3-D Virtual Studio for Natural Inter-"Acting"

Namgyu Kim, Woontack Woo, *Member, IEEE*, Gerard J. Kim, *Member, IEEE*, and Chan-Mo Park

***Abstract*—Virtual studios have long been used in commercial broadcasting. However, most virtual studios are based on "blue screen" technology, and its two-dimensional (2-D) nature restricts the user from making natural three-dimensional (3-D) interactions. Actors have to follow prewritten scripts and pretend as if directly interacting with the synthetic objects. This often creates an unnatural and seemingly uncoordinated output. In this paper, we introduce an improved virtual-studio framework to enable actors/users to interact in 3-D more naturally with the synthetic environment and objects. The proposed system uses a stereo camera to first construct a 3-D environment (for the actor to act in), a multiview camera to extract the image and 3-D information about the actor, and a real-time registration and rendering software for generating the final output. Synthetic 3-D objects can be easily inserted and rendered, in real time, together with the 3-D environment and video actor for natural 3-D interaction. The enabling of natural 3-D interaction would make more cinematic techniques possible including live and spontaneous acting. The proposed system is not limited to broadcast production, but can also be used for creating virtual/augmented-reality environments for training and entertainment.**

***Index Terms*—Augmented reality, calibration, image segmentation, multiview and stereo cameras, registration, three-dimensional (3-D) interaction, three-dimensional (3-D) model reconstruction, virtual environment (VE), virtual studio, z-keying.**

I. Introduction

VIRTUAL studios have long been in use for commercial broadcasting and motion pictures. Most virtual studios are based on "blue screen" technology, and its two-dimensional (2-D) nature restricts the user from making natural three-dimensional (3-D) interactions. Actors have to follow prewritten scripts and motion paths, and pretend as if directly interacting with the synthetic objects. This often creates an unnatural and seemingly uncoordinated output. This is quite apparent when the broadcast is live and the processing to correct such out-of-synch output is nearly impossible on the fly. For instance, it is common to notice, in a live broadcast of weather-forecast news, an incorrect timing on the part of the announcer (e.g., the weather map being pulled out after a pause).

Another limitation of the current virtual studio is in the creation of the backdrop. In the example of the weather forecast, the backdrop was quite simple with 2-D drawings of weather maps and temperature charts. 2-D images or videos can be used as a backdrop as well (to make the user seem being somewhere else), but the current technology does not allow for the actor to interact with the environment. To situate the actor in a more complex environment and enable 3-D interaction with it, the operating environment must be made in 3-D. In general, however, the modeling of realistic 3-D virtual environments (VE) is difficult and labor intensive. In addition, the rendering of the VE on the fly (plus mixing it with the image of the actor) requires expensive special-purpose rendering hardware.

In this paper, we introduce an improved framework for virtual studios that enables actors/users to interact, in 3-D, more naturally with the synthetic environment and objects. The proposed framework consists of four main parts: 1) image-based 3-D-environment generation using a portable camera with stereoscopic adapter; 2) real-time video-actor generation using a multiview camera; 3) mixing graphical objects into the 3-D scene; and 4) support for "natural" interaction such as collision detection, first-person view, and tactile feedback. The proposed system is a hybrid approach of augmented- and virtual-reality techniques, and is a relatively inexpensive method for generating a 3-D image-based interactive environment in real time, without special rendering hardware and a blue-screen-studio setup.

This paper is organized as follows. In Section II, we review prior researches related to this work such as virtual studios, 3-D information-processing techniques, object registration, image-based rendering (IBR), and 3-D interaction techniques. Section III will present an overview of the proposed interactive virtual studio, and detailed technical details about the four main components of the system follow in the subsequent sections. In the final sections, we illustrate the implemented system and evaluate its merits and limitations in the context of virtual studios. Finally, we conclude the paper with an executive summary and a discussion of new possible applications and future work.

II. Related Work

In general, virtual-studio sets require "blue-screen" (chromakeying) technology, high-end graphic workstations, camera-tracking technology and signal compositing for high realism, and exact mixing results [5]. The 2-D nature of current virtual studios as developed and used in the current broadcast industry limits its use to situations where the camera angle is fixed and there is minimal user interaction [6].

Manuscript received February 10, 2004; revised July 27, 2004, November 14, 2004, and December 23, 2004. The work described in this paper was supported in part and carried out at Advanced Telecommunications Research (ATR) Media Integration and Communications (MIC) Laboratory, Japan, from 2000 to 2001, and also by the Korean Ministry of Education's Brain Korea (BK) 21 program and the Korean Ministry of the Information and Communication's Information Technology Research Center (ITRC) program, and continued at the Pohang University of Science and Technology (POSTECH) Virtual Reality (VR) Laboratory. This paper was recommended by Associate Editor I. Gu.

N. Kim, G. J. Kim, and C.-M. Park are with the Department of Computer Science and Engineering, Pohang University of Science and Technology, Pohang, Korea (e-mail: ngkim@postech.ac.kr; gkim@postech.ac.kr; parkcm@postech.ac.kr).

W. Woo is with the Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), Gwangju, Korea (e-mail: wwoo@gist.ac.kr).

Digital Object Identifier 10.1109/TSMCA.2005.855752

Yamanouchi *et al.* overcame the limitations of the cost and space for the conventional blue-screen setups in their "Real Space-based Virtual Studio" system [7]. This system combines the real- and virtual-space images. The virtual-environment image sequences are precaptured by a 360° omnidirectional camera. The real space (foreground) images are obtained by the Axi-vision camera, which can capture color and depth information. The real and virtual images are mixed using the depth information in real time, but their algorithms are limited only to indoor studio sets.

Advanced video-conferencing systems share some common functionalities as those of the virtual studio. Daniillidis and co-workers developed a large-scale teleconferencing system [26], [27] and it included functionalities, such as 3-D model acquisition from bi/tri-stereo image sequences, view-dependent scene description, and also large-scale data transmission for a remote site. Schreer and co-workers presented a teleimmersive conferencing system called the VIRtual Team User Environment (VIRTUE) [28], [29], and a similar system called the Coliseum was developed by Baker *et al.* at [30]. However, VIRTUE emphasized the problem of stereo calibration, multiple-view analysis, tracking and view synthesis, rather than issues like interacting with the environment.

While chromakeying enables simple 2-D compositing, z-keying (distance-keying) allows compositing of objects and environments in 3-D using pixelwise depth information [12], [14]. However, z-keying is mostly used for separating foreground objects from a video sequence in a blue-screen environment [7], [12]. Kanade demonstrated a 3-D compositing technique using full 3-D-environment depth information [13]. However, it required extensive equipment and a complex setup to extract and mix the objects.

The most straightforward method of extracting 3-D information from a scene is to use multiple camera views and stereo correspondences. Scharstein and Szeliski gave a very good survey and taxonomy of the different algorithms and approaches to the stereo-correspondence problem [34]. There also have been approaches to 3-D depth-information acquisition by methods other than the traditional expensive stereo or multiple-camera setups. Yang *et al.* introduced a new method for using commodity graphics hardware to achieve real-time 3-D depth estimation by a plane-sweeping approach with multiresolution color-consistency tests [31]. The 3-D depth information can also be calculated by a motion analysis in short image sequences. For instance, Brodsky *et al.* investigated the processes of smoothing as a way of estimating structure from 3-D motion [33]. Nayar *et al.* described a nontraditional stereo image-capture method (and system design guidelines), called the "catadioptric stereo," using a single camera with a mirror [32].

IBR offers a solution to viewing images (e.g., the virtual-studio backdrop) at different angles, thus producing highly realistic output [15], [16], [18]. However, with IBR techniques, it is necessary to establish correspondences among pixels of two given images to produce a view-independent output, which amounts, again, to the 3-D-information extraction problem. Even if the extraction problem was solved, it would still be difficult for the user to interact meaningfully with image-based rendered scene because the technique was developed mostly for static images. However, techniques for simple and limited interactions, like navigation [17], [20], [37], selection, and manipulation [19], in an IBR environment have been proposed.

Another essential element in enabling natural 3-D interaction in a virtual-studio setting is the extraction of the actor and tracking his/her movements (whole or partial). Gavrila [8] has demonstrated a human-extracting and -tracking algorithm for user interaction for virtual reality (VR) systems and Cheung [9] devised a similar algorithm for a smart surveillance system. The Artificial Life Interactive Video Environment (ALIVE) system [11] used the extraction and tracking algorithm developed by Wren *et al.* [10] to analyze user intentions and control an avatar in a VE, although not a demonstration of a user's direct interaction with his/her environment. Urtasun and Fua developed an approximate vision-based tracking system for a walking or running human using a temporal motion model [38] and, similarly, for articulated deformable objects [39]. In general, real-time vision-based reconstruction of general articulated motions in a robust way still remains a challenging problem. Using separate sensors can surely make the problem somewhat more feasible as demonstrated by [40] and [41].

Interaction in the 3-D virtual space (virtual studio being one such example) has been one of the main research topics for VR researchers [49]–[51]. An important design goal for 3-D interaction is "naturalness," which refers to the resemblance to the way humans interact in the real world. The main characteristics of natural interaction include the use of multimodality, particularly the use of tactility for close-range interaction, and direct and noncognitive interaction [46]. In the context of virtual studios, our goal is to provide a way to produce "natural-looking" interaction to convince the viewers. Ironically, this, in turn, also asks for the actor to act (and interact) naturally. Thus, the actor, if possible, must be able to see and directly touch the interaction objects in the virtual-studio setting.

The final phase in a virtual-studio production is the mixing of the backdrop and actor image. Apostoloff and Fitzgibbon [35] and Rother *et al.* [36] introduced video-matting algorithms based on background/foreground extraction by considering the spatiotemporal color-gradient relationships. Their algorithms can also be applied to the actor-segmentation process for scenes with a general static background.

