UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

COMPUTER SCIENCE HONOURS
FINAL PAPER
2015

Title: A usability and performance comparison for mouse-based rotation controllers

Author: Steven Rybicki

Project Abbreviation: 3DUI

Supervisor: Prof. James Gain, Dr. Brian DeRenzi, Dr. Pedro Wolf

CATEGORY	MIN	MAX	CHOSEN
Requirement Analysis and Design	0	20	0
Theoretical Analysis	0	25	0
Experiment Design and Execution	0	20	20
System Development and Implementation	0	15	5
Results, Findings and Conclusion	10	20	20
Aim Formulation and Background Work	10	15	15
Quality of Paper Writing and Presentation	10		10
Adherence to Project Proposal and Quality of Deliverables	10		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	0	10	0
Total marks		**80**	80

A usability and performance comparison for mouse-based rotation controllers

Steven Rybicki
Computer Science Department
University of Cape Town
steveryb@gmail.com

ABSTRACT

Rotation controllers are used to orient models in important applications in 3D computer graphics and visualisation. Unfortunately, previous studies do not provide clear guidance on which rotation controller is most effective for particular applications as they struggle to obtain significant results, bias participants towards individual performance measures and do not include a prevalent rotation controller: the Discrete Sliders.

In this paper, we present the results of a broad quantitative user experiment ($n = 46$) to compare the three most prevalent controllers (Arcball, Two Axis Valuator, and Discrete Sliders). We aimed to determine whether there was a difference in the suitability of these controllers for the two main classes of orientation tasks: orientation matching and inspection. Participants were instructed to complete tasks as quickly and accurately as they could to avoid bias towards a particular measure.

We found no statistically significant differences between Arcball and Two Axis Valuator. However, we found the Discrete Sliders to be significantly more accurate for simple matching tasks (a medium to large effect) but slower for complex matching, and inspection tasks (a small to medium effect of approximately two seconds). This indicates that the Discrete Sliders are most appropriate in situations where fine-grained accuracy is valued over speed, while for other tasks, such as inspection, either Arcball or Two Axis Valuator are better suited.

CCS Concepts

•**Human-centered computing → Graphics input devices;** *Usability testing; Laboratory experiments;* Heuristic evaluations;

Keywords

3D User Interfaces, 3D Rotation, 2D mouse, Arcball, Discrete sliders, Two Axis Valuator, Usability study

This thesis was completed as part of a Bachelor of Science Honours degree in Computer Science at the University of Cape Town in 2015. For more information about this project, including a proposal and literature review, please see http://3dui.github.io/. The software used in this evaluation can be viewed at http://3dui.github.io/testing-rotation-controllers/, with its source code accessible at https://github.com/3DUI/testing-rotation-controllers. Raw results and software used in the analysis can be found at https://github.com/3DUI/rotation-data-analysis.

1. INTRODUCTION

Three-Dimensional (3D) scene manipulation is integral to many significant tasks, including creating virtual scenes for game and film; examining 3D models for applications such as medicine, architecture, computer-aided design; and the exploration of virtual worlds [6]. Such applications enable a user to select, position, and rotate objects into the desired orientation [27]. However, these operations are not all equally fast: users can position objects in $2 - 3$ seconds [26] but take upwards of 10 seconds to orient them [14]. Interestingly, an equivalent task with a physical cube can be completed in under a second [25]. This suggests that there is room to improve rotation controllers, which would in turn improve 3D scene manipulation as a whole.

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 seem to indicate that controllers that perform well for orientating models, such as a Virtual Trackballs [9], do not perform as well for "inspecting" a model [3].

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 modelling suites. Our aim was to determine whether one of the controllers tested was more effective or usable than another for orientation or inspection tasks. To this end, we had two hypotheses:

- Discrete Sliders controller would perform more accurately and quickly in fine-grained orienting tasks
- Two Axis Valuator controller would be faster for inspection tasks.

From our experiment, we conclude that Discrete Sliders was significantly more accurate for simple fine-grained accuracy tasks ($r = 0.30$, $p < 0.001$) than Arcball and more accurate for rotations about the x axis than Two Axis Valuator ($r = 0.54$, $p < 0.01$). Discrete Sliders took longer (approximately 2 seconds difference in the median time) to obtain this accuracy. No significant difference in accuracy was found between the controllers for complex orientation or inspection tasks. However, 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, $p < 0.0001$). This shows that 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 the controllers.

These studies used two types of tasks: orientation matching and inspection [13]. In orientation matching, participants manipulate a model to match a reference model [9]. In inspection, participants rotate an object while searching for specific properties. For example, Bade et al. [3] had participants rotate an object looking for a target. Once the target was found, they had to rotate it to the centre of the screen and "shoot" it to confirm they had identified it. The former is used, for example, in composing 3D animated scenes, while the latter applies to evaluating a 3D brain scan result [3] as it requires identifying and framing details on a model. 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 [16] 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.

