

Hand Interface for Immersive Virtual Environment Authoring System

Chan-Su Lee, Jeong-Dan Choi, Kwang-Man Oh, Chan-Jong Park

Motion Information Team, VR Center, Electronics and Telecommunications Research Institute (ETRI)

161 Kajong-Dong, Yusong-Gu, Taejon, 305-350, KOREA

{chanslee, jdchoi, okman, cjpark} @ etri.re.kr

Abstract: Virtual Environment Authoring System (VEAS) can provide advanced graphical tools for creating, editing, optimizing and packaging its contents. Immersive VEAS (IVEAS) can additionally endow users with direct perception and interaction while authoring virtual environment (VE)s. However, it is difficult to make seamless changes among command generation, object selection, and manipulation. In this paper, we describe our VE authoring system, called Gaudi, which provides seamless mode changes between authoring commands and direct manipulation. Using hand gesture recognition and state automata, we generate authoring commands. Also, using collision detection between hand cursor controlled by CyberGlove™ and objects in VEs, our system provides direct manipulation and finger-button function. Hand interface, which includes gesture recognition and by state automata for direct manipulation and command generation, provides useful interaction for IVEAS.

Keywords: immersive virtual environments, human-computer interaction, gesture recognition, automata, 3D cursor, hand interface

1. Introduction

Generally, virtual reality (VR) systems require 3D computer graphic models that can be interactively experienced by participants [1]. Well-developed immersive VR systems will provide three-dimensional environment in which user can directly perceive and interact with objects. The underlying belief motivating most VR research is that it will lead to more natural and effective human-computer interface [2]. We have developed IVEAS to build more realistic three-dimensional computer graphic models so as to help perceive 3D world as real one and to provide easy interaction with the environment.

Let us consider authoring virtual model house for architecture. Well-developed virtual model house, which is generated by 3D graphics system, may give same feeling to customers as touring real model house. It may be helpful to author the house with perception of 3D world for the realistic 3D-model house authoring. Especially it is helpful in selecting the location of the furniture and in selecting wall color or design with realistic feeling. 3D Model loading, property changing and saving, selecting are commands for authoring system of 3D model house. Translation manipulation, rotation manipulation and scaling show various manipulation methods required. Customers who hope to find their favorite house may also want to navigate the model house, to change color of house, and to locate additional furniture in the house. Many commands, various manipulation methods and easy navigation method are required in the tasks like those.

Human use hands in many tasks such as manipulating object and communicating with other person [3]. Hands can be used for two kinds of interaction in VE. One is for direct object manipulation and the other for command generation. To generate command, hand gesture recognition is performed [4-6]. Commands are generated or

parameters are changed to interact with model object using the result of hand gesture recognition [4]. However, sensed data are not used to manipulate objects directly in most command generation approach. Direct manipulation using hands provides intuitive manipulation of VE object [7-9]. In many cases, direct manipulation systems have difficulties in command generation. Direct manipulation with hand also provides only primitive manipulation method in spite of dexterous function of hand. In spite of many researches using hand to interact with VE, there are few methods to smoothly connect command with direct manipulation. If these two kinds of interactions are combined properly, hand interface for IVEAS can provide natural and useful interaction.

This paper approaches these problems by interaction mode, state automata and hand cursor with finger-button function. In the next section, we will define interaction mode and states for hand interface. After that, we will explain dexterous hand cursor action with finger-button function, and then shows detailed implementation result and flow of interaction in each mode. At last, we discuss conclusions and future work.

2. Interaction mode and state automata

The basic functions for VE interaction, especially in authoring VE, can be divided into four kinds of actions. The first is to generate command such as loading model into VE, saving authored environment or exiting the program. The second is to select objects in the environment. It includes releasing the selected object or activating menu. The third is to manipulate the selected object. Selected objects may be rotated, translated or scaled. The last one is to navigate VE to watch previously authored VE model or move to some other place to author other part of the model.

