A Haptic Virtual Environment for Industrial Training

Mojtaba Hosseini
DISCOVER Lab, University of Ottawa School of Information Technology and Engineering, Ottawa, Canada
1(613) 5625800-2148
mojtaba@discover.uottawa.ca

Francois Malric
DISCOVER Lab, University of Ottawa School of Information Technology and Engineering, Ottawa, Canada
1(613)5625800-6247
frank@discover.uottawa.ca

Nicolas D. Georganas
DISCOVER Lab, University of Ottawa School of Information Technology and Engineering, Ottawa, Canada
1(613) 5625800-6225
georganas@discover.uottawa.ca

Abstract

This paper presents a haptic virtual environment application for industrial training. A choice of haptic devices (PHANTOM[15] and CyberForce/ CyberGrasp[10]) is used to perform a training exercise with the possibility of sharing virtual scene states with remote users to allow collaboration and remote observation. Furthermore, the application employs video textured avatars to improve face-to-face communication, a novel area of interest management mechanism to reduce network bandwidth usage, standard encoding of state updates for compatibility and is used in conjunction with various projection technologies such as a Head Mounted Display, CAVE configuration or regular PC monitor. The paper serves to underline the various components needed for the development of a haptic virtual environment and our experiences in doing so.

1. Introduction

Virtual Environments (VE) are usually referred to as computer generated 3D worlds wherein user(s) reside and interact with through various computer human interfaces. Haptic Virtual Environments (HVEs) form a subset of these where in addition to the visual feedback, force or touch feedback is also provided to the user through what is usually called a haptic device. HVEs are themselves categorized based on the location of the haptic device and the haptic simulation. In standalone HVEs, both the haptic device and the haptic environment are on the same machine. Example applications of these are training simulations [11], education [16] or virtual prototyping [2]. In distributed HVEs or otherwise known as tele-haptic applications, the haptic device is separate from the environment and remotely affects and manipulates it, examples of which are remote medicine [5] and tele-operation [1]. A special case of distributed HVEs is a collaborative HVE where multiple users each with his/her own haptic device collaboratively manipulate a shared virtual environment [9].

This paper presents a haptic virtual environment for remote industrial training whose discussion serves to highlight the comprehensive set of technologies needed for the development of such applications and the related issues and experiences we encountered in doing so.

Section 2 introduces the particular training scenario addressed in our application followed by section 3 outlining the design of the system. Section 4 presents the details of haptic device interfaces incorporated into the system while section 5 discusses the remaining components of the system. The paper concludes with a discussion of related issues and future work as well as a summary of our collective experiences.

2. Training Scenario

The training exercise addressed in our application involves teaching users how to deal with malfunctioning equipment. More specifically, users are trained to remove a defective control card from an ATM switch card rack and replace it with a functioning card.

The participants start by familiarizing themselves with the environment and the other participants. The trainer walks the trainees through the necessary pre-exercise safety procedures and then proceeds to identify the malfunctioning control card, releases it from the rack, removes it safely and installs the replacement part.

A sense of touch and force feedback is essential to the exercise as are effective modes of communication and interaction. The training can be a stand-alone application where one user goes through the exercise alone or can be distributed whereby remote users are able to observe, learn and comment on the exercise. The application requires high quality visual display of the virtual environment, realistic touch and force feedback, multiple avenues for communication (voice, video, body language) and efficient use of resources to allow for high performance. The next section presents our design of a system enabling the development of a haptic virtual environment with such characteristics.

3. System Design

By virtue of being virtual environments, HVEs share many common components with other virtual environments such as a 3D rendering mechanism for displaying the environment, a scene management mechanism for keeping a consistent copy of the virtual environment and a communication layer for informing remote users of changes in the environment. In order to improve scalability and efficiency of the virtual environment, a filtering mechanism commonly referred to, as Area of Interest Management is needed to limit the distribution of updates to only those users that are interested. In addition, an HVE needs to have an interface between each haptic device and the scene management mechanism. Figure 1 shows the design of our system for developing haptic virtual environments.

