Usability Comparison of Mouse-based Interaction Techniques for Predictable 3d Rotation

Ragnar Bade[1], Felix Ritter[2], and Bernhard Preim[1]

[1] University of Magdeburg, Universitätsplatz 2, D-39106 Magdeburg, Germany,
`bade@isg.cs.uni-magdeburg.de`
[2] MeVis - Center for Medical Diagnostic Systems and Visualization,
Universitätsallee 29, D-28359 Bremen, Germany

Abstract. Due to the progress in computer graphics hardware high resolution 3d models may be explored at interactive frame rates, and facilities to explore them are a part of modern radiological workstations and therapy planning systems. Despite their advantages, 3d visualizations are only employed by a minority of potential users and even these employ 3d visualizations for a few selected tasks only. We hypothesize that this results from a lack of intuitive interaction techniques for 3d rotation. In this paper, we compare existing techniques with respect to design principles derived by clinical applications and present results of an empirical study. These results are relevant beyond clinical applications and strongly suggest that the presented design principles are crucial for comfortable and predictable interaction techniques for 3d rotation.

1 Introduction

3d visualization offers a great potential for medical applications, such as therapy planning and surgical education. 3d models of relevant anatomic structures may be derived by means of CT or MRI data. Due to the progress in computer graphics hardware high resolution 3d models may be explored at interactive frame rates, and facilities to explore them are a part of modern radiological workstations. Despite their advantages, 3d visualizations are only employed by a minority of potential users and even these employ 3d visualizations for a few selected tasks only. Instead, medical doctors prefer to ”read” CT or MRI data slice by slice.

We hypothesize that one reason for this situation is the lack of intuitive interaction techniques for 3d rotation. Medical doctors attempt to explore geometric models systematically (looking at each part of an organ for example). Hence, they require a predictable behavior which enables them to anticipate the effect of an interaction and to return to a previous viewing direction.

Software systems for geometric modelling, Computer-Aided Design and surgery planning provide a variety of interaction techniques for 3d rotation[3]. In this

---

[3] For readability we will use “rotation techniques” as a synonym for “interaction techniques for 3d rotation”.

paper, we compare such techniques with respect to design principles inspired by clinical applications. The comparison of 3d rotation techniques as well as the discussed principles are relevant beyond clinical applications. We focus on users controlling 3d rotation by means of a standard 2d mouse and do not consider dedicated 3d input devices since these are still rare.

We evaluate direct manipulation techniques because we assume that this interaction style is superior to indirect controllers such as scroll wheels for 3d rotation. Bimanual (two-handed) interaction techniques [2] as well as interaction techniques which require 3d pointing devices are not considered.

First, we present a list of principles for predictable and convenient rotation techniques in Sect. 2. Subsequently, we discuss state of the art direct manipulation rotation techniques for 2d pointing devices in Sect. 3 and in Sect. 4, we report and discuss results of an empirical user study.

2 Design Principles for 3D Rotation Techniques

Based on our experience in clinical applications (cp. [13]), we identified the following four general principles as crucial for predictable and pleasing rotation techniques:

1. *Similar actions should provoke similar reactions.*
2. *Direction of rotation should match the direction of 2d pointing device movement.*
3. *3d rotation should be transitive.*
4. *The control-to-display (C/D) ratio should be customizable.*

The first is a general principle for human-computer interaction which emphasizes the importance of a predictable and reproducible behavior. As an example, the same mouse movement should not result in varying cursor movements.

The second principle applies to stimulus-response (S-R) compatibility and kinesthetic correspondence [4] between the direction of user action (e.g. mouse movement) and the direction of computer reaction (e.g. object rotation).

The third principle completes the first two by taking into account the natural transitive behavior of movements in an Euclidean space (human world). Since a pointing device movement from point A to point B and then to point C ends at the same location as a direct movement from A to C, this should also be true for the corresponding reaction of such an action. We identified this principle as crucial to enable users to return to the initial viewing position.

While the first three principles are important for a predictable behavior, the fourth principle takes user and application needs into account. Customizing the control-to-display (C/D) ratio [5] is necessary to find the best compromise between speed and accuracy according to the task and user preferences and is therefore crucial for speed, accuracy and user satisfaction.

