Somsak Walairacht
somsak@pi.titech.ac.jp

Keita Yamada
kyamada@pi.titech.ac.jp

Shoichi Hasegawa
hase@pi.titech.ac.jp

Yasuharu Koike
koike@pi.titech.ac.jp

Makoto Sato
msato@pi.titech.ac.jp
Precision and Intelligence Laboratory
Tokyo Institute of Technology
Tokyo, Japan

4 + 4 Fingers Manipulating Virtual Objects in Mixed-Reality Environment

Abstract

This paper describes the construction of a prototype system that allows the user to manipulate virtual objects in a mixed-reality environment. The user can perceive haptic sensations at his or her fingertips and see a virtual world with a live image of his or her hands manipulating the virtual objects. The difficulty of constructing such a visual presentation is the problem of merging the real image with correct geometrical occlusion with images of virtual objects in the simulated virtual world. In this paper, a method of solving registration of real and virtual images to produce mixed-reality scenes is proposed. We have combined this method with our haptic interface device, SPIDAR-8, to construct a system that integrates these realities by using the sense of touch and vision of real images in the virtual world that can be realized during the manipulation of virtual objects. The implemented results are shown and remaining problems are discussed.

1 Introduction

Merging real and virtual images requires the correct geometric alignment of both images in one single reference frame, and such issues have been the focus of research projects on augmented-reality (AR) systems, which try to augment computer graphics with real-world scenes (Feiner, MacIntyre, Haupt, & Solomon, 1993; Milgram & Kishino, 1994; Azuma, 1997). Meanwhile, a virtual reality (VR) environment is a completely synthetic world. The merging of a live image of the user's interaction with the VR system enhances the user's sense of immersion inside the virtual world. Adding a genuine video scene adds a significant degree of richness and realistic detail at minimal computational expense to the completely modeled environment. The integration of haptic sensation realized in VR into AR to become a mixed-reality (MR) system is an attractive improvement.

The WYSIWYF (what you see is what you feel) system is an example of a MR system (Yokohohji, Hollis, & Kanada, 1996). The system provides haptic sensation by employing a PUMA robot arm and displays an image of the user's real hand manipulating the virtual objects. It can perform adequately, but the object manipulation task is limited by the interface device itself.

Many AR researchers describe ways to compose real and virtual images using real-time vision-based registration (Edwards, Rolland, & Keller, 1993; Okuma, Kiyokawa, Takemura, & Yokoya, 1998) to create a realistic 3-D illusion of immersion in the virtual environment. Other researchers implement

Presence, *Vol. 11, No. 2, April 2002,* 134–143

their systems using video see-through or optical see-through displays with a head-mounted display (HMD) (Yokoya, Takemura, Okuma, & Kanbara, 1999) or head-mounted projector (HMP) (Inami et al., 2000). Not only does the user have to put on the HMD or HMP, which limits his or her field of view and sometimes becomes uncomfortable, but also the display of the images is often too dim on the screen of the HMD or on the half-mirror of the HMP.

In this paper, we propose a prototype system for virtual object manipulation in a mixed-reality environment. We have developed a two-handed, multifingered, string-based, haptic interface device. By using this interface device, the user can perceive force feedback at four fingertips (thumb and index, middle, and ring fingers) on each side of both hands. When manipulating an object, the user can see his or her real hands in the virtual environment. A sense of immersion into the virtual world is perceived. In the process of modeling 3-D virtual hands, depth cues about the hands and virtual objects can be determined. In this process, image sequences of real hands are superimposed on the virtual hands, thus ensuring accurate alignment of the real and virtual images.

This paper is organized as follows. Section 2 reviews the current technologies of haptic interface devices and gives a brief description of SPIDAR-8, our prototype haptic interface device. In section 3, we describe the configuration of the proposed MR system. In section 4, we provide details of the implementation of the system. In section 5, we show the implementation of an MR system and discuss the results. In section 6, we conclude and mention our future research plans.

2 Haptic Interface Devices

2.1 Related Works

A haptic interface device allows users to touch and feel the simulated virtual objects with which they interact. Today's devices can be classified in many ways depending on the type of actuators used or the grounding arrangement. Electrical actuators are commonly used because they are clean, safe, and easy to install, whereas hydraulic or pneumatic interfaces are less common. Haptic interface devices need to be attached to an immovable support that can resist the actions of the user. The devices are fastened to either a desktop or to parts of the user's body.

An example of the most popular desktop force feedback interface device is the PHANToM, which is manufactured by SensAble Technologies (Massie & Salisbury, 1994). A weight-counterbalanced and back-drivable arm with six degrees of freedom, the PHANToM is well suited for point interaction by a finger. More dextrous manipulation of virtual objects needs at least two PHANToM arms (one for the thumb and index finger).