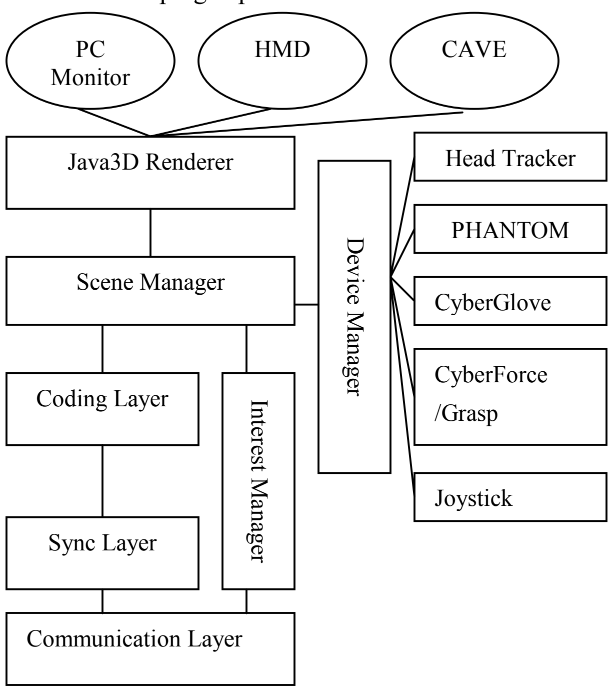

Figure 1 HVE System Design

The core part of the application logic lies in the Scene Manager component where the description of the virtual environment resides. The Renderer (Java3D API in our case) uses the scene description to render it to the display device. In our system, a choice of three different display devices is possible: a regular PC monitor, a Head Mounted Display or using three screens in a CAVE [3] configuration. Java3D accommodates all three displays as well as generating stereoscopic views without a need for extensive programming on the part of the developer. Haptic devices and trackers interface with the Scene Manager through a Device Manager. The next section will elaborate on the haptic devices and the interfaces involved but at the moment it suffices to say that the Scene Manager shares a part of the virtual scene with the haptic devices and the interfaces allow both to maintain a consistent version of it. Better scalability and efficiency is achieved through the use of a novel interest management scheme and standardized coding is used for state updates as well as scene description through an MPEG-4 based framework as will be discussed later. Though not apparent from the figure above, the system allows for the integration of video media inside the 3D environment for an extra avenue of communication. Figure 2 shows one possible configuration of the system where a PHANTOM device is used for haptic feedback, a CyberGlove for finger tracking, PC monitor for display and a video stream to represent the head of the avatar. The next section presents the details of the haptic devices used as well as the interfaces involved.

Figure 2. An example configuration of the system

4. Haptic devices and their interfaces

The first step in introducing haptics into the virtual environment was to incorporate a PHANTOM Premium 1.0 [15] device into the application. It would allow for the trainee to feel the appropriate force needed to unclip and pull out the defective control card, reflecting the reality and underlining the important aspects of the manipulation. This procedure on real equipment requires that an appropriate and not excessive level of force be applied to remove the card, otherwise, damage could occur. Also the card must be unclipped first in order for it to be removed.

The incorporation of the PHANTOM involved writing a simple Java Native Interface (JNI) to the C-libraries to allow for the loading of geometry into the haptic scene as well as the communication between the haptic device and the Java application. The limited operation space of the PHANTOM (a 13cm×18cm×25cm cube) meant that only a small part of the entire scene could be loaded into the haptic scene and therefore felt by the user. Figure 3 below shows the subsection of the switch that was loaded in our particular application.

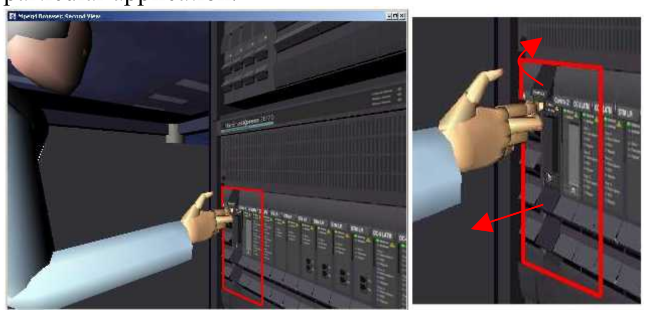

Figure 3. The subsection of the virtual world with haptic feedback (left) and procedure for removing card (right)

The Java application retrieves the position of the PHANTOM device every 50ms and moves the right hand of the avatar accordingly. In fact, the user is given the impression that he/she controls the tip of the index finger of the avatar and must use it to unclip the card by pushing it up and then remove the card by pulling on it as shown in Figure 3.

