INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION, *20*(3), 271–284

A 3D Vision-Based Ambient User Interface

Dongpyo Hong
Woontack Woo
Ubiquitous Computing and Virtual Reality Lab, GIST

This article proposes a 3-dimensional (3D) vision-based ambient user interface as an interaction metaphor that exploits a user's personal space and its dynamic gestures. In human–computer interaction, to provide natural interactions with a system, a user interface should not be a bulky or complicated device. In this regard, the proposed ambient user interface utilizes an invisible personal space to remove cumbersome devices where the invisible personal space is virtually augmented through exploiting 3D vision techniques. For natural interactions with the user's dynamic gestures, the user of interest is extracted from the image sequences by the proposed user segmentation method. This method can retrieve 3D information from the segmented user image through 3D vision techniques and a multiview camera. With the retrieved 3D information of the user, a set of 3D boxes (SpaceSensor) can be constructed and augmented around the user; then the user can interact with the system by touching the augmented SpaceSensor. In the user's dynamic gesture tracking, the computational complexity of SpaceSensor is relatively lower than that of conventional 2-dimensional vision-based gesture tracking techniques, because the touched positions of SpaceSensor are tracked. According to the experimental results, the proposed ambient user interface can be applied to various systems that require real-time user's dynamic gestures for their interactions both in real and virtual environments.

1. INTRODUCTION

With the rapid progress of technologies in the computers and communications fields, future computing environments will provide users with a large volume of information and services. In such computing environments, users will be able to receive just-in-time services from any (invisible) computer, anytime and anywhere, due to ubiquitous computers and pervasive networking (Dey, Salber, & Abowd, 1999; Shafer et al., 2000; Weiser, 1991). Eventually, users will request more natural and comfortable user interfaces to interact seamlessly with computing environ-

This research was supported by CTRC at GIST.

Correspondence should be addressed to Woontack Woo, U-VR Lab, #C308, Dept. of Info. & Comm., GIST, Gwangju, 500–712, Korea. E-mail: wwoo@gist.ac.kr

ments. Through such interfaces, users can exploit their intentions or emotions when they want personalized services.

In spite of the great demand of natural and comfortable user interfaces, conventional user interfaces such as the keyboard and the mouse still remain very limited with regard to expressing the user's intentions or emotions. In virtual reality applications, for example, most systems still make participants use two-dimensional (2D) user interfaces even though the systems can support three-dimensional (3D) interactions. In this regard, new types of user interfaces have been introduced to overcome the limitations of 2D user interfaces in the last few decades. For example, eye-gaze tracking, gestures, voice, and even haptic user interfaces have drawn public attention because of the expectation of replacing conventional user interfaces with more natural and seamless ones (Freeman et al., 1998; Hayward, 2001; Maes, Darrell, Blumberg, & Pentland, 1995; Park, Lee, & Kim, 2002). However, these types of user interfaces are still complicated or bulky when the user is engaged in interacting with a system, because the user has to wear a data glove or motion tracking equipment (VRLOGIC Co., 2004). To compensate for these obstacles, many researchers have been exploiting vision techniques. The advantages of a vision-based user interface over other interfaces are its relatively easier calibration and more natural interactions with the systems by removing obtrusive devices.

In general, vision-based user interfaces are categorized into two types (Woo, Kim, Wong, & Tadenuma, 2001). One type is the contact vision-based user interface, which generally exploits markers worn by the user (Billinghurst & Kato, 1999). The other is the noncontact vision-based user interface, which generally exploits dynamic gesture tracking techniques (Ebihara et al., 1998; Haritaoglu et al., 1998). In the contact vision-based user interface, we can extract information of interest from tracking markers, which is simple and easy. However, there are several drawbacks in marker-based tracking techniques. When markers are occluded or when multiple markers are used, it is difficult to track them robustly and correctly. Furthermore, marker-based user interfaces require a user to wear markers or carry marker-attached objects. The noncontact vision-based user interface overcomes these limitations by removing distracting wires and markers. Although the noncontact vision-based user interface has many advantages, it is still vulnerable to various environmental conditions such as interference from lighting sources, casting shadows on the user, and so on. To resolve these restrictions, many vision algorithms have been developed, such as background subtractions, which can extract information robustly regardless of environmental changes (Elgammal, Duraiswami, Harwood, & Davis, 2002; Elgammal, Harwood, & Davis, 2000; Horprasert, Harwood, & Davis, 1999). Meanwhile, the previous vision-based dynamic gesture tracking techniques have been used in simple applications like hand-movement tracking, because they are relatively complicated and their computational complexity is comparatively higher to compute 3D information from 2D vision techniques (Freeman & Weissman, 1995; Kohler, 1997; Lenman, Bretzner, & Thuresson, 2000; Nishikawa, Ohnishi, & Miyazaki, 1998). Consequently, we believe that another technique is required to compensate complexity of dynamic gesture tracking for natural user interactions.

