VisHap: Augmented Reality Combining Haptics and Vision*

Guangqi Ye[1], Jason J. Corso[1], Gregory D. Hager[1], Allison M. Okamura[1,2]
Departments of [1]Computer Science and [2]Mechanical Engineering
The Johns Hopkins University
grant@cs.jhu.edu aokamura@jhu.edu

Abstract – *Recently, haptic devices have been successfully incorporated into the human-computer interaction model. However, a drawback common to almost all haptic systems is that the user must be attached to the haptic device at all times, even though force feedback is not always being rendered. This constant contact hinders perception of the virtual environment, primarily because it prevents the user from feeling new tactile sensations upon contact with virtual objects. We present the design and implementation of an augmented reality system called VisHap that uses visual tracking to seamlessly integrate force feedback with tactile feedback to generate a "complete" haptic experience. The VisHap framework allows the user to interact with combinations of virtual and real objects naturally, thereby combining active and passive haptics. An example application of this framework is also presented. The flexibility and extensibility of our framework is promising in that it supports many interaction modes and allows further integration with other augmented reality systems.*

Keywords: Haptics, human-computer interaction, computer vision, visual tracking

1 Introduction

In recent years, many human-computer interaction and virtual environment systems have incorporated haptic devices. A general survey reveals that in most haptic systems, the user must be constantly attached to the haptic device in order to feel the generated forces. The user typically grasps a stylus or places a fingertip in a thimble. Continuous contact with a physical tool hampers the perception of the virtual environment and human-computer interaction in many ways. One primary reason is that it prevents the user from feeling new tactile sensations upon contact with virtual objects. In general, haptics includes both kinesthetic (force) and cutaneous (tactile) information. Most commercially available devices only apply force feedback. Devices specifically designed for tactile feedback, e.g. [4, 10], are typically complex and not yet integrated with force feedback devices.

Furthermore, most haptic devices have a very limited workspace. For example, the workspace of the PHANToM Premium 1.0A model [1, 7], which is our experimental platform, is approximately a $13cm \times 18cm \times 25cm$ rectangular solid. If the user has to attach his hand to the device all the time and move his or her hand in such a limited space, it would be hard to extend the virtual environment to incorporate rich interaction elements. In addition, constant contact impairs the experience of virtual environment because it acts as a constant reminder to the user that he or she is interacting with a virtual world through a tool.

To overcome these drawbacks, we designed and implemented an augmented reality system that employs visual tracking to seamlessly integrate force feedback with tactile feedback in order to generate a "complete" haptic experience. The basic idea is to incorporate a computer vision system to track the user's movement so direct contact via a physical tool becomes unnecessary. The framework also allows the user to interact with combinations of virtual and real objects naturally, thereby combining active and passive [5] (or kinesthetic and cutaneous) haptics. Our work builds upon previous research in encountered-type haptic displays [12, 14], with the goal of creating a higher fidelity haptic interaction with both passive and active elements.

The VisHap system is composed of three subsystems: computer vision, the haptic device, and an augmented environment model, which is also called the world subsystem. The vision subsystem tracks the movement of the user's finger and transfers the 3D position and velocity of the finger to the augmented environment model. Our current implementation includes stationary static stereo cameras and XVision [2]. The haptic device is controlled as a robot to meet the finger at the point of contact with a virtual object. Once contact is made, a hybrid control allows the user to feel virtual dynamics in the direction normal to the object surface, while maintaining the position of the haptic device directly behind the finger. The character of this feedback depends on

*0-7803-7952-7/03/$17.00 © 2003 IEEE.

the environment model. The augmented environment model defines the physical configuration and interaction properties of all the virtual and real objects in the environment, such as the position and surface properties of the virtual objects and the passive objects attached to the haptic device end-effector. Once the user is close enough to a virtual object, the augmented environment model sends a command to the haptic device to prepare to meet the user in space. When the user is interacting with the object, the augmented environment model continuously sends the position of the user to the haptic model, and the haptic device applies force feedback to the user's finger according to the positions of the finger and the object, as well as the dynamic properties of the object. For example, the virtual object may be designed to allow the user to push a virtual button, slide along a virtual wall, or hit a virtual ball. In each case, a proper passive element is attached to the haptic device end-effector and the corresponding interaction mode is defined. Our framework is flexible and extensible because it supports many interaction modes and allows further integration with other augmented reality systems, such as head-mounted displays.

