

A VIEW CONTROL INTERFACE FOR 3D TELE-IMMERSIVE ENVIRONMENTS

Morihiko Tamai[†], Wanmin Wu[‡], Klara Nahrstedt[‡], Keiichi Yasumoto[†]

[†]Nara Institute of Science and Technology
Ikoma, Nara 630-0192, JAPAN
{morihiko-t,yasumoto}@is.naist.jp

[‡]University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
{wwu23,klara}@uiuc.edu

ABSTRACT

The 3D tele-immersive (3DTI) environments are emerging as the next generation technique for tele-communication. In this paper, we present a novel and generic interface for view control in 3D environments. The interface uses Wii Remote, a wireless controller for a video game console. It allows the user to manipulate the virtual camera and 3D objects intuitively with buttons and through motions (pitch and roll). We conducted user studies to evaluate the interface with a professional dancer and average audiences. The results demonstrate that the Wii Remote interface is much better suited to view control in 3DTI environments since it is much easier to use, more effective, and more accurate than the conventional mouse-based interfaces.

1. INTRODUCTION

Due to the rapid cost drop of digital cameras, a multi-view video system has gained increasing research attention. In a multi-view video system, a specific event scene is captured simultaneously from a group of video cameras with different view directions. Such systems involve a wide variety of research topics such as tele-immersion, CSCW (Computer-Supported Cooperative Work) and 3D TV [1]. In particular, multi-party 3D tele-immersive (3DTI) environments [2] are considered as one of the most promising next generation tele-communication environments because of their ability to allow effective collaboration among distant users.

In a multi-party 3DTI environment, each party is a site hosting multiple 3D cameras, which are capable of capturing images including the depth information. Each site produces multiple 3D video streams, where each 3D video stream is captured from one view direction of a 3D camera (see Fig. 1 (a)). 3D video streams are then exchanged and aggregated through the overlay network among sites. At each site, the received 3D video streams are rendered as a set of 3D points in the shared virtual space together with virtual graphics objects (see Fig. 1 (b)). This virtual space promotes geographically distributed participants to do a wide variety of collaborative activities such as artistic performance, remote education/training, and video-conferencing.

Compared with conventional fixed-view video systems, in 3DTI environment, one of the most important features is to allow the user to customize his/her view for the virtual space. For example, in a collaborative dancing within a 3DTI environment [3], dancers may want to change their view angle to check dancing pose from different directions. They also may

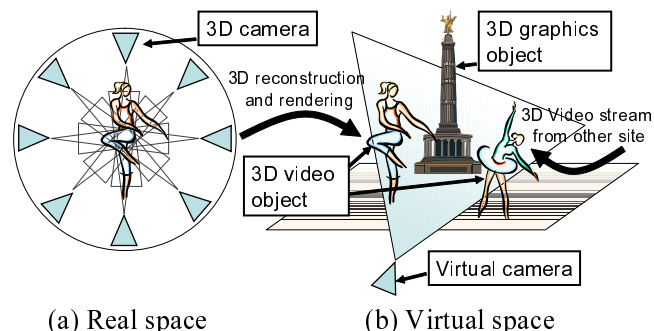


Fig. 1. Real space and rendered objects in the virtual space

want to rearrange the positions of rendered 3D videos in the virtual space to specify the dancing formation. In this paper, we focus on the user interface for view control in 3DTI environments.

In 3DTI environments, each user is not only a viewer of a virtual space but also a participant of a collaborative activity. Thus, the user interface for view control should neither hinder user's activity nor be difficult to use. Since wired devices such as wired mouse severely restrict user's activity, such a wired device is not appropriate in 3DTI environments. On the other hand, there are wireless user interfaces such as wireless mouse and wireless laser pointer [4]. However, these devices are pointing-based interfaces, and are difficult to operate correctly during activities involving body movement. Moreover, these interfaces are mainly focused on 2D environments, and it is difficult to map 3D operations to these interfaces. There are also hands-free user interfaces using head tracking device [5, 6] and motion capture devices [7, 8]. However, the user needs to wear a special tracking device, and this device is sensitive to tracking error and needs calibration [9].

