

A Design and Implementation of Immersive Virtual Space Authoring System

Jeongdan Choi, Chansu Lee, Kwangman Oh, Chanjong Park
VR Lab., ETRI-CSTL
{jdchoi, chanslee, okman, cjpark}@etri.re.kr

ABSTRACT

This paper describes an immersive virtual space authoring system to help developing the VR applications. Our system supporting the immersive user interface on the desk-top is composed of VR hardware controlling software modules, 3D authoring software modules and VR I/O devices such as CyberGlove and HMD. We also develop the immersion toolkit to support high-level programming for VRML object manipulation. This paper suggests an extended VRML format that allows preview of 3D VRML objects prior to load them. It is useful to support drag-and-drop authoring interface while immersed, and to reduce the retrieval cost of ultimate 3D VRML data on the Internet.

1. INTRODUCTION

Recent technologies of computer graphics enable that even the end user is able to manipulate the 3D objects as well as 2D objects on the desktop environment. Moreover, the VRML (Virtual Reality Markup Language) that is the standard 3D graphics file format[1][2] for the Internet and World Wide Web, opens the way to getting and distribution the 3D objects through the Internet.

Now, the VRML browsers and authoring tools are proliferating, and the content-based information is becoming increasingly importance for various areas such as the digital library and the interactive dataware house. Also, many authors who want to create the virtual space need more and more convenient interface and powerful authoring functions. We can use the 3D devices such as CyberGlove and HMD (Head Mounted Display) at a price and install on the desktop platform. These devices give the immersion to the author[3][4][5][6].

In general, the VE (Virtual Environment) application should be provide the simulation and the navigation in the virtual space[7]. It requires the precise controlling of the spatial input device and the processing of a number of commands for object loading, object placement, property specification and animation in the immersive environment[8]. Until now there are a few high-level programming tools for application developers importing VRML. In order to help the programmers to make the VE application, we develop the immersion toolkit that includes software functions for virtual space authoring and its APIs in this paper. It also provides higher level of abstraction than that offered by low-level graphics libraries to the VE application programmers.

While authoring, the author wants to get the 3D objects through the Internet to create the virtual space. But, it is hard to collect these objects because that many users are connected to the Internet using the low-speed modem while the 3D object data has extensive volume size[9]. We suggest an extended VRML to allow previewing prior to load the entire VRML object, and develop the VRML browser supporting this.

In this paper, we develop an immersive authoring system using 3D devices such as CyberGlove and HMD to support realistic authoring works. This paper laid out as follows: in Section 2, we introduce the related studies for VR authoring system. In Section 3, we describe our immersive virtual space authoring system in detail. In Section 4, we present the results of implementation. Finally, we conclude this paper and suggest future studies in Section 5.

2. RELATED STUDIES

The promise ERGO which is studied by Deneb[10] is software for analyzing bio-engineering aspects by implementing posture and movement of a human being naturally in the 3-dimensional work space by using macro. A fast prototyping system may be implemented by offering inverse kinematics. Additionally, using immersive virtual reality devices develops simulation analysis tools and training systems of various types. Telepresence Research Institute developed a virtual experience system called Menagerie[11]. This is a system in which interactions are exchanged by wearing a stereo color viewer (FAKESPACE BOOM-2C Viewer) in the virtual space in which animal models and 3-dimensional voice (Crystal River Engineering Beachtron) are reproduced. Immersive Systems Company developed a software package called Meme[12]. The company manufactures virtual space application programs which support multiple participants. This program is advantageous in that users wear HMD and DataGloves and participate in the virtual space through network, and a platform-independent system is developed. MechDyne Company developed a

virtual reality system supporting the surround screen VR[13]. The company develops simulators for the ocean, air, and land. Trimension system Company developed a system called ReACTor[14] in which CrystalEyes lenses are worn and virtual images which are projected on all wall surfaces are viewed through the stereo view.