Wearable haptic interface devices, such as haptic gloves, attach to the user's body and allow free movement and a larger workspace than do their desktop counterparts. The only commercialized haptic glove is the CyberGrasp (Virtual Technologies), which uses a position-sensing CyberGlove and a cable-driven exoskeleton structure attached to the back of the user's hand. The complex structure guides tendons transmitting forces to the hand produced by electrical motors in a remote control box. This results in high backlash and friction. Moreover, the glove is quite heavy; it can easily lead to user fatigue during prolonged use.

The Rutgers Master II is a research prototype haptic glove, developed at Rutgers University (Burdea, Roskos, Silver, & Lagrana, 1992). This haptic glove is lighter than the CyberGrasp due to the use of pneumatic actuators, which also have low friction and do not overheat. Unfortunately, the Rutgers Master places the actuators inside the user's palm, so the user cannot completely close his or her fist.

To construct a mixed-reality environment in which the user sees a live image of his or her hands manipulating virtual objects, the haptic interface device should not be seen in the composite scene of the virtual world. The aforementioned haptic interface devices are unsuitable because either the user's hand is attached with a structure of actuators or the user's hand is covered and hidden inside a glove. We use our prototype haptic interface device that transparently attaches the user's fingers to lightweight strings. Attaching the device to the hands does not interfere with the view of the user's

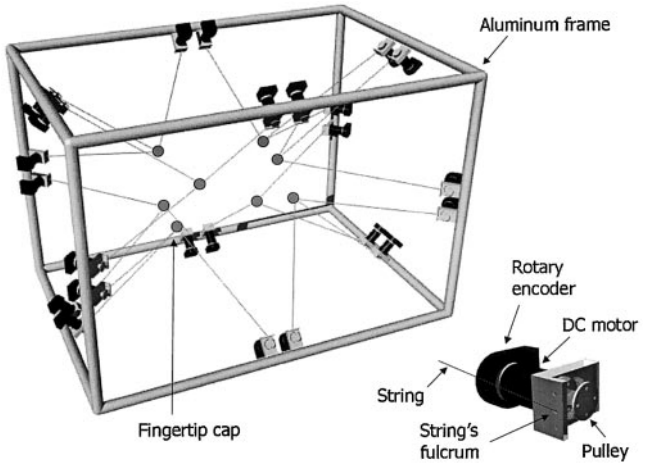

Figure 1. *SPIDAR-8.*

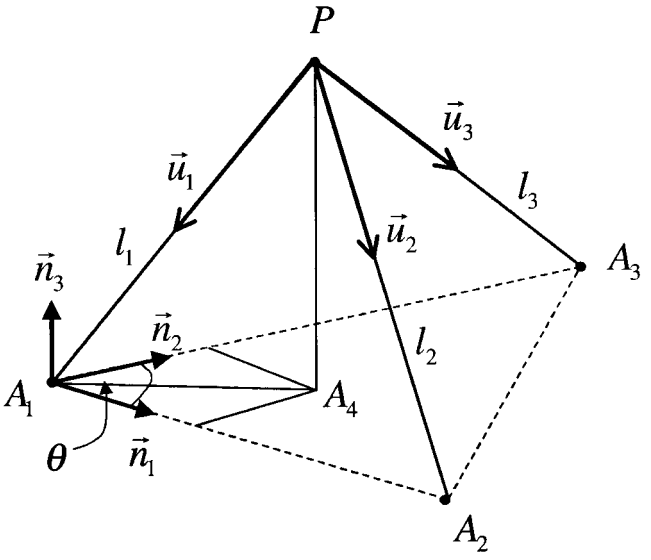

Figure 2. *Position measurement and force feedback generation of SPIDAR-8.*

hands or the virtual objects in the virtual world. A brief description of SPIDAR-8, our prototype haptic interface device, is given in the following section.

2.2 The SPIDAR-8 Two-Handed Multifinger Haptic Interface Device

SPIDAR-8 is a kind of string-based haptic interface device designed for multiple fingers of each hand, as shown in figure 1 (Walairacht, Ishii, Koike, & Sato, 2001). It is used to provide force feedback while the user is manipulating virtual objects in a simulated virtual world. Forces are displayed at four fingertips—the thumb and the index, middle, and ring fingers—on the left and right hand of the user. The device calculates each fingertip position in the virtual space by using the length of three strings. When the position of any fingertip comes into contact with the position of a virtual object, force feedback is generated by controlling the tension of the strings. Methods of position measurement and force feedback generation are briefly explained as follows.

Let l_i $(i = 1, 2, 3)$ be the length of each string measured from a fingertip position, P, to the corresponding string's fulcrum, A_i $(i = 1, 2, 3)$. The vectors $\vec{n}_1$ and $\vec{n}_2$ are unit vectors along the vectors $\vec{A}_2\vec{A}_1$ and $\vec{A}_3\vec{A}_1$, which connect between the string fulcrums A_1A_2 and A_1A_3, respectively. A vector $\vec{n}_3$ is the cross product of the vectors $\vec{n}_1$ and $\vec{n}_2$, which defines the position of P that always lies inside the space enclosed by the frame.

