Three-Finger-Tap Methods for Navigation in 3D Scenes on Tablet

Hongyu Wang(✉), Takeshi Umezawa, and Noritaka Osawa

Graduate School of Advanced Integration Science, Chiba University,
Chiba, Japan
{homyetw,ume}@chiba-u.jp, n.osawa@faculty.chiba-u.jp

Abstract. This paper proposes two three-finger-tap methods designed for navigation in three-dimensional (3D) scenes. The proposed methods can be implemented in 3D applications installed on a tablet. One method integrates rotation, scaling, and translation (RST) into the specification of a triangle as a region of interest. The other is a point-of-interest (POI) method using multi-touch techniques without a virtual widget. Two experiments showed that the proposed methods perform better than a conventional RST method in two kinds of 3D scenes. Moreover, questionnaire results support the conclusion that the two methods not only performed better than a conventional POI method, but also enabled users to interact as freely and precisely as with a conventional RST method.

Keywords: 3D virtual environment · Multi-touch · 3D navigation

1 Introduction

Various three-dimensional (3D) application programs have been implemented on tablets. A 3D virtual environment in an application typically includes a virtual space that is wider than a scene viewed on a tablet display from one viewpoint. Users must change viewpoints to navigate in the virtual environment and want to control the viewpoint camera easily. Although viewpoint movement has been studied extensively in 3D virtual environments, investigating how to use fingertips on a multi-touch screen of a tablet to move the viewpoint on a tablet remains difficult. In this paper, we propose using three-finger-tap navigation methods in 3D virtual environments.

Generally speaking, if there is an area of interest in a scene, we want to navigate efficiently to see that area. In conventional 3D navigation methods, navigation must be decomposed into actions such as rotation, zooming, and translation. Moreover, we must apply these actions in an appropriate order. However, decomposition is unnecessary when changing viewpoints in the real world, and it is difficult for many people who are not skilled to navigate on a tablet using decomposition. User interfaces for 3D navigation without decomposition have not been sufficiently investigated. Therefore, we propose and evaluate methods of navigating by designating a region of interest using multi-touch contact and moving the viewpoint to ensure that the region enters the field of view appropriately.

M. Kurosu (Ed.): HCI 2017, Part II, LNCS 10272, pp. 276–287, 2017.
DOI: 10.1007/978-3-319-58077-7_22

We organize the remainder of this paper as follows. We present some previous work in Sect. 2. Section 3 describes our two methods in detail. We show our experiments and discuss our user study in Sects. 4 and 5, respectively. Finally, in Sect. 6, we draw conclusions and suggest directions for future work.

2 Related Work

In 3D virtual environments, typical interaction tasks [1] include navigation, object selection and manipulation, and system control. Navigation and manipulation in 3D scenes require six degrees of freedom (DOF), but the input of each contact finger on a touch screen provides only two DOF. In addition, Jankowski and Hachet [7] showed that over the past few decades, input devices have mostly included a mouse, a keyboard, and a touch screen, and we expect that this will persist till the next decade. Therefore, researchers have proposed many solutions to the problem of how to use those input devices for more effective interaction.

2.1 Rotation-Scaling-Translation Methods

The rotation, scaling, and translation (RST) method has been used widely for interaction in two-dimensional (2D) contexts, with many 3D applications also using RST on multi-touch tablets. The RST method enables the control of rotation, scaling, and translation using multi-touch gestures. Reisman *et al.* [12] described a screen-space method that extends 2D RST semantics into 3D environments. Their method captures the semantics of the traditional 2D RST multi-touch method and extends its principles into the 3D environment. The principles define a controllable mapping between points in the environment and on the screen. This RST method can be used not only to manipulate 3D objects, but also to navigate in a 3D virtual environment and can provide solutions for curbing ambiguities and rotational exhaustion. However, the screen-space method has two drawbacks, *bimanual interaction* and *decomposition*. A user can interact with three or more touch points, as he or she can use his or her two hands for navigation or manipulation, perhaps creating a limitation because one hand cannot pass through another. A user must decompose navigation into rotation, scaling, and translation operations.

