A Novel Multi-touch Approach for 3D Object Free Manipulation

Jiechang Guo[1], Yigang Wang[1(✉)], Peng Du[1], and Lingyun Yu[1,2]

[1] Hangzhou Dianzi University, Hangzhou 310018, China
yigang.wang@hdu.edu.cn
[2] Center for Medical Imaging North East Netherlands, University of Groningen, University Medical Center Groningen, Hanzeplein 1, 9713 GZ Groningen, Netherlands

Abstract. In the field of scientific visualization, 3D manipulation is a fundamental task for many different scientific datasets, such as particle data in physics and astronomy, fluid data in aerography, and structured data in medical science. Current researches show that large multi-touch interactive displays serve as a promising device providing numerous significant advantages for displaying and manipulating scientific data. Those benefits of direct-touch devices motivate us to use touch-based interaction techniques to explore scientific 3D data. However, manipulating object in 3D space via 2D touch input devices is challenging for precise control. Therefore, we present a novel multi-touch approach for manipulating structured objects in 3D visualization space, based on multi-touch gestures and an extra axis for the assistance. Our method supports 7-DOF manipulations. Moreover, with the help from the extra axis and depth hints, users can have better control of the interactions. We report on a user study to make comparisons between our method and standard mouse-based 2D interface. We show in this work that touch-based interactive displays can be more effective when applied to complex problems if the interactive visualizations and interactions are designed appropriately.

Keywords: Direct-touch interaction · 3D manipulation · Multi-touch gesture

1 Introduction

Scientific visualization focuses on the comprehension of many different scientific datasets, such as particle data in physics and astronomy, fluid data in aerography, and structured data in medical science. By effectively exploring and interacting with data, scientists can understand, clarify, and gain insight from their dataset. Large multi-touch interactive displays have become commonplace in our daily life as a promising device, as it provides numerous significant advantages for displaying and manipulating scientific data. For instance, the large size and high-resolution screen for visualizing scientific data [1], the extra input bandwidth provided by multi-points [2] and the somesthetic perception feedback that offers users the sense of control of their data [3]. There is also great potential for direct-touch interaction to promote the process of scientific visualization, as it can meet the need of direct manipulation on the data rather

J. Chang et al. (Eds.): AniNex 2017, LNCS 10582, pp. 159–172, 2017.
https://doi.org/10.1007/978-3-319-69487-0_12

than being restricted to traditional indirect mouse/keyboard-based interaction in desktop environments. Those benefits of direct-touch devices motivate us to take full advantage of them to build a more intuitive touch interaction of scientific 3D data.

3D manipulation is the fundamental task for scientific visualization. Unlike the touch interaction of 2D data, interacting with 3D data in a virtual 3D world using a 2D multi-touch display is a tough and challenging task that demands an instinctive mapping from 2D touch input to 3D manipulation. For positioning tasks, the absence of depth information makes it complicated for users to translate the object along the depth direction. For orienting tasks, it is even harder for users to determine the rotation angles respectively for each of the three axes let alone defining the rotation center in a 3D space. Conventionally, 2D interfaces such as mouse typically uses 3D widgets or combinations of mouse and keyboard to manipulate the 3D models. However, these interactions are indirect. Besides, 3D widgets in system control mode is not suggested for the touch-based large screen since it is difficult for users to reach the menu or buttons on a large screen. In contrast, multi-touch devices provide the possibilities to accomplish different exploration tasks by multi-touch gestures. Most of the previous works [2, 4, 5] introduced the basic interactions of 3D manipulation, but to place and orient a 3D object correctly is still difficult.

Previous works have proven that the touch interaction facilitates precise control over 3D particle space [3]. In this work, we focus on the interactions of the structured 3D model, which have even higher requirements in terms of the precision of the manipulations. We present a novel multi-touch approach for positioning and orienting structured models in 3D visualization spaces, which combines multi-touch gestures and an additional assistant axis. Our method supports 7-DOF manipulation (translation along the x-/y-/z-axis, rotation around x-/y-/z-axis, and uniform zoom), two free rotation modes (trackball rotation and rotation around user-defined center), and viewpoint control to view and manipulate 3D models from different viewpoints. We evaluate our method by a user study, with a controlled experiment of eight independent interaction tasks and one complex integrated task.