3 State of the Art 3d Rotation Techniques

In this section, we analyze and compare different state of the art rotation techniques. The list of principles in Sect. 2 is utilized to formally classify these techniques and to point out existing drawbacks. Since this comparison focuses on usability concerns, details of the underlying mathematics and implementation are not covered but can be retrieved from the cited publications.

3.1 Chen et al.'s Virtual Trackball

The virtual trackball (VT) implemented by Chen et al. [6], the so-called Virtual Sphere, can be described as a 3d sphere located behind the viewport. The 2d viewport location of the moving mouse is projected onto this sphere to get corresponding 3d points on the sphere's surface. Thus, the 2d motion of the mouse is mapped to a 3d rotation from one projected point on the sphere's surface to a second one. (See Henriksen et al. [8] for a mathematical description of this and all other discussed techniques.) Unfortunately, the resulting rotation axis is not necessarily perpendicular to the mouse displacement vector [8]. The VT implemented by Bell [3] as well as Shoemake's VT [18] (both discussed further on) corrects this. Therefore, we will not further investigate Chen et al.'s VT.

3.2 Bell's Virtual Trackball

The VT implemented by Bell [3] is an improved version of Chen et al.'s VT [6] [8]. Instead of projecting the 2d mouse location onto a sphere, Bell projects it to a combination of a sphere and a hyperbola in 3d space. Thus, rotating a 3d scene using this approach appears very smooth. Nevertheless, with respect to our principles list Bell's VT fails in several points.

Moving the mouse by a vector $(\Delta x, \Delta y)$ at different locations on the viewport results in different rotations. As one example, a horizontal mouse movement in the center of the viewport rotates the 3d scene about the up-vector of the virtual camera. In contrast, the same movement at the bottom or top border of the viewport rotates the 3d scene about the viewing-axis of the virtual camera. Thus, even simple horizontal or vertical mouse movements may result in unpredictable rotations – Bell's VT violates Principle 1.

Even if similar actions do not provoke similar reactions and the rotation direction changes over the viewport space using this VT, the direction of the 3d rotation remains similar to the direction of the 2d mouse movement. For example, moving the mouse at the top border of the viewport to the right rotates the 3d scene clockwise and moving the mouse at the bottom border of the viewport to the right rotates the 3d scene counterclockwise. This behavior violates Principle 1 but fulfills Principle 2.

In contrast to Principle 3, a combination of two rotations using Bell's VT is not transitive. Moreover, the 3d scene starts to tumble if the mouse is moved on a circular trajectory. Thus, closed loops of mouse motion may not produce the expected closed loops of rotation (see Fig. 1). However, even in this case the

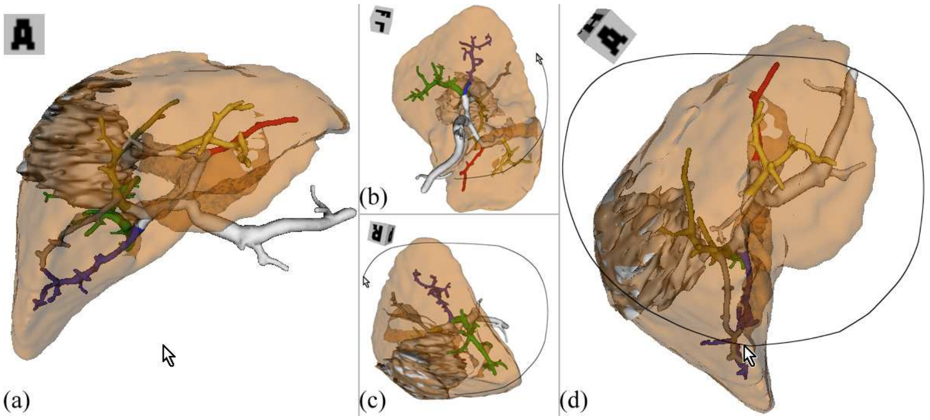

Fig. 1. Bell's VT is used to rotate a liver with tumor and vessels. (a)-(d) show different time steps. As illustrated, closed loops of mouse motion (black line) may not produce the expected closed loops of rotation (cp. (a), (d)).

rotation direction remains similar to the direction of mouse movement. As an example, while moving the mouse on a counter-clockwise circular trajectory, the object spins counter-clockwise too.

