Tiny Immersive Virtual Reality System with Avatar Control

Ji-Hyung LEE, ManKyu SUNG, ChanJong PARK

VR Team, Human Computing Research Department, ETRI
161 Kajong-dong, Yusong-gu, Taejon, 305-350 KOREA
ijihyung@etri.re.kr

Abstract

The avatar is considered the essential metaphor in multi-participant VR environment. As most user of VR systems desire to feel the immersion, we should generate more realistic scene. The avatar is becoming more human and complex, it needs that we must control the avatar with more detail methods. To navigate the virtual space with these controls, it is required to have the mapping between the navigation and the avatar control, and between the navigation and the input device.

If the participant navigates with his avatar in immersive virtual environment, his immersion will be enlarged. But for the expensive costs of immersive VR system or several reasons, there are few immersive systems having avatars. Therefore, it is required to apply the general immersive VR into desktop VR. In this paper, we propose the tiny immersive virtual reality system with the avatar control.

Key words: VR, Immersion, Interface, Avatar Control, Human-computer interaction, Navigation

1. Introduction

Nowadays, Virtual Reality (VR) enters a new phase. The networked or multi-participant VR system is becoming feasible system for developing or improving the network performance and technology. In multi-participant VR, the avatar which represents a participant becomes essential.

To increase the immersion, the VR systems try to imitate the real worlds and make the humanoid avatar which is made considering human structure, resembles the real human, and acts naturally as if it is the human [1]. Considering the importance of the avatar, there are many attempts to standardize the human avatar for virtual environment. Many people watched activities of Living worlds, Universal Avatar, and H-ANIM VRML Humanoid [2]. The humanoid avatar and animation specification is involved in MPEG-4 SNHC, recently.

Another aspects, immersive VR systems are introduced. Immersive VR system is more intuitive and better in the view of human-computer interaction than typical desktop VR system [3]. Respectively VR has more interfacing devices, and immersive VR Systems consist of several interfacing devices that are not used conventionally in the desktop computer environment [4]. These devices are classified by the function; the viewing, the navigation, the manipulation, and so on. Generally speaking, the HMD (Head Mounted Display) has been used to view the virtual worlds. Sometimes the see-through HMD or the large screen, for example Immersive desk, CAVE (Cave Automatic Virtual Environment) and CABIN (Computer Aided Booth for Image Navigation), has been used [5]. The mouse, the joystick and the spaceball are used to navigate the virtual worlds. And some systems take advantage of the glove and the tracker system with gesture recognition [6]. At the same time, the navigation devices are used as the manipulation devices. In some systems, the unique manipulation device is provided. And there is a noteworthy research about device interface, which desire to design the device independent metaphor [7].

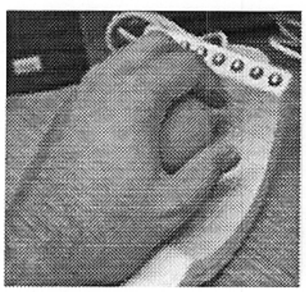
At any rate, as I mentioned above, though the avatar has a significant role in VR, there are few immersive VR systems with the avatar's motion control. Of course, there are several research results which provide avatar's motion slaving with 4 or 8 magnetic sensors [8][9]. But they are not applied into immersive VR system yet. So there is the lack of avatar motion control, the reason is categorized into three roughly. The first is that most of immersive VR systems have an eye to the navigation and the avatar's motion control is not important when the participant uses only his own view. Moreover if only one participant is in the virtual world, it is not necessary to control the avatar's motion. The second is that some large screen systems use the transparent glasses like crystalEYEs. In this kind of immersive VR system, one participant could see another participant through the glass, and the collaborative environment is more important than the avatar's existence. The last one is that some immersive VR system is designed only to model the some graphics objects. In this system, the participants can show only parts of other's avatar body, and the motion controls of these parts which might be tracked by interfacing devices is not required.

If so, how could the networked multi-user VR system provide the immersive environment? In this paper, we propose the tiny immersive virtual reality system with the avatar control.

