Development of a Hand Device Emulation System for Realizing a Virtual Environment on a Personal Computer by using it3d and Java3D

Akimitsu Hashimoto[1], Hiroshi Nakano[2], Noritaka Osawa[3], Shinichi Orita[4] and Hidenori Akiyama[1]

[1]*Graduate School of Science and Technology, Kumamoto University, 2-39-1 Kurokami, Kumamoto 860-8555, Japan*
[2]*Center for Multimedia and Information Technologies, Kumamoto University, 2-39-1 Kurokami, Kumamoto 860-8555, Japan*
[3]*National Institute of Multimedia Education, 2-12 Wakaba, Mihama-ku, Chiba 261-0014, Japan*
[4]*Graduate School of Human Informatics, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan*

hashi@cae.cc.kumamoto-u.ac.jp

Abstract

We are developing a hand device emulation system for realizing a virtual environment on a generic personal computer by using it3d and Java3D. It3d is an interactive toolkit library for developing 3D applications, which utilizes virtual reality technologies. It was implemented by using the Java language and the Java3D class library to enhance its portability. Manipulation devices, such as a hand device, are necessary for treating 3D objects (3D menus, buttons and so on) given by it3d library. Such manipulation devices are rather special and too expensive for personal use. It is therefore not realistic for online education in virtual environments where learners are studying on their PCs at home. This is the motive force to develop a hand device emulation system. The system enables 3D applications for virtual environment to run on generic PCs without any modifications. The hand's position, gradient and limited shapes of fingers can be controlled even only with mouse actions; click, double-click, scroll, drag and so on. It is not necessary for the hand device emulation system to run on the same PCs where the 3D application is running, because our system uses the distributed network functions provided by it3d.

1. Introduction

The Internet speed, the performance of the personal computers (PCs), and the Internet accessibility at home have been improved in recent years. Specially, three-dimensional (3D) graphics performance of PCs is being enhanced dramatically and the cost is also going down by the development of multimedia technology. It is therefore becoming possible to run virtual reality (VR) applications on generic PCs.

We are studying 3D virtual experiments on web browsers[1,2] by the use of Java[3] and Java3D application programming interface (API)[4]. These applications are portable and runnable on many browsers and operating systems because of using the Java and Java3D API.

We adapted some of the applications for VR[5] by the use of it3d[6] (Interactive Toolkit library for 3D applications), and they can run on the immersive VR system named TEELeX[7]. TEELeX is an abbreviation for Tele-Existence Environment for Learning eXploration developed by National Institute of Multimedia Education (NIME) in Japan and the origin is the CAVE[8] developed by University of Illinois.

Using TEELeX, learners can face an experience of full-scale and real-time experiments with 3D user interfaces in the virtual environment, however, such a large-scale virtual environment systems are not easy to access usually. We thought that much simpler home VR systems should be necessary for applying VR for the distance education.

VR systems are generally consisted of a stereo display(s) and input/output devices. The stereo view can be provided by some kinds of Red/Blue 3D glasses, stereo shutter glasses, stereo LCD displays and so on. Some of them, for example the Red/Blue glasses and the stereo shutter glasses, are not expensive for home-use. The most important one of the input/output devices for the user interface in VR might be a manipulation device. Nevertheless, the manipulation devices, such as a glove

device, can not be prepared easily at home because they are rather special and too expensive. We therefore decided to develop the hand device emulation system.

2. Development of the emulator

This section describes the program which constitutes the emulator, and its role. The developed emulator assumes that client programs are using it3d[6] as a graphical user interface (GUI) library for virtual environment. It3d is an abbreviation for Interactive Toolkit library for developing 3D applications. It3d has a 3D widget library for multimodal interfacing, and an interaction-recognition library. The input/output library for distributed devices has a uniform programming interface style for various types of devices, and it supports position/orientation trackers and the sensor gloves which are used by our experiment system. The library utilizes multicast peer-to-peer communication between devices and drawing clients and it reduces load of clients. It3d is implemented by using the Java language and the Java 3D class library to enhance its portability.

