

# Virtual Environment Interaction Based on Gesture Recognition and Hand Cursor

Chan-Su Lee

Kwang-Man Oh

Chan-Jong Park

VR Center, ETRI  
161 Kajong-Dong, Yusong-Gu  
Taejon, 305-350, KOREA  
+82-42-860-{5319, 5884, 5836, 1050}

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

## ABSTRACT

Virtual Reality (VR) system can provide an intuitive and natural user interface. Hand gesture can be used for more effective and easy interaction in virtual environment (VE). Immersive VE authoring system (IVEAS) required to generate many commands and to manipulate objects with various ways. This paper suggests hand interface to generate commands and to manipulate object directly. It also proposes interaction mode and state automata for combining command and direct manipulation effectively.

## Keywords

Hand interface, immersive virtual environments, HCI, gesture recognition.

## 1. INTRODUCTION

Everyday, hands are common means to manipulate objects and to communicate with other people. However, when we work with a computer or computer-controlled application, we are constrained by clumsy intermediary devices such as keyboards, mice, and joysticks.

The VR system is afforded to provide new intuitive and natural interface. VR system can support three-dimensional (3D) viewing, dynamic display and closed-loop interaction [1]. Well-developed immersive VEs will provide three-dimensional environment in which a user can directly perceive and interact with virtual objects. The underlying belief motivating most virtual reality research is that this will lead to more natural and effective human-computer interface [2]. Although immersive VE promised that a user could directly perceive and interact with virtual objects naturally and effectively in the environment, interface in immersive VE is in some aspects much more difficult than 2D desktop interface.

The approaches to interact with VE using hands can be divided into two categories. One is to generate command for functional interaction. The other is to manipulate object directly. Many systems use hand gesture recognition to generate command. Other systems may use 3D menu with 3D-cursor to support command generation in 3D environment [3]. However, menu selection is more difficult in 3D space than in 2D display. Using the result of gesture recognition, commands are generated [4-5] or parameters to manipulate object in VE are changed [6]. Most systems based on gesture recognition do not use sensed data directly to manipulate objects. Direct manipulation of VE using 3D cursor provides more intuitive manipulation method for VE object [7-9]. Still, these systems are difficult to generate various commands.

We developed Immersive Virtual Environment Authoring System (IVEAS) to provide more effective interface in VE with perception of 3D objects. This authoring system not only manipulates virtual object directly but also generates many kinds of commands in immersion environment.

We review previous works related to gesture recognition and interaction in VR system in chapter 2. After that, we describe interaction mode and state automata to combine command generation and direct manipulation. In chapter 4, we show implemented system detail of IVEAS. In the last follows conclusion and future work.

## 2. RELATED WORK

VR interface using hand gesture is motivated previously developed our hand gesture recognition system. And direct manipulation is required for more effective interaction in VR system.

### 2.1 Hand Gesture recognition system

At first, we developed hand gesture recognition system, especially Korean Sign Language (KSL) recognition system. Sign language is well-structured code gesture and has many vocabularies [10]. This sign language recognition system is applied to communication system between deaf person and normal person [11]. The recognition results of the hand gestures by deaf person wearing glove devices generate sound and text for the normal person to understand the sign language. And the texts typed by normal person are displayed by graphically generated sign language with 3D model. We applied these techniques to VR system for command generation and avatar motion control [5]. Control of avatar motion using hand gesture provides user with easy interface for virtual environment object interaction by gesture recognition and command generation. User can control avatar motion in immersive virtual environment by hand gesture. Still, it is difficult to manipulate virtual environment object directly.

### 2.2 Interaction in VR system

VR system does not have dominant metaphor or uniform framework in manipulation and interaction as desktop metaphor in 2D interaction. There are many approaches to interact with VE effectively. Arm-extension techniques and ray-casting techniques are developed to select object 3D world, [12, 13, 14]. Two-hand interface [8] and Worlds in Miniature (WIM) [15, 16] are used to manipulate virtual world object directly or to navigate virtual world.

Hand interaction gives better knowledge of spatial relationship in the virtual environment and gives more effective usage of 3D interaction [8]. Hand interface is focused on manipulating virtual objects intuitively and directly.