2. Tiny Immersive Virtual Reality System

Our system take advantage of a HMD with a tracker, a spaceball, and a mouse. The participant wears the HMD, and grasps the spaceball with left hand, the mouse with right hand, which is drawn in Fig. 1. He can only view the scene through the HMD, and the camera of that view is located in the eye position of the avatar. He can navigate the virtual world and control the avatar's motion by the spaceball. Using the mouse, he can show the pop-up menu in his view. If he presses the mouse button when the cursor is located at an object in virtual world, he can select that object and manipulate.

(a) The participant weared HMD



(b) Spaceball with left hand (c) Mouse with right hand

Fig. 1. System configuration.

Our system can be applied to as an interface of desktop VR application directly. In this paper, we also try to attach our system to VOES (virtual office environment system; one of projects in ETRI, KOREA) [10]. VOES has distributed network structure and protocol for multiple user's virtual space navigation and interaction through their avatar. To minimize the network traffic, it takes a protocol as it so called VSTP (virtual space transfer protocol) [11].

3. Avatar Control

3.1. Avatar

The avatar in our system is represented by a set of the segment geometry connected by joints. The avatar is composed of with 17 segments: *head, neck, torso, waist, pelvis, left/right upper arm, forearm, hand, upper leg, foreleg*, and *foot*. The avatar has a hierarchy similar to that of H-ANIM VRML Humanoid [2]. The avatar model has 21 skeletal links and the number of joints is 16. These joints have 34 DOF (Degree of Freedom), and the avatar itself has 6 DOF in three-dimensional space. Therefore, our avatar has 40 DOF. The joints of our avatar can rotate properly for the motion, and our avatar can take 10 basic motions; *walk, side-walk, turn, sit, stand, jump, agree, deny, bow, and bye*. We expect the proper utilization of these 10 motions give us the simple navigation and communication in the virtual world [10].

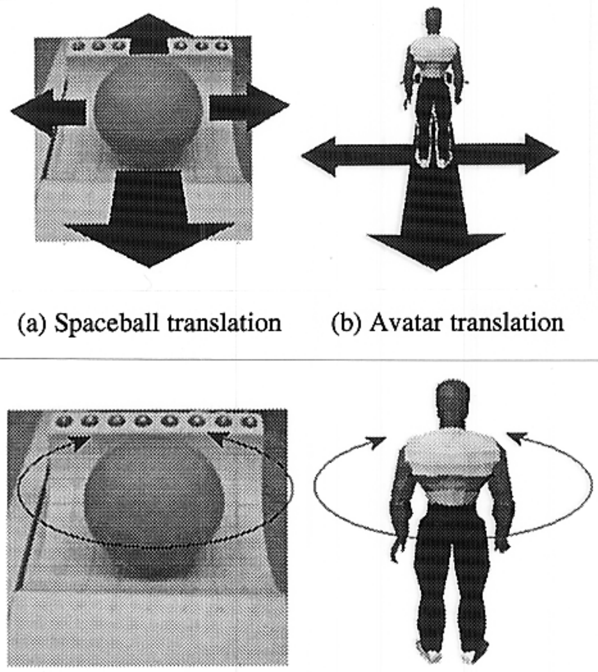
3.2. Spaceball

The spaceball represents the six DOF in three-dimensions; three translations (TX, TY and TZ) and three rotations (RX, RY and RZ) [12]. We connect the three translations and one rotation to six motions of the avatar; TZ is connected to *walk* forward and backward, and TX is connected to left/right *side-walk*, and TY is connected to *sit/stand/jump*, and RY is connected to *turn* the avatar's direction. These mappings are summarized in Table 1. The spaceball has 9 buttons, also. We use the button 1 for *bye* and the other buttons are reserved for future. The other three motions (*agree, deny and bow*) could be assigned to the rest of buttons of the spaceball, and are capable to use the head movement, which deals in section 3.4.

