Usability and Performance of Mouse-based Rotation Controllers

Steven Rybicki*
Department of Computer Science
University of Cape Town

Brian DeRenzi†
Department of Computer Science
University of Cape Town

James Gain‡
Department of Computer Science
University of Cape Town

Abstract

Rotation controllers are used to interactively orient models in many important applications in 3D computer graphics and visualisation. Unfortunately, previous studies do not provide clear guidance on which rotation controller to use in a particular situation, either because they assess performance measures and rotation tasks in relative isolation or because they do not achieve statistical significance.

In this paper, we present the results of a broad quantitative user experiment ($n = 46$) to compare the three most prevalent rotation controllers (Arcball, Two-Axis Valuator, and Discrete Sliders) according to both speed and accuracy across two classes of tasks (orientation matching and inspection). While we found no significant differences between Arcball and Two-Axis Valuator, Discrete Sliders were found to be significantly more accurate for simple orienting tasks (a medium to large effect), but slower across all tasks (a small to medium effect, median approximately two seconds). Thus, a Discrete Sliders controller is better suited to situations where fine-grained accuracy is valued over speed and in other instances, e.g., inspection, either an Arcball or Two-Axis Valuator is appropriate.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques

Keywords: 3D User Interfaces, 3D Rotation, 3D object manipulation, Arcball, Discrete sliders, Two-Axis Valuator, Usability

1 Introduction

Three-dimensional scene manipulation is integral to many significant tasks, including creating virtual scenes for games and film, and examining 3D models in applications such as medicine, architecture, computer aided design, and the exploration of virtual worlds [Bowman et al. 2004]. Such applications enable a user to select, position, and rotate objects into a desired orientation [Wenjun 2008]. However, these operations are not all equally fast: users can position objects in $2 - 3$ seconds [Ware and Rose 1999] but orientation takes upwards of 10 seconds [Hinckley et al. 1997]. Interestingly, an equivalent task with a physical cube can be completed in under a second [Wang et al. 1998]. This suggests there is room to improve rotation controllers, which would improve 3D scene manipulation.

One possible improvement would be to match rotation controllers with classes of tasks for which they excel. Most systems offer one rotation controller for all tasks, but studies suggest that controllers that perform well for orientating models, such as a Virtual Trackball [Chen et al. 1988], do not perform as well for "inspecting" a model [Bade et al. 2005].

Rotation is a ubiquitous task in 3D visualisation, modeling and animation, be it orbiting a scene, orienting a cutting plane, aligning scene elements or rotating kinematic frames. While there have been a number of previous experimental evaluations of rotation controllers, further study is warranted, given the importance of model and view rotation, and the difficulty that previous studies have had in finding statistically significant differences. With this in mind, we performed a quantitative evaluation of three popular mouse-based rotation controllers (Arcball, Two-Axis Valuator, and Discrete Sliders) with two classes of tasks. These controllers were chosen based on performance in other studies as well as their use in popular 3D modeling suites. Our aim was to determine whether one of the controllers tested was more effective or usable than another for a orientation or inspection tasks.

Our contribution over previous work is that we test the same set of prevalent controllers across two sets of tasks (rather than a single task type) and measure both speed and accuracy (rather than biasing towards one or the other). Our experiment includes complex orientation-matching that we believe is more representative of real-world use. Our study design (a randomised single factor repeated measures experiment with 12 sub-tasks and 46 subjects) allows us to detect medium to large effect sizes with statistical significance. Additionally, we look specifically at confounding factors.

We found that Discrete Sliders were significantly more accurate for simple fine-grained accuracy tasks ($r = 0.29$, $p < 0.001$) than Arcball but took longer (median approximately 2 seconds) to obtain this accuracy. No significant difference in accuracy was found between the controllers for Complex Orientation or Inspection tasks. However, the Two-Axis Valuator and Arcball are faster for Complex Orientation ($r = 0.28$ and $r = 0.33$, respectively, $p < 0.01$) and inspection tasks ($r = 0.47$ and $r = 0.44$, respectively, $p < 0.0001$). The Discrete Sliders controller is better suited where fine-grained accuracy is required over speed, while the Arcball and Two-Axis Valuator are better suited where speed is valued over very accurate model orienting.

2 Related Work

To our knowledge, there have been seven experiments from six studies that performed a quantitative evaluation of mouse-based 3D rotation controllers. The objectives and methodologies of these studies are summarised in Table 1. All studies used a within-subjects design to evaluate the performance and usability of various controllers.

These studies used two types of tasks: orientation matching and inspection [Henriksen et al. 2004]. In orientation matching, participants manipulate a model to match the orientation of a reference [Chen et al. 1988], and in inspection, participants rotate an object while searching for specific properties. The former is applicable when composing 3D animated scenes, while the latter would

---

*e-mail:steveryb@gmail.com
†e-mail:bderenzi@cs.uct.ac.za
‡e-mail:jgain@cs.uct.ac.za

Graphics Interface Conference 2016
1-3 June, Victoria, British Columbia, Canada

Table 1: *Summary of the methods used by empirical quantitative evaluations of mouse-based 3D rotation controllers.*

