The Eighteenth International Symposium on Artificial Life and Robotics 2013 (AROB 18th ’13),
Daejeon Convention Center, Daejeon, Korea, January 30-February 1, 2013

Spatial operation using skeletal recognition for virtual 3D work space

Yusuke Saito[1] and Hiroyuki Nishiyama[1]

[1] Graduate School of Science and Technology, Tokyo University of Science,
2641 Yamazaki, Noda-shi, Chiba-ken, Japan
(Tel:+81-47122-1106; Fax:+81-47122-1106)
(j7412611@ed.tus.ac.jp)

Abstract: 3D desktop application software for the general environment as well as professional work environment has been increasing. However, the input method for existing applications requires large-scale facilities and special equipment. This study proposes an operation method using skeletal recognition technology with depth information only. In addition, the method compares placement and specifying rotation tasks with the wheel mouse to verify its operability and to evaluate its efficiency and usability. The efficiency of the method, excluding the adjustment and some usability aspects was found to be good, but the overall efficiency was poor compared with using a mouse because the operation of the space is unstable and the estimated position of the skeleton was inaccurate.

Keywords: 3D operation, Input device, Skeletal recognition, kinect

1 INTRODUCTION

In recent years, there have been improvements in working environments and applications that use 3D space represented on a 2D plane screen. 3dsMax[1] and GoogleEarth[2] are current examples. However, a virtual 3D work environment like this increases the degrees of freedom required for operation compared to a 2D workspace. Several operations in space have been studied to resolve these problems. However, these studies had to employ hand-held devices, large-scale systems with a marker at the site, or a camera in a room.

This paper proposes an operation method using skeletal recognition technology[4]. this technology can acquire the depth of an object by a single infrared sensor device.[3] With the skeletal recognition technology for virtual 3D workspace on the computer, at operation in real 3D space that is no tabletop surface, 3D coordinate of the dominant hand corresponds to the transition, we suggest a technique employing a complex device consisting of a 2D cursor and a 3D cursor. With respect to efficacy, we create tasks that mimic the operation of placement and of rotation in workspace virtual 3D, participants did the experiment compared with the mouse wheel, to consider and feeling its efficiency also.

2 RELATED WORK

Interfaces suitable for a 3D virtual working environment have been proposed. These include the Globe Fish and the Globe mouse[1], which decouple translation and rotation at the device level and Wii Remote[5], which achieves high degrees of freedom in space by employing infrared and tilt sensors. Operating in virtual 3D space requires much zooming and rotation, even for simple point of view (POV) and objects. However, input devices have a limited number of buttons, which hinders scalability for these mode changes to perform complex tasks in a virtual 3D work environment.

On the other hand, operations with the body have been actively researched. Examples include Kimura et al.[2] who designed and implemented a gesture-interface system for a wide-view electronic workspace and gesture operation in a large space called the ”g-stalt”[6]. In addition, Nancel et al.[3] conducted a comparative study considering input devices and differences between the dimension of the method of manipulation.

In this study, we selected operation on a tabletop surface and in space for a comparative evaluation and implemented operations in space with skeletal recognition assuming that a two-button click of the mouse is a pseudo hand gesture. During the evaluation, we decided to use only mouse buttons because hand gesture recognition accuracy can be a factor.

---

[1] 3dsMax, http://usa.autodesk.com/

[2] GoogleEarth, http://www.google.com/intl/en/earth/index.html

[3] Kinect, http://www.xbox.com/en-US/kinect

3 DESIGN PRICIPLES

3.1 Operation target

This study seeks to implement POV and object manipulation with reference to manipulations in 3D space of modeling software Metasequoia[4].

- Changing the POV (camera control)
  - Vertical and parallel movement of the POV
  - Changing POV in the depth direction
  - Moving POV along a curve (Rotating the scene)
- Changing object positions
  - Vertical and parallel movements of the object
  - Rotating to change the object orientation

Operations on a tabletop surface, such as mouse movement, are restricted to a plane, and be forced to change view when adjusting or moving the depth of the object in the above operation. However, for operation in space, if both objects and the target point are displayed on the screen, there will be less need to change the POV before moving the target point of the object.