Table 1. Avatar's behavior mapping

Motion of Avatar	Spaceball movement
Walk	TZ
Side-walk	TX
Turn	RY
Jump	TY
Sit	TY
Stand	TY
Bye	Button 1

The spaceball translation event (Fig. 2 (a)) is mapped with avatar translation. As mentioned above, if the direction of translation is TZ (forward or backward), the avatar walk forward or backward as in Fig. 2 (b). If the direction is TX, the avatar takes a side-walk toward left or right. Drawn in Fig. 2 (c), rotation event in the spaceball is occurred, the avatar turns.



(a) Spaceball translation (b) Avatar translation

(c) Spaceball rotation (d) Avatar rotation

Fig. 2. Avatar's behavior mapping with spaceball

The TY event among the spaceball translations is mapped with 3 motions (*sit, stand and jump*), but TY has only 2 directions; upward and downward. However, 3 motions have some properties. As it were, one seated could only stand, not jump. If one is standing, he might sit or jump. If one wants to jump, he would be back to stand after the jump. Therefore, we design the event state diagram illustrated in Fig. 3.

Fig. 3. State transition Diagram

3.3. Mapping method

Our Avatar's motions are 2 types. One is the motion using the direct kinematics in VOES [10], the other is from the motion capture. But in order to navigate virtual space with avatar control, there's no difference between them besides the motion's quality. It is more important how to map the spaceball event to the avatar's motion than which type the avatar's motion is. We classify the 2 kinds of mapping methods: trigger and response. Trigger method means that an event triggers the avatar's motion, and the avatar continues the motion. Response method means that an event is occurred, the avatar responses and moves at once, and when a motion cycle is end, the avatar cease and wait the next event.

VOES takes both methods. The motions in VOES have 10 primitive motions, which is same as this paper. VOES has the motion engine and the database for the avatar's motions. When user in VOES commands his avatar to take a motion, the motion engine gets the data from the motion database, and the motion engine operates the avatar using direct kinematics. Some primitive motions (*walk, side-walk*) have toggle commands. For example, user commands to walk, his avatar would walk for-wards/backwards, and it would stop when user commands to stop. These toggle commands take the trigger method and the others take response method.

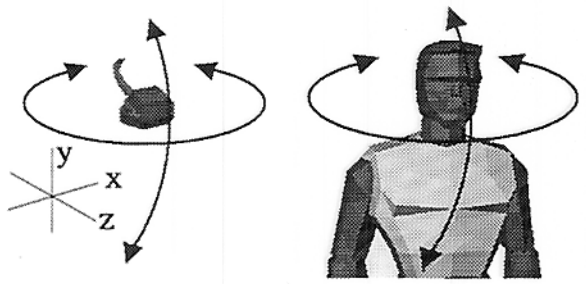
The toggle command consists of 3 stage; initial stage, intermediate stage and finish stage. Initial stage means the avatar's movement from the neutral pose (stand) to the motion's beginning pose. Intermediate stage means the motions to iterate, and finish stage means the movement from a pose in the motion to the neutral pose.

These toggle commands in VOES are for the purpose of navigation. Therefore, to navigate with the spaceball, the translation event of the spaceball is occurred, the avatar would go to walk. If the translation event of the reverse direction is occurred, the avatar would stop. In this type, the spaceball event is to trigger the avatar's motion. To use a interface of VOES, we implement our system in trigger method.

However, as we want to make a system independent on VOES, we implement to map in response method also. We take advantage of the motion capture data not to use the motion engine in VOES. We design our systems that, when a translation event of the spaceball is occurred, the avatar takes one walking cycle in the navigation. In this case, we encounter the synchronization problem. It may happen to occur the next event during processing the avatar's motion which is generated by the current event. So, it is necessary to develop the motion modification for the synchronization later.

3.4. Head tracker

The viewing direction of the man is mainly changed by his head movement. Though the head movement belongs to the body movement, the head movement is independent of the body movement. As the direction of the sight is not decided by the body's direction, we attach the tracker to the HMD to obtain the natural head movement. Our system senses the tracker rotation values of x, y axes, and apply those to the head movement. Drawn in Fig. 4, (a) illustrates the movement of the head tracker, and (b) shows the avatar's corresponding head movement. Because the camera is attached in the head of the avatar, the head movement of the participant changes the sight. And excluded three motions in the spaceball (*agree, deny and bow*) is to move the head of the avatar, the participant can perform these motions by his own movement.