All studies had difficulty obtaining statistically significant results. There are a number of possible explanations including small sample sizes, leading to low statistical power. As seen in Table 1, 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. [13] hypothesise that the task difficulty was insufficient to generate large differences. For instance, the orientation matching task of Chen et al. [9] (on which all other orientation matching tasks are based) is simple: given a reference orientation of a cartoon-like house, rotate an identical house using a controller 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 (there is often significant variation in ability across gender [24, 19]) and experience with 3D rotation interfaces (found to predict participants' performance on mental rotation tests [22]). Only three experiments (#4, #6, #7) detailed their analysis of confounding factors. Even these studies only looked at two confounding factors: gender and previous experience with 3D modelling. This makes it unclear how the others might have been affected by confounding effects.

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

All the experiments did some degree of user preference analysis, undertaken with varying degrees of sophistication. Chen et al. [9] informally asked subjects how they felt, Hinckley et al. [14] and Partala [20] had subjects rank the controllers according to their overall experience. The others captured specific usability metrics, focusing on ease of use, predictability of the controller's behaviour, and perceived performance attributes.

It would be useful to compare the suitability of all the controllers tested for different classes of tasks. However, it is difficult to compare controllers across studies. This is due to the differences in their experimental design, limited overlap in the controllers tested, and the small number of statistically significant differences detected. As such, when comparing controllers, we can only consider the intra-study differences. To perform these intra-study comparisons, we extracted the statistically significant results from each study and used these to rank each controller. These rankings are presented in Table 2.

From these rankings, we can see that previous experiments have struggled to differentiate controllers on accuracy. Only two experiments had any statistically significant comparisons for accuracy. Even when results were significant, they were fairly limited. Jacob and Oliver [16] found Evan's controller less accurate than others tested for orientation matching tasks. Bade et al [3] found Bell's VT, a variation of the Virtual Sphere, was more accurate than the other controllers.

We can also see some indication that some controllers being suited better for certain classes of tasks. The Overlapping Sliders performed slowly for both orientation matching [9] and inspection [16], but was found to be as accurate as other controllers for orientation matching [20]. This implies that while sliders are slow, they are accurate. While Bell's VT was ranked lower for usability than the Two Axis Valuator for inspection tasks [3], it was ranked more highly for usability for orientation matching tasks [28].

3. TECHNICAL APPROACH

We conducted a survey of popular 3D modelling suites, including Blender [5], Maya 2016 [1], and 3DS Max 2016 [2], revealing three main classes of rotation 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 untested controller: Discrete Sliders. These controllers and their methods of rotation are illustrated in Figure 1.

As Henriksen et al. [13] 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:

$$f : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^4, f(p_c, p_l) = q_r$$

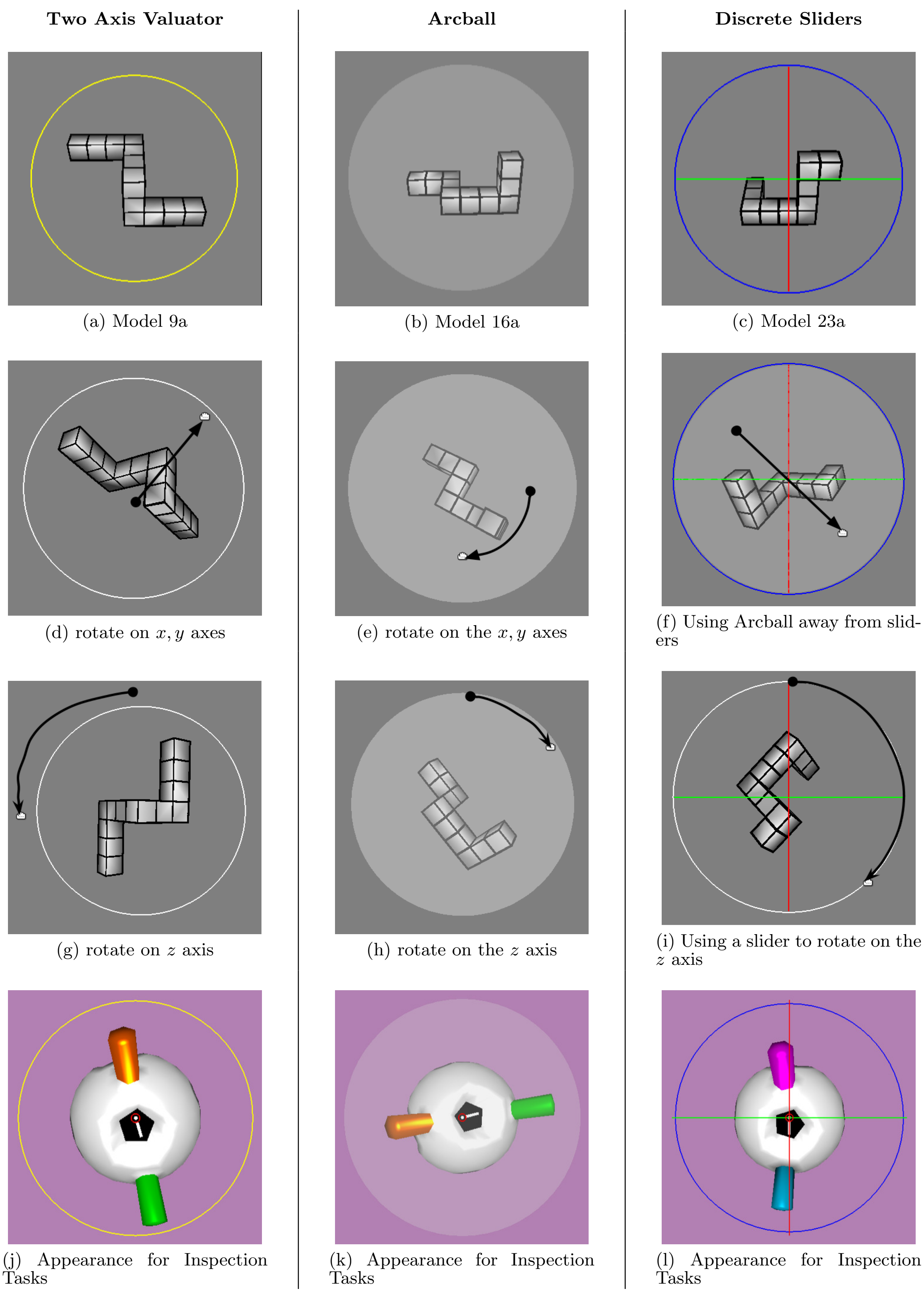
Table 1: Summary of the methods used by empirical quantitative evaluations of mouse-based 3D rotation controllers.

