A Study in Interactive 3-D Rotation Using 2–D Control Devices

Michael Chen
Department of Electrical Engineering/
Dynamic Graphics Project
University of Toronto[†]

S. Joy Mountford
Human Interface Group
Apple® Computer Inc.[‡]

Abigail Sellen
Institute for Cognitive Science
University of California, San Diego[#]

ABSTRACT

This paper describes and evaluates the design of four virtual controllers for use in rotating three-dimensional objects using the mouse. Three of four of these controllers are "new" in that they extend traditional direct manipulation techniques to a 3-D environment. User performance is compared during simple and complex rotation tasks. The results indicate faster performance for complex rotations using the new continuous axes controllers compared to more traditional slider approaches. No significant differences in accuracy for complex rotations were found across the virtual controllers.

A second study compared the best of these four virtual controllers (the Virtual Sphere) to a control device by Evans, Tanner and Wein. No significant differences either in time to complete rotation task or accuracy of performance were found. All but one subject indicated they preferred the Virtual Sphere because it seemed more "natural".

CR Categories and Subject Descriptors: I.3.6 [Computer Graphics]: Methodology and Techniques – interaction techniques; D.2.2 [Software Engineering]: Tools and Techniques – User interfaces; H.1.2 [Models and Principles]: User/Machine Systems – Human factors; B.4.2 [Input/Output devices]; J.6 [Computer-Aided Engineering]: Computer-aided manufacturing.

General Terms: Algorithms, Experimentation, Performance, Human Factors.

Additional Key Words and Phrases: input devices, virtual controllers, I/O devices, virtual sphere, mouse, interactive graphics, 3-D graphics, real-time graphics, rotation control, user performance.

1. INTRODUCTION

The recent increase in the available power of special purpose computer graphics machines has extended the operational range of capabilities for users. Objects can now be more easily generated in 3-D (in wireframe, solid and shaded forms), and manipulated in real-time. Despite advances in the ability to display 3-D objects, there is a lack of methods by which the user can easily manipulate and control the position of an object on the screen. Currently, simple direct manipulation controllers do not exist for 3-D object positioning. The design of such controllers could be important interface contributions for application environments such as manufacturing, architecture, and engineering design, which rely heavily on the display and control of three dimensions. The mouse is a successful interface tool, performing well for direct manipulation control of two-axis problems, either through manipulation of x and y separately, or the coupled control of x and y axes together. However, the issue of how best to extend the use of the mouse to accommodate the additional capabilities afforded by three-dimensional graphics is still relatively unexplored.

The ultimate goal is to provide users with an easy way of performing translation, rotation and sizing operations for complete manipulation of 3-D objects. This current performance study focuses on the use of virtual controllers in conjunction with a mouse to perform tasks involving rotation. In performing rotations users can manipulate all three axes simultaneously, whereas in performing translations and sizing operations users more often use fewer axes.

Most 3-D graphics machines use a mouse with one to three discrete buttons as the main input control device. Currently, there are four popular display techniques used to control object rotations:

1) Sliders: Typically the user adjusts the x, y and z sliders graphically displayed on the screen to indicate the amount of rotation in each axis independently. (Alternatively, physical sliders can be used).
2) Menu selection: The user first selects the axis from a text menu and then holds down the mouse button while moving the mouse in one dimension to indicate the amount of rotation.
3) Button press: The user holds down one of three buttons on the mouse or keyboard, and moves the mouse in one dimension to indicate the amount of rotation.
4) Two-axes valuator: The user moves the mouse in two dimensions to control rotation in two of the three axes.

The first three conventional approaches do allow access to rotation on all three axes but use the mouse as a one-dimensional input device. For example, the same left-and-right motion is used to control different rotation directions. However, there is little stimulus-response (S-R) compatibility or kinesthetic correspondence between the direction of mouse movement and direction of object rotation [7] Pique, 1986. The fourth conventional technique, (the two-axis valuator), does provide better S-R correspondence.

[†] Department of Electrical Engineering, University of Toronto, Toronto, Ontario, Canada, M5S 1A4. Tel # (416) 978-6619.

[‡] Apple Computer Inc., 20525 Mariani Ave MS 27A0, Cupertino, CA, 95014. Tel # (408) 973-4801.