III. System Overview

Fig. 1 provides the overall structure of the proposed framework. As shown in Fig. 1, we first generate the image-based environment using a portable stereo camera, e.g., camcorder with stereoscopic adapter [1]. Then, we capture the user, using a multiview camera, e.g., Digiclops [22] (which can extract color and depth information in real time) and segment the user out from natural video sequences with cues such as color, edge, motion, disparity, etc [2], [3]. Next, given the camera parameters, we render computer-graphic objects (on a large screen display for the user to see) with the backdrop to provide users with the illusion of interacting with the environment. The mixing of the 3-D real-image-based environment, the user image, and computer-graphic objects is performed by "z-keying"

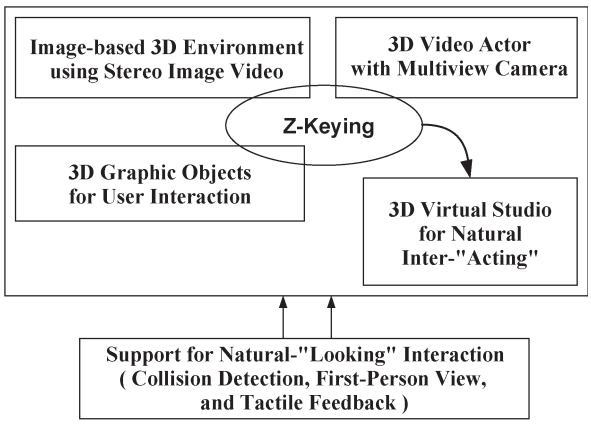

Fig. 1. Basic structure of generating the proposed virtual-studio set. We first generate an image-based environment using a portable stereo camera. Then, we capture the user with a multiview camera and segment the users out from natural video sequences. Given the camera parameters, we render computer-graphic objects to provide interaction. We also add support for natural-looking interaction, such as collision detection, first-person view, and tactile feedback.

that compares pixelwise depth information [4]. According to the users' interaction events, we update the computer-graphic objects and the relevant part of the 3-D backdrop environment rather than re-rendering the whole environment. We also add support for natural-looking interaction such as collision detection, first-person view for the actor (where possible), and direct interaction and tactile feedback.

IV. Three-Dimensional Real Image-Based Environment

In this section, we explain how we generate the 3-D environment to be used as the backdrop for acting in the virtual studio. Recent image-based techniques allow real-time rendering without using expensive high-end computers and avoid the overhead found in polygonal rendering. Thus, we propose to use an image-based approach by first taking stereo video sequences of the environment and to use epipolar geometry analysis to extract 3-D information. Stereo image sequences obviously provide various advantages for our purpose, over using a single-frame image, providing 3-D information (such as orientations and positions) of the objects in the scene. However, the difficulty in stereo imaging mainly rise in capturing well-controlled stereo images, which is a key step to acquiring an 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. For one, two cameras will generally have slightly different physical characteristics, and accurate camera calibration is a must for estimation of accurate 3-D information and, furthermore, realistic 3-D effects on the screen. In our system, to avoid the aforementioned problem, we use a single camera with a stereoscopic adapter, called the NuView [21] system pictured in Fig. 2.

The optical adapter, placed in front of the lens of the camera, allows for the camera to capture stereo video sequences. As a result, while it alleviates many of the problems that usually

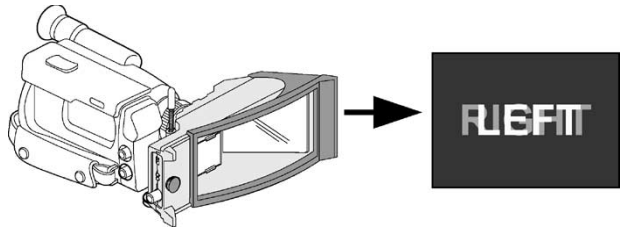

Fig. 2. Stereoscopic video camera (a camera with the NuView stereoscopic adapter).

arise in using different cameras for different views, the quality of the resulting stereo video sequences is reduced. First, there exist distortion introduced by the mirror and the associated optical system. Furthermore, the adapter captures each image of the stereo pair in a different line, i.e., field-sequential format (i.e., in half the resolution). Therefore, it is essential that these shortcomings are compensated for the camera to be used in real 3-D-vision-based applications. In the following sections, we focus on the compensation method for these 3-D distortions.

A. Stereoscopic Video With a Single Camera

As shown in Fig. 2, the stereoscopic 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 $(1.45'' \times 1.25'')$ 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 on the odd field and the right image on the even field. The synchronization of the light valves with the alternating fields of the camcorder is achieved through the cable connecting the video-out of the camcorder and the connector in the adapter.

It is then necessary to convert the video sequence to the "above/below" format. As explained earlier, the adapter produces a field-sequential stereoscopic 3-D video by simultaneously recording the second eye view to the camcorder. The resulting field-sequential video can be displayed on a 2-D (TV) monitor or a 2-D screen with special stereo glasses. The field-sequential format, however, is an inconvenient format to be used in various other vision applications. For example, applying image processing, such as filtering or transformation, to such field sequential video-data format, can cause a loss in quality of the stereo images because such processing propagates the effects into the interlaced lines, and thus produces 3-D artifacts. For the same reason, the available video-compression scheme cannot be exploited to save the hard-disk space or limited channel bandwidth. Therefore, we first separate the field-sequential format to the "above/below" format, where the left image is placed to the top part of the image and the right image to the bottom part.

After field separation, we transform the image to a "side-by-side" format. Spatial and temporal interpolations are applied to each image to produce high-resolution 2-D/3-D images/video sequences. There exists flickering effects when displaying the stereoscopic 3-D videos, when captured using the adapter in 60 Hz. The stereoscopic 3-D video in 60 Hz is not as smooth as compared to the 2-D video in 60 Hz, because the effective

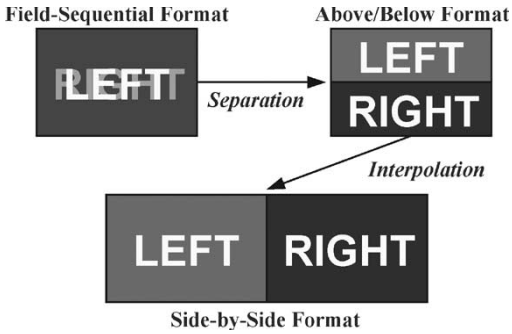

Fig. 3. Capturing stereo video sequences: From field-sequential format to side-by-side format.

refresh frequency drops to 30 Hz with the monitor allocating 30 Hz to the left and the right images, respectively. In addition, displays (such as head-mounted display, polarized screen, or autostereoscopic display) require projecting an image in the original size to provide comfortable 3-D display. Spatial interpolation is also required in 2-D applications using only 3-D depth information (e.g., z-keying). Spatial interpolation is achieved by first doubling the size (by line copy), then by linear interpolation between lines as follows

$$\begin{cases} F_{\mathrm{L}}^{2i} = G_{\mathrm{L}}^{i} \\ F_{\mathrm{L}}^{2i+1} = \frac{\left(G_{\mathrm{L}}^{i} + G_{\mathrm{L}}^{i+1}\right)}{2} \end{cases}. \tag{1}$$

F_{L} and G_{L} denote the left images in the side-by-side and above/below formats, respectively. The superscript i represents the index of the row in the image. The right image can be interpolated in a similar way. Fig. 3 shows procedures for producing a stereoscopic image.

B. Stereo Calibration and Error Compensation

To estimate the 3-D information from the stereo video sequences, we first need to perform a camera calibration that determines the distortion model and model parameters. We first capture the relationship between the 3-D world coordinate and its 2-D perspective projection onto the virtual stereo cameras by observing a calibration object, e.g., a black and white noncoplanar square grid pattern, whose geometry is known with respect to the 3-D coordinate system attached to this stereo camera. The resulting camera parameters allow us to learn the distortions that occurred during projection through the stereoscopic adapter also.

With the list of 3-D world coordinates and the corresponding 2-D image coordinates, camera parameters are estimated for each camera by solving a set of simultaneous equations. Finally, the epipolar geometry is constructed from the projection matrices. Here, we use the Tsai's algorithm to determine the distortion model and model parameters [23]. The algorithm estimates 11 model parameters: five intrinsic and six extrinsic parameters. The intrinsic camera parameters include the effective focal length f, the first-order radial-lens distortion coefficient $k1$, the principal point (the center of the radial lens) $(c_x,\ c_y)$, and the scale factor to account for any uncertainty due to frame-grabber horizontal scanline resampling s_x. The extrinsic parameters include the rotation matrix R and the translational component T. After performing Tsai's algorithm, the rectification can be accomplished by exploiting the transformation matrix obtained from the relationship between two sets of extrinsic parameters, $\{R, T\}_{\mathrm{left}}$ and $\{R, T\}_{\mathrm{right}}$ [1].

Before we exploit the stereo images to estimate the pixelwise depth, we need to calibrate for the color values. The color distortion occurs due to another inherent weakness of capturing stereo video with the stereoscopic adapter. The orthogonally positioned polarizing surfaces in the adapter yield stereo video sequences with different levels of color. Note that color equalization not only allows comfortable stereoscopic 3-D display, but also helps to estimate accurate 3-D depth information, especially when the depth is estimated based on the intensity level. The color-equalization approach usually takes advantage of the method of "histogram matching" [42]. Our approach is similar, but we use a new relational transition function based on the images' color statistics.

To equalize the color levels of both images in the stereo pairs, we use three test pairs, where each pair contains of only one color, i.e., red, green, or blue. We first select the region of interest from the given pairs of stereo images and then estimate statistics regarding the color distortion. Given the statistics, we normalize and modify the color histograms. The new intensity (color) of the right image, the projected image through the mirror, $F_{\mathrm{R\_N}}$ is normalized using the formula defined as follows

$$F_{\mathrm{R\_N}} = \frac{\sigma_{\mathrm{L}}}{\sigma_{\mathrm{R}}}(F_{\mathrm{R}} - m_{\mathrm{R}}) + m_{\mathrm{L}}. \tag{2}$$

F, σ, and m represent the image, standard deviation and average, respectively. The subscripts L and R denote left and right images, respectively.

C. Disparity Estimation

After proper calibration and color equalization, we estimate disparity using a collection of cues. Disparity estimation is recognized as the most difficult step in stereo imaging. The task of disparity estimation in a 3-D reconstruction is to find correspondences among a pair of stereo images and estimate 3-D positions by triangulation. We use a hierarchical block-matching scheme with several cues like the edges and intensity similarity based on the Markov random field (MRF) framework [2]. This framework involves two steps: 1) the hierarchical overlapped block matching; and then 2) pixelwise refinement. First, the disparity is estimated at a coarse level, e.g., block size of 16×16, using the full-search block matching. Then, the block size decreases to 8×8, only if the block yields higher compensation error than a prefixed threshold value. Given the initial disparity field and edge information of the stereo pair, we perform a pixelwise disparity estimation based on a coupled MRF/Gibbs random field (GRF) model [3].