	Experiment	Controllers Tested	No. of Subjects	No. of Tasks	Task Type(s)	Emphasised Measure	Usability Measures
#1	Chen et al. [1988] Experiment 1	Individual Sliders Overlapping Sliders Virtual Sphere XY+Z Controller	12	27	Orientation matching	Accuracy	N/A
#2	Chen et al. [1988] Experiment 2	Evans Controller Virtual Sphere	6	27	Orientation matching	Accuracy	N/A
#3	Jacob and Oliver [1995]	Evans Controller Overlapping Sliders Virtual Sphere XY+Z Controller	137	18	Orientation matching & Inspection	Accuracy	Ease of use Perceived accuracy, speed
#4	Hinckley et al. [1997]	Arcball Virtual Sphere	24	15	Orientation matching	Accuracy	Ranking controllers
#5	Partala [1999]	Keyboard Controls Virtual Rectangle Virtual Sphere	12	24	Orientation matching	Speed	Ranking controllers
#6	Bade et al. [2005]	Arcball Bell's VT Two-Axis Valuator Fixed Trackball	42	25	Inspection	Speed	User comfort Predictability of controller behaviour
#7	Zhao et al. [2011]	Arcball Bell's VT Two-Axis Valuator	12	40	Orientation matching	Accuracy	Ease of Use Perceived performance, predictability, accuracy Overall usability

typically apply in evaluating a 3D brain scan result [Bade et al. 2005]. These studies controlled task difficulty by varying textural and geometric model complexity, as well as the number of axes required during rotation.

Five of these experiments tested only orientation matching (#1, #2, #4, #5, #7), one tested only inspection (#6), and one tested both (#3). This is unfortunate, as Jacob and Oliver found that while all controllers had comparable accuracy for inspection, one was far less precise for orientation matching. This reinforces the hypothesis that certain controllers might be better suited to certain classes of tasks.

Many of the studies had too few participants to measure statistically significant results. Table 1 shows four experiments were conducted with twelve participants or fewer (#1, #2, #5, #7) and a further one had fewer than twenty five (#4). Furthermore, Henriksen et al. [2004] hypothesise that the task difficulty was insufficient to generate large differences. for a large effect size. For instance, the orientation matching task that all others are based on [Chen et al. 1988] is simple: given a reference orientation of a cartoon-like house, rotate an identical house to match its orientation.

Another possible explanation is confounding effects in the subject pools used by these studies. Every study employed university students, but these pools differed widely. All have a different selection of backgrounds, from postgraduate Engineering students to undergraduate Psychology majors; gender ratios, which can account for difference in performance [Voyer et al. 1995] [Parsons et al. 2004]; and experience with 3D rotation interfaces, which has been found to predict performance on mental rotation tests [Terlecki and Newcombe 2005]. Only two studies (#6, #7) detailed their analysis of confounding factors.

In these studies, task performance was measured by recording either speed or accuracy. Participants were biased towards a particular measure either by instructions (#6) or task feedback (#1, #2). This bias makes it difficult to conclude how effective controllers are. Controllers which are accurate but very slow, or quick but inaccurate are potentially less usable than those not on the extremes.

All experiments performed user preference analysis, undertaken with varying degrees of sophistication. Chen et al. [1988] informally asked subjects how they felt, Hinckley et al. [1997] and Partala [1999] had subjects rank the controllers according to their experience. The others captured specific usability metrics, focusing on ease of use, predictability of the controller's behaviour, and perceived performance attributes.

3 Technical Approach

A survey of popular 3D modeling suites, including Blender [1], Maya 2016 [2], and 3DS Max 2016 [2], reveals three main classes of controller: virtual trackballs, sliders and two-axis valuators. We chose the controller in each class that performed best in previous quantitative evaluations, namely the Two-Axis Valuator with Z rotation and Arcball. We also included a previously un-tested controller: Discrete Sliders. These controllers, and their methods of rotation, are illustrated in Figure 1.

As Henriksen et al. [2004] note, some studies of rotation controllers omit important implementation details. This makes it difficult to replicate the controllers, replicate the results or compare results across studies. In light of this, we provide full specifications of the controllers here.

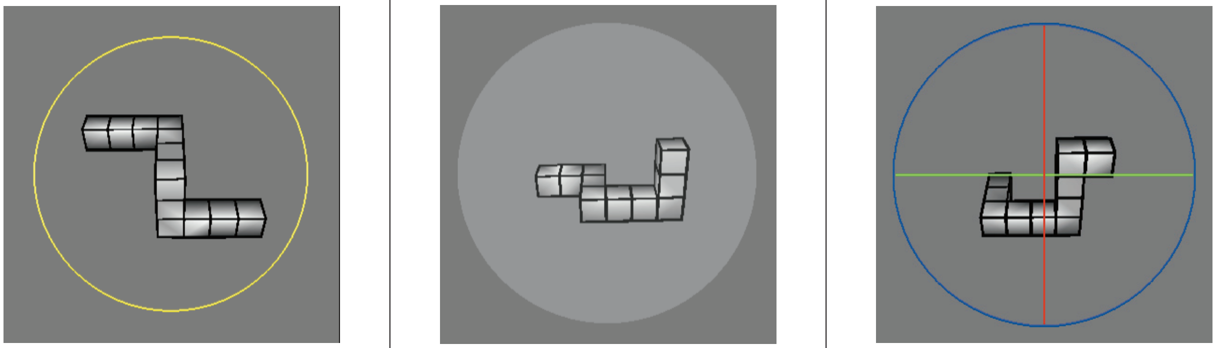
Our rotation controllers are operated by holding down the left mouse button and tracking mouse movement. The two most recently sampled mouse positions are projected onto the plane $z = 0$ as $p_c = (x_c, y_c, 0)$ and $p_l = (x_l, y_l, 0)$, respectively. A rotation controller is then defined as a function: $f : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow$