	Experiment	Controllers Tested	No. of Subjects	No. of Tasks	Task Type(s)	Emphasised Measure	Usability Measures
#1	Chen et al. [9] Experiment 1	Individual Sliders Overlapping Sliders Virtual Sphere XY+Z Controller	12	27	Orientation matching	Accuracy	N/A
#2	Chen et al. [9] Experiment 2	Evans Controller Virtual Sphere	6	27	Orientation matching	Accuracy	N/A
#3	Jacob and Oliver [16]	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. [14]	Arcball Virtual Sphere	24	15	Orientation matching	Accuracy	Ranking controllers
#5	Partala [20]	Keyboard Controls Virtual Rectangle Virtual Sphere	12	24	Orientation matching	Speed	Ranking controllers
#6	Bade et al. [3]	Arcball Bell's VT Two Axis Valuator Two Axis Valuator with Fixed Up-vector	42	25	Inspection	Speed	User comfort Predictability of controller behaviour
#7	Zhao et al. [28]	Arcball Bell's VT Two Axis Valuator	12	40	Orientation matching	Accuracy	Ease of Use Perceived performance, predictability, accuracy Overall usability

Table 2: Ranks of controllers per measure in studies which obtained statistically significant results

Rotation Method	Speed	Accuracy	Usability
Arcball	2/3‖	2/2‖	2/3 ‖ 3/3 ¶
Bell's VT	3/3‖	1/2‖	2/3 ‖ 1/3 ¶
Evans Controller	2/2†	2/2‡	2/2*
Individual Sliders	4/4*		
Keyboard	2/2§		3/3§
Overlapping Sliders	3/4* 2/2†	1/2‡	
Two Axis Valuator	1/3‖	2/2‖	1/3 ‖ 2/3 ¶
Two Axis Valuator with Fixed Up-vector	3/3‖	2/2‖	3/3 ‖
Virtual Rectangle	1/2§		1/3§
Virtual Sphere	1/4* 1/2† 1/2§	1/2‡	1/2* 2/3 §
XY+Z Controller	2/4* 2/2†	1/2‡	

Underlined: this position is tied
* Chen et al. [9] Orientation Matching Tasks
† Jacob and Oliver [16] Inspection Tasks
‡ Jacob and Oliver [16] Orientation Matching Tasks
§ Partala [20] Orientation Matching Tasks
‖ Bade et al. [3] Inspection Tasks
¶ Zhao et al. [28] Orientation Matching Tasks



(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

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 mouse cursor icon changes to provide feedback: an open hand in areas where rotation is possible; an extended index finger when over a slider; and a closed hand during active rotation. Unfortunately, due to problems with CSS on our test machines, the closed hand was replaced with a black text cursor, which made it difficult to distinguish against the dark background. While this was present for all three controllers, participants noted it most acutely 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. [9], as it allows rotations about the z axis and performs well for inspection tasks [3].

The Two Axis Valuator has two distinct behaviours: 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:

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

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

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

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 modelling suites as well as its performance for inspection tasks [3]. In the Arcball stipulated by Shoemake [21], the model is positioned at the centre 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. However, instead of Shoemake's Arcball [21], we use a controller more in line with modern 3D packages and better differentiated from the two-axis valuator. In our Arcball, we ignore all movement outside the central sphere and only allow z-rotations in a thin band on the edge of the sphere.

To implement our Arcball, we show the user a larger sphere around the model, called the visible sphere, outside of which we ignore any mouse movements. We then do all calculations on a smaller inner sphere. This means any motion on the edge of the visible sphere is outside the inner sphere, providing z rotations as desired.

