

Image-based Panoramic 3D Virtual Environment Using Rotating Two Multi-view Cameras*

Sehwan Kim¹, Eun-Young Chang², Chung-Hyun Ahn² and Woontack Woo¹

¹ KJIST U-VR Lab.
1 Oryong-dong, Puk-gu, Kwangju, Korea
E-mail: {skim, wwoo}@kjist.ac.kr

² ETRI Realistic Broadcasting Research Team
161 Gajeong-dong, Yuseong-gu, Daejeon, Korea
E-mail: {eychang, hyun}@etri.re.kr

ABSTRACT

In this paper, we propose a new method for generating an image-based 3D panoramic virtual environment (VE). The panoramic VE is generated using 3D depth information estimated from rotating two multi-view cameras. Even though conventional 2D image-based mosaicking methods provide a wide view, they have limitations in providing a user with a navigation-enabled virtual environment. In order to resolve such obstacles, we first estimate the depth of the scene using two calibrated multi-view cameras and then stitch 3D point clouds instead of images. By rotating two cameras using a turn-table it enables users to navigate the resulting 3D virtual environment with HMD.

1. INTRODUCTION

Panoramic image provides a wide field of view or allows a user to look around the whole scene. It provides immersion when it combines with special personal display systems such as HMD (Head Mounted Display). The virtual environment (VE) constructed with an image-based panorama provides more realism with relatively simple rendering than a CG-based VE. As a result, the image-based panorama has been adopted in constructing various types of image-based virtual reality systems (IBVR) [1].

Up to now, several approaches have been proposed to generate panoramic images. For example, a panorama can directly be generated using a Catadioptric omnidirectional camera [2]. However, it needs complicated calibration and compensation due to the inherent characteristics of the camera. A panorama can be constructed using an uncalibrated camera [3] or a moving camera [4]. However, both have limitations in providing 3D feeling or allowing navigation, since such systems generate a panorama based on 2D images. To add 3D feeling, Peleg et al. construct a 3D virtual panorama while changing baseline to get different disparity using an off-center rotating camera

[5]. Shum and Szeliski construct a similar system by using range information from several viewpoints [6]. Benosman and Devars get a depth map after rotating two linear image sensors with respect to an axis to generate two cylindrical projection images [7]. However, those 3D systems have difficulties in allowing users to navigate the VE.

In this paper, we propose a new method for generating an image-based 3D panoramic VE. First, we estimate depth of the scene using two calibrated multi-view cameras. Then, we stitch 3D point clouds instead of images by back-projecting the pixelwise depth into the 3D VE. Finally, we rotate two cameras using a turn-table. As a result, the proposed depth-based stitching enables a user with a HMD to navigate the resulting photo-realistic VE.

The rest of the paper is organized as follows: In chapter 2, we describe the proposed 3D VE generation method. Experimental results and analysis are explained in chapter 3. Conclusion and future work will be mentioned in chapter 4.

2. 3D VIRTUAL ENVIRONMENT GENERATION

Image-based 3D panoramic virtual environment overcomes weak points of the 2D image-based panorama by allowing users to navigate the 3D VE through an HMD. The proposed navigation-enabled 3D VE is generated by the following steps: (i) calibration and 3D depth information estimation, (ii) depth-based stitching, and (iii) panoramic 3D VE generation.

2.1. Camera calibration and depth-based stitching

Point clouds estimated from two multi-view cameras are stitched to generate a panoramic 3D VE based on the geometric relationship between two cameras. Thus, an accurate camera calibration process is required. In this paper, two multi-view cameras are calibrated based on modified Zhang's algorithm [9]. However, the algorithm has problems when it is applied to VR applications since the accuracy becomes worse as distance increases. Thus,

* It was supported in part by ETRI and in part by KJIST

we propose an effective method to get reliable results at a long distance using two multi-view cameras whose three lenses are already calibrated [10].

To lessen the projection errors, we capture several images of the static pattern. Then, grid points of the pattern, obtained through mean or median filters, are used for getting intrinsic parameters of the cameras. After obtaining intrinsic parameters, extrinsic parameters are calculated for three lenses. The lenses of the multi-view camera see the same direction and are positioned on a single plane. The extrinsic parameters of each camera are obtained by averaging the extrinsic parameters of three lenses. The resulting extrinsic parameters provide rotation and translation matrices from the origin of the world coordinate to the centers of each camera as shown in Figure 1 [11][12].

