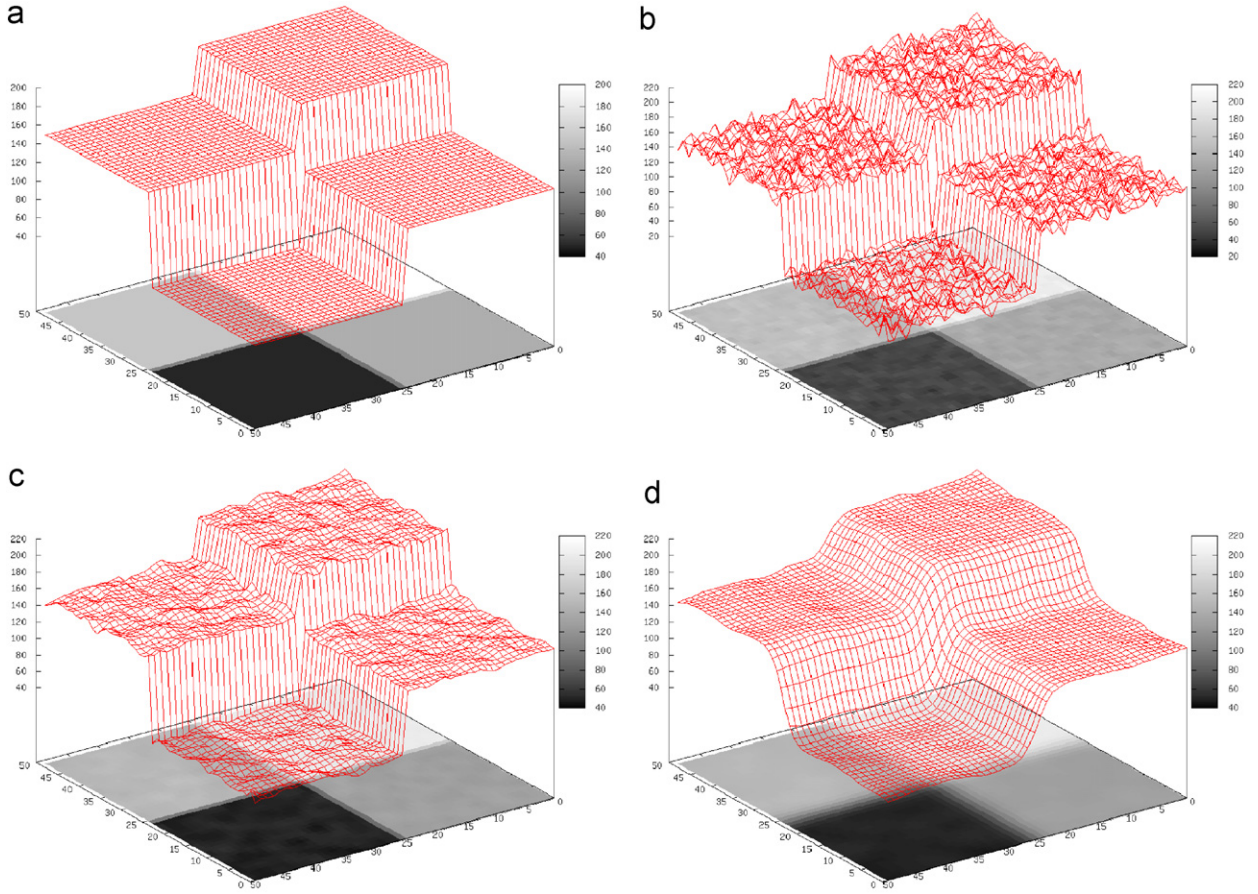

Fig. 4. Comparison of bilateral filtering and Gaussian filtering. (a) Original status, (b) additive Gaussian noise, (c) bilateral filtering and (d) Gaussian filtering.

