Photo-realistic Interactive Virtual Environment Generation Using Multiview Cameras

Namgyu Kim†, Woontack Woo and Makoto Tadenuma

ATR MIC Labs
Kyoto 619-0288, Japan
E-mail:{ngkim,wwoo,tadenuma}@mic.atr.co.jp

ABSTRACT

In this paper, we report on a convenient and unified framework for generating a photo-realistic interactive virtual environment (piVE) using heterogeneous multiview cameras, while not using bluescreen techniques and special rendering hardware. In spite of the rapid growth of computer hardware, rendering a photo-realistic virtual environment on the fly is still a challenging problem. With the proposed framework, exploiting stereo images/videos, piVE can be rendered in realtime without using expensive high-end computer with rendering hardware. The proposed framework consists of three main parts, i.e. (i) photo-realistic virtual space generation exploiting a camera with stereoscopic adapter, (ii) generation of a video avatar (a special object representing the user) by exploiting multiview camera, and (iii) graphics object rendering according to the given camera parameters and the user's interaction. We also address z-keying issues among background video, graphics objects and video avatar.

Keywords: photo-realistic environment, interactive environment, virtual environment, z-keying, multiview

1. INTRODUCTION

While virtual reality is a powerful tool for a range of applications, its usefulness is mainly limited by the real time rendering of a realistic virtual environment (VE)*. Traditionally, virtual reality systems use 3D computer graphics to model and render virtual environments in real-time. In general, the modeling of photo-realistic VE (pVE) is difficult and labor-intensive. In addition, the rendering of complicated pVE on the fly requires expensive special purpose rendering hardware. In spite of the rapid growth of computers, the rendering quality and scene complexity are often limited because of the real time constraint.

Over the last few decades, geometry-based rendering (GBR) techniques have been widely adopted in virtual reality (VR) systems to generate VE. In general, the GBR techniques use geometric model to build virtual space and objects. With a well-calibrated tracking system, interaction or operation in the GBR-based VE is relatively easy, because each object in the space is well modeled and controlled. However, the GBR-based VE has difficulties in representing pVE on the fly, since required computational power goes far beyond the performance of available computers. Especially, a real-world scene is too complex to be rendered, *e.g.* a normal room consists of millions of polygons, *i.e.* giga bytes of data.

Meanwhile, as an alternative to traditional GBR, image-based rendering (IBR) techniques can generate pVE. The IBR gives the virtual space fairly realistic feeling since 3D space is represented based on captured images. However, there are still many challenges to be met before IBR gains the wide acceptance in VR systems. For example, the interaction with objects in IBR-based VE is very difficult since there is no geometric model in IBR-based approaches. The more modeling, the better interaction we experience.

In this paper, we present a new hybrid approach which uses a combination of image-based and geometry-based rendering approaches [1] . The proposed unified framework is a convenient way for generating a photo-realistic interactive virtual environment (piVE) on the fly, without using expensive rendering hardware. In addition, the user can interact with CG objects in piVE through the video avatar, a special object representing the user, which is generated without using bluscreen techniques. The proposed framework consists of three main parts, *i.e.* (i)

† Namgyu Kim is a Ph.D. candidate at Pohang University of Science and Technology (POSTECH), Pohang, Korea

*Though there is no standard definition for VE, we use the term VE to describe computer generated world in which users can interact with virtual objects and navigate through the environment.

photo-realistic virtual space generation exploiting a camera with stereoscopic adapter, (ii) video avatar generation exploiting multiview camera, and (iii) graphics object rendering according to the given camera parameters and the user's interaction.

In this framework, we exploit two heterogeneous multiview cameras to construct piVE. As shown in Figure 1, we first generate IBR-based VE using a portable stereo camera, *e.g.* camcorder with stereoscopic adapter [2]. Then, we capture the user using a multiveiw camera and segment the user out from natural video sequences by exploiting hybrid cues such as color, edge, motion, disparity, etc [3, 4]. The segmented user with depth information is used to generate a video avatar, which replace a CG avatar in piVE. Next, given the camera parameters, we render only a few CG objects to provide users with the illusion of interaction in piVE. The mixing among the virtual space, video avatar and CG objects, what is called *z-keying*, is performed by comparing pixelwise depth information. Finally, according to the user's interaction event (*e.g.* collision), we update the CG objects, instead of re-rendering the whole VE.