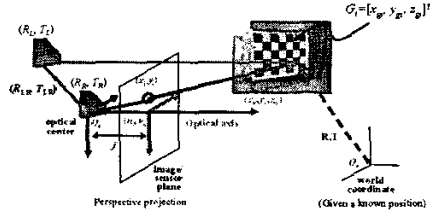


Figure 1. Camera calibration

Note however that, in general, the back-projected results from the left and right camera are not matched in VE because independently estimated extrinsic parameters, (R_L, T_L) and (R_R, T_R) , have some inherent errors. Thus, we need to estimate the extrinsic parameters of two cameras using dependent calibration, i.e. iterative calibration of two cameras.

For 3D depth-based stitching in VE, it is essential to acquire accurate R_{LR} and T_{LR} . The relationship between two cameras, R_{LR} and T_{LR} , can be expressed as follows:

$$\begin{aligned} R_{LR} &= R_R R_L^{-1} \\ T_{LR} &= T_R - R_{LR} T_L \end{aligned} \quad (1)$$

where, R_{LR} and T_{LR} are the rotation and translation matrices of right camera with respect to the left camera.

We obtain the optimum values by exploiting back-projected 3D coordinates of K grid points for each camera through an optimization process. That is, given K grid points, R_{LR} and T_{LR} are found such that the distance between corresponding grid points may be minimized, i.e.,

$$\begin{aligned} &\text{Given two sets of corresponding points,} \\ &\text{Find } \{R_L, T_L\} \text{ \& } \{R_R, T_R\} \\ &\text{such that } \arg \min_{\{R_L, T_L\} \text{ \& } \{R_R, T_R\}} \sum_{i=0}^{K-1} (G_{Li} - G_{Ri})^2 \end{aligned} \quad (2)$$

where, G_{Li} and G_{Ri} represent back-projected 3D coordinate of each grid point from left and right camera in VE, respectively. In real case, it is almost impossible to obtain

exactly same values even though G_{Li} and G_{Ri} should be the same. The difference is expressed as a function of $\{R_L, T_L\}$ and $\{R_R, T_R\}$. Thus, the accurate geometric relationship between the two cameras (or extrinsic parameters) can be found by minimizing the difference.

Let us denote (u_{G_u}, v_{G_u}) as an image coordinate of a 3D grid point $G_i = [x_{gi}, y_{gi}, z_{gi}]^T$ of the pattern for the left camera. The back-projected point G_{Li} into VE is represented as follows:

$$G_{Li} = \begin{bmatrix} x_{gLi} \\ y_{gLi} \\ z_{gLi} \\ 1 \end{bmatrix} = \begin{bmatrix} C_L & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_L & T_L \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u_{G_u} \\ v_{G_u} \\ 1 \end{bmatrix} = P_L^{-1} \begin{bmatrix} u_{G_u} \\ v_{G_u} \\ 1 \end{bmatrix} \quad (3)$$

where, C_L denotes the camera parameter. The back-projected point G_{Ri} through the right camera can be expressed similarly. Therefore, by estimating R_{LR} and T_{LR} minimizing Eq. (2), the relationship between two cameras can be obtained. Finally, the relationship enables two sets of point clouds to be stitched in the VE. Note however that there still remain some errors due to inaccurate 3D depth estimation.

The following method is used to remove noise over overlapping areas after back-projecting the point clouds taken from both cameras into VE using R_{LR} and T_{LR} . Let us assume that a cube, whose side is l , at the current moment. The median filter is applied to eliminate noise.

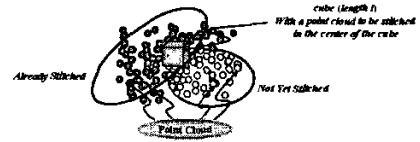


Figure 2. Noise removal

$$I_{LR_i} = \text{Median}_{j \in \eta_i} (I_{L_j}, I_{R_j}) \quad (4)$$

where, I_{LR_i} denotes stitching result for a point i . The subscript η_i represents the area specified by an i^{th} cube. I_{L_j} and I_{R_j} denote point clouds from the left and right camera within the cube. The median filter is applied upon the point clouds for the overlapping area to remove noise, while maintaining a smooth depth change. The color of the resulting point after stitching is used as that of i^{th} point. As a final step, bilinear interpolation is performed to display the point cloud in 3D VE to the user.

2.2. Panoramic 3D virtual environment generation

By rotating two multi-view cameras using a turn-table, VE can be generated and users can navigate the resulting 3D VE with a HMD. A user can see in any direction within a constant radius of VE as shown in Figure 3. The

procedure for generating VE is as follows. First, we closely locate two cameras to estimate more accurate extrinsic parameters of both cameras. Then, the baseline between two cameras is adjusted according to applications. Note that only translation vectors are affected and the relative rotation matrix does not change because cameras move only in a horizontal direction along the bar on the turn-table.