---

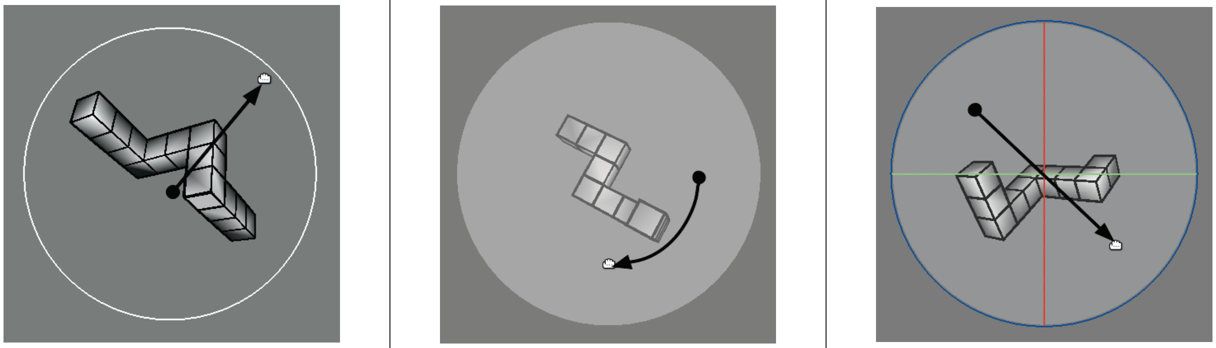
[1] Blender Foundation, www.blender.org

[2] Autodesk, www.autodesk.com

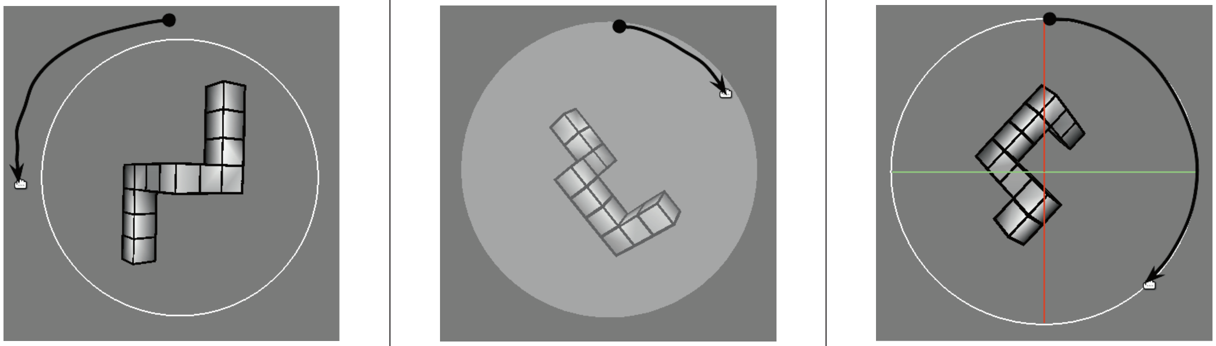
Two-Axis Valuator | **Arcball** | **Discrete Sliders**



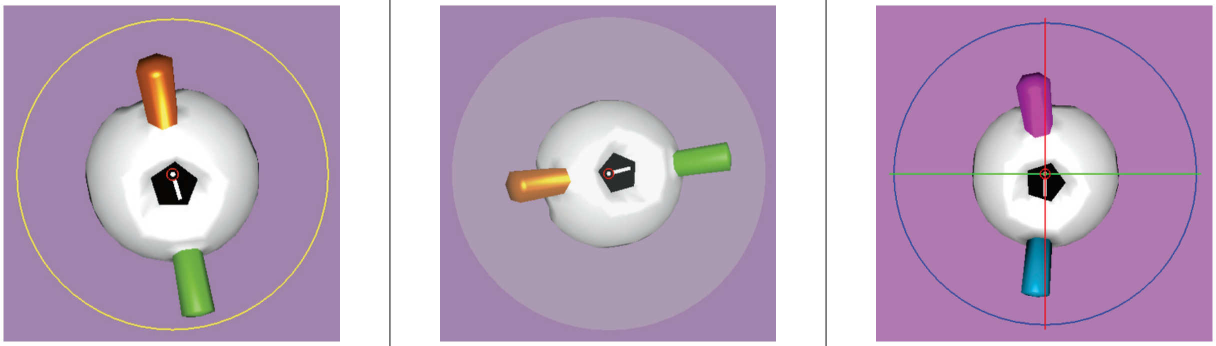
(a) *Model 9a* **(b)** *Model 16a* **(c)** *Model 23a*



(d) *rotate on x, y axes* **(e)** *rotate on the x, y axes* **(f)** *Using Arcball away from sliders*



(g) *rotate on z axis* **(h)** *rotate on the z axis* **(i)** *Using a slider to rotate on the z axis*



(j) *Appearance for Inspection Tasks* **(k)** *Appearance for Inspection Tasks* **(l)** *Appearance for Inspection Tasks*

Figure 1: *Appearance & interaction design for rotation controllers used in this study*

$\mathbb{R}^4, f(p_c, p_l) = q_r$, which maps p_c and p_l to a quaternion q_r representing the rotation applied to the current model. For consistency of use, each controller has approximately the same range of motion: one rotation action can lead to at most a π rotation on a single axis for x, y and a 2π rotation for z.

The cursor changes to provide feedback: an open hand in areas where rotation is possible; an extended index finger over a slider; and a closed hand during active rotation. Unfortunately, due to a CSS problem on the test machines, the closed hand was replaced with a black text cursor, which made it difficult to distinguish from the dark background. While this was present for all controllers, participants noted it mostly with regard to the Two-Axis Valuator.

3.1 Two-Axis Valuator

Blender implements a version of this controller that rotates only on the x and y axes. We instead chose the XY+Z version of Chen et al. [1988], as it allows rotations about the z axis and performs well for inspection tasks [Bade et al. 2005].