As Shoemake [21] 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:

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

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.

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

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

Our Arcball differs from the Two Axis Valuator in its z rotation and in that it does not suffer from hysteresis [21]. Rotating the Two Axis valuator in a circular motion inside the circular guide will leave the model in a different orientation than when it started, which some users find unexpected. For Arcball, closed loops of motion produce closed loops of rotation.

3.3 Discrete Sliders

Discrete sliders are widespread in Autodesk products, such as Maya 2016 and 3DS Max 2016. They employ a controller with 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 has 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 implementation, 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 after this is interpreted using that rotation controller until mouse-button release, even if it is no longer in the corresponding region.

3.4 Design and Implementation of Evaluation Programs

The controllers were implemented using three.js [8]. This was chosen to allow for easy deployment on the test equipment. The code was hosted on a static web page, with results saved onto the participant's computer using the Local Storage API in a JSON format.

As the system consisted of a number of discrete stages, it was implemented using a pipeline architecture. To ensure that the experiment can be resumed after a crash, the current stage of the pipeline is saved to Local Storage. Upon restart, if the browser detects a previously unfinished experiment it prompts the participant to resume. Refresh and back navigation display a warning to the user to prevent accidental data loss.

All rotations are performed using three.js's quaternion libraries. This was done to avoid Gimbal lock and to ensure uniform behaviour across all controllers.

4. METHOD

4.1 Heuristic Evaluation and Pilot

To ensure the quality of the rotation controllers, we had four 3DUI experts conduct a heuristic evaluation with Nielsen's Revised Heuristics [18] using early versions of the tasks used in the experiment. The evaluators reported problems detected, and potential solutions, using a standard form [23].

To further validate the design, we ran a remote pilot with 4 undergraduate students majoring in computer science obtained through convenience sampling. These students were given a link to the experiment via email and were told to allocate an hour of distraction-free time. They were incentivised to participate with a small monetary remuneration fee of 30ZAR. After they completed the experiment, we conducted an informal debriefing. This experiment followed the same method as outlined in Section 4.2 with earlier versions of the tasks.

4.2 Experimental Evaluation

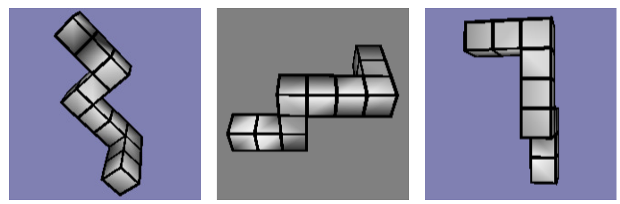
4.2.1 Experimental Design

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 modelling 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 (SRRA) [22]. This captured demographics and potential covariates such as gender, experience with 3D modelling software, and frequency playing video games. Since no participant listed themselves as less than moderately skilled with computers, all participants were eligible.

Experiments were performed in a closed lab with identical screen and mouse models. Participants received on-screen instructions before each stage of the experiment and were provided a printed manual for reference as needed. One participant was excluded from the analysis due to misreading the instructions, leaving 45 participants.

Table 3: 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



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

Figure 2: Screen shots of a simple and complex task matching task alongside the starting orientation of the model

Before the evaluations, participants were given 10 minutes to complete a Mental Rotation Test (MRT) [11], a common test for assessing ability to mentally rotate shapes [12]. 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, 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 presentation 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) [7] giving us a usability score as a third measure. Participants were also asked to include any other feedback they had as text.

4.2.2 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 3.

Our orientation tasks are similar to Chen et al. [9] but replace simple coloured houses with more complex models from the MRT. These models are designed to be orient mentally, making rotation tasks more challenging. We further divided the orientation tasks into two groups: Simple and Complex. Simple tasks require rotation using on one or two axes while Complex tasks require rotation about all three. This difference is illustrated in Figure 2.

The inspection tasks are based on those of Bade et al. [3]: 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 black circle visible on the screen.

Participants were first presented with Simple orientation tasks, then the Complex orientation tasks, and finally with the inspection tasks. Within each of these groups, the order of tasks was randomised.

4.2.3 Measures

For every task, we captured 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 calculated as the minimum angle required to rotate the participant’s 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 [15].

For inspection tasks, accuracy is the minimum angle required to orient the participant’s model so that the centre of the side containing only an “i” is in the centre 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 [4], discarding the z rotation component and transforming these angles to the corresponding quaternion. We then compare the users quaternion with the reference quaternion using the formula in 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).

5. DATA ANALYSIS AND RESULTS

5.1 Heuristic Evaluation and Pilot Experiment

The heuristic evaluation revealed twelve problems – eight high priority and four medium priority. All twelve problems detected were addressed before pilot experiments began.

In the pilot experiments, participants identified problems with instructions as well as reporting that the original inspection task, a colour finding task, was too easy. Because the task required participants to find the red side of the model, they could quickly manipulate the model and stop when the desired colour was spotted. This led to the creation of inspection tasks where duplicated letter “i”s on the sides of the model forced participants to spend more time examining each side of the shape.

