Review of Evaluations of Mouse Based 3D Rotation Techniques

Steven Rybicki, University of Cape Town

3D Rotation is one of the major bottlenecks for manipulating a 3D scene. In a mouse based 3D manipulation program, we are able to translate objects in a couple seconds. However, it takes tens of seconds to rotate an object to a desired orientation. As this hampers our ability to perform tasks such as editing 3D scenes or viewing 3D medical scans, it has been a research priority to find better 3D rotation techniques. To assess our progress towards finding these techniques, I have evaluated the literature on experiments evaluating the efficacy of mouse based 3D rotations. I determined that effective evaluations of rotations should include analysis of their use when performing representative tasks, measuring user speed and accuracy (task performance metrics). They should also record user presence, learnability and comfort (user preference metrics). Of the six studies I found, every study had issues with obtaining statistically significant results. Analysis revealed many possible causes, including simplistic task design and small sample sizes. Additionally, I found problems with replicability, the types of tasks used to assess rotations, the task performance metrics measured and the user preferences gathered. Together, these issues made it difficult to determine which 3D rotation technique was most performant, though there is evidence to suggest that Virtual Sphere and the Two Axis Valuator techniques show promise.

Categories and Subject Descriptors: I.3.6 [**Computer Graphics**]: Methodology and Techniques - Interaction Techniques; H.1.2 [**Models and Principles**]: User/Machine Systems - human factors

General Terms: Human Factors, Design, Experimentation, Performance

Additional Key Words and Phrases: 3D User Interfaces, 3D Rotation, 3D Modeling, Direct Manipulation, 3D object manipulation, 2D mouse, Arcball, Virtual Sphere, Usability study

1. INTRODUCTION

3D scene manipulation techniques are important as they facilitate a number of applications with great significance. These include creating virtual scenes using modeling software such 3DMax [Autodesk 2015], examining 3D models created from medical scans [Bade et al. 2005] and the exploration of virtual worlds [Bowman et al. 2004].

The act of manipulating a 3D scene can be divided into three different sub-tasks. These are: the selection of objects, positioning those objects in an environment and rotating those objects into a desired orientation [Wenjun 2008]. We have effective methods for object selection and translation, allowing simple positioning in two to three seconds [Ware and Rose 1999]. However, while we can rotate physical cubes to a desired orientation in under a second [Wang et al. 1998], we take tens of seconds to undertake an equivalent task using a virtual system [Hinckley et al. 1997]. It is not surprising that in systems where orientation is important, users spend the majority of their time trying to get the correct rotation for an object [Shuralyov and Stuerzlinger 2011]. Thus there is potentially room for improvement for the usability of 3D rotation methods, allowing significant gains for the usability of 3D manipulation systems as a whole.

In this review we are critically examining evaluations of 3D rotation methods. I am focusing exclusively on mouse based rotations, as opposed to 3D rotation with either two handed techniques or techniques that involve the use of other peripherals. There is evidence that specialised external peripherals can be more effective than mice alone [Hinckley et al. 1997] and there is active research into translating 3D rotation interactions into mobile devices [Buda 2012]. However, at the time of writing, mice are the most prevalent input devices used for 3D rotations. This means any gains for mouse based rotations will have a large impact for professionals needing to manipulate 3D scenes. For a more complete summary of different input methods for 3D rotations, see Buda [2012].

In Section 2 I outline the standard evaluation techniques to evaluate a 3D interface. Section 3 lists and explains all the 3D rotation methods I found that have been experimentally verified. I then summarise and critique the experiments founds in Section 4. Finally I conclude with a holistic overview in Section 5, suggesting areas where future work could be done.

2. TECHNIQUES FOR EVALUATING 3D ROTATION INTERFACES

To determine the efficacy of an interaction technique, one usually does a heuristic analysis along with a user evaluation. Unfortunately, at the time of writing there are no experimentally verified guidelines for 3D UI Design [Bowman et al. 2004]. Without such guidelines, it is not feasible to perform credible analysis on 3D rotation methods using heuristics alone. Analysis of a technique is therefore done through an experimental evaluation measuring user satisfaction and performance metrics. I outline the current literature for all three methods in this section.