In Section 2 we describe the framework and implementation of our system, as well as some example applications. We present experimental results for our system in Section 3. Section 4 provides the conclusions drawn from this work.

1.1 Related Work

Incorporating haptics into virtual environments is a promising field. Insko et al. [5], show that augmenting a high-fidelity visual virtual environment with low-fidelity haptics objects, which they call "passive haptics," can increase participant's sense of presence as measured by subjective questionnaires, observed participant behaviors and physiological responses. Experiments show that in navigating an identical real environment while blindfolded, those participants trained in a virtual reality (VR) augmented with passive haptics performed significantly faster and with fewer collisions than those trained in a non-augmented virtual environment.

Salada et al. [9] investigated the use of fingertip haptics to directly explore virtual environments instead of via an intermediate grasped object (a tool). They render the relative motion between the fingertip and the object surface using a rotating drum or sphere. Their experiments show that relative motion between the fingertip and the object surface is a key psychophysical aspect of fingertip haptics.

Touch/force display system [14] is probably the first system based on the encountered-type haptic device concept. An original optical device was designed to track the finger position. When the finger is in contact with a virtual object, the device contacts the skin. However, it not possible to attach additional passive objects to the device.

Yokokohji et al. [12] implemented a haptics/vision interface, WYSISYF (What You See Is What You Feel), for VR training system for visuo-motor skills. Using visual tracking and video keying, the system is registered so that the visual and haptic displays are consistent spatially and temporally. A Puma robot is used to simulate the physical interaction, so high fidelity haptic sensations cannot be conveyed. Yokokohji et al. [13] also studied the problem of multiple objects in an encountered-type virtual environment. They present path planning techniques to avoid collisions between the hand and the robot, for a single robot attempting to apply multiple encountered-type virtual haptic objects.

2 System Design and Implementation

The VisHap system is composed of three parts, called the vision, haptics and world subsystems. Figure 1 illustrates the configuration of the system. The user is interacting in an augmented reality that is composed of several virtual planes around the workspace and a passive key (removed from a conventional computer keyboard) attached to the end of a haptic device. A stereo camera system tracks the finger in 3D space. The user can observe the scene on a standard PC monitor or a head mounted display (HMD).

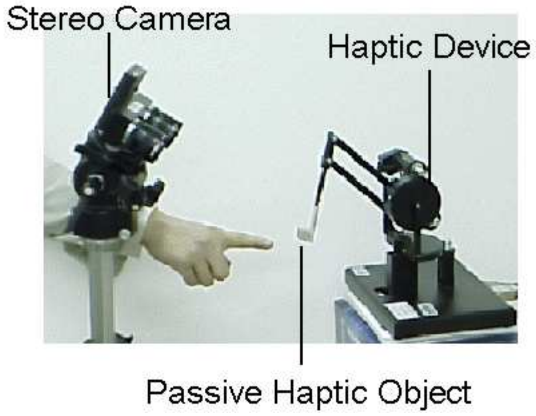

Figure 1: Image of the VisHap system.

The vision subsystem is composed of a stereo camera and tracking software. It is responsible for capturing the scene and tracking the user's finger in real time. Since 3D registration between the vision and haptics modules is required, the cameras are calibrated in order to calculate the 3D position of the finger in the coordinate system of one of the cameras. To enhance the flexibility of the system, we used automatic finger detection and tracking without any attached marker or special imaging media.

The haptic subsystem, which consists of a 3D haptic device, is the interface between the virtual/augmented environment and the user. When the user is far from objects in the environment (set by a threshold), the haptic device is motionless and the user moves his finger freely in space. When the user is in contact with a virtual object, the haptics module simulates the sensation of the interaction through force feedback to the user's finger. To simulate different objects under different interaction scenarios, the haptic module produces corresponding force feedback to the user. In order to display appropriate tactile sensations, a real "passive" haptic object is attached to the end-effector of the haptic device.