The Two-Axis Valuator has two distinct behaviors: if the mouse is moved inside the circular guide, then the model rotates about the x and y axes according to x, y changes in mouse position; while outside the circular guide the model is rotated clockwise or anti-clockwise about the z-axis depending on the arc traced by the mouse.

Let us have a circular guide of radius r centered at the origin and facing the camera. Then, we can calculate the angle of rotation for the x, y and z axes, θ_x, θ_y and θ_z, as:

$$\begin{aligned}
\theta_x &= tav_x(p_c, p_l) := \frac{\pi(x_l - x_c)}{2r}, \\
\theta_y &= tav_y(p_c, p_l) := \frac{-\pi(y_l - y_c)}{2r}, \\
\theta_z &= tav_z(p_c, p_l) := \operatorname{sgn}((p_c \times p_l) \cdot (0,0,1)) \cos^{-1}\left(\frac{p_c \cdot p_l}{|p_c||p_l|}\right).
\end{aligned}$$

We convert these axial rotations to corresponding quaternions q_x, q_y, q_z and define the Two-Axis Valuator rotation function as:

$$f_{tav}(p_c, p_l) = q_x q_y \text{ if } |p_c| <= r, \quad q_z \text{ otherwise}$$

In our implementation, the circular guide is yellow to differentiate it from the Discrete Sliders and it turns white while rotating.

3.2 Arcball

We chose the Arcball controller due to its prevalence in 3D modeling suites as well as its performance for inspection tasks [Bade et al. 2005]. In Shoemake's original Arcball [1992], the model is positioned at the center of a sphere, onto which the mouse position is projected. Rotation then mimics what would happen if you were to nudge the sphere in the corresponding direction. Mouse movements outside the sphere are translated to z rotations. We modify this slightly by ignoring all movement outside the central sphere, only allowing z-rotations in a thin band on the edge of the sphere. This is more in line with modern 3D packages and better differentiated from the two-axis valuator.

As Shoemake [1992] outlines, we first map p_c and p_l onto the surface of a sphere. We then use these locations to create rotation quaternions. Let r_v and r_p be the radius of the visible and projecting spheres, respectively, with $r_p < r_v$. We then define the function that takes a point on the $z = 0$ plane to the projecting sphere as:

$$\text{project}(x, y, 0) = \begin{cases} \frac{\sqrt{x^2+y^2}(x,y,0)}{r_p^2} & \text{if } \frac{x^2+y^2}{r_p^2} > 1 \\ (\frac{x}{r_p}, \frac{y}{r_p}, \sqrt{1 - \frac{x^2+y^2}{r_p^2}}) & \text{otherwise} \end{cases}$$

We then take the corresponding points on the sphere, $s_c = \text{project}(p_c), s_l = \text{project}(p_l)$ and create a quaternion from the angle between them using their cross product. However, unlike the original Arcball, this is only for points within the visible sphere.

$$f_{arcball}(p_c, p_l) = \begin{cases} (\frac{s_c \times s_l}{|s_c \times s_l|}), \cos^{-1}(\widehat{s_c} \cdot \widehat{s_l}) & \text{if } |p_c| < r_v \\ (1,0,0,0) & \text{otherwise} \end{cases}$$

In our implementation, we set $r_v : r_p$ as $8 : 9$. During rotation, the visible Arcball becomes more opaque as a form of visual feedback.

3.3 Discrete Sliders

Discrete sliders are widespread in Autodesk products: Maya 2016 and 3DS Max 2016, which employ three circular sliders aligned with the three orthographic planes surrounding a virtual trackball. Each slider allows for rotation on a different axis. A user can rotate a model using either the sliders or the virtual trackball (implemented as an Arcball in our case).

Unlike other controllers, the Discrete Sliders have different modes depending on where rotation begins. If the mouse starts over a slider, then all subsequent mouse dragging until button up will be interpreted as rotations on that slider's axis, otherwise Arcball applies. Let the initial mouse position be p_s. Then, let the regions where the x, y and z sliders accept input be R_x, R_y, R_z, respectively. We assume that $R_i \cap R_j = \emptyset$ for $i \neq j$ and $i, j \in \{x, y, z\}$. Finally, let the visible radius of our arcball controller be r_v and the radius of the sphere we are projecting onto be r_p, with $r_p < r_v$. Then,

$$f_r(p_c, p_l) = \begin{cases} tav_x(p_c, p_l) & \text{if } p_s \in R_x \\ tav_y(p_c, p_l) & \text{if } p_s \in R_y \\ tav_z(p_c, p_l) & \text{if } p_s \in R_z \\ f_{arcball}(p_c, p_l) & \text{if } p_s \notin R_x \cup R_y \cup R_z \end{cases}$$

In our case, R_x, R_y, R_z are thin bands bracketing each individual slider. Note that the model of rotation depends on p_s, not p_c or p_l. This means that once a user has selected a slider or Arcball, any motion is interpreted using that rotation controller until mouse-button release, even if it is no longer in the corresponding region.

4 Method

4.1 Experimental Design

We performed a randomised single factor repeated measures experiment with 46 students–42 undergraduates and 4 postgraduates. Participants were recruited through convenience sampling by advertising with posters and on social media. A small monetary incentive was offered to encourage participation. Our target population was computer literate users capable of using 3D modeling software. To ensure computer literacy, and capture potentially confounding factors, participants were screened using an adapted version of the Survey of Spatial Representation and Activities [Terlecki and Newcombe 2005]. This captured demographics and potential covariates such as gender, experience with 3D modeling software, and frequency playing video games. No participants were excluded, as all listed themselves as at least moderately skilled with computers.