Although we perform stereo calibration, there are still some errors (e.g., a few line pixel differences) that occur from the spatial interpolation process by (1). Our hierarchical block-matching algorithm can overcome these pixel errors. That is, by

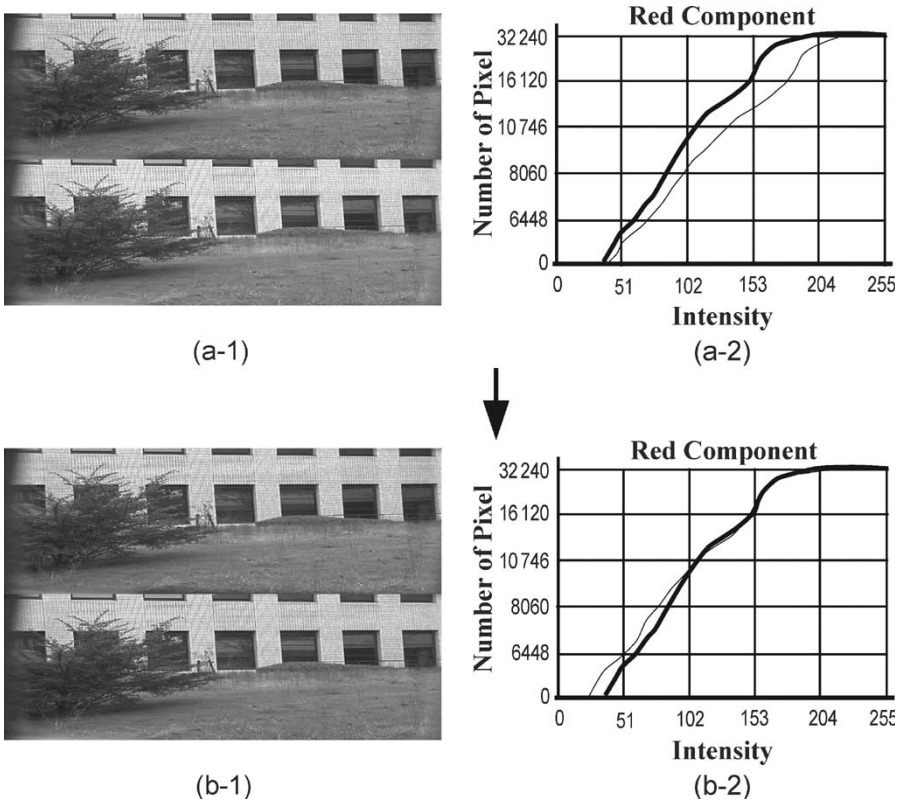

Fig. 4. Color compensation: (a-1) Original above/below image, (a-2) red-component histogram of (a-1), (the thick line represents data of the image above and the thin line represents data of the image below), (b-1) compensated above/below the image, (b-2) red-component histogram of (b-1).

performing the block matching from a large to a small size, we can correctly find the corresponding pixels within an ignorable error range. The error range depends on the estimation block size, e.g., in the case of a 8×8 block, the block matching can cover about two or three pixel line errors [1].

D. Environment Generation—Implementation

With the NuView adapter on the Sony TRV900 camera, we capture a grid test pattern for 3-D calibration and red-green-blue (RGB) color patterns for color compensation. We then digitize the sequences using IEEE 1394 interface architecture. To maintain a stereo sync signal, the captured images are set to full size [720×480 at 30 frames per second (fps)]. The even lines contain the picture for the left eye and the odd lines for the right eye. As explained in Sections IV-A and IV-B, we first transform the images in the field-sequential format to the side-by-side format, and then interpolate them using (1) to recover the "full" size.

In our experiments, a noncoplanar test pattern was used to estimate camera parameters for the left (reference) and right (virtual, image coming through the adapter) cameras. The left and right cameras (with the calculated extrinsic parameters) were not positioned on the same plane as the reference (left) camera. According to our calibration results, the virtual (right) camera was set back at about 50 mm from the reference camera due to the mirror in the adapter.

After the camera calibration and image rectification, we performed color modification using (2). To analyze the characteristics of the color distortion, we captured three test images, each containing only one color component, i.e., red, green, and blue, respectively. Note that the statistics (mean and variance of each sequence) can be estimated during the camera-calibration process and applied on the fly. Fig. 4 compares the histograms of the original and the compensated pairs of stereo images for the red color. The effect of the color equalization is illustrated. The calibration and rectification improved the disparity estimation dramatically.

In addition to the procedures, as explained in Section IV-C, we also adopt a moving-object segmentation scheme to estimate depth information for moving objects along object boundaries [1]. As shown in Fig. 5(a) and (b), we first perform disparity estimation for a static scene and then for the scene with moving objects. By using the statistics of the static scene, we can separate moving objects from the scene (reasonably for scenes with few and small moving objects). The segmented moving objects with disparity can be combined back into the static-background scene. A more detailed description of this process is given in Section V. The resulting background video with depth information is ready to be used as a virtual backdrop.

V. User Video Generation

The "user/actor video" is generated using a multiview camera that can capture color and depth information simultaneously in real time. While the environment video was captured and reconstructed using the stereoscopic video camera (with adapter)

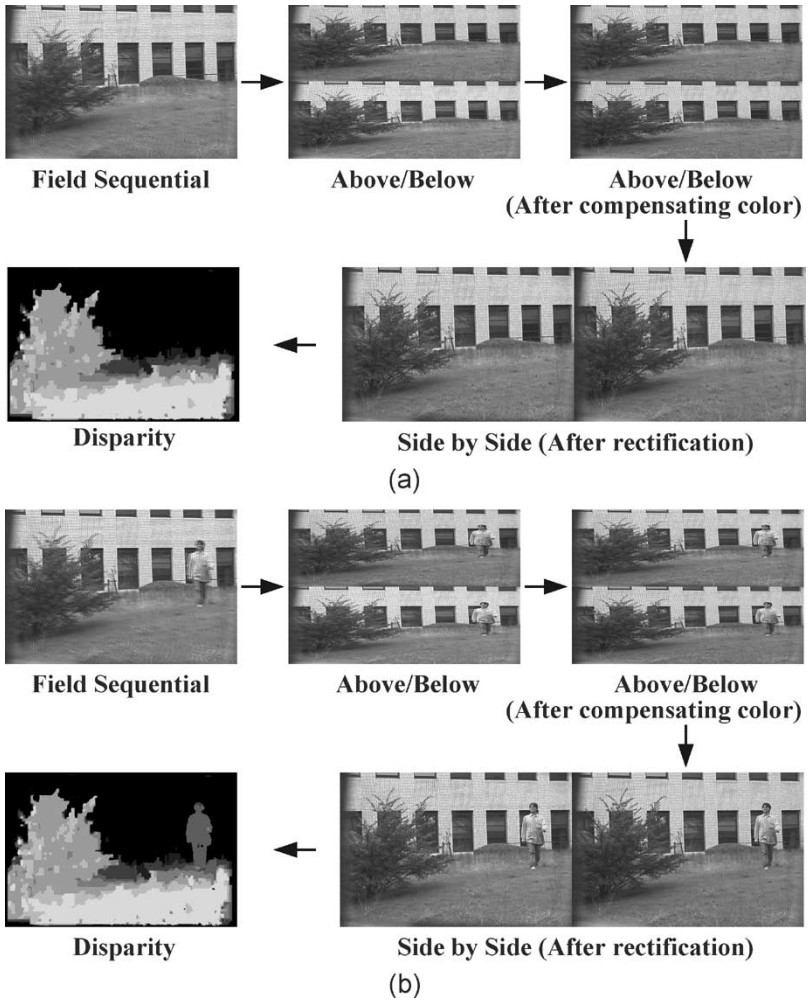

Fig. 5. Three-dimensional real image-based environment generation: (a) Static background, and (b) background with a moving object (a walking human in the right side).

offline, the user video must be captured on the fly for mixing and interacting with the 3-D graphic object. The multiview camera that we use is called the Digiclops[1] and provides color and depth information in real time (at about 12 fps in a resolution of 320×240).

There has been a growing interest in object segmentation for various applications with object-based functionalities such as interactivity. The capability of extracting moving objects from video sequences is a fundamental and crucial problem of many vision applications [8]. The difficulties in automatic segmentation mainly come from determining the semantically meaningful areas. Even in cases where we have a clear definition about the areas of interest, an accurate and reliable segmentation is still 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 schemes have been proposed and among these, the use of motion information has been widely accepted as an important cue, under the assumption that the objects of interest (OOI) exhibit coherent motion. However, methods based on motion similarity may not work because of occlusions and inaccurate motion estimation. Therefore, additional information is usually necessary to detect accurate boundaries of the moving objects.

In our framework, we separate objects from the static background by subtracting the current image from a reference image (i.e., a static background). The subtraction leaves only the nonstationary or new objects. The proposed segmentation algorithm consists of the following three steps: 1) static background modeling; 2) moving-object segmentation; and 3) shadow removal.

A. *Static-Background Modeling*

We present a robust and efficient algorithm to separate moving objects from the natural background by exploiting the color properties and statistics of the background image. 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 [8], [43]. However, many of these algorithms are susceptible to both global and local illumination changes that cause the consequent processes to fail.

[1]The NuView is used for generating the virtual backdrop due to its portability and simplicity, and the DigiClops [22] for user video for its real-time processing features.

In addition, the algorithms mostly focus on the approximated object regions shaped like 2-D rectangles or ellipses, not the precise objects's shape. Schreer *et al.* gave a solution to real-time shadow detection and elimination for extracting exact object shape in their video-conferencing system [44]. They used the hue and saturation approximation method in the hue-saturation-value (HSV) color space. Our approach is similar to theirs in the aspect of the object-detection criterion, but differs in the color spaces used.

Our proposed normalized color space is able to cope with reasonable changes in the illumination conditions. The proposed algorithm for detecting moving objects from a static background scene works fairly well on real image sequences from outdoor scenes, as shown in Fig. 5.