The world subsystem acts as the overseer of both vision and haptic subsystems. It is responsible for defining the configuration of the whole augmented reality, specifying the properties (e.g., position, orientation, dimensionality, surface property, etc.) of all virtual and real objects, rendering the virtual environment to the user if necessary, and carrying out the 3D registration between vision subsystem and haptic subsystem. At any time, it queries the vision system to check whether the user's finger is in the scene, and if so, transforms the 3D position and velocity of the finger to the world coordinate system. When the finger is close to certain virtual or real objects in the environment, which indicates that an interaction is possible, it sends the positions of the finger and object to the haptic subsystem and notifies the haptic subsystem to prepare for the coming interaction. During interaction, it continues sending the necessary information to the haptics module. Graphical rendering of the environment can be implemented using standard computer graphics and video processing techniques. For augmented reality, the video of the scene is displayed to the user, with the haptic device "deleted" and a virtual object overlaid. A head mounted display can be used to achieve better immersiveness.

Our framework has several advantages compared to the encountered-type systems discussed in Section 1 and those that require constant contact with a haptic device. First, it allows a much richer set of haptic interaction, including kinesthetic and cutaneous sensations. Interaction with real "passive" haptic objects is easy to configure. Second, the system is very flexible. The configuration of the scene, the properties of virtual objects and the mode of interaction can all be customized by editing corresponding configuration files, without many changes to the hardware. Furthermore, almost all hardware components in our design are standard devices, therefore easing the difficulties of implementation and integration. A standard PC, with a stereo camera system and a haptic device, is the minimum requirement. Some of the software implementations are recent research topics [1, 3, 6, 8] and even free to download [2].

In our current implementation, the entire system runs on a Pentium III PC with the Linux operating system. A SRI Small Vision System (SVS) with a STH-MDCS stereo head by Videre Design is the imaging unit. A PHANToM Premium 1.0A model [1, 7] from SensAble Technologies is the device used to simulate haptic interaction.

2.1 Vision Subsystem

As mentioned earlier, the main task of the vision subsystem is to track the user's finger and provide 3D information and video to the world subsystem. Using the SVS system, we capture real-time image pairs of the scene and calculate disparity information to acquire 3D data. In our current implementation, we assume that the user is interacting with the scene using a single finger. We perform fingertip tracking on the left color image and compute the 3D position of the finger in the coordinate system of the left camera.

2.1.1 Appearance-based Hand Segmentation

An advantage of our system is that it does not require any special marker on the user's finger. Instead we perform efficient and robust appearance-based skin segmentation on the color image [11]. The basic idea of the appearance model is to split the image into small image tiles and build a hue histogram for each of the image patches. At the start of each session, we carry out a fast on-line learning procedure of the background scene by averaging the first several images and building the appearance model for the averaged image. For future images, we also build the histogram for image patches in the region of interest and carry out pairwise histogram comparison with the background model. Histogram intersection [11] is an efficient way to match histograms.

$$H(I, M) = \frac{\sum_{j=1}^{n} min(I_j, M_j)}{\sum_{j=1}^{n} M_j} \qquad (1)$$

Here I and M refer to the model and measure histograms, respectively. The match score is the criterion of foreground segmentation.

Another offline procedure is carried out to learn the color appearance model of human skins. We collect the training data by recording image sequences with segmented hands. We convert all skin pixels from RGB color space to HSV color space and learn a single Gaussian model of the hue distribution. We perform a skin/non-skin check on every foreground pixel by thresholding the probability that the given pixel belongs to the skin model, and then filter out non-skin points. Then a median filter operation is used to remove noise. The result of the whole hand segmentation procedure is a binary image with foreground pixels indicating the skin region. Figure 2 shows an example segmentation result.

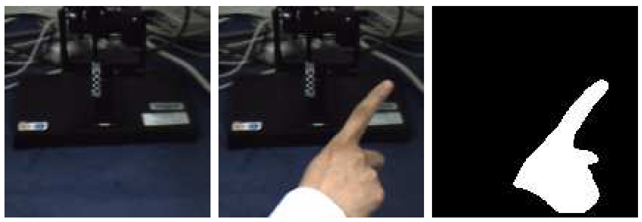

Figure 2: An example of background image (left), foreground image (middle) and segmentation result (right).

2.1.2 Fingertip Detection and Tracking

An efficient way to detect the fingertip is to exploit the geometrical property of the finger [3, 8]. We use a cylinder with a hemispherical cap to approximate the shape of the finger. The radius of the sphere corresponding to the fingertip (r) is approximately proportional to the reciprocal of the depth of the fingertip with respect to the camera (z). We model this relationship with $r = K/z$. K is determined by the experiment configuration.