Considering the diagram shown in figure 2, we have

$$\vec{n}_1 = \frac{A_2 - A_1}{\|A_2 - A_1\|}, \tag{1}$$

$$\vec{n}_2 = \frac{A_3 - A_1}{\|A_3 - A_1\|}, \tag{2}$$

$$\vec{n}_3 = \vec{n}_1 \times \vec{n}_2, \tag{3}$$

and the position of point P can be calculated by

$$P = A_1 + \alpha_1\vec{n}_1 + \alpha_2\vec{n}_2 + \alpha_3\vec{n}_3, \quad \alpha_3 \geq 0, \tag{4}$$

where α_1, α_2, and α_3 can be derived from the following equations:

$$\alpha_1 = \frac{1}{\sin^2\theta}\left(\frac{K_1}{d_1} - \cos\theta \cdot \frac{K_2}{d_2}\right), \tag{5}$$

$$\alpha_2 = \frac{1}{\sin^2\theta}\left(\frac{K_2}{d_2} - \cos\theta \cdot \frac{K_1}{d_1}\right), \tag{6}$$

$$\alpha_3 = \sqrt{l_1^2 + \|\alpha_1\vec{n}_1 + \alpha_2\vec{n}_2\|^2}, \tag{7}$$

$$\theta = \cos^{-1}(\vec{n}_1 \cdot \vec{n}_2), \tag{8}$$

$$d_1 = \|A_2 - A_1\|, \tag{9}$$

$$d_2 = \|A_3 - A_1\|, \tag{10}$$

$$K_1 = \frac{1}{2}\{d_1^2 - (l_2^2 - l_1^2)\}, \tag{11}$$

and

$$K_2 = \frac{1}{2}\{d_2^2 - (l_3^2 - l_1^2)\}. \tag{12}$$

To display force feedback at each fingertip of the user, SPIDAR-8 controls the amount of electric current entering each motor. The tension force on each string, t_i ($i = 1, 2, 3$), and the unit vectors, u_i ($i = 1, 2, 3$), are used to compose the resultant force vector as shown by the following equation:

$$\vec{f} = \sum_{i=1}^{3} t_i \vec{u}_i, \tag{13}$$

where

$$\vec{u}_i = \frac{A_i - P}{\|A_i - P\|}. \tag{14}$$

Three strings from three different fulcrums connected to a point for each finger form a triangular cone of possible force displayable to that finger. In fact, force feedback can be correctly displayed only if the resultant force vector lies inside this force cone. However, if the computed resultant force vector lies outside the force cone, the force vector is projected onto the force cone and the resultant force vector is recomposed. In this way, SPIDAR-8 can display force feedback in the appropriate direction and magnitude. The details of the force projection mechanism of SPIDAR-8 can be found in the paper by Walairacht et al. (2001).

3 Mixed-Reality System Setup

The mixed-reality environment setup consists of our prototype haptic interface device, SPIDAR-8, a CCD video camera, and an 18 in. LCD flat-screen display, as shown in figure 3.

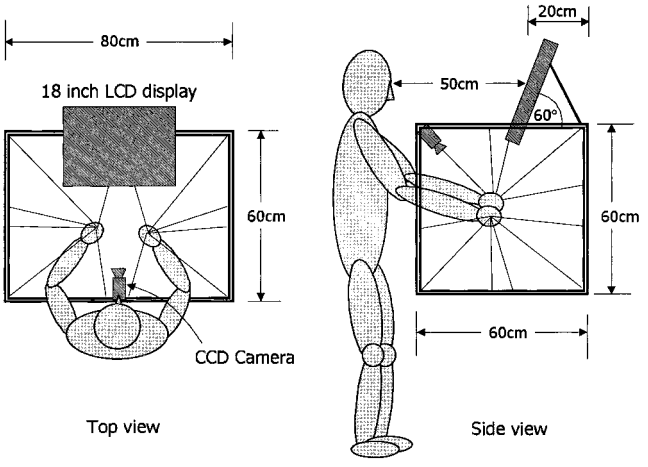

Figure 3. *System configuration.*

3.1 Hardware Configuration

- The frame of the SPIDAR-8 is 80 cm wide, 60 cm deep, and 60 cm high, and is placed on a tabletop 70 cm above the floor.
- A CCD video camera is mounted at the top center of the SPIDAR-8 frame.
- The LCD flat-screen display is mounted on top of the frame, and the screen is inclined about 60 deg. to reduce the reflection of the room's lights for a comfortable and clear view. The screen is 50 cm away from the user's eyes when he is standing 10 cm in front of the frame.

The CCD camera is used to obtain images of the left and right hands of the person using the system. Image sequences of the user's real hands are to be merged into the simulated virtual world and displayed to the user. Because the camera is installed at about a 50 deg. inclination from the center of the frame, we display a simulated virtual world on the LCD display with the same viewing angle.