The developed classes for the emulator consist of two groups. One is the group of classes which were created from the original it3d classes with some modifications, where the source codes of it3d are open in its web site[6]. Another group of classes are almost newly created for the emulator. Each of the two groups is explained in the next two subsections.

2.1. The classes modified from it3d

Table 1 shows the list of the classes created from the original it3d classes with some modifications. The original classes contain native methods for accessing input/output devices through dynamic link libraries (DLLs), and these methods are completely cleaned out and some of them are replaced with emulated methods. The original classes work only on Microsoft Windows system, nevertheless, the new classes can work on any systems which support Java and Java3D API keeping compatibility with the original classes.

2.2. The newly created class

It3d provides the TrackerDeviceManager class for the tracker device which detects the position of the glove device, and also provide the GloveDeviceManager class for recognizing the direction and shapes of the glove. These classes are originally designed for the standalone application which has loops checking the real devices, however, they should be runnable as threads in the emulator program because these classes are not checking real devices but must respond to the mouse event. These two original classes are rewritten and the new classes (TrackerDeviceManagerEmulator and GloveDeviceManagerEmulator) are created without compatibility, nevertheless, they can construct emulator program easily. These two classes keep the original distributed network functions. It is therefore not necessary for the emulation system to run on the same PCs where the 3D application is running.

2.3. The emulator program

The GUI parts of the emulator system consist of three programs named EmuItConstants, HandControl, and EmuIt3D classes.

By using the emulation system, when the application program which needs a glove device is running, a hand is displayed on a screen and can be moved according to the mouse, even if there are no real glove devices. It is necessary to recognize the position, direction and the form

Table 1. The classes and their original function

source code for the class (original name in it3d)	Function
DeviceAddressManagerEmulator.java (DeviceAddressManager.java)	Distributed device library
DeviceAddressServantEmulator.java (DeviceAddressServant.java)	Management of an address
GloveDeviceManagerEmulator.java (GloveDeviceManager. Java)	Recognition of a direction and form of a glove device
TrackerDeviceManagerEmulator.java (TrackerDeviceManager.java)	Recognition of a position of a glove device
Vht6DofDeviceEmulator.java (Vht6DofDevice.java)	Support of a 6 degree of freedom device
VhtCyberGloveEmulator.java (VhtCyberGlove.java)	Management of a sensor of a finger joint and an angle to bend
VhtDeviceEmulator.java (VhtDevice.java)	Indication of connection information
VhtGloveEmulator.java (VhtGlove.java)	Get an angle of a finger joint
VhtHumanFingerEmulator.java (VhtHumanFinger.java)	A class group for other Virtual Hand Toolkit (finger connection)
VhtHumanHandEmulator.java (VhtHumanHand.java)	A class group for other Virtual Hand Toolkit
VhtIOConnEmulator.java (VhtIOConn.java)	Setting of connection with device manager
VhtPhalanxEmulator.java (VhtPhalanx.java)	The acquisition of a transform of a finger joint
VhtTrackerDataEmulator.java (VhtTrackerData.java)	Data of tracker
VhtTrackerEmulator.java (VhtTracker.java)	Support of 3 dimensions of tracker device
VhtTransform3DEmulator.java (VhtTransform3D.java)	The acquisition of shape data of a glove

of the glove device in the real system, however, such information cannot be acquired in the emulation system. We therefore keep the fundamental data of five fingers in the EmuItConstants class. We simplified the finger state to only two types; opened and closed states. There are two hands (left and right hands), and each hand has five fingers, and each finger has three joints, and each joint has two states. We therefore have sixty states in total. Each states are described in 4 X 4 matrix stored in a Transform3D object of Java3D. The Transform3D not only has a matrix but also include such operations as rotation, scaling, translation and many useful methods. The Transform3D objects were obtained by these methods, for example, setRotation(AxisAngle4d a1) method which sets the rotational component using the rotation of angle about the vector; AxisAngle4d object a1.

The HandControl class is a program for receiving mouse events and changing the hand states though TrackerDeviceManagerEmulator and GrloveDevice ManagerEmulator classes using the parameters in EmuItConstants class. The most important point in this emulator is the correspondence between mouse actions and hand responses. We investigated many times and we decided the best set of mouse actions to treat 3D widgets in the VR space as shown in Table 2.