We first set up the multiview camera with known camera extrinsic parameters (pitch value and height from floor, which was measured physically) and internal parameters (horizontal and vertical field of view, and focal length, which are calculated from multiview camera specifications). These camera parameters (with the relative stereoscopic camera-position information obtained in the previous camera-calibration stage) are used for rendering and mixing the computer-graphic objects with the already constructed virtual backdrop.

We then gather the 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

$$I_m(R,G,B) = \frac{1}{L} \cdot \sum_{t=0}^{L-1} I_t(R,G,B) \tag{3}$$

$$I_\sigma(R,G,B) = \sqrt{\frac{1}{L} \cdot \sum_{t=0}^{L-1} \left(I_t(R,G,B) - I_m(R,G,B)\right)^2} \tag{4}$$

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

B. Moving-Object Segmentation

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 of the moving objects from pixels of the background, we measure the Euclidian distance in the RGB color space, and then compare it with the threshold. The threshold is adjusted in order to obtain a desired detection rate in the subtraction operation. The distance D_t and threshold Th are defined as follows

$$D_t(R,G,B) = \sqrt{\sum_{s=R,G,B} \left(I_t(S) - I_m(S)\right)^2} \tag{5}$$

$$\mathrm{Th}(R,G,B) = \alpha \cdot \left(I_\sigma(R) + I_\sigma(G) + I_\sigma(B)\right) \tag{6}$$

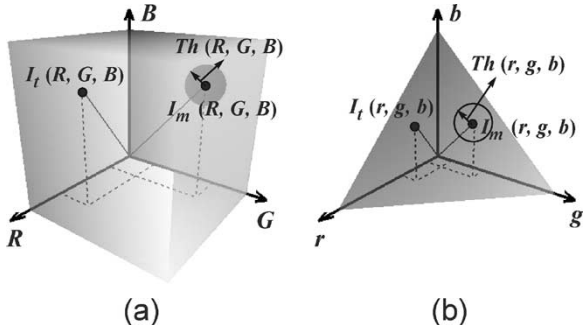

Fig. 6. Classification in the color space: (a) Classification in RGB space, and (b) classification in the normalized RGB space.

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

C. Shadow Removing

Normally, moving objects make shadows and shading effects according to (changing) lighting conditions. However, the RGB color space is not a proper space to deal with the (black and white) shadow and the shading effects.

Instead, as shown in Fig. 6, we use the normalized color space, which represents the luminance property better [25], [43]. For classification in the normalized RGB space, the normalized mean and color images, respectively, are defined as follows

$$I_m(r,g,b) = \frac{I_m(R,G,B)}{S_m(R,G,B)} \tag{7}$$

$$I_t(r,g,b) = \frac{I_t(R,G,B)}{S_t(R,G,B)} \tag{8}$$

where S_m and S_t, respectively, denote the summation of the (R, G, B) component of the mean reference and tth frame images, i.e.

$$S_m(R,G,B) = I_m(R) + I_m(G) + I_m(B) \tag{9}$$

$$S_t(R,G,B) = I_t(R) + I_t(G) + I_t(B). \tag{10}$$

The distance D_t and threshold for the normalized color space Th are defined as follows

$$D_t(r,g,b) = \sqrt{\sum_{s=(r,g,b)} \left(I_t(s) - I_m(s)\right)^2} \tag{11}$$

$$\mathrm{Th}(r,g,b) = \alpha \cdot \left(I_\sigma(r) + I_\sigma(g) + I_\sigma(b)\right) \tag{12}$$

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

The current image $I_t(R, G, B)$ is first compared pixelwise with the mean reference image $I_m(R, G, B)$ and classified as the background, if the difference is less than the threshold values $\mathrm{Th}(R, G, B)$. Meanwhile, the objects can further be classified into shadows or shadings in the normalized 2-D color space.

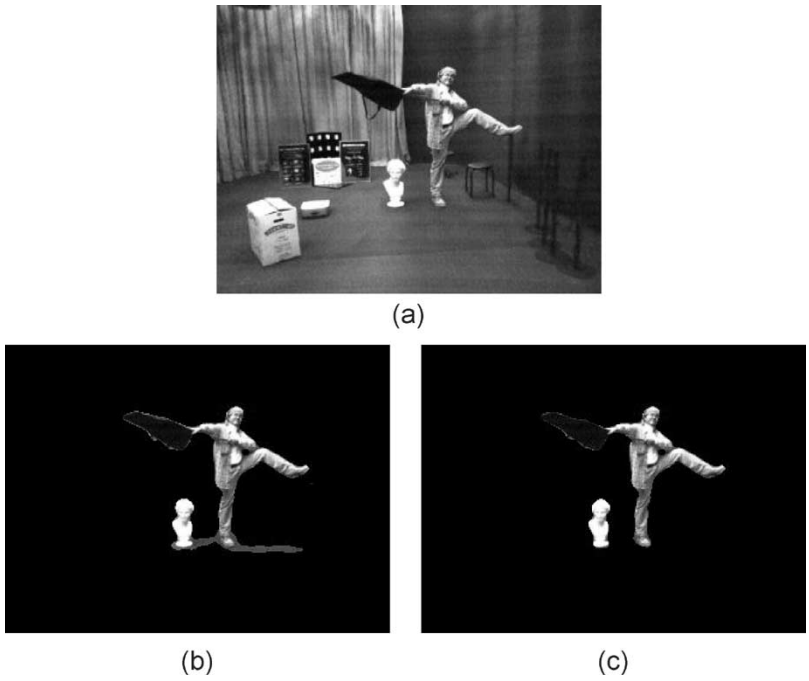

Fig. 7. Moving-object segmentation: (a) Current image, (b) initial object segmentation, and (c) segmented object after shadow removal.

In the normalized color space, we can separate the shadows from the 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 Fig. 6(b), if the difference of the pixel is less than the threshold, the pixel is classified as part of the background. Note that we can speed up the process by only checking the difference in the normalized color space, if the pixel was classified as part of the objects. If the pixel is classified as part of the objects (in the normalized color space), then we further decompose the color difference into the brightness and chromaticity components. Then, based on the observation that a 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 shadows.

We also exploit the depth information to reduce misclassification, while segmenting out the moving object. The underlying assumption is that the moving objects have a limited range of depth values (i.e., relatively small or thin). Using this assumption, we remove spot noises in the segmented object. In addition, we apply two types of median filters (5×5 and 3×3) to fill the hole while maintaining sharp boundaries.

Fig. 7 shows the segmented user from the natural-background scene, without using blue-screen techniques. As shown in Fig. 7(c), we have segmented the object with sharp boundaries, after applying the proposed separation scheme in the normalized color spaces.

Note that in the chromakeying-based virtual studios, real "nonblue" props cannot be inserted (if the purpose is to extract just the human actor). In our static background-based approach, we can insert and remove any colored objects (the inserted objects would be part of the static background). See Section IX for more explanation and examples.

VI. Graphic-Object Rendering

The scenes made using a virtual studio usually introduce computer-generated synthetic objects, which may be interacted upon and exhibit behaviors. In general, we can add three types of computer-graphic objects into the environment: 1) isolated objects that have no interaction with the user or other objects in the environment; 2) objects that interact with one another; and 3) objects that interact with the users. The environment evolves according to the interaction events.

In mixing computer-graphic objects into the VE, one of the most important technical issues is the registration between computer-graphic objects and background scenes. To provide a natural-looking environment, the exact position of the real camera must be known to place the objects correctly in the environment.

We exploit the camera parameters, which are obtained in the camera-calibration stage while generating the virtual space, as explained in Sections IV and V. Another important issue to achieve a more natural composition is taking the shadow and shading effects into account, according to the changing lighting conditions. In our approach, we assume that the lighting conditions are not significantly changing and the directions of light sources are known. These assumptions work fairly well because we use outdoor scenes as background video and, according to the given lighting conditions, we can control the shading of the graphic objects appropriately.[2]

Our framework has two methods for detecting possible interaction between the graphic objects and users. One is based on an inclusion test that tests whether part of the users' 3-D

[2]The graphic objects are rendered in the Open Graphics Library (OpenGL) environment [45]. In OpenGL, the color image and the corresponding depth image are obtained from the front-color and depth buffer.

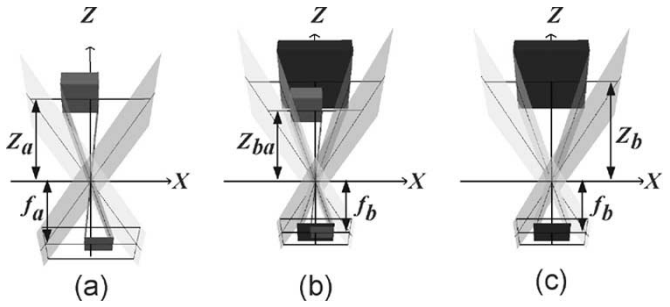

Fig. 8. z-keying maintaining size consistency: (a) User video, (b) background video, and (c) composite video.

points are within another objects' boundaries. This method is very sensitive to the computed 3-D position error acquired from the multiview camera, and thus, we use the portional number, that is, the ratio of one's points included in the other to the total number of one's points (and vice versa). According to our experiments, the inclusion portional ratio was tuned to about 0.3%. The other method is based on the common geometry-based collision detection. To realize this method, we have to reconstruct the 3-D geometry from the captured 3-D-point data, and such a reconstruction is computationally costly to be done at every frame. In our framework, all interactable objects are organized as a hierarchy of (invisible) bounding spheres or boxes. When a sufficient number of users' 3-D points are included in a bounding sphere of another object, then only those included points are reconstructed into triangle patches. Then, a more exact second test can be carried out between the triangle patches and part of the object contained in the intersected bounding sphere.

VII. Mixing (z-Keying)

To compose the final output, the z-keying operation is carried out between the video avatar and the background. To maintain the consistency between the real space in which the user operates and the virtual space of the graphic objects, we need to use a unified set of camera parameters. Thus, the parameters of the camera with the adapter (i.e., NuView) are used as the reference. We first use the precalculated camera parameters to set up a multiview camera. Then, we compensate for the position and rotation errors of the multiview cameras by finding a principal axis in 3-D from the disparity information of the segmented user.