2 Related Work

The sense of touch is significant in human-computer interaction (HCI) for its somesthetic capabilities [6]. Thus, enabling touch interaction with scientific visualization is essential. Interaction in scientific visualization is unique in HCI domains for complex analysis tasks and datasets [7]. The challenge of touch interaction in scientific visualization is defining an instinctive mapping from 2D touch input to 3D manipulation [1].

Several researchers and tool developers have addressed this challenge of 2D-to-3D mapping, and their work gave us some inspiration. Based on the de facto standard technique RST for manipulating 2D data [8], Screen-space technique extended it into 3D manipulation which controls the 6 DOF in an integral way with three or more fingers [9]. We have benefited a lot from this technique. The Z-technique [4] for performing depth positioning by adding another finger can be considered as a baseline for designing intuitive 3D position interactions. Other techniques like the Sticky tools designed full 6-DOF interactions by dividing the DOFs of 3D manipulation tasks [10].

Depth-Separated Screen Space (DS3) [11] extended the concept of separation DOFs which consisted of the Z-technique [4] to control translation and Screen-space technique to control rotation. Studies like Sticky Tools [10] and DS3 [11] pointed out that by separating the degrees of freedom, the RST technique can perform better, from which we derived our work. Besides, the work of Mendes [12] inspires us to provide unconstrained viewing angles for interactions which most researches did not involve in. Eden, a professional multi-touch tool, was designed for constructing virtual organic environments that used conjoined touch instead of single touch to differentiate interactions [5]. Yu et al. [3] proposed a frame-based touch interaction for manipulating astronomy particle data in 7-DOF, with the assistant of the frame they allowed full 7-DOF manipulations using one or dual touch input. FI3D focused on the scientific exploration of particle data in astronomy and supported the manipulation of the data as a whole rather than specific objects in the scene. When there are multiple structured 3D models in the scene, the interactions of FI3D will be inappropriate. We learned from the lessons of the mentioned works and made improvements.

3 Proposed Approach

3.1 Design Goals

To overcome the challenges presented above, we developed a number of supplementary goals so as to help us design better interactions for users to manipulate the 3D structured model. We designed our method to:

G1: support all 7-DOF for 3D manipulations,
G2: provide constrained manipulations as well as free manipulations,
G3: enable users to define rotation center in 3D space,
G4: provide extra depth hints for precise manipulations,
G5: provide unconstrained viewing angles for interactions,
G6: design sophisticated interactions without influencing the usability of the basic interactions,
G7: ensure a smooth switch between different types of interactions,
G8: construct clear and intuitive gestures, and
G9: be more effective than standard 2D interface when applied to complex problems.

In the following subsection, we state the method we use to achieve the above goals. We discuss how instinctive mappings from 2D input to 3D manipulations have been developed and introduce the user controlled axis tool and gestures that are designed for interacting with the 3D model.

3.2 Interactions Design for 3D Model Manipulation

Our method supports eleven operations for interacting with the 3D model, and these operations are realized by a user controlled axis tool and a set of elaborately designed gestures. Figure 1 shows a screenshot of rotating the object about the x-axis. Note that we use the shadow to provide depth hint (G4).

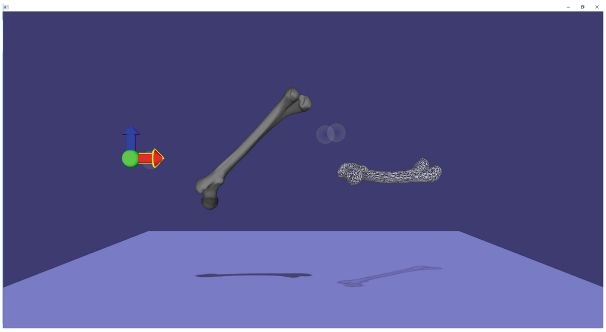


Fig. 1. Screenshot of rotating the object about x-axis.

User Controlled Axis Tool. We provide an additional user controlled axis for the following three reasons:

1. To avoid the use of system control mode metaphors, the axis appears only when users trigger it by dedicated gestures.
2. We are able to construct a set of intuitive and uniform gestures for the interactions based on the tool (G6, G7, G8).
3. Unlike the extra frame in FI3D [3], having an axis next to the object is a better choice since it is easy to reach and understand for manipulation.