In this article, we propose a 3D vision-based ambient user interface as an interaction metaphor that exploits the user's personal space and their dynamic gestures. To exploit the gestures in interactions, we need 3D information of the user, which is extracted from a sequence of acquired images, and a gesture-tracking method. In retrieving 3D information of the user, we exploit the proposed user segmentation method, which takes advantages of RGB and normalized RGB color space (Hong & Woo, 2003). Because of the proposed user segmentation method, we do not have to use special facilities or devices to segment the user from the given background, like a blue screen or chroma-keying device. In addition, we can retrieve 3D information from the segmented user image because we utilize a multiview camera. In tracking the user's dynamic gestures, we construct a set of invisible 3D boxes and augment it around the user dynamically based on 3D information of the user. This set of 3D boxes is called SpaceSensor. Consequently, the user can interact with the system by touching the augmented SpaceSensor. Regarding the user's dynamic gesture tracking, the computational complexity of SpaceSensor is relatively lower than that of conventional 2D vision-based gesture tracking techniques, because we do not have to calculate 3D information additionally from the segmented images. In particular, we adopt the concept of "Ambient Media" in Ishii's "Tangible Bit" in the proposed ambient user interface and exploit air (or space) as the medium for the user to interact with a system (Ishii & Ullmer, 1997). Therefore, we expect that the proposed ambient user interface enhances naturalness while it removes the cumbersome devices.

This article is organized as follows. In section 2, we explain the key components of the proposed techniques (i.e., user segmentation algorithm, design of SpaceSensor and dynamic gesture tracking). Experimental results and some applications are shown in section 3. Discussion and future works follow in section 4.

2. SpaceSensor: 3D VISION-BASED AMBIENT USER INTERFACE

2.1. User Segmentation

The general background subtraction technique is to subtract a current image from the reference image. Although various cues (color, motion, block, etc.) are utilized in many studies, the proposed method exploits the characteristics of the pixel's color values in the well-known two color spaces (RGB and normalized RGB). In addition, we need to determine the optimal threshold values in the background subtraction techniques.

In the proposed method, we train the background images in RGB and normalized RGB color space, respectively. Then, we can evaluate the mean and standard deviation at pixel i's (R,G,B) color channels in the reference image during the background training. Each pixel of the reference image is modeled as follows:

$$Rf_i = < \mu_i, \sigma_i, \bar{\mu}_i, \bar{\sigma}_i > \quad I_i = \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} \quad \bar{I}_i = \begin{bmatrix} r_i \\ g_i \\ b_i \end{bmatrix} = \frac{1}{|I|}\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} \tag{1}$$

where Rfi is the tuple of reference image. μ_i and σ_i are the vector of the mean and standard deviation of pixel i's color channels in RGB color space. μ_i and σ_i are the vector of the mean and standard deviation of pixel i's color channels in the normalized RGB color space. I_i and I_i is the intensity of each pixel in RGB color and normalized RGB color space, respectively.

The following equations show how to compute the vector of the mean and standard deviation at pixel i in RGB and normalized RGB color space.

$$\mu_i = \frac{1}{N}\sum_{j=0}^{N-1} I_{ij} \qquad \bar{\mu}_i = \frac{1}{N}\sum_{j=0}^{N-1} \bar{I}_{ij} \tag{2}$$

$$\sigma_i = \frac{1}{\sqrt{N}}\left(\sum_{j=0}^{N-1}\left(I_{ij} - \mu_i\right)^2\right)^{\frac{1}{2}} \qquad \bar{\sigma}_i = \frac{1}{\sqrt{N}}\left(\sum_{j=0}^{N-1}\left(I_{ij} - \bar{\mu}_i\right)^2\right)^{\frac{1}{2}} \tag{3}$$

where N is the number of trained images.

When we observe the variations of pixels in the image of a static background scene, they are easily modeled as a Gaussian distribution. From this observation, the threshold value of pixel *i* is mapped by function of standard deviation of pixel *i*.

$$Th_i = \alpha \bullet \sigma_i \qquad \overline{Th}_i = \beta \bullet \bar{\sigma}_i \tag{4}$$

where Th_i and $\overline{Th}_i$ is threshold values of pixel i in RGB and normalized RGB color space, respectively. The determinant constants α and β determine the confidence interval.