2.2 Point-of-Interest Methods

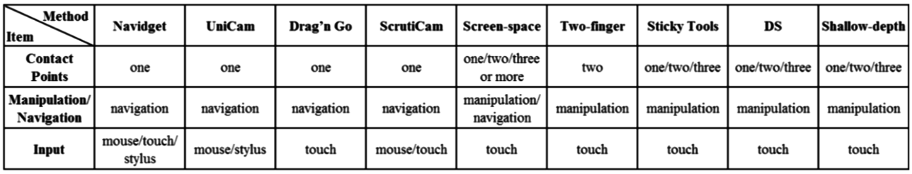
Mackinlay *et al.* [10] proposed a viewpoint movement technique called point-of-interest (POI). Given a user-selected location point on the surface of a target, the POI technique determines a normal vector at the given point and calculates the distance between the point and the new camera and then rotates the virtual camera and face toward the point automatically. This technique suggests the metaphor of “go to” and has several advantages such as ease-of-use and speed. As a typical example, Hachet *et al.* proposed a method, called Navidget [4], that permits speedy and easy positioning of a camera by a user. A user first selects the POI by a pen or mouse, then a spherical widget is popped up, and it guides the user to specify the orientation of the camera. This method extends the POI technique by enabling additional controls, exercised by circling motion and virtual widget. In other words, this method overcame the drawback of the user's inability to specify the distance between the positions of a viewpoint and a target (POI) and the orientation of the viewpoint. We have noticed that while Navidget [4] works well for navigating in 3D virtual environments, a user must first draw a circle to select the target and then wait for the virtual widget to appear. In other words, navigation is decomposed into two steps.

Mackinlay *et al.* [10] categorized navigation into *general*, *targeted*, *specified coordinate*, and *specified trajectory movement*. The POI technique is included in targeted movements, and many researchers have extended this technique using different key points. UniCam [13] is implemented as a click-to-focus method that aims at automatically choosing the endpoint of the camera orbit according to the proximity of the edges of some object. Drag'n Go method [11] combines the POI technique with some features of direct manipulation. Because it is based on a trajectory path between camera and target, the method can fully control its position and distance relative to a target and its traveling speed. Declec *et al.* [3] proposed a method that extends the POI technique using a trackball to control camera movement. Users drag the POI on the surface of the target to keep the area of interest visible. However, this method can be used only to examine models closely using a touch screen or mouse.

2.3 Other Conventional Methods

Not only in navigation, many two- or three-touch 3D manipulation techniques have been proposed (cf. Table 1). One two-finger method [9] is based on the idea of encoding the DOF by the movement of two fingers of one hand. Two tabletop methods, shallow-depth [5] and sticky tools [6], use one-, two-, or three-touch contact to control five or six DOF. Knoedel and Hachet [8] showed that it takes less time for a user to manipulate objects by directly touching the screen where the object is displayed than with indirect touch, i.e., touching a touchpad instead of touching the screen, while indirect interaction improves efficiency and precision, especially in 3D virtual environments.

Table 1. Classification of each method by features



Method Item	Navidget	UniCam	Drag'n Go	ScrutiCam	Screen-space	Two-finger	Sticky Tools	DS	Shallow-depth
Contact Points	one	one	one	one	one/two/three or more	two	one/two/three	one/two/three	one/two/three
Manipulation/ Navigation	navigation	navigation	navigation	navigation	manipulation/ navigation	manipulation	manipulation	manipulation	manipulation
Input	mouse/touch/ stylus	mouse/stylus	touch	mouse/touch	touch	touch	touch	touch	touch

2.4 Region-of-Interest

Brinkmann [2] describes use of a user-specified rectangle as a region of interest (ROI) in digital compositing. The ROI is widely used in computer vision, but not in 3D navigation. We use this concept for 3D navigation. When a user specifies an ROI in 3D scenes, the viewpoint is changed to obtain a new position and orientation to enable the ROI to be seen appropriately within a display. We think it is important for a user to be able to move a viewpoint easily to see his or her ROI in 3D navigation. Jankowski and Hachet [7] observed that touch input favors direct and fast interaction. Especially, since the appearance of the iPhone in 2007, multi-touch techniques have been used widely. In light of these considerations, we propose and evaluate three-finger-tap methods.