Table 2: *Specification of the orientation and inspection tasks. The* (x, y, z, w) *values form a quaternion* q *which if applied to the model rotates it to the orientation required to solve the task*

Name	Type	x	y	z	w
Simple X	Orientation	0.26	0.00	0.00	0.97
Simple Z	Orientation	0.00	0.00	0.71	0.71
Simple X+Y	Orientation	0.33	0.46	0.19	0.80
Simple Y+Z	Orientation	0.22	0.22	0.67	0.67
Complex 1	Orientation	0.05	0.79	0.12	−0.61
Complex 2	Orientation	−0.91	0.36	0.16	−0.12
Complex 3	Orientation	0.06	0.74	−0.10	−0.66
Bottom	Inspection	0.62	−0.47	0.36	−0.51
Top	Inspection	−0.53	−0.57	−0.46	−0.43
Back	Inspection	−0.05	0.99	−0.09	0.00
Left	Inspection	−0.00	0.77	0.00	0.64
Right	Inspection	0.04	0.78	0.02	−0.63

Experiments were performed in a closed lab with up to five participants at a time. Each participant used a desktop computer running Windows 8.1. In the interests of reproducibility [Casiez and Roussel 2011] we report the mouse and screen characteristics, as follows: LG 22EA53 displays were used, with dimensions of 47.5 cm × 26.7 cm and with resolutions of 1920 × 1080. Each participant used a Proline MSU0767 mouse, with mouse sensitivity specified in the operating system at 50%. On screen the Two Axis Valuator circular guide and Discrete Slider z rotation slider were 1 mm wide on screen, with the circular guides, z rotation slider and the Arcball sphere all 177 mm wide on screen. For the Discrete Sliders, there was a buffer of 17 mm around a slider for selection.

Participants were provided a printed manual and received on-screen instructions before each stage of the experiment. One participant was excluded for misreading instructions, leaving 45 for analysis.

Before the evaluations, participants were given 10 minutes to complete a Mental Rotation Test (MRT) [Vandenberg and Kuse 1978], a common test for assessing ability to mentally rotate shapes [Harle and Towns 2010]. They were asked to complete the test as quickly and accurately as possible, but refrain from guessing.

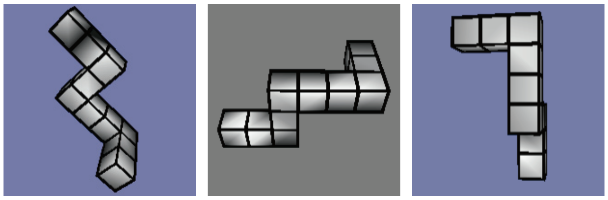
Controller type is our single factor, with three levels: the Two-Axis Valuator, Arcball, and Discrete Sliders. For each controller evaluation, participants were given controller-specific instructions, and then allowed three minutes to train before performing a total of 12 tasks with that controller. Before each evaluation, participants were again instructed to complete each task "quickly and accurately".

To prevent memorisation of the tasks, the order of tasks was randomised within each group and each controller's orientation matching tasks was presented using one of three models taken from the MRT. These models are shown in Figure 1a, Figure 1b, and Figure 1c The order of models and controllers was counterbalanced. The primary measures for tasks were accuracy and time.

After each set of evaluation tasks, participants assessed the usability of the controllers using the System Usability Scale (SUS) [Brooke 1996] giving us a usability score as a third measure.

4.2 Task Design

Each controller was evaluated with 12 tasks: 7 orientation matching, and 5 inspection tasks. The models used for these tasks are shown in Figure 1 and the specification of rotation required to solve each task is listed in Table 2.



(a) *Complex 2* **(b)** *Start* **(c)** *Simple Z*

Figure 2: *Screenshots of a simple and complex task matching task alongside the starting orientation of the model*

Our orientation tasks are similar to Chen et al. [1988] but replace simple coloured houses with more complex models from the MRT, which are designed to make rotation tasks more challenging. We further divided the orientation tasks into two groups: Simple, which require rotation on one or two axes, and Complex, which require rotation about all three. This difference is illustrated in Figure 2.

The inspection tasks are based on those of Bade et al. [2005]: participants need to rotate a complex geometric shape while looking for the side containing only the letter "i". Once located, they position the dot of the "i" inside a blank circle visible on screen.

Participants were first presented with Simple orientation tasks, followed by Complex orientation tasks, and finally inspection tasks.

4.3 Measures

For every task, we recorded accuracy and time measures and used these to calculate the accuracy normalised over time. Per controller, we captured usability scores using the SUS questionnaire. Below we define these measures more explicitly.

Accuracy: in radians, how closely the participant's model is oriented to the reference model. For orientation matching, this is the minimum angle required to rotate the model to match the reference. We represent the reference model's orientation as a unit quaternion q_r and the participant's submitted orientation as the unit quaternion q_p. We then calculate the minimum angle between q_r and q_p by

$$\theta = \cos^{-1}\left(2\langle q_r, q_p\rangle^2 - 1\right) \tag{1}$$

where $\langle q_r, q_p\rangle$ is the inner product [Huynh 2009].

For inspection tasks, accuracy is the minimum angle required to orient the participant's model so that the center of the side containing the "i" is in the center of the participant's camera. To measure this, We calculate the difference in orientation, ignoring any roll around the dot of the "i". That is, the "i" does not have to be upright, the participant merely has to position the circle on the dot of the "i".

This was calculated by transforming the quaternion to the corresponding Tait-Bryan rotation angles, discarding the z rotation component and transforming these angles to the corresponding quaternion. We then compare the users quaternion with the reference quarternion using Equation 1.

Speed: in seconds, the length of time between starting a task and submitting it.