In the proposed method, we show how to effectively use the determined threshold values to subtract the user from a background scene. Equations 5 and 6 are the determinant functions that compare the color channels differences of pixel i and the determined threshold values in RGB and normalized RGB color space, respectively.

$$F_i = \sum_{c=1}^{3} u\left(\left\|D_{i,c}\right\| - \left\|Th_{i,c}\right\|\right) \tag{5}$$

$$f_i = \sum_{c=1}^{3} u\left(\left\|\bar{D}_{i,c}\right\| - \left\|\overline{Th}_{i,c}\right\|\right) \tag{6}$$

$$D_i = I_i - \mu_i \qquad \bar{D}_i = \bar{I}_i - \bar{\mu}_i \tag{7}$$

where . $\|\bullet\| \equiv \sqrt{\left(x_1 - x_0\right)^2 + \left(y_1 - y_0\right)^2 + \left(z_1 - z_0\right)^2}$. $F_i(0 \le F_i \le 3)$ and $f_i(0 \le f_i \le 3)$ are the determinant functions that characterize pixel i in each color space and c is the number of channels. Here, u is a unit step function, and it has either 0 or 1. Each of D_i and $\bar{D}_i$ is the vector difference between current image and reference image at pixel i in RGB color space and normalized RGB color space, respectively. Thus, if $D_i > Th_i$, then it is 1. Otherwise, it is 0.

Using Equations 5 and 6, we can determine pixel *i* as follows.

$$Obj_i = \begin{cases} B: & F_i = c_i \\ H^s: 0 \le F_i < c_i \\ B^s: & f_i = c_2 \\ H: 0 \le f_i < c_2 \end{cases} \tag{8}$$

where B is the background image and B^s is the background image with cast shadows. H^s is the segmented user image with shadows, and H is the segmented user image without shadows. c_1 and c_2 are the numbers of color channels. In RGB and normalized RGB color space, its range is $0 \le c_1 \le 3$ and $0 \le c_2 \le 3$, respectively.

As shown in Figure 1, each stage has two steps in the proposed method. In the first stage, we train background images and make the reference image in RGB and normalized RGB color space, respectively. Then in the second stage, we subtract the current image from the reference image in each color space. In training background scenes, we model background using Equation 1 and determine the threshold at pixel i through Equation 4. After background modeling is done in each color space, we separate the user with the cast shadows from the background scene in RGB color space using Equation 5. After quantizing the image as a binary map, it can be used as mask image in normalized RGB color space. When we apply the mask image into the reference image and current image in normalized RGB color space at the same time, we simply discard the cast shadows from the user, because shadows have only effects on luminance. Through these two stages, we can easily achieve the user image (H) without cast shadows.

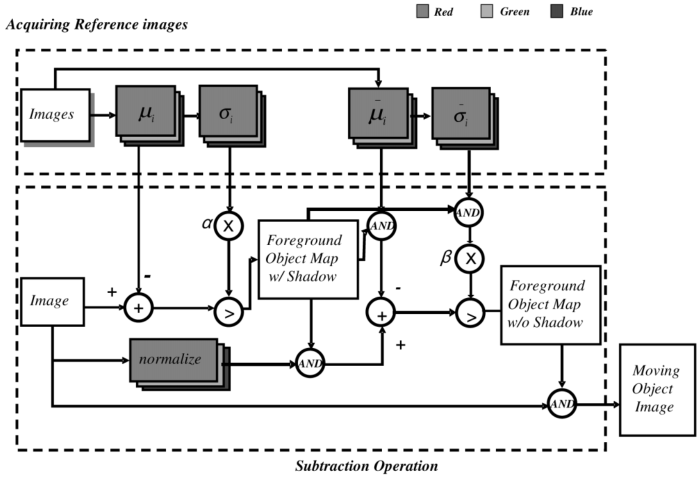


FIGURE 1 The proposed background subtraction.

2.2. Design of SpaceSensor

In human–computer interaction, a natural user interface is an important component of an interactive system. In the proposed user interface, we exploit the user's dynamic gestures as an interaction metaphor. In general, however, it requires intensive computational power and complicated algorithm to process 3D dynamic gestures when a vision technique is exploited. Therefore, we adopt a set of 3D box-based invisible virtual sensors, SpaceSensor, which improves the efficiency of tracking the user's dynamic gestures while maintaining its simplicity.