Considering these basic functions in VE interaction, we can define 4 basic interaction modes as follows. **Normal Mode:** mode to generate commands. It is basic mode in interaction and can be changed to any other state or mode with a few exceptions. **Selection Mode:** mode to select object. In order to select far away object easily, casting line segment with varying length is done. To select menu and activate it using collision detection in hand cursor is also done in this mode. **Manipulation Mode:** mode to change properties of selected object. Basically, the selected object is changed according to user's movement in this mode. **Navigation Mode:** mode to navigate VE according to defined metaphor. Moving to selected point is also provided in this mode. Most of interactions during authoring VE are related to one of these four interaction modes.

This interaction mode can be regarded as basic situation of interaction. Interpretation domain of user's action can be divided according to interaction mode. Same user action in different interaction mode can be interpreted differently. For example, hand movement with touching object by index finger can be interpreted as selecting the object or manipulating it by translation. Like this way, we can associate several meanings for single user's action with different interaction mode.

We designed state automata to seamlessly change mode using hand gesture in VE authoring. State is changed using the result of gesture recognition according to state transition function. State transition function can be shown by state transition diagram. Figure 1 shows developed state transition diagram for VE interaction system. Output of each state in the diagram defines interaction mode. The automata have 25 states. 18 hand gestures are used for the transition of these states. The symbol located on the upper part of arrow in figure 1 is the recognized gesture, which generates state transition to the arrow direction. 'Any' symbol means that state transition is occurred for any gesture recognition result to the defined next state. NORMAL, SELECT MANIPULATE and NAVAGATE are basic states for each interaction mode. The result of gesture P5, which is defined in figure 2, can make state translation to the state of ACTIVATE in selection mode and the state of GOTO in navigation mode. They have different meanings by different state in spite of same user action. ACTIVATE state is interpreted as activating menu function and GOTO state as navigating selected position. LOAD, SAVE, EXIT and RELEASE are command-generating states and can be occurred in any interaction modes. So these states can

be transitioned from any interaction state, which is not shown in state transition diagram for simplicity. In some case, it is required to make limitation in state transition. For example, states belonged to selection mode should not be changed directly to states of navigation mode because immediate change to navigation mode from selection mode may cause unintended result. This system requests to pass normal state as intermediate states, which reduce accidental mode change from selection mode to navigation mode. By using state automata, we can control interaction flow too.



Figure 1: Hand interaction automata with interaction mode output

Interpreting user's action depending on interaction mode, we can generate various functions for the same user's gesture command or hand cursor action. And state automata control flow of interaction and generate smooth interaction with reduction of unintentional interaction flow

3. Hand cursor and button function

A cursor is required to interact with 3D virtual world. The 3D cursor controlled by a 3D input device is more useful to interact in 3D VE [10]. Direct manipulation systems with hand interface also use 3D cursor for user's hand representation. However, in most case, 3D cursor has simple arrow-shape [7,8] in spite of dexterous function of hands. Simple arrow-shaped cursor makes it difficult to use diverse abilities of fingers.

We use 3D hand model as 3D cursor. Each finger joint angle is controlled by sensed data. Line segment is attached to the end of each finger to improve usability of each finger in hand cursor. And it detects collision with VE objects. So each finger of hand cursor can be checked whether it collides with object or not. Collision of each finger with object can be used as independent finger-button function, which is similar to picking object in 3D computer graphic system with 2D display. If a finger collides with an object, state of finger-button is set to 'ON'. If not, then finger-button is 'OFF' state. In such ways, finger button of 3D-hand cursor can be defined for each finger of hand cursor with the state of ON or OFF.

In manipulating with hand, the working space is constrained by the length of finger. So it is difficult to grasp large model or to select far away object. How can we expand the working space? The metaphor we used in order to solve this problem is chopstick metaphor. In Korea, chopsticks are used frequently when eating food. We can

pick far away food using these tools. The line segment attached at each finger can be used such purpose. If we extend line segment from each finger outward by the finger length, then the working space will be doubled. Of course, line segment attached to each finger seems to be a little strange, but it can be considered as instruments attached to the finger. We can manipulate large object more easily by extending attached line segment.