5.2 Experimental Evaluation

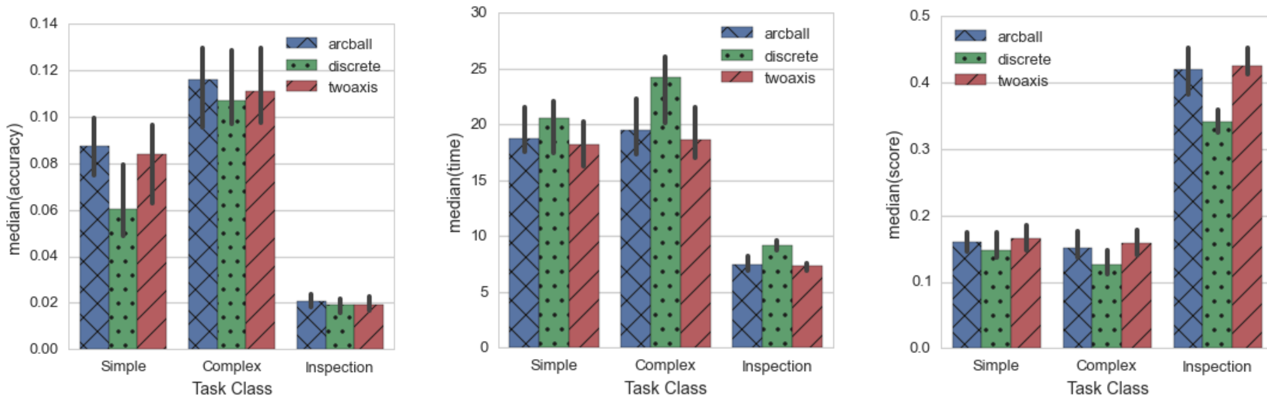
We analysed data from 45 participants performing 12 tasks (4 Simple orientation, 3 Complex orientation, and 5 inspection) and a usability questionnaire. This gave us 1620 task and 135 usability observations.

Results for a task submission were discarded if the accuracy of the task submissions was above 0.5 radians. This threshold was chosen as we regard anything above this as considered too inaccurate for a task to be considered successful. All values above this threshold were identified as outliers by a Grubb Outlier test. As this is a repeated measures experiment, we removed the corresponding task submissions for that participant across all 3 controllers. This resulted in 138 results of the 1620 total results gathered to be 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 D’Agostino and Pearson’s omnibus test of normality [10] 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 and its tasks, complex orientation, and inspection). This was done to determine if there was a significant difference in performance between the three controllers for that group of tasks. 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. The effect size r between the controllers was then calculated using the matched-airs rank-biserial correlation using the Wilcoxon Signed Rank statistic [17]. The results of these tests are presented in Table 4, Table 5, and Table 6. In these tables, $\tilde{c_1} - \tilde{c_2}$ is the difference of the median values (median diff.) of the pair of the controllers listed in the header for that column.



(a) Accuracy of submissions (b) Time of submissions (c) Score of submissions

Figure 3: Bar Charts with error bars (95th confidence interval) displaying the median values of the measures for task submissions

Table 4: Summary of results and statistical analysis of task results for task accuracy

	Friedman test		Median ($\tilde{c}$)		Arcball, Discrete		Arcball, Two Axis		Discrete, Two Axis	
	$\chi^2(2)$	Arcball	Discrete	Two Axis	r	$\tilde{c_1} - \tilde{c_2}$	r	$\tilde{c_1} - \tilde{c_2}$	r	$\tilde{c_1} - \tilde{c_2}$
Orientation	12.7357 *	0.0981	0.0863	0.0958	0.1727 *	0.0118 *	0.0627	0.0023	0.1152	-0.0095
Simple	15.5404 **	0.0874	0.0603	0.0842	0.2957 **	0.0272 **	0.0652	0.0032	0.1936	-0.0239
X	17.3023 **	0.0628	0.0293	0.0600	0.4618 *	0.0335 *	0.0720	0.0028	0.5415 *	-0.0307 *
Z	7.4286 †	0.0897	0.0416	0.0441	0.3984 ‡	0.0481 ‡	0.1080	0.0456	0.0871	-0.0026
X + Y	0.2105	0.1414	0.1667	0.1466	0.0469	-0.0253	0.0413	-0.0052	0.0185	0.0201
Y + Z	3.2105	0.0864	0.0674	0.0850	0.3457	0.0190	0.1067	0.0014	0.1380	-0.0176
Complex	0.8235	0.1159	0.1068	0.1109	0.0161	0.0091	0.0477	0.0050	0.0158	-0.0041
Inspection	2.8131	0.0207	0.0192	0.0192	0.1727	0.0015	0.0854	0.0014	0.0356	-0.0000

Table 5: Summary of results and statistical analysis of task results for task time