For the design of SpaceSensor, we need to acquire the segmented information as well as the depth information of the user. The design of the proposed SpaceSensor is focused on tracking natural movements by making it dynamically augmented around the user instead of fixing it at a certain location (Woo et al., 2001). Furthermore, it is augmented into a reachable space from the movements of the user to track their gestures as accurately as possible. To implement the proposed SpaceSensor, we calculate the center position of the user ($H_c = \{H_x, H_y, H_z\}$) as follows:

$$H_c = \frac{1}{N_j}\sum_{i=1}^{N_j} SUD_{j,i}\,, J = \{x, y, z\} \tag{9}$$

where Segmented User Depth information, SUD_j, represents the 3D points and N_j is the number of 3D points in each coordinate within the segmented user's disparity map image. From Equation 9, we can compute requisite parameters for the implementation of SpaceSensor as the following equation:

$$SS_{width} = SS_{height} = SS_{depth} = SUD_{\max\{y_1\}} - SUD_{\min\{y_1\}} \tag{10}$$

where SS_{width}, SS_{height}, and SS_{depth} represent width, height, and depth of SpaceSensor which is based on the user's center point, respectively. $SUD_{\max\{y_i\}}$ and $SUD_{\min\{y_i\}}$ represent the top and bottom points of regions being occupied by the user, respectively. Equation 10 is based on "Leonardo da Vinci: The Vitruvian man" (Morphvs, 2004). From Equation 9 and 10, the proposed SpaceSensor is augmented around the user as shown in Figure 2.

In our design of SpaceSensor, we allocate eight invisible 3D box-based virtual sensors around the personal space. The more 3D box-based sensors, the more accurate the tracking of movement will be. However, there is a trade-off between accuracy and computational complexity of processing the users' dynamic gestures because the increased latency or time delay caused by an increased number of boxes will distract the users when they interact with the system. This allocation of SpaceSensor is based on the height of the segmented user. We determined that eight 3D virtual boxes are enough to track the user's dynamic gestures in a 3D space.

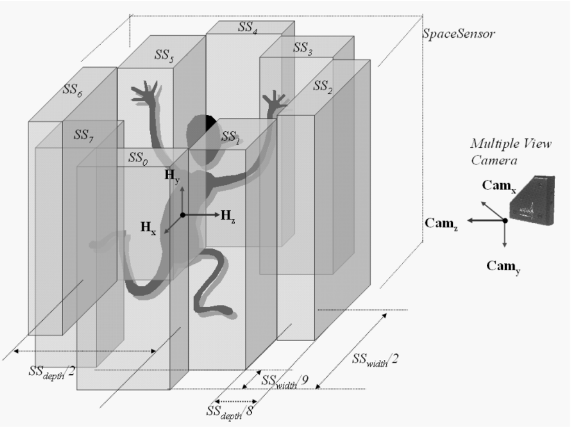


FIGURE 2 Allocation of SpaceSensor.

2.3. Dynamic Gesture Tracking

As explained in the previous subsections, the proposed gesture-tracking technique is simple but accurate. With regard to keeping the tracking technique as simple as possible, we created eight different regions in SpaceSensor, in which each region has its state. The state of SpaceSensor can be represented by SS = {ss0, ss1, …, ssN}, where N denotes the number of 3D box-based virtual sensors. The state of the i-th box, ssi, is denoted by 1 or 0, where 1 indicates that the user is touching the box when the i-th box is observed and 0 indicates that the user is not touching the box. Unlike other vision-based approaches, we keep only the state of SpaceSensor, SS, to track dynamic gestures rather than keep tracking the movement of the user's body parts such as arms, legs, feet, and torso.

As shown in Figure 2, SS{0,…,7} represents the regions of SpaceSensor that cover the user's personal space sufficiently. Through the sequence of states of SS{0,...,7}, a user is able to manipulate virtual objects directly for explicit interactions as well as make gestures for implicit interactions. When a user touches one of eight 3D boxes, its state is changed to 1, and the touched position is calculated as follows:

$$P(x,y,z)=\left(\frac{1}{N_x}\sum_{i=1}^{N_x}x_i,\frac{1}{N_y}\sum_{i=1}^{N_y}y_i,\frac{1}{N_z}\sum_{i=1}^{N_z}z_i\right) \tag{11}$$

where P(x,y,z) represents the touched position in SpaceSensor. Nx, Ny, and Nz are the number of touched points in SpaceSensor.

Figure 3 shows the measurement of the touched position in SpaceSensor. Given the segmented user with the depth information and SpaceSensor, gestures can be tracked by observing how the touched position moves through SpaceSensor. The proposed SpaceSensor is able to track the user's gestures as well as extract additional information using a combination of states of SpaceSensor in real time. For example, the proposed user interface could be applied in determining the move-