One of immersive authoring tool systems thus developed is SmartScene of MultiGen Company[15]. This system provides both-hand interface using PinchGloves or CyberGloves, and increased feeling for reality in the visual programming method called LynX by using a modeling tool called Vega. Also, dVISE of Division Company[16] supports 3-dimensional tool foxes or menus, and provides a function for facilitating implementation of the 3-dimensional immersive system.

An object of the present study is to implement the virtual environment desired by users without using the programming technology but by using data of already existing models and changing attributes or re-arranging positions. However, it has been disadvantageous in that it takes a great deal of time and efforts since lower-rank structures are developed first based on the graphic library in implementing the virtual reality application programs. In order to remove the above disadvantages, the class library was implemented by observing the VRML2.0 standard and expanding necessary functions. Then the prototype system of the immersive authoring tool was developed based on the above.

3. IMMERSIVE AUTHORING SYSTEM

3.1 Software Modules

The software modules in our system are classified into the HW devices interface modules, the authoring function modules and the immersion toolkit.

The HW interface modules such as 3D input module and 3D display module mainly support the interface with 3D input/output devices, while the authoring function modules provide the general authoring functions.

- 3D input module: This module processes the received 3D data from VR input device (e.g, CyberGlove). In other words, it generates the authoring command by the angles among finger joints and widths among each finger.
- 3D display module: This module divides the received scene from the channel converter to two channels in order to make the stereo graphic, then sends out to the VR output device (e.g, HMD).
- Authoring management module: This module provides the general authoring functions such as data loading, data storing, data manipulating, property specifying and data positioning.
- Icon viewer: This module collects VRML objects from the Internet, then previews the geometry information of collected objects using our suggested the extended VRML format.
- Immersion toolkit: This supports visual authoring functions in an extensible, native VRML2.0 environment based on IRIS Performer. It provides APIs of VRML object manipulating functions such as rotation, stretching, scaling, combining and multi-channel multi-processing if hardware supported.

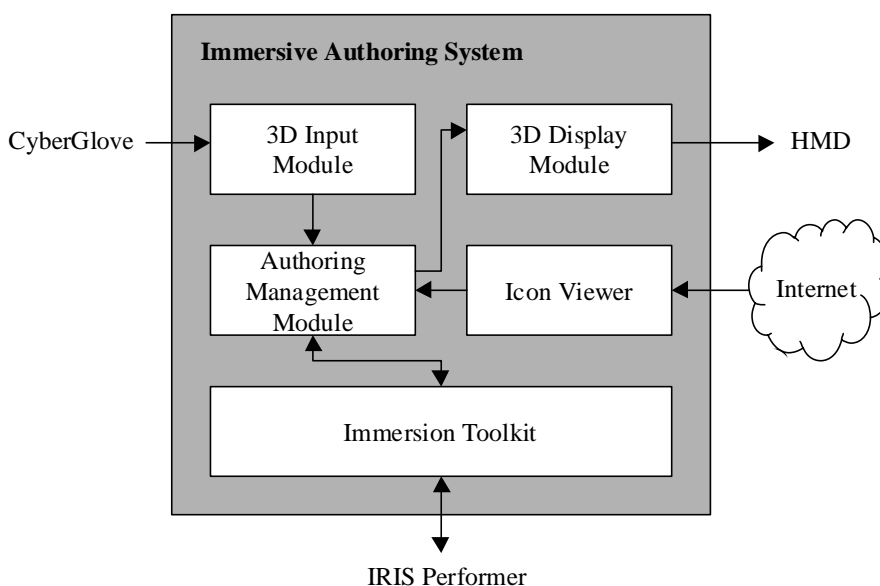


Figure 1 Software Modules and Interface

Figure 1 shows the layout of these software modules and the interface among them. As shown in Figure 1, the authoring management module and the icon viewer use our proposed immersion toolkit for authoring the VRML object.

3.2 Immersion Toolkit

In order to provide convenient VRML manipulation, we construct a library called immersion toolkit that is object-oriented and basis for VRML 2.0 specifications based on IRIS Performer. This toolkit simplifies and abstracts the task of writing graphics programming into a set of easy to manipulate 3D objects.