3.2 Calibration

Calibration is the process of initiating parameter values that map the physical environment to the simulated virtual environment. The parameter values are the position and orientation of the camera, the sensed posi-

tions of SPIDAR-8, and the position and orientation of the virtual objects.

The coordinate systems of each component are related to each other. The central reference is the world coordinate system, which is fixed at the center of the frame of SPIDAR-8. During the operation of the system, all components need to operate in the common framework with reference to the world coordinate system.

Calibration requirements of this system consist of the calibration of SPIDAR-8 and the camera.

3.2.1 SPIDAR-8 Calibration. In this process, an initial position measurement is obtained of the user's eight sensing fingertips. Because the user is allowed to use both hands cooperatively in one working space, we need to calibrate all sensing fingertips in one common framework: the world reference coordinate system.

A calibration pole is placed in the middle of the base of the SPIDAR-8 frame. The tip of the pole points upward to the center of the frame, which represents the origin of the world reference coordinate system ($x = 0$, $y = 0$, $z = 0$). Calibration is performed manually by moving each fingertip while wearing its fingertip cap, attached by three strings, to carefully touch the tip of the calibration pole. The position measurement parameters (the position of the finger, the length of the three strings, and the counter values of each rotary encoder) are initialized when the current calibrating fingertip touches the pole. Because the process of calibration is performed manually, it is difficult to achieve high accuracy. However, the calibration error at this time averages 0.5 cm for each finger.

3.2.2 Camera Calibration. The camera is calibrated by tracking three known positions. We use the method proposed by Yokoya to find the relationship between the world coordinate system and the camera coordinate system. (Yokoya et al. 1999). The camera coordinate system, c, is described by the model-view matrix, M, of a point, w, in the world coordinate system, where

$$c = \mathrm{M}w, \tag{15}$$

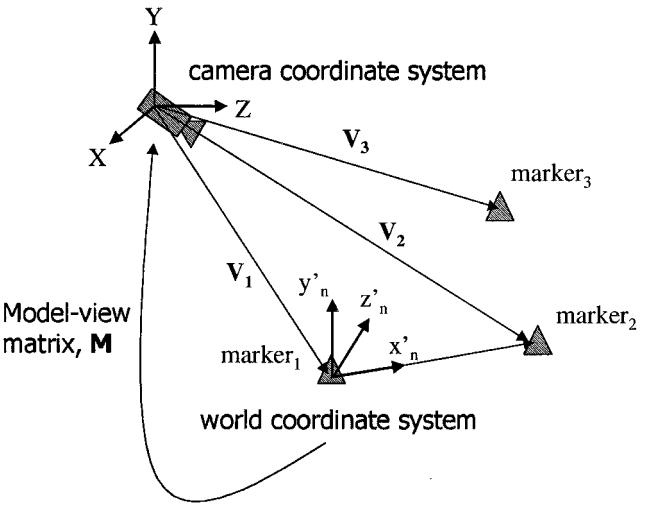

Figure 4. *Camera calibration.*

and

$$\mathrm{M} = \begin{bmatrix} \mathrm{R} & \mathrm{T} \\ 0 & 1 \end{bmatrix}. \tag{16}$$

The positions of the markers (m_1, m_2, and m_3) are given by vectors $\vec{V}_1$, $\vec{V}_2$, and $\vec{V}_3$, respectively. The direction vector on each axis of the world coordinate system can be defined as

$$x_n = \vec{V}_2 - \vec{V}_1, \tag{17}$$

$$y_n = (\vec{V}_3 - \vec{V}_1) - \frac{x_n \cdot (\vec{V}_3 - \vec{V}_1)}{x_n \cdot x_n} \cdot x_n, \tag{18}$$

and

$$z_n = x_n \times y_n, \tag{19}$$

which are normalized as

$$x'_n = \frac{x_n}{\|x_n\|}, \quad y'_n = \frac{y_n}{\|y_n\|}, \quad z'_n = \frac{z_n}{\|z_n\|}. \tag{20}$$

The rotation matrix of the model-view matrix can be determined using $\mathrm{R} = [x'_n, y'_n, z'_n]$, whereas T is the transformation matrix of marker m_1 to the world coordinate system. The alignment of the camera to the world coordinate system can be achieved by using this model-view matrix.

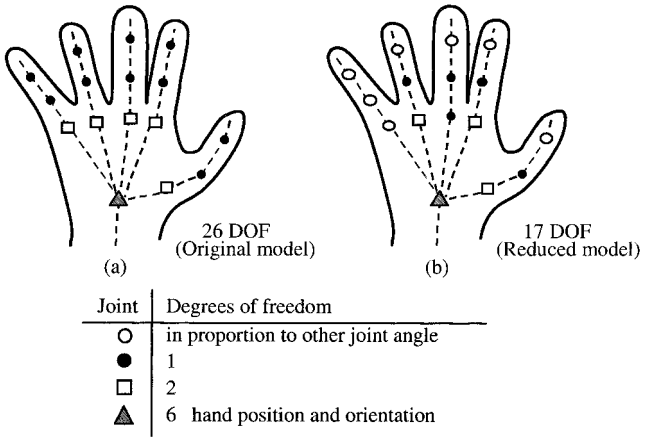

Figure 5. *Model of the real hand and the reduced version of the virtual hand.*

4 Implementation of the Mixed-Reality Environment

First, the virtual hands are modeled from the fingertip positions measured by SPIDAR-8. Next, the virtual hands are compared with the projected image of the user's real hands and adjusted in position to try to make them coincide. Then, the image of the real hands is merged into the virtual world. In this way, correct occlusion of the user's real hands and the virtual objects can be achieved.