A series of criteria are checked on the candidate fingertips to filter out false fingertips. The following pseudo-code segment illustrates the algorithm.

```
$\forall p(x, y) \in$ search region
  1. IF ( $p(x, y)$ is not a skin pixel ) CONTINUE LOOP
  2. Calculate the $z$-coordinate of p in the camera system
  3. Compute desired fingertip radius $r = \frac{K}{z}$
  4. Calculate number of skin pixels $S_{filled}$ in the circle $C$ centered at $p$ with radius $r$
  5. IF ( $S_{filled} <$ area of the circle $C$ ) CONTINUE LOOP
  6. Compute number of skin pixels $S_{square}$ along a square $S$ centered at $p$ with side-length $r_2 = r + \delta r$
  7. IF ( $S_{square} < 2 \times r_2$ or $S_{square} > 4 \times r_2$ ) CONTINUE LOOP
  8. Check pairwise diagonal pixels along $S$ and calculate number of pairs $N$ that both pixels are skin points
  9. IF ( $N > 1$ ) CONTINUE LOOP
  10. Record $p(x, y)$ as a candidate fingertip with $score = S_{filled}/$ area of the circle $C$
```

The circle assumption of the fingertip is enforced by checking the number of points in the neighboring circle. The square is examined to make sure that there is a cylinder of reasonable size connected to the circle. The diagonal check aims to remove false fingertips that may appear in the middle of the finger. In most cases this algorithm outputs multiple candidates around the true location of the fingertip. We select the one with the highest score to be the fingertip. Figure 3 displays an example detection result.

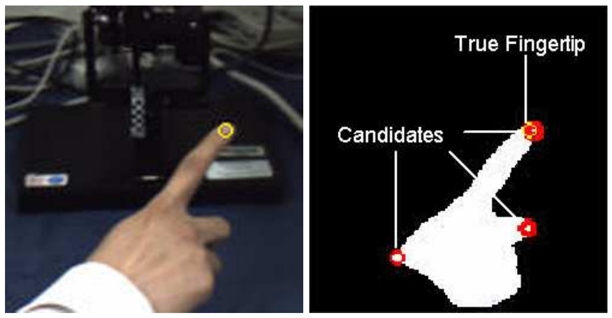

Figure 3: Fingertip detection result: the foreground image and detected fingertip (left) and the candidate fingertips on the segmented foreground image (right).

We implement a simple Kalman Filter to predict the position of the fingertip in each frame to improve real time fingertip tracking. We assume that the fingertip moves approximately at a constant velocity and in a straight line. We search for the fingertip in a small window around the predicted position using method similar to that used for finger detection. Our experiments show that this algorithm tracks the fingertip quite accurately and robustly. The position of the tracked fingertip is computed in the coordinate system of the left camera frame by combining calculated disparity image and the camera parameters.

2.2 World Subsystem

As the overseer of the entire VisHap system, the world subsystem is responsible for performing 3D vision/haptics registration, scene rendering, notifying the haptic device about imminent interaction, etc.

Since the coordinate systems of the camera and haptic device are canonical Euclidean frames, they are related by a rigid transformation. During the system calibration process, we move the haptic device around in the field of view of the camera system and record more than three pairs of coordinates of the end-effector in both the camera and haptic frames. Then the rotation and translation between the two systems are calculated as the optimal absolute orientation solution [6]. We carry out the calibration using the SVS system and PHANToM 1.0A model and achieve highly accurate results. The average error in each dimension is less than $0.5mm$.

We implemented example applications of the framework in which the user interacts with a virtual wall or button. We define the wall as a plane with parameter $P = (\vec{n}, d)$ where $\vec{n}$ and d are the normal and the distance of the plane to the origin, respectively. The button is defined as $B = (\vec{p}, \vec{n}, w, h)$ with $\vec{p}$ and $\vec{n}$ specifying the position of the button and the normal of the surface, while w and h indicating the size of the button. To enhance the fidelity of haptic experience, we attach appropriate passive objects to the haptic device, such as a flat metal piece, a computer keyboard key, etc.