3 Proposed Methods

We propose two methods for viewpoint movements in 3D virtual environments using a three-finger tap. The principal idea of one method, the ROI method, is to let a user specify a triangle area he or she wants to see as an ROI. The other method, the Tripod method, is a multi-touch extension of Navidget. Both of the proposed methods integrate translation and rotation (moreover the ROI method can also integrate scaling into one operation) rather than decomposing movement into RST, in contrast to Navidget and the screen-space method [12] Both of our methods determine the direction of the viewpoint using three-finger touch in order, and thus it is possible to move the viewpoint to see the rear of an object.

3.1 ROI Method

If a user is interested in an area, he or she can specify the area. The ROI method enables a user to specify an ROI by a triangle with three-finger touch and then move the viewpoint to see the ROI appropriately on the screen in one operation. A user taps three points U, P, W on the screen, causing rays to be cast through the three points from the current viewpoint, hitting the surfaces of target 3D virtual objects on A, B, C, respectively (cf. Fig. 1). Points A, B, C form an ROI (ΔABC) in a 3D scene. The new viewpoint will face the ROI. The viewpoint V is on the normal $\vec{n}$ of ΔABC, which is through G, the center of mass of ΔABC. The distance between V and G is adjusted to m, and the ROI is fully and appropriately displayed on the screen. The distance m is expressed by the formulas

$$m = \frac{\frac{1}{2}h}{tan\frac{\alpha}{2}} = \frac{\frac{1}{2} \times \gamma h_{max}}{tan\frac{\alpha}{2}} \tag{1}$$

$$h_{max} = max\{h_B, h_C\} \tag{2}$$

where $h_B = \frac{\overrightarrow{AB} \cdot \overrightarrow{AG}}{\left|\overrightarrow{AG}\right|}$, $h_C = \frac{\overrightarrow{AC} \cdot \overrightarrow{AG}}{\left|\overrightarrow{AG}\right|}$, α is the angle of the vertical field of view, $max\{h_B, h_C\}$ is the maximum of h_B and h_C, h is the height of the vertical field of view,

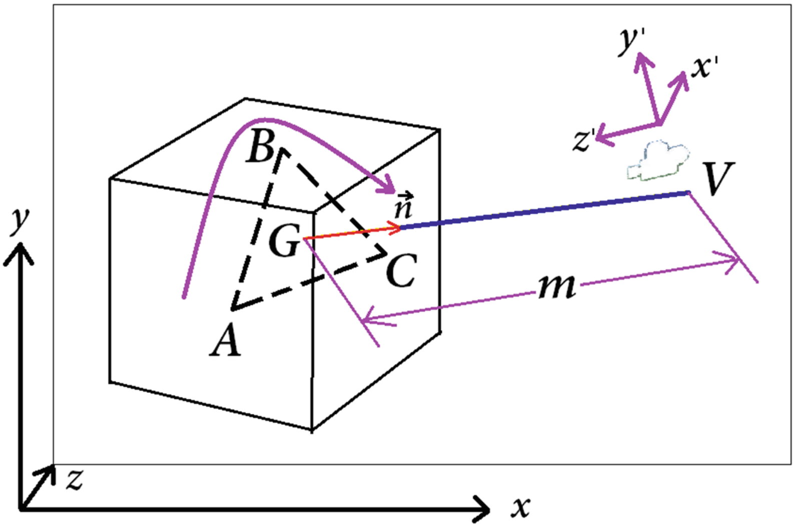


Fig. 1. ROI method

and γ is a size coefficient that we set to $\gamma = \frac{10}{9}$ in our experiment to ensure that the screen displays a scene that includes the complete ROI from the new viewpoint.

3.2 Tripod Method

The Tripod method aims to improve Navidget [4], which uses the metaphor of circling for selection and the mode of rotating the viewpoint orientation using a spherical widget. The principal idea of Tripod method is that we do not need to use widget, but instead can use terrain or objects near the target to complete the navigation. Differently from Navidget, we do not use a widget for changing orientation. The Tripod method uses a three-finger tap to define a center of rotation *B* (the second touched point) in the scene and the position and direction in a conical space (using the first and third touched points *A* and *C*) which is defined by the normal vector of the triangle (cf. Fig. 2). The definitions of symbols in the figures are the same as in the ROI method except that in this method *GV* is a definable value whose best value depends on the target size (a value of one to five times the target size is appropriate for viewing all of the target).