Due to the underlying mathematics of Bell's VT, it is not possible to customize the C/D ratio (Principle 4) in a convenient manner. Since using this technique the C/D ratio depends on the viewport size – changing the size on the screen (e.g. changing window size of the 3d view) results in a modified C/D ratio. As one result, the smaller the viewport the faster the 3d scene is rotated and vice versa. In addition, the rotation using Bell's VT is limited to 90° from the center of the viewport to its borders. As described before, this fixed rotation ratio results in a dependency between viewport size and C/D ratio. This limitation can be eliminated by introducing a scale factor for the calculated rotation angles. Unfortunately, this results in very different rotation behaviors as will be discussed in Sect. 3.3.

Despite these problems, Bell's VT is integrated in a wide range of applications especially due to its implementation in the Open Inventor graphics toolkit [17], [19].

3.3 Shoemake's Virtual Trackball: ArcBall

Shoemake's VT [18], the so-called Arcball, is a special version of Chen et al.'s VT (see [8]). Shoemake as well utilizes the projection of the mouse location onto a sphere to calculate rotation axis and angle. However, due to a carefully chosen scale factor for the calculated rotation angles using Shoemake's VT rotations are transitive. Thus, Shoemake's VT fulfills Principle 2 and 3 but similar to Bell's VT the Arcball violates Principle 1. However, in contrast to Bell's VT using

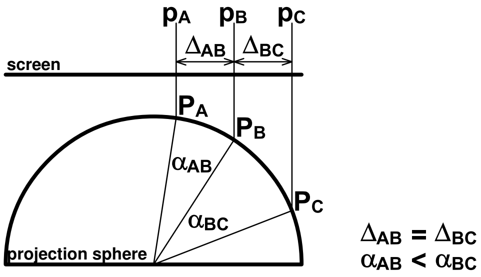

Fig. 2. Discontinuity of the resulting rotation angle of Shoemake's VT at the rim of the projected sphere: The distance Δ_{AB} between viewport points p_{A} and p_{B} equals the distance Δ_{BC} between p_{B} and p_{C}. However, the resulting rotation angle α_{AB} between projected points P_{A} and P_{B} is smaller than the angle α_{BC} between P_{B} and P_{C}.

Shoemake's Arcball the 3d scene is rotated twice about the viewing-axis if the mouse is moved once around the projection sphere, which may lead to confusion and slightly violates Principle 2.

Due to the underlying mathematics (similar to Bell's VT), Shoemake's VT does not provide a convenient way to customize the C/D ratio. The smaller the viewport the faster the 3d scene is rotated. Changing the size of the underlying projection sphere inversely proportional to the viewport size would fix this problem. However, this would either result in large viewport regions characterized by a rotation restricted to the viewing-axis of the camera or in a projection sphere much larger than the viewport hampering the 3d rotation about all axes. Similar to Bell's VT the rotation by means of Shoemake's VT is limited to 180° from the center of the viewport to its borders. Thus, the C/D ratio depends on the viewport size. As suggested for Bell's VT, this limitation can be eliminated by introducing a different scale factor for the calculated rotation angles. However, using a factor that halves the rotation speed of Shoemakes's VT will cause its behavior to adapt to Chen et al.'s and Bell's VT (see [8]) which violates Principle 3. Furthermore, due to the projection geometry of Shoemake's VT (a sphere) the C/D ratio changes from the center of the viewport to the rim of the projected sphere as illustrated in Fig. 2. As one result, the rotation seems to snap to the rim with a loss of user control and accuracy. This phenomenon has also been reported in [8] as a discontinuity of the rotation.

Even though the implementation of Shoemake's VT (see [10]) is simpler compared to Bell's VT, it is not wide-spread.

3.4 Two-Axis Valuator Trackball

In the Two-Axis Valuator approach (cp. [6]) the horizontal mouse movement is mapped to a rotation about the up-vector of the virtual camera and the vertical mouse movement is mapped to a rotation about the vector perpendicular to the up-vector and the view-vector of the camera. Diagonal mouse movement is mapped to a combined rotation about both axes.[4] As one result, no explicit rotation about the view-vector is possible and thus, it is often separately controllable by pressing another mouse button or an additional control key.