The HandControl class has additionally radio buttons for selection of hand shapes; close, open and index, and has sliders for setting the position by the rectangular coordinates.

The EmuIt3D class has a main method and runs as a main program of the emulator system which implements EmuItConstants and HandControl classes.

3. Experiments

The emulator program must run first as a following command.

```
% java EmuIt3D
```

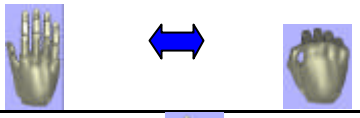
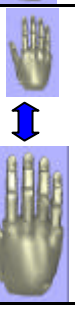
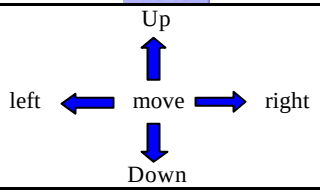
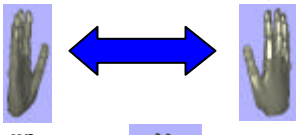
After several seconds for initializing network functions, the emulator window is shown as Fig.1. At this time, there are no applications using the it3d library, the hand is not displayed on a screen.

Now, since the emulator is ready, a test application can be started ether on the same PC or on the other PCs in the same segment of the network or in the network area where the multicast protocol is reachable. Here, for example, we selected program named LRHandTest distributed with it3d and wake up it by the following command.

```
% java LRHandTest
```

It is an application to test whether three balls displayed on a screen can be caught by the right hand and the left hand. When an application of LRHandTest is started, it begins looking for a tracker device and a glove device using the Remote Method Invocation (RMI) and the multicast communication.

Table 2. Mouse actions

mouse action	hand response
click	Open  Close
double click	Index 
scroll	far  near
left button drag	Up  left move right Down
center button drag	the same as scroll
right button drag	Left roll right  up  Roll Down 

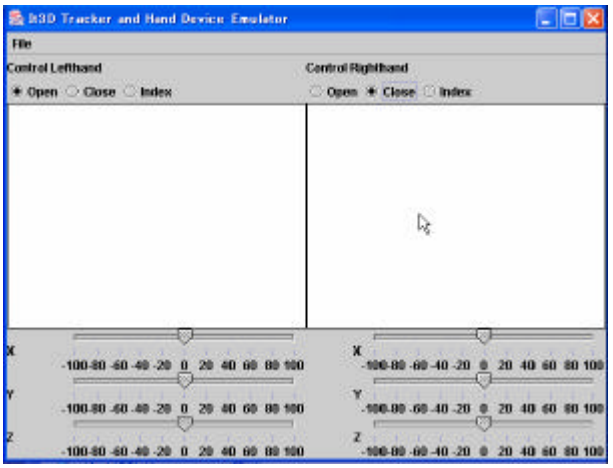


Figure 1. The window of an emulator

If they could be discovered, a hand position and direction are recognized. Because there are no real glove devices, information of the position, the direction and the shape set as initial values by the emulation program. After the left and right hands are displayed sequentially, three colored balls are displayed. Each ball has a different function; "holdable only by the left hand", "holdable by either the right or left hand", and "holdable only by the right hand" respectively. All of the functions can be achieved without any difficulty by mouse actions as shown in Figs.2 and 3. The emulator is working perfectly.

4. Conclusion

We developed the hand device emulation system for realizing a virtual environment on a generic personal computer by using it3d and Java3D. The emulation system enables 3D applications for virtual environment to run on generic PCs without any modifications. The hand's position, gradient and limited shapes of fingers can be controlled only by mouse actions; Click, double-click, scroll, drag and so on. It is not necessary for the hand device emulation system to run on the same PCs where the 3D application is running, because the system uses the distributed network functions provided by it3d. The emulation system can run on any system which supports Java and Java3D API, because the native methods are completely removed from the original sources. The usability has been checked by the experiments and all of the functions can be achieved without any difficulty only by the mouse actions.

We are planning to improve the system by the following ways. New hand forms should be added to three forms; "open", "close", "index". Two mouses should be used simultaneously for left and right hands. We will also try to use USB cameras as input devices by recognizing the position information.