### 3. INTERACTION MODE AND STATE AUTOMATA

We use hand gesture to provide various interaction methods in VE without use of other device input. Most of the gesture recognition system use different gesture for different command. If user want to increase command to be used in immersive VE system, gesture to be recognized should be increased by the increased command number. Generally as the number of gesture to recognize is increase, the accuracy of gesture recognition may be decrease. Recognition of unintentional gesture or inaccurate recognition of intentional gesture could make severe problem. If we define interaction domain properly we can use same gesture with different meaning in different domain. And if we can specify acceptable command in current situation, we can reduce erroneous gesture command or unintentional gesture recognition. For such a purpose we define interaction mode and interaction state.

#### 3.1 Interaction Mode

The basic function for VE interaction, especially in authoring VE, can be divided into 4 kinds of functions. We can define 4 basic interaction modes by considering these basic functions in VE interaction. **Normal Mode:** mode to generate gesture commands and transformed to any other. After command generation, the interaction mode will be changed to normal mode. Loading model in VE or saving authored VE or exiting the program is done in this mode. **Selection Mode:** mode to select object and to cast ray for far away object selection with variation of line segment length. It includes releasing selected object or activating menu. **Manipulation Mode:** mode to manipulate object. Basically in this mode selected object is changed according to user's movement. Various property changes can be done in this mode. **Navigation Mode:** mode to navigate VE according to defined metaphor or to move user's position to defined place.

These interaction modes specify interpretation domain of gesture and hand cursor action. Same gesture can be interpreted in a different meaning with different mode. Hand cursor interaction can also have different meaning for the same action by different interaction mode. So we can associate several meanings for the same gesture or hand cursor interaction with different mode.

#### 3.2 State Automata & Gesture Recognition

We designed state automata for VE interaction. Figure 2 shows developed state transition diagram for the interaction of immersive VE. Gesture recognition and cursor interaction are used to make state transition. There are several states that are different specific action in each mode. Primarily, the state change is done by the result of gesture recognition. Additionally, hand cursor interaction can also used to make state transition for different interaction method. Output of each state defines interaction mode. Defined automata have 20 states. And 14 hand gestures are used for the transition of these states.

In figure 2 the symbol located on the upper part of arrow are gesture recognition result acting as input event, gesture recognition result. 'Any' symbol means that state transition will be occurred for any input event except previously specified event

like command generating event. **ACTIVATE** in selection mode and **POINT\_GOTO** in navigation mode are transited by the result of same gesture but interpreted as different meaning because of different interaction mode. By preventing state transition for improper or unintentional gesture, we can reduce unintentional command generation or state transition.

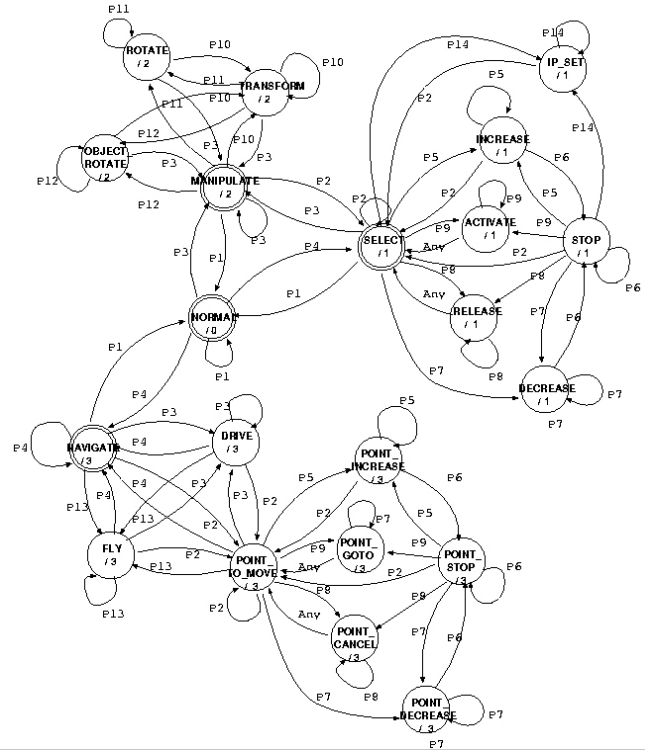


Figure 2. State transition diagram for VE interaction

#### 3.3 Hand Cursor Interaction

3D cursor controlled by a 3D input device is more useful to interact with 3D VE than 2D interaction device such as mouse [17]. System supporting direct manipulation by hand also uses 3D cursor for hand representation. But most of the 3D cursors are simple arrow-shaped 3D cursor [7,8] in spite of hands dexterous function. Simple arrow-shape makes it hard to use diverse ability of hand.