Compared to our first prototype [6], which did not include any haptic devices, the inclusion of the PHANTOM greatly enhanced the interactivity as well as the realism of the training exercise. The ability to feel the resistance of the clip as it is being pushed open and the weight and inertia of the card as it is pulled out of the rack took the application one step closer to working with the physical equipment. Although greatly enhancing the training exercise by adding haptic feedback, the PHANTOM was found to be limited in several aspects however.

Other than the limited operation space of the PHANTOM already mentioned, the fact that this device only provides one point of interaction with the virtual world was the most limiting factor. The act of grabbing a card proved most unnatural given only one point of interaction especially for first-time users of the system, used to manipulating the physical world with two hands and millions of sensory points.

The particular PHANTOM device used being a 3DOF output device (no feedback on rotation), limited the training exercise further. For example attempting to rotate the card while it is inside the rack should be met with a resistance force (generated by the adjacent cards), however there are no actuators on the PHANTOM device to provide such resistance to rotational movements. The maximum exertable force of 8.5N can also be limiting when dealing with rather heavy equipment.

In order to address these limitations, the PHANTOM device was replaced by a CyberGlove-CyberGrasp-CyberForce setup [10]. The CyberGlove tracks the finger movements of the user while the CyberGrasp provides feedback on the five fingers by pulling on them and the CyberForce restricts the movements of the hand itself. Figure 4 illustrates the setup.

Figure 4. Inclusion of CyberGlove, CyberGrasp, CyberForce and an HMD into the application

In order to enhance the experience, an HMD is used for display and an Ascension miniBird for tracking of the head.

By providing 6 points of haptic interaction with the virtual world (5 for each finger and one for the hand), the training's realism is increased significantly as compared to the one using the PHANTOM. Users can simulate the grabbing motion of the card as they would in the physical world and feel the weight of the card when they hold it.

Being able to feel the shape of the various objects as well as their weight allows users to gain an expectation of how the equipment will behave in the real physical world. The use of the HMD also contributed to the sense of immersion by providing the user with a natural way of looking around and focusing on the exercise.

Though an improvement on the previous system, this setup still had significant limitations.

Once again as compared to the physical world where millions of sensors in the skin provide a very rich experience of object manipulation, haptic feedback on only 5 points (the tip of each finger) and no touch feedback (to feel the texture and temperature of the object) still falls short of expectations. A related problem is that fine manipulation and feeling of small objects is not possible in this setup (for example a screw in the card is not easily felt using the CyberGrasp). The inclusion of tactile feedback devices could assist in this regard.

As with the PHANTOM setup, writing a Java Native Interface (JNI) to the provided C libraries was necessary to incorporate the use of these devices into our java framework. The haptic simulation was simplified to modeling the area of interest only (the card needing repair and it's neighboring cards). Throughout the simulations, all card geometries in the area of interest can be felt with any of the 6 points of feedback. The simulation of the unclipping of the card is enabled at first, providing the sensation of a clip, held by one finger, rotating with some resistance around its hinge. Once completed, this haptic simulation is replaced by one modeling the removal of the card from the rack, restraining the lateral movements while grabbing and pulling the card.

Although the operation space of the CyberForce (12" x 12" square swept through 133 degrees with radius of 19.5") is greater than that of the PHANTOM (thus allowing for the inclusion of a greater area of the virtual environment in the haptic scene), the immobility of the device is a significant limitation. Being attached to the CyberForce prevents the user from walking around the room freely, forcing another form of navigation (mouse, keyboard or joystick) and hence takes away from the sense of immersion. Also, the ability to move equipment from one place to another (for example transferring a card from the rack to the repair table) is difficult given the operation space of the equipment.