	Friedman test		Median ($\tilde{c}$)		Arcball, Discrete		Arcball, Two Axis		Discrete, Two Axis	
	$\chi^2(2)$	Arcball	Discrete	Two Axis	r	$\tilde{c_1} - \tilde{c_2}$	r	$\tilde{c_1} - \tilde{c_2}$	r	$\tilde{c_1} - \tilde{c_2}$
Orientation	5.7429	19.1120	21.6550	18.3195	0.1799 *	-2.5430 *	0.0067	0.7925	0.1443	3.3355
Simple	0.9814	18.7830	20.5850	18.2470	0.0477	-1.8020	0.0098	0.5360	0.0266	2.3380
X	0.3256	16.3650	13.0880	13.3270	0.1916	3.2770	0.1052	3.0380	0.0011	-0.2390
Z	0.7619	20.5750	22.0465	18.8050	0.0476	-1.4715	0.0105	1.7700	0.0337	3.2415
X + Y	0.3684	21.0100	21.3525	19.0835	0.0811	-0.3425	0.0498	1.9265	0.1721	2.2690
Y + Z	2.7368	20.2230	23.2575	21.5960	0.2546	-3.0345	0.2262	-1.3730	0.0071	1.6615
Complex	10.7059 *	19.4940	24.2790	18.6020	0.3331 *	-4.7850 *	0.0152	0.8920	0.2790 *	5.6770 *
Inspection	37.3551 ***	7.4050	9.1560	7.2955	0.4364 ***	-1.7510 ***	0.0116	0.1095	0.4746 ***	1.8605 ***

Table 6: Summary of results and statistical analysis of task results for task score

	Friedman test		Median ($\tilde{c}$)		Arcball, Discrete		Arcball, Two Axis		Discrete, Two Axis	
	$\chi^2(2)$	Arcball	Discrete	Two Axis	r	$\tilde{c_1} - \tilde{c_2}$	r	$\tilde{c_1} - \tilde{c_2}$	r	$\tilde{c_1} - \tilde{c_2}$
Orientation	5.8357	0.1567	0.1411	0.1645	0.1184	0.0156	0.0013	-0.0078	0.1298	-0.0234
Simple	0.9814	0.1605	0.1471	0.1662	0.0168	0.0134	0.0017	-0.0057	0.0356	-0.0191
X	0.3256	0.1888	0.2383	0.2285	0.1340	-0.0496	0.0166	-0.0397	0.0188	0.0098
Z	0.6190	0.1492	0.1380	0.1635	0.0430	0.0113	0.1823	-0.0143	0.0244	-0.0255
X + Y	0.3684	0.1457	0.1393	0.1580	0.1579	0.0064	0.0156	-0.0123	0.1636	-0.0187
Y + Z	3.0000	0.1512	0.1328	0.1426	0.1721	0.0184	0.2489	0.0086	0.0014	-0.0098
Complex	10.6387 *	0.1511	0.1261	0.1586	0.2377	0.0250	0.0110	-0.0075	0.2500 ‡	-0.0325 ‡
Inspection	36.3364 ***	0.4203	0.3411	0.4253	0.4797 ***	0.0792 ***	0.0044	-0.0050	0.5024 ***	-0.0842 ***

† $p < 0.05$
‡ $p < 0.0167$
* $p < 0.01$
** $p < 0.001$
*** $p < 0.0001$

5.2.1 *Accuracy*

The accuracy results are presented in Figure 3a and Table 4. We found a significant difference in the controllers for the orientation matching tasks ($\chi^2(2) = 12.74$, $p < 0.01$), with a statistically significant difference between Discrete Sliders and Arcball (median diff. $= 0.01$, $r = 0.17$, $p < 0.01$) in favour of the Discrete Sliders. Drilling into the matching tasks, we find significant differences in Simple Tasks ($\chi^2(2) = 15.54$, $p < 0.001$) with post hoc tests finding Discrete Sliders more accurate than Arcball (median diff. $=$ 0.03, $r = 0.30$, $p < 0.001$). We also found significant differences in the Simple X task ($\chi^2(2) = 17.30$, $p < 0.001$) with Discrete Sliders significantly more accurate than Two Axis Valuator (median diff. $= 0.03$, $r = 0.54$, $p < 0.01$).

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

5.2.2 *Speed*

The results of the time taken is presented in Figure 3b and Table 5. While no significant difference in time taken was found for completing the Simple orientation matching tasks, there was a significant effect of controller for the Complex matching tasks ($chi^2(2) = 10.71$, $p < 0.01$). The post hoc test showed that the Discrete Sliders controller was slower than both Arcball (median diff. $= 4.79$, $r = 0.33$, $p < 0.01$) and Two Axis Valuator (median diff. $= 5.68$, $r = 0.28$, $p <$ 0.01).

There was also a significant effect on speed for the inspection task ($\chi^2(2) = 37.36$, $p < 0.0001$). Again, significant differences between the Discrete Sliders and Arcball (median diff. $= 1.75$, $r = 0.44$, $p < 0.0001$), and between the Discrete Sliders and Two Axis Valuator (median diff. $=$ 1.86, $r = 0.47$, $p < 0.0001$) were found with the post hoc test, with the Discrete Sliders slower in both cases.