2.1. Heuristic based analysis

Heuristic based analysis focuses on using heuristics to predict how well an interface will accomplish a task. Bade et al. [2005] was the only source I could find that posited a set heuristics solely for 3D rotation. However, these heuristics were based purely on their own personal experience and were not formally verified except shallowly with one experimental study in the same paper.

Despite there not being a viable heuristic framework, there are a number of heuristics that are mentioned consistently across the literature, namely Schneiderman's Direct Manipulation [Shneiderman 1993], hysteresis [Shoemake 1992] and the Control-Display Ratio [Bade et al. 2005] [Dominjon et al. 2005].

Direct Manipulation, as explained by Schneiderman [1993] is an interface where "what you see is what you get". Practically, for 3D rotation methods, this means

(1) Having a continuous representation of the object you are rotating.
(2) Rotating the object using straightforward physical actions, like mouse drags and presses.
(3) Implementing rotations through rapid, incremental and reversible operations which are immediately applied.
(4) Creating a rotation system which has layers of features, allowing novices to accomplish rotations with minimal learning curve, allowing them to practice a certain set of features until they feel like they have mastered them.

Application of the third principle of direct manipulation gives us that rotations should be easily reversible. Intuitively this means that if a user applies a set of rotations, it should be obvious how to reverse these rotations. However, Shoemake [1992] notes that certain 3D rotation methods exhibit a form of hysteresis: if you rotate an object by $\frac{\pi}{2}$ in two axes, then $\frac{-pi}{2}$ in those axes in the same order, certain systems will not lead you back to the same orientation you started with. This breaks a user's intuition and makes it harder to reason about rotations. As such, systems that do not have this hysteresis effect are seen as desirable.

The Control-Display Ratio (C-D Ratio) is the ratio of amplitudes of movements of a user's hand and the corresponding movements of the virtual cursor that user controls in a virtual environment [Dominjon et al. 2005]. This can be tweaked to allow for different use cases: a C-D Ratio ≈ 1 makes the connection between mouse movement and object motion more connected, while a small C-D ratio makes it easier to accomplish fine motion controls [Chen et al. 1988]. This makes it advantageous to be able to adjust the C-D Ratio, making such systems more dynamic over systems that cannot.

2.2. Task Performance Metrics

User task performance metrics generally refer to how effective an interface is at accomplishing a particular task [Bowman et al. 2004]. These can be either general (length of time to match orientation of two platonic solids) or domain specific (length of time taken to identify area of interest on a model of a human heart).

Broadly, there are two types of tasks that are generally defined as important for 3D rotation interfaces: orientation matching and inspection tasks [Henriksen et al. 2004]. In orientation matching, one is given a model in a particular orientation and asked to manipulate a copy of that model until it matches that orientation. Inspection tasks has a user rotating a shape looking for areas of interest.

Orientation matching is intended to be the representative task for manipulating 3D models to the desired orientation in a modeling package [Chen et al. 1988]. Inspection tasks are meant to emulate tasks where you want to determine whether a particular model contains certain properties, like evaluating a 3D brain scan result [Bade et al. 2005]. These tasks can be specified at different degrees of complexity. This usually accomplished by increasing the number of axes to rotate the object, or by modifying the object itself. This is done through increasing the number of features of the object or using objects that are less familiar to the user.

The task performance metric used by studies to assess these tasks is speed and accuracy. This means determining how accurately and quickly a user is able to perform a rotation. These can further be tuned by biasing a user towards a particular metric. For example, Chen et al. [1988] gave users a report on how accurate they were after a task. Bade et al. [2005] told users to perform the task as quickly as they could.