The fact that the CyberForce is a 3DOF output device (no feedback on rotation), resulted in the same limitation as that already mentioned with the PHANTOM: mainly that the user could not be prevented from rotating the card inside the rack though this should not be physically possible.

The weight of the CyberForce and CyberGrasp also made training exercises of long duration cumbersome and tiring for the user. Combined with the setup and calibration procedure that must be done for every user, the usability of this system becomes questionable.

Another usability issue is that since the right hand is attached to the CyberForce/CyberGrasp, it is no longer available for directing navigation (using mouse, keyboard or joystick) and since the setup is immobile, the user cannot freely navigate the virtual world by walking in the physical world (using the HMD's tracker for getting the position of the user). Therefore other forms of navigation must be made available to the user (such as using hand gestures to navigate). In our case, the user navigates with his/her left hand using a joystick which requires some initial orientation and getting used to. A much better approach is to develop mobile wearable haptic devices such as in [12] so that the tracked movements of the user wearing an HMD would be shown in the virtual world while the haptic device would easily follow him/her.

A limitation of the software interface provided with the haptic device package was the absence of an inter-object physics-based collision reaction engine (though collision detection is provided). This implies that the developer must program the physical responses resulting from the collision of objects together in terms of the appropriate forces that must be exerted by the CyberForce on the user hand. For example if the user picks up an object and that object collides with another, the developer must calculate the resulting force on the hand as shown in Figure 5 below.

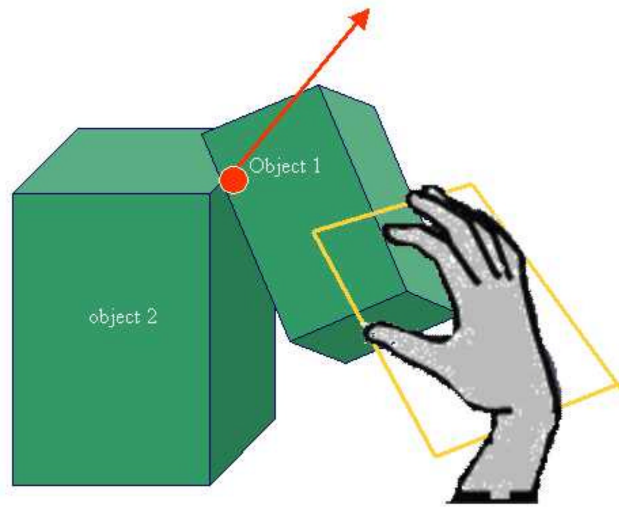

Figure 5. Inter-object collision and haptic feedback

Such calculations would depend on the collision point/surface, center of gravity of object, position of the user hand, etc. The inclusion of a more complete collision engine in the haptic device's software interface would greatly reduce the amount of work an HVE developer would have to undertake to integrate the device into his/her application.

It should be mentioned at this point that in both systems (using PHANTOM or CyberForce), since the position of the avatar's hand is that of the haptic device and if there are no other trackers determining the position of the elbow and shoulder of the user, inverse kinematics are necessary to calculate the appropriate elbow and shoulder rotations in order keep the hand of the avatar attached to the body.

So far we have only discussed the issues related to the stand-alone version of our training prototype. One requirement of our application however was for remote users to be able to observe and comment on the exercise as it takes place with the future extension of allowing remote users to also haptically manipulate the shared virtual world. In order to achieve this the applications mentioned so far were built on top of our framework facilitating the development of distributed virtual environments called COSMOS [4][8]. COSMOS is a framework based on the MPEG-4 standard [13] written in Java and implementing a VRML-to-Java3D loader, a VRML-to-MPEG-4 encoder and a Java3D/MPEG-4 compositor/decoder. This framework allows for loading of 3D worlds from VRML and rendering them in Java3D, encoding/decoding of the scene in MPEG-4's Binary Format for Scenes (BIFS) as well as the encoding and decoding of update messages representing changes to the scene. Another feature of