In contrast to all other discussed techniques, here the same mouse movement always results in the same rotation. Thus, the Two-Axis Valuator is the first (and only) approach that fulfills Principle 1. Furthermore, it fulfills Principle 2 since the direction of rotation always equals the direction of the mouse movement. In contrast to Principle 3, a combination of two rotations using the Two-Axis Valuator is not transitive. Furthermore, a clockwise circular mouse movement rotates the 3d scene counter-clockwise – Principle 2 is violated in this special case. Due to the direct mapping of the 2d mouse movement to a corresponding rotation about the two axes, the C/D ratio can be easily customized by changing the mapping factor. Thus, the Two-Axis Valuator fulfills Principle 4.

The Two-Axis Valuator is widely used in applications such as the Visualization Toolkit (VTK) [12]. This is probably due to the simple implementation. Another reason might be the fact that the Two-Axis Valuator fulfills three of the principles stated in Sect. 2 (with a slight deduction concerning Principle 2).

3.5 Two-Axis Valuator with Fixed Up-vector

For 3d modeling purposes (e.g. in 3D-Studio-Max [7]) or associated tasks (e.g. in Deep Exploration [16]) often a special version of the Two-Axis Valuator is applied. This version uses the world's up-vector for horizontal rotation. Since this vector is fixed, the approach is called Two-Axis Valuator with fixed up-vector.

Using a Two-Axis Valuator with fixed up-vector offers the advantage of transitive rotations – Principle 3 is fulfilled. Unfortunately, this leads to discrepancies between mouse movement direction and rotation direction – Thus, Principle 1 is violated. Furthermore, the rotation direction is contrary to the direction of mouse movement if the 3d scene is rotated upside down – Principle 2 is violated too. Nevertheless, Principle 4 is fulfilled due to the direct mouse movement mapping of the Two-Axis Valuator.

As mentioned before, the Two-Axis Valuator with fixed up-vector is used in 3d modeling applications. This is probably due to the transitivity of rotations and the customizable C/D ratio which are both crucial for the level of precision in those applications.

3.6 Tabular Comparison

As discussed in Sect. 3.1 – 3.5 each rotation technique exhibits different behavior and different features such as transitivity or a direct stimulus-response (S-R) compatibility and kinesthetic correspondence. None of the rotation techniques fulfills all principles from Sect. 2 (see Tab. 1).

---

[4] In a right-handed coordinate system with the negative z-axis being the view-vector the horizontal rotation is about the y-axis and the vertical rotation is about the x-axis of the camera.

Table 1. Comparison of state of the art rotation techniques.

Rotation Technique:	Bell's VT	Shoemake's VT	Two-Axis Valuator	Two-Axis Valuator with fixed up-vector
Principle 1	−	−	+	−
Principle 2	+	+	+/−	−
Principle 3	−	+	−	+
Principle 4	−	−	+	+

4 Evaluation of 3d Rotation Techniques

In Section 3 we systematically compared 3d rotation techniques with respect to a list of design principles which are inspired by clinical applications (see Sect. 2). As one result, we identified certain drawbacks of the discussed techniques and pointed out different usability crux of each of them. However, their influence on usability aspects such as performance, accuracy and user satisfaction is widely unknown.

As Henriksen et al. [8] stated, only four studies have empirically evaluated virtual trackballs ([6], [9], [15], [11]). Three of them (Chen et al. [6], Hinckley et al. [9] and Partala [15]) used relatively simple rotation tasks. In these "orientation matching tasks" the users were instructed to rotate an object to a target orientation presented by an image of this object in the desired orientation. Only Jacob and Oliver [11] stepped beyond and included a more complex inspection task (e.g. find the number of windows in a house). Since they found differences between rotation techniques even for different simple tasks, rotation techniques have to be empirically evaluated for more complex tasks to reveal trade-offs among them. Moreover, we notice that all these studies used only common, well-known objects with an inherent orientation (e.g. a house or a human head) which is strongly different from real world tasks for example in therapy planning and surgical education applications. Here, the shape or appearance of an object (e.g. a tumor) may not provide conclusive hints of the object's orientation.