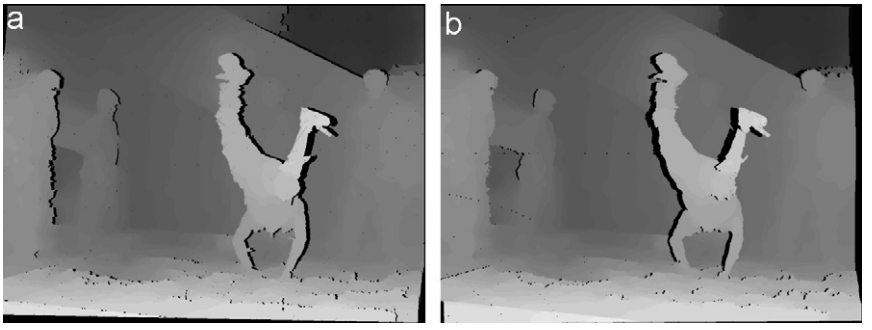

Fig. 5. Post-filtered depth maps. (a) Projection from the left side and (b) projection from the right side.

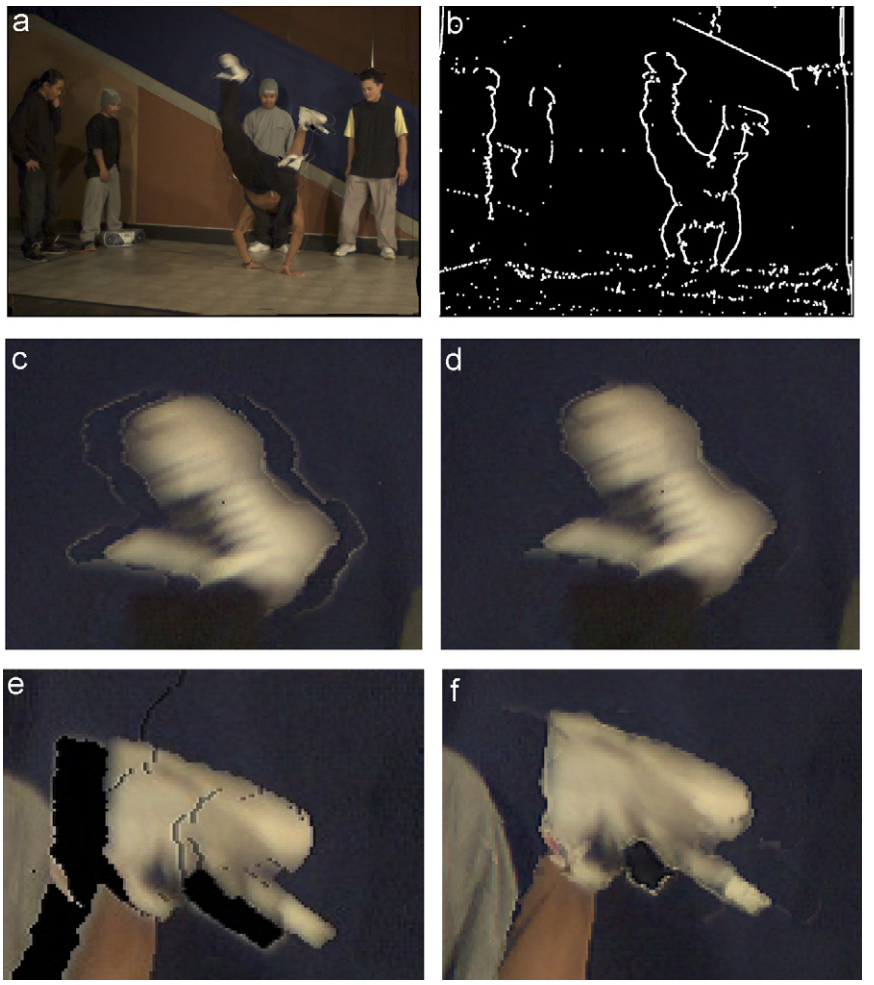

Fig. 6. Sample results of rendering. (a) Blended image, (b) occluded region, (c) before matting, (d) after matting, (e) 3D warping and (f) proposed method.

rendering of the virtual view image with the smoothed depth maps.