[#] Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, CA, 92093. Tel # (619) 534-2541.

©1988 ACM-0-89791-275-6/88/008/0121 $00.75

The amount of left-and-right and up-and-down movement of the mouse can proportionally rotate the object left-and-right and up-and-down on screen. Rotation about an arbitrary axis on a plane can also be done by moving the mouse diagonally. However, this technique does not allow the user to rotate the object clockwise or counter-clockwise. Therefore systems that use this technique often require the user to work with 3 independent orthogonal views to execute complete 3-D manipulations.

One possible solution to permit full object manipulation is to use input devices with additional degrees of freedom. However, few people seem able to construct reliable mental models about the relative contributions and effects of all the coupled axes which are associated with these extra degrees of freedom. An earlier study described by [5] Mountford, Spires and Korner, 1986, showed how much time subjects spent using all the different axes involved in 3-D control (i.e. the single axes x, y, z; the coupled axes xy, yz, xz; and all axes, xyz attached/coupled together). In this study, subjects performed translation, rotation and sizing operations during an object construction task. The results indicated that during rotation operations subjects used mostly the single independent axis, x, y or z; during translation mostly the coupled xy axis; and during sizing, all three axes together, xyz. Very few subjects in this study used (or had use for) coupled axes of control, except for the familiar xy coupled axis. Subjects did not use either of the other pairs of axes (xz, yz) to move, rotate or size objects.

This performance evaluation study suggests that users did not have enough familiarity or experience with coupled axes (ie xz, yz, or xyz) to successfully perform fully integrated 3-D control manipulations using all the different combinations of axes. Users are particularly unfamiliar with the visual appearance and movement associated with rotating an object around xz or yz. If this is indeed the case, then it is unlikely that users will want to have new devices that make simultaneous use of **all** of the additional degrees of freedom that can be provided for 3-D object manipulation. It is possible that for more complex manipulation tasks such as docking, a device with some extra degrees of freedom may be appropriate. A full six degrees-of-freedom controller called the IIISPACE™ Digitizer (Polhemus) is available, but such input devices are not yet affordable for most users. Traditional 2-D input devices will continue to be the most available and dominant devices. Thus it is important to design 3-D manipulation techniques assuming such a 2-D device.

The current paper describes the conventional slider approach as well as three alternate "virtual rotational controllers" that allow users to directly manipulate 3-D objects using a one-button mouse. These controllers were designed not to have any knobs, drag boxes or menus that could distract the user from the task of rotating the object. Furthermore, each controller was designed to be overlaid on top of the object to be rotated, helping the user focus attention on the object being manipulated. This suggested another constraint, that the controllers be as transparent as possible for a clear view of the object. Finally, the intention was that the controllers be easily understood by novices and be as natural to use as possible. That is, the goal was to make them "transparent" and easy to use. We designed the controller operations to perform as analogously to real object manipulations as possible. This was achieved by extending the use of successful 2-D direct manipulation techniques to a 3-D environment.

This paper also describes two studies which were carried out to evaluate the controllers by comparing subjects' performance in rotating object in 3-D. The first study compares relative performance of all four controllers, the traditional slider and the 'new' three virtual controllers developed by Chen. The second study compares the best of these controllers to a controller developed by [4] Evans, Tanner and Wein, 1981.

2. DESCRIPTION OF VIRTUAL ROTATIONAL CONTROLLERS

Figure 1 shows a representation of the displayed house used in all rotation tasks. Rotations in x, y and z correspond to rotating the object up-and-down, left-and-right and clockwise-counter-clockwise, respectively. Thus, in this study, rotation is with respect to the user's (camera's) frame of reference. Even though there are systems that perform rotations about the object's frame, (e.g. [1] Bier, 1986, [6] Nielson and Olsen, 1986], it has been suggested that inexperienced users can perform rotations more easily in the user's reference frame [5] Mountford et al, 1986.