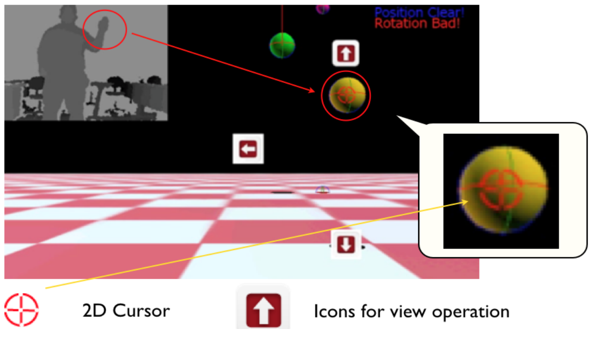

Fig. 1. Example of complex cursor.

3.2 Proposed method

In operation in space, it is difficult to control the point of view (POV) with only a 3D cursor. Therefore, we sought to design a system suitable for controlling objects with a 3D cursor and controlling the POV with a 2D cursor. Figure1 illustrates an example of using a complex 3D and 2D cursor. The user move objects and specify the direction directly with a 3D cursor. The 2D cursor controls the POV until the arrow is reached.

3.3 Systemflow

3.3.1 Skeletal information acquiring module

This module employs a depth sensor to obtain the distance between the sensor and something in sight of the sensor. Thus, it is able to acquire rate of positions for the 2D image of each skeletal site, by orthogonal projection that 3D position converted to 2D image.

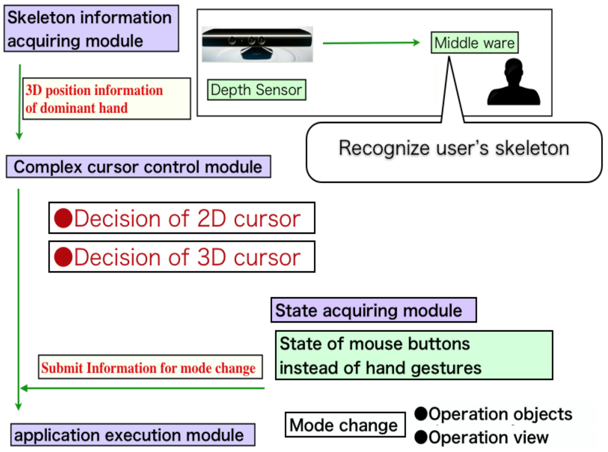

Fig. 2. Systemflow

3.3.2 Complex cursor control module

The position of the 2D cursor is determined by multiplying rate of 2D position that is determined by the orthogonal projection of the 3D hand and the screen size. tmpX(tmpY) is referred to as ratio of the dominant position in the image depth for the x (y) axis, and the position is represented by equation (1).

$$2DCusorPosition(X,Y) = (tmpX \times DisplayWide, tmpY \times DisplayHeight) \quad (1)$$

The position of the 3D cursor is determined by the position of the 2D cursor and the distance from the sensor to the dominant hand. In particular, *SD* is the distance from the work field to the sensor, *HD* is the distance between the dominant hand and the sensor, *HS* is the sensitivity of the movement distance in virtual space and the real operating distance of the hand, and equation (2) represents the position.

$$3DCusorDepth = \{(SD \times 2) - HD\}^{HS} \quad (2)$$

3.3.3 State-acquiring module

Information for switching operation mode is needed when working with objects and viewpoints. In this system, we assume that an object is clicked when the 3D cursor touches the object and that the POV is clicked when the 3D cursor is not touching the object.

3.3.4 application-execution module

This module reflects the complex cursor and the mode change process of the state-acquiring module in this application.

---

[4] Metasequoia, http://www.metaseq.net/metaseq/

4 EXPERIMENT

Participants continuously perform the two tasks of placing an object in the specified location and turning the object to the specified direction. They performed these tasks using both the proposed method and the mouse wheel, with placement distance and difficulty of angle adjustment divided into three levels. The difficulty of adjusting the setting angle is divided into three levels, beginning from the easiest: less than 45 degrees, sideways, and diagonal front. As representing the efficiency, not only in the overall time for consecutive task, it took time to adjust location and direction of the object, to reach the goal and such as measuring the number of errors.