In the following we present our empirical usability study of the discussed rotation techniques from Sect. 3 which focusses on the intuitiveness and appropriateness of these techniques for complex rotation tasks with complex-shaped objects.

4.1 Evaluation Goals

We attempt to compare different state of the art rotation techniques with focus on the degree of intuitive usage, performance and user satisfaction in conjunction with complex scan and hit tasks. This allows us to identify suitable rotation techniques for further usage, deeper evaluations and as a basis for future improvements.

4.2 Evaluation Strategy

Concerning therapy planning and surgical education applications, rotating a 3d scene is only a tool for solving various tasks. In this working environment it is crucial that the rotation techniques are comfortable and predictable. Therefore, our evaluation strategy focuses on the intuitive usage. No instructions, explanations and demonstrations of the usage and functions of the rotation techniques were presented to our subjects. The evaluation program could be downloaded from our project web-site [1] which provides further information and instructions on how to use this program as well.

Apparatus The evaluation was carried out by each user on his/her own computer system. Thus, users where not confronted with unfamiliar input devices and foreign device performance settings. For the evaluation we used a world-in-hand metaphor which let users feel to rotate the 3d scene and not the camera which had proven to be best suited for orbiting (cp. [15]). The evaluation program including questionnaire and all rotation techniques is implemented in Macromedia Director 8.5 [14].

Task In order to compare the rotation techniques, we identified the systematic exploration (scanning for a target) of a mostly unknown 3d scene in conjunction with rotating the target to the center of the view (e.g. for further inspections) as crucial. Thus, subjects were asked to perform a series of scan and hit (search and shoot) tasks.

Subjects were confronted with a complex-shaped symmetric and unknown object without an inherent orientation and had to scan this object for a target by rotating it (see Fig. 3). Once the target had been located they had to rotate it to the center of the screen and to shoot the target by pressing the space bar or the enter key (as they prefer). They were instructed that speed is important.

A defined level of accuracy was forced by the system such that the targets had to be rotated to a position inside a small circle on the screen. The allowed fault tolerance between the target center and the center of shooting had been 13°. The number of bullets required to shoot all targets was counted to quantify inaccuracy in solving the task.

After shooting a bullet the user received both visual and aural feedback indicating a hit (explosion of the target) or miss (impact on the rotated object, target remains in position to retry).

Design The evaluation was designed in the form of a game with different levels which had to be completed in random order. In each level, one of the rotation techniques from Sect. 3 is used to scan an object for 25 targets which had to be localized and shoot. Only one target at a time is randomly positioned in one of 18 funnels of the object that had to be scanned. After one target had been shoot another one appeared with a short delay.

Procedure At the beginning all subjects had to answer a questionnaire concerning their experience and familiarity with a mouse as input device, with 3d tools, 3d games and 3d rotation. Afterwards, subjects were given a general description of the experimental procedure.

Before entering a new level, a training scene was presented to become familiar with the rotation technique of the current level and to adjust the rotation speed to personal needs (by dragging a small slider) if the rotation technique provides this. At the beginning of each training session, subjects were instructed to click and drag with the mouse to rotate the scene and to press the space bar or enter key to shoot. The practice was neither limited in time nor in number of targets and could be terminated by the subjects whenever they felt familiar enough with the rotation technique. Subsequently, in the level subjects had to shoot 25 targets as fast as possible. To ensure reliable completion times, no adjustments of the rotation speed were allowed within a level.

After each level, subjects had to assess how comfortable they felt with hitting the targets and how close the rotation technique in this level followed their expectations. They were also informed about their time required to finish the level and a top ten of the fastest subjects in this level was presented. The same procedure was then repeated for the remaining levels (rotation techniques). The entire procedure took about 30 minutes.

Subjects Forty-two unpaid subjects (30 male, 12 female) took part in the evaluation. All of them had long-term experience with computer mice. On a scale from 1(low) to 7(high) all but four assessed their experience as 7 whereas these four subjects assessed their experience as 5 or 6, respectively. In contrast, all subjects had very different 3d navigation and rotation experience. To avoid a bias caused by different input device types only users which employed a mouse were included in the study.