The camera position and rotation information can be estimated based on an assumption that we can find a line passing through the user perpendicular to the floor. To preserve the original 3-D geometry of the composed image, the user video has to be enlarged or reduced in proportion to the focal distance.

Let the user video and background video be a and b, respectively. As shown in Fig. 8, the depth between the camera and the object in the composed image Z_{ba} can be approximated as follows [24]

$$Z_{ba} = Z_a \times \frac{f_b}{f_a} \tag{13}$$

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 user video in the composed image remains unchanged.

Fig. 9 shows the z-keying process and the final result, respectively. Upon interaction, the appropriate parts (instead of the whole environment) of the scene are updated.

We compare the depth values of the environment, segmented actor, and graphic objects. The composite scene chooses the color pixel corresponding to the highest depth value (i.e., closest). If there is a conflict (e.g., same depth value from two different entities), priority is given in the following order: actor, graphic objects, and the environment.

VIII. Support for Natural Interaction

In a typical virtual-studio setting, the user needs to see oneself through a monitor to interact with the synthetic objects. This often creates unnatural looks and behavior, because the location of the display monitor (where the actor is looking at) cannot always coincide with that of the interaction object. Sometimes, the actor has to rely solely on prechoreographed motion or producer's moment-by-moment verbal instructions and act without any visual feedback at all.

One popular solution to the above problem is to use "blue" props. The actor would interact naturally with the blue prop, a "physical" representation of the interaction object. In the final production, the blue prop is replaced with the appropriate graphic object through simple chromakeying. However, this is only a partial solution because not every interaction object can conveniently be represented with blue objects (e.g., water, flying objects, etc.) In addition, there may be too much geometrical differences between the blue prop and the actual interaction object, again resulting in unnatural-looking interaction (e.g., noticeable seams between the hand and the object).

Therefore, in our proposed system, we solve the view-direction problem by including display devices that can be erased later from the final production by the static-background segregation technique covered in Section V. The display device may even be stereoscopic. For nonautostereoscopic display devices, the user would have to wear a glass of some sort. Image-processing techniques proposed by Wu *et al.* [48] can be used to artificially remove glasses from human faces in video streams. In order to supply tactility and improve natural-object manipulation, we adopt a whole-body vibrotactile display device that can be worn on strategic places on the body in a discreet manner. We use a vibrotactile display wear called the POStech Tactile Wear (POS-T Wear), a computer-controlled set of small vibration motors, custom developed in our laboratory. The vibration device is small enough to be hidden by the camera (e.g., can be put on the palm of the hand or under the shirt; see Fig. 18). The vibration upon collision helps the user's perception of collision (occurrence and position of collision) and results in more natural acting and event responses, especially for segments with timed interaction (e.g., hitting a flying tennis ball). The examples in the next section illustrates these features.

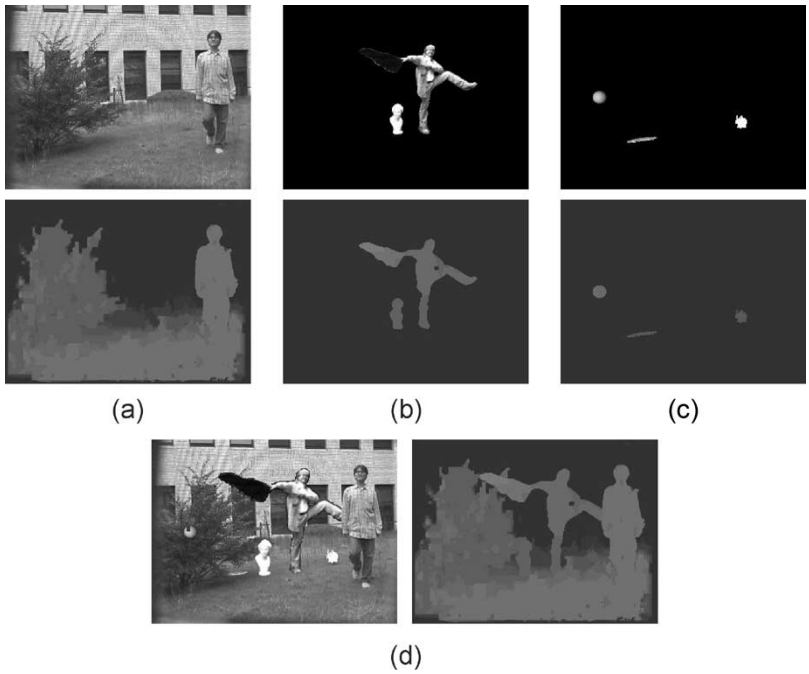

Fig. 9. Mixing result: (a) Environment scene, (b) segmented-user video, (c) graphic-object render scene, and (d) composed-result scene.

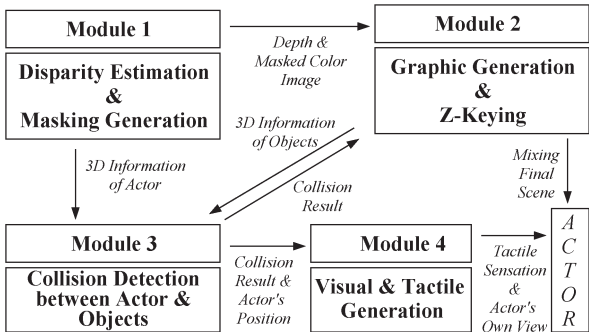

Fig. 10. System-implementation details. The system consists of four internal modules. This figure shows data flows among each module. Practically, we use three computers: One includes module 1 and 3, the other includes module 2, and another includes module 4.

Our system consists of four internal processing modules. Fig. 10 shows the important modules of the overall system. There are lots of data traffic among these modules, with the most delay occurring between module 1 and 2 due to the mask and color-image transmissions. To alleviate this networking delay, we implemented a JPEG compression algorithm with the Intel Performance Primitives library [47]. Module 3 computes collision detection and distributes the results to module 4 (for actor's visual and tactile feedback) and module 2 for the final mixing of the scene. Our system operates at about 16 ft/s using desktop personal computers (PCs). The overall system can match broadcast quality in terms of speed and image quality with better computing hardware.

IX. Applications

In this section, we illustrate five examples of applications of the proposed approach to a virtual studio for enhanced interaction capability. The example in Fig. 11 illustrates a 3-D outdoor scene (including the gazebo) mixed and rendered with a 3-D actor and graphic objects (a sphere, a rabbit, a lamp, and a lizard). The actor is able to "go behind" the columns of the gazebo "naturally" in real time according to the reconstructed 3-D information, not through postprocessing or prewritten scenarios.

Fig. 12 shows an example with a more complex user interaction with the environment (or graphic object) by the detection of collision with the 3-D actor's hand. In this case, the sphere is circling around the actor, and by touching the ball, the circling direction can be changed. The sphere moves counterclockwise around the actor in (1) and (2), then a collision with the user's hand is detected in (3), and its circling direction is changed in (4). Now, it circles in a clockwise direction as seen in (5)–(9), then upon another collision in (10), changes its direction again. Such interaction (or actually pretending to interact) with the synthetic objects occur frequently in a virtual-studio setting, for instance, the user invoking to bring up graphic data or animation to continue the given sketch. By allowing the user to naturally interact with synthetic objects, the acting can come more naturally, and also the scripts can be specified simpler without specification of the detailed motion and timing.

The third example in Figs. 13 and 14 shows the user interacting based on his location with respect to the environment, a perspective map in this case. In a typical virtual studio, an actor acts and "pretends" to interact with the environment according to the signals from the director. The transition from one scene to another occurs either by the actor pressing a hidden button in his hand or by the behind-the-curtain operator. We often notice that even trained professionals do make mistakes once in a while in live broadcast situations. In this example, the actor (or the weather forecaster) is able to see his location relative to the perspective map and by stepping to the appropriate region, he can invoke the pop-up of the temperature or other related

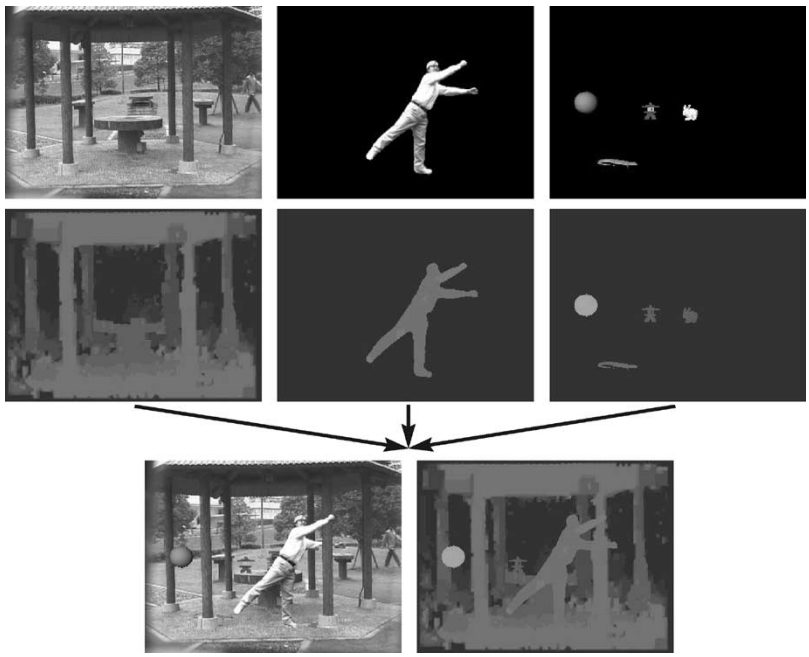

Fig. 11. Example: An outdoor scene with natural occlusion effect.