(a) Tracker movement (b) Avatar head Movement

Fig. 4. Mapping between tracker movement and avatar head movement

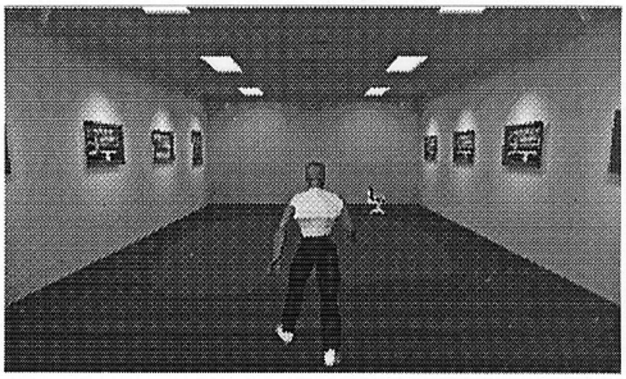
3.5. Mouse

The mouse by right hand is used to generate the pop-up menu which user could modify the options like view change during navigating the virtual space, and to select /manipulate the objects. Using this function, user could do the simple virutal space authoring.

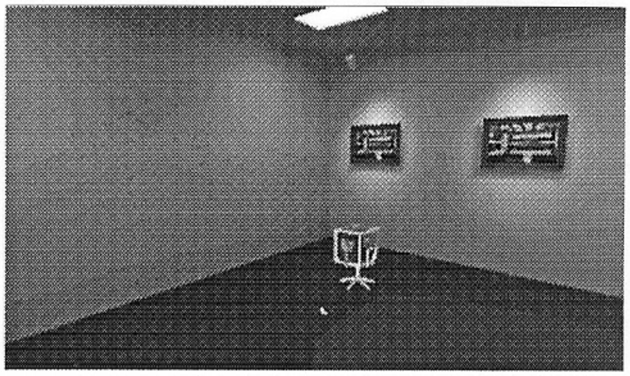
4. Experiments

We implement the system in Pentium II 300 MHz PC, Microsoft Windows NT 4.0, and program using Microsoft Visual C++ 5.0 and TGS Open Inventor library 2.4. Polhemus *Fastrak* with 1 sensor is used as a head tracker. We use XVS-Link as a device interface library and refer to the technical manual of the specific devices.

We implement our system in 2 ways: dependent on VOES, and independent on VOES. Fig. 5 shows the results, (a) is a snapshot of user's navigating with his avatar, and (b) is a snapshot of user's picking the chair to manipulate. Wearing HMD, user could show like in Fig. 5 (b) without his avatar, but changing option, we make user show the virtual space behinds the avatar's back like in (a).



(a) walk in virtual space



(b) pick the virtual object to manipulate with mouse

Fig. 5. navigate with avatar and mainipulate the virtual environment

As we want to make the avatar metaphor, we try to use some devices; Joystick, Logitech 3D Mouse, Ascension Flock of Birds, and StereoGraphics CrystalEYEs.

The joystick, generally speaking, is the input device with several buttons and the stick which supports 2 translation axes. The joystick which has been used in the flight simulations or games, is very popular and cheap. Some virtual reality applications adopted the joystick as a navigation device. The joystick in Microsoft windows 95/NT supports the maximum 6 translation axes and 32 buttons. But all of them is not supported simultaneously. Only 2 joysticks with 2 ~ 3 axes movement and 4 buttons can be supported [13]. As it were, the joystick provides only translation. It may be the problem, when applying the joystick to our system. According to the typical flight games, the joystick's movement is processed as the flight view rotation, and the flight translation is controlled by other input devices (keyboard or additional joystick accessary). Therefore, there is no problem to navigate the virtual space, but some additional techniques are required to control the avatar.