Gestures Design. Based on the use of axis tool, aseries of multi-touch gestures are built, using up to four touch points to complete the 11 different interactions. Meanwhile, the advanced interaction was designed without influencing the usability of the basic interaction (G6). We divide the gestures into four types including translation,

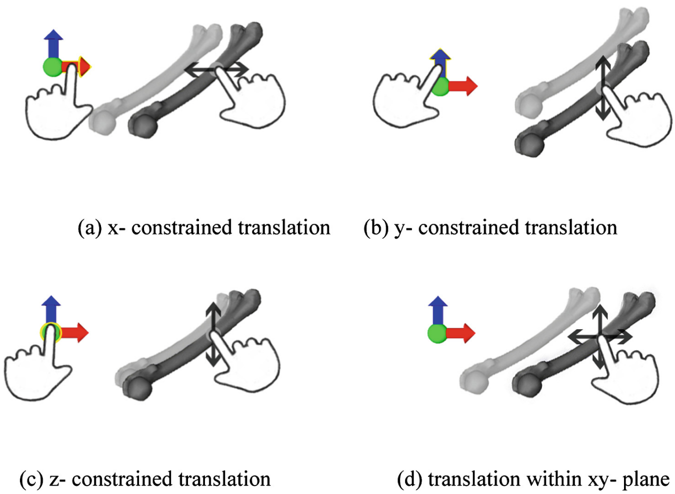


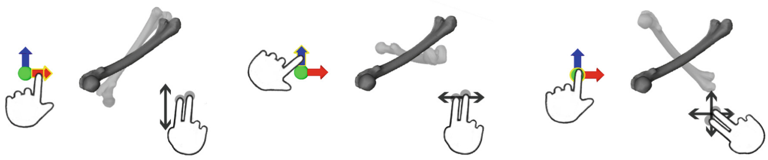
Fig. 2. Translation interaction.

rotation, scaling and viewpoint controlling. The gestures can be manipulated smoothly and users can switch between different types of gestures on the fly to realize different operations (G7).

Translation. For translation, the axis tool is triggered when one finger touches the model. After that users can move the object directly within the plane parallel to the view plane (Fig. 2d). In this condition, the model is "sticking" with the finger so as to perform a precise movement. By adding one more finger from the other hand to pick x-/y-/z-axis on the axis tool, the model will be moved separately in the three directions (Fig. 2).

Rotation. We design three types of interactions for orienting the model including constrained rotation, trackball rotation, and rotation around user-defined center.

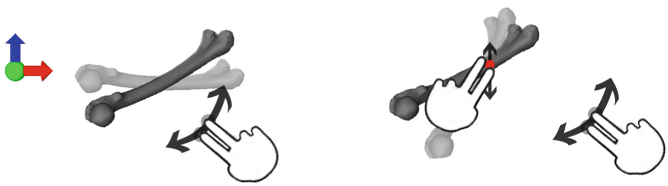
For constrained rotation, after triggering the axis tool by two adjacent touches on the screen, users can add one touch point on the axis tool to a specific axis that the model is expected to rotate around (Fig. 3). By dragging two touches on the screen, users perform trackball rotation (Fig. 4a).



(a) x- constrained rotation (b) y- constrained rotation (c) z- constrained rotation

Fig. 3. x-/y-/z-constrained rotation interaction

Additionally, we provide users the ability to rotate the model around a user-defined center (Fig. 4b). After triggering the rotation interaction by two touches on the screen, users can add two more touches to define the rotation center. The 2D location of the rotate center is defined at the first intersecting point on the object corresponding to ray-casting of the middle point of the two adjacent touches. By moving fingers up and down, users can adjust the rotation center in the depth direction. The rotation center is shown as a small red ball (Fig. 4b).



(a) trackball rotation (b) rotate around user-defined center

Fig. 4. Free rotation interaction (Color figure online)

Scale. For uniform scaling task (Fig. 5a), we choose the standard split-close and split-apart gestures from two hands to support large scale. We map the changes of distance between two touch points to the scale factor. With increasing distance, the object is enlarged, and vice versa.

Viewpoint Control. We provide unconstrained viewing angles for interactions so as to enable users to view and manipulate models from different angles (G5). This is important for the task of positioning and orienting an object relative to another. Since we manipulate 3D objects in the object coordinate system, changing the position of the camera will cause a problem when users want to align objects based on viewpoint changing. In order to realize viewpoint control as well as manipulate the object from different angles correctly, we decided to rotate the scene itself instead of changing the position of the camera. Users can swipe three fingers on the screen to view the models from different angles (Fig. 5b).