MPEG-4 and hence COSMOS is the ability to integrate various media types (video inside 3D worlds for example) into the scene as well as providing a layer for synchronization of the various media streams.
Space restrictions prohibit us from presenting the details of the framework but it suffices to say that having developed the stand-alone version of our prototype (be it with or without haptic devices), it was then easy to encode and transmit the virtual world as well as all updates to the scene to remote clients using the COSMOS framework. There are only two issues worth mentioning.
Though developing distributed HVEs is very similar to developing other distributed VEs, the large number of frequent but small in size updates that result from tracking of various body parts (head, hand and fingers in our case), necessitates the packing of these updates into one message to reduce the overhead of transmitting them individually. In other words, the communication layer must combine the updates to the user's avatar at a given point in time into one update, otherwise both the sender and receiver won't be able to communicate with acceptable performance.
The other issue is that the additional modes of interaction provided by haptic devices demand more realistic avatars than those used in other VEs. At the very least, each avatar must have a detailed hand geometry to represent the hand gestured captured by the CyberGlove for instance. In order to increase the sense of presence of a remote user in the same virtual world, we found articulated avatars not to be adequate enough and therefore moved towards video textured avatars as will be discussed in the next section.
Collaborative HVEs (where more than one user has haptic devices and they collaboratively manipulate a shared object), present further issues to overcome, the most important one of which is dealing with delay and delay jitter between the remote sites. The need for a delay compensation technique as well as Quality of Service (QoS) guarantees with respect to delay and especially jitter become essential for such applications. The delay can be compensated through predictive techniques and delay jitter must be bounded since it has been shown that jitter is a more significant factor in making the completion of a collaborative task between two remote users more difficult [14]. These are some of the issues we hope to investigate in our future work.

5. Other system components

In order to provide a better avenue of communication between the user performing the training exercise and remote users, his/her video is captured and shown on the remote user's machine as the head of the avatar as shown in Figure 1. Video avatars allow for remote users to gain confidence in the presence of a human as opposed to a machine controlled avatar or 'bot'. The video textured head of the avatar is made possible through the use of the Java Media Framework for capture and transmission of the video and its integration with Java3D to render it as changing texture in the 3D scene.
The use of video textured avatars in distributed VEs in turn brings about the question of scalability. Receiving video streams as well as tracking updates from every participant in the distributed exercise may require more bandwidth than is available. It is therefore necessary to filter out some of these streams based on the interest of individual users or in other words employ an Area of Interest Management (AoIM) scheme. The topic of AoIM has been a focus of research in the distributed VE community especially with regards to large-scale VEs. We have, however, developed a novel interest management scheme more suited to collaborative VEs employing video textured avatars and tracking equipment [7]. The scope of this paper does not include a discussion of our AoIM technique but it is worth mentioning that developers of advanced distributed HVEs will have to include some sort of AoIM scheme to improve the scalability or even the feasibility of their application.

6. Related issues

Developers of HVE applications must delve into a myriad of issues ranging from haptic device interfaces and device limitations to psychology and usability studies as well as more conventional issues such communication delay and jitter, multimedia integration, coding and synchronization and 3D graphics. So far we have discussed some of the specific issues we have encountered in developing an industrial training HVE using specific haptic devices.

We now turn to more generic issues involved in the development of HVE applications. As can be deduced from the discussion in section 4, we encountered several limitations with current haptic device technologies, leading us to the impression that today's haptic technology is still in the early stages of commercial development and much more needs to be done in developing better force feedback devices as well as their integration with touch and temperature feedback devices in order to have a truly realistic touch and force feedback from the virtual environment.

Though the CyberGrasp is wearable, neither the PHANTOM nor the CyberForce are wearable or even mobile, making them inappropriate for use inside a CAVE or other immersive environments. There is therefore a need for lighter, less cumbersome and wearable haptic devices.

Finally, the possibility of using augmented/mixed reality settings instead of purely virtual reality settings for related application is of interest. In the case of our application for instance, using a mock physical switch and card but overlaid with computer graphics to complete the details would possibly eliminate the need for the intricacies of the currently used haptic devices to simulate the entire switch haptically. In this regard, mixed or augmented reality could present a compromise between real world training and pure virtual reality training.