4.1 Model of Virtual Hand

The human hand has a very complex structure and is difficult to model because each finger alone has many joints and linkages. The total number of degrees of freedom (DOFs) found on one hand is 26, which includes all joints on all fingers and six DOFs of motion at the wrist. In this system, the number of DOFs on a virtual hand is reduced to 17 to make it possible to estimate the virtual hand pose using the eight fingertip positions measured by SPIDAR-8 (Walairacht, Yamada, Koike, & Sato, 1999). By the following criteria, we can reduce the number of DOFs of a virtual hand. (See figure 5.)

- It is almost impossible to move the joint of the last link, θ_1, (nearest to the fingertip) without moving the penultimate joint, θ_2, or vice versa. After measuring the flex angle of both joints for various amounts of bending, we can approximate the relation between both joints as $\theta_2 = 1.1341\theta_1^2 - 0.286\theta_1$. Therefore, the DOF of the joint of the last link is assigned to be proportional to the penultimate joint on the thumbs and each of the fingers.
- It is assumed that the middle finger cannot be moved side to side in most cases when grasping an object. Hence, the joint at the link of the palm to the middle finger is reduced from two DOFs to one DOF.
- The little finger is not measured by SPIDAR-8; therefore, the joint motion of the little finger is assumed to resemble the joint motion of the ring finger. No DOFs of the little finger are computed.

Because only four fingertips on each of the user's two hands are measured, the information is insufficient to solve the registration problem. Therefore, additional hand position information is necessary; the following data issued for this purpose are the projected wrist position and the joint-ball.

4.1.1 Projected Wrist Position. A projected wrist position can be determined from screen data when an image of the user's real hand (taken by the camera) is displayed on the screen. The user wears a red bracelet marker on the wrist, as shown in figure 6. The screen data is raster-scanned, and the marker can then be extracted by the color-filtering technique. Using the ends of the extracted marker, a position at the center is computed. This position represents a projected wrist position of that hand.

4.1.2 Joint-Ball. In many cases, the projected images of the user's real hands are not completely coincident with the virtual hands, especially at the finger joints. Such errors can be corrected by adjusting the finger joints of the virtual hands. By comparing the silhouettes of the real hand and the virtual hand in the camera image, a joint-ball is defined at each finger joint on the silhouette of the virtual hand, as shown in figure 7.

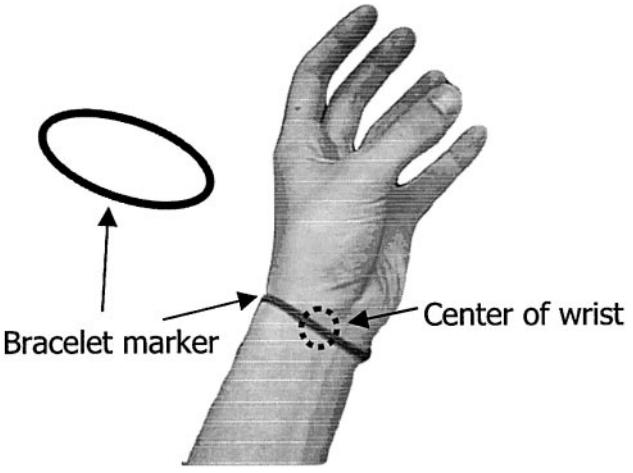

Figure 6. *Bracelet marker.*

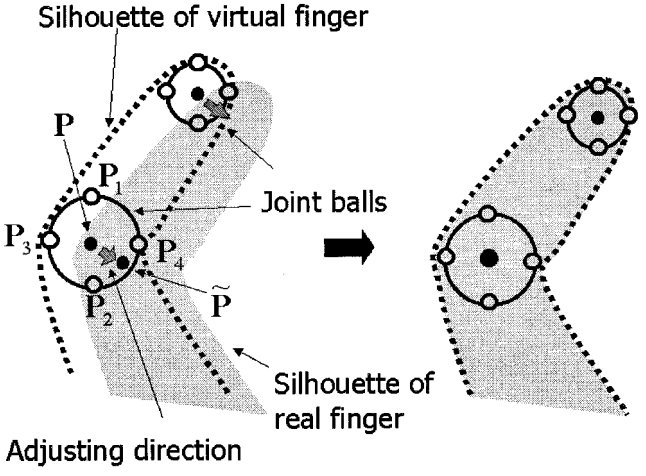

Figure 7. *Joint-ball.*

Point P is assigned to be an initial position, and four adjacent points (P_1, P_2, P_3, and P_4) define the region of a joint-ball. By considering the region where a joint-ball overlaps the silhouette of the real hand, a target position, $\tilde{P}$, can be computed by

$$\tilde{P} = P + \alpha \sum_{i=1}^{4} \tau_i (P_i - P). \qquad (21)$$

In this equation, the value of τ_i is equal to 1 if P_i overlaps the silhouette of the real hand, and otherwise is equal to 0. The constant α is defined as an acceleration coefficient for moving a joint-ball to its target position.

