A One-Handed Multi-touch Method for 3D Rotations

Douglas Scheurich, Wolfgang Stuerzlinger

A One-Handed Multi-Touch Method for 3D Rotations

Douglas Scheurich[1] and Wolfgang Stuerzlinger[1]

[1]York University, Toronto, Canada

{dougsch, wolfgang}@cse.yorku.ca

Abstract. Rotating 3D objects is a difficult task. We present a new rotation technique based on collision-free "mating" to expedite 3D rotations. It is specifically designed for one-handed interaction on tablets or touchscreens. A user study found that our new technique decreased the time to rotate objects in 3D by more than 60% in situations where objects align. We found similar results when users translated and rotated objects in a 3D scene. Also, angle errors were 35% less with mating. In essence, our new rotation technique improves both the speed and accuracy of common 3D rotation tasks.
Keywords: 3D rotations, 3D user interfaces, multi-touch, tablets

1   Introduction

The most common interaction tasks in three dimensional, 3D, virtual environments are navigation, object selection, and manipulation, such as translation and rotation. Object translation positions objects within the scene, whereas rotations orient objects. Despite being a fundamental task, there is no established standard for rotating 3D objects. One issue is that there is no "best" input device for 3D manipulation. In practice users use two-dimensional, 2D, pointing devices, including the mouse and touchscreens. 2D pointing devices offer good control of 2 degrees of freedom (DOF). However and in a 3D environment, control over 3 DOF is required for translations or rotations (yaw, pitch, and roll), or 6 DOF for both simultaneously. In many user interfaces this is handled through combinations of different widgets or touch gestures, i.e. through a combination of 2 DOF and 1 DOF controls. A mouse button is often assigned to control 2 DOF rotations. The third DOF is typically controlled via a modifier or the scroll wheel.

The computer-aided design program Solidworks recently introduced a simple form of object mating. There, clicking on a specific surface of an object followed by a click on another surface snaps these two together, so that the first surface "mates" onto the second. However, this simple mating technique may leave objects in positions where they interpenetrate.

Mating is a simple way to orient objects. Thus, we were surprised to discover that there was no documented work on mating methods that avoided or resolved colli-

sions. This encouraged us to explore the idea of mating for rotating objects on a touchscreen interface while avoiding object interpenetration.

2   Previous Work

Relevant other work on 3D rotations uses either 2D or 3D input devices. For 2D devices we first discuss the mouse and then touchscreen methods.

With a mouse, Bade et al. [1] evaluated four different methods, Bell's [2] and Shoemake's [3] virtual trackballs, and two variants of the Two-Axis Valuator [4], in a user study. They found the Two-Axis Valuator to be best. Jacob et al. [8] investigated inspection tasks requiring rotations and found a similar result. However, the tasks in both of these studies required only 2DOF rotation control! Again, investigating only a subset of all 3D rotations, Partala [7] found virtual trackballs to be superior. Zhao et al. [5] recently investigated tasks that require full 3D rotation control. They did not identify significant differences between Bell's and Shoemake's trackballs and the Two-Axis Valuator.

Reisman et al. [9] presented a multi-touch method to control the position and rotation of 3D objects. The solver-based method tries to keep the object stable under the fingers. Yet, the result is not always predictable and rotations are often limited to 90 degrees in two of three directions (e.g. when a cube is facing the viewer). Rotations then require clutching. Martinet et al. [11] used this method in their 6 DOF manipulation system. Hancock et al.'s [10] technique permits 1 DOF rotation with a 2-finger "rotate" gesture and controls the other 2 DOF with a two-handed 2+1 finger gesture. Kin et al. [12] controls 2 DOF rotations with a single finger and the third DOF with a two handed gesture.

Bier [19] developed a system called Gargoyle3D based on the idea of "snap-dragging". This system uses a general-purpose gravity system combined with snapping in 3D to enable users to accurately position and orient objects in 3D. In the user interface Gargoyle3D offered more than 40 distinct commands to achieve these operations. This level of complexity makes it infeasible for tablet computing. Related work on SKETCH by Zeleznik et al. [20] requires at a minimum a three-button mouse and a keyboard for a modifier. While less complex than Bier's system, the sheer number of mouse and keyboard combinations again does not lend this idea to a simple or efficient implementation on a tablet.

Zhai et al. [13] and Boritz et al. [14] investigated 3D rotations as part of 6 DOF docking tasks. Hinckley et al. [6] compared 3D rotation techniques using 2D and 3D input devices and found that those based on 3D input devices were approximately a third faster. Recently Kratz et al. [15] proposed a system where the orientation of a user's hand controls 3D rotations. Their comparison with a virtual trackball found also a ~30% improvement. Liu et al. [21] proposed a system for full 6 DOF manipulation using four distinct one or two-finger touch gestures. However, the authors note that their system is not well suited for fine control of model transformations.