4.3 Statistical Analysis

We first applied the One-Sample Kolmogorov-Smirnov-Test to test if the given data distributions are significantly different from a normal distribution. Task completion times exhibit normal distribution ($p >= .05$) whereas the number of shots required, experience and rotation technique assessment data differ significantly from a normal distribution. All subjects had to complete one level after another thus, we used paired samples tests to analyze differences between the rotation techniques.

To analyze differences in task completion time, we used the paired samples t-test. To analyze differences in the number of shots required, we applied the Wilcoxon-Signed-Ranks-Test and to analyze differences in the ordinal scaled assessment of the rotation techniques, the Sign-Test was employed.

4.4 Evaluation Results and Discussion

Statistics on the mean task completion times are shown in Tab. 2 and in Fig. 4, respectively. Comparison of completion times (Tab. 3) revealed that with high significance ($p <= .001$) users performed faster with Shoemake's VT and the Two-Axis Valuator than with the Two-Axis Valuator with fixed up-vector and Bell's VT. Furthermore, the Two-Axis Valuator was significantly ($p <= .05$) faster than Shoemake's VT. Between Bell's VT and the Two-Axis Valuator with fixed up-vector no significant difference could be observed.

Using Shoemake's VT and the Two-Axis Valuator subjects required significantly ($p <= .05$) more shots to complete the task compared to Bell's VT (Tab. 3, Fig. 5). However, users performed significantly faster. We did not observe any other significant correlations of shots required between the rotation techniques.

Statistics on the assessment of all interaction techniques concerning (1) comfortable task completion and (2) whether the behavior was perceived as predictable are shown in Fig. 6. As expected, the Two-Axis Valuator that fulfills most of the principles presented in Sect. 2 was assessed significantly ($p <= .01$) best with respect to both questions. In contrast, Bell's VT, Shoemake's VT and the Two-Axis Valuator with fixed up-vector were assessed very similar concerning question (1). Concerning question (2), Bell's VT and Shoemake's VT were assessed similar but significantly ($p <= .001$) superior to the Two-Axis Valuator with fixed up-vector.

A separate analysis for males and females revealed a slightly better performance for male subjects for any of the 3d rotation techniques. Using Bell's VT, females were significantly ($p <= .05$) slower. The survey also revealed that all females were less experienced in 3d navigation, 3d rotation, 3d modeling programs and 3d games. To verify if that explains the detected differences, we subdivided the subject pool into two groups according to their experience. A separate analysis of these two groups revealed no significant performance differences. Consequently, the significant difference in completion time using Bell's VT between males and females suggests that sex significantly influences user performance with Bell's VT.

To summarize, the Two-Axis Valuator as well as Shoemake's VT performed best even though using both techniques more shots had been required to hit all targets and to complete the task. Furthermore, subjects attested the Two-Axis Valuator to be the most convenient rotation technique which strongly supports our principles list.

5 Conclusion

We compared Bell's VT, Shoemake's VT, the Two-Axis Valuator and the Two-Axis Valuator with fixed up-vector by means of an experiment where we measured task completion times and shots required for complex scan and hit tasks. Our experiment design is inspired by clinical applications involving the complete

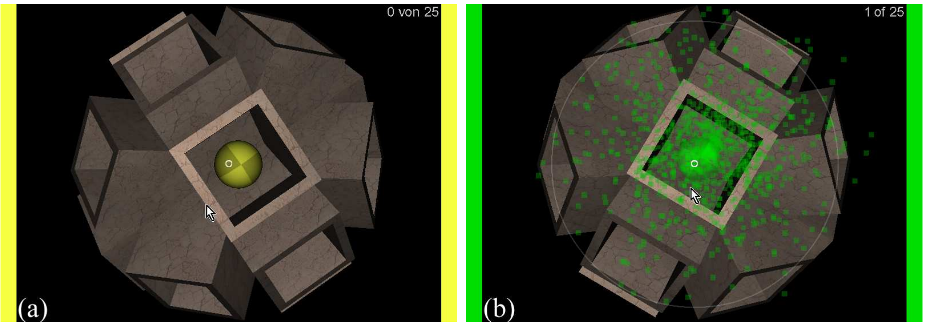

Fig. 3. Screenshots of the experiment software. Subjects rotated the 3d scene (e.g. by Bell's VT (a) and Shoemake's VT (b)) to locate the targets and to position them in the center of the viewport to shoot them. The viewport center is marked by a small white circle.