After translation and rotation matrices are estimated from the reference position, two virtual cameras can be positioned in a VE. Finally, 3D depth information from each camera is obtained while rotating the turn-table every constant degree as shown in Figure 3. 3D information is illustrated as point cloud and rendered onto the VE. We can obtain the point cloud in all directions and final 3D VE through 3D depth-based stitching.

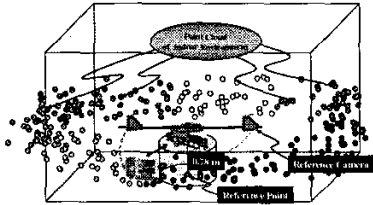


Figure 3. Virtual environment generation

We can measure how much observation gain we can get by comparing the case with that of one camera-based panorama. Figure 4 shows a top-view of a general indoor environment and a turn-table with two multi-view cameras. The upper circle represents an area where users can see the scene in 3D. Thus, by comparing maximum circle circumference seen from a centered-camera with that seen from the proposed method, we can see the observation gain of the proposed method.

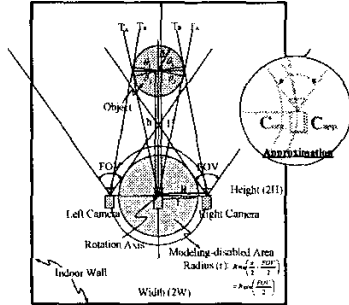


Figure 4. Gain of user navigation area

When one camera is used and a user sees the modeled cylinder from the rotation axis, the maximum angle between two tangential lines is $\pi - 2\theta_2$, where θ_2 denotes maximum angle formed by a normal. When the radius of the modeled human is d , the circle circumference A is as follows.

$$A = d \left(\pi - 2 \sin^{-1} \left(\frac{d}{h} \right) \right) \quad (5)$$

where, h is the distance from rotation axis to the center of an object and r is a radius of modeling-disabled area.

$$r = R \cos \left(\frac{FOV}{2} \right) \quad (6)$$

where, R is rotation radius, FOV is field of view of a multi-view camera.

On the other hand, in case of the proposed method, the observable maximum angle is $\pi + 2\theta_1$, where θ_1 is a maximum angle by a normal when tangential lines are formed from each camera to the cylinder. However, as shown in the right side of Figure 4, the difference between the position of original camera C_{org} and that of a camera by tangential line with a circle radius R , is negligible because the rotational axis is far enough from the environment. The maximum length of circumference B is expressed by approximation as follows:

$$B = d \left(\pi + 2 \sin^{-1} \left(\frac{R-d}{h} \right) \right) \quad (7)$$

Thus, the maximum gain C of an observable range by the proposed method is given as Eq. (8) and is represented as a function of R .

$$C = \left\{ \pi + 2 \sin^{-1} \left(\frac{R-d}{h} \right) \right\} / \left\{ \pi - 2 \sin^{-1} \left(\frac{d}{h} \right) \right\} \quad (8)$$

Accordingly, given h and d , C increases almost linearly with R increasing.

3. EXPERIMENTAL RESULTS AND ANALYSIS

We have performed experiments in a general indoor environment under a normal fluorescence lighting condition. We employed two IEEE 1394 cameras (Digiclops) to capture background. The camera calibration was accomplished using OpenCV library and a Xeon 1.0GHz CPU computer was used. In the experiments, the size of indoor environment is $7m \times 5m$ and pattern size is $1.75m \times 1.25m$.

Table 1. Error range of extrinsic parameters

Image Size	640×480	640×480
Distance	≈ 1m	≈ 4m
Pattern Size	0.3m×0.2m	1.75m×1.25m
Error Range	within ±0.01m	within ±0.04m

Figure 5 shows depth-based stitching results after the proposed optimization procedure. Figure 5(a) and Figure 5(b) are the back-projected results of point clouds from both left and right cameras, respectively. Figure 5(c)

shows a composite image after 3D depth-based stitching. As shown in Table 1, the error becomes bigger with the distance increasing. Thus, we reduced the errors by applying the optimization process to get accurate R_{LR} and T_{LR} as shown in Figure 5.