3 A New Multi-Touch 3D Rotation Technique

Tablets and tablet use is now a permanent addition to computing, considering the sustained explosive growth of the tablet industry in the last few years. Especially since the advent of smaller tablet form factors, such as the 7.9" iPad mini, one cannot assume that a tablet is going to be used with two hands simultaneously. Such devices are used almost anywhere by holding it in one hand while interacting with the system with the other. A great example is a video on the Autodesk website showcasing their Inventor Publisher Mobile Viewer[1] application for iOS and Android. Parts of these videos show an engineer working on a tablet. The engineer never once puts the tablet on a table or desk to work, and holds it with one hand while working with the other the entire time. Given this constraint, we noticed a lack of touch-based systems that require only a single hand for 3D manipulation. This also means that all previously presented, touch-based 3D rotation methods, which require two hands, are not ideally suited for tablet use.

We designed our new technique explicitly for one-handed use. We also wanted to avoid the unpredictable nature and the limitations of Reisman's [9] approach. Moreover, we were inspired by a recent set of observations on 3D user interface design by Stuerzlinger et al. [16]. They point out that in the real world the vast majority of objects are aligned with planes or other objects. The reason is that (almost) all objects are in contact with others on our planet and "floating" objects are a rare exception. Tables usually stand on floors; pictures are attached to walls; light fixtures to the ceiling. Many such objects have only one free rotational DOF in their "normal" placement. In other words, truly random orientations are the exception in the real world. Consequently, we focus on user interfaces that are optimized for these pervasive cases and design our technique accordingly.

The idea of "mating" two surfaces fits this observation very well, except that naïve mating easily results in object interpenetration. Mating the seat of a chair "onto" the ground would put the backrest of the chair into the ground, which novices often find confusing [16]. This is demonstrated in Figure 1 below. Therefore, we enhance basic mating by always putting the chair into a position that avoids collisions, while keeping the seat plane parallel to the ground. As an added bonus mating also translates the object, which may lead to additional timesaving. Given that our enhanced form of mating always put objects into contact, we globally prevent objects from "floating" in our system. This limits the system to 2 DOF positioning, but also matches the capabilities of touchscreens better as fewer DOFs need to be controlled and simplifies the user interface. Objects can still assume any 3D rotation in our system, which is the main focus of our work here.

---

[1] http://m.autodesk.com/mobile/servlet/product?siteID=17221380&id=17774143
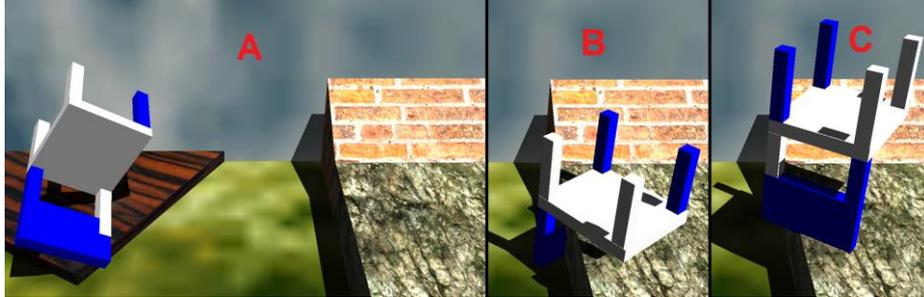This website is accurate and available as of January 20th, 2013

Fig. 1. The original orientation is shown in panel A. The user seeks to mate the seat of the chair from the left to the stone ledge on the right. Panel B illustrates the result of mating without preventing collisions. Such an operation leaves the chair in a confusing position, where the plane of the seat of the chair is parallel with the plane of the stone. However, the chair interpenetrates the floor. Panel C demonstrates mating with our collision resolution. The plane of the chair seat is still parallel with the stone surface. Yet, the chair has been "raised" so that it is in a collision free position.

3.1 Two Multi-Touch Rotation Methods

As discussed above, previous research did not identify a clearly superior 3D rotation technique, especially one with a limited number of input controls suitable for deployment on a tablet. Hence, we base our multi-touch system on the Two-Axis Valuator to directly control 2 DOF rotations. To control the third DOF we use a different form of multi-touch gesture compared to previous work. We implemented two variations for this. One is designed for systems that permit only 3D rotations, the other for systems that support both positioning and rotation.

3.2 Rotation Only Method

Our first method targets 3D rotations only. It interprets a single finger drag as Two-Axis Valuator manipulation. This is usually done with the index finger. Dragging the finger left and right will rotate the object about the *y axis* while dragging the finger up and down will rotate about the *x axis* of the camera coordinate system. A two-finger touch rotates around the view direction (i.e. the *z axis* of the camera coordinate system, see Figure 3 below). Here we implement a new technique: if one finger stays in place and a second "scrolls" this is interpreted as a rotation. We found users preferred to accomplish this in one of two ways. When rotating objects in the middle of the screen, putting the index finger down and flicking the thumb left or right below it is a natural way to access this functionality. When operating on the right edge of the screen, placing the index finger down and scrolling with the middle finger seems most natural. Placing the middle finger down and scrolling with the index works well when working near the left edge of the screen. To summarize, a single finger drag rotates about the *x* and *y axes*, while a single finger anchored in place (i.e. index finger) plus a finger dragging (i.e. the thumb) rotates the about the *z axis*. Double tapping a point