We define different interaction properties corresponding to different objects. A simple way to implement this is to define a database of various interaction modes and object surface properties. For example, a button allows a click, and a hard wall only allows a sliding along the surface. For each object we only need specify the index into a database that describes its interaction property.

2.3 Haptic Subsystem

The main task of the haptic subsystem is to simulate the touching experience by presenting suitable force feedback to the user's fingertip. For combined tracking and force feedback, a control scheme is necessary.

2.3.1 Control Law for the Haptic Device

We implement a closed-loop PD control law based on error space to guide the haptic device to a target position which is determined by the world model and current interaction status. For example, if the user is interacting with a virtual plane in space, the target position of the haptic device is the projection of the fingertip onto the plane, i.e., the intersection of the plane and the line that is parallel to the normal of plane and passes through the point corresponding to the current position of the fingertip. The control law is :

$$\ddot{e} + K_v \dot{e} + K_p e = f \text{ where } e = X_d - X \qquad (2)$$

Here X_d and X refer to the desired and current position of the device, respectively. Parameters K_p and K_v are adjusted experimentally to make the system critically or over damped. f is the impedance force applied to control the haptic device. The force f is scaled appropriately for interaction with virtual objects, as described in Section 2.3.3.

To solve the problem of the large difference in frequency between the vision subsystem, which normally runs at no more than 20Hz, and the haptic subsystem, which typically runs around 1KHz, we add a low pass filter on e and $\dot{e}$ to achieve smooth control and to remove high-frequency noise.

$$y = \frac{ax}{s+a} \quad \text{or in time space} \quad y_t = \frac{ax + y_{t-1}}{1+a} \qquad (3)$$

Here y and y_t refer to the filtered result of x, while a is a constant that characterizes the filter. This control law is used in each degree of freedom of the haptic device.

2.3.2 Gravity Compensation for PHANToM

The manufacturer of the PHANToM provides a counter-weight attached to one of the motors that maintains the equilibrium of the device without additional forces. In our experiment, we attach other objects to the endpoint for the purpose of interaction instead of the gimble that is typically counterbalanced. Without additional gravity compensation, the device has a tendency to fall into a degenerate configurations from which it is almost impossible to recover. Thus, we implement a simple gravity compensation scheme. As shown in [1], the following equation gives the motor torques required to counteract the wrench F applied to the manipulator:

$$\tau = {J^b}^T R^T F \text{ or } F = ({J^b}^T R^T)^{-1} \tau \qquad (4)$$

where J^b is the body Jacobian of the manipulator and R is the rotation matrix of the forward kinematics.

We calculate the total torque caused by the gravity of all the parts of the device as:

$$\tau_g = \begin{vmatrix} 0 \\ g(m_a l_1 + 0.5 l_1 m_c + m_{be} l_5) \cos\theta_2 \\ g(0.5 m_a l_2 + m_c l_3 - m_{df} l_6) \sin\theta_3 \end{vmatrix} \qquad (5)$$

The definitions of the variables used in this equation are the same as in [1].

By adjusting m_a in Equation 5 we can calculate the gravity torque τ_g for the device with attached objects. Using Equation 4, the force F_{GC} is computed to compensate for gravity. Combining F_{GC} and weighted f calculated from control law, we are able to achieve smooth and stable trajectory tracking.

$$F = F_{GC} + \Lambda_{gain} f \qquad (6)$$

where Λ_{gain} is the matrix that controls the force gain. When the user is not interacting with any object, Λ_{gain} is the identity matrix.

2.3.3 Interaction with Objects

When the user's finger is touching a virtual or real object, we simulate the interaction forces by adjusting the force gain Λ_{gain} in Equation 6 according to the object properties and interaction mode. For convenience, we define ${}^O\Lambda_{gain}$ for each object in its own coordinate system as this object's gain matrix. A similar transform converts the object's gain matrix to that of the haptic device ${}^H\Lambda_{gain}$.

$${}^H\Lambda_{gain} = {}^H_O R^T \ \ {}^O\Lambda_{gain} \ \ {}^H_O R \qquad (7)$$

where ${}^H_O R$ is the rotation matrix between the frame of the object and the haptic device.