2.3. User Preference Metrics

While accuracy and speed are important, these metrics alone do not measure the usability of a system. Metrics such as how easy a system is to learn, how comfortable a user is using that system, and how present a user feels in the virtual environment, also affect how well a user performs tasks in a given system [Bowman et al. 2004]. These are user preference metrics, subjective perception of well the interface performs, and are usually gathered using a questionnaire.

Learning effects are related to the fourth principle outlined for direct manipulation in Section 2.1. We want users to be able to quickly learn how to use and navigate an interface to complete rotation tasks. We also want them to feel like this is intuitive.

User comfort in 3D interfaces is primarily concerned with simulator sickness as well as the strain on arms and eyes from use with a 3D device [Bowman et al. 2004]. These issues tend to not arise using a mouse based 3D rotation interface, but are still important to capture.

Presence for 3D interfaces is defined as the feeling of being present in the 3D environment. This is the product of using a transparent user interface - an interface that makes a user feel like they are working directly on the problem. This is usually measured qualitatively on a scale, but can be examined using physical reactions [Bowman et al. 2004].

Other more traditional user preference metrics are gathered by studies, including ease of use, predictability of the system and a subjective comparison of the tested systems. See Table I for a list of the metrics used.

3. METHODS OF ROTATION

Early rotation methods tended to focus on users selecting the rotation they wanted per axis, via individual sliders, buttons or menus [Chen et al. 1988]. Individual sliders would be placed under the object, one per axis, allowing the user to accurately choose their rotation. Chen et al. [1988] proposed an improvement of this: instead of a single slider per axis underneath the object, they overlaid overlapping sliders on top of the object. XY sliders were put at right angles with each other across the object, with a Z slider placed as a circle around the outside of them.

A further extension of this idea by Chen et al. [1988] was the Continuous XY with added Z controller (XY+Z controller), where the user specified rotations using a cursor's movement inside and outside a circle on the center of the screen. Distance from the center was used to specify XY rotations, while movement outside in a clockwise or anticlockwise direction was used to control Z rotation. Evans et al. [1981] created a controller without any screen overlay with a similar idea, which is known as Evans controller. This is a gesture based controller, where mouse movement in straight lines controls the XY rotation, while clockwise and anti clockwise motions control the Z rotation.

These approaches were supplanted with the idea of a virtual trackball [Shoemake 1994]. Instead of discretely specifying each axis of rotation separately, you would imagine that your object was in a ball and you were pushing that ball in different directions. This idea was extended by Hanson's rolling ball [Hanson 1992], where the object of rotation was inside a ball that was viewed to be constantly rolling in a direction, with the user using the mouse to control it rotating in certain directions. Due to a property of rotating groups, this allowed one to achieve any 3D orientation despite only having two dimensions of input. This was also explored with Chen's Virtual Sphere [Chen et al. 1988], where users dragged a sphere with their mouse to indicate their direction. Chen's Virtual Sphere was extended with Bell's Virtual Trackball (Bell's VT) [Henriksen et al. 2004], which improved the implementation to focus on using a hyperbolic arcs rather than just a sphere's surface.

Shoemake [1992] identified that the above models suffered from a hysteresis effect: for Virtual Sphere and related interfaces, a sequence of rotations followed by a reversed sequence of rotations in the opposite direction does not necessarily lead to the same initial orientation. They then specified Arcball, a virtual trackball interface that avoids hysteresis. Instead of rolling a sphere, a user draws arcs on the surface of a sphere. The sphere is then rotated according to this arc. Due to the mathematics used, the sphere is set at a fixed C-D. This restriction, along with its system of half length arcs, allows it to avoid hysteresis.