5.2.3 *Score*

The results of the submissions' score (accuracy normalised by time) is presented in Figure 3c and Table 6. Again, no significant difference was found for the Simple orientation matching tasks. However, for Complex orientation tasks, we found a significant difference between controllers ($\chi^2(2) = 10.64$, $p < 0.01$) with post hoc tests detecting a difference between the Discrete Sliders and Two Axis Valuator (median diff. $= 0.03$, $r = 0.25$, $p < 0.0167$) with the Two Axis Valuator having the higher score.

Furthermore, there was also a significant effect on score for the inspection task ($\chi^2(2) = 36.34$, $p < 0.0001$), with significant differences between the Discrete Sliders and Arcball (median diff. $= 0.08$, $r = 0.50$, $p < 0.0001$), and Discrete Sliders and Two Axis Valuator (median diff. $= 0.08$, $r =$ 0.50, $p < 0.0001$).

5.3 Usability Score and Feedback

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 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 preferred 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.

5.4 Confounding Factors

We collected data on twenty-six possible categorical 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 one to twice a year ($r = 0.42$ for speed); having no prior skill with 3D modelling software ($r = 0.32$ for speed). No factors impacted the difference in measures between Discrete Sliders and Arcball, or Discrete Sliders and Two Axis Valuator.

6. DISCUSSION

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

We also failed to detect significant differences in accuracy for inspection tasks. This can possibly be explained by the use of polished controllers and providing clear feedback in the inspection task. The inspection tasks provided implicit feedback for accuracy through the position of the dot of the "i" relative to the circular guide in the centre of the screen. Since the controllers are all capable of accurate orienting, as supported by the low median accuracy found, users could use this clear feedback with any controller to accurately complete the inspection task. This could lead to the lack of differentiation in accuracy seen. Bade et al [3] also failed to differentiate between inspection accuracy for Arcball and the Two Axis Valuator.

When compared to Discrete Sliders, the Arcball and Two Axis Valuator are moderately faster for both Complex Orientation and Inspection tasks. The speed increase may be attributed to the free-form nature of these controllers. It also may be due to the small clickable area covered by the sliders in the Discrete Sliders controller. 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 sliders, dividing the click target into four smaller regions.

The Discrete Sliders has the lowest median scores for all groups of tasks other than the Simple X orientation matching task. The differences in score are much smaller for the Simple orientation matching task than it is for the Complex or inspection tasks. This re-enforces the findings above: the Discrete Sliders are best suited for tasks where fine-grained accuracy is required above speed, whereas either the Arcball or Two Axis Valuator would be more effective if speed is more important for a task.

Taken together, these 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.

We only detected significant differences between Arcball and Two Axis Valuator for one task: the Bottom Inspection task (see Table 3). This contradicts results found by Bade et al. [3] where the Two Axis Valuator is faster Arcball across all inspection tasks. The difference here could be explained by our use of a modified Arcball instead of Shoemake's Arcball [21] as used in this studies. Another possible explanation is that while the method of rotation is dissimilar mechanically between them, their interfaces were more similar to each other than that of the Discrete Slider controller. It is also possible that the CSS cursor bug affected the comparison here, minimising any possible differences.

Previous experiments found usability differences between Arcball and Two Axis Valuator for both orientation matching [28] and inspection [3]. One possible explanation of our failure to detect usability differences is the usability questionnaire chosen. SUS, while robust, is not thorough as it only contains ten questions. It is difficult to compare our usability analysis to other studies, as they did not detail the questionnaires used to gather their usability measures.

Considering the effect of confounding factors, as all the significant differences found are for Arcball and Two Axis Valuator, we can conclude that there are no strong confounders affecting the comparisons between the Discrete Sliders and other controllers outlined above.

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.

Our first hypothesis was that the Discrete Sliders controller would perform more accurately and quickly in fine-grained orienting tasks. The Discrete Sliders was more accurate for simple (one- or two-axis) tasks than the Arcball controller (with a small effect size) and for simple one-axis tasks than 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 and to confirm or reject the hypothesis.

We also hypothesised that the Two Axis Valuator controller would be faster for inspection tasks. While we found the Two Axis Valuator and Arcball both faster than the Discrete Sliders, we did not manage to find differences in the performance of the Two Axis Valuator and Arcball aside from one of the five inspection tasks.

Overall, results suggest that the Discrete Slider are most useful in circumstances where accuracy is paramount, such us in 3D modelling suites. When absolute accuracy is less important, or speed is more important, 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 time span and then measure the accuracy achieved by participants.

Experimentation could also investigate novel controllers that make it easier to use the Arcball or Two Axis Valuator for initial model orientation and then use the sliders for more fine-grained orientation.

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 modelling and visualisation software.

8. ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisors Prof. James Gain and Dr. Brian DeRenzi, as well as my project partner Siobhan O'Donovan, for their guidance and immense knowledge. This project is far better having had their input.