(a) scaling (b) viewpoint controlling

Fig. 5. Scaling and viewpoint controlling

4 User Study

A proven fact is that distinct input modalities such as mouse, keyboard and multi-touch display, each has its unique properties of benefits and disadvantages relying on the different application fields and the given interaction tasks [13]. To take an important step towards a better comprehension of pros and cons of these interaction techniques when dealing with structured 3D model, we report on a user study to compare our method with the standard mouse-based 2D interface. Meanwhile, we evaluate our method by users' performance and preference, particularly for our goals G8 and G9.

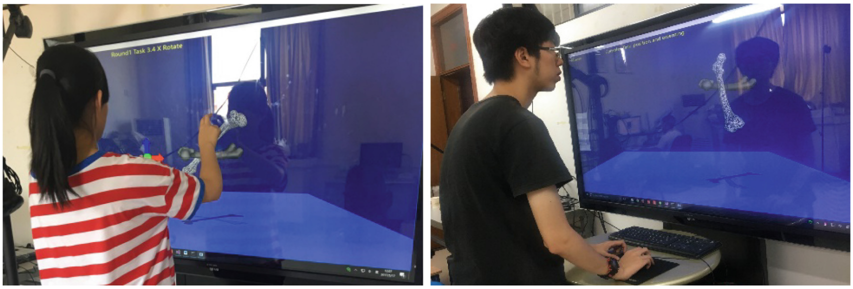
4.1 Participants

We invite twelve students (six male, six female) to attend the user study. Nine of them have experience of computer graphics or 3D computer games. Four students have experience with large multi-touch displays. Their ages varies from 20 to 25 (M = 21.538, SD = 1.45). All the users were right-handed.

4.2 Apparatus

We use a 65″ LED display with high resolution (1920 * 1080 pixels) and a G5 Multi-touch screen overlaying the display from PQLab, which supports 50+ touch points detectable simultaneously.

For standard 2D interface, we chose a 450 dpi average speed mouse and keyboard as interaction devices as shown in Table 1 instead of using widgets so that we can avoid computing the extra time and distance for mouse to reach the menus or buttons. The examples of setups for the experiments are shown above (Fig. 6). For touch interactions, users stood in front of the large display, while a table was placed in front of the screen for mouse/keyboard interactions condition. The application ran under Windows 10.



(a) touch interaction (b) mouse and keyboard interaction

Fig. 6. Example setup of touch and mouse/keyboard interaction.

Table 1. 3D manipulation of mouse and keyboard

Task	Mapping event
x-/y-translation	Left button drag
z-translation	Left button down + scroll up and down
x-/y-/z-constrained rotation	Key {x|y|z} down + right button drag
Trackball free rotation	Right button drag
Rotation around user-defined center	Define center: key {c} down + left button down + scroll up and down rotate: right button drag
Scaling	Scroll up and down
Viewpoint controlling	Key {left|right} down

4.3 Tasks

We tested eight tasks of independent basic manipulations and one complex integrated task of positioning and orienting one model according to another. The independent tasks test the performance of our interactions method separately with exact task instructions for 3D manipulation, while the complex integrated task requires users to think and make decisions on specific interactions they would like to use to translate and rotate the 3D object to the target location.

The eight independent tasks are the basic components of the integrated task. We tested three interaction techniques, translation, rotation, and scale. There were two translation tasks, one for translating on the x-/y-plane and the other for translating in 3D space. We tested constrained rotation and free rotation separately within a total of five rotation interactions. Table 2 below shows the description of each task.

Table 2. Independent tasks sequence per round

	Task	Description
1	x-/y-translation	Translate 3D object to match the target
2	x-/y-/z-translation	Translate 3D object to match the target
3	x-constrained rotation	Rotate 3D object to match the target
4	y-constrained rotation	Rotate 3D object to match the target
5	z-constrained rotation	Rotate 3D object to match the target
6	Trackball free rotation	Rotate 3D object to match the target
7	Defining center free rotation	Rotate 3D object to match the target
8	Scaling	Scale 3D object to fill the target

We asked users to position and orient a bone relative to a gray wireframe bone which indicates the target model as precisely and quickly as possible. We gave instructions of each task on the top of the viewport to guide users to perform corresponding interaction (Fig. 7). We calculated whether the bone has matched the desired position when the users' fingers left the screen or the mouse button was released, the task would be stopped automatically if matched. Participants were allowed to give up the task.