Another method that avoids hysteresis is the Two-Axis Valuator Trackball or the Two-Axis Valuator [Chen et al. 1988]. This method restricts the user by allowing them to only rotating in two dimensions. This is usually specified as the XY dimensions relative to the current camera view. However, in certain applications with an objective fixed direction system, the rotations are given as XY dimensions relative to some reference plane. This is termed Fixed-right/up Trackball (Fixed Trackball) [Buda 2012] and is used in applications where a high degree of precision is necessary such as in 3DMax [Autodesk 2015]. Partala [1999] proposed a modification of this model, known as the virtual rectangle. A rectangular plane is placed inside the object of rotation. The X and Y axes of rotation are controlled by the xy position of the cursor on the rectangle. Z rotations are controlled by clockwise and anticlockwise motions outside of this rectangle.

4. EXPERIMENTS EVALUATING 3D ROTATION METHODS

4.1. Analysis of Experimental Designs Used

I found six experiments that focus on testing mouse based 3D rotation techniques. The studies objectives and methodologies are summarised in Table I. Their results are summarised in Table II.

From looking at the papers in aggregate, we can observe problems with attaining statistically significant results. For example, Hinckley et al. [1997] failed to get any statistically significant differences in performance between Arcball and Virtual Sphere. Jacob et al. [1995] failed to get any statistically significant differences in speed of rotation from their orientation tasks, while Chen et al. [1988] did using the same evaluation techniques and rotation methods. The number of statistically significant different results obtained was low as seen by the number of ties in the results summary table.

Another problem I found is with the replicability of results. When Hinckley et al. [1997] performed an experiment with the same tasks used by Chen et al. [1988], they failed to get comparable task completion times for Virtual Sphere. Even using the same tests, Chen et al.[1988] got subtly different results between their two experiments.

While the experiments that use orientation matching tasks are all close to the design by Chen et al. [1988], the task completion times are often very different. Experiments also fail to completely specify the implementation of the controllers used. This makes it difficult to assess whether the methods, although named the same, are actually able to be compared directly. For example, Jacob et al. [1995] failed to fully specify the tasks they used, making the possible replication of their study difficult.

There are a number of explanations for why these problems are across the field. One is the number of subjects - besides the 137 person study by Jacob et al. [1995], the average number of subjects in an experiment was 15, with the smallest being 6.

Another is that there are possibly confounding effects in their subject pool. Every study used university students, but these pools differed widely. The pools all have a different selection of backgrounds (from Masters Engineering students to undergraduate psychology majors), sex ratios (there is often significant variation in ability across sex [Voyer et al. 1995] [Parsons et al. 2004]) and amount of experience with 3D rotational interfaces.

Some studies did control for these confounding factors. Bade et al. [2005] divided their subjects into pools based on sex, finding that with the exception of one method their pool did not have significant differences in performance based on sex. Zhao et al. [2011] conducted an ANOVA analysis to control for individual variation. But, with the exception of two experiments, these confounding effects were not controlled for. Hinckley et al. [1997] noted a difference of ability across sex and said it merited further study, but did not go further. Jacob et al. [1995] tested their sample for 3D rotational aptitude but did not factor this into their analysis.

A problem noted by Henriksen et al. [2004] is that the tasks used by the studies prior to 2004 tended to be basic, orientation matching tasks. Chen et al. [1988] created a matching task based on 3D houses, where the subject has to orientate a simple house with differently coloured sides to match a reference image. This house is rotated in one or two axes in fixed intervals. The only experiment that I found that did not use a task very similar to this was Bade et al [2005], who did an inspection task based on complex geometric shapes. Zhao et al. [2011] used a similar task structure, but with more complex rotations required. This task is problematic, as it does not effectively measure real world performance: objects of rotations are often not simple nor are they at nicely predefined angles. Henriksen et al. [2004] also suggested that this simplicity could reduce the differences in performance of the different controllers. This could mask the differences in controllers, leading to the problems with results I have found.

Henriksen et al. [2004] also noted that experiments tended to focus on biasing subjects towards accuracy rather than speed. This was either accomplished by explicitly telling subjects to focus on accuracy, by giving feedback on accuracy at the end of experiments [Chen et al. 1988] or by giving the subjects a time that was acceptable to finish the task in[Zhao et al. 2011]. Coupled with the focus on Orientation Matching, the majority of research seems to be orientated towards the use case of manipulating 3D scenes. However, there are fewer studies focusing on speed or inspection tasks. This makes it harder to determine which controller would be best suited for other tasks, such as 3D medical scan viewing. The only study we could find examining inspection tasks was done by Bade et al. [2005].