4 Experiments

We conducted a pilot study to improve our understanding of how our two methods perform in different scenes.

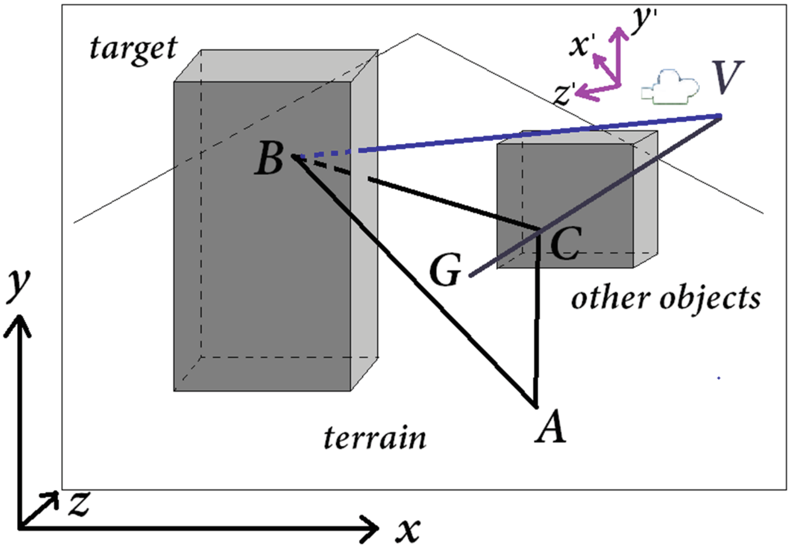


Fig. 2. Tripod method

4.1 Goal

We compared our two methods with two conventional methods: RST and POI. We evaluated the performance of all of the methods against two different kinds of task scenes. After the experiments, we conducted a questionnaire survey. The questionnaire and its results helped us to understand subjects' attitudes and feedback on each method regarding which method is simpler and more effective. In addition, we evaluated how effective each method is for each scene, and how users feel about each method.

4.2 Subjects

Ten subjects (five women and five men) participated in the experiments. They are all right handed, averaging 24 years of age. All of them are university students who had no previous familiarity with our test environment or methods.

4.3 Apparatus

We implemented our proposed methods in a virtual environment using the game development platform Unity (version 5.3.4f1). The experiments used ASUS ZenPad3s 10 (2016), a 9.8-inch tablet with a 2,048 × 1,536 pixel multi-touch screen, with Android OS v6.0 installed. Users placed the tablet on the desk and interacted with our environments using their right hands.

4.4 Experiment 1

We conducted Experiment 1 to evaluate the methods' performances when a user performs some precise navigation (Fig. 3). A subject was asked to move the viewpoint to see three marked points in the 3D scenes and put them into the corresponding rectangular frames drawn on the screen. We selected ten different scenes and placed three marks in each scene, asking subjects to complete the scenes one by one, and then we measured the average completion time. To cover many situations, we designed the scenes as follows:

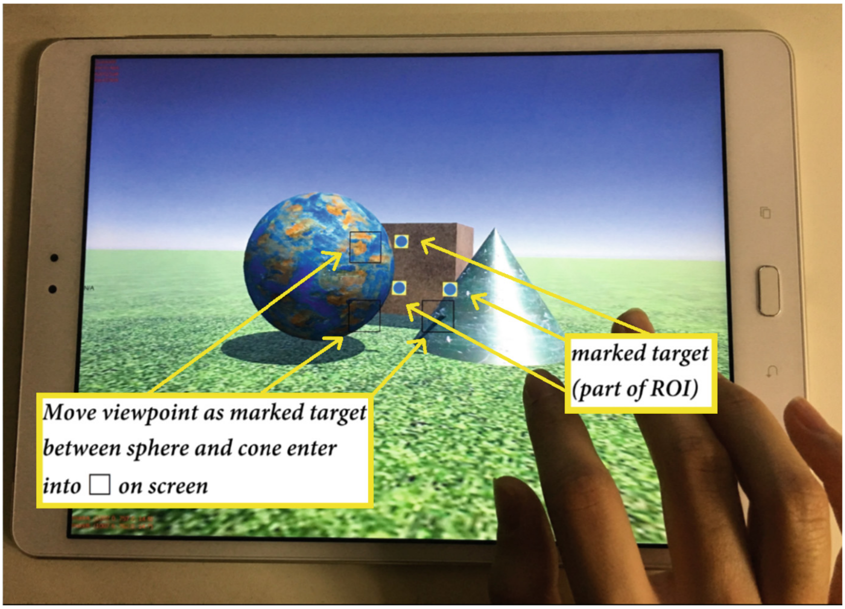

Move viewpoint as marked target between sphere and cone enter into □ on screen
marked target (part of ROI)