In this paper, we present a view control interface using Wii Remote [10], which is a wireless controller for a video game console. Since Wii Remote has many buttons (e.g., up, down, left, and right) compared with mouse, we can map view control operations (e.g., changing view angle) to these buttons. This allows the user to control view correctly during activities involving body movement. Our interface also uses accelerometers of the Wii Remote for view control operations. This allows the user to change his/her view more intuitively using hand movement (e.g., rotating the view direction rightward by rotating user's hand rightward). We design the interface to be generic for the purpose of view control in 3D

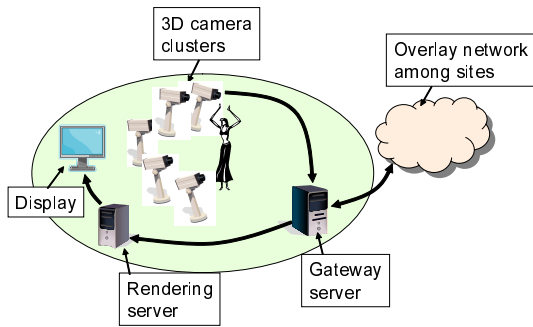


Fig. 2. A local 3D tele-immersive environment

environment, and it can be applied to not only 3DTI but also other systems such as 3D TV. We conducted user studies to investigate the usability of our interface, and the results show that it greatly outperforms a mouse-based interface in terms of ease of use, effectiveness and accuracy.

The rest of the paper is organized as follows. Section 2 summarizes our 3DTI environment. In Section 3, we first summarize the functions of Wii Remote, and then, we describe how to map these functions to user operations for view control. In Section 4, we show results of user studies. Section 5 concludes the paper.

2. 3D TELE-IMMERSIVE ENVIRONMENT

A multi-party 3DTI environment is constructed with multiple sites, where each site hosting a *local 3DTI environment*. Fig. 2 illustrates a local 3DTI environment, which consists of (1) multiple 3D camera clusters, (2) a gateway server, (3) a rendering server, and (4) a 2D (or 3D) display. Each 3D camera cluster is constructed by four 2D cameras. Three of them are black and white, and are used to obtain the depth information of pixels using trinocular stereo algorithm [11]. The fourth camera is a color camera, and is used to obtain the RGB color information of pixels. These four cameras are synchronized to take pictures at the same instant. Each 3D camera cluster is mounted to take the same physical scene from different angles to cover a wide field of view for the scene ($> 120^\circ$). Each 3D video stream produced by a 3D camera cluster is forwarded to the gateway server. After the gateway server receives 3D video streams, it forwards the streams to the rendering server and remote gateway servers which are requesting the streams. The rendering server receives 3D video streams from the gateway server, renders them into a 3D graphics scene representing a virtual space using OpenGL commands, and displays the virtual space to the screen of the display connected to the rendering server.

In the virtual space, 3D video streams from different sites are rendered simultaneously. Since all 3D camera clusters of one site are capturing the same physical scene, the streams are semantically correlated, and rendered as one 3D object (called *3D video object*) in the virtual space. Each 3D video object can be seen from different angles. We may also render *3D graphics objects* (e.g., virtual desk in tele-conferencing application) to support collaborative activity in the virtual space.

3. WII REMOTE INTERFACE DESIGN FOR 3DTI VIEW CONTROL

We use Wii Remote as the user input device. Wii Remote is designed as a one-handed remote controller. Fig. 3 shows the Wii Remote interface and its layout of buttons. The **B** button is assumed to be pushed by the index finger, while the other buttons are assumed to be pushed by the thumb. We call the group of **UP**, **DOWN**, **LEFT**, and **RIGHT** buttons the *arrow button*. The Wii Remote also has a three axes accelerometer capable of measuring accelerations along three perpendicular axes. We use the accelerometer to detect pitch and roll motions of the user's hand. The current state of the buttons and the accelerometer can be obtained through Bluetooth radio. In our 3DTI environment, a Bluetooth USB adapter is plugged into the rendering server of each site, and the current state of Wii Remote is periodically (about 30 ms) obtained.