Usability: as ordinal data captured through the post-test usability questionnaire as well as an overall SUS score.

Score: this was a synthetic measure, used to combine both speed and accuracy into a single value. The score is calculated per completed task by computing $\frac{\pi - \text{accuracy}}{\text{time}}$, meaning the larger the score is the better a participant performed. This measure was created to balance the relationship between accuracy and time (given more time, we expect participants to be more accurate). While this does not replace the speed or accuracy measures, it allows us to compare a notion of relative efficiency given by radians of accuracy per second achievable with a given controller.

5 Data Analysis and Results

We analyzed data from 45 participants. In terms of demographics, we had 13 participants identify as female, 32 as male with 19 participants between the ages of 18 and 21, 24 between the ages of 22 and 25 and 2 over the age of 25. In terms of prior skill, 9 participants rated themselves as moderately skilled with 3D modelling software, 19 rated themselves as being not very skilled, 16 rated themselves as having no skill, and one participant abstained from answering the question.

Each participant performed 12 tasks (4 Simple orientation, 3 Complex orientation and 5 inspection) and a usability questionnaire, for each of the three controllers. This gave us 1620 observations and 135 usability questionnaires.

Results for a task submission were discarded if the accuracy was worse than 0.5 radians. This threshold was chosen as we regard anything above this as too inaccurate for a task to be considered successfull. As this is a repeated measures experiment, we removed the corresponding task submissions for that participant across all 3 controllers. This resulted in 138 of the 1620 observations being discarded. Additionally, partially-completed SUS questionnaires were also removed, along with corresponding questionnaires for the other controllers, resulting in 3 surveys being discarded.

This gave us 483 Simple orientation task observations, 357 Complex orientation task observations, 642 inspection task observations, and 132 usability surveys for analysis.

Before performing our analyses, we applied the D'Agostino and Pearson [1973] omnibus test of normality to our data and found that, with statistical significance, all our outcome measures obeyed a non-parametric distribution. We therefore used non-parametric statistical methods for our analysis.

For each performance measure (accuracy, speed, and score), we applied the Friedman test to each set of observations (simple orientation, complex orientation, and inspection) to determine if there was a significant difference in performance between the three controllers for that task. When a significant difference was reported by the Friedman test, a post-hoc test using Wilcoxon Signed Rank tests with Bonferroni correction was performed to determine which groups differed.

5.1 Accuracy

The median accuracy in radians, per task and controller (see Figure 3a) was as follows:

	Simple	Complex	Inspect
Arcball	0.087	0.116	0.021
Discrete	0.060	0.107	0.019
Two-Axis	0.096	0.111	0.019

A Friedman test revealed a significant effect of controller on accuracy for the simple orientation task ($\chi^2(2) = 15.5, p < 0.001$). A post-hoc test using Wilcoxon Signed Rank tests with Bonferroni correction showed significant differences between the Arcball and Discrete controllers ($p < 0.001, r = 0.30$) and also between the Two Axis Valuator and Discrete controllers ($p < 0.01, r = 0.54$).

No significant difference was found in the accuracy achieved between controllers for either the complex orientation or inspection tasks.

5.2 Speed

The median speed in seconds, per task and controller (see Figure 3b) was as follows:

	Simple	Complex	Inspect
Arcball	18.8	19.5	7.41
Discrete	20.6	24.3	9.16
Two-Axis	18.2	18.6	7.30

While no significant difference was found in speed for completing the simple orientations, there was a significant effect of controller on speed for complex orientations ($\chi^2(2) = 10.7, p < 0.01$). The post-hoc test showed significant differences between Discrete and Arcball controllers ($p < 0.01, r = 0.33$) and between Discrete and Two-axis controllers ($p < 0.01, r = 0.28$).

There was also a significant effect on speed for the inspection task ($\chi^2(2) = 37.36, p < 0.001$). Again, significant differences between the Discrete and Arcball controllers ($p < 0.001, r = 0.44$) and between the Discrete and Two-axis controllers ($p < 0.001, r = 0.47$) were found with the post-hoc test.

5.3 Score

The median score normalising accuracy by time, per task and controller (see Figure 3c) was as follows:

	Simple	Complex	Inspect
Arcball	0.161	0.151	0.420
Discrete	0.147	0.126	0.341
Two-Axis	0.166	0.159	0.425

Again, no significant difference was found for the Simple orientation tasks. However, the Friedman test revealed a significant effect of controller on the score for the complex orientation task ($\chi^2(2) = 10.6, p < 0.01$). A significant difference between the Discrete and Two-axis controllers ($p < 0.05, r = 0.25$) was detected.

Furthermore, there was also a significant effect on score for the inspection task ($\chi^2(2) = 36.3, p < 0.01$), with significant differences between the Discrete and Arcball controllers ($p < 0.01, r = 0.48$) and between the Discrete and Two-axis controllers ($p < 0.01, r = 0.50$).