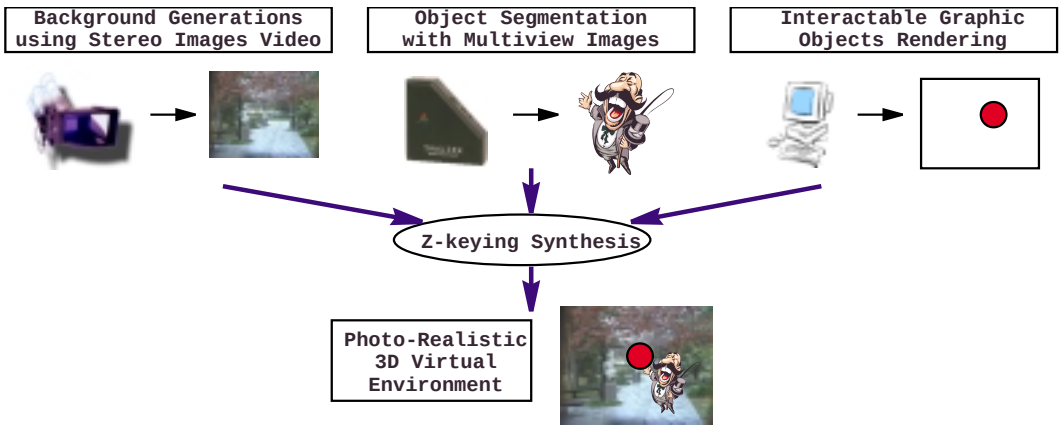


Figure 1. Basic structure for generating photo-realistic interactive virtual environment. we first generate IBR-based VE using a portable stereo camera. Then, we capture the user using a multiveiw camera and segment the user out from natural video sequences. The segmented user with depth information is used to generate a video avatar. Given the camera parameters, we render only a few CG objects to provide users with the illusion of interaction in piVE. Finally, according to the user's interaction event (*e.g.* collision), we update the CG objects, instead of re-rendering the whole VE.

This paper is organized as follows. In section 2, we explain the image-based background generation scheme exploiting stereo video. The video avatar generation scheme is explained in section 3. In section 4, we describe how CG objects are consistently mixed into the image-based virtual space. A resulting photo-realistic interactive virtual environment and discussion about possible applications are provided in Section 5 and 6, respectively.

2. PHOTO-REALISTIC BACKGROUND WITH STEREO VIDEO

With the proposed framework, exploiting stereo images/videos, IBR-based VE can be rendered on the fly without using expensive high-end computers. IBR has developed many new algorithms that avoid much of the overhead found in polygonal rendering. As a first step towards IBR-based piVE, we incorporate epipolar geometry analysis to process stereo video sequences. Exploiting stereo images in modeling of piVE provides various advantages over using an image. For example, stereo vision provides 3D information (such as orientations and distances) of the objects in the scene. However, the difficulties in stereo imaging mainly stem from capturing well-controlled stereo images, which is a key step toward accurate depth estimation.

In general, stereo images can be captured using a pair of stereo cameras, where each camera captures a scene from a slightly different perspective. However, several well-known problems arise from capturing stereo images/video sequences, since two cameras will generally have slightly different physical characteristics. Without accurate camera calibration, we may fail to estimate accurate 3D information and to provide realistic 3D effects on the screen. Meanwhile, a camera with a stereoscopic adapter, *e.g.* NuView system, can be considered as a new way to capture

stereo images/video sequences, because it can alleviate those problems arising from the different characteristics of a pair of stereo cameras [2]. Note that it also allows users to access all the functions built into the camcorder, e.g. zoom, auto-focus, auto-exposure, special effects, etc.

The stereoscopic adapter (e.g. Nu-View system) captures stereo video in field sequential format. The adapter consists of a sturdy black plastic housing, a reflecting mirror and liquid crystal shutters (LCS). The prismatic beam splitter and the orthogonally positioned polarizing surfaces (1.45" × l.25") in the LCS open and close the light valves, to record either the direct image or the mirror reflected image on alternate fields of the video. As a result, the left image is recorded during the "odd" field, and the right image is recorded during the "even" field, or vice versa. The field sequential format is necessary to be transformed to other formats, *e.g.* above/below format and/or side-by-side format, since stereo images in field sequential format makes further processing difficult. We transform the sequences to side-by-side format after a spatial interpolation such as linear or bilinear interpolation between lines.

To estimate 3D information from the stereo video sequences, we need to perform camera calibration which determines the distortion model and model parameters. In general, standard stereo camera calibration techniques follow 3-step procedures. First, they establish a list of 3D world coordinates and corresponding 2D image coordinates. Given this list, camera parameters are estimated for each camera using a set of equations. Finally, the epipolar geometry is constructed from the projection matrices. We calibrate the camera with a stereoscopic adapter using Tsai's calibration algorithm [5].

The Tsai algorithm estimates 11 model parameters: five intrinsic (also called internal or interior) and six extrinsic (also called external or exterior) parameters. The intrinsic camera parameters include the effective focal length f, the first order radial lens distortion coefficient κ_1, the principal point (the center of radial lens) $[c_x, c_y]$, and the scale factor s_x. The extrinsic parameters include the rotation matrix R (rotation angles for the transformation between the world and camera coordinate frames) and transformation matrix T (translational components for the transformation between the world and camera coordinate frames). After performing Tsai's algorithm, the rectification can be accomplished by exploiting the transformation matrix obtained from the relationship between the two sets of extrinsic parameters, $\{R, T\}_1$ and $\{R, T, \}_2$, where the subscript 1 and 2 denote left and right, respectively [2].