4.2 Successive Adjustment of Model Parameters

In this application, a model of the real hand consists of four fingertip positions measured by SPIDAR-8 and the projected wrist position. A model of the virtual hand is constructed using the initial fingertip positions and wrist position. If all parameters of both models are equal, the projected image of the real hand can completely coincide with the virtual hand. However, in practice, they are always different. The model of the virtual hand is adjusted at every update. The adjusting amounts can be found from the model of the real hand at the current step and the approximately adjusted model of the virtual hand in the next step. The successive adjustment can be described as follows.

Let the model parameters of the virtual hand be defined as

$$Q = (Q_1, Q_2, Q_3, Q_4, Q_5), \qquad (22)$$

where Q_1, Q_2, Q_3, and Q_4 are the fingertip position of the thumb and the index, middle, and ring finger, respectively, and Q_5 is a position at the wrist.

In the same way, the model parameters of the real hand are defined as

$$\tilde{Q} = (\tilde{Q}_1, \tilde{Q}_2, \tilde{Q}_3, \tilde{Q}_4, \tilde{Q}_5), \qquad (23)$$

where $\tilde{Q}_i$ ($i = 1, 2, 3, 4$) are measured fingertip positions measured by SPIDAR-8, and $\tilde{Q}_5$ is a projected wrist position.

The initial positions of the joint-balls are

$$P = ((x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)), \qquad (24)$$

and the target positions of the joint-balls are

$$\tilde{P} = ((\tilde{x}_1, \tilde{y}_1), (\tilde{x}_2, \tilde{y}_2), \ldots, (\tilde{x}_k, \tilde{y}_k)). \qquad (25)$$

The differences in the model parameters, ΔQ, can be computed from the model parameters of the real and virtual hands by using

$$\Delta Q = \tilde{Q} - Q. \tag{26}$$

The differences in the position of the joint-balls, ΔP, can be found from the initial and target positions of the joint-balls by using

$$\Delta P = \tilde{P} - P. \tag{27}$$

The integrated error, J, is defined to be the sum of the normed differences of the model parameters, ΔQ, and the joint-balls, ΔP, with a relative weighting coefficient c between ΔQ and ΔP:

$$J = \|\Delta Q\|^2 + c\|\Delta P\|^2. \tag{28}$$

If the integrated error is small enough, the model parameters of the real hand and the model parameters of the virtual hand are assumed to be the same. In this case, the projected image of the real hand is nearly coincident with the virtual hand.

In the following algorithm, an adjustment of the model of a virtual hand, θ, at the n^{th} step determines the model parameters for the $n + 1^{\text{st}}$ step. The adjustment amounts can be computed based on an optimization criteria.

1. Construct the model parameters of the real hand, $\tilde{Q}$, as in equation (23), from the updated fingertip positions measured by SPIDAR-8 and the extracted projected wrist position.
2. Compute the differences of the positions of the joint-balls, ΔP, by equation (27).
3. Compute the differences of the model parameters, ΔQ, by equation (26).
4. Consider the integrated error, J. If the integrated error is small enough, the process is finished. Otherwise, continue.
5. Compute the adjustment amounts, $\Delta\theta$, and adjust model parameters, θ_n, to be model parameters, θ_{n+1}.
6. Repeat the algorithm from step 2.

The computation of the adjustment amounts, $\Delta\theta$, in step 5 is described in the following details.

Generally, the adjustment amounts, $\Delta\theta$, are the differences between the model parameters in the next step and the model parameters at the current step:

$$\Delta\theta = \theta_{n+1} - \theta_n. \tag{29}$$

Because the model parameters $\tilde{Q}$ are available, the model parameters of the $n + 1^{\text{st}}$ step should be $Q(\theta_{n+1})$. Then, the integrated error can be expressed as

$$J(\theta_{n+1}) = \|Q(\theta_{n+1}) - \tilde{Q}\|^2 + c\|P(\theta_{n+1}) - \tilde{P}\|^2. \tag{30}$$