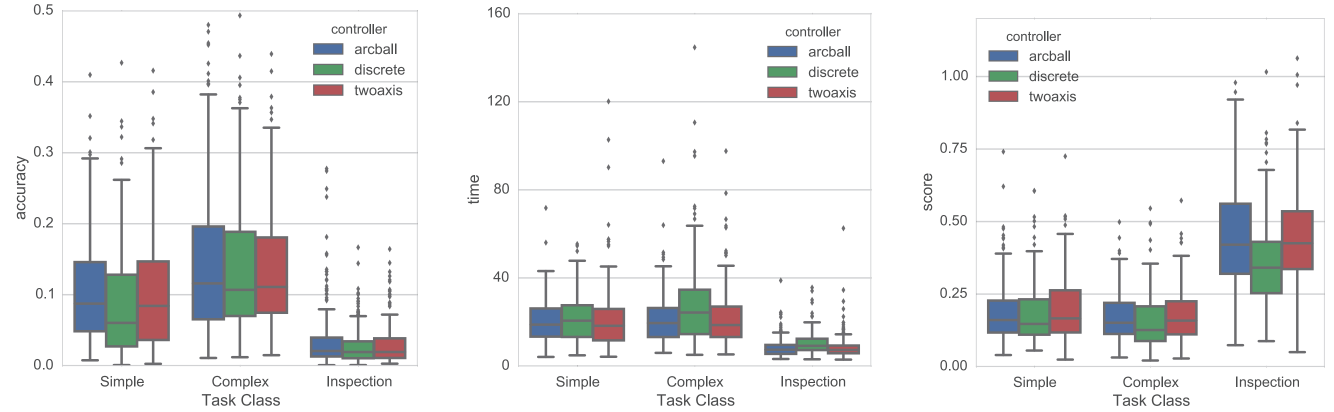
5.4 Usability score

A Friedman-based quantitative analysis of the SUS questionnaire responses found no statistically significant differences between controllers for individual questions or total score.

Qualitatively, participants reported in the SUS questionnaire that, while the Arcball was fairly intuitive, they found it difficult to achieve fine-grained accuracy.

The Discrete Sliders were thought to achieve better accuracy. However, this controller was not considered helpful for inspection tasks and the sliders were reported as difficult mouse targets.

For the Two-Axis Valuator, participants prefered the larger z-rotation zone, but found the discontinuity between the different forms of rotation jarring when they accidentally dragged the mouse over the circle barrier.



(a) *Box and Whisker plot for the accuracy of submissions in each group*

(b) *Box and Whisker plot for the time of submissions in each group*

(c) *Box and Whisker plot for the score of submissions in each group*

Figure 3: *Box and Whisker plots showing the distribution of the measures for task submission*

5.5 Confounding factors

We collected data on twenty-six possible confounding factors to assess for potential bias in our study. For each of the confounding factors, we created a stratum for each level of the factor, leading to a total of 97 individual strata.

We performed a within-subject analysis of the controllers for each individual stratum. For example, for the "Gender" factor, we performed a Friedman test across controllers for all the individuals of each sex separately. Again, for stratum with a significant result, a post-hoc test using Wilcoxon Signed Rank tests with Bonferroni correction was performed to determine which groups differed.

Four factors were found to have a significant impact on speed between the Arcball and Two-Axis Valuator and two of those factors also had a significant effect on the score measure: order of controllers during the study, namely "Two-Axis Valuator, Discrete Sliders, Arcball" ($r = 0.47$ for time, $r = 0.33$ for score); participation in Intramural sports teams ($r = 0.33$ for speed, $r = 0.37$ for score); playing board games once or twice a year ($r = 0.42$ for speed); having no prior skill with 3D modeling software ($r = 0.32$ for speed). No differences were found based on the age, gender or prior 3D modelling software experience of participants.

6 Discussion

The Discrete Sliders was the most accurate controller for Simple orientation tasks, performing significantly better than the Arcball and Two-Axis Valuator, but we did not see this difference for Complex orientation or Inspection tasks. One possible explanation is that the sliders, which are constrained to a single axis, favour simple axial rotations over free-form multi-axis rotations.

When compared to Discrete Sliders, the Arcball and Two-Axis Valuator controllers are moderately faster for both Complex Orientation and Inspection tasks. The speed increase may be attributed to the free-form nature of these controller.

Taken together, these two findings motivate a new controller that makes it easier to use the Arcball (or Two-Axis Valuator) for larger, multi-axis orientation tasks and the precision of the Discrete Sliders for the final stages. Our implementation of the Discrete Sliders provided both in one, but perhaps a more explicit, user-toggled separation between the interfaces would provide the benefits of both.

The longer task time required for Discrete Sliders might be due to the relatively small clickable area covered by the sliders. In contrast, mouse targets for the Arcball and Two-Axis Valuator span most of the screen. By Fitts's Law, the difference in area means that sliders will generally be slower than using the Arcball or Two-Axis Valuator. Even if a participant were to exclusively use the Arcball functionality within the Discrete Sliders controller, this would need to be selected between the various sliders, dividing the click target into four smaller regions.

There is only one task where there are significant differences between Arcball and Two-Axis Valuator: the Bottom Inspection task (see Table 2). This contradicts results found by Bade et al. [2005] where the Two-Axis Valuator is faster than Arcball across all inspection tasks. The difference here could be explained by our use of a modified Arcball rather than Shoemake's Arcball [Shoemake 1992] as used in Bade et al's studies. Another possible explanation is that while the method of rotation is fairly dissimilar mechanically, the interface of the Two-Axis Valuator controller and the Arcball controller were more similar to each other than that of the Discrete Slider controller.

7 Conclusions

We performed a quantitative evaluation of Arcball, Two-Axis Valuator, and Discrete Sliders to determine whether one of these controllers was more effective or usable than another for a particular class of tasks.

The Discrete Sliders controller more accurate for simple (one- or two-axis) tasks than both the Arcball controller (with a small effect size) and the Two-Axis Valuator (with a medium effect size). However, it was significantly slower than the other controllers for complex (three-axis) tasks. More evidence is required to effectively determine whether the controller is slower for simpler fine-grained orienting tasks.

Overall, our results suggest that the Discrete Sliders are most useful in circumstances where accuracy is paramount, such as in 3D modeling suites. When absolute accuracy is less of a concern and speed is more important, both the Two-Axis Valuator and Arcball controllers appear to be more effective (with no detectable difference between them).