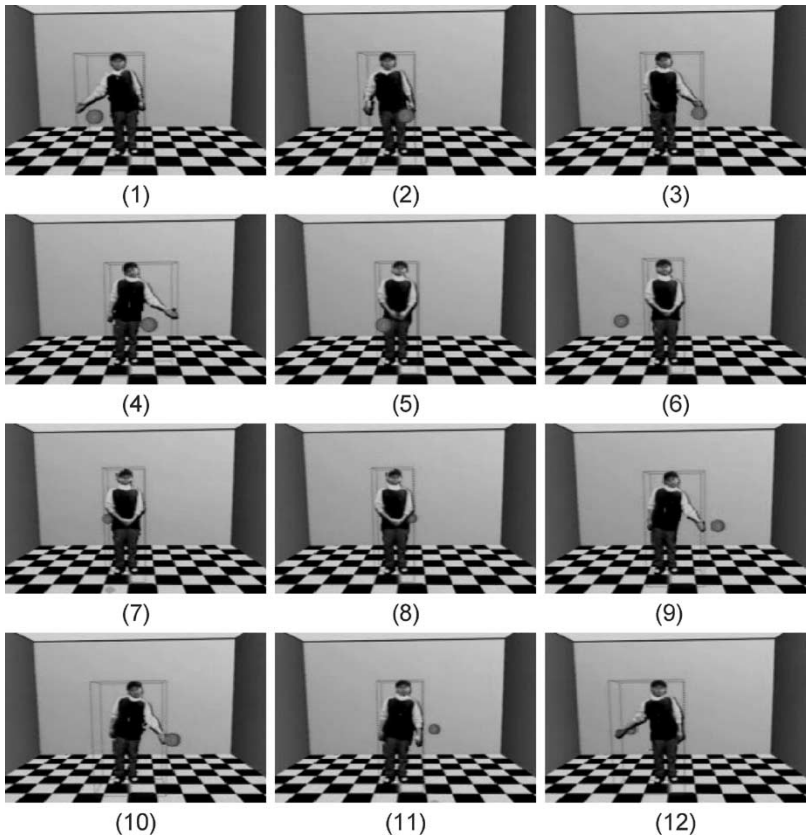

Fig. 12. Collision-based inter-"acting" virtual studio (image sequences). The sphere moves counterclockwise around the actor in (1) and (2), then collision with the user's hand is detected in (3), and its circling direction is changed in (4). Now, it circles in a clockwise direction as seen in (5)–(9), then upon another collision in (10), changes its direction again.

information pertaining to that region very naturally without any help from an operator or extra device.

The fourth example in Figs. 15–17 shows user interaction with real props. As mentioned in Section V-C, a general chromakeying-based virtual-studio set cannot include nonblue-color props. The upper-left image of Fig. 15 shows the static background model that includes the real props, the chair, and the monitor. Our system extracts only the actor without the real props. Fig. 16 shows the synthetic-background scene (and the composed one)—the actor sits on the virtual chair

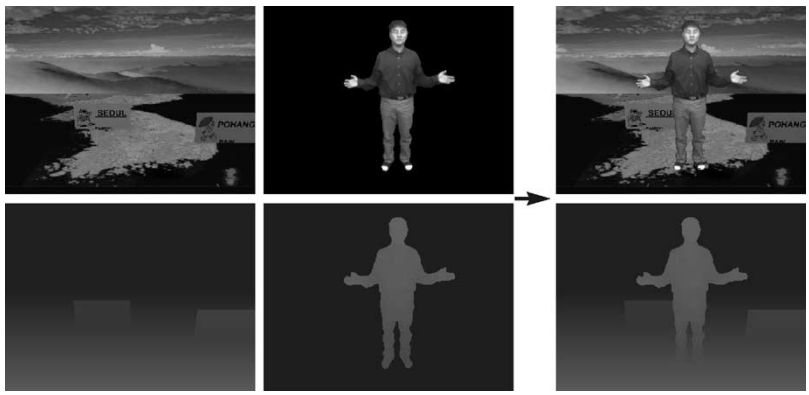

Fig. 13. Location-based inter-"acting" virtual studio: Weather-forecasting application.

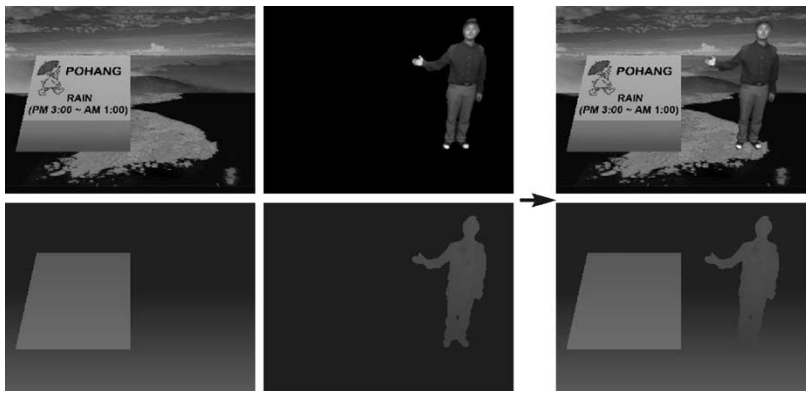

Fig. 14. Location-based inter-"acting" virtual studio: Weather-forecasting scenario (actor detected to be in a particular region on the map).

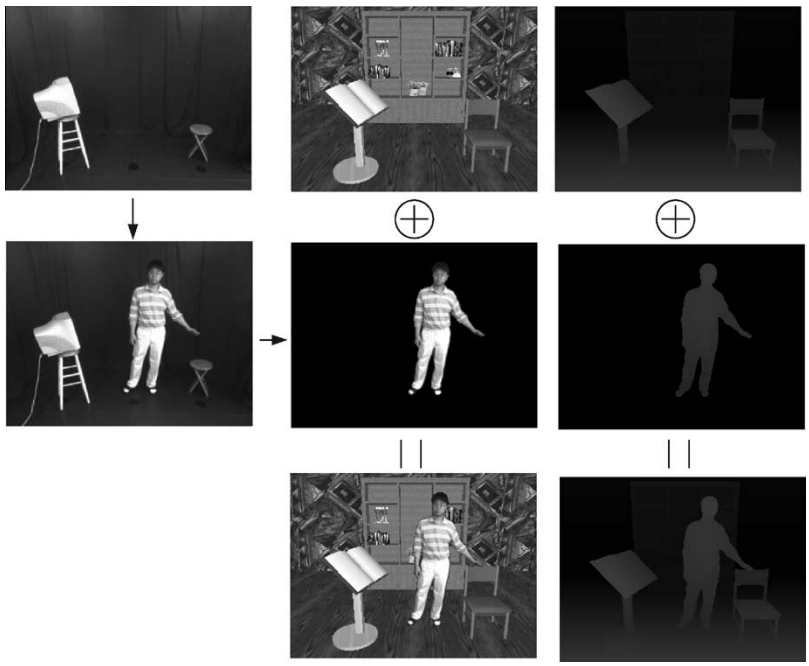

Fig. 15. Actor segmentation with real props. The upper-left image is the synthetic background. The segmented actor image is composited with the background (center column is the color images and the right column the depth images).

looking at the virtual book. In the real-studio set, the actor sits on the real chair looking at the monitor. Fig. 17 illustrates a situation of the actor manipulating the virtual book. The actor can act in a very natural way, looking at the changing contents of the monitor and even interacting with it. The interactable objects (such as the virtual book) have invisible hierarchical bounding volumes. If, for instance, the actor's 3-D points are included in the book's bounding volume, the

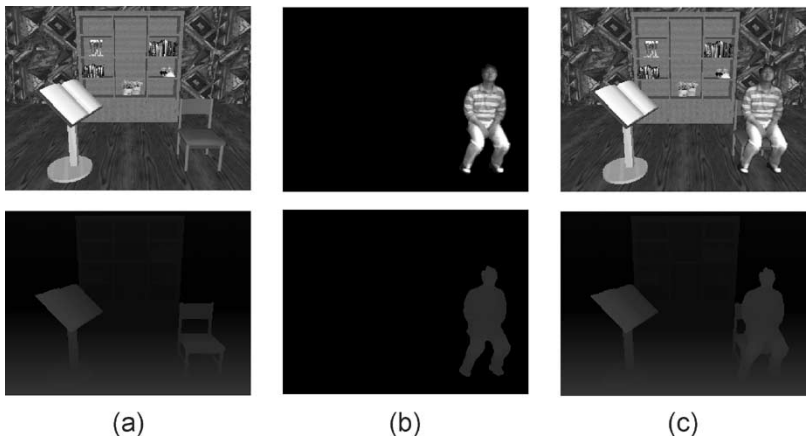

Fig. 16. Sitting on the prop chair: (a) Rendered virtual set, (b) actor sitting on the real chair, and (c) the composited virtual scene.

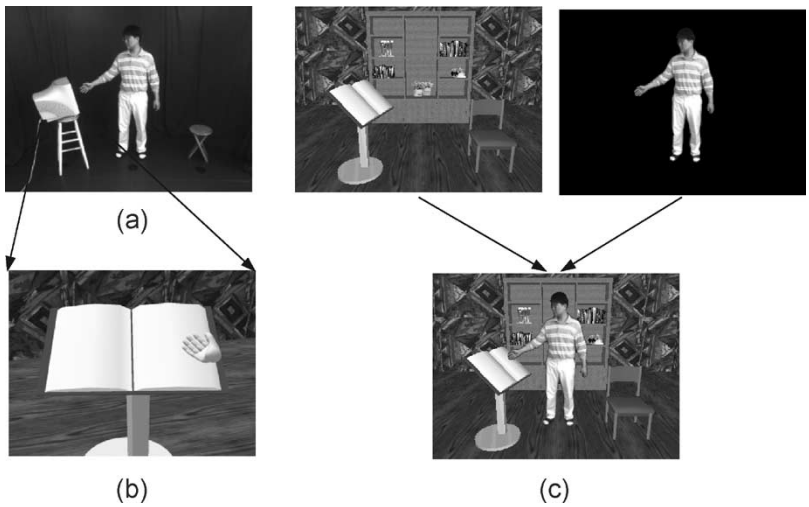

Fig. 17. Visual feedback for actor's interaction: (a) Interaction in the virtual studio, (b) what is seen in the display monitor to the actor, and (c) the composited final scene.