The model parameters $Q(\theta_{n+1})$ can be linearly approximated as the sum of the model parameters $Q(\theta_n)$ with the adjustment amounts that can be found from the Jacobian matrix, $\mathrm{J_Q}$:

$$Q(\theta_{n+1}) = Q(\Delta\theta + \theta_n), \tag{31}$$

$$\cong Q(\theta_n) + \mathrm{J_Q}(\theta_n)\Delta\theta, \tag{32}$$

where the Jacobian metric $\mathrm{J_Q}(\theta_n)$ is

$$\mathrm{J_Q}(\theta_\mathrm{n}) = \frac{\partial Q(\theta_n)}{\partial \theta_n}. \tag{33}$$

In the same way, we can linearly approximate the target positions of the joint-balls $P(\theta_{n+1})$ as

$$P(\theta_{n+1}) \cong P(\theta_n) + \mathrm{J_P}(\theta_n)\Delta\theta. \tag{34}$$

Therefore, we can linearly approximate the integrated error in equation (30) by the approximation of $Q(\theta_{n+1})$ and $P(\theta_{n+1})$ from equation (32) and (34), respectively, as

$$J(\theta_{n+1}) \cong \|\mathrm{J_Q}(\theta_n)\Delta\theta - (\tilde{Q} - Q(\theta_n))\|^2 + c\|\mathrm{J_P}(\theta_n)\Delta\theta - (\tilde{P} - P(\theta_n))\|^2. \tag{35}$$

Because the integrated error, $J(\theta_{n+1})$, is assumed to be very small, we can solve for $\Delta\theta$ by rearranging the second-degree polynomial equation (35) into the simple form

$$\mathrm{A}\Delta\theta = \mathrm{b}, \tag{36}$$

which implies

$$\Delta\theta = \mathrm{A}^{+}\mathrm{b}, \tag{37}$$

where

$$A = J_Q(\theta_n)^T J_Q(\theta_n) + cJ_P(\theta_n)^T J_P(\theta_n) \quad (38)$$

and

$$b = J_Q(\theta_n)^T(\tilde{Q} - Q(\theta_n)) + cJ_P(\theta_n)^T(\tilde{P} - P(\theta_n)). \quad (39)$$

4.3 Merging the Image of Real Hands into a Virtual World

A CCD video camera is used to record the images of the user's real hands when performing a manipulation task. The chroma key technique is used to extract only the image of the hands from a blue-screen background. We implemented a software routine for chroma keying on the graphics server. After the model of the virtual hands has been adjusted as already described in the previous sections, the color pixels of the virtual hands are blitted to be transparent on the screen. From the three-dimensional position information from our model of the virtual hands, the virtual hands always occlude the virtual objects in the correct geometrical position. The extracted image of the real hands are then superimposed on the transparent virtual hands. In the virtual world scene, the image of the real hands can be merged with the virtual objects with correct occlusion.

5 Implementation Result and Discussion

Figure 8 shows the constructed mixed-reality environment in which the user is manipulating a virtual Rubik's cube with force sensations and can see his real hands in the virtual world manipulating the virtual cube.

We can observe the registration errors in the mixed-reality scenes in figure 9. Space appeared between the user's fingers and the virtual cube in some cases, and sometimes certain parts of the finger were covered by the virtual cube. We suspect that these errors are caused by the heavy computational requirements of this process. The estimation of the virtual hands' poses is slow, and it is very difficult to make the virtual hands coincide exactly with the real hands. In addition to the limited measurement position information provided by the interface device, the estimated position information using the joint-balls and the projected wrist position were employed. Although the estimation improved, some error still remains. Further, the graphics server currently has inadequate computational power to handle these tasks quickly enough to eliminate these errors. Replacing the graphics server with a higher-performance computer could help reduce registration errors.

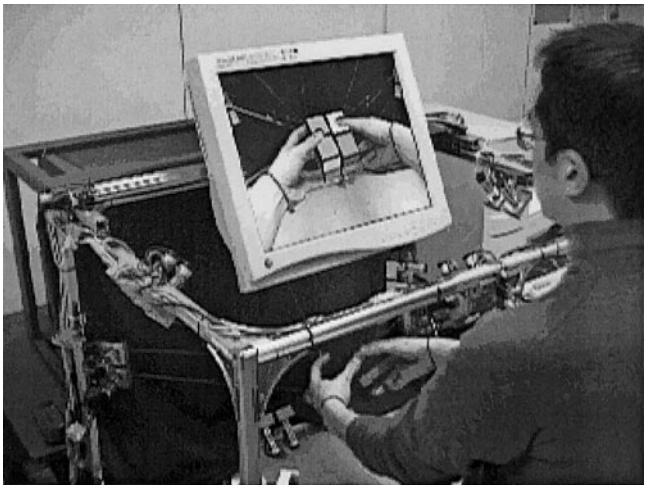

Figure 8. *The constructed system providing a mixed-reality environment.*

6 Conclusions

In this paper, we have proposed a mixed-reality environment system in which an image of the user's real hands is merged into a virtual world when performing virtual object manipulation. The user can see his or her real hands and perceive force feedback at the fingertips when manipulating virtual objects. The composition of real and virtual images with correct registration can be achieved by a successive adjustment algorithm that we have introduced. The implementation results have shown a feasible mixed-reality environment. Although the problem of registration remains in some cases, the proposed system is promising for a feasible implementation of a mixed-reality system.