To determine how users would respond to the controller in real world use cases, all the experiments did some degree of user preference analysis. This was done at 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 overall experience. The others captured specific usability metrics, focusing on ease of use, predictability of the controller's behaviour and perceived performance attributes. No paper captured all the metrics Bowman [2004] outlined as important, namely user comfort, ease of learning and presence in the virtual environment. This makes it difficult to determine how users would react to using these controllers when released as part of an integrated 3D scene manipulation package.

4.2. Results

The results for the experiments found are summarised in Table II.

Due to the construction of the experiments, it is difficult to determine which method of rotation is most performant or usable. As seen in Table I, of the eleven methods tested, only two were tested in more than two experiments: Virtual Sphere was tested five times; Arcball was tested three times. Of the remaining nine methods, five were tested in two different studies and four were only tested in one. This would be acceptable if all the studies used the same experimental design, but as outlined in the previous section we can see that there are differences that make this comparison difficult. As such, we can only consider the intra-study differences and try to use that to infer the performance of the controllers.

Non-trackball methods tend to do poorly for performance based metrics. Slider based methods along with Evans Controller and the Keyboard tend to be slower than the other methods. We can see that Overlapping Sliders and XY+Z Controllers are useful for accurate results, but that they were slow.

The Virtual Sphere was ranked highly in the pre-2005 orientation matching studies for the metrics of speed and accuracy. However, Bell's VT, a slight improvement of this controller, was much slower than Arcball and the Two-Axis Valuator for inspection tasks of Bade et al. [2005]. Arcball was intended to be more usable than Virtual Sphere due to its lack of hysteresis, and it did perform more quickly than Bell's VT in the study by Bade et al. [2005] However, in both Bade et al. [2005] and Zhao et al. [2011] it was rated as less usable than Bell's VT.

The Two-Axis Valuator technique seems promising, for inspection tasks, performing quickest and being chosen as most usable in the study by Bade et al. [2005]. However, the Fixed Trackball - the variation of the Two-Axis Valuator with a fixed up/right vector, performed differently. It came dead last in speed, accuracy and usability. This suggests that the actual implementation of a system is incredibly important to its functioning.

5. CONCLUSION

To evaluate the experimental evidence present for methods of 3D rotation, I had to determine what criteria one should be evaluating such experiments with. While most user interface analysis is done using heuristics, there are unfortunately no experimentally verified heuristics I could use to analyse rotations.

Instead, I rely on task performance and user preference metrics gained through experimentation. For rotation, task performance is defined as the speed and accuracy a user is able to undertake representative 3D rotation tasks. I then identified three important user preference metrics that are important for a user to be able to use a controller: user comfort, presence and learnability. To be confident in the experimental evidence for a method, it needs to be tested against these metrics. This means testing how users perform in an array of representative tasks and having them polled for these preference metrics.

After evaluating these studies, I found all but one by Bade et al. [2005] used simple tasks that were not representative of the complexity faced in real rotation tasks. I also found that the majority of research was biased towards accuracy and orientation matching rather than speed and inspection. With the exception of Bade et al. [2005], no paper captured a user preference metric identified as important in our analysis.

Continuing my analysis of the research, I found that there were problems with the design and reporting of the experiments. All but Bade et al. [2005] failed to systematically report and control for confounding factors. Papers specified the mathematics of their techniques, but did not specify how exactly they were implemented.

These issues together with the large number of statistically insignificant results make it difficult to analyse the methods of rotation. Using the available evidence, I was able to identify Virtual Sphere and Two-Axis Valuator as having potential to be the most performant and usable methods. Stronger claims would require further experimentation to be done.