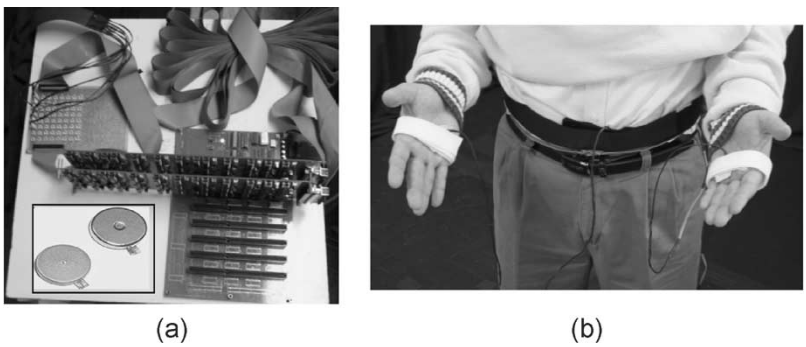

Fig. 18. POS-T Wear tactile-feedback hardware: (a) Vibration motors and the control hardware and (b) actor wearing the device on two hands and on the body.

center point of included points are calculated and a virtual hand can be drawn around that position as an interaction feedback [see Fig. 17(b)].

The fifth example in Figs. 18 and 19 illustrate the interaction using tactile feedback (see a movie clip [52]). Fig. 18 shows the POS-T Wear vibrotactile display device. The vibratory motor is shaped like a flat coin with a radius of about 7 mm, and a thickness of about 3.5 mm. It has a voltage range of 2.5–3.8 V and can produce between 8500 and 15 000 rpm. For this example, we use three vibrators on the two hands and on the body [see Fig. 18(b)]. Fig. 19(a)–(c) show the initial VEs. In Fig. 19(d), user is pushing a 3-D button with his right hand. The color of the button changes from blue to red upon detecting collision by the hand. A vibration is given to the user's hand at the time of the collision. Compared to reacting to the event only by visual feedback, the subtle addition of the tactility results in a much more natural (looking), smooth, and confident acting. After the button push, an object is floated in the air and the user

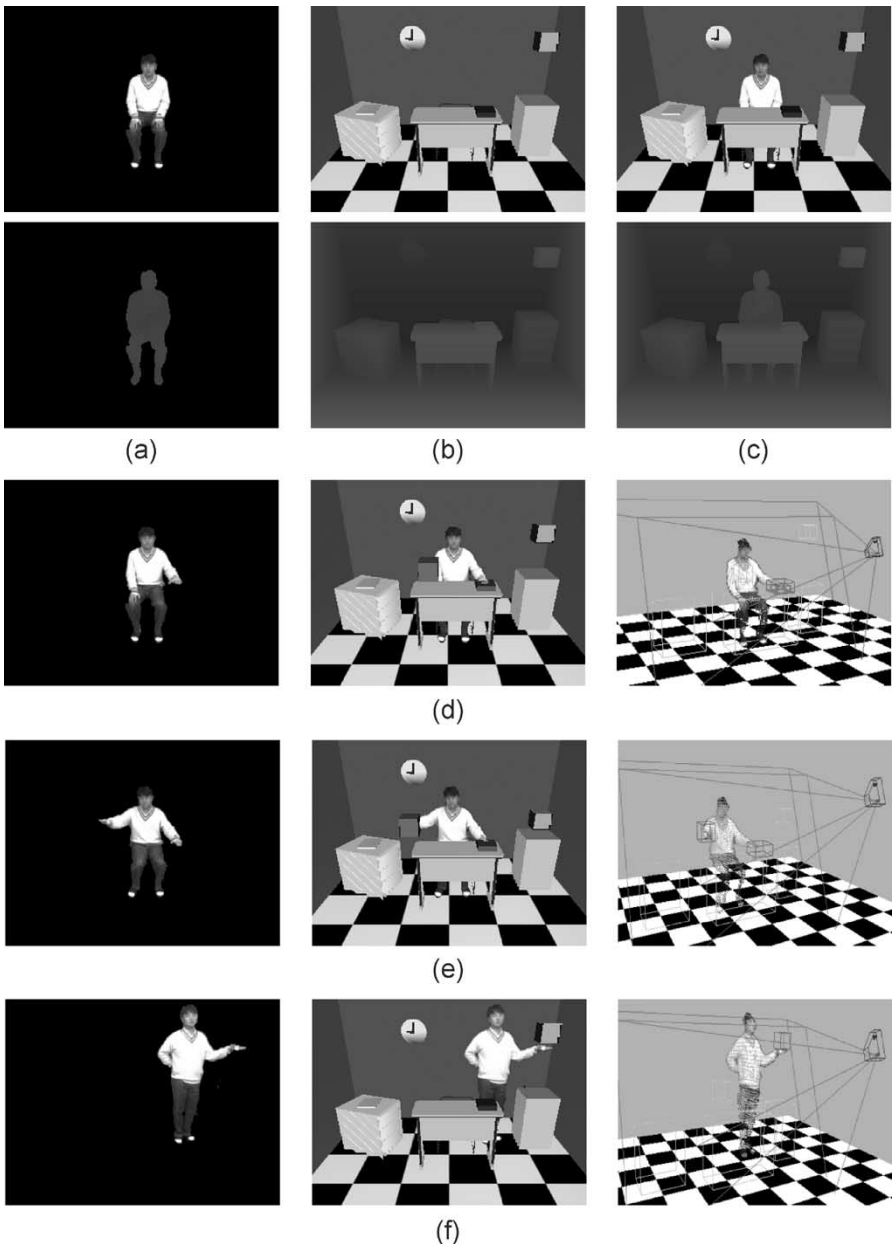

Fig. 19. Tactile feedback for actor's interaction: (a) Actor image, (b) graphic image, (c) mixed image, (d) user interaction (3-D-button push), (e) direct interaction with virtual objects, and (f) collision-based interaction.

can push the object up and down using the touch feedback on his hands [see Fig. 19(d)–(f)].

X. Conclusion and Future Works

Virtual studios have long been used in commercial broadcasting and are starting to evolve into the next level with improved technologies in image processing and computer graphics. Movies in which real life actors "seem" to act and interact with the synthetic characters are very common now. In this paper, we introduced a relatively inexpensive virtual-studio approach [personal-computer (PC)-based and inexpensive specialty cameras] framework to enable actors/users to interact in three-dimensional (3-D) systems more naturally with the synthetic environment and objects. The proposed system uses a stereo camera to first construct a 3-D environment (for the actor to act in), a multiview camera to extract the image and 3-D information about the actor, and a real-time registration and rendering software for generating the final output. Synthetic 3-D objects can be easily inserted and rendered, in real time, together with the 3-D environment and video actor for natural 3-D interaction.

The enabling of natural 3-D interaction would make more cinematic techniques possible including live and spontaneous acting. For the actor to act naturally and interact with the virtual environment (VE), the actor must be able to see the composited output in real time, preferably in the first-person viewpoint. Providing a first-person viewpoint is problematic, because the actor would have to wear a special display device [like a head mounted display (HMD)] and this does not go well with the

broadcasting purpose. The next best alternative is to provide a large display near or where the user is looking at (e.g., camera or interaction object) and removing it (if necessary) from the final composited scene, as was demonstrated in our system.

We acknowledge that the proposed system uses many of the already known techniques in the fields of computer vision, graphics, and VR, such as camera calibration, collision detection, and 3-D reconstruction. However, several practical engineering details and issues that go along when building a working system like this were well addressed, including color compensation, shadow removal, "seemingly" believable collision handling, image formatting, image distortion, etc.

The proposed system is not limited to broadcast production, but can also be used for creating virtual/augmented-reality environments for training and entertainment. In fact, even home-brewed digital contents can be produced for presentations or short educational materials due to its low cost.

We are currently working to further improve our system to be used for actual broadcasting. For instance, the image quality as proposed in this paper is not sufficient for general broadcasting quality (about 640×480 resolution). To achieve higher quality, the actor segmentation and 3-D point-based interaction must be computed in real time for images with 640×480 resolution. The alternative solution may be to use two cameras, one of high quality for shooting the actor, and the other, multiview, for computing the 3-D information of the actor, and fuse the two types of information in a seamless manner.

The vibrotactile device is currently not wireless, making it difficult to use, plus, it has latency problems (slow response time). We hope to improve the quality of the hardware in these regards. Another weakness of the proposed system is that it does not allow any camera movement. Furthermore, for a more natural interaction and acting, it is necessary that the whole body of the actor be tracked (or at least parts other than just the end of the limbs).

REFERENCES

[1] W. Woo, N. Kim, and Y. Iwadate, "Stereo imaging using a camera with stereoscopic adapter," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Nashville, TN, Oct. 2000, pp. 1512–1517.

[2] W. Woo and Y. Iwadate, "Object-oriented hybrid segmentation using stereo image," in *Proc. Society Optical Engineering (SPIE) PW-EI-Image and Video Communications and Processing (IVCP)*, Bellingham, WA, Jan. 2000, pp. 487–495.

[3] W. Woo, N. Kim, and Y. Iwadate, "Object segmentation for Z-keying using stereo images," in *Proc. IEEE Int. Conf. Signal Processing Proceedings (ICSP)*, Beijing, China, Aug. 2000, pp. 1249–1254.

[4] N. Kim, W. Woo, and M. Tadenuma, "Photo-realistic interactive 3D virtual environment generation using multiview video," in *Proc. Society Optical Engineering (SPIE) PW-EI-Image and Video Communications and Processing (IVCP)*, Bellingham, WA, Jan. 2001, vol. 4310, pp. 245–254.

[5] S. Gibbs, C. Arapis, C. Breiteneder, V. Lalioti, S. Mostafawy, and J. Speier, "Virtual studios: An overview," *IEEE Multimedia*, vol. 5, no. 1, pp. 18–35, Jan.–Mar. 1998.

[6] A. Wojdala, "Challenges of virtual set technology," *IEEE Multimedia*, vol. 5, no. 1, pp. 50–57, Jan.–Mar. 1998.

[7] Y. Yamanouchi, H. Mitsumine, T. Fukaya, M. Kawakita, N. Yagi, and S. Inoue, "Real space-based virtual studio—Seamless synthesis of a real set image with a virtual set image," in *Proc. ACM Symp. Virtual Reality Software and Technology (VRST)*, Hong Kong, Nov. 2002, pp. 194–200.

[8] D. M. Gavrila, "The visual analysis of human movement: A survey," *Comput. Vis. Image Understand.*, vol. 73, no. 1, pp. 82–98, 1999.

[9] K. M. Cheung, "Visual hull construction, alignment and refinement for human kinematic modeling, motion tracking and rendering," Ph.D. dissertation, Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-44, Oct. 2003.