Fig. 7. Independent tasks setting.

For the integrated tasks, users were asked to position and orient the bone relative to the other without instructions. The bone is placed in a random place with a random orientation. To match the models, several interactions may need to be taken. The viewpoint control interaction was enabled during this task for users to view the scene and manipulate the object from different angles. In this condition, users needed to make decisions on which interactions they would like to use to match the models. They could use the eleven types of interactions we provided to complete the task.

4.4 Design

We used a repeated-measures design for the independent tasks with two input devices (multi-touch display, mouse/keyboard). Each user should complete tests separately for the two devices, half of the users tested mouse first while the other half tested multi-touch display first.

For the eight independent tasks, each user performed four rounds of tests, four times on each task and input devices. The tasks were always shown in the same sequence (Table 2). For each of the translation tasks, the target wireframe bone was placed in the middle of the screen, and the position of the manipulated bone was placed on 4 different starting positions varied by a Latin square. For rotation tasks, the bone and target bone were placed in the middle of the scene with two different rotation directions varied between rounds using the Latin square. For zoom tasks, the bone and target bone were placed on four unique positions with two different zoom factors. The first two rounds were considered as practice rounds which we do not consider them for the final analysis of the data.

In total, we have 12 users $\times$ 2 input devices $\times$ 8 tasks $\times$ 4 times $\times$ 4 rounds = 3072 interactions for the independent tasks. We first introduced the interaction mappings for each input devices to users before they began the test. Between the independent tasks of each input devices, we asked users to fulfill a questionnaire to give the subjective evaluation of the usability of the input device, including the degree of difficulty to complete each task and the memory difficulty for each interaction mappings on a seven-point Likert scale. After finishing all the independent tasks, users had to express their preference for the two input devices and their reasons.

For complex integrated tasks, there was no time limitation. We measure the errors between the models. Meanwhile, we record users' interactions and subjective assessment such as preference.

Eventually, when all tasks finished, users were asked to describe their overall feelings without consideration of the tasks, including their favors for the input devices, preference for constrained or free manipulation, and evaluation for the depth hints we provided by rendering model shadows and viewpoint controlling interaction. In addition, we collected their previous experiences of computer graphics, 3D game, and multi-touch large display. We also communicated with each user for advice and inspiration on our future work.

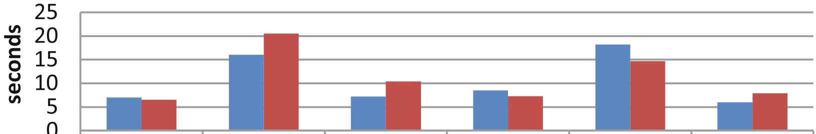
5 Results

We report the results of user study from three parts, the independent tasks' results, the complex integrated task results, and users' overall preference. In this section, we analyze the result and discuss our lessons learned.

5.1 Independent Tasks

We compared the average completion time for each task and each input device and present the results in Table 3. The task of x-/y-translation shows no obvious difference between mouse and touch. In contrast, mouse showed a significant difference when dealing with the task of z translation. The results for rotation showed a significant difference. For the tasks of constrained rotation, the mouse was much faster than touch. However, for tasks related to free rotation, the touch outperformed mouse, especially when dealing with rotate around user-defined center tasks. Mouse was faster when dealing with scale tasks.

Table 3. Task completion time of independent tasks.



	xy translate	z translate	constrained rotate	trackball rotate	rotate around point	scale
Mouse	6.98	16.07	7.22	8.52	18.23	6.01
Touch	6.53	20.54	10.42	7.27	14.71	7.89

From the questionnaire, users were asked to compare the two interactions of the input devices according to the difficulty of completing each task, the ease of remembering for each interaction on a 7-point Likert scale, the results are shown in Table 4. Overall, both interaction techniques rated high, we think we achieved our goal G8 and G9.

Table 4. The scores for each interaction.

Interaction	Mouse	Touch
x-/y-translation	6.91	6.83
z-translation	6.12	6.12
Constrained rotation	6.33	5.91
Trackball free rotation	6.25	6.25
Defining center rotation	4.87	5.58
Scaling	6.83	6.62

5.2 Complex Integrated Tasks

The position bias is defined as the ratio of the length of position deviation to the length of the actual path [11]. The result of position bias showed a significant difference between mouse and touch input devices. The position bias for mouse condition is

1.19% and 0.73% for touch. The rotation bias is described by the Tait-Bryan angle as shown in Table 5. This time, the mouse and touch show no significant difference. The total rotation bias for mouse condition is 6.15° and for touch is 5.54°. As the result, the manipulation is more accurate in touch condition.