Fig. 3. Experiment 1

(1) Selecting three primitive shapes: sphere, cube, and cone;
(2) Designing ten scenes to show an area of interest: The front, side, top, and back of an object; containing two objects with the gap between the objects visible; containing two objects with one blocked by the other; containing two objects with both objects visible from an aerial view; containing three objects and watching from the side of the object; containing three objects, with all three objects from the gap between two objects visible; and containing three objects with all three objects visible from an aerial view;
(3) Using different conditions in different scenes: width of gap, size of object, and initial position of the viewpoint camera.

4.5 Experiment 2

We conducted Experiment 2 to evaluate the methods when a user navigates in a 3D landscape, such as on the street or plaza, with regard to how he or she navigates to a specified target fast (Fig. 4). Virtual buttons with ordinal numbers were placed in a 3D landscape, and then a subject was asked to press the virtual buttons in ascending order. This task included ten scenes, in each of which a subject was required to enlarge a virtual button sufficiently for it to be seen, but we did not prescribe a strict distance or direction. To cover various situations, we designed the scenes as follows:

(1) Designing a large space, such as a city block, including buildings and a plaza;
(2) Selecting ten different locations for target buttons: on a roof, in a window, and on the ground of a plaza;
(3) A virtual arrow is displayed to mark the position of a virtual button to reduce search time;
(4) When we are sufficiently close to a button, its color is changed to show proximity.

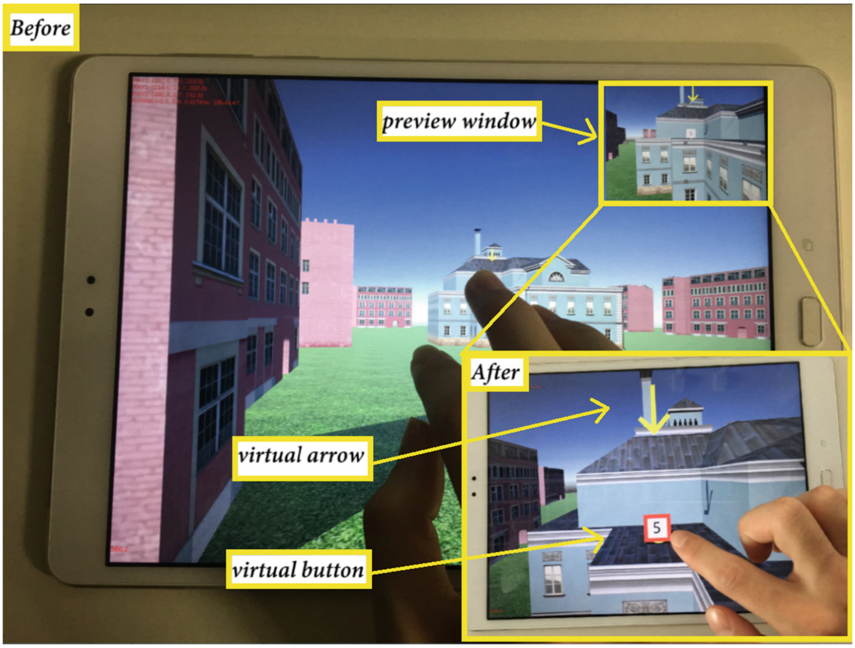


Fig. 4. Experiment 2

4.6 Procedure

Each subject was asked to sit in a chair, and the tablet was placed on a desk. Subjects performed two experiments in the same order, Experiment 1 first and then Experiment 2.

We designed a training session with five scenes. Subjects first practiced each method, and then participated in the experiment. In each experiment, a Latin square was used to counterbalance methods with subjects.