on the object will use the enhanced mate functionality to mate the specified surface of the object with the plane behind it, as illustrated in Figure 2 below.
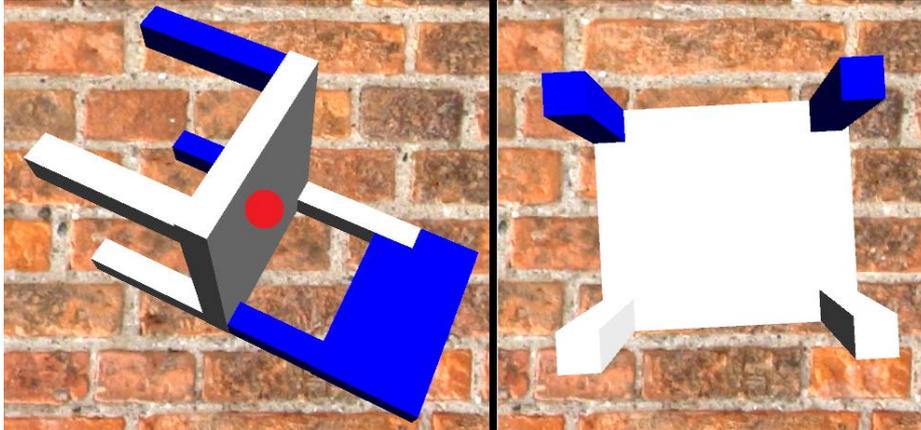

Fig. 2. Illustrating of mating for 3D rotations using method 1. In this image we can see the brick wall as a background object behind the chair. By double tapping on the seat of the chair, marked with the red dot, the object's surface that corresponds with the double tap location (i.e. the chair seat), will become aligned with the plane of the object behind it (i.e. the brick wall). Consequently and by double tapping the seat of the chair, the chair is automatically rotated so that the seat plane is parallel with the background surface. This figure also illustrates a trial in the first phase of the experiment. The user is able to manipulate the object on the left. The task is to match the pose of the left object with that of the right object.

3.3 Rotation and Translation Method

Our second method targets both rotations and translations. As a result this method is more targeted at "real world" usage compared to the first method, since in an interactive environment some (or many) objects may be moveable. In the second method single finger movements control the (constrained) translation of an object along the surfaces of the scene using a variant of the technique presented by Oh et al. [17]. Basically, objects always snap to and/or slide on the surface beneath them. For example, imagine wanting to drag a block around a staircase. If you are dragging the block down the staircase, as you move your finger the block will slide across the surface of each step, and when it reaches the end of a step, it will snap from that step' surface onto the surface of the next step and continue to slide along that, etc.
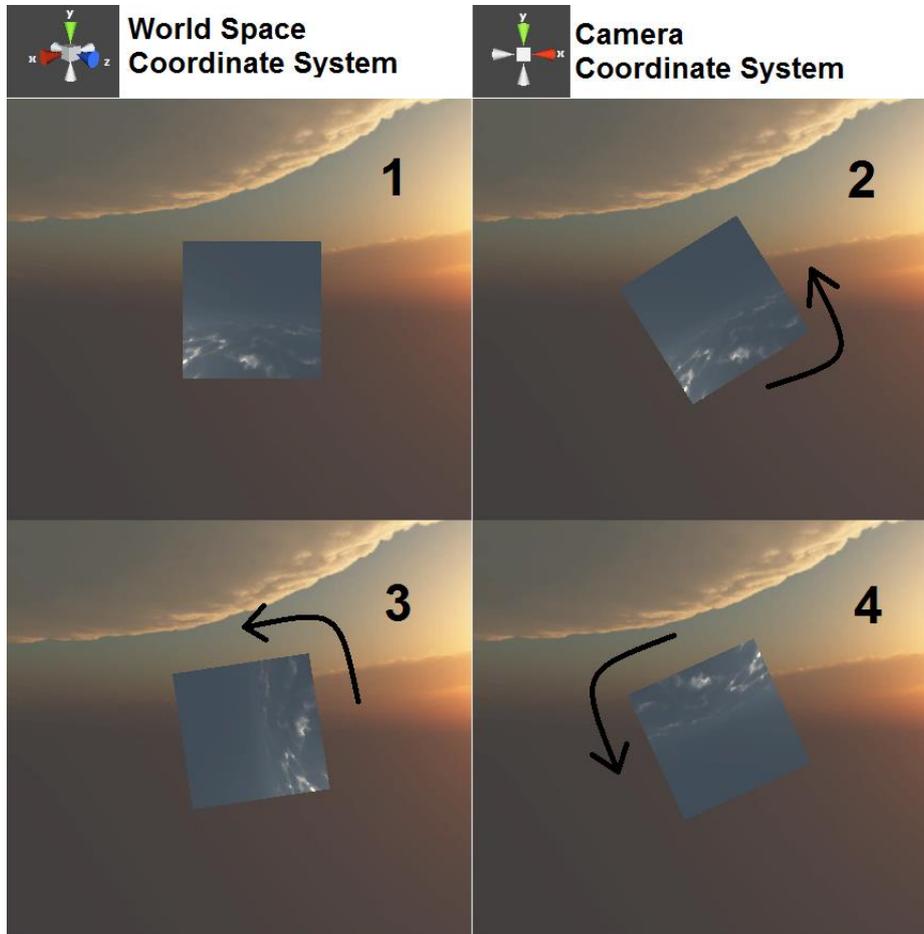
Fig. 3. For all frames the world space and camera coordinate systems are shown at the top of this illustration. The user is always looking down the *z-axis*. Frames 1 through 4 illustrate rotating a cube along the view direction (i.e. the *z-axis*). We can clearly see that only the camera coordinate system influences the rotation of the object.