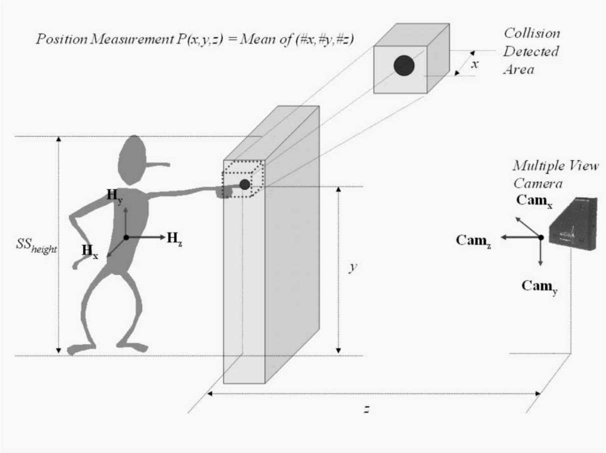


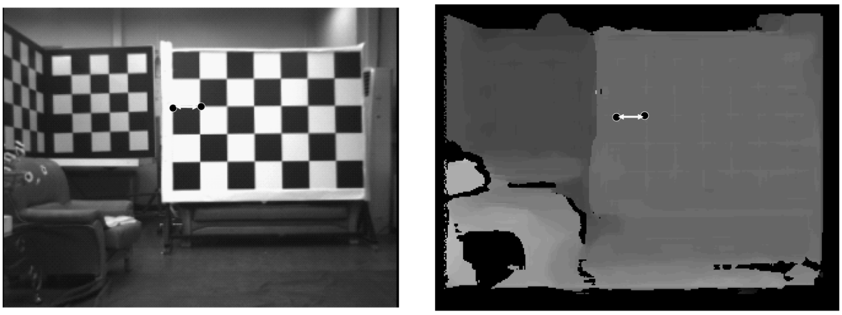
FIGURE 3 Position measurement of SpaceSensor.

ments of the user (Speed), the usage of the personalized space (Large or Small), the weight of movements (Acceleration), and so on. Therefore, the proposed SpaceSensor can be employed as a new type of user interface in a virtual environment without distracting the user.

3. EXPERIMENTAL RESULTS

For the proposed user interface, we utilize the commercially available imaging device, Digiclops^{TM}, and Intel® XeonTM CPU 2.80GHz with 2GB RAM (Point Grey Research Inc., 2002). With this experimental setup, we are able to acquire image sequences in 320 × 240 size with up to 25 fps. Figure 4 shows the experimental configuration to measure the accuracy of SpaceSensor.

As shown in Figure 4, we measure the distance of the two known points (0.25 m) from the acquired disparity map to calculate the error bounds. When the multiview



(a) Real Image

(b) Disparity Map

FIGURE 4 Experimental configuration for SpaceSensor's accuracy.

camera is positioned at 2 m to about 5 m, the minimum and maximum errors are from 0.02 to 0.07 m. It is reasonable to apply SpaceSensor into interactive systems that require large motions of the users. As we indicated in the previous sections, the accuracy of SpaceSensor is directly proportional to the accuracy of segmentation and disparity map.

In general, the higher the resolution of image size, the more information we are able to extract. However, high-resolution images require not only more computational power but also the loss of real time. Thus, we selected relatively low resolution (320 × 240) of image to guarantee real-time user interactions. Table 1 shows the frame rate according to the applied algorithms.

As shown in Table 1, the proposed user segmentation algorithm is adequate to be applied into real-time applications because it uses only 1/23 seconds or 1/12 seconds to separate the user from the given image sequences.

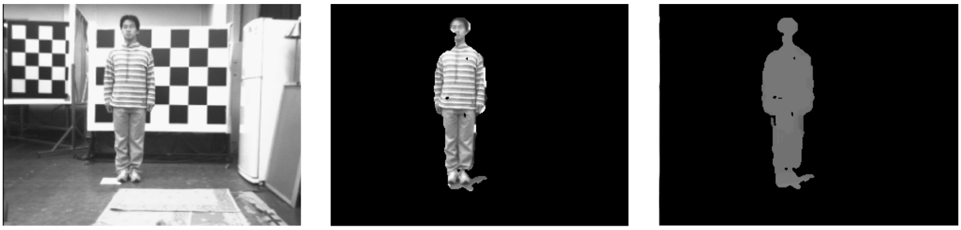
In fact, it is hard to extract the 3D information of the user directly from the disparity image acquired by a multiview camera. This disparity image includes not only the user but also information about background or other objects. Figure 5 shows the results of the proposed user segmentation method.

As shown in Figure 5c, we can extract accurate 3D information of the user of interest if we exploit both the segmented user image and the disparity image.