After the experiments, we asked the subjects to fill in a five-point Likert scale questionnaire, including the items “ease-of-use,” “ease-of-understanding” “interact-fast,” “interact-precisely,” and “interact-freely,” and also interviewed them. They rated the items between 1 (strongly disagreement) and 5 (strongly agreement), indicating strong disagreement, disagreement, neutrality, agreement, and strong agreement.

4.7 Result

As shown in Fig. 5, Experiment 1 showed that the average completion times of the Tripod and POI methods were shorter than RST and ROI methods, and Experiment 2 showed that the average completion times of the ROI and Tripod methods were shorter than those of Method RST and Method POI.

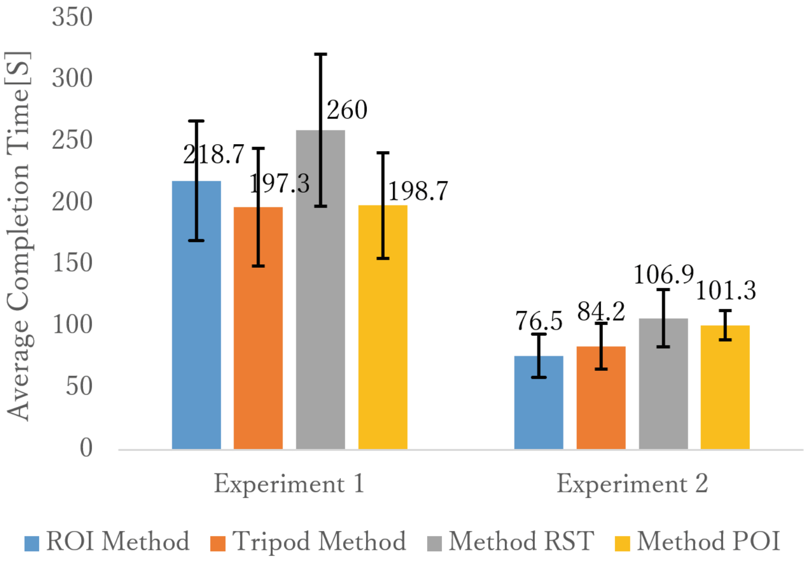


Fig. 5. Average completion time with standard deviation bars

Using one-way ANOVA, the average completion time of Experiment 1 of each method shows that there are significant differences among the performances of the methods Experiment 1 ($F(3,36) = 2.997$, $p < 0.043$). And in Experiment 2, the result also shows that there are significant differences within groups ($F(3,36) = 5.503$, $p < 0.003$).

The Tukey’s HSD test shows that the differences are significant ($p < 0.05$) between group pair (1) ROI and POI and (2) ROI and RST, and marginally significance between Tripod and RST in Experiment 2.