Table 1 shows how can we associate finger button ON/OFF states with function for interaction. ON BUTTON shows which finger-button collides with an object. For example 'B1' means thumb finger is touched to an object and 'B1B2B3' means that thumb finger, index finger and middle finger are touched to an object. Same finger contact with an object may activate different function in different interaction mode. If index finger collides with an object, contacted object is selected in selection mode, and position of the object is translated in manipulation mode.

Table 1: Example of Button function according to interaction mode

ON BUTTON	Interaction mode	Defined function description	Related state
B1	Select	release contacted object if selected previously	RELEASE
B2	Select	select touched object	SELECT
	Manipulation	change translation according to hand cursor position change	TRANSFORM
B1B2	Manipulation	change rotation according to hand cursor orientation change	ROTATE
	Navigation	change viewpoint to the collision detected position	GOTO
B2	Select	Activate selected object	ACTIVATE
B1B2B3	Manipulation	change rotation according to hand cursor movement related to object center	OBJECT ROTATE
B1B2B3B4	Manipulation	change translation and rotation according to hand cursor movement	MANIPULATE

Index finger segment can be used another purpose. It can be extended to select long distant object. It is similar to ray-casting techniques [2,11,13]. But the length of segment can be changed to any length depending on selecting object. It also used to point some special region for navigation. Using finger-button function and special segment in index finger, we increased usability of dexterous hand function.

4. Implementation

We developed IVEAS with hand interface using CyberGlove™ and Polhemus Fastrak™ as input device and Virtual Research V6 HMD as display equipment. We called this system, Gaudi, who is the most famous architect of Spain in recent history. We use Performer™ as basic 3D graphics libraries for real time graphic processing. We developed VRML 2.0 API based on Performer™ for easy implementation and management of 3D scene. As the system is developed using VRML nodes and scene graph, authored 3D model can be easily imported from or exported to VRML format.

4.1 State change and command generation using hand gesture recognition

Automata change states according to the result of gesture recognition. We used hand gesture recognition, especially hand posture recognition, to change state of hand interaction automata. Posture is hand shape of gestures. And it is defined without regard to position or orientation movement of hand, which is proper to our system. So posture is used for hand gesture recognition. As shown in figure 2, we defined 18 intuitive and distinct postures, which comes from Korean Sign Language. For recognition, we used Fuzzy Min-Max Neural Networks [6,12] which has on-line adaptation capability. The classification result can be regarded as fuzzy value. It shows how much current gesture is similar to trained gestures. Recognition ratio for these 18 postures is 95.1%. By considering interaction mode, proper gesture number in each interaction mode can be reduced. Recognition

ratio can be improved by considering interaction mode in the interpretation of the gesture meanings. Command is generated according to changed state. So hand gesture recognition results can change interaction mode and generate command.