After proper calibration and rectification, we estimate disparity using various hybrid cues [2–4, 6]. Disparity estimation is recognized as the most difficult step in stereo imaging. Many disparity estimation algorithms have been proposed, but the resulting disparity is not accurate enough to be used in real applications. To overcome the weakness of available estimation schemes, we adopt a hierarchical block matching scheme with hybrid cues, which exploit edges as well as intensity similarity, based on MRF framework [2–4, 6]. We start with a block to maintain consistency of the estimation, and then segment the block according to the estimation error and variance level. The edge information is exploited to estimate accurate disparity along the object boundaries.

In addition, we adopt a moving object segmentation scheme to estimate relatively accurate depth information for moving objects, especially along object boundaries. As shown in Figure 2 (a) and (b), we first perform disparity estimation for a static scene and then we estimate disparity of the scene with moving objects. By exploiting statistics of the static scene, we can separate moving objects from the scene. The segmented moving objects with disparity can be combined to the static background scene. The more detailed explanation on the scheme is in Section 3. Given disparity information and camera parameters, 3D position can be calculated by triangulation. The resulting background video with depth information is ready to be used as a virtual space for piVE.

3. VIDEO AVATAR IN VIRTUAL ENVIRONMENT

The video avatar of the user is generated by exploiting a multiview camera, *i.e* moving object separation and video texture mapping. In VE, the user can usually be represented by a CG avatar, which has some communication ability in VE. The CG avatars have been widely accepted in VE since the shapes of avatars need not be an accurate representation of the user, or be human at all. However, the behavior of the avatar needs to be synchronized to the actions of the user so that the user experience interaction with VE through the avatar. In the proposed framework, we replace the CG avatar with a video avatar to improve realism of interaction. The video avatar is generated by mapping video texture of the segmented user on to the avatar.

There has been a growing interest in object segmentation to provide various applications with object-based functionalities such as interactivity and scalability. The capability of extracting moving objects from video sequences is a fundamental and crucial problem of many vision applications. The difficulties of automatic segmentation mainly

(a)

(b)

Figure 2. Photo-realistic virtual space generation. (a) static background (b) background with moving objects. We estimate more accurate disparity along object boundaries by adopting moving object segmentation scheme from a static scene.

come from defining semantically meaningful areas. Even in cases where we have a clear definition about the areas of interest, an accurate and reliable segmentation is a challenging and demanding task because those areas are not homogeneous with respect to low-level features such as intensity, color, motion, disparity, etc.

To alleviate these difficulties, several hybrid schemes have been proposed. Among those, motion information has been widely accepted as a crucial cue, under the assumption that the objects of interest (OOI) can be characterized by a coherent motion. Even in case of moving objects, motion similarity may not work well due to various error sources including occlusions and inaccurate motion estimation. Therefore, additional information is necessary to detect accurate boundaries of objects.

In the proposed framework, we adopt a slightly different approach to separate objects from the static background. The basic idea of separating moving objects from the background scene is to subtract the current image from a reference image which is acquired from a static background during a period of time. The subtraction leaves only non-stationary or new objects. The proposed segmentation algorithm consists of the following three steps: (1) static background modeling (2) moving object segmentation and (3) shadow removing.

The object separation technique based on static background modeling has been used for years in many vision systems as a preprocessing step for object detection and tracking. However, many of these algorithms are susceptible to both global and local illumination changes which cause the consequent processes to fail. The proposed normalized color space is able to cope with the slight change of illumination conditions. The proposed algorithm for detecting moving objects from a static background scene works fairly well on real image sequences of outdoor scenes, as shown in Figure 2.

We first set up the multiview camera according to the camera position information, $\{T, R\}$, which is obtained in camera calibration stage of the camera with stereoscopic adapter. We then gather statistics over a number of static background frames, *i.e.* $N(m, \sigma)$, where m and σ denote the pixelwise mean and standard deviation of the image. Let the color image be $I(R, G, B)$. The resulting mean image, $I_m(R, G, B)$, is used for the reference background image and the standard deviation, $I_\sigma(R, G, B)$, is used for extracting moving objects as threshold values. The mean image and standard deviation are calculated as follows:

$$I_m(R,G,B)_{ij} = \frac{1}{L} \cdot \sum_{t=0}^{L-1} I_t(R,G,B) \tag{1}$$

$$I_\sigma(R,G,B)_{ij} = \sqrt{\frac{1}{L} \cdot \sum_{t=0}^{L-1} (I_t(R,G,B) - I_m(R,G,B))^2} \tag{2}$$

where L denotes the total number of image frames used to estimate statistics. We choose L to be 30.

Each pixel can be classified into objects or background by evaluating the difference between the reference background image and the current image in the color space in terms of (R,G,B). To separate pixels in the moving objects from pixels in background, we measure euclidian distance in RGB color space, and then compare it with the threshold. The threshold is determined in order to obtain a desired detection rate in the subtraction operation. The distance, D_t, and threshold, Th, are defined as follow.

$$D_t = \sqrt{(I_t(R) - I_m(R))^2 + (I_t(G) - I_m(G))^2 + (I_t(B) - I_m(B))^2} \quad (3)$$

$$Th = \alpha \cdot (I_\sigma(R) + I_\sigma(G) + I_\sigma(B)) \quad (4)$$

where variable α is determined according to the changing lighting condition.