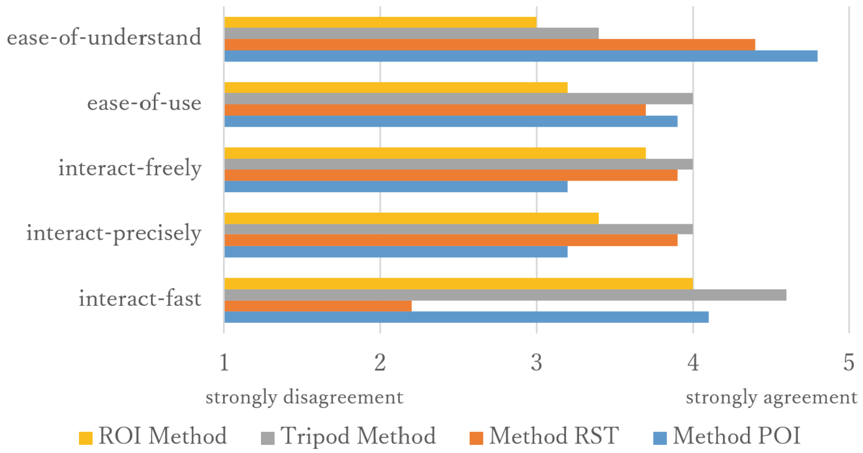


Fig. 6. Subjective evaluation of the four methods

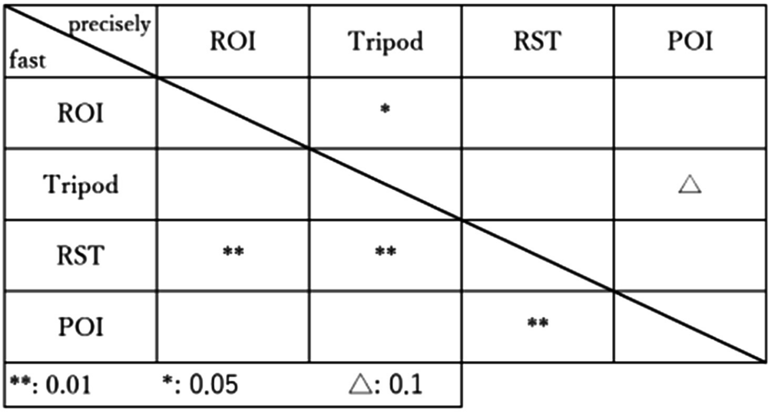
These results suggest that the proposed methods are effective in both Experiment 1 for accurate viewpoint adjustment and Experiment 2 for fast navigation. Differences between our proposed methods are not clearly shown in the experiments.

The questionnaire (cf. Fig. 6) result suggested that the average ratings of "You can use the system quickly" in the POI method is 4.1, in the Tripod method is 4.6, and in the ROI method is 4, better than the 2.2 in the RST method. The Wilcoxon signed rank test (cf. Table 2) showed that compared with the RST method, the ROI, Tripod, and POI methods were significant for "interact-fast." Tripod method was marginally significant for "interact-freely," compared with the POI method. Furthermore, the test also indicates that the Tripod method performed better on "interact-precisely" than the POI and ROI methods. However, the average rating on the item "You can understand the system easily" for the ROI method is 3 and for the Tripod method is 3.4, while the RST method scores 4.4 and the POI method scores 4.8. The Wilcoxon signed rank test also showed that the RST and POI methods were more significant than the ROI and Tripod methods. It seems that the subjects did not fully understand the relationship between the proposed methods and the 3D scenes, because the subjects underwent training in the methods only once, and several of them did not have a sense of 3D environments. We need to do further experiments to clarify the features of the proposed methods.

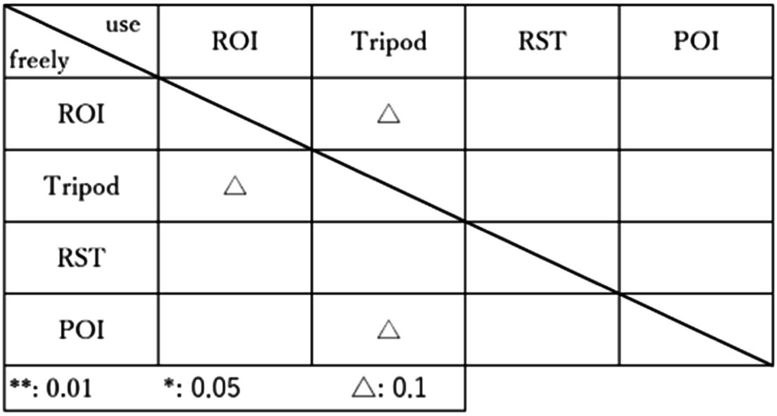
5 Discussion

Regarding the scenes of Experiment 1, in which all three objects have smooth surfaces, we did not consider including some cases of irregular objects, such as sculptures, when designing the experiment. In Experiment 2, when the button is located in the window or in a very far position, two subjects could not control the camera appropriately and thus moved the viewpoint to within a building or to a very far position through a building or an unexpected location, by using the POI method. Therefore, considering its characteristics, the performance of the POI method in Experiment 1 requires reverification.

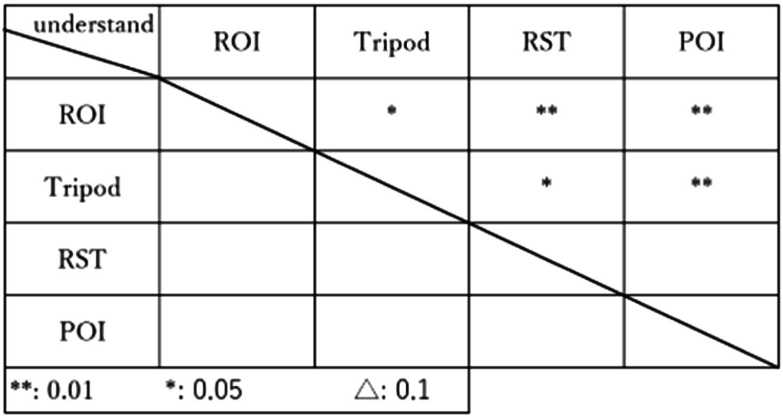
Table 2. Summary of Wilcoxon signed rank test