Table 5. Rotation bias

Tait-Bryan angle	Mouse	Touch
Yaw	0.98	0.07
Pitch	0.56	0.28
Roll	4.61	5.07

The number of interactions users took to complete the tasks is 1963. For mouse condition, the average interactions number is 67 times and for touch condition, the average interactions number is 80 times. The switching between different interactions is smoother in the touch condition than mouse, with 1.16 s for touch and 1.32 s for mouse. Given this result, the goal G7 was achieved. We specifically count up the interaction times for free rotations and constrained rotations. As expected, we found that users preferred free rotation than constrained rotation for they took 412 times with 20.98% of the total for free rotation interactions and only 174 times with 8.86% of the total for constrained rotation interactions. What's more, the times of viewpoint control interaction is 440 times with 22.41% of the total interactions.

5.3 Overall Preferences

After completing all independent tasks with each input devices, we asked users to describe their preference when dealing with the time-limited independent tasks. Half of the users preferred mouse and keyboard interactions because they are familiar with mouse (3×), the speed of mouse is faster (2×) and less body movement (1×). While the other half preferred touch for the interactions are more intuitive (3×), the somesthetic feedback (2×) and the sense of immersive (1×).

After the more complex integrated task some users changed their mind. At this time, 83.3% users preferred touch interactions. The reasons were that they had more control over their model (4×), the transitions between different interactions were smoother (2×), more interesting (1×), more immersive (1×) and more intuitive (2×). Out of the two users who preferred mouse, they insisted that the mouse is more fast and precious.

After finishing all tasks, users were asked to report their overall preference. All users preferred touch interaction without considering other factors because it is more interesting. For constrained and free manipulations, 70% preferred free manipulations. The scores for the depth hints we provided by casting the shadows of models and the viewpoint controlling is 6.21, indicating that they thought the depth hints were quite useful.

6 Discuss and Lessons Learned

Touch and mouse input devices show no significant effect of x-/y-translation tasks. While for z-translation tasks, the reason might be that the scroll wheel of mouse is much faster and more accurate than moving touch points.

For constrained rotation tasks, the fact that touch outperformed mouse largely due to the hardware. Unlike pressure sensitive touch screen or capacitive touch screen, PQlab's multi-touch screen detects the touch points by infrared ray, some extra touch points are inevitably detected when users' fingers are close to the screen, which makes some failures on the gesture recognition process. When doing repetitive movements on the multi-touch display, it was difficult for users to move their fingers smoothly due to the friction force. From those result, we can learn that we should use at most two fingers from each hand for frequent interactions to reduce the friction. Besides, during the movement, the two-touches we use to control rotation were often detected as one touch point unless the users were intentionally careful to keep these gestures. We can learn from this that the two-touches are not suitable for frequently-used interactions. Instead, a conjoined touches gesture combined with two fingers together may solve this problem, which needs to be recognized technically. For the trackball rotation tasks, we think the reason for touch's slender lead is the same as x-y-translation tasks. While for tasks of rotation around the user-defined center, touch was faster than mouse. We conjecture that the interaction can be performed with a single multi-touch gesture, but needs three steps to perform with mouse and keyboard due to the limited input bandwidth of mouse, which is time-consuming. This result gives expression to the advantages of multi-touch interaction for the extra input bandwidth. As a user reported, we restrained the movement too much and the rotation is only triggered when the movement of two-near-touches is exactly vertical or horizontal. We believe that there is still room for the progress of the rotation interactions.

Due to the same reason as z-translation tasks, scale tasks for mouse was faster than touch for the scroll wheel was faster and precious. It is difficult to achieve fine adjustment. But as users reported, the scale on touch display is more intuitive and the increment is smoother.