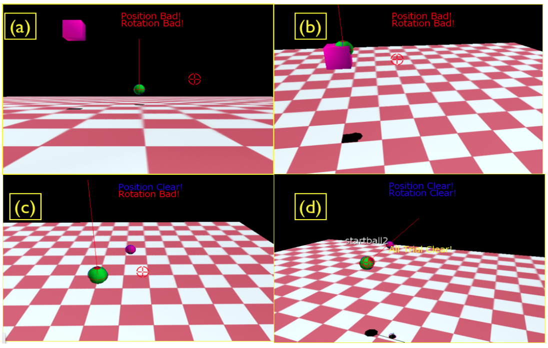

Fig. 3. Set of tasks.

4.1 Consecutive tasks

In the placement task, participants transported the green sphere to the target (purple cube). Immediately after the start, the POV is located just in front of the object. Verifying the target area requires pulling some of the POV, it has become random whether the left or right (figure3(a)). When the controlled object remains in the range of the target for 1.0 seconds, we assume the object position is stable (figure3(b)). When the rotation task is specified, the target (purple cube) will disappear when the placement task is completed (figure3(c)). Purple spheres that has become random whether the left or right indicate the direction in which the occurrence of the specified. As with objects, the consecutive task is considered completed when the red light indicating the direction of the object intersects the sphere and is stable for 1.0 second (figure3(d)).

4.2 Experiment environment

To measure the performance efficiency, we recorded the time to first pass through the destination, to adjust placement, to fit the direction completely, to adjust the direction, and to complete both tasks. In this experiment, participants performed three consecutive tasks after some practice. We recruited nine participants (one female), ranging in age from 20 to 24 years old. All are right-handed daily computer users. None are color-blind. We measure each of Operation time and number of overshoots for each of 486 trials: $2interface \cdot 3distance \cdot 3rotation \cdot 9participants \cdot 3replications$.

We asked questions based on the five-point Likert scale. We ask that question a total of 10 times per participant in the entire experiment: each distance of placement task, each difficulty of rotation adjustment, and each of 2 interfaces. Participants rated the usability in each operation object and POV and in each of the two tasks, and of fatigue. We also encouraged comments after every question.

In both the mouse and the proposed control methods, the distance between the participant and the screen was a uniform 3[m]. Depth sensors are placed as near 1.3 [m] is about space and work from the sensor. The mouse had a wheel and was made by Logicool, using a kinect depth sensor. The system and the trial were implemented on the XNA framework using `C#` language.

4.3 Results and Consideration

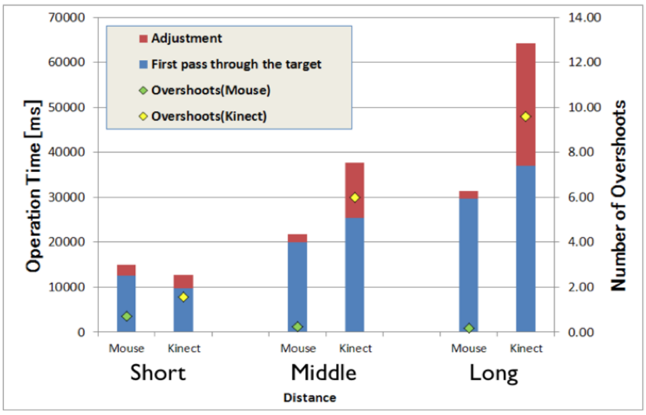

Fig. 4. Operation time : Placement task.