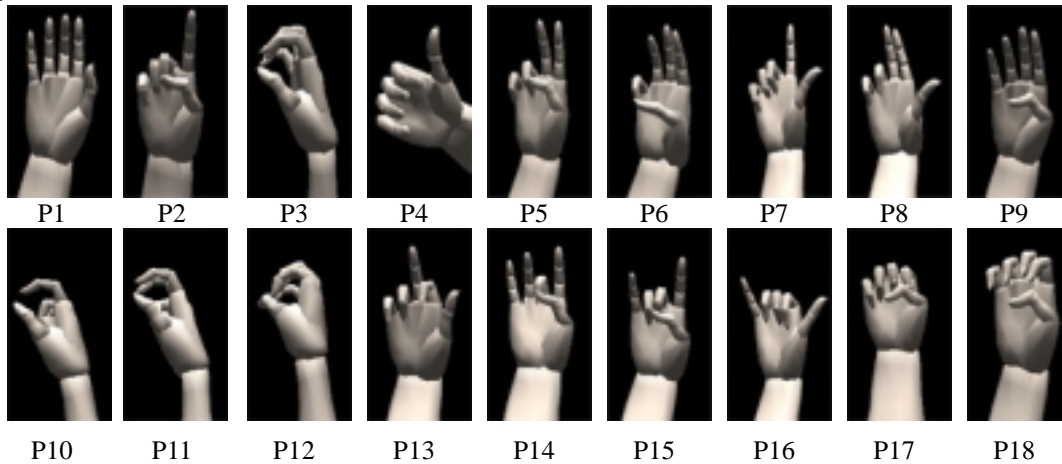


Figure 2: Defined 18 hand postures

4.2 Selecting object or menu

Selection is done for two purposes. One is to choose object to manipulate and the other is to handle menus and generate commands by them. Increasing length of line segment to select far distant object, stopping the line segment increase, and selecting object to manipulate are the common interaction cycle in selection mode. If 3D menu is activated, then the action that is defined by each menu is performed after increasing or decreasing line segment. In order to release selected object, state is changed to RELEASE by hand posture P8. The selected object is released by touching the object in this state. Bounding box is drawn for selected object and removed for released object.

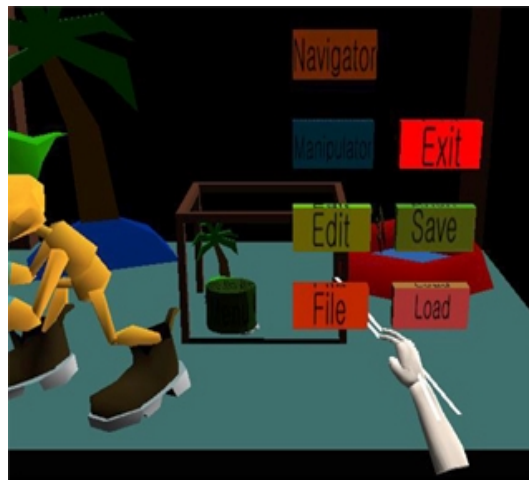


Figure 3: Menu selection

User can also select objects or activate menus by hand cursor in selection mode. The collision between index finger and object makes the object selected. After that, touching with thumb finger will release selected object. Touching index finger and middle finger together will make menu activate and do defined action or call predefined function. Figure 3 shows activating menu by extended finger segment of hand cursor. Sub-menus are displayed by 'File' menu activation. Figure 4 shows the selection of far distant object with long line-segment of

index finger. By combining gesture recognition and hand cursor, it is possible to select, release or activate menu more easily and intuitively.

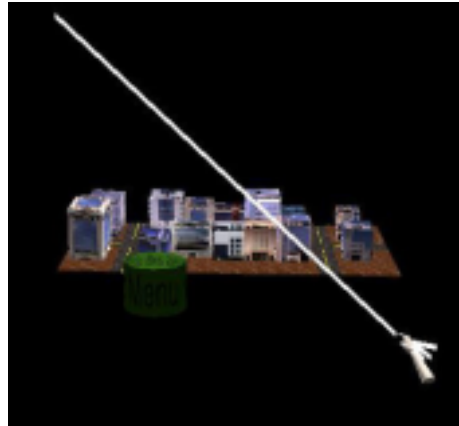


Figure 4: Long line segment

4.3 Manipulating object

Manipulating objects can also be done similar to object selection by gesture recognition or by hand cursor. For example, transform manipulation of selected object can be done in the state of TRANSFORM by gesture recognition of P10. Selected object can be rotated at the state of ROTATE by hand cursor. In the state of OBJECT_ROTATE, rotation of object is changed relative to object center. Figure 5 shows manipulating selected object by gesture recognition and hand movement.

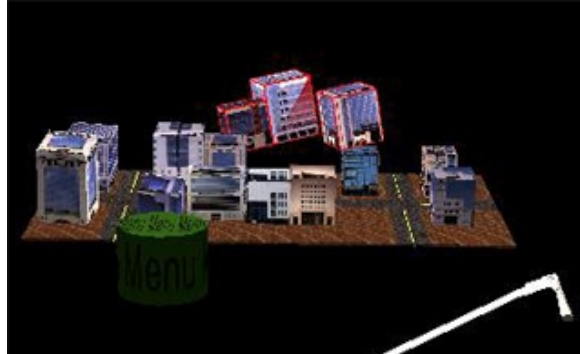


Figure 5: Direct manipulation

Direct manipulation is also done as table 1 shows by hand cursor in manipulation mode. Simple index finger contact and movement of hand cursor for an object translate the contacted object as the state of TRANSFORM does. Touching of thumb and index finger together rotates the object as done in the state of ROTATE. Direct manipulation with hand cursor is effective in near object and precise control. In manipulation, hand cursor manipulation is more intuitive and easy to interact with VE. We can manipulate far away object as well as near object easily by combining selection, manipulation and finger-button function.