Table 2. Mean completion time for each rotation technique with separate means achieved by male (30), female (12), experienced (23) and inexperienced (19) subjects.

Rotation Technique	Mean Completion Time (in sec)				
	all subjects	males	females	experienced	inexperienced
Bell's VT	148.06	141.42	164.67	142.39	154.92
Shoemake's VT	133.24	130.37	140.42	130.75	136.26
Two-Axis Valuator	122.92	125.79	115.75	127.48	117.39
Two-Axis Valuator with fixed up-vector	147.71	148.59	145.50	146.54	149.12

Table 3. Comparison of rotation techniques regarding completion time and shots required.

Comparison			T-Test Completion Time		Wilcoxon-Test Shots Required	
Bell's VT	vs.	Shoemake's VT	t = -4.240	p = .000	Z = -2.083	p = .037
Bell's VT	vs.	Two-Axis Valuator	t = 5.360	p = .000	Z = -2.181	p = .029
Bell's VT	vs.	Two-Axis Valuator with fixed up-vector	t = 0.080	p = .937	Z = -0.796	p = .426
Shoemake's VT	vs.	Two-Axis Valuator	t = 2.231	p = .031	Z = -0.585	p = .558
Shoemake's VT	vs.	Two-Axis Valuator with fixed up-vector	t = -3.053	p = .004	Z = -0.766	p = .444
Two-Axis Valuator	vs.	Two-Axis Valuator with fixed up-vector	t = 7.153	p = .000	Z = -1.187	p = .235

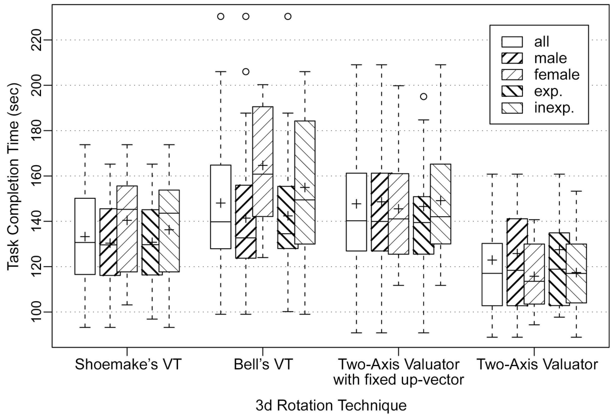

Fig. 4. Boxplot of completion times for all rotation techniques for all, male (30), female (12), experienced (23) and inexperienced (19) subjects.

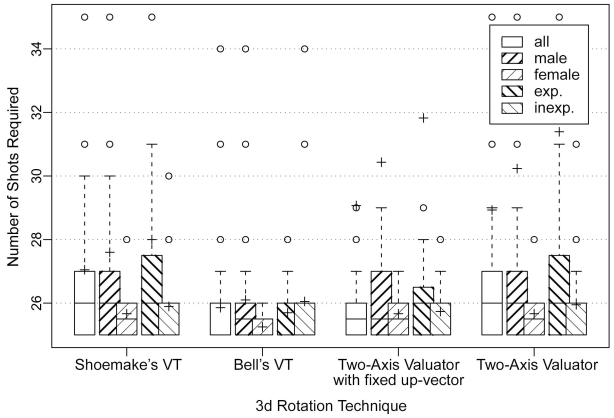

Fig. 5. Boxplot of number of shots required for all rotation techniques for all, male (30), female (12), experienced (23) and inexperienced (19) subjects.