To improve usability of hand cursor, we use 3D-hand model as 3D cursor and attached line segment on the end of each finger. Each finger joint angles are controlled by sensed data. Each attached line segment detects collision with VE objects. So each segment can be checked whether it collides with object or not. Collision with object is assigned new function like button in mouse. If a finger collide with an object then the state of collided finger is 'ON'. If not, then the state is 'OFF'. Like that, each finger can have 'ON' or 'OFF' state. We can change state by combination of each finger state.

Table 1: Button interaction in hand cursor with finger state

| # of ON Button | interaction mode | state change | function description    |
|----------------|------------------|--------------|-------------------------|
| B1             | Selection        | RELEASE      | Release selected object |

|            |              |                      |  |
|------------|--------------|----------------------|--|
| B2         | Selection    | <b>ACTIVATE</b>      | select collided object   |
| B1B2       | Manipulation | <b>TRANSFORM</b>     | change position according to hand cursor movement                          |
| B1B2B3     | Manipulation | <b>ROTATE</b>        | change rotation according to hand cursor movement                          |
| B1B2B3B4   | Manipulation | <b>OBJECT ROTATE</b> | change rotation according to hand cursor movement related to object center |
| B1B2B3B4B5 | Manipulation | <b>MANIPULATE</b>    | change translation and rotation according to hand cursor movement          |

Table 1 shows how can we use button ON/OFF states to interact with VE object by hand cursor. ‘# of ON Button’ shows which button is ON state. For example ‘B1’ means thumb finger collides with an object and ‘B1B2B3’ means thumb finger, index finger and middle finger collide with an object. Same finger collision activates different function in different interaction mode. Like this way, we increase usability of dexterous hand function. Each of these button interactions changes interaction state and enables easy interaction of VE.

#### 4. IMPLEMENTATION

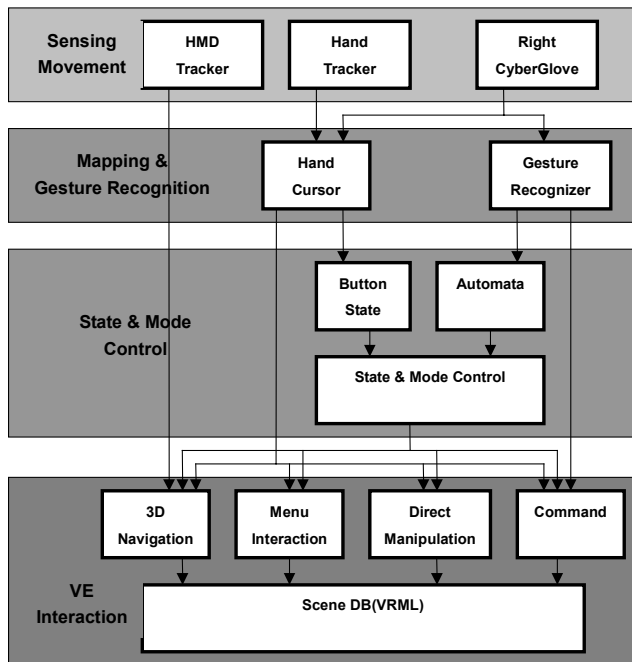


Figure 3. System configuration for Hand Interface

We developed IVEAS assuming that there are previous 3D model data made from various 3D modeling software to be used for 3D virtual object modeling. Our system can set each model’s location and change scale or color with perceiving 3D virtual world. We use CyberGlove™ and Polhemus Fastrak™ as input device and Virtual Research V6 as output display. The Virtual Reality

Modeling Language (VRML) [18] is a popular virtual world modeling language. We use VRML format as basic importing and exporting model format. And Performer™ is useful for real time graphic processing. So we developed VRML 2.0 Toolkit based on Performer™ for easy implementation and description of 3D VE and manage VRML data easily.