4.4 Navigating authored environment

It is necessary not only to look around authored environment, but also to move to some other place in order to author that space. We provide navigation metaphors such as flight vehicle metaphor and driving vehicle metaphor. And we also provide direct movement to a desired place. Using the segment attached on index finger, user can point the place. In the state of GOTO, user's viewpoint changed to the selected point. This gives easy navigation to the desired position. It is useful especially in large-scale VE authoring.

5. Conclusion and Future work

We developed IVEAS. For more intuitive and easy interaction, we used hand gesture recognition, interaction mode and state automata. By defining interaction mode, various interactions can be developed for same user action. To connect direct manipulation and gesture command smoothly, state automata and 3D-hand cursor are also used. To use dexterous function of finger we define finger-button function in hand cursor. Segment attached at each finger enables user's working space extension by changing its length.

In the future, we will develop two-hand interface for large scale VE authoring. And also we will improve the accuracy in direct manipulation.

6. Acknowledgements

I thank for the reviewer of this paper and recommendation. This research is the partial result of "Virtual environment authoring system with VR technology" project, funded by MIC.

Reference

- [1] Woodrow Barfield and Thomas A. Furness III, "Virtual Environments and Advanced Interface Design," Oxford Univ. Press, 1995
- [2] Mark R. Mine et al. "Moving Objects in Space:Exploiting Proprioception in Virtual-Environment Interaction," *Proceedings of SIGGRAPH 97*, pp.19-26, August, 1997
- [3] David J. Sturman and David Zelter. "A Survey of Glove-based Input," *IEEE Computer Graphics & Applications*, pp.30-39, January 1994.
- [4] Hiroaki NISHINO et. al. "Interactive Two-Handed Gesture Interface in 3D Virtual Environments," *In Proceedings of ACM VRST'97*, pp. 1-8, 1997.
- [5] Yanghee Nam and KwangYun Wahn, "Recognition of Space-Time Hand-Gestures using Hidden Markov Model," *In Proceedings of ACM VRST'96*, pp. 51-58, July 1996.
- [6] ChanSu Lee et. al. "The Control of Avatar Motion Using Hand Gesture," *In Proceedings of ACM VRST'98*, pp. 59-65, Nov. 1998.
- [7] Daniel P. Mapes and J. M. Moshell, "A Two-Handed Interface for Object Manipulation in Virtual Environments," *Presence*, Vol. 4, No. 4, pp. 403-416, 1995.
- [8] Kiyoshi Kiyokawa et. al. "VLEGO: A Simple Two-handed Modeling Environment Based on Toy Blocks," *In Proceedings of ACM VRST'96*, pp. 27-34, 1996.
- [9] Joris Groen and Peter J. Werkhoven, "Visuomotor Adaptation to Virtual Hand Position in Interactive Virtual Environments," *Presence*, Vol. 7, No. 5, pp. 429-446, 1998.
- [10] Geoff Leach et al, " Elements of a Three-dimensional Graphical User Interface," *in Proceeding of Human-Computer Interaction:INTERFACE'97*, pp. 69-76, 1997.
- [11]M. R. Mine, "ISAAC: A Virtual Environment Tool for the Interactive Construction of Virtual Worlds", UNC Chapel Hill Computer Science Technical Report TR95-018, 1995.
- [12]P. Simpson, "Fuzzy Min-Max Neural Networks-Part 1:Classification," *IEEE Trans. on Neural Networks*, Vol. 3, pp. 776-786, Sep. 1992.
- [13]Bowman, D. et al. "An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments," *in Proceedings of Symposium on Interactive 3D Graphics*, pp. 35-38, 1997