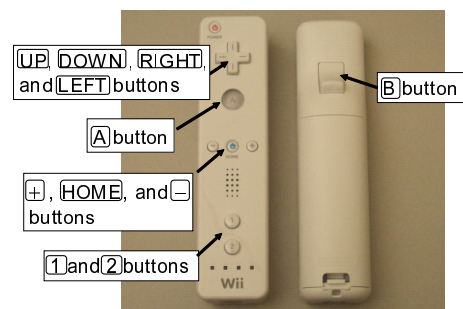


Fig. 3. The Wii Remote interface

In general, a collaborative multi-view virtual space has two major elements which the user can control: the *virtual camera* and *3D objects*. The virtual camera captures the content of the virtual space rendered on the screen of the display from one view direction (see Fig. 1 (b))¹. 3D objects consist of 3D video objects and 3D graphics objects, which are rendered in the virtual space². Our Wii Remote interface has two control modes: *camera mode* and *object mode*. The camera mode is used to control the virtual camera, while the object mode is used to control the 3D objects.

Camera Mode: For collaborative activity, the user may want to walk around the virtual space, especially when he/she is collaborating with other participants in a huge virtual space like Second Life [12]. Also the user may want to look at a specific target object (e.g., dancing partner in collaborative dancing application) from different angles. For these requirements, our interface supports the following two methods for controlling the virtual camera: *first-person view (FPV) control method* and *target-centered view (TCV) control method*. The FPV control method has the following operations: forward/backward movement and rotation. Fig. 4 illustrates a situation where the user first apply the forward movement operation, and then apply the rotation operation to the virtual

¹ If the virtual camera is controlled, i.e. moved, by the user, it looks like the user goes around the virtual space and looks at the captured 3D objects in the virtual space from different view directions.

² If the 3D object is controlled by the user, it looks like the user holds the object and rotates it to see it from different view directions. All other 3D objects stay static.

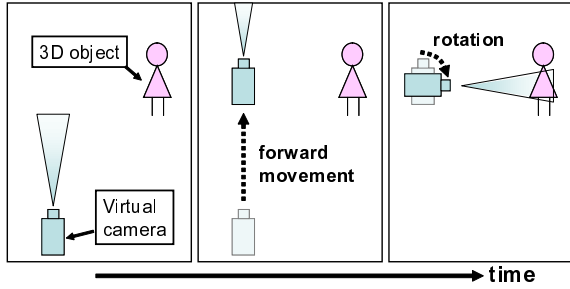


Fig. 4. First-person view control method

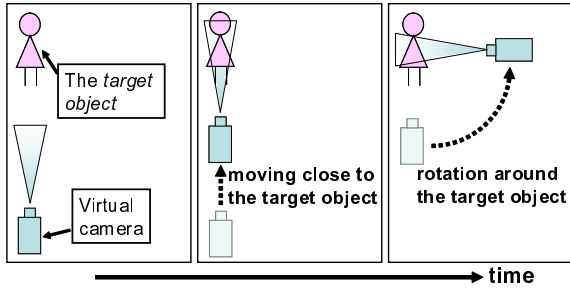


Fig. 5. Target-centered view control method

camera in the FPV control method. This view control method is popularly used in FPS (First-Person Shooter) games to walk around a 3D virtual space. On the other hand, the TCV control method has the following operations: moving close/away to/from the target object, and rotation around the target object. Fig. 5 illustrates a situation where the user first apply the moving close to the target object operation, and then apply the rotation around the target object operation to the virtual camera in the TCV control method. This view control method is often used in the movie *The Matrix*.

For each view control method, we can intuitively map operations for the virtual camera to the arrow button. Table 1 shows an example of the mapping for the FPV control method. In addition, in the TCV control method, the user needs to select the target object. For this purpose, our interface uses the $\boxed{+}$ / $\boxed{-}$ button to change the currently selected target object to the next/previous object.

We can also map pitch and roll motions to the corresponding operations of view control methods. An example of the mapping for the TCV control method is shown in Table 1. Since a user moves his/her hand not only for view control but also for collaborative activity, the user needs to explicitly specify whether he/she is moving hand for view control or not. For this purpose, we make the user to hold the \boxed{B} button during view control using motions.

Object Mode: For 3D objects, our interface supports the following two operations: *move* and *rotate*. In the *move* operation, a target object is moved forward/backward along the direction of the virtual camera, and moved leftward/rightward along the orthogonal direction to that of the virtual camera. In the *rotate* operation, a target object is rotated around the axis vertical on the ground of the virtual space. These operations can be mapped to buttons and motions as shown in Table 1. In the camera and object modes, the same buttons and motions

Table 1. Mapping between buttons/motions and operations

	Camera Mode	Object Mode
\boxed{UP} / \boxed{DOWN}	F/B movement	F/B movement
\boxed{LEFT} / \boxed{RIGHT}	L/R rotation	L/R movement
$\boxed{+}$ / $\boxed{-}$	Next/Prev object	Next/Prev object
Pitch motion	Moving close/away	N/A
Roll motion	Rotation around target	Rotation
\boxed{A}	SW to object mode	SW to camera mode

are used for some of the operations. For example, the \boxed{UP} button is used for camera movement and object movement operations. Thus, the user needs to switch between these two modes. For this purpose, the \boxed{A} button is used.