Rotating objects in the second method is similar to that in method one described above, except there is one additional finger for each technique. Thus a two-finger drag gesture, typically used with two fingers side-by-side, controls the rotation through the Two-Axis Valuator. Moving two fingers left or right rotates the object about the *y axis* while moving them up or down rotates the object about the *x axis* of the camera coordinate system. With two-fingers in place, moving a third finger rotates the object around the view direction (i.e. the *z axis* of the camera coordinate system). Users commonly use the middle and index finger in place while moving the thumb below them, especially when rotating objects in the middle of the screen. When space is constrained at the edge of the screen, a different possibility is to use the middle, index, and to move the ring finger to rotate objects. A single finger tap on a surface of

an object followed by a finger tap elsewhere in the scene mates the two surfaces. This is illustrated in Figures 4 and 5 below.
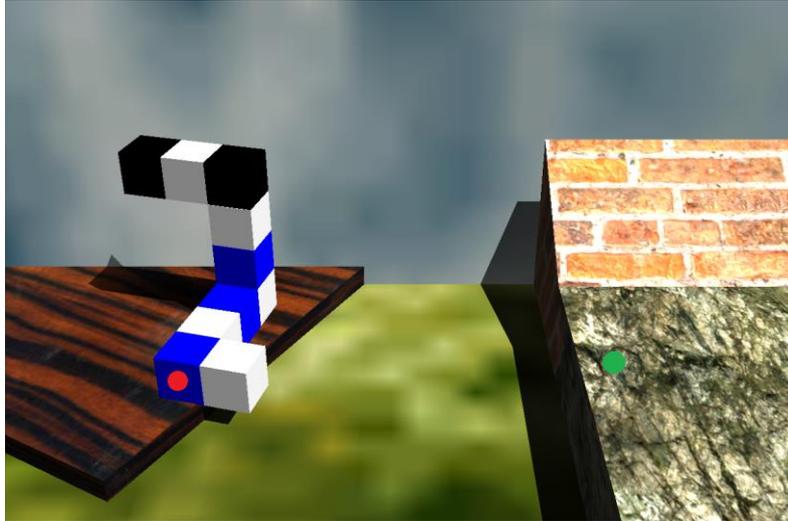

Fig. 4. The initial scene before performing a mating operation. By taping the Shepard-Metzler object at the location indicated by the red dot, and then taping elsewhere in the scene, say the green dot on the right hand side, the object is simultaneously rotated and translated (i.e. mated) onto the new surface. The final position is illustrated in Figure 5 below.
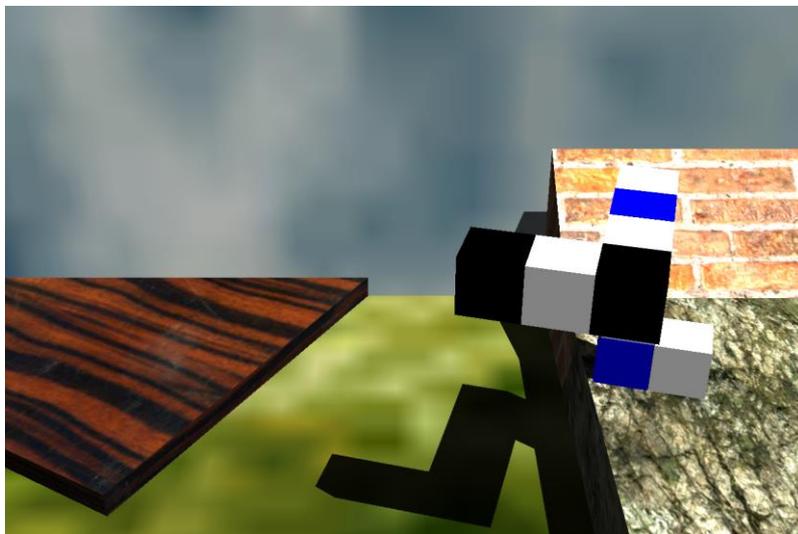

Fig. 5. Final position of the Shepard-Metzler object after the mating operation.