We note that the VRML 2.0 is the file format adopted by the Internet community and the ISO (International Standards Organization) for 3D geometry data on the World Wide Web, and the IRIS Performer provides a suitable multi-channel and multi-processing for creating real-time Virtual Reality simulation.

Our developed immersion toolkit manipulates VRML 2.0 for supporting both user interaction and constant frame rate, also is implemented using IRIS Performer.

The immersion toolkit has three API categories as the following:

- Scene managing APIs: These APIs are used to define the data structure that holds VRML nodes relevant to the entire scene, allowing programmers to manage all scene-related data in a hierarchical fashion. It is to treat over 400 VRML base classes that take care of the rendering and interaction nodes such as geometry data, cameras, lights, viewers, manipulation and the built-in culling and collision detection operations with bounding box or sphere.
- Event handling APIs: These APIs are used to process sensor, script and route node for supporting the activation of the virtual world. After the sensor or the script has generated an initial event, the event is propagated along any route to other nodes. These other nodes may response by generating additional events and so on. The sensors generate the events according to the timer or author's actions such as touching object and navigating close to a particular object. The scripts allow that the author defines arbitrary behaviors or simple animation for an object.
- Interaction processing APIs: In the interactive environment, the user grabs an object to scale, position, delete, cloning, and so on. The programmers may take advantage of high-level interactions such as responding to the user manipulate an object using by the selector and the manipulator. The manipulator has the bounding box with 8 handles for move object and for scale object along each X-, Y-, and Z-axis.

3.3 Extended VRML Format

In general, the author gets the desired VRML objects from the Internet to make the virtual space. In order to get an **object** the author may choose one in the list of VRML text description. Using this way, the author may choose undesired object because that the text description may contain incorrect information for an **object**. On the other hand, the author may choose an object after downloading the entire object data. This way incurs heavy retrieval time due to extensive volume size of the VRML object.

Note that VRML 2.0 has the capability to define new nodes. In order to avoid above weaknesses of the native VRML format, we suggest an extended VRML format to allow preview the VRML object to prior to downloading it. The suggested format adds an icon paragraph. An icon, a bitmap image in fixed size, represents the geometry information of VRML object viewed by the VRML browser. Our developed scan converter automatically generates the icon if an object not contains icon paragraph in the VRML data, and then adds icon paragraph to it.

We also developed the icon viewer to preview icons of collected VRML objects through the Internet. Figure 2 shows an active window of the icon viewer. It is assumed that the author searches for 'chair' to put the 'park bench' on the virtual space.