#### 4.1 System Configuration for Hand Interface

This system has 4 states for hand interface as figure 3 shows. At the first state, movements of user are sensed. Using CyberGlove™, angle of each finger’s joint is measured. Polhemus Fastrak™ is used to detect hand movement. The movement of head is also measured to control viewpoint. After that hand gesture recognition is done to generate commands and to change automata state. User’s hand movements control hand model. In the third stage, current interaction state and interaction modes are changed to interpret the meaning of user’s action. Collision is detected in this stage to determine state of finger button. State transition is also done according to gesture recognition result by state automata. If the result of gesture recognition is not defined in current state, warning message will be sent and no change of state will be done. In the last state, proper function is activated according to interaction mode and state. Activating function is one of the functions related to 3D navigation, menu interaction, direct manipulation, or command generation.

#### 4.2 Gesture Recognition & Command Generation



Figure 4: Defined 20 hand postures

We used hand gesture recognition, especially hand posture recognition for state change and command generation. Posture is independent of arm movement and easy to recognize, which is proper to our system. We defined 20 basic and distinct postures as shown in figure 4, many of which are from Korean Sign Language. We used Fuzzy Min-Max Neural Networks [5,19] which has on-line adaptation capability and classification result can be regarded as fuzzy member function. Input data is 15 angle data of each finger, which are normalized during calibration. Each person's angle data range is measured and normalized to [0,1]. The maximum angle is 1 and the minimum angle is 0 in each finger. To move hand model properly, offset and gain for model control is also calculated during calibration. Recognition ratio for these 20 postures is 95%. If we consider interaction states, proper gesture number in each state is reduced much. So recognition ratio is improved by considering interaction states during interpretation. We can easily change interaction mode and state, and generate command by hand gesture recognition.

### 4.3 Hand Cursor and Line Segment

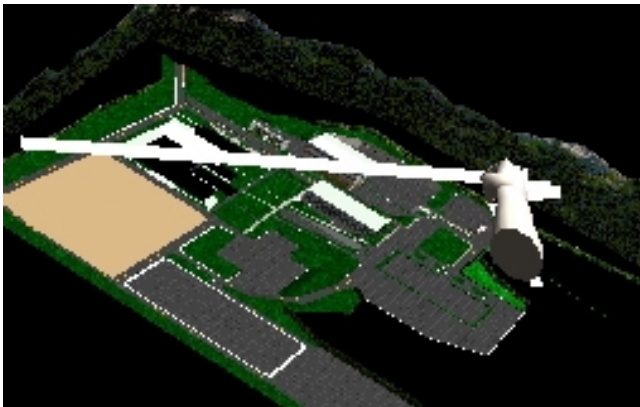


Figure 5: Long line segment

In manipulating with hand, the working space has some constraint by finger length. It may be difficult to select far away object without change of hand position and it is impossible to grasp large object model for manipulation with fingers. The metaphor we used to expand interaction space is chopstick metaphor. In Korea, chopsticks are used when eating food. We can pick food far away using these tools. The line segment attached in each finger can be used such a purpose. As we extend line segment outward, the working space also is extended by the increased length. Proper change of length of line segment enables user to select far away object from long distance away. It's similar to ray-casting techniques [2,12,13], but the length of segment can be changed freely and easily to fit to selecting object.

### 4.4 Authoring Interaction

#### 4.4.1 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 command from them. In selection mode, objects are selected by direct touch with finger or by line segment after proper length

change by increase or decrease of line segment. Menu action is executed by changing state to ACTIVATE state. RELEASE state makes selected object set free from selection list.

Using hand cursor, object selection is also done. Index finger detection with object, which makes index finger button ON state, makes the object selected. Touching index finger and middle finger will make menu function executed as ACTIVATE state. By combining gesture recognition approach and hand cursor approach, user can easily select object or activate menu.

#### 4.4.2 Manipulating object

Manipulating object can also be done similar to selecting object by two ways. In manipulation mode, MANIPULATE state enable mapping position and orientation movement of cursor to selected object directly. TRANSFORM state is only for position movement of cursor. Both ROTATE state and OBJECT\_ROTATE state are manipulation of orientation by hand cursor movement. OBJECT\_ROTATEW state is to rotate selected object related to object center.

Direct Manipulation by hand cursor is done as Table 1 shows. Contact of Index finger to the object model makes position change with state transition to TRANSFORM state. Thumb and index touch makes rotation change as in ROTATE state. In manipulation mode, hand cursor direct manipulation is more intuitive and easy than gesture command.