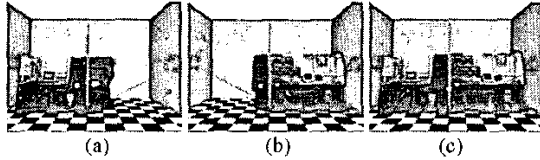


Figure 5. Depth-based 3D stitching (a) left image (b) right image (c) stitched image

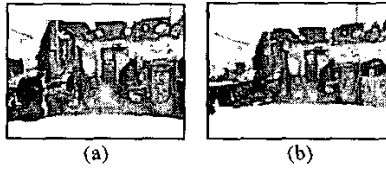


Figure 6. Zoom-in and zoom-out

Several scenes seen from a navigation-enabled area are shown in the Figure 7, and Figure 8 is a top-view of VE. As can be seen from Figure 8, the VE forms a rectangular shape. The right side shows another space out of room as shown in Figure 7(b).

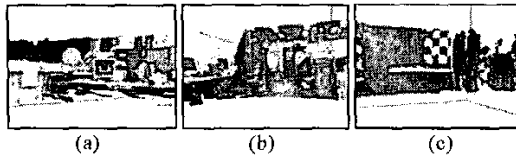


Figure 7. Scenes from seen different directions



Figure 8. Top-view of VE

Figure 9 explains the observable gain C (Eq. (8)) of a user according to the distance between two cameras.

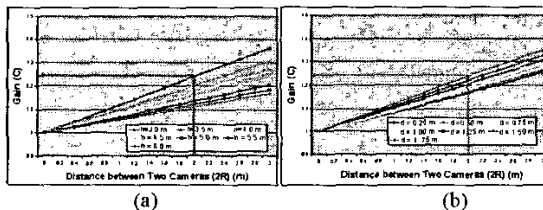


Figure 9. Observable gain according to the distance between two cameras (a) observable gain according to h in case of $d=0.5m$ (b) observable gain according to d in case of $h=4m$

For example, let us put a cylindrical object with a radius 0.5m in front of the rotational axis by 3m. If the distance between the two cameras is 2m, 24% gain is obtained. On the other hand, if the object radius is 1.75m and distance from the rotation axis to the object center is 4m, 24% gain is obtained.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new method to generate an image-based 3D panoramic VE. The proposed method provides a user with a wide FOV and enables the user with a HMD to naturally navigate the VE within a constant radius. It also provides a photo-realistic VE with less rendering time than the conventional CG-based VE. It also obtains observable modeling gain as the rotation radius increases. There are several remaining challenges. To reduce stitching error, more accurate camera calibration is needed. To apply the proposed method to real-time application we need a fast stitching algorithm. A natural composition between virtual objects and background VE requires light source estimation and analysis to match illumination condition of the VE.

5. REFERENCES

- [1] H. Chen, "Building Panoramas from Photographs Taken with a Hand-held Camera," Ph.D. Dissertation, University of Hong Kong, 2002.
- [2] S. K. Nayar, "Catadioptric Omnidirectional Camera," *IEEE Computer Society Conf. on CVPR'97*, pp.482-488, 1997.
- [3] H. Y. Shum and R. Szeliski, "Construction of Panoramic Image Mosaics with Global and Local Alignment," *Int'l. Journal of Computer Vision*, vol. 36(2), pp. 101-130, 2000.
- [4] S. Peleg, B. Rousso, A. Rav-Acha and A. Zomet, "Mosaicking on adaptive manifolds," *IEEE Trans. on PAMI*, vol. 22(10), pp. 1144-1154, 2000.
- [5] S. Peleg et. al. "Cameras for stereo panoramic imaging," *CVPR*, vol. 1, pp. 208-214, 2000.
- [6] H. Y. Shum and R. Szeliski, "Stereo reconstruction from multiperspective panoramas," *ICCV*, vol.1, pp.14-21, 1999.
- [7] R. Benosman and J. Devars, "Panoramic stereovision sensor," *Proc. 14th Intl. Conf. on PR*, pp. 767-769, 1998.
- [8] W. Woo, N. Kim and Y. Iwadata, "Stereo imaging using a camera with stereoscopic adapter," *Proc. of IEEE - SMC 2000*, vol.2, pp. 1512-1517, Oct. 2000.
- [9] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *Proc. of the Seventh IEEE Intl Conf.*, vol. 1, pp. 666-673, 1999.
- [10] Point Grey Research Inc., <http://www.ptgrey.com>
- [11] S. Kim and W. Woo, "Image-based 3D Mosaicking using Multiple 3D Cameras," *Korean Institute of Communications Sciences (KICS2002)*, vol.26, pp.58-61, 2002.
- [12] S. Kim and W. Woo, "Virtual Environment Generation using 3D Image-based Panorama," *15th Workshop on Image Processing and Image Understanding*, pp. 111-116, 2003.