Future work could focus on more sophisticated experiments. These could include controlling for all known confounding factors for rotation, using larger sample sizes and using more complex rotation tasks. All these together could help with attaining more statistically significant results as well as allow the results to be more applicable to real life applications. Studies could also study the usability aspects more closely, focusing on presence and learnability of the systems.

REFERENCES

Autodesk. 2015. 3d Modeling and Rendering Software — 3ds Max — Autodesk. (2015). http://www.autodesk.co.za/products/3ds-max/overview

Ragnar Bade, Felix Ritter, and Bernhard Preim. 2005. Usability comparison of mouse-based interaction techniques for predictable 3d rotation. In *Smart graphics*. Springer, 138–150.

Doug A Bowman, Ernst Kruijff, Joseph J LaViola Jr, and Ivan Poupyrev. 2004. *3D user interfaces: theory and practice*. Addison-Wesley.

Vlad Buda. 2012. *Rotation Techniques for 3D Object Interaction on Mobile Devices*. Ph.D. Dissertation. Utrecht University.

Michael Chen, S Joy Mountford, and Abigail Sellen. 1988. A study in interactive 3-D rotation using 2-D control devices. In *ACM SIGGRAPH Computer Graphics*, Vol. 22. ACM, 121–129.

Lionel Dominjon, Anatole Lécuyer, J-M Burkhardt, Paul Richard, and Simon Richir. 2005. Influence of control/display ratio on the perception of mass of manipulated objects in virtual environments. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE.* IEEE, 19–25.

Kenneth B Evans, Peter P Tanner, and Marceli Wein. 1981. Tablet-based valuators that provide one, two, or three degrees of freedom. In *ACM SIGGRAPH Computer Graphics*, Vol. 15. ACM, 91–97.

Andrew J Hanson. 1992. The rolling ball. *Graphics Gems* 3 (1992), 51–60.

Knud Henriksen, Jon Sporring, and Kasper Hornbæk. 2004. Virtual trackballs revisited. *Visualization and Computer Graphics, IEEE Transactions on* 10, 2 (2004), 206–216.

Ken Hinckley, Joe Tullio, Randy Pausch, Dennis Proffitt, and Neal Kassell. 1997. Usability analysis of 3D rotation techniques. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*. ACM, 1–10.

Inés Jacob and Javier Oliver. 1995. Evaluation of techniques for specifying 3D rotations with a 2D input device. In *BCS HCI*. 63–76.

Thomas D Parsons, Peter Larson, Kris Kratz, Marcus Thiebaux, Brendon Bluestein, J Galen Buckwalter, and Albert A Rizzo. 2004. Sex differences in mental rotation and spatial rotation in a virtual environment. *Neuropsychologia* 42, 4 (2004), 555–562.

Timo Partala. 1999. Controlling a single 3D object: viewpoint metaphors, speed and subjective satisfaction. In *Proc. of INTERACT*, Vol. 99. 486–493.

Ben Shneiderman. 1993. 1.1 direct manipulation: a step beyond programming languages. *Sparks of innovation in human-computer interaction* 17 (1993), 1993.

Ken Shoemake. 1992. ARCBALL: a user interface for specifying three-dimensional orientation using a mouse. In *Graphics Interface*, Vol. 92. Citeseer, 151–156.

Ken Shoemake. 1994. Arcball Rotation Control. *Graphics Gems IV* (1994).

Dmitri Shuralyov and Wolfgang Stuerzlinger. 2011. A 3D desktop puzzle assembly system. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 139–140.

Daniel Voyer, Susan Voyer, and M Philip Bryden. 1995. Magnitude of sex differences in spatial abilities: a meta-analysis and consideration of critical variables. *Psychological bulletin* 117, 2 (1995), 250.

Yanqing Wang, Christine L MacKenzie, Valerie A Summers, and Kellogg S Booth. 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., 312–319.