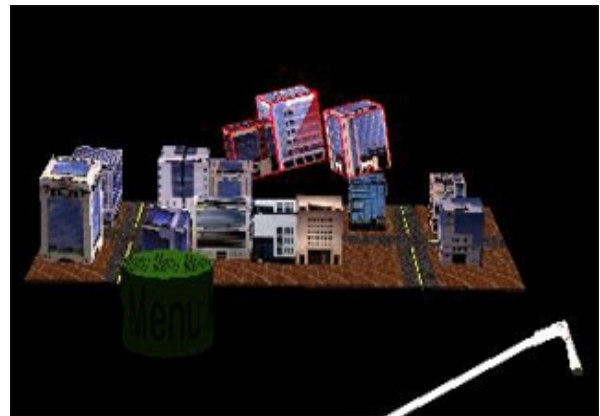


Figure 6. Direct manipulation

Figure 6 shows manipulation of far-away object after selecting with long line segment.

#### 4.4.3 Navigation

During authoring VE, it is necessary not only to look around authored environment but also to move some other place to author different place. We provide navigation metaphor such as flight vehicle, driving vehicle and trackball. Moving desired place directly is also supported. Using segment attached in index finger, user can select desired place. And by changing current state to POINT\_GOTO state, user's viewpoint changed to the selected point. This gives easy movement to the precisely defined destination.

## 5. CONCLUSION AND FUTURE WORK

We developed immersive VE authoring system. For more intuitive and easy interaction we used interaction mode and state. Using

interaction mode, diverse interaction methods are provided by same action. Direct manipulation and gesture command smoothly connected by interaction states that can be changed state either by gesture recognition or by button function of 3D-hand cursor.

In the future, we will develop two-hand interface for large scale VE authoring.

## 6. REFERENCES

- [1] Woodrow Barfield and Thomas A. Furness III, *Virtual Environments and Advanced Interface Design*, Oxford University 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] Dong A. Bowman et. al, "The Virtual Venue: User-Computer Interaction in Information-Rich Virtual Environments," *Presence*, Vol. 7, No. 5, pp. 478-493, 1998.
- [4] Yanghee Nam and KwangYun Wohn, "Recognition of Space-Time Hand-Gestures using Hidden Markov Model," *In Proceedings of ACM VRST'96*, pp. 51-58, July 1996.
- [5] ChanSu Lee et. al. "The Control of Avatar Motion Using Hand Gesture," *In Proceedings of ACM VRST'98*, pp. 59-65, Nov. 1998.
- [6] Hiroaki NISHINO et. al. "Interactive Two-Handed Gesture Interface in 3D Virtual Environments," *In Proceedings of ACM VRST'97*, pp. 1-8, 1997.
- [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] D. Morris, "Man Watching: A field Guide to Human Behaviour," Grafton, 1968
- [11] Gyu-Tae Park ZeungNam Bien, Chan-Su Lee, Won Jang and Jong-Hyeong Kim, "Real-Time Sign Language Recognition/Generation for Two-Way Communication," *World Automation Congress 98*, Anchorage, May 10-14, 1998.
- [12] 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
- [13] 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.
- [14] Ivan Poupyrep, M. Billinghurst, S. Weghorst and T. Ichikaawa "The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR," *in proceedings of the ACM Symposium on User Interface Software and Technology(UIST)*, pp. 79-80, 1996
- [15] Randy Pausch & Tommy Burnette, "Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures," *Proceedings of SIGGRAPH 95*, pp. 399-400, 1995.
- [16] Dahlan Nariman et. al., "Multi-scale Virtual Environment with Gesture Interface," *Proceedings of VSMM'98*, pp. 318-323, 1998.
- [17] Geoff Leach et al, "Elements of a Three-dimensional Graphical User Interface," *in Proceeding of Human-Computer Interaction:INTERFACE'97*, pp. 69-76, 1997
- [18] Rikk Carey and Gavin Bell, "The Annotated VRML 2.0 Reference Manual," Addison-wesley, 1997.
- [19] P. Simpson, "Fuzzy Min-Max Neural Networks-Part 1:Classification," *IEEE Trans. on Neural Networks*, Vol. 3, pp. 776-786, Sep. 1992.