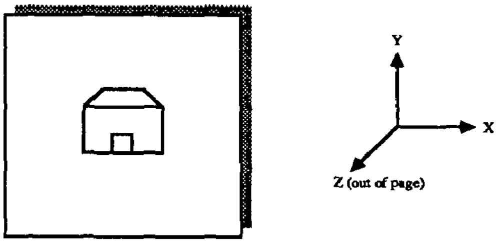
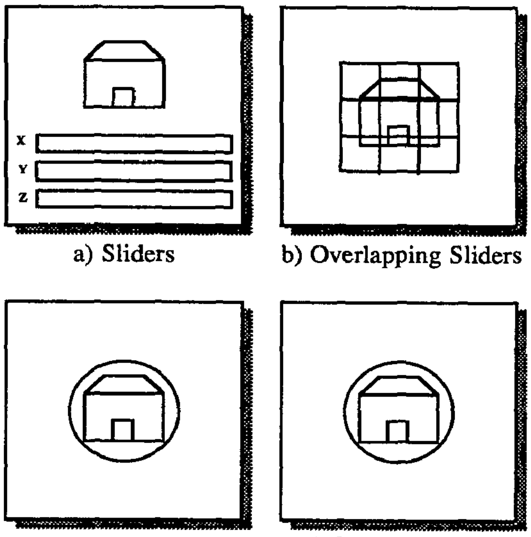


Figure 1. Definition of the coordinate axes.

The four controller displays used in the evaluation test are shown in Figure 2. Note that the Continuous XY with additional Z and the Virtual Sphere controllers have the same displays. They differ in the rotation axes available inside the circular region (described later).



a) Sliders

b) Overlapping Sliders

c) Continuous XY+ Z

d) Virtual Sphere

Figure 2. Screen displays of the four virtual controllers with object in centre.

2.1. Graphical Sliders Controller

The Graphical Sliders controller uses a traditional approach to allow users to perform 3-D rotations and serves as a control for performance comparisons. In this study, we chose horizontal sliders and placed them below the object to be rotated (see Figure 2a), similar to other graphical control interfaces. The sliders simulate "treadmills" and therefore provide relative control over the amount of rotation. A full sweep across a slider provides 180 degrees of rotation about an independent axis. As long as the mouse button is initially depressed inside one slider, the user can rotate about the corresponding axis even if accidentally crossing into another slider.

2.2. Overlapping Sliders Controller

The Overlapping Sliders controller [3] Chen, 1987, is a modification of the conventional slider approach in three respects:

1) The x, y, and z axes are represented by a vertical, horizontal and circular slider, respectively.

2) All three sliders are overlapped (as shown in Figure-3a) and then simplified to look like a nine-square grid (Figure 3b).

3) The grid is superimposed over the object to be rotated (Figure 2b)

In this implementation, a full sweep of the vertical or horizontal slider rotates the object 180 degrees about the x or y axis respectively. A full circle around the outside squares rotates the object 360 degrees about z (see Figure 3b). Note that only near vertical, horizontal and circular movement of the mouse inside the middle column, middle row and outside squares (respectively) are recognized by this controller. A diagonal movement in the middle square, for example, is ignored since this is a coupled rotation in x and y (i.e. the rotation axis lying somewhere on the x-y plane). Thus, this controller still operates on the basis of single axis control. The difference between this controller and conventional sliders, though, is increased controller-display compatibility. The direction of movement of the mouse more closely corresponds with the direction of rotation. In addition, superimposing the controller on the object is intended to give the user more of a sense of directly manipulating the object.

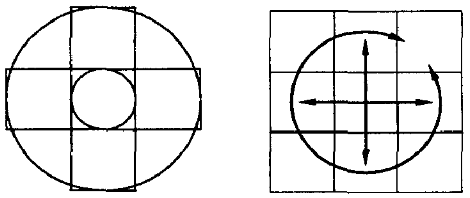


Figure 3. a) Three overlapped sliders, b) idealized version

2.3. Continuous XY with Additional Z Controller

The Continuous XY with added Z controller (Figure 2c) operates in two modes. If the mouse button is depressed while the mouse cursor is inside the circle, left-and-right and up-and-down movement of the mouse will rotate the object left-and right and up-and-down on the screen. Diagonal movement will rotate the object the proportional amount about the x-axis and y-axis (i.e. the axis of rotation is on the x-y plane and is perpendicular to the direction of mouse movement). If the mouse button is depressed while the mouse cursor is outside the circle, the user can rotate the whole object clockwise by going around the outside of the circle. Thus, this controller provides either 1) continuous rotation on the x-y plane, or 2) exact rotation about the z-axis. In this implementation, a full sweep of the mouse across the circle rotates the object 180 degrees about the corresponding axis in the x-y plane. A full circle around the outside rotates the object 360 degrees about z.