I would also like to thank the Computer Science department for allowing the use of experimental space and equipment. Without their generosity, this project would not have been possible.

9. REFERENCES

[1] Autodesk. 3D Animation And Modeling Software | Maya | Autodesk. http://www.autodesk.co.za/products/maya/overview, 2015. [Online; accessed 2015-07-29].

[2] Autodesk. 3D Modeling & Rendering Software | 3ds Max 2016 | Autodesk. www.autodesk.co.za/products/3ds-max/overview, 2015. [Online; accessed 2015-07-29].

[3] R. Bade, F. Ritter, and B. Preim. Usability comparison of mouse-based interaction techniques for predictable 3D rotation. In *Proceedings of the 5th International Conference on Smart Graphics*, SG'05, pages 138–150, Berlin, Heidelberg, 2005. Springer-Verlag.

[4] P. Berner. Technical concepts: Orientation, rotation, velocity and acceleration, and the SRM. Technical report, SEDRIS, 2008.

[5] Blender Foundation. blender.org - Home of the Blender project - Free and Open 3D Creation Software. https://www.blender.org/, 2015. [Online; accessed 2015-07-29].

[6] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Company Inc., Redwood City, CA, USA, 2004.

[7] J. Brooke. SUS: a 'quick and dirty' usability scale. In P. Jordan, B. Thomas, I. McClelland, and B. Weerdmeester, editors, *Usability evaluation in industry*, pages 189–194. Taylor & Francis Ltd., London, UK, 1996.

[8] R. Cabello. three.js - javascript 3D library. http://threejs.org/, 2015. [Online; accessed 2015-07-30].

[9] M. Chen, S. J. Mountford, and A. Sellen. 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*, SIGGRAPH '88, pages 121–129, New York, NY, USA, 1988. ACM.

[10] R. D'Agostino and E. S. Pearson. Tests for departure from normality. empirical results for the distributions of b^2 and $\sqrt{b^1}$. *Biometrika*, 60(3):613–622, 1973.

[11] S. G. Gandenberg and A. R. Kuse. Mental rotations, a group test of three-dimensional spatial visualization. *Perceptual and motor skills*, 47(2):599–604, 1978.

[12] M. Harle and M. Towns. A review of spatial ability literature, its connection to chemistry, and implications for instruction. *Journal of Chemical Education*, 88(3):351–360, 2010.

[13] K. Henriksen, J. Sporring, and K. Hornbæk. Virtual trackballs revisited. *Visualization and Computer Graphics, IEEE Transactions on*, 10(2):206–216, March 2004.

[14] K. Hinckley, J. Tullio, R. Pausch, D. Proffitt, and N. Kassell. Usability analysis of 3D rotation techniques. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, pages 1–10, New York, NY, USA, 1997. ACM.

[15] D. Q. Huynh. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[16] I. Jacob and J. Oliver. Evaluation of techniques for specifying 3D rotations with a 2D input device. In *Proceedings of the HCI'95 Conference on People and Computers X*, HCI '95, pages 63–76, New York, NY, USA, 1995. Cambridge University Press.

[17] D. S. Kerby. The simple difference formula: an approach to teaching nonparametric correlation. *Innovative Teaching*, 3(1), 2014.

[18] J. Nielsen. Heuristic evaluation. *Usability inspection methods*, 17(1):25–62, 1994.

[19] T. D. Parsons, P. Larson, K. Kratz, M. Thiebaux, B. Bluestein, J. G. Buckwalter, and A. A. Rizzo. Sex differences in mental rotation and spatial rotation in a virtual environment. *Neuropsychologia*, 42(4):555–562, 2004.

[20] T. Partala. Controlling a single 3D object: viewpoint metaphors, speed and subjective satisfaction. In *Proc. of INTERACT*, volume 99, pages 536–543, Edinburgh, UK, 1999. IOS Press.

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

[22] M. S. Terlecki and N. S. Newcombe. 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, 2005.

[23] UX Direct. FAQ: How to conduct heuristic evaluation. Technical report, Oracle Corp., 2012.

[24] D. Voyer, S. Voyer, and M. P. Bryden. Magnitude of sex differences in spatial abilities: a meta-analysis and consideration of critical variables. *Psychological bulletin*, 117(2):250, 1995.

[25] Y. Wang, C. L. MacKenzie, V. A. Summers, and K. S. Booth. The structure of object transportation and orientation in human-computer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, pages 312–319, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[26] C. Ware and J. Rose. Rotating virtual objects with real handles. *ACM Trans. Comput.-Hum. Interact.*, 6(2):162–180, June 1999.

[27] H. Wenjun. The three-dimensional user interface. In S. Pinder, editor, *Advances in Human Computer Interaction*. INTECH Open Access Publisher, Rijeka, Croatia, 2008.

[28] Y. J. Zhao, D. Shuralyov, and W. Stuerzlinger. Comparison of multiple 3D rotation methods. In *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2011 IEEE International Conference on*, pages 1–5, Ottawa, ON, 2011. IEEE, IEEE.