3.3. Boundary matting and inpainting

After performing post-filtering on depth maps, these two depth maps are projected to each real camera image. Then, the virtual view image shown in Fig. 6(a) is rendered by blending two neighboring images, as described in Eq. (9)

$$occ_L(u,v)=\begin{cases}1 & (Z_R(u,v)<\text{threshold})\\ 0 & (Z_R(u,v)>\text{threshold})\end{cases}$$
$$occ_R(u,v)=\begin{cases}1 & (Z_L(u,v)<\text{threshold})\\ 0 & (Z_L(u,v)>\text{threshold})\end{cases} \quad (8)$$

$$I(u,v)=\begin{cases}(1-\alpha)I_L(u_L,v_L)+\alpha I_R(u_R,v_R) & (occ_L(u,v)=0, occ_R(u,v)=0)\\ I_L(u_L,v_L) & (occ_L(u,v)=0, occ_R(u,v)=1)\\ I_R(u_R,v_R) & (occ_L(u,v)=1, occ_R(u,v)=0)\\ 0 & (occ_L(u,v)=1, occ_R(u,v)=1)\end{cases} \quad (9)$$

$$\alpha=\frac{|t-t_L|}{|t-t_L|+|t-t_R|} \quad (10)$$

In Eq. (9), $I(u, v)$ means the pixel value at (u, v) virtual image plane, I_L, I_R mean the reference image plane, occ is the occlusion map defined in Eq. (8), (u_L, v_L) and (u_R, v_R) are the projected points to the reference camera from (u, v) on the virtual image plane, t is the translation vector of the extrinsic matrix. α is a weighting coefficient and it is defined in Eq. (10). Here, subscript L and R indicate left and right cameras, respectively. t_L, t_R and t are the translation vectors of the left camera, the right camera, and the virtual camera, respectively. In Eq. (8), Z_L and Z_R are the depth values at the virtual image plane projected from the left and right cameras. As defined in Eq. (8), the occlusion map is obtained by mapping the area where the projected depth value is less than a pre-defined threshold after post-filtering. No pixel is expected to be projected to these areas. Left-side occlusion is assumed to happen in the image projected from the right side image and vice versa. The area where $occ(u, v)=1$ is regarded as the occluded area. Fig. 6(a) shows blending of two images obtained using the two

smoothed depth maps, and Fig. 6(b) shows the area where either occ_L or occ_R is true.

Due to the miss-focus and half-pixel problems, the objects may be ill-defined at the borders. It makes depth estimation difficult and as a result, unnatural pixel appears around the boundary region. To reduce this, we conducted boundary matting defined in Eq. (11).

$$\begin{aligned} occ_L(u,v) &= 1 \quad (occ_L(u+1,v)=1, Z_R(u,v) < Z_R(u+\Delta u, v)) \\ occ_R(u,v) &= 1 \quad (occ_R(u-1,v)=1, Z_L(u,v) < Z_L(u-\Delta u, v)) \end{aligned} \tag{11}$$

In Eq. (11), Δu is the width of occlusion (u+Δu is the nearest pixel which has non-zero depth value). This process actually expands the border of the occlusion, meaning the area where the pixel value is copied from one reference image. It erases the mixture of foreground and background colors. Fig. 6(c) and (d) show the same part of resulted image obtained before matting and after matting. The remaining blank area was filled with inpainting [9], which is usually used, for example, to recover damaged part of the image or erase subtitle. Fig. 6(e) and (f) show close-up of images generated with 3D warping and the proposed method.

4. Experimental results and discussion

4.1. Experimental conditions

For experiments, we used sequences named "breakdancers" and "ballet", generated and distributed by Interactive Visual Group at Microsoft Research [7]. This data includes a sequence of 100 images captured from 8 cameras. Depth maps computed from stereo are also included for each camera with the calibration parameters. The captured images have a resolution of 1024×768 pixels. More detailed descriptions, for example, depth map generation method, are explained in Ref. [11].