Aerial View: In a virtual space, users generally collaborate with each other using the side view (view from the horizontal direction to the ground) since this view is similar to the one used in our daily life. However, in the side view, it is difficult to arrange multiple 3D objects in the virtual space correctly, because the user can not grasp the spatial relationship between 3D objects well. To overcome the problem, our interface has the aerial view (view from above of the virtual space). We can switch between the side view and aerial view with the $\boxed{1}$ (to the side view) / $\boxed{2}$ (to the aerial view) button. In the aerial view, the user can select the target object with the $\boxed{+}$ / $\boxed{-}$ button, and move around the target objects with the arrow button. Also, users can zoom in/out the virtual space through pitch motion.

4. EXPERIMENT

To evaluate our interface, we have conducted two user studies. The purpose of the user studies is to understand the usability of our interface for 3DTI in comparison with the wired and wireless mouse device³. In our experimental setup, mouse has the capability similar to Wii Remote for view control. The camera mode and the object mode can be switched by clicking a check box on the screen. The virtual camera and 3D objects can be rotated and moved by dragging them on the screen with the mouse.

Users and Tasks: In the first user study, we have invited a professional dancer and asked her to perform the following tasks. *Task1 (Co-located teaching of a dancing pose):* The dancer and a student are immersed together in the joint virtual space. The dancer visually shows the dancing pose to the student. The student follows/repeats the visual instructions of the dancer. The dancer checks the student's pose from different view directions. *Task2 (Remote teaching):* A student is alone immersed in the virtual space. The dancer is not in the virtual space, but she is watching the student through a remote display. The dancer verbally describes movement gestures to the student. The student follows the verbal instructions. The dancer checks the student's pose from different view directions. *Task3 (Improvisation dance):* The dancer in the virtual space explores and plays with body movement ideas. We note

³In the first user study, we used a wired mouse. In the second user study, we used a wireless mouse.

Table 2. Results of the first user study

	Task1 (Mouse/Wii)	Task2 (Mouse/Wii)	Task3 (Mouse/Wii)
Ease of use	3 / 5	4 / 5	3 / 5
Efficiency	4 / 5	4 / 5	3 / 5
Effectiveness	3 / 5	2 / 5	3 / 4
Accuracy	4 / 4	2 / 4	N/A
Creativity	N/A	N/A	1 / 5

that the dancer is familiar with using mouse ⁴.

In the second user study, five graduate students and one postdoctoral researcher participated in the experiment. We asked each participant to do the task, *Learning Tai Chi*: a 3D multi-view recorded video of a person doing Tai Chi is played in the virtual space. Each participant learns Tai Chi by watching the 3D video from different view directions using our Wii Remote interface and the mouse interface ⁵.

Apparatus: In the first user study, we used Point Grey Dragonfly cameras as a 3D camera cluster, a Dell Precision 470 (equipped with an Intel Xeon 3GHz and NVidia Quadro NVS 280 PCI-E) as the rendering and gateway server, and a 61-in NEC plasma display. In the second user study, we used a Dell Precision 470 as the rendering server and a 19-in Dell LCD display.

Ranking Questions: In the first user study, we asked the dancer the following ranking questions (1: low rating, 5: high rating) at the end of each task. *Ease of use*: Does the interface make it easier for the user to control objects and the virtual camera? *Efficiency*: How quickly can user control objects and the virtual camera? *Effectiveness*: How effectively can user control objects and the virtual camera? *Overall accuracy of the task*: How accurate was the information delivery for each task? *Creativity*: How does the interface support creativity in body movement?

In the second user study, we asked each participant to answer the ranking questions about ease of use, efficiency, and effectiveness described above.