precisely fast	ROI	Tripod	RST	POI
ROI		*		
Tripod				△
RST	**	**		
POI			**	
**: 0.01	*: 0.05	△: 0.1		



use freely	ROI	Tripod	RST	POI
ROI		△		
Tripod	△			
RST				
POI		△		
**: 0.01	*: 0.05	△: 0.1		



understand	ROI	Tripod	RST	POI
ROI		*	**	**
Tripod			*	**
RST				
POI				
**: 0.01	*: 0.05	△: 0.1		

The RST method is considered to be not bad, except for “interact-fast.” For the items “interact-freely” and “interact-precisely,” the POI method is the worst. If the user is trained adequately, our two methods might provide effective user interfaces for navigating precisely, freely, and fast in most 3D scenes.

6 Conclusion

We proposed two 3D camera-positioning methods for 3D navigation and compared them with two conventional methods. Two experiments suggested that the proposed techniques perform better than the conventional RST method without three-touch contact in two different settings, although further investigation is needed. In the future, we will evaluate the proposed methods in further scenes including some that contain difficult shapes and irregular objects.

References

1. Bowman, D.A., Kruijff, E., Laviola, J.J., Poupyrev, I.: An introduction to 3-D user interface design. Presence: Teleoper. Virtual Environ. **10**, 96–108 (2001)
2. Brinkmann, R.: The Art and Science of Digital Compositing, p. 184. Academic Press, San Diego (1999)
3. Declec, F., Hachet, M., Guitton, P.: Tech-note: ScrutiCam: camera manipulation technique for 3D objects inspection. In: IEEE Symposium on 3D User Interfaces 2009, pp. 19–22 (2009)
4. Hachet, M., Decle, F., Knödel, S., Guitton, P.: Navidget for 3D interaction: camera positioning and further uses. Int. J. Hum. Comput. Stud. **67**(3), 225–236 (2009)
5. Hancock, M., Carpendale, S., Cockburn, A.: Shallow-depth 3D interaction: design and evaluation of one-, two- and three-touch techniques. In: Proceedings of SIGCHI Conference on Human Factors in Computing Systems, pp. 1147–1156 (2007)
6. Hancock, M., Ten Cate, T., Carpendale, S.: Sticky tools: full 6DOF force-based interaction for multi-touch tables. In: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, pp. 133–140 (2009)
7. Jankowski, J., Hachet, M.: Advances in interaction with 3D environments. Comput. Graph. Forum **34**(1), 152–190 (2015)
8. Knoedel, S., Hachet, M.: Multi-touch RST in 2D and 3D spaces: studying the impact of directness on user performance. In: Proceedings of the 2011 IEEE Symposium on 3D User Interfaces, pp. 75–78 (2011)
9. Liu, J., Au, O.K.C., Fu, H., Tai, C.L.: Two-finger gestures for 6DOF manipulation of 3D objects. Comput. Graph. Forum **31**(7), 2047–2055 (2012)
10. Mackinlay, J.D., Card, S.K., Robertson, G.G.: Rapid controlled movement through a virtual 3D workspace. In: Proceedings of the SIGGRAPH Computer Graphics, vol. 24, pp. 171–176 (1990)
11. Moerman, C., Marchal, D., Grisoni, L.: Drag’n Go: simple and fast navigation in virtual environment. In: IEEE Symposium on 3D User Interfaces, pp. 15–18 (2012)
12. Reisman, J.L., Davidson, P.L., Han, J.Y.: A screen-space formulation for 2D and 3D direct manipulation. In: Proceedings of the 22nd Annual ACM SIGGRAPH Symposium on User Interface Software, pp. 69–78 (2009)
13. Zeleznik, R., Forsberg, A.: UniCam—2D gestural camera controls for 3D environments. In: Proceedings of the 1999 Symposium on Interactive 3D Graphics, pp. 169–173 (1999)