7. Conclusion

This paper presents a series of Haptic Virtual Environments used for an industrial training application built on a generic framework for developing distributed virtual environments and making use of various commercial haptic devices. The paper serves as a discussion of the various haptic-related issues encountered in developing these prototypes as well as a presentation of the multitude of components required to develop advanced distributed HVEs.

8. Acknowledgements

This research was supported in part by the National Capital Institute of Telecommunications (NCIT) and by the Canada Research Chairs Program.

9. References

[1] H. Baier, M. Buss, F. Freyberger, J. Hoogen, P. Kammermeier, G. Schmidt, "Distributed PC-Based Haptic, Visual and Acoustic Telepresence System- Experiments in Virtual and Remote Environments". *Proceedings of IEEE VR'99*, Houston, Texas, 1999.

[2] P. Buttolo, P. Stewart, A. Marsan, "A haptic hybrid controller for virtual prototyping of vehicle mechanisms", *Proceedings of 10th IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Orlando, Florida, 2002, pg. 249-254.

[3] C. Cruz-Neira, D.J. Sandin, T.A. DeFanti, "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE," *Computer Graphics (Proceedings of SIGGRAPH '93)*, ACM SIGGRAPH, Anaheim, California, 1993, pp. 135-142.

[4] V. Darlagiannis and N.D. Georganas, "Virtual Collaboration and Media Sharing using COSMOS,", *Proceedings of 4th World Multi-conference on Circuits, Systems, Communications & Computers (CSCC 2000)*, Athens, Greece, 2000.

[5] A. Guerraz, B. Hennion, P. Thorel, F. Pellissier, A. Vienne, I. Belghit, "A haptic command station for remote ultrasound examinations", *Proceedings of 10th IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Orlando, Florida, 2002, pg. 88-96.

[6] M. Hosseini, N.D.Georganas, "Collaborative Virtual Environments for Training", *Proceedings of ACM Multimedia 2001*, Ottawa, Canada, 2001, pp.621-622.

[7] M. Hosseini, S. Pettifer and N.D. Georganas, "Visibility Based Interest Management in Collaborative Virtual Environments", *Proceedings of ACM conference on Collaborative Virtual Environments (CVE 2002)*, Bonn, Germany, 2002, pg. 143-144.

[8] M. Hosseini and N. D. Georganas, "Suitability of MPEG4's BIFS for Development of Collaborative Virtual Environments", IEEE International Workshop on Enabling Technologies for Collaborative Enterprises (WET ICE'01), Boston, Massachusetts, 2001.

[9] R. Hubbold, "Collaborative Stretcher Carrying: A Case Study", *Proceedings of 2002 EUROGRAPHICS workshop on Virtual Environments*, Barcelona, Spain, 2002.

[10] Immersion Corporation, http://www.immersion.com/products/3d/interaction/hardware.shtml

[11] J. Kim, S. De, M.A. Srinivasan, "Computationally efficient techniques for real time surgical simulation with force feedback", *Proceedings of 10th IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Orlando, Florida, 2002, pg. 51-57.

[12] T. Mitsuda, S. Kuge, M. Wakabayashi, S. Kawamura, "Wearable Haptic Display by the Use of a Particle Mechanical Constraint", *Proceedings of 10th IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Orlando, Florida, 2002, 153-158.

[13] MPEG-4 Systems: ISO/IEC 14496-1, "Information Technology – Coding of audio-visual objects, Part 1: Systems", January 1999.

[14] K.S, Park, R., Kenyon, "Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment", *Proceedings of IEEE conference on Virtual Reality (VR'99)*, Houston, Texas, 1999.

[15] SensAble Technologies Inc., http://www.sensable.com/haptics/products/phantom.html

[16] J. Solis, C.A. Avizzano, M. Bergamasco "Teaching to write Japanese characters using an haptic interface", *Proceedings of 10th IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Orlando, Florida, 2002, pg. 255-261.