In this paper, image quality is evaluated with peak signal-to-noise ratio (PSNR). PSNR is the evaluation function that is based on the squared-error between the two images. In our method, the virtual viewpoint image is generated from the two nearest images. In this case, the virtual viewpoint is set to be exactly the same as the actual reference camera, and the virtual viewpoint image is generated from the two nearest cameras other than the reference camera. PSNR is calculated between the generated image and the reference image. Prior to the calculation of PSNR, the image in RGB color space is converted to the image in YUV color space and Y channel is used for calculation. The conversion to Y channel is defined in Eq. (12). After conversion, PSNR is calculated as described in Eq. (13).

$$Y(u,v) = 0.299R(u,v) + 0.587G(u,v) + 0.114B(u,v) \tag{12}$$

$$10\log_{10}\frac{255^2}{(1/W \times H)\sum_{s,t=0}^{W-1,H-1}(Y(s,t)-\hat{Y}(s,t))^2} \tag{13}$$

In Eq. (13), W and H are the image width and image height, respectively. Y and $\hat{Y}$ are the Y channel of the reference image and the generated image, respectively. For image generation, we set each parameter as follows: $\sigma_d = 20$, $\sigma_r = 5$ and *threshold* $= 5$. The *threshold* here is the brightness in the depth map. The relationship between the pixel value of the depth map and the actual depth value is described in Eq. (14).

$$Z = \frac{1}{P/225((1/MinZ)-(1/MaxZ))+(1/MaxZ)} \tag{14}$$

In Eq. (14), P is the pixel value in the depth map, *MinZ* and *MaxZ* defines the depth range. In the "breakdancers" sequence, *MinZ* is 44 and *MaxZ* is 120, and in the "ballet" sequence, *MinZ* is 42 and *MaxZ* is 130.

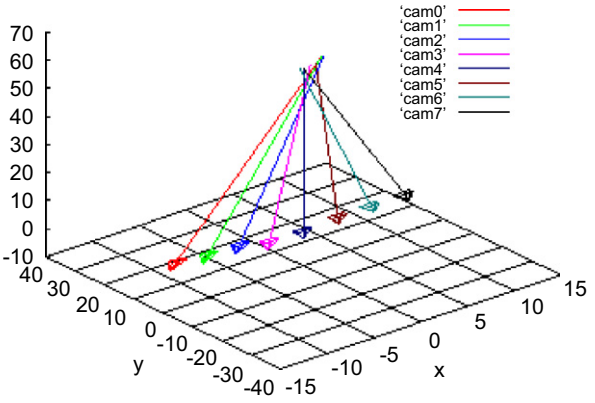

Fig. 8. The camera configuration.

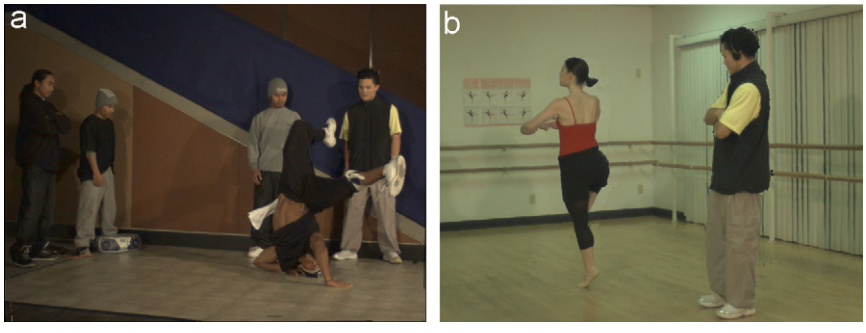

Fig. 7. Sample results of rendering. (a) Breakdancers and (b) ballet.

4.2. Experimental results

The experiments were conducted to confirm the effectiveness of the proposed method. Fig. 7 shows an example of the view synthesis results. Fig. 8 shows the camera arrangement of the "breakdancers" sequence. We generated the image of camera4 that is located at the center of camera array (Fig. 8) and the origin of world coordinates to calculate PSNR. Figs. 9 and 10 show the result of calculating PSNR with changing distance between cameras given by $|t-t_L|+|t-t_R|$. As a comparison, we calculated PSNR values for the images generated by view-centered method and 3D warping as well. Our method improved PSNR for maximum 4 dB compared to the other two methods.