In our current implementation, we define ${}^O\Lambda_{gain}$ as a diagonal matrix with λ_x, λ_y and λ_z referring to its diagonal elements. The z-axis of the object's frame is along the normal of the surface of the object. In our experiments where the user interacts with buttons or planes, we adjust λ_z to simulate interaction force while λ_x and λ_y stay constant. For example, in the case that the object is a solid wall, which allows no piercing along its normal direction, we use a very large λ_z when the fingertip is under the plane. Effectively, this creates a linear spring whose force is proportional to the depth of the finger into the virtual object. When the user's finger enters the object, the haptic device presents a strong force in the direction normal to the surface of the object, in order to push the finger back to the point of contact back on the surface. Figure 4 illustrates the relationship of λ_z to the distance of the fingertip under the plane.

Another example is to push a button or a key on the keypad or keyboard. Similar to the wall, we define the destination point of the haptic device as the center of the surface of the button at the time of initial contact. Once the user pushes the button down and enters the object, we increase λ_z to a proper value to simulate the resistance of the button, until after some point the user triggers the button and feels a lower stiffness. Then we adjust the destination point of the haptic device to the surface of the bottom board of the button and increase λ_z to a much greater value. Thus a much stronger stiffness is felt when the finger pushes the button all the way down to the bottom board. The relationship between λ_z and the depth of the finger into the surface of the button is shown in Figure 4.

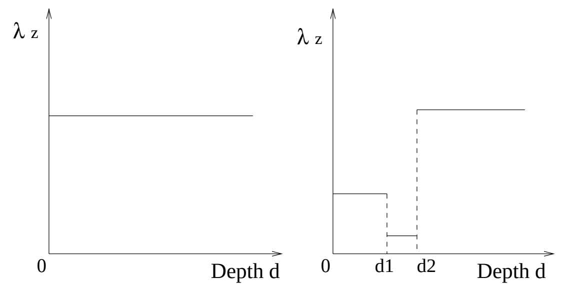

Figure 4: Relationship of force gain λ_z and the depth d of the fingertip under a plane (left) or the surface of a button (right).

A more complex situation is interaction with multiple objects in the same scene. The distance of the fingertip to each object is updated at each frame. The object nearest to the fingertip is chosen to be the current interaction subject. Some methods for approaching this problem are presented in [13].

3 Experimental Results

For foreground segmentation, we use the first 10 frames to learn the appearance model of the background. We build hue histograms of 8 bins for each of the 5×5 image patches. To test the algorithm, we record image pairs of the background and foreground. By comparing the segmentation result and the ground-truth classification image, which is generated by manually marking the foreground part of the scene, we are able to evaluate the scheme. We captured more than 20 pairs of background/foreground images with different background scenes and carried out the experiment on these images. The test set also included 6 pairs of images that undergo illumination changes. As a result, the average correct ratio was 98.16%, with average false positive ratio of 1.55% and false negative ratio of 0.29%.

We set up several interaction scenes and tested the overall performance of the VisHap system. A simple virtual environment consists of a virtual plane in space. The user moves his finger to interact with the plane. Figure 5 shows the relationship of the distance of the fingertip to the plane and the average PHANToM force feedback along the normal to the plane. Note that the force shown here does not include the component to compensate for gravity. It corresponds to component of $\Lambda_{gain} f$ in Equation 6 in the direction of the normal to the plane. Thus, it is the net force that the user feels along the plane's normal. It can be seen that the force feedback matches our model of the plane as shown in Figure 4 very well, i.e., the plane feels like a linear spring.

Figure 5 also shows a more complex case in which the user interacts with a fixed button. When the user is not in contact with the button, the haptic device is stationary at the position of the button. When the fingertip touches the button, force feedback is presented in a manner very similar to the model described in Figure 4, i.e., two distinct linear springs along the normal of the surface of the button models the force feedback before and after the button is triggered. During the stage when the button is triggered, the user feels much smaller resistance. Note that, in actual experiments, the change of force is a gradual process when the button is triggered. This is necessary to ensure stability of the PHANToM device.