If mating results in a collision, the object is pushed away from the target surface until the object is only in contact. This was illustrated in Figure 1 above. The reason

for this is that in the real world objects very rarely interpenetrate and we designed our system to conform to this. If in a 3D world the user wants to place a chair on the ground, it is unlikely their goal is to have parts of the chair above the ground and parts of the chair disappearing into the floor. The effect of pushing objects away from the surface is relative to the user and the object. For example and if an object is mated onto a wall and interpenetrates it, the object will be pushed towards the user so that the object remains visible and inside the scene. The alternative of pushing it away from the user and the object becoming potentially invisible, is much more confusing. This effect is illustrated in Figures 6 and 7 below.
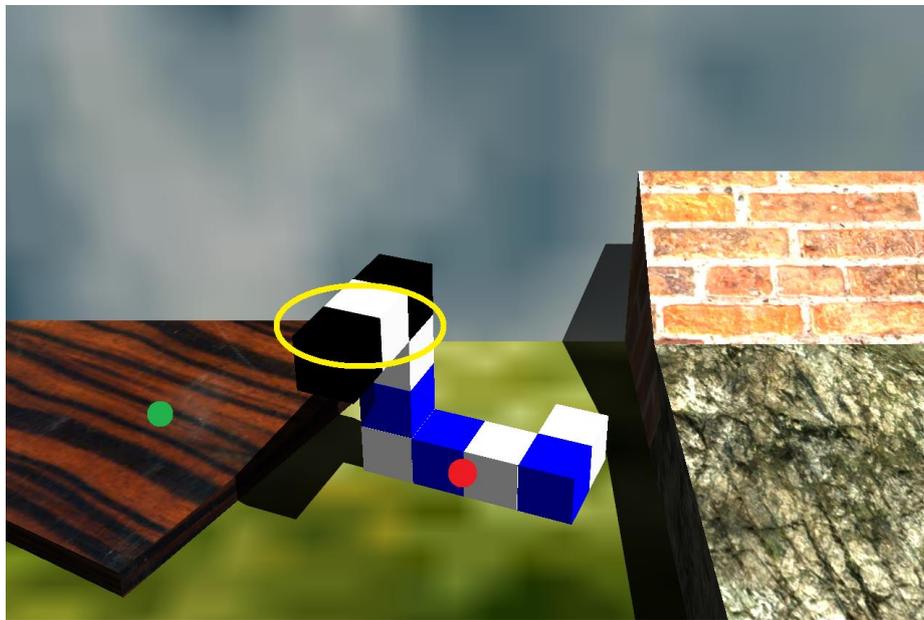

Fig. 6. In this example the Shepard-Metzler object is mated with the wooden floor to the left. If the user taps the object at the red dot and then taps the floor at the green dot, a naïve mating operation would result in the circled yellow part of the object disappearing into the wooden floor.
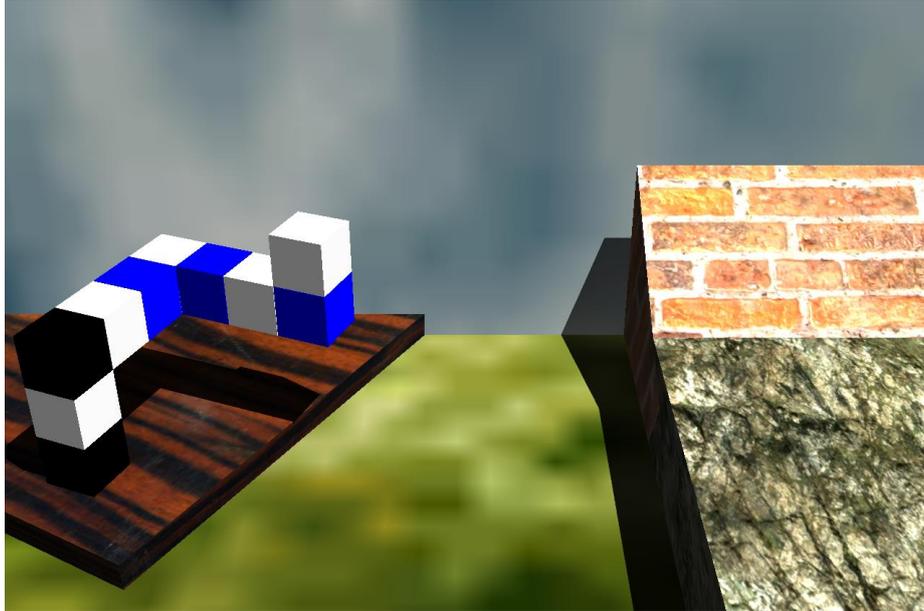
Fig. 7. The final position of the object in the scenario of Figure 6. The object has been pushed upwards, so that no parts interpenetrate the floor.

The movement from the initial position to the final position is animated to help users understand the result and to prevent them from having to visually reacquire the object position and orientation. This is especially helpful if an object is being moved over large distances over the scene. Also, before the objected is moved, our system pre-determines the final position and checks for collisions. If a collision is found, our system pushes the object away from the surface until there are no collisions, and this becomes the new final position of the object. Since the rotation and translation of the object is animated, if the object ends up in an unintended position or location, say as the result of a tap at the wrong location, the user will be able to see this visually and make appropriate corrections.