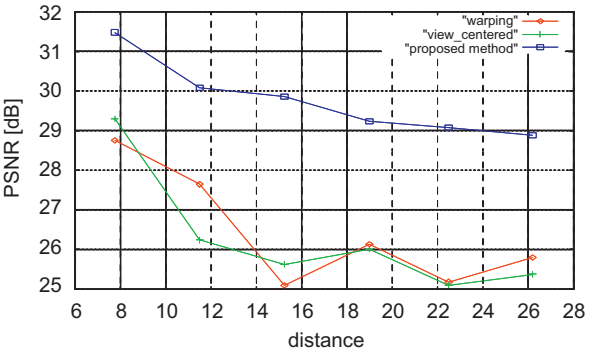

Fig. 9. PSNR (breakdancers).

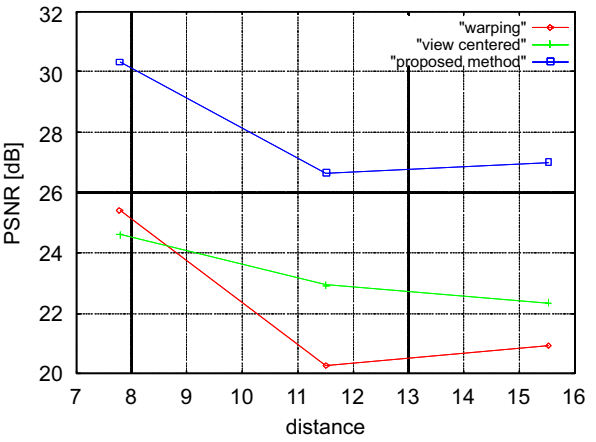

Fig. 10. PSNR (ballet).

4.3. Discussion

To inspect how much possibility for improvement there was in PSNR, we calculated PSNR using a mask generated from the absolute difference of the two projected depth maps (Fig. 11). It is assumed that if the absolute difference of the values for a pixel in the two depth maps is larger, it is expected to have a lower signal-noise ratio.

Fig. 12 shows the PSNR of the part where absolute difference exceeds the pre-defined threshold indicated by the horizontal axis, and Fig. 13 shows the PSNR of the part where the absolute difference do not exceed the threshold. In Fig. 12, PSNR is converged to the one calculated by using the entire image. As expected, PSNR decreases as the absolute difference grows. Fig. 11 shows that for maximum 3 dB difference as the threshold is varied. However, the area where the two projected depth values differ occupies comparably a small percentage of the

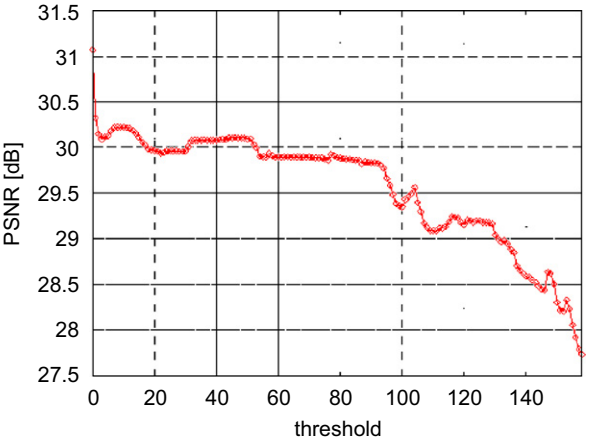

Fig. 12. PSNR of the part where the threshold is exceeded.

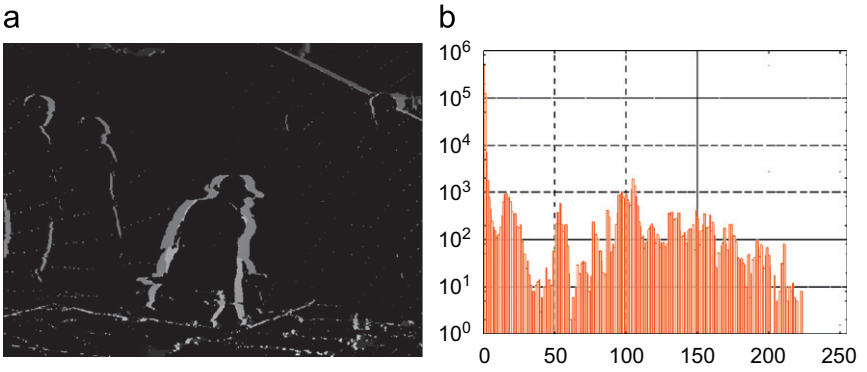

Fig. 11. The absolute difference of two projected depth maps. (a) Visualized image and (b) histogram.