The results of the complex integrated task reveal that the manipulation is more accurate in touch condition than mouse. We speculate that is because users were more concentrated on the models in touch condition. They did not need to spend extra effort on the keyboard. Thus, we think the touch interaction is suitable for tasks in which the precise manipulations are required. Besides, the touch interaction was more attractive than normal mouse interaction so that users would like to spend more time on the task. The switching between different interactions was smoother in touch condition. Therefore, we think we designed the interaction appropriately. According to our statistics, unsurprisingly most of the users preferred the free manipulation. The reasons might be that the free manipulation is more intuitive and this interaction offers users more freedom. The frequent use of viewpoint control interaction strongly demonstrated that providing unconstrained viewing angles for interactions is essential for 3D manipulations via 2D input and output devices.

7 Conclusion

In this paper, we presented a novel multi-touch approach for 3D object free manipulation. Our method is combined with a user-controlled axis tool and a set of elaborately designed multi-touch gestures. The method supports eleven operations for interacting with the 3D model in 7-DOF (goals G1, G2). We designed an advanced interaction for defining rotation center in 3D space (G3), which existing methods did not support. With the axis tool, gestures were designed in a simple and uniform way (G6). The switching between gestures can be on the fly (G7). Besides, we support 3D manipulations with unconstrained viewing angles, which provide users extra depth hints (G4, G5). We reported on a user study comparing our method with standard mouse and keyboard interactions. The results showed that our method was competitive when dealing with complex integrated tasks (G9). According to users report, the gestures are easy to learn and remember (G8). In the future, we would like to address the issues presented in the Lesson Learned section and design better multi-touch interactions for 3D manipulation.

Acknowledgements. We would like to thank Chenling Tang, Minda Chen and all the user study participants for their invaluable input. This work was supported by National Natural Science Foundation of China (Grant No. 61502132, No. 61502130) and Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education.

References

1. Isenberg, T.: Position paper: touch interaction in scientific visualization. In: Proceedings of the Workshop on Data Exploration on Interactive Surfaces—DEXIS, pp. 24–27 (2011)
2. Martinet, A., Casiez, G., Grisoni, L.: The effect of DOF separation in 3D manipulation tasks with multi-touch displays. In: Proceedings of VRST10 ACM Symposium on Virtual Reality Software & Technology, pp. 111–118 (2010)
3. Yu, L., Svetachov, P., Isenberg, P., Everts, M.H., Isenberg, T.: FI3D: direct-touch interaction for the exploration of 3D scientific visualization spaces. IEEE Trans. Vis. Comput. Graph. **16**(6), 1613–1622 (2010)
4. Martinet, A., Casiez, G., Grisoni, L.: The design and evaluation of 3D positioning techniques for multi-touch displays. 3D User Interfaces **30**, 115–118 (2010)
5. Kin, K., Miller, T., Bollensdorff, B., Derose, T., Hartmann, B., Agrawala, M.: Eden: a professional multitouch tool for constructing virtual organic environments. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1343–1352 (2011)
6. Roblesdelatorre, G.: The importance of the sense of touch in virtual and real environments. IEEE Multimed. **13**(3), 24–30 (2006)
7. Keefe, D.F.: Integrating visualization and interaction research to improve scientific workflows. IEEE Comput. Graph. Appl. **30**(2), 8–13 (2010)
8. Hancock, M.S., Vernier, F.D., Wigdor, D., Carpendale, S.: Rotation and translation mechanisms for tabletop interaction. In: IEEE International Workshop on Horizontal Interactive Human-Computer Systems, Tabletop IEEE, p. 8 (2006)
9. Reisman, J.L., Davidson, P.L., Han, J.Y.: A screen-space formulation for 2D and 3D direct manipulation. In: ACM Symposium on User Interface Software and Technology, Victoria, DBLP, pp. 69–78 (2009)

10. Hancock, M., Cate, T.T., Carpendale, S.: Sticky tools: full 6DOF force-based interaction for multi-touch tables. In: ACM International Conference on Interactive Tabletops and Surfaces, pp. 133–140 (2009)
11. Martinet, A., Casiez, G., Grisoni, L.: Integrality and separability of multitouch interaction techniques in 3D manipulation tasks. IEEE Trans. Vis. Comput. Graph. **18**(3), 369–380 (2012)
12. Mendes, D., Ferreira, A.: Evaluation of 3D object manipulation on multi-touch surfaces using unconstrained viewing angles. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) INTERACT 2011. LNCS, vol. 6949, pp. 523–526. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23768-3_73
13. Besançon, L., Issartel, P., Ammi, M., Isenberg, T.: Usability comparison of mouse, touch and tangible inputs for 3D data manipulation (2016)