Moreover, a recently mated object is temporarily constrained. The system then rotates the object in the local coordinate system of the target surface. This enables users to mate an object onto any surface and then to quickly adjust the remaining DOF using either a two- or three-finger rotation. This permits users to quickly snap an object, for example a coffee cup, onto a table and then to turn the cup on the table to correctly orient the cup. The initial mating operation will ensure that the bottom of the cup is in complete contact with the surface, ensuring that it is not floating, leaving the user with the task of controlling only a single DOF to correctly orient the cup. This is substantially easier than having to always control all 3 DOF.

4 Methodology

4.1 Participants

Twelve paid volunteer participants were recruited from the local university campus and city. The age of the 6 male and 6 female participants ranged from 19 to 35 years ($mean = 26.17$, $SD = 4.47$). All had never participated in a 3D study before. All were right handed and preferred to use the tablet with their right hand. 3D video game usage was between 0 to 4 hours per week ($mean = 1.92$, $SD = 1.62$). Potential participants were asked in advance of the study if they had previously participated in a 3D study from our lab (even with another examiner) and we excluded people who indicated they had performed a user study involving rotating or translating 3D objects.

4.2 Apparatus

We conducted the study on an 8" Gadmei T883-3D tablet with Android 4.0.3. A widescreen monitor on a desktop computer was used in the second half of the study (i.e. during method 2) to display target scenes. The reasoning for this is explained below in the methodology section. We built a variety of commonly recognizable 3D objects, as well as several inspired by the Shepard-Metzler test [18]. After a pilot study we decided on a car, chair, dog, and one Shepard-Metzler object as shown in Figure 8 below. Colorings were introduced to disambiguate poses as, for example, a view onto the bottom of a chair with legs with identical colors does not reveal the full 3D rotation of the whole object at a glance. The white parts of objects highlighted in different colors for feedback. This reasoning for this is explained in more detail below.
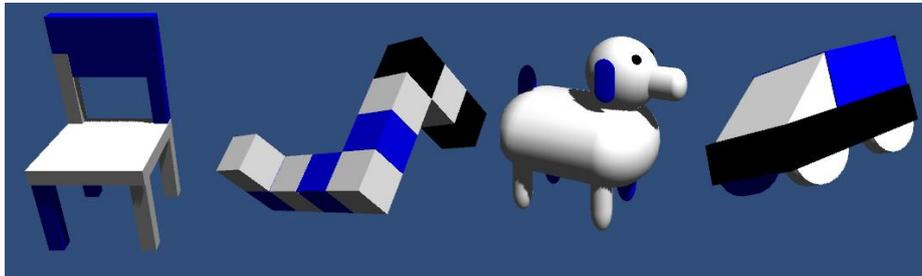

Fig. 8. Objects selected for use in both studies.

4.3 Procedure

The study was conducted in a quiet room with the investigator, the sitting participant, and at most one other student present. The software was first configured to the participant's handedness. This affected the first part of the study where the rotating object was on the side of the user's preference and the target displayed on the other one.

Then the study was explained along with a system demonstration. All participants acknowledged they understood how the system worked and had no questions.

Overall we used a 2x2 within subject design with 2 phases. The first phase targeted only 3D rotations, whereas the second investigated rotations with constrained translations. The conditions were mating enabled or not, and surface aligned target orientations or not. These conditions were counterbalanced over all trials. In each set of 48 trials, subjects were asked to rotate 4 models 6 times with either mating enabled or not. Targets were aligned three of these six times, while the others had random target orientations. The order of each of these 24 trial blocks was determined using a Fisher-Yates shuffle. To determine the starting pose of the rotatable object we used two randomly shuffled copies of a list of 12 difference angles: 15, 30, 45, … , 165, 180 degrees. Each copy of the list matches to targets being aligned or not. To compute the final starting 3D orientation we first defined a rotation axis by generating a random point on a unit sphere. The user rotatable object was then rotated "back" from the target orientation about this axis by the angle chosen above. The participants' task was to rotate the object to within a quaternion angle of 10 degrees from the target orientation. Users could not abort trials.

Figure 2 above illustrates a trial in the first phase. Participants were presented with 2 objects. The target object on the left (or right) could not be manipulated and was the shown in the desired orientation. The object on the other side was rotatable. When the user touched the rotatable object the white segments turned magenta to indicate selection. When the object was within the 10 degree limit the white segments turned green to signal the user that the object had been rotated within an acceptable accuracy tolerance (the quaternion angle of 10 degrees mentioned above). A button then appeared in the upper corner of the screen on the side of the un-moveable object. Users pressed this button to move to the next trial, where it disappeared until the next object was correctly aligned. In phase one the focus was only on rotating objects. Thus, it was feasible to display both the target orientation and the user controlled object on the tablet screen side-by-side.

The task in the second phase shown in Figures 4 and 5 was to both translate and rotate the object into the target pose, a 5 DOF task. Said target pose was displayed on a second, desktop monitor in this phase. Since users performed tasks involving translations and rotations, which requires more space for interaction and scene display, we were unable to display both an interactive scene and the target scene on the same device. Hence the target scene was displayed on the monitor, giving the user a larger space to translate objects on the tablet. For example, the tablet would display the scene from Figure 4, while the desktop monitor would display the scene from Figure 5. The participants' task was then to orient the object from his scene to match the scene displayed on the desktop.