As shown in Figure 3, to incorporate chromaticity, we introduce normalized color space. Normally, moving objects make shadow and shading effects according to (changing) lighting conditions. However, the (R,G,B) color space is not a proper space to deal with shadow and shading effects. Normalized mean and color images, respectively, defined as follows.

$$I_m(r,g,b) = \frac{I_m(R,G,B)}{S_m(R,G,B)}, \quad (5)$$

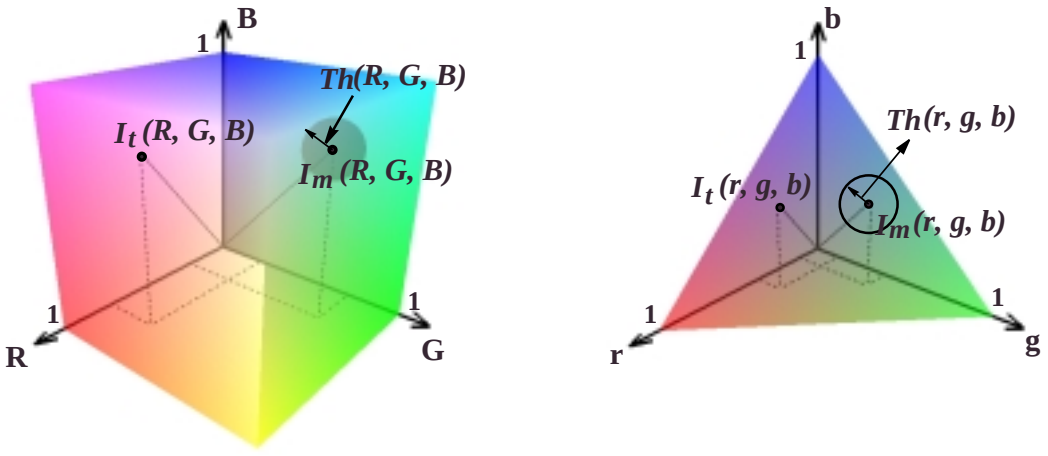
$$I_t(r,g,b) = \frac{I_t(R,G,B)}{S_t(R,G,B)} \quad (6)$$

where S_m and S_t, respectively, denote the summation of (R,G,B) component of the mean reference and t-th frame images, *i.e.* $S_m(R,G,B) = I_m(R) + I_m(G) + I_m(B)$ and $S_t(R,G,B) = I_t(R) + I_t(G) + I_t(B)$. The distance and threshold for the normalized color space, D_t and Th, are defined as follow.

$$D_t = \sqrt{(I_t(r) - I_m(r))^2 + (I_t(g) - I_m(g))^2 + (I_t(b) - I_m(b))^2} \quad (7)$$

$$Th = \alpha \cdot (I_\sigma(r) + I_\sigma(g) + I_\sigma(b)) \quad (8)$$

where variable α is determined according to the changing lighting condition.



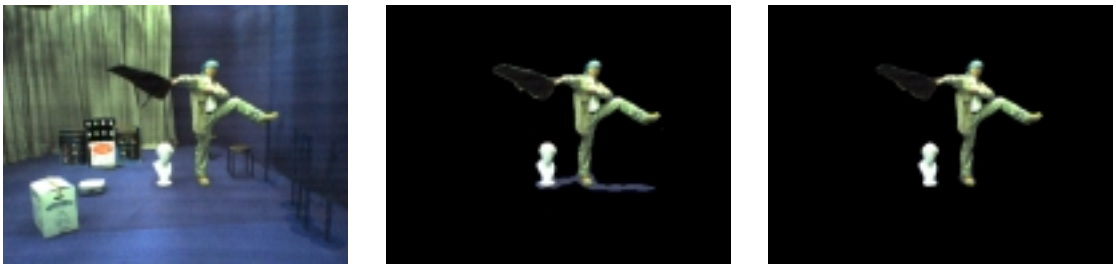
(a) (b)

Figure 3. Classification in color space. (a) classification in RGB space (b) classification in normalized RGB space. The current image $I_t(R,G,B)$ is performed pixelwise comparison with mean reference Image $I_m(R,G,B)$ and classified into background, if the difference is less than the threshold values, $f(I_\sigma)$, a function of $I_\sigma(R,G,B)$. Meanwhile, the objects further can be classified into shadow or shading in the normalized 2D color space. In the proposed framework, we classify the shadows as a part of the background.

In the proposed normalized color space, we can separate moving objects from background. The difference between the normalized color image $I_t(r,g,b)$ and the normalized mean of the reference image $I_m(r,g,b)$ is calculated. As

shown in Figure 3 (b), if the the difference of the pixel is larger than the threshold, then the pixel is classified as a part of background. Note that we can speed up the process by only checking the difference in the normalized color space, if the pixel is classified as a part of the objects. If the pixel is classified as objects, then we further decompose the color difference into brightness and chromaticity components. Then, based on the observation that shadow has similar chromaticity but slightly different (usually lower) brightness than those of the same pixel in the background image, we can label the pixels into a third category, *i.e.* shadow. Note however that, in the proposed framework, we classify the shadow as a part of the background, since we only need to separate moving objects from the background.