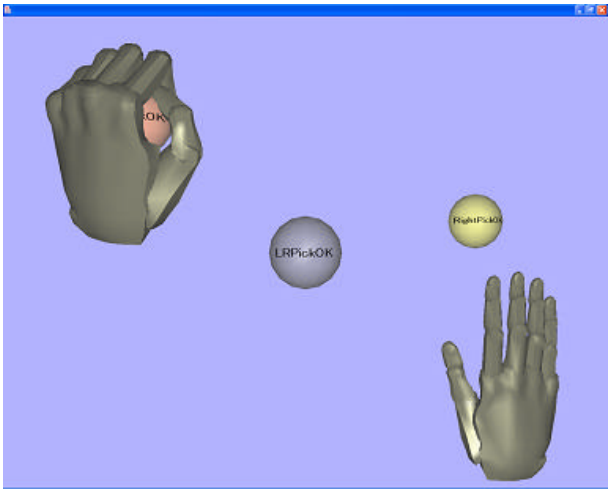


Figure 2. A ball is held and moved by the left hand.

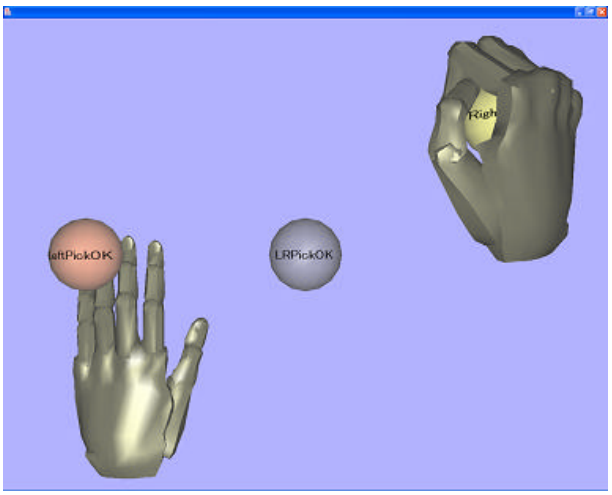


Figure 3. A ball is held and moved by the right hand.

5. References

[1] Y. Nakamura and H. Nakano, "Development of 3D simulation programs for classical mechanics - Using Java 3D -", *ICCE/ICCAE 2000, Taipei, Taiwan*,

Vol.2, 2000, pp.1602-1603.

[2] H. Nakano, K. Tokunaga and Y. Nakamura, "Visualization Effect on Virtual Experiments in Dynamics", *Proceedings Information Technology Based Higher Education and Training (ITEHT2002), Budapest, Hungary, July 7-9*, 2002.

[3] M. Campione, K. Walrath, A. Huml and Tutuorial Team, *The Java Tutorial Continued: The Rest of the JDK*, Addison-Wesley, 1998.

[4] H. Sowizral, K. Rushforth and M. Deering, *The Java™ API Specification*, Addison-Wesley,1998. http://java.sun.com/products/java-media/3D/

[5] H. Nakano, K. Tokunaga, N. Osawa and H. Akiyama, "Full-scale and Real-time Virtual Experiments in Dynamics by using an Immersive Projection Display and Hand Manipulation", *Proceedings Information Technology Based Higher Education and Training (ITEHT2003), Marrakech, MOROCCO, July 7-9*, 2003, pp.184-189.

[6] N. Osawa, K. Asai and F. Saito, "An Interactive Toolkit Library for 3D Applications: it3d", *Proceedings Eighth Eurographics Workshop on Virtual Environments (EGVE2002)*, 2002, pp.149-157. http://www.nime.ac.jp/it3d/

[7] K. Asai, N. Osawa, and Y. Y. Sugimoto, "Virtual Environment System on Distance Education", *Proceedings EUROMEDIA'99*, 1999, pp.242-246. http://www.nime.ac.jp/teelex-w/index-e.html

[8] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround- screen projection-based virtual reality: The design and implementation of the CAVE",*Proceedings ACM SIGGRAPH '93*, 1993, pp.135-142. http://www.evl.uic.edu/pape/CAVE/