4.3.1 Operation Time

Figure4 and Figure5 presents the operation time for each task. First, in placement task, when placement distance is short, the average time has reduced by 23% compared to using the mouse (average 9,673[ms] vs. 12,492[ms]). However, as the distance increased, the time for moving to the target and the time for adjustment increased because time is lost in the proposed

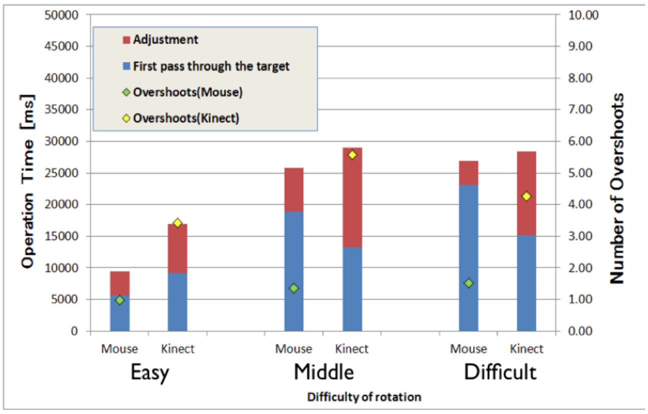

Fig. 5. Operation time : Rotation task.

method from the time the participant presses the button to change the perspective until he/she moves the 2D cursor to the icon to change the POV. Second, the number of overshoots increases considerably as the distance increases. Cursor shaking instability is a possible cause. Third, the overall efficiency using our method to specify a rotation task is less than that using mouse. However, as regards suitable time in the approximate direction, our method is not affected by the complexity of direction. This can be specified directly in spatial orientation, which means that users can easily suited to a complex direction.

4.3.2 Operability

To evaluate the operability, participants evaluated ease of controlling the object and POV on a five-point scale with 5 being comfortable and 1 being uncomfortable; fatigue was rated on a similar scale with 5 being most fatiguing. The proposed method was harder to use for moving the object (mouse 327 points vs. proposed method 284 points). In comments, some participants say that moving the object to the target was easy, but it could seem difficult due taking a long time to adjust and the instability of the complex cursor. And, there were no significant differences between the mouse and the proposed method for vertical and lateral control of the POV (339 points vs. 304 points) or for scene rotation (319 points vs. 301 points). This indicates that 2D feedback such as with an arrow icon is more efficient. About tiredness, The standard deviation of the proposed method is also greater than that of the mouse (3.57 points vs. 7.87 points). Some participants didn’t feel tired, and were able to control the object and direction smoothly in space.

4.3.3 Cause of cursor shaking instability

We will now address the issue of specific shaking instability of skeletal recognition that is considered to have the greatest impact on efficiency and usability. In placement and rotation tasks, the cursor became unstable when approaching the goal, and this did not simply seem to relate to holding the hand steady in space. When skeletal positions closely overlapped, the middleware ’s estimated 3D coordinates became inaccurate, and the tracking of the hand ’s 3D position information became unstable in the final stages.

5 SUMMERY AND FUTURE WORK

In this study, we designed and implemented a method of controlling virtual 3D objects and POV based on skeletal recognition. The proposed system used a complex cursor composed of 2D and 3D cursors. Experiments compared the complex cursor and mouse cursor with respect to efficiency and usability. In addition, using skeletal recognition technology in virtual 3D space clarified the issues that require precise operation. The future direction of this study will include designing an algorithm to account for the instability of the skeletal position recognition and improving the POV control in virtual space.

REFERENCES

[1] Froehlich, B et al.(2006), The globefish and the globe mouse: two new six degree of freedom input devices for graphics applications. In: ACM Conference on Human Factors in Computing Systems (CHI), pp. 191—199. ACM Press

[2] Kimura, A et al.(2005), Design and Implementation of Wide-view Electronic Working Space, In: Proceedings of Interaction 2005, pp. 143-150

[3] Nancel, M et al.(2011), Mid-air Pan-and-Zoom on Wall-sized Displays. In: ACM Conference on Human Factors in Computing Systems (CHI), pp. 177-186, ACMPress

[4] J. Shotton et al(2011), Real-time human pose recognition in parts from a single depth image. In Proc. CVPR

[5] WiiRemote, http://wii.com/

[6] Zigelbaum, J. et al.(2010), g-stalt: a chirocentric, spatiotemporal, and telekinetic gestural interface. In Proc. TEI ’10, pp.261-264, ACM