We also exploit depth information to reduce misclassification, while segmenting the moving object. The underlying assumption is that the moving objects have limited range of depth. Using this assumption we remove spot noises in the segmented object. In addition, we apply two types of median filter (7×7 and 3×3) to fill the hole while maintaining sharp boundaries. Figure 4 shows the segmented user from the natural background scene, without using bluescreen techniques. As shown in Figure 4 (c), we have a segmented object with sharp boundary, after applying the proposed separation scheme in the normalized color space.



(a) (b) (c)

Figure 4. Moving object segmentation. (a) current image (b) initial object segmentation (c) segmented object after shadow removing. To maintain sharp object boundaries we exploit hybrid cues including color, edge, depth, etc. To reduce mis-classification rate between objects and background, we also use median filters, which smooth out spot noise.

4. INTERACTIVE VIRTUAL ENVIRONMENT WITH CG OBJECTS

The interactive virtual environment (iVE) is usually constructed by adding CG objects, whose behaviors are controlled by external user input. In general, we can add three types of CG objects into the VE: (i) isolated objects that have no interaction with oyher objects in the VE, (ii) objects that interact with each other, and (iii) objects that interact with video avatars. The VE evolves according to the interaction events of those CG objects and video avatars.

In mixing CG objects into pVE, one of the most importance issues is the registration between CG objects and background scenes. To provide a natural looking VE, the exact position of the real camera must be known to place the CG objects correctly in the pVE. We exploit the camera parameters, which are obtained in the camera calibration stage while generating virtual space [2,5].

Another important issue to achieve a more natural composition is taking into account the shadow and shading effects, according to changing lighting conditions. In the proposed framework we assume that lighting conditions are not significantly changing and the direction of light sources are known. These assumptions work fairly well because we use outdoor scenes as background video and, according to given lighting condition, we can control lighting condition of the CG objects and thus accordingly change shadow/shading effects for the rendered CG objects, as shown in Figure 5.

5. PHOTO-REALISTIC INTERACTIVE VIRTUAL ENVIRONMENT

In general, natural 3D mixing of two videos/images is still a challenging task. The well-known chroma-keying is a 2D technique, which cuts out foreground objects using chroma-keying and then composites them with separately shot background scene [7,8]. As a result, the real objects are always in front of 2D background scene. The constraint of chroma-keying technique can be overcome by the z-keying (distance-keying) that composites objects with natural

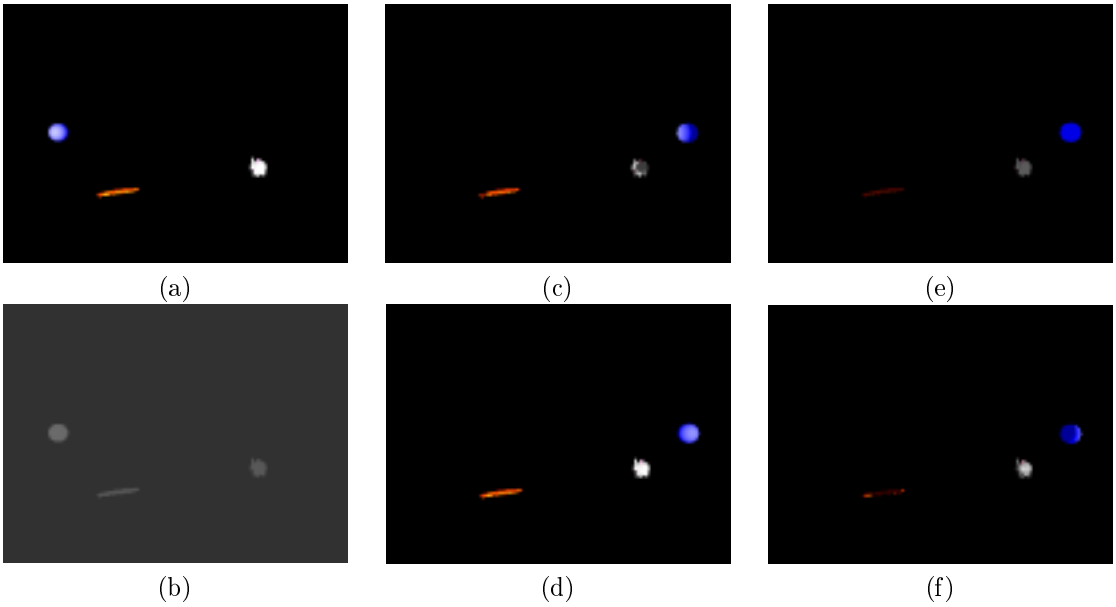


Figure 5. CG objects with changing lighting conditions. (a) image (b) depth value (c) light source in West (d) light source in North (e) light source in South (f) light source in East. According to given lighting condition, we can control lighting condition of the CG objects and thus accordingly change shadow/shading effects for the rendered CG objects.

or/and synthetic imagery using pixelwise depth information [9, 10]. The z-key switch properly composites real and real/virtual objects by comparing depth information of both images for each pixel, *e.g.* the nearer one to the camera will occlude the others. However, the usage of the conventional z-keying scheme is usually restricted to the bluescreen environment to separate objects from one video sequence [7, 8].