[10] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[11] P. Maes, T. Darrell, B. Blumberg, and A. Pentland, "The ALIVE system: Wireless, full-body interaction with autonomous agents," *Multimedia Syst.*, vol. 5, no. 2, pp. 105–112, 1997.

[12] T. Kanade, K. Oda, A. Yoshida, M. Tanaka, and H. Kano, "Video-rate Z keying: A new method for merging images," The Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-95-38, Dec. 1995.

[13] T. Kanade, *Virtualized Reality*. Pittsburgh, PA: Robotics Inst., Carnegie Mellon Univ. [Online]. Available: http://www.ri.cmu.edu/labs/lab_62.html

[14] L. Blonde, M. Buck, R. Calli, W. Niem, Y. Paker, W. Schmidt, and G. Tomas, "A virtual studio for live broadcasting: The Mona Lisa project," *IEEE Multimedia*, vol. 3, no. 2, pp. 18–29, Summer 1996.

[15] C. Zhang and T. Chen, "A survey on image-based rendering—Representation, sampling and compression," *EURASIP Signal Processing: Image Communication*, vol. 19, no.1, pp. 1–28, 2004, Invited Paper.

[16] S. B. Kang, "A Survey of Image-Based Rendering Techniques," Digital Equipment Corp., Cambridge Res. Lab., Cambridge, MA, Tech. Rep. 97/4, Aug. 1997.

[17] S. B. Kang and P. K. Desikan, "Virtual navigation of complex scenes using clusters of cylindrical panoramic images," in *Proc. Graphics Interface*, Vancouver, BC, Canada, 1998, pp. 223–232.

[18] S. E. Chen, "QuickTimer VR: An image-based approach to virtual environment navigation," in *Proc. Computer Graphics, SIGGRAPH*, Los Angeles, CA, 1995, pp. 29–38.

[19] J. S. Pierce, A. Forsberg, M. J. Conway, S. Hong, R. Zeleznik, and M. R. Mine, "Image plane interaction techniques in 3D immersive environments," in *Proc. Symp. Interactive 3D Graphics*, Providence, RI, 1997, pp. 39–43.

[20] H. Shum, "Rendering by manifold hopping," in *Proc. SIGGRAPH*, Los Angeles, CA, 2001, p. 253.

[21] 3-D Video Inc., *NuView Owner's Manual*. [Online]. Available: http://www.stereo3d.com/nuview.htm

[22] Point Grey Research Inc., Digiclops and Triclops: Stereo Vision SDK. [Online]. Available: http://www.ptgrey.com

[23] R. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robot. Autom.*, vol. RA-3, no. 4, pp. 323–344, Aug. 1987.

[24] NHK STR Labs, *Virtual System for TV Program Production*. NHK Labs Note, No. 447.

[25] Y. Gong and M. Sakauchi, "Detection of regions matching specified chromatic features," *Comput. Vis. Image Understand.*, vol. 61, no. 2, pp. 263–269, 1995.

[26] J. Mulligan, N. Kelshikar, X. Zampoulis, and K. Daniilidis, "Stereo-based environment scanning for immersive telepresence," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 3, pp. 304–320, Mar. 2004.

[27] N. Kelshikar, X. Zabulis, J. Mulligan, K. Daniilidis, V. Sawant, S. Sinha, T. Sparks, S. Larsen, H. Towles, K. Mayer-Patel, H. Fuchs, J. Urbanic, K. Benninger, R. Reddy, and G. Huntoon, "Real-time terascale implementation of teleimmersion," in *Proc. Int. Conf. Computational Science (ICCS)*, Melbourne, Australia, 2003, pp. 33–42.

[28] O. Schreer and P. Kauff, "An immersive 3D video-conferencing system using shared virtual team user environments," in *Proc. ACM Collaborative Environments*. Bonn, Germany, 2002, pp. 105–112.

[29] F. Isgro, E. Trucco, P. Kauff, and O. Schreer, "3D image processing in the future of immersive media," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 3, pp. 288–303, Mar. 2004.

[30] H. Baker, D. Tanguay, I. Sobel, D. Gelb, M. Goss, W. Culbertson, and T. Malzbender, "The coliseum immersive teleconferencing system," HP Labs, Tech. Rep. HPL-2002-351, Dec. 2002.

[31] R. Yang, M. Pollefeys, H. Yang, and G. Welch, "A unified approach to real-time, multi-resolution, multi-baseline 2D view synthesis and 3D depth estimation using commodity graphics hardware," *Int. J. Image Graph.*, vol. 4, no. 4, pp. 1–25, 2004.

[32] J. Gluckman and S. K. Nayar, "Rectified catadioptric stereo sensors," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Hilton Head Island, SC, 2000, pp. 380–387.

[33] T. Brodsky, C. Fermuller, and Y. Aloimonos, "Structure from motion: Beyond the epipolar constraint," *Int. J. Comput. Vis.*, vol. 37, no. 3, pp. 231–258, 2000.

[34] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1, pp. 7–42, 2002.
[35] N. Apostoloff and A. Fitzgibbon, "Bayesian video matting using learnt image priors," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Washington, DC, 2004, pp. 407–414.
[36] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut—Interactive foreground extraction using iterated graph cuts," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 2004, pp. 309–314.
[37] M. Uyttendaele, A. Criminisi, S. Kang, S. Winder, R. Szeliski, and R. Hartley, "Image-based interactive exploration of real-world environments," *IEEE Comput. Graph. Appl.*, vol. 24, no. 3, pp. 52–63, May–Jun. 2003.
[38] R. Urtasun and P. Fua, "3D human body tracking using deterministic temporal motion models," Computer Vision Lab, Swiss Federal Inst. Tech., Lausanne, Switzerland, Tech. Rep. IC/2004/03, 2004.
[39] R. Plaenkers and P. Fua, "Articulated soft objects for multiview shape and motion capture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1182–1187, Sep. 2003.
[40] J. Starck, G. Collins, R. Smith, A. Hilton, and J. Illingworth, "Animated statues," *J. Mach. Vis. Appl.*, vol. 14, no. 4, pp. 248–259, 2003.
[41] W. Sun, A. Hilton, R. Smith, and J. Illingworth, "Layered animation of captured data," *Visual Comput., Int. J. Comput. Graph.*, vol. 17, no. 8, pp. 457–474, 2001.
[42] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2002, pp. 94–102.
[43] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," *Comput. Vis. Image Understand.*, vol. 80, no. 1, pp. 42–56, 2002.
[44] O. Schreer, I. Feldmann, U. Goelz, and P. Kauff, "Fast and robust shadow detection in videoconference applications," in *Proc. VIPromCom, 4th EURASIP IEEE Int. Symp. Video Processing and Multimedia Communications*, Zadar, Croatia, 2002, pp. 371–375.
[45] M. Woo, J. Neider, and T. Davis, *OpenGL Programming Guide*, 3rd ed. Reading, MA: Addison-Wesley.
[46] V. Alessandro, Notes on Natural Interaction. [Online]. Available: http://naturalinteraction.org/NotesOnNaturalInteraction.pdf
[47] Intel Com., Intel Performance Primitive Library. [Online]. Available: http://www.intel.com/software/products/ipp/index.htm
[48] C. Wu, C. Liu, H. Y. Shum, Y. Q. Xu, and Z. Zhang, "Automatic eyeglasses removal from face images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 3, pp. 322–336, Mar. 2004.
[49] C. Hand, "A survey of 3D interaction techniques," *Comput. Graph. Forum*, vol. 16, no. 5, pp. 269–281, Dec. 1997.
[50] D. Bowman, J. Gabbard, and D. Hix, "A survey of usability evaluation in virtual environments: Classification and comparison of methods," *Presence, Teleoperators Virtual Environ.*, vol. 11, no. 4, pp. 404–424, 2002.
[51] D. Bowman, E. Kruijff, J. LaViola, and I. Poupyrev, *3D User Interfaces: Theory and Practice*. Boston, MA: Addison-Wesley, 2004.
[52] N. Kim, A movie clip about the 3-D interaction using tactile feedback. [Online]. Available: http://home.postech.ac.kr/~ngkim/Studio/vsnatural.wmv

Namgyu Kim received the B.S. degree in computer science in 1995 from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, and earned the M.S. degree in 1998 in computer science and engineering from the Pohang University of Science and Technology (POSTECH), Pohang, Korea, where he is currently pursuing the Ph.D. degree.

He has an interest in the system and interaction design of a virtual/mixed-reality environment.

Woontack Woo (S'94–M'00) received the B.S. in electronics engineering from Kyungbuk National University, Daegu, Korea, in 1989, and the M.S. degree in electronics and electrical engineering from Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 1991. In 1998, he received the Ph.D. degree in electrical engineering systems from University of Southern California (USC), Los Angeles.

He joined Advanced Telecommunications Research (ATR), Kyoto, Japan, in 1999, as an Invited Researcher. Since February 2001, he has been with the Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, where he is an Assistant Professor in the Department of Information and Communications. The main thrust of his research has been implementing holistic smart environments, which include UbiComp/WearComp, virtual environment, human–computer interaction, three-dimensional (3-D) vision, 3-D visual communications, and intelligent signal processing.

Gerard J. Kim (S'92–M'05) received the B.S. in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1987, the M.S. degree in computer engineering in 1989, and the Ph.D. degree in computer science in 1994 from the University of Southern California (USC), Los Angeles.

He had worked as a Computer Scientist with support from the U.S. National Research Council Postdoctoral Fellowship for two years at the National Institute of Standards and Technology (NIST). He has been an Associate Professor in Computer Science and Engineering at the Pohang University of Science and Technology (POSTECH), Pohang, Korea, since 1995. His research interests include various topics in virtual reality (VR) (3-D interaction and innovative applications), computer music, and computer-aided design, such as intelligent design and design reuse.

Chan-Mo Park received the B.S. degree in chemical engineering from Seoul National University, Seoul, Korea, and the M.S. and Ph.D. degrees from University of Maryland, College Park. In 2001, he was awarded the Honorary Doctor of Letters degree at University of Maryland University College, Adelphi.

He has been a Professor of Computer Science at University of Maryland, College Park, Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, The Catholic University of America, Washington, DC, and Pohang University of Science and Technology (POSTECH), Pohang, Korea, since 1969. Currently, he is the President of POSTECH. His main research interests are digital image processing, computer graphics and computer vision, system simulations, and VR.