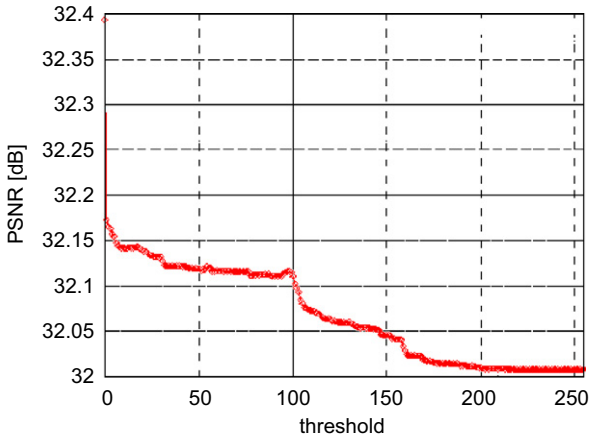

Fig. 13. PSNR of the part where the threshold is not exceeded.

whole image (see Fig. 11(b) for the log scale). The influence for the entire PSNR is small and it is about 0.4 dB (Fig. 13). From this investigation, we conclude that it is difficult to improve the image quality in PSNR, using a certain depth map to render a novel view image.

5. Conclusion

In this paper, we propose a novel 3D warping-based rendering method for FTV. This method solves some of the problems of depth-image-based rendering(DIBR) by performing post-filtering at the virtual image plane with an assumption that the depth value should vary smoothly inside a region of similar color. Furthermore, Ray-Space compatibility can be achieved by the fact that the depth map is reconstructed at the virtual image plane. Experimental results show the efficiency of our algorithm. The implementation of the proposed algorithm is adopted as the reference software for MPEG standardization experiments for Free-viewpoint TV (MPEGFTV).

Future work will include better handling of occlusion around the depth boundaries, more sophisticated depth smoothing, and the implementation of a Ray-Space-Based high-interactive FTV system.

Acknowledgments

We would like to thank Interactive Visual Media Group, Microsoft Research for distributing multi-camera video with fine quality depth data and camera parameters.

References

[1] T. Fujii, T. Kimoto, M. Tanimoto, Ray-space coding for 3D visual communication, in: Proc. Pict. Coding Symp., Vol. II, 1996, pp. 447–451.
[2] N. Fukushima, T. Yendo, T. Fujii, M. Tanimoto, Free viewpoint image generation using multi-pass dynamic programming, in: Proc. SPIE Stereoscopic Displays Virtual Reality Syst., Vol. XIV, 2007, pp. 460–470.
[3] S. Gortler, R. Grzeszczuk, R. Szeliski, M. Cohen, The lumigraph, in: Proc. ACM SIGGRAPH'96, 1996, pp. 43–54.
[4] A. Klaus, M. Sormann, K. Kamer, Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure, in: Proc. ICPR'06, Vol. III.
[5] M. Levoy, P. Hanrahan, Light field rendering, in: Proc. ACM SIGGRAPH'96, 1996, pp. 31–42.
[6] W. Mark, L. Mcmillan, G. Bishop, Post-rendering 3D warping, in: Proc. Symp. I3D Graphics, 1997, pp. 7–16.
[7] Microsoft Research, Image-Based Realities-3D Video Download, 〈http://research.microsoft.com/ivm/3DVideoDownload/〉.
[8] M. Tanimoto, Overview of free viewpoint television, Signal Process.: Image Commun. 21 (2006) 454–461.
[9] A. Telea, An image inpainting technique based on the fast marching method, in: J. Graphics Tools, Vol. IX.
[10] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: Proc. ICCV'98, 1998, p. 839.
[11] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality video view interpolation using a layered representation, in: ACM Trans.Graphics, in: Proc. SIGGRAPH, vol. 23(3), 2004, pp. 600–608.