Here, white parts of the object highlighted in cyan when within 10 degrees of the correct orientation, in yellow when within $1/50^{th}$ of size of the scene, and green when close to the correct pose. When the user was close enough, a button appeared in the upper right corner of the tablet screen which when pressed went onto the next trial. When pressed, the scene on the computer monitor changed. Thus users were not able to see the next trial in advance of completing the current trial.

5 Results

We found that using our mating system decreased rotation times substantially. According to a repeated measures ANOVA and in the 3D rotation task investigated in phase 1, there were significant effects of mating on completion time ($F_{1,11} = 23.06$, $p < .001$) and target alignment ($F_{1,11} = 100.92$, $p < .0001$). Both mating and aligned targets were significantly faster. There was also a significant interaction between the conditions. A Tukey-Kramer posthoc test shows that aligned scenarios with mating were approximately 65% faster than all other combinations. Figure 9 below illustrates average completion times. The results for the error angles show a significant effect for mating ($F_{1,11} = 93.83$, $p < .001$) and also confirm that aligned targets were positioned significantly more accurately ($F_{1,11} = 63.75$, $p < .001$).
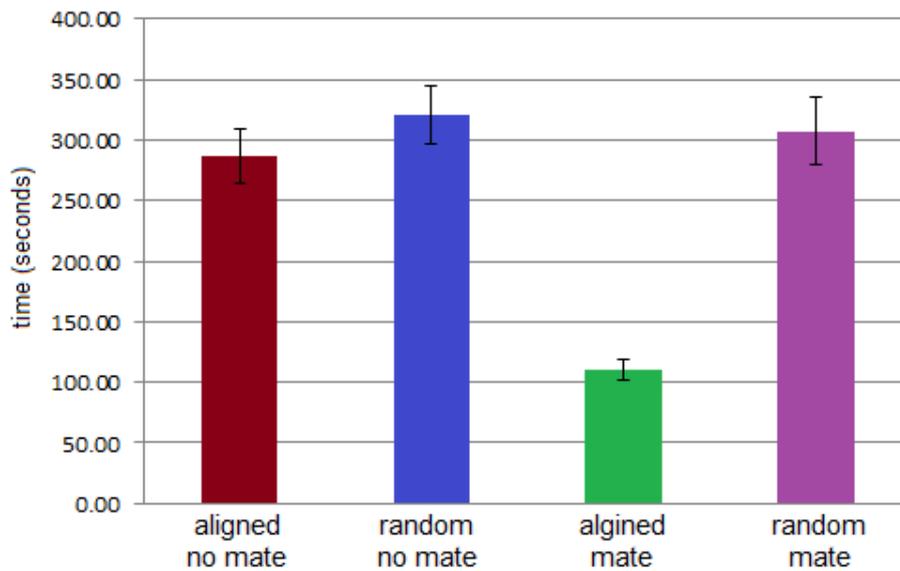

Fig. 9. Mean completion times for study 1 with standard error shown.

In the 3D translation and rotation task in phase 2 there was again a significant effect of mating ($F_{1,11} = 37.68$, $p < .0001$) and aligned targets ($F_{1,11} = 61.7$, $p < .0001$) on task completion time. There was also a significant interaction between them and Tukey-Kramer identifies that aligned scenarios with mating were completed approximately 64% faster than all other combinations. Figure 10 below illustrates the average completion time per participant. In this phase, there was a significant effect on error angles for both mating ($F_{1,11} = 108.75$, $p < .0001$) and aligned targets ($F_{1,11} = 24.71$, $p < .0005$), as well as a significant interaction between them. Tukey-Kramer reveals that aligned objects were posed in the mating condition approximately 35% more accurately in terms of error angles compared to all other combinations.
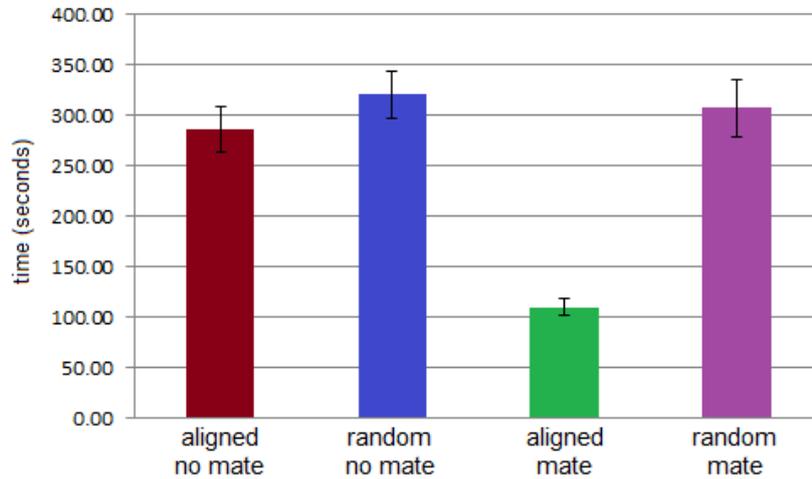
Fig. 10. Mean completion times for study 2 with standard error shown.