Figure 6 shows the dynamic augmentation of SpaceSensor around a user. As shown in this figure, we projected the 3D information of the segmented user in the virtual 3D space to show whether the invisible virtual sensors are accurately allocated around the user. As a result of the augmentation, SpaceSensor can change its shape dynamically using the requisite parameters, for example, shrinking and growing SpaceSensor itself. These parameters are automatically calculated in real time. In this regard, the proposed user interface is able to interpret the movements or gestures of the user as values. For example, the volume of SpaceSensor in Figure

Table 1: Processing Result of the Proposed Algorithm

Image Size	*Step in the Proposed Algorithm*	*Frame Per Sec.*
320 × 240	Acquiring live images	25
320 × 240	Acquiring disparity images	14–15
320 × 240	Applying user segmentation algorithm with disparity images	12–13
320 × 240	Applying user segmentation algorithm with live images	23–24



(a) Captured live image (b) The segmented user (c) The user's disparity map

FIGURE 5 Extraction 3D information of the user.

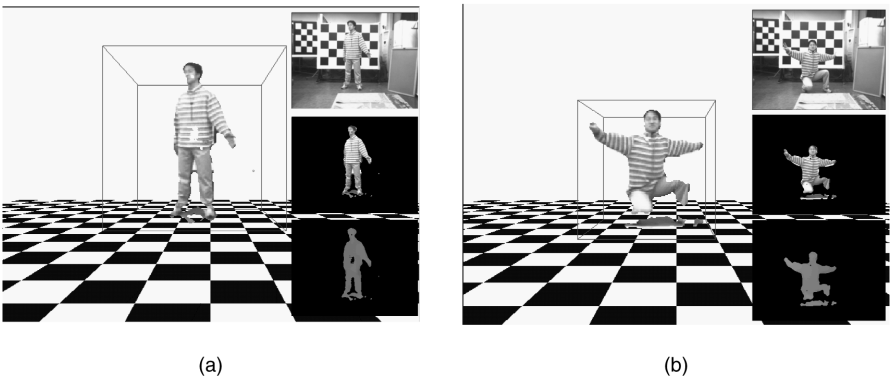


FIGURE 6 Augmentation of SpaceSensor around a user.

6a is larger than Figure 6b, thus we can assume that the user in the pose of Figure 6a is more active compared to the user in Figure 6b.

Figure 7 depicts the top view of the augmented SpaceSensor. As shown in this figure, we can determine the center position of the user approximately even though the depth information is only computed for objects that are located at the closest position to the camera. Thus, the proposed method cannot retrieve full 3D information of the user, but exploits full 3D information when the user interacts with the proposed user interface.

Figure 8 shows the experimental results of detecting the touched positions in SpaceSensor and tracking gestures. As shown in this figure, the 3D information of touched positions is calculated by using Equation 3 when the user touches one of the areas of SpaceSensor. As explained in the previous section, we trace the trajectory of the touched positions to track the user's gestures. The result indicates that SpaceSensor is able to track gestures in any direction around the user. However, in

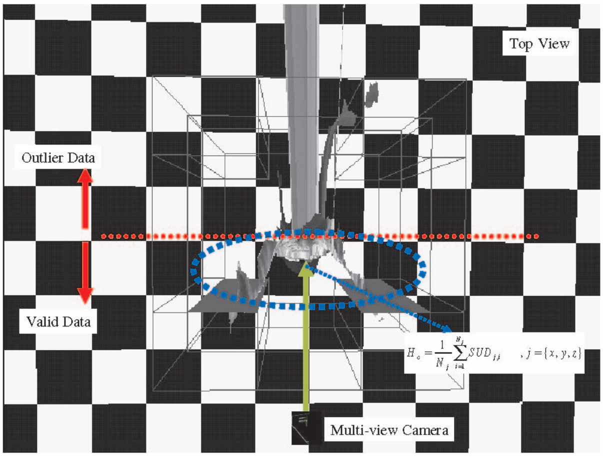

Top View
Outlier Data
Valid Data
$H_c = \frac{1}{N_j}\sum_{i=1}^{N_j} SUD_{i,j} \quad , j = \{x, y, z\}$
Multi-view Camera

FIGURE 7 Top view of SpaceSensor.

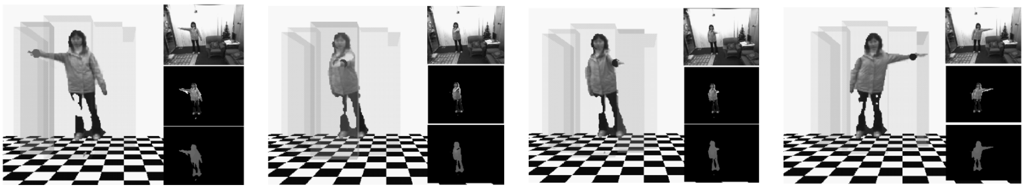


FIGURE 8 Tracking and detection of SpaceSensor.