In the proposed framework, the z-keying between video avatar and background is performed without using bluscreen techniques. As explained in Section 3, the user is segmented from nature scene using proposed object segmentation scheme, exploiting normalized color space and depth information. In addition, the segmented object result in video avatar which has depth information by exploiting stereo geometry. We also maintain consistency in z-keying among virtual space, video avatars and CG objects by exploiting camera parameters of multiview cameras.

We use two two different types of multiview cameras, a camera with stereoscopic adapter (SONY TRV900 with NuView) [11] and a well-calibrated multiview camera (DigiClops) [12]. Due to its portability and simplicity, the Nuview adapter [2] is used to generate photo-realistic virtual space. Meanwhile, the DigiClops is used to generate video avatar on the fly. Since we use two different types of multiview cameras, in order to make z-keying natural-looking, we need to know exact camera parameters including position and rotation matrices for both cameras.

To maintain the consistency between the real space of the user and the virtual space of the video avatar, we need to use unified camera parameters. In this framework, the parameters of the camera with adapter are used as a reference. We first exploit pre-calculated camera parameters to set up the multiview cameras, to make procedure simple. Then, we compensate for the position and rotation errors of the multiview camera by finding a principal axis in 3D from the disparity information of the segmented user. The camera position and rotation information can be estimated based on the assumption that a line passing through the user is perpendicular to the floor. To preserve the original 3D geometry of the composed image, the video avatar has to be enlarged or reduced in proportion to the focal distance. Let the video avatar and background video be a and b, respectively. As shown in Figure 6, the depth between the camera and the object in the composed image Z_{ba} can be approximated as follows [13]

$$Z_{ba} = Z_a \times \frac{f_b}{f_a} \tag{9}$$

where Z_a denotes the distance between the camera and the object. The focal lengths of the foreground and background cameras are f_a and f_b, respectively. As a result, the user would feel that the absolute size of the video avatar

in the composed image remains unchanged. In case of camera movements, it may also require synchronizing camera movements or synthesizing views of real background video scene. Without loss of generality, we also add CG objects into the piVE.

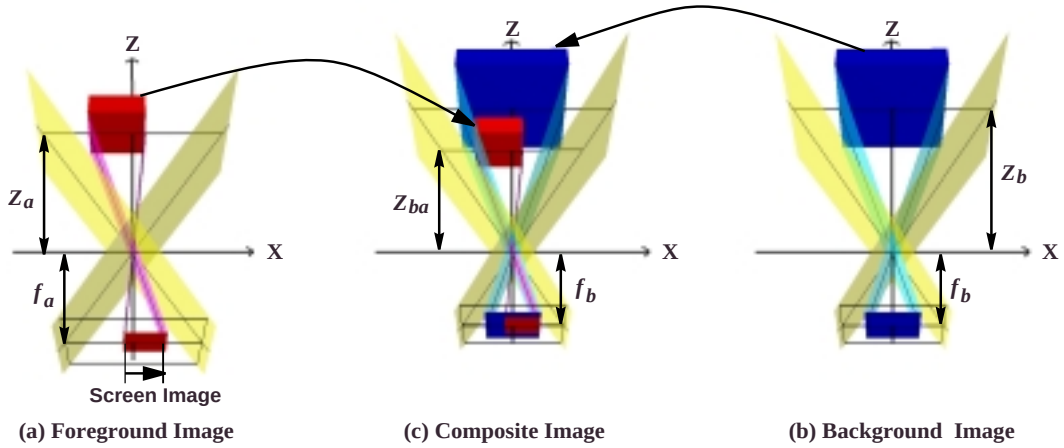


Figure 6. Z-keying maintaining size consistency [13]. (a) the video avatar (b) the background video (c) composite video. To preserve the original 3D geometry of the composed image, the the foreground or background has to be enlarged or reduced in proportion to the focal distance. Let the focal lengths of the foreground and background cameras be f_a and f_b, respectively. The depth between the camera and the object in the composed image Z_{ba} can be approximated as $Z_{ba} = Z_a \times \frac{f_b}{f_a}$, where Z_a denotes the distance between the camera and the object.

Figure 7 and 8 show the z-keying process and the final piVE, respectively. The whole objects have disparity information by exploiting multiview images and hardware z-buffer. Finally, the user can interact with CG objects in the piVE through the video avatar. We update the virtual CG objects, instead of the whole VE, if the interaction between the video avatar and the CG objects are detected. Although the interaction in the resulting piVE is limited since there is no exact geometric model in this method, we still experience the illusion of interaction by the manipulation of the CG objects, which give fairly realistic feeling. We also can add shadow and shading effects to the video avatar as well as CG objects, by exploiting the information of the light source in the background video. [3, 4].