5.1 Discussion

Our enhanced mating-based 3D rotation technique substantially decreases the time required to match aligned target orientations by 64% or more, while significantly improving accuracy. Given that most objects are aligned to others in many real world scenarios, this is substantial and exceeds all improvements found in previous work. Participants generally found either the chair or the dog the easiest object to rotate. Unanimously, the Shepard-Metzler model was judged the most difficult. Participants found the mating interface simple to use and all perceived it as faster. Although our original design for the flick gesture was targeted at thumb flicking, about half the participants preferred to use the ring finger instead for this technique.

Although this is not currently implemented in our system, we can easily adapt our method to enable mating even when there are no other surfaces present. For this we would simply add a (potentially invisible) back plane surface that is orthogonal to the view direction and proceed with mating to said surface. Optionally, this surface may be partially transparent to offer visual guidance to the user. One limitation of our current implementation is that we do not support mating objects to two (or more) surfaces simultaneously. This is an area we plan to investigate in future work.

5.2 Conclusion

We presented a new multi-touch 3D rotation technique based on mating to accelerate common tasks. It is targeted at one-handed touchscreen use, especially on tablets. Our user study revealed that the new technique improves manipulation times by more than 60% for common 3D rotation tasks. Rotation accuracy is significantly improved

as well. In the future, we plan to investigate the performance of this mating technique for full 6 DOF manipulation tasks.

6　References

1. Bade, R., Ritter, F., Preim, B., Usability comparison of mouse-based interaction techniques for predictable 3D rotation, Smart Graphics 2005, 138-150.
2. Bell. G, Bell's Trackball, http://members.tripod.com/professor_tom/index.html, 1988.
3. Shoemake, K., Arcball: A user interface for specifying three-dimensional orientation using a mouse, Graphics Interface '92, 151-156.
4. Chen, M., Mountford, S. J., Sellen, A., A study in interactive 3-D rotation using 2-D control devices, SIGGRAPH '88, 121-129.
5. Zhao, Y. J., Shuralyov, D., Stuerzlinger W., Comparison of multiple 3d rotation methods, VECIMS 2011, 13-17.
6. Hinckley, K., Tullio, J., Pausch, R., Proffitt, D., Kassell, N., Usability analysis of 3D rotation techniques, UIST '97, 1-10.
7. Partala, T., Controlling a single 3D object: Viewpoint metaphors, speed and subjective satisfaction, INTERACT '99, 486-493.
8. Jacob, I., Oliver, J., Evaluation of techniques for specifying 3D rotations with a 2D input device, HCI '95, 63-76.
9. Reisman, J., Davidson, P., Han, Y., A screen-space formulation for 2D and 3D direct manipulation, UIST '09, 69-78.
10. Hancock, M., ten Cate, T., Carpendale, S., Sticky tools: Full 6DOF force-based interaction for multi-touch tables, ITS '09, 133-140.
11. Martinet, A., Casiez, G., Grisoni, L., The effect of DOF separation in 3D manipulation tasks with multitouch displays, VRST '10, 111-118.
12. Kin, K., Miller, T., Bollensdorff, B., DeRose, T., Hartmann, B., Agrawala, M., Eden: A professional multitouch tool for constructing virtual organic environments, CHI '11, 1343-1352.
13. Zhai, S., Milgram, P., Quantifying coordination in multiple DOF movement and its application to valuating 6 DOF input systems, CHI '98, 320-327.
14. Boritz, J., Booth, K.S., A study of interactive 6 DOF docking in a computerized virtual environment, VRAIS '98, 139-146.
15. Kratz, S., Rohs, M., Guse, D., Müller, J., Bailly, G., Nischt, M., PalmSpace: Continuous around-device gestures vs. multitouch for 3D rotation tasks on mobile devices, AVI '12, 181-188.
16. Stuerzlinger, W., Wingrave, C., The value of constraints for 3D user interfaces, Virtual Realities: Dagstuhl Seminar 2011, 203-224.
17. Oh, J.-Y., Stuerzlinger, W., Moving Objects with 2D Input Devices in CAD Systems and Desktop Virtual Environments, Graphics Interface 2005, 195-202.
18. Shepard, R., Metzler. J., Mental rotation of three dimensional objects, Science 1971. 71(972):701-3.
19. Bier, Eric A., Snap-Dragging in Three Dimensions, Proceedings of the 1990 symposium on Interactive 3D graphics, ACM: New York, 193-204.
20. Zeleznik, Robert C., Herndom, Kenneth P., and Hughes, John F., SKETCH: an interface for sketching 3D scenes, ACM SIGGRAPH 2007, ACM: New York, Article 19.
21. Liu, J., Au , O. K-C., Fu, H., and Tai, C-L., Two-finger gestures for 6dof manipulation of 3d objects, Comp. Graph. Forum 31, 7pt1 (September 2012), 2047-2055.