Future experiments could focus on further differentiating the performance of these controllers across time and accuracy. One approach might be to fix the acceptable accuracy for a task (e.g., by providing real-time feedback and preventing the participant from moving on to the next task before attaining a required accuracy threshold) and measure the time required to complete the task. This accuracy threshold, along with the complexity of the models, could be varied to control the difficulty of the task. Conversely, one could provide a number of models to be oriented in a fixed timespan and then measure the accuracy achieved by participants.

It is clear that interactively orienting 3D models is an important component in a wide range of applications. This paper contributes to a growing body of evidence that no single rotation controller works best across all tasks for both speed and accuracy. Further exploration is required to fully understand the situations under which different controllers are most effective. This information can then be fed back into the design of 3D modeling and visualisation software.

Acknowledgements

This work was funded by NRF/THRIP grant TP14080788335 and the Center for High Performance Computing, South Africa.

References

BADE, R., RITTER, F., AND PREIM, B. 2005. Usability comparison of mouse-based interaction techniques for predictable 3d rotation. In *Proceedings of the 5th International Conference on Smart Graphics*, Springer-Verlag, Berlin, Heidelberg, SG'05, 138–150.

BOWMAN, D. A., KRUIJFF, E., LAVIOLA, J. J., AND POUPYREV, I. 2004. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

BROOKE, J. 1996. Sus: a 'quick and dirty' usability scale. In *Usability evaluation in industry*, P. Jordan, B. Thomas, I. McClelland, and B. Weerdmeester, Eds. Taylor & Francis Ltd, London, UK, 189–194.

CASIEZ, G., AND ROUSSEL, N. 2011. No more bricolage!: Methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ACM, New York, NY, USA, UIST '11, 603–614.

CHEN, M., MOUNTFORD, S. J., AND SELLEN, A. 1988. A study in interactive 3-d rotation using 2-d control devices. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '88, 121–129.

D'AGOSTINO, R., AND PEARSON, E. S. 1973. Tests for departure from normality. empirical results for the distributions of b2 and b1. *Biometrika 60*, 3, 613–622.

HARLE, M., AND TOWNS, M. 2010. A review of spatial ability literature, its connection to chemistry, and implications for instruction. *Journal of Chemical Education 88*, 3, 351–360.

HENRIKSEN, K., SPORRING, J., AND HORNBÆK, K. 2004. Virtual trackballs revisited. *Visualization and Computer Graphics, IEEE Transactions on 10*, 2, 206–216.

HINCKLEY, K., TULLIO, J., PAUSCH, R., PROFFITT, D., AND KASSELL, N. 1997. Usability analysis of 3d rotation techniques. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, ACM, New York, NY, USA, UIST '97, 1–10.

HUYNH, D. Q. 2009. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision 35*, 2, 155–164.

JACOB, I., AND OLIVER, J. 1995. Evaluation of techniques for specifying 3d rotations with a 2d input device. In *Proceedings of the HCI'95 Conference on People and Computers X*, Cambridge University Press, New York, NY, USA, HCI '95, 63–76.

NIELSEN, J. 1994. Heuristic evaluation. *Usability inspection methods 17*, 1, 25–62.

PARSONS, T. D., LARSON, P., KRATZ, K., THIEBAUX, M., BLUESTEIN, B., BUCKWALTER, J. G., AND RIZZO, A. A. 2004. Sex differences in mental rotation and spatial rotation in a virtual environment. *Neuropsychologia 42*, 4, 555–562.

PARTALA, T. 1999. Controlling a single 3d object: viewpoint metaphors, speed and subjective satisfaction. In *Proc. of INTERACT*, IOS Press, Edinburgh, UK, vol. 99, 536–543.

PETERS, M., LAENG, B., LATHAM, K., JACKSON, M., ZAIYOUNA, R., AND RICHARDSON, C. 1995. A redrawn vandenberg and kuse mental rotations test-different versions and factors that affect performance. *Brain and cognition 28*, 1, 39–58.

SHEPARD, R. N., AND METZLER, J. 1971. Mental rotation of three-dimensional objects. *Science 171*, 3972, 701–703.

SHOEMAKE, K. 1992. Arcball: A user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of the Conference on Graphics Interface '92*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 151–156.

TERLECKI, M. S., AND NEWCOMBE, N. S. 2005. How important is the digital divide? the relation of computer and videogame usage to gender differences in mental rotation ability. *Sex Roles 53*, 5, 433–441.

VANDENBERG, S. G., AND KUSE, A. R. 1978. Mental rotations, a group test of three-dimensional spatial visualization. *Perceptual and motor skills 47*, 2, 599–604.

VOYER, D., VOYER, S., AND BRYDEN, M. P. 1995. Magnitude of sex differences in spatial abilities: a meta-analysis and consideration of critical variables. *Psychological bulletin 117*, 2, 250.

WANG, Y., MACKENZIE, C. L., SUMMERS, V. A., AND BOOTH, K. S. 1998. The structure of object transportation and orientation in human-computer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, CHI '98, 312–319.

WARE, C., AND ROSE, J. 1999. Rotating virtual objects with real handles. *ACM Trans. Comput.-Hum. Interact. 6*, 2 (June), 162–180.

WENJUN, H. 2008. The three-dimensional user interface. In *Advances in Human Computer Interaction*, S. Pinder, Ed. INTECH Open Access Publisher, Rijeka, Croatia.

ZHAO, Y. J., SHURALYOV, D., AND STUERZLINGER, W. 2011. Comparison of multiple 3d rotation methods. In *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2011 IEEE International Conference on*, IEEE, Ottawa, ON, IEEE, 1–5.