Logitech 3D mouse uses the ultrasonic waves to track the movement and the rotation of mouse in 3D space. It consists of the transmitter, control unit and reciever. As 3D mouse gives the 6 DOF, there is no problem same as in the joystick. But there is the range limit (100-degree cone, 5 feet) [14]. When applying this to our system, similar mapping which is done in spaceball is made. However, to navigate and control, user should move his arm and hand in 3D space. Due to user's fatigue, it is hard to hold logitech 3D mouse for a long time. Ascension Flock of Birds has the magnetic 3D mouse. This is the same case of Logitech 3D mouse.

We try to test the StereoGraphics CEVR (CrystalEYEs Virtual Reality). We considered CrystalEYEs to replace HMD with cheaper one. But CEVR supports the same tracking as that of Logitech 3D mouse. Therefore, we consider 2 aspects; head track and immersion. First, we test this glasses to track the head movement as in section 3.4, and get the same result. Then, we try Head Directed Navigation[15]. However, we didn't adopt the navigation by the head movement because we want to guarantee the freedom of the sight irrespective of the navigation or the avatar's control. In the second aspect, as user wearing this glasses can see the real worlds, it doesn't provide the immersion as much as HMD.

5. Conclusion and Future work

In this paper, we propose the tiny immersive VR system. Our system provides both hands interface to control the avatar intuitively in condition of HMD, to navigate very easily, and to give the simple authoring. To make the avatar metaphor, we experimented several input devices besides spaceball and mouse, and found possibilities of those devices.

As we know that it is necessary to modify the avatar's motion in the situation of the event burst, we want to research about it in near future. And we didn't consider the feeling difference between the real world navigation and that of the virtual world in this paper, but we will try to solve this gap.

Acknowledgement

Korea Ministry of Information and Communication (MIC) has provided the support for this work. The Project Management Number of ETRI is 8MG2600.

References

1. Bruce Damer: "Avatars!", Peachpit Press, (1998).
2. VRML Humanoid Animation Working Group, http://ece.uwaterloo.ca:80/~h-anim/
3. Casey Boyd: "Does Immersion Make a Virtual Environment More Usable?", http://www.acm.org/sigchi/proceedings/short-talk/cb.htm
4. Dieter Schmalstieg et al.: "Consturcting a Highly Immersive Virtual Environment: A Case Study", *Technical Report of Institute of Computer Graphics,* Vienna University of Technology, TR-186-2-95-12, Austria, (1995).
5. Michitaka Hirose: "CABIN-A Multiscreen Display for Computer Experiments", *Proceeding of VSMM'97, IEEE Computer Society*, pp.78-83, (1997).
6. ChanSu Lee et al.: "The Control of Avatar Motion Using Hand Gesture", *Proceeding of the VRST'98*, pp.59-66, Taipei Taiwan, (1998).
7. Chris Faisstnauer et al.: "Computer-Assisted Selection of 3D Interaction and Navigation Metaphors", *Proceedings of 1998 workshop on Computer graphics*, pp.40-43, Taipei Taiwan, (1998).
8. Norman I. Badler et al.: "Real-Time Control of a Virtual Human Using Minimal Sensors", *Presence*, Vol.2 No.1, pp.82-86, (1993).
9. Sudhashu K.Semwal et al.: "Mapping Algorithms for Real-Time Control of an Avatar Using Eight Sensors", *Presence*, Vol.7 No.1, pp.1-21, (1998).
10. Ji-Hyung Lee et al.: "Avatar's Gesture Recognition using Anthropometry in Real-time", *Proceeding of the 13th Symposium on Human Interface*, pp.423-426, Osaka Japan, (1997).
11. Man-Kyu Sung et al.: The Avatar Navigation of Distributed Virtual Environment By Using Multiview Client", *Proceeding of APCHI 98*, pp108-113, (1998).
12. SpaceTec homepage, http://www.spacetec.com
13. Micheal Morrison et al: "Windows 95 Game developer's guide", Sams publishing Inc., (1996).
14. 3d mouse and head tracker technical reference manual, Logitech, (1992).
15. Head Directed Navigation in Virtual Environments, http://www.cg.tuwien.ac.at/research/vr/hdn.