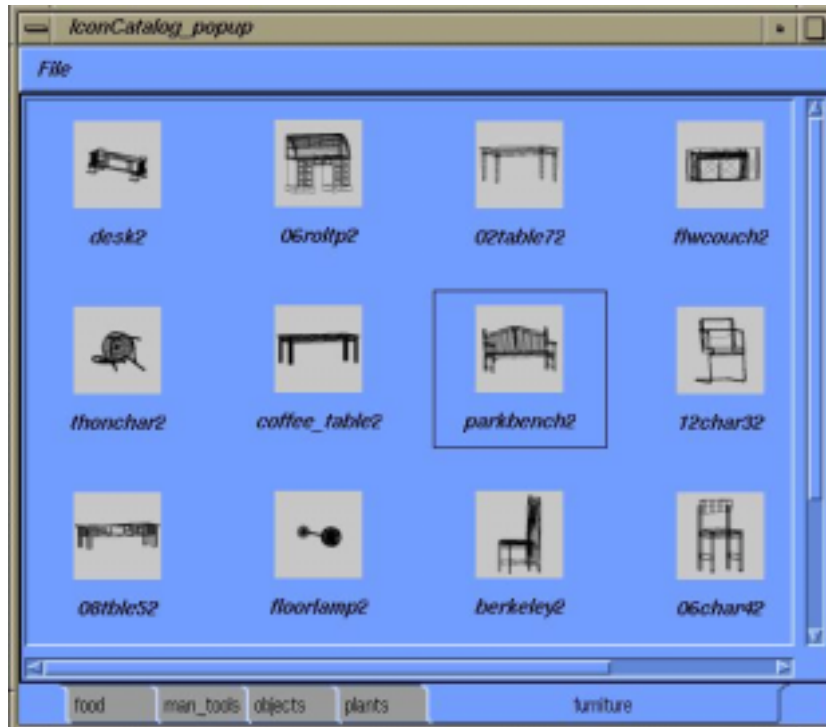


Figure 2 Active Window of Icon Viewer

3.4 Hardware Modules

Our system uses the CyberGlove to input 3D data instead of the conventional input devices such as keyboard and/or mouse. The CyberGlove is composed of optic fibers and 18 Pohlemus sensors, and detects various 3D information from the gesture of the author who wore it. The Pohlemus sensor tracks 3D coordinates data of x-, y-, and z-axis and 6 DOF values of the receiver that is mounted on the palm or the wrist of the author.

The CyberGlove interface unit is used to connect the CyberGlove and host computer with DC cable. On the other hand, the system electronics unit contains the HW and SW necessary to generate and sense the magnetic fields, compute positions and orientation, and interface with the host computer via an RS-232 port. The transmitter emits the magnetic fields, while the receiver detects the magnetic fields emitted by the transmitter.

We also use the HMD as output device in order to give highly immersion to the author as well as to provide realistic way of authoring work. The 3D database that is being written maintains a constant frame through graphic pipelines and passes through the display management module. Also, the HMD controller combines separated images into one, and synchronize the right and left images, then displays synchronized scene on the LCD panel of HMD.

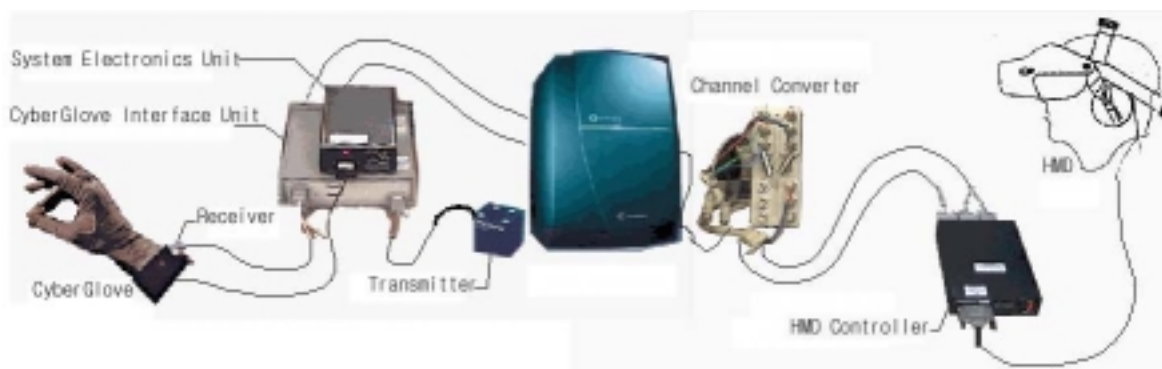


Figure 3 Configuration of Hardware Devices

Figure 3 shows the configuration of the input/output devices and related HW components on our system.

4. IMPLEMENTATION

This section describes the result of the immersive virtual space authoring system. We implemented our system using IRIS Performer 2.2 and VRML 2.0 immersion toolkits v1.0. We also use SGI's Octane MXE system as host computer.

Figure 4 and Figure 5 show the examples of authoring works. Figure 4 shows that the author is creating a virtual architectures with four channels. Figure 5 shows authoring tool windows such as the scene graph tree view, the material property editor, the texture editor and the texture control dialog.