Colin Ware and Jeff Rose. 1999. Rotating Virtual Objects with Real Handles. *ACM Trans. Comput.-Hum. Interact.* 6, 2 (June 1999), 162–180. DOI:http://dx.doi.org/10.1145/319091.319102

Hou Wenjun. 2008. *The Three-Dimensional User Interface*. INTECH Open Access Publisher.

Yao Jun Zhao, Dmitri Shuralyov, and Wolfgang Stuerzlinger. 2011. Comparison of multiple 3d rotation methods. In *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2011 IEEE International Conference on*. IEEE, 1–5.

Table I: Summary of the methods used by experiments evaluation mouse based 3D rotation methods

Experiment	Methods Tested	No. of subjects	Rotation Task	Task Performance Metric	User Preference Metrics
Chen et al. [1988] Experiment 1	Individual Sliders Overlapping Sliders Virtual Sphere XY+Z Controller	12	Orientation Matching	Accuracy	N/A
Chen et al. [1988] Experiment 2	Evans Controller Virtual Sphere	6	Orientation Matching	Accuracy	N/A
Jacob et al. [1995]	Evans Controller Overlapping Sliders Virtual Sphere XY+Z Controller	137	Orientation Matching and Inspection	Accuracy	Easy of Use Perceived Precision and Speed
Hinckley et al. [1997]	Arcball Virtual Sphere	24	Orientation Matching	Accuracy	Comparative overall usability
Partala [1999]	Keyboard Controls Virtual Rectangle Virtual Sphere	12	Orientation Matching	Speed	Comparative overall usability
Bade et al. [2005]	Arcball Bell's VT Two-Axis Valuator Fixed Trackball	25	Inspection	Speed	User comfort, Predictability of Controller Behaviour
Zhao et al. [2011]	Arcball Bell's VT Two-Axis Valuator	12	Orientation Matching	Accuracy	Ease of Use, Perceived Performance, Predictability, Accuracy, Overall Usability

Table II: Summary of the experimental results per rotation method

Rotation Method	*Rank for Speed*	*Rank for Accuracy*	*Rank for Usability*
Overlapping Sliders	$3/4^{*}$ $\underline{2/2}^{\dagger}$	$\underline{1/2}^{\ddagger}$	
Individual Sliders	$4/4^{*}$		
Evans Controller	$\underline{2/2}^{\dagger}$	$2/2^{\ddagger}$	$2/2^{*}$
XY+Z Controller	$2/4^{*}$ $\underline{2/2}^{\dagger}$	$\underline{1/2}^{\ddagger}$	
Virtual Sphere	$1/4^{*}$ $1/2^{\dagger}$ $\underline{1/2}^{\S}$	$\underline{1/2}^{\ddagger}$	$1/2^{*}$ $2/3^{\S}$
Bell's VT	$\underline{3/3}^{\|}$	$1/2^{\|}$	$\underline{2/3}^{\|}$ $1/3^{\P}$
Arcball	$2/3^{\|}$	$\underline{2/2}^{\|}$	$\underline{2/3}^{\|}$ $3/3^{\P}$
Two-Axis Valuator	$1/3^{\|}$	$\underline{2/2}^{\|}$	$1/3^{\|}$ $2/3^{\P}$
Fixed Trackball	$\underline{3/3}^{\|}$	$\underline{2/2}^{\|}$	$3/3^{\|}$
Virtual Rectangle	$\underline{1/2}^{\S}$		$1/3^{\S}$
Keyboard	$2/2^{\S}$		$3/3^{\S}$

Underlined text: If a rank is underlined, e.g. $\underline{2/2}^{\dagger}$, then that method tied with at least one other method for that metric

*Chen et al. [1988] Orientation Matching Tasks
†Jacob et al. [1995] Inspection Tasks
‡Jacob et al. [1995] Orientation Matching Tasks
§Partala [1999] Orientation Matching Tasks
$^{\|}$Bade et al. [2005] Inspection Tasks
¶Zhao et al. [2011] Orientation Matching Tasks