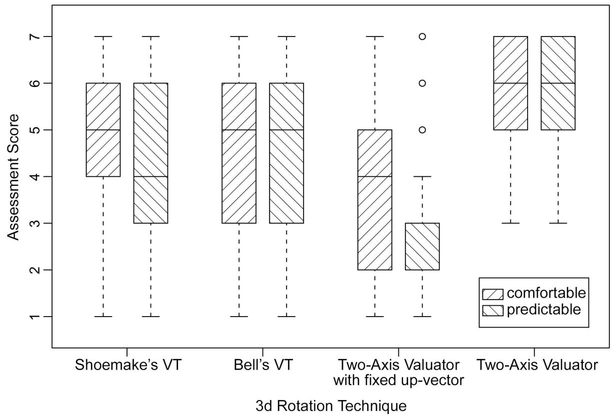

Fig. 6. Boxplot of assessment score for all rotation techniques.

exploration of arbitrary positioned and oriented complex shaped objects. A questionnaire was employed to assess whether 3d rotation techniques are perceived as comfortable and predictable.

Our evaluation revealed significant performance and user satisfaction differences between common 3d rotation techniques. The Two-Axis Valuator turned out to be the best 3d rotation technique while Bell's VT which is wide-spread in general 3d user interface toolkits is rated lowest with respect to speed and satisfaction. We are currently working on a refined version of interactive 3d rotation since we hypothesize that still better user performance is possible by an appropriate combination of flexibility and user-guidance.

References

1. Bade, R.: Evaluation of 3d rotation techniques, http://wwwisg.cs.uni-magdeburg.de/cv/projects/LST/rotation, April (2005)
2. Balakrishnan, R. Kurtenbach G.: Exploring bimanual camera control and object manipulation in 3D graphics interfaces. Proc. of ACM SIGCHI'99 (1999) 56–62
3. Bell, G.: Bell's Trackball. http://members.tripod.com/professor_tom/index.html, (1988)
4. Britton, E.G. Lipscomb, J.S. Pique, M.E.: Making nested rotations convenient for the user. Proc. of SIGGRAPH'78 (1978) 222–227
5. Chapanis, A. Kinkade, R.: Design of Controls. H. vanCott, R. Kinkade (ed.) Human Engineering Guide to Equipment Design, US Government Printing Office (1972)
6. Chen, M. Mountford, S.J. Sellen, A.: A study in interactive 3D rotation using 2D control devices. Computer Graphics, **22(4)** (1988) 121–129
7. Discreet: Discreet 3D-Studio-Max. http://www.discreet.com/3dsmax, April (2005)
8. Henriksen, K. Sporring, J. Hornbæk, K.: Virtual trackballs revisited. IEEE Transactions on Visualization and Computer Graphics, **10(2)** (2004) 206–216
9. Hinckley, K. Tullio, J. Pausch, R. Proffitt, D. Kassel, N.: Usability analysis of 3D rotation techniques. Proc. of ACM UIST'97 (1997) 1–10
10. Hultquist, J.: A Virtual Trackball. Graphics Gems, Academic Press (1990) 462–463
11. Jacob, I. Oliver, J.: Evaluation of Techniques for Specifying 3D Rotations with 2D Input Device. Proc. of HCI'95 (1995) 63–76
12. Kitware Inc: VTK Home Page. http://www.vtk.org, April (2005)
13. Krueger, A. Tietjen, C. Hintze, J. Preim, B. Hertel, I. Strauss, G.: Interactive Visualization for Neck Dissection Planning. Proc. of IEEE EuroVis'05, (2005) 295–302
14. Macromedia: Macromedia Director.
    http://www.macromedia.com/software/director, April (2005)
15. Partala, T.: Controlling a Single 3D Object: Viewpoint Metaphors, Speed and Subjective Satisfaction. Proc. of INTERACT'99, IOS Press (1999) 486-493
16. Right Hemisphere: Deep Exploration.
    http://www.righthemisphere.com/products/dexp, April (2005)
17. SGI: SGI - Developer Central Open Source | Open Inventor.
    http://oss.sgi.com/projects/inventor, April (2005)
18. Shoemake, K.: ARCBALL: A user interface for specifying three-dimensional orientation using a mouse. Proc. of Graphics Interface (1992) 151–156
19. Strauss, P.S. Carey, R.: An object-oriented 3D graphics toolkit. Proc. of SIGGRAPH'92 (1992) 341–349