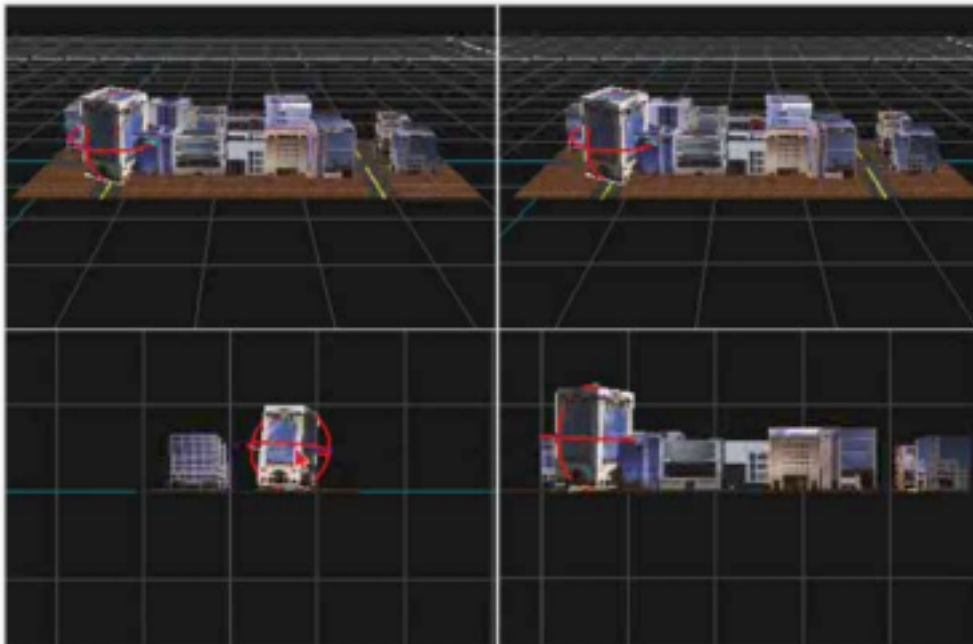


Figure 4 Creating a Virtual Space with 4 Channels

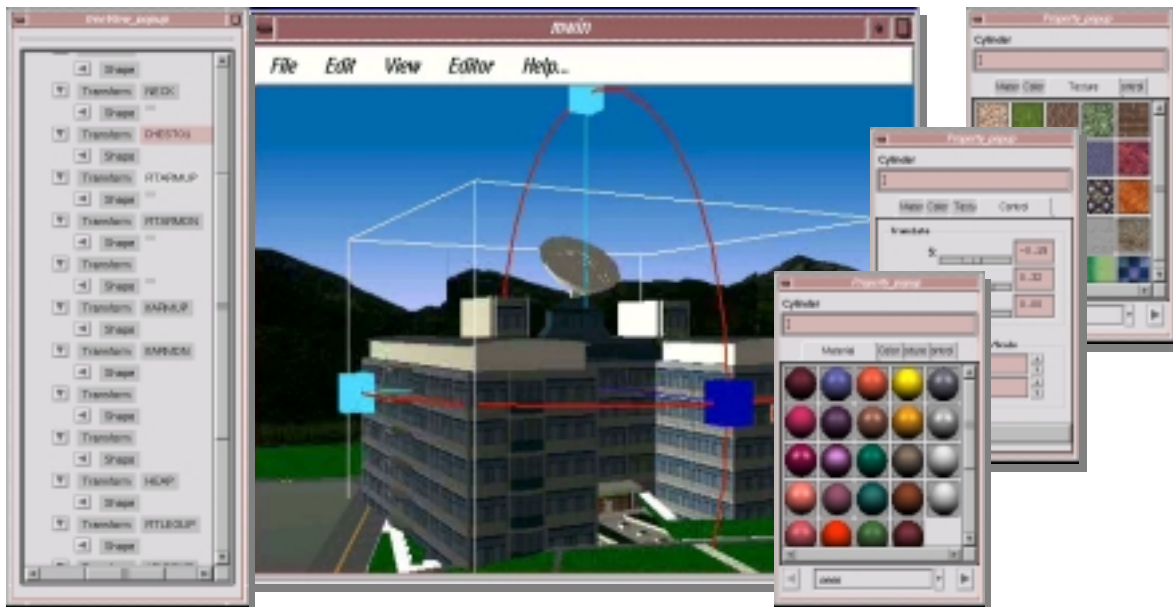


Figure 5 Virtual Authoring Windows (Left: Scene Graph Tree View, Middle: Main Authoring Window, Right three: Material Property Editor, Texture Editor and Texture Mapping Control Dialog)