6. SUMMARY AND DISCUSSION

In this paper, we proposed a cost-effective way to yield photo-realistic interactive virtual environment by exploiting multiview images/video. We first generate photo-realistic virtual environment by exploiting stereo video. We then segment the objects from natural static background to generate video avatars and then mix the video avatars into photo-realistic virtual environment. We finally add computer generated virtual objects into the environment, which makes user experience the illusion of interaction with the photo-realistic virtual environment, without modeling and rendering the whole virtual environment.

According to preliminary experiments, we show how the proposed photo-realistic interactive VE can replace computer-generated 3D graphics VE. The proposed framework will play a key role in developing immersive entertainment systems such as ATR MIC Interactive DAnce System (MIDAS) [3, 14] and ATR I-cubed Tangible Music System [15, 16]. The remaining technical challenges for the proposed framework to enable wide-ranging applications are to include the camera movement and to improve interactivity.

REFERENCES

1. W. Woo, http://www.mic.atr.co.jp/ wwoo/Resaerch/ATR-MR, *PMR: Photo-realistic Mixed Reality.*
2. W. Woo, N. Kim, and Y. Iwadate, "Stereo imaging using a camera with stereoscopic adapter,," *Proc. of IEEE Intl. Conf. on SMC* , pp. 1512–1517, Oct. 2000.
3. W. Woo and Y. Iwadate, "Object-oriented Hybrid Segmentation using Stereo Images," *Proc. SPIE PW-EI-IVCP* , pp. 487–495, Jan. 2000.

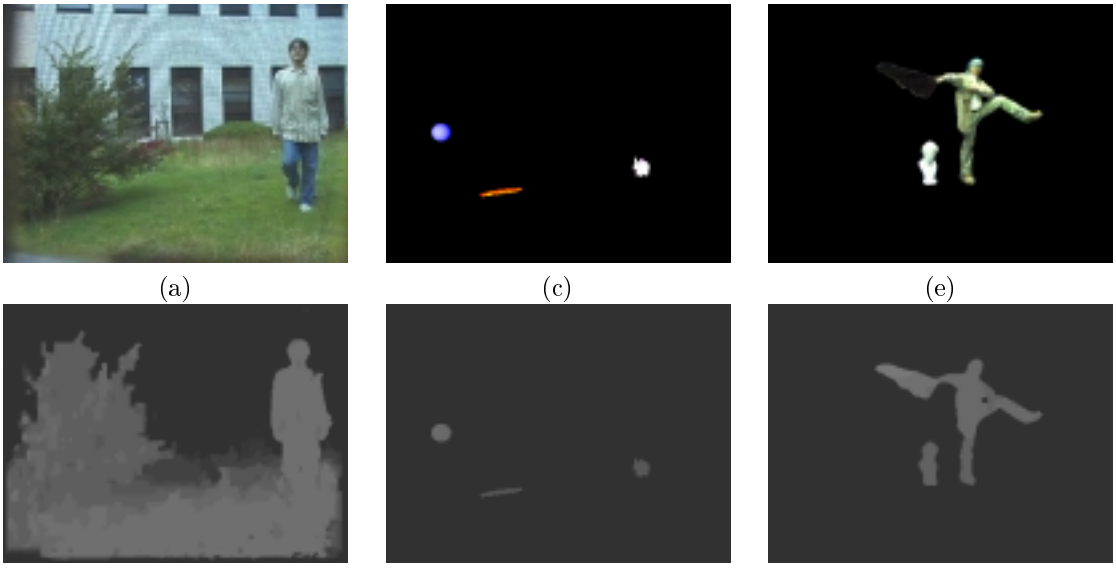


Figure 7. Components of virtual environment. (a) Background video (b) depth image of the background video (c) CG object (d) depth image of the CG objects (e) video avatar (f) depth image of the video avatar.

4. W. Woo, N. Kim, and Y. Iwadate, "Object segmentation for z-keying using stereo images," *Proc. of IEEE Intl. Conf. on ICSP* , pp. 1249–1254, Aug. 2000.
5. R. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal of Robotics and Automation* **3**, pp. 323–344, August 1987.
6. W. Woo and A. Ortega, "Overlapped block disparity compensation with adaptive windows for stereo image coding," *IEEE Trans. on CSVT* **10**, pp. 194–200, Mar. 2000.
7. S. Gibbs, C. Arapis, C. Breiteneder, V. Lalioti, S. Mostafawy, and J. Speier, "Virtual Studios: An Overview," *IEEE Multimedia* **5(1)**, January/March 1998.
8. M. Hayashi, "Image Compositing Based on Virtual Cameras," *IEEE Multimedia* **5(1)**, January/March 1998.
9. L. Blonde, M. Buck, R. Calli, W. Niem, Y. Paker, W. Schmidt, and G. Thomas, "A Virtual Studio for Live Broadcasting: The Mona Lisa Project," *IEEE Multimedia* , pp. 18–29, Summer 1996.
10. K. Oda, M. Tanaka, A. Yoshida, H. Kano, and T. Kanade, "A Video Rate Stereo Machine and its Application to Virtual Reality," *Proc. of ISPRS* , 1996.
11. 3-D Video Inc., http://www.3-dvideo.com, *NuView Owner's Manual.*
12. Point Grey Research Inc., http://www.ptgrey.com, *Triclops: Stereo Vision SDK.*
13. NHK STR Labs, NHK Labs Note No. 447, *Virtual Studio System for TV Program Production.*
14. R. Suzuki, Y. Iwadate, M. Inoue, and W. Woo, "MIDAS : MIC Interactive Dance System," *Proc. of IEEE Intl. Conf. on SMC* , pp. 751–756, Oct. 2000.
15. W. Woo, http://www.mic.atr.co.jp/ wwoo/Resaerch/AIMS, *ATR IMS: ATR I-cubed Media System.*
16. W. Woo, N. Kim, and M. Tadenuma, "Sketch on dynamic gesture tracking and analysis exploiting vision-based 3d interface,," *Proc. of SPIE Intl. Conf. on VCIP* **4310**, Jan. 2001.