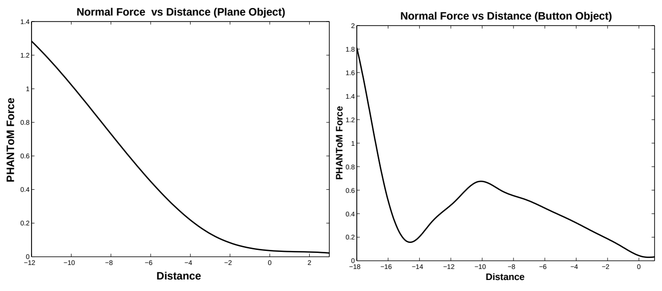

Figure 5: Relationship of haptic force feedback along the normal of the object surface and the distance of the fingertip to the plane or the button. A negative distance indicates that the finger is under the surface.

We also experimented with more complex scenes with multiple objects. The VisHap system is capable of automatically switching interaction subjects according to the scene configuration and current fingertip position.

4 Conclusions

A drawback common to almost all haptic systems is that the user must be attached to the haptic device continuously even though force feedback is not always being rendered. In this paper we present the design and implementation of a haptics-based interaction system that uses finger tracking to overcome this problem and to generate a "complete" haptic experience. We present a modular framework that consists of computer vision, the haptic device and an augmented environment model. The key elements of the implementation of the system were presented. We implemented several example applications on a standard PC and achieved real time performance. The experimental results justify our design and show the flexibility and extensibility of our framework.

Future improvements of the system include incorporating a head mounted display and rendering the scene and the user's hand directly on HMD. The advantage of an HMD is that it can achieve higher immersiveness and fidelity. Another goal is to incorporate richer sets of objects and interaction modes to extend the virtual environment. For example, the user could play a puzzle game by moving tiles around on a board, and a rotating drum could simulate sliding.

Ackowledgments

The authors thank Jake Abbott for his assistance with PHANToM control. The work was supported in part by the Johns Hopkins University and National Science Foundation Grant #EEC-9731478.

References

[1] M. C. Cavusoglu, D. Feygin, and F. Tendick. A critical study of the mechanical and electrical properties of the phantom haptic interface and improements for high performance control. *Presence*, 11(5):555–568, 2002.

[2] G. D. Hager and K. Toyama. Xvision: A portable substrate for real-time vision applications. *Computer Vision and Image Understanding*, 69(1):23–37, 1996.

[3] C. Hardenberg and F. Berard. Bare-hand human-computer interaction. In *Workshop on Perceptive User Interfaces*. ACM Digital Library, November 2001. ISBN 1-58113-448-7.

[4] V. Hayward and M. Cruz-Hernandez. Tactile display device using distributed lateral skin stretch. *Proceedings of the 8th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, ASME IMECE*, DSC-69-2:1309–1314, 2000.

[5] B. Insko, M. Meehan, M. Whitton, and F. Brooks. Passive haptics significantly enhances virtual environments. Technical Report 01-10, Department of Computer Science, UNC Chapel Hill, 2001.

[6] C.-P. Lu, G. D. Hager, and Eric Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000.

[7] T. Massie and J. K. Salisbury. The phantom haptic interface: A device for probing virtual objects. *Proceedings of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, ASME WAM*, 1994.

[8] K. Oka, Y. Sato, and H. Koike. Real-time fingertip tracking and gesture recognition. *IEEE Computer Graphics and Applications*, 22(6):64–71, 2002.

[9] M. Salada, J. E. Colgate, M. Lee, and P. Vishton. Validating a novel approach to rendering fingertip contact sensations. In *Proceedings of the 10th IEEE Virtual Reality Haptics Symposium*, pages 217–224, 2002.

[10] C. R. Wagner, S. J. Lederman, and R. D. Howe. A tactile shape display using rc servomotors. *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 354–355, 2002.

[11] G. Ye, J. Corso, D. Burschka, and G. Hager. Vics: A modular vision-based hci framework. In *Proceedings of the 3rd International Conference on Computer Vision Systems(ICVS 2003)*, pages 257–267, 2003.

[12] Y. Yokokohji. Wysiwyf display: a visual/haptic interface to virtual environment. *PRESENCE, Teleoperators and Virtual Environments*, 8(4):412–434, 1999.

[13] Y. Yokokohji, J. Kinoshita, and T. Yoshikawa. Path planning for encountered-type haptic devices that render multiple objects in 3d space. *Proceedings of IEEE Virtual Reality*, pages 271–278, 2001.

[14] T. Yoshikawa and A. Nagura. A touch/force display system for haptic interface. *Presence*, 10(2):225–235, 2001.