Figure 6 shows that the author who wore CyberGlove and HMD is navigating the virtual space. Also, Figure 7 shows the navigation screens. Figure 7(a) is a screen shot of the virtual office, and Figure 7(b) is that of the virtual lobby.



Figure 6 Navigating in the virtual lobby



Figure 7 Navigation Screens

5. CONCLUSIONS

In this paper we developed an immersive virtual space authoring system. Our developed system has the following benefits:

First, our authoring system supports CyberGlove as input device and HMD as display device. It gives the immersion to the author in the virtual space. Second, the immersion toolkit provides various modules and its APIs to manipulate the VRML objects. It is useful to program the VE application what includes various VRML manipulation functions. It also allows for easy incorporation of additional functionality and allows for rapid prototyping of virtual spaces such as training system, science space world, electronic library, Internet cafe, Internet school, etc. Third, our suggested extended VRML format allows previewing VRML object prior to download it. It reduces the retrieval cost of VRML objects through the Internet. We further expect that it may be proposed as one of the next VRML standard.

As the result, our developed system may be used widely to developing various VE applications such as digital studios and simulation programs.

In the future, we will develop the retrieve 3D object data on the basis of a set of characteristic. And In our immersive authoring system requires for continuous study from the point of view of the development of a low-price and accurate system. And also in the method of natural interaction, 3D interface metaphor as an actual life where he lives remains to be the subject of future study.

6. REFERENCES

- [1] Randy Stiles, Sandeep Tewari, "Adapting VRML For Free-form Immersed Manipulation", In VRML 98, 1998
- [2] Rikk Carey, and Gavin Bell, "The Annotated VRML 2.0 reference Manual", Addison Wesley Developers Press.
- [3] Galia Givaty, Hendrik A.H.C. van Veen, et al. "Tele-experiments-Experiments on Spatial Cognition using VRML-based Multimedia", In VRML 98, 1998
- [4] Robert C. Zeleznik et al. , "Two Pointer Input For 3D Interaction", ACM Interactive 3D Graphics, pp. 115-120, 1997
- [5] Hiroaki Nishino et al. "Interactive Two-Handed Gesture Interface in 3D Virtual Environments", VRST'97, pp.1-8, 1997
- [6] Daniel P. Maps and J.M.Moshell, "A Two-Handed Interface for Object Manipulation in Virtual Environments", Presence, Vol.4, No.4, pp.403-416, 1995
- [7] G.Drew Kessler, "A Framework for Interactors in Immersive Virtual Environments", IEEE VR'99, Houston, TX, pp.190-197.1999

- [8] Yoshiaki Araki, "A Interaction Model for Dynamic Establishments of Realations among 3D-objects in Multi-user Virtual Environments", VSMM'98, pp.392-397, 1998
- [9] Fances L. Van Scoy, " A Virtual Environment Based File Management System: Preliminary Work", VSMM'98, pp.386-391, 1998
- [10] Fisher, S. S., Bolas, M.T., and Merritt, J. O., "Interactive Virtual Environments for the World Wide Web," Stereoscopic Displays and Virtual Reality Systems IV, Proc. SPIE 3012, 1997.
- [11] http://www.virtex.com/prod_cyberglove.html
- [12] <http://www.immersive.com/>
- [13] <http://www.mechdyne.com/>
- [14] <http://www.trimension-inc.com/>
- [15] Multigen Inc. , "SmartScene™ GUI Toolkit Programmer's Guide", Oct. 1997
- [16] <http://www.division.com>