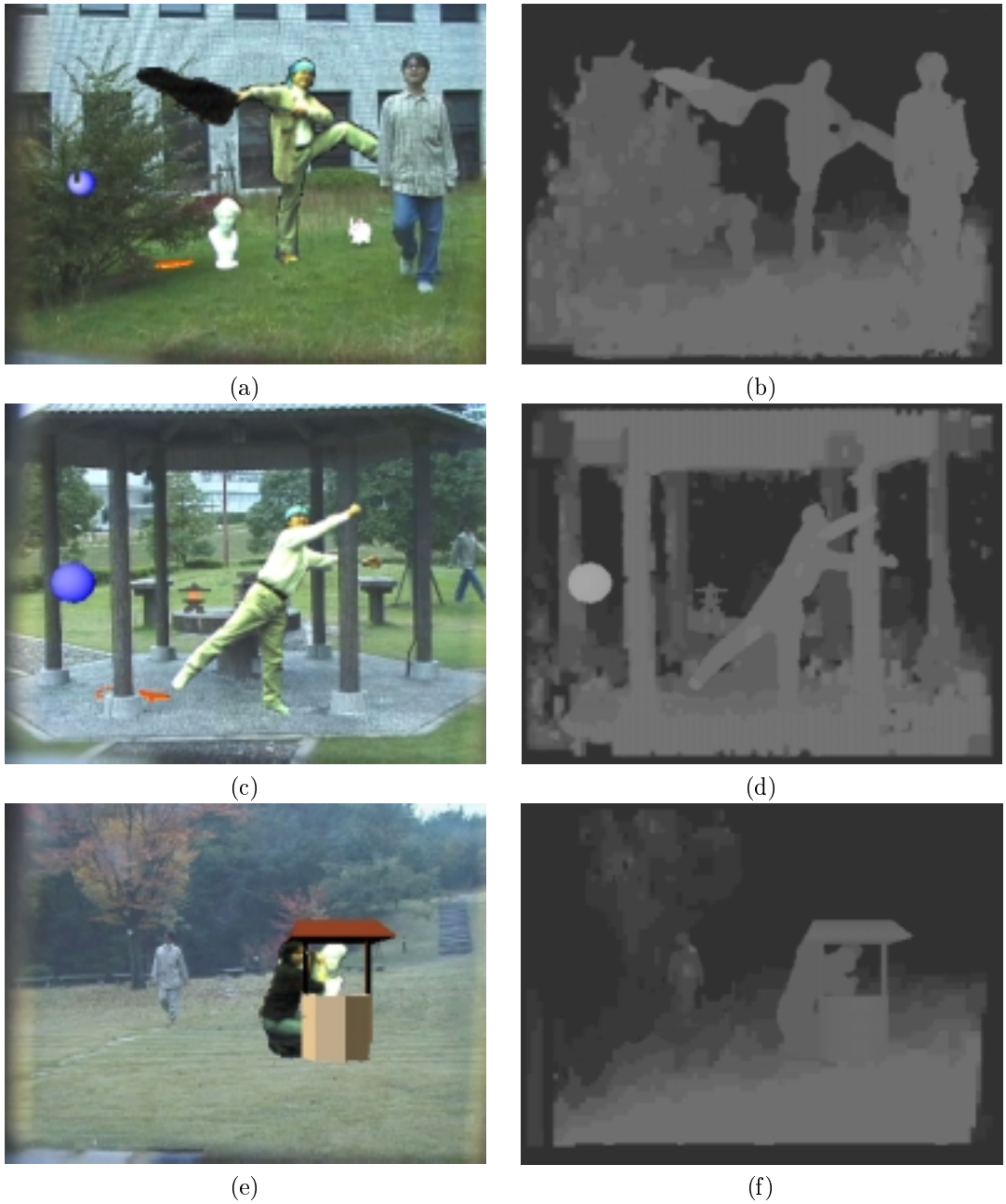


Figure 8. Z-keying for photo-realistic virtual environment (piVE). (a) composed image I (b) depth image of image I. (c) composed image II (d) depth image of image II. (e) composed image III (f) depth image of image III. By comparing pixelwise depth information, we mix CG objects and video avatar into ipVE. The user interact with the CG objects in the piVE through the video avatar. As a result, the piVE rendered without high-end computers and rendering hardware by combining image-based rendering and model-based rendering techniques.