Results: Table 2 shows results of the ranking questions in the first user study. From Table 2, we can see that our interface is ranked much higher than mouse for almost all questions. Specifically, creativity of our interface is ranked significantly higher than that of a mouse. The dancer commented that our interface establishes direct connection between view control functionality and body movement. This encourages the user to manipulate the 3D objects and the virtual camera more naturally when doing a particular task. Also, more interestingly, the dancer commented that our interface is useful for choreography design, because it creates another dimension/element (i.e., view angle) in the design process. The dancer also mentioned that the color of Wii Remote affects the visibility in the virtual space. In the experiment, we used Wii Remote with white color, and it is clearly noticeable compared to darker one. So, we can use black one to make Wii Remote invisible in the virtual space.

⁴When using the mouse interface, the dancer determines the viewing angle of the virtual space prior to the dance, and during the dance, she moves freely without any mouse-based view control.

⁵We will not evaluate the learning effectiveness of participants using Wii Remote, just the ease of use and effectiveness in getting information to the participants.

In the second user study, average rank for each question was as follows: Ease of use (Mouse/Wii) = 3.0 / 4.0; Efficiency (Mouse/Wii) = 3.3 / 4.8; Effectiveness (Mouse/Wii) = 3.2 / 4.3. The results show that our interface is ranked much higher than the mouse interface for all questions. From these results we conclude that our interface provides better usability for view control in 3DTI than the mouse interface.

5. CONCLUSION

We presented a view control interface for 3DTI environments using Wii Remote. Our interface allows the user to control view intuitively with buttons and through motions. To evaluate our interface, we conducted user studies and compared our interface with mouse. Experimental results show that our interface is preferred by most of the participants, because it is much easier and more natural to use.

6. ACKNOWLEDGMENT

We would like to acknowledge the support of this research by the National Science Foundation (NSF SCI 05-49242 and NSF CNS 05-20182). The presented views are those of authors and do not represent the position of NSF. We would like to thank Mahsa Kamali, Renata Sheppard, and Zhenyu Yang for their help. We would also like to thank all the participants of our experiments for their cooperation.

7. REFERENCES

- [1] W. Matusik and H. Pfister, 3D TV: A Scalable System for Real-Time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes, *ACM Trans. on Graphics (TOG)*, Vol. 23, No. 3, pp. 814–824, 2004.
- [2] Z. Yang, W. Wu, K. Nahrstedt, G. Kurillo, and R. Bajcsy, View-Cast: View Dissemination and Management for Multi-party 3D Tele-immersive Environments, *Proc. of the 15th Annual ACM Int'l Conf. on Multimedia (MM'07)*, pp. 882–891, 2007.
- [3] Z. Yang, B. Yu, R. Diankov, W. Wu, and R. Bajcsy, Collaborative Dancing in Tele-immersive Environment, *Proc. of the 14th Annual ACM Int'l Conf. on Multimedia (MM'06)*, pp. 723–726, 2006.
- [4] X. Chen and J. Davis, LumiPoint: Multi-User Laser-Based Interaction on Large Tiled Displays, *Displays*, Vol. 23, No. 5, pp. 205–211, 2002.
- [5] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE, *Proc. of the 20th Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH'93)*, pp. 135–142, 1993.
- [6] S. Gottschalk and J. F. Hughes, Autocalibration for Virtual Environments Tracking Hardware, *Proc. of the 20th Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH'93)*, pp. 65–72, 1993.
- [7] H. J. Shin, J. Lee, M. Gleicher, and S. Y. Shin, Computer Puppetry: An Importance-Based Approach, *ACM Trans. on Graphics*, Vol. 20, No. 2, pp. 67–94, 2001.
- [8] P. T. Chua, R. Crivella, B. Daly, N. Hu, R. Schaaf, D. Ventura, T. Camill, J. K. Hodgins, and R. F. Pausch, Training for Physical Tasks in Virtual Environments: Tai Chi, *Proc. of the IEEE Virtual Reality Conf. 2003 (VR'03)*, pp. 87–94, 2003.
- [9] V. V. Kindratenko, A survey of electromagnetic position tracker calibration techniques, *Virtual Reality*, Vol. 5, No. 3, pp. 169–182, 2000.
- [10] Wii controllers page, <http://wii.nintendo.com/controller.jsp>.
- [11] J. Mulligan and K. Daniilidis, Real time trinocular stereo for tele-immersion, *Proc. of the 2001 Int'l Conf. on Image Processing*, pp. 959–962, 2001.
- [12] Second Life official website, <http://secondlife.com/>.