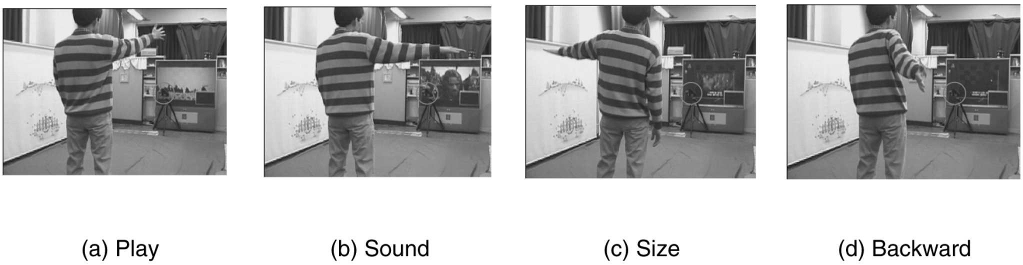
our current scheme, a user must touch certain regions of SpaceSensor to track gestures continuously. Currently, we let the hand position be (0,0,0) when there is no collision with SpaceSensor .

To show the effectiveness of the proposed user interface, we have applied it into real environments and virtual environments, respectively. Figure 8 shows how a user controls a home appliance with his or her gestures.

As shown in Figure 9, the red circle indicates a multiview camera. In this application, SpaceSensor enables the user to play, play forward, and play backward movies through the user's gestures. In addition, the user is able to control the level of sound.

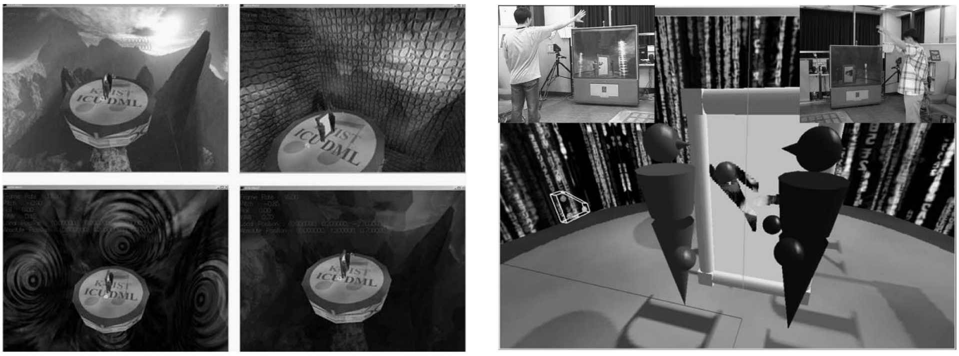
As a user interface of virtual environments, we also apply SpaceSensor into an interactive virtual reality system, in which users are able to experience the interactive expressions and share their experiences over the network in real time. Figure 10 depicts the users' interacting with either virtual objects or each other in a virtual space using the proposed user interface, SpaceSensor.

As shown in Figure 10a, there are several background scenes in the virtual environment that are changeable by the users' interactions. Through the experiments, the participants are able to express their intentions interactively in real time over the network. In the explicit interactions, however, it is difficult for them to scratch the hidden layer on the virtual object due to the restrictions in the accuracy of SpaceSensor. Therefore, we need to improve the collision detection algorithm and accuracy of SpaceSensor to provide interactive expressions.



(a) Play (b) Sound (c) Size (d) Backward

FIGURE 9 Control Movie Player using SpaceSensor.



(a) Virtual environments (scene changing)

(b) User Interactions (scratching)

FIGURE 10 Virtual environments and user interactions through the network.

4. CONCLUSION AND FUTURE WORK

In this article, we proposed a 3D vision-based ambient user interface that enables a user to interact with a system by tracking the user's dynamic gestures in real time. The proposed ambient user interface is an expansion of interaction metaphors as well as interface metaphors in the conventional computing environment. As tested in various applications, SpaceSensor can be exploited as a user interface in various systems because it tracks a user's real-time gestures without distracting the user. Furthermore, it not only overcomes some restrictions of 2D vision-based user interfaces but also resolves the complexity of real-time gesture-tracking algorithms. Eventually, we can utilize SpaceSensor to extract various key features from the user's dynamic gestures such as uses of personal space, activities in the personal space, and emotional cues. If various analytical methods of the extracted key features are possible, then the proposed ambient user interface can also be used as a personalized user interface. However, the proposed ambient user interface is as yet dependent on both the robustness of user segmentation technique and the accuracy of disparity estimation due to its pixel-wise computation algorithm. Therefore, we need to investigate both the user segmentation technique and disparity estimation method at the same time.

REFERENCES

Billinghurst, M. & Kato, H. (1999a). Collaborative mixed reality. In *Proceedings of International Symposium on Mixed Reality* (pp. 261–284).