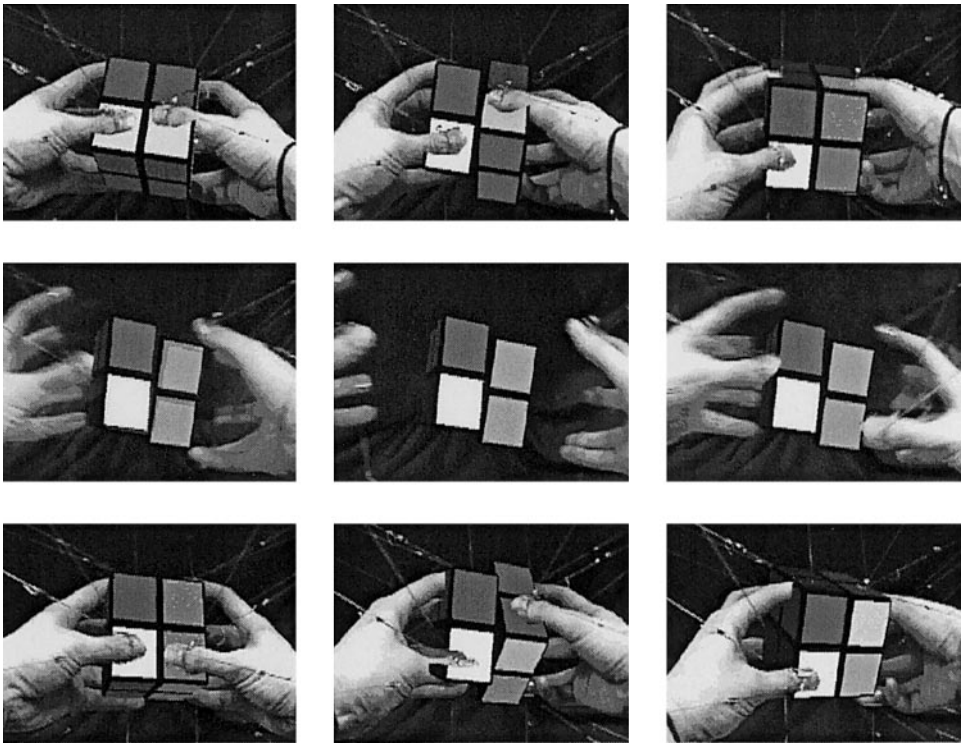

Figure 9. *Screens of the mixed-reality environment.*

References

Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments, 6*(4), 355–385.

Edwards, E. K., Rolland, J. P., & Keller, K. P. (1993). Video see-through design for merging of real and virtual environments. *Proceedings of VRAIS'93,* 197–204.

Feiner, S., MacIntyre, B., Haupt, M., & Solomon, E. (1993). Windows on the world: 2D windows for 3D augmented reality. *Proceedings of UIST'93,* 145–155.

Inami, M., Kawakami, N., Sekiguchi, D., Yanagida, Y., Maeda, T., & Tachi, S. (2000). Visuo-haptic display using head-mounted projector. *Proceedings of VR2000,* 233–240.

Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality virtual displays. *IEICE Transaction on Information and Systems, E77-D*(9), 1321–1329.

Okuma, T., Kiyokawa, K., Takemura, H., & Yokoya, N. (1998). An augmented reality system using a real-time based registration. *Proceedings of ICPR'98, 2,* 1226–1229.

Massie, T., & Salisbury, K. (1994). The PHANToM haptic interface: A device for probing virtual objects. *ASME Winter Annual Meeting, DSC, 55-1,* 295–300.

Burdea, G., Roskos, E., Silver, D., & Lagrana, N. (1992). A portable dextrous master with force feedback. *Presence: Teleoperators and Virtual Environments, 1*(1), 18–27.

Tsai, R. Y. (1986). An efficient and accurate camera calibration technique for 3D machine vision. *Proceedings of CVPR,* 364–374.

Walairacht, S., Yamada, K., Koike, Y., & Sato, M. (1999). Modeling virtual hands with haptic interface device. *Proceedings of ICAT'99,* 233–236.

Walairacht, S., Ishii, M., Koike, Y., & Sato, M. (2001). Two-handed multi-fingers string-based haptic interface device. *IEICE Transactions on Information and Systems, E84D*(3), 365–373.

Yokohohji, Y., Hollis, R. L., & Kanada, T. (1996). What you see is what you can feel—Development of a visual/haptic interface to virtual environment. *Proceedings of VRAIS'96,* 46–53.

Yokoya, N., Takemura, H., Okuma, T., & Kanbara, M. (1999). Stereo vision based video see-through. In Y. Ohta & H. Tamura (Eds.), Mixed Reality: Merging Real and Virtual Worlds (pp. 131–145). Tokyo: Ohmsha.