Billinghurst, M., & Kato, H. (1999b). Real world teleconferencing. In *Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI'99)* (pp. 194–195).

Dey, A. K., Salber, D., & Abowd G. D. (1999). A context-based infrastructure for smart environments. In *Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99)* (pp. 114–128).

Ebihara, K., Davis, L. S., Kurumisawa, J., Horprasert, T., Haritaoglu, R.I., Sakaguchi, T., et al. (1998). Shall we dance? Real time 3D control of a CG puppet. *SIGGRAPH98 Conference Abstract and Applications* (pp. 87–90).

Elgammal, A., Duraiswami, R., Harwood, D., & Davis, L. S. (2002). Background and foreground modeling using non-parametric kernel density estimation for visual surveillance. *Proceedings of the IEEE, 90,* 1151–1163.

Elgammal, A., Harwood, D., & Davis, L. S. (2000). Non-parametric model for background subtraction. In *Proceedings of the 6th European Conference on Computer Vision-Part II, LNCS 1843* (pp. 751–767).

Freeman, W. T., Anderson, D. B., Beardsley, P. A., Dodge, C. N., Roth, M., Weissman, C. D., et al. (1998). Computer vision for interactive computer graphics. *IEEE Computer Graphics and Applications, 18,* 42–53.

Freeman, W. T., & Weissman, C. D. (1995). Television control by hand gestures. In *Proceedings of International Workshop on Automatic Face and Gesture Recognition* (pp. 179–183).

Haritaoglu, I., Harwood, D., & Davis, L. (1998). W4: Who, when, where, what: A real time system for detecting and tracking people. In *Proceedings of Third Face and Gesture Recognition Conference* (pp. 222–227).

Hayward, V. (2001). Survey of haptic interface research at McGill University. In *Proceedings of Workshop on Advances in Interactive Multimodal Telepresence Systems* (pp. 91–98).

Hong, D., & Woo, W. (2003). A background subtraction for a vision-based user interface. In *Proceedings of Pacific-Rim Conference on Multimedia (PCM2003)* 1B3.3.

Horprasert, T., Harwood, D., & Davis, L. S. (1999). A statistical approach for real-time robust background subtraction and shadow detection. In *Proceedings of IEEE ICCV'99 FRAME-RATE Workshop* (pp. 1–19).

Ishii, H., & Ullmer, B. (1997). Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of Conference on Human Factors in Computing Systems (CHI '97)* (pp. 234–241).

Kohler, M. (1997). System architecture and techniques for gesture recognition in unconstraint environments. In *Proceedings of International Conference on Virtual Systems and Multimedia VSMM'97* (pp. 137–146).

Lenman, S., Bretzner, L., & Thuresson, B. (2002). *Computer vision based hand gesture interfaces for human-computer interaction* (Tech. Rep. No. TRITANA-D0209, CID Report). Retrieved January 2003, from http://cid.nada.kth.se/en/publicat/all.html

Maes, P., Darrell, T., Blumberg, B., & Pentland, A. (1995). The ALIVE system: Full-body interaction with autonomous agents. In *Proceedings of Computer Animation '95* (pp. 11–18).

Morphvs. (2004). *Vitruvian Man on planning of temples.* Retrieved December 2004, from http://www.aiwaz.net/modules.php?name=News&file=article&sid=24

Nishikawa, A., Ohnishi, A., & Miyazaki, F. (1998). Description and recognition of human gestures based on the transition of curvature from motion images. In *Proceedings of IEEE International Conference on Automatic Face Recognition* (pp. 552–557).

Park, K., Lee, J., & Kim, J. (2002). Facial and eye gaze detection. In *Proceedings of the Second International Workshop on Biologically Motivated Computer Vision (BMCV 2002), LNCS 2525* (pp. 368–376).

Point Grey Research, Inc. (2002). *Digiclops.* Retrieved February 2002, from http://www.ptgrey.com/products/digiclops/index.html

Shafer, S., Brumitt, B., & Meyer B. (2000). The Easyliving Intelligent Environment system. *CHI Workshop on Research Directions in Situated Computing.*

VRLOGIC Co. (2004a). *DataGlove.* Retrieved July 2004, from http://www.vrlogic.com/html/datagloves.html

VRLOGIC Co. (2004b). *Polhemus.* Retrieved July 2004, from http://www.vrlogic.com/html/tracking_systems.html

Weiser, M. (1991). The computer for the 21st century. *Scientific American, 265*(3), 94–104.

Woo, W., Kim, N., Wong, K., & Tadenuma, M. (2001). Sketch on dynamic gesture tracking and analysis exploiting vision-based 3D interface. In *Proceedings of SPIE PW-EI-VCIP'01, 4310* (pp. 656–666).