2.4. Virtual Sphere Controller

The virtual sphere controller simulates the mechanics of a physical 3-D trackball that can freely rotate about any arbitrary axis in 3-space. On the display screen (see Figure 2d), the user can imagine viewing an object encased in a glass sphere. Rotation is then a matter of rolling the sphere and therefore the object with the mouse cursor. Up-and-down and left-and-right movement at the centre of the circle is equivalent to "rolling" the imaginary sphere at its apex and produces rotation about the x-axis and y-axis respectively. Movement along (or completely outside) the edge of the circle is equivalent to rolling the sphere at the edge and produces rotation about z. The amount of rotation is adjusted so that a full sweep of the mouse across the circle rotates the object 180 degrees about the corresponding axis in the x-y plane; a full circle around the outside rotates the object 360 degrees about z. The implementation of the Virtual Sphere is outlined in Appendix A.

The difference between this and the Continuous XY with additional Z, is that the Virtual Sphere allows continuous rotation about all three axes inside the circle[1] while the latter only allows continuous control of two axes inside. To rotate in z, the user must go outside the circle.

3. EXPERIMENT 1

This first experiment was designed to compare the subject performance using the four controllers described above. The main performance measures recorded were time to complete rotation task and accuracy in performing that task. The experimentor gave minimal instruction in the use of each controller, so that no explicit conceptual model was imparted to the subjects. For example, the subjects were not told that the Virtual Sphere controller simulated a physical 3-D trackball.

The previously described four controllers were presented to subjects in order of increasing computational and cognitive complexity. It may be reasonable to assume that users would have more difficulty in grasping the idea behind the latter controllers. We were especially interested in how novices would perform without first being told the conceptual models of the controllers. We wanted to find out how easy the controllers were to learn by allowing subjects to just start trying to use them.

3.1. Method

3.1.1. Subjects

Twelve right-handed, male subjects were tested, consisting of both undergraduate and graduate students at the University of Toronto. All were familiar with using a mouse while none had any experience with any of the four controllers. Only three of the twelve had any experience with 3-D graphics systems.

3.1.2. Apparatus

The experiment was run entirely on an Silicon Graphics IRIS 3020 workstation. The IRIS (Integrated Raster Imaging System) is a high-performance, high-resolution (1024 by 768) colour computing system for 2-D and 3-D graphics. The heart of the IRIS is a custom VLSI chip called the Geometry Engine. A pipeline of ten or twelve Geometry Engines accepts points, vectors, polygons, characters and curves in user-defined coordinate systems and transforms them to screen coordinates, with rotations, translation, scaling and clipping. The four virtual controllers, the solid rendered house and the testing programs were written in C.

In addition to the Geometry Pipeline, an IRIS system consists of a general-purpose microprocessor, a raster sub-system, a high-resolution colour monitor, a keyboard and a three-button optical mouse. Only the left button of the mouse was used for these controllers and the mouse worked best using stroke-lift-stroke tactics. The mouse acceleration algorithm was disabled so that the amount of cursor movement was not affected by the speed of the mouse movement. An IRIS was used because it is a very fast machine and runs in real-time and can provide full colour rendering of solid objects.

[1] The Virtual Sphere controller may actually be better than a real physical 3-D trackball in at least one respect. With a physical trackball, it is impossible to have the entire top hemisphere of the ball exposed. This is because one of the rotation sensors must be placed at the "equator" of the sphere. Thus it is nearly impossible for the user to physically twist the trackball while rolling it. Accordingly, a 3-D trackball is better described as a 2+1D device (Buxton, 1986).

3.1.3. Task

In order to compare user performance on all four virtual controllers, subjects were asked to perform a series of matching tasks. Subjects were shown a solid-rendered, upright house in colour on the right-hand side of the screen and were asked to match its orientation to a tilted house on the left-hand side of the screen. The house was coloured differently on all of its faces so as to aid the subject in identifying its various surfaces. The centre of rotation was fixed at the centre of gravity of the house. Subjects were told to press the space bar when satisfied with the match, and were instructed that both speed and accuracy were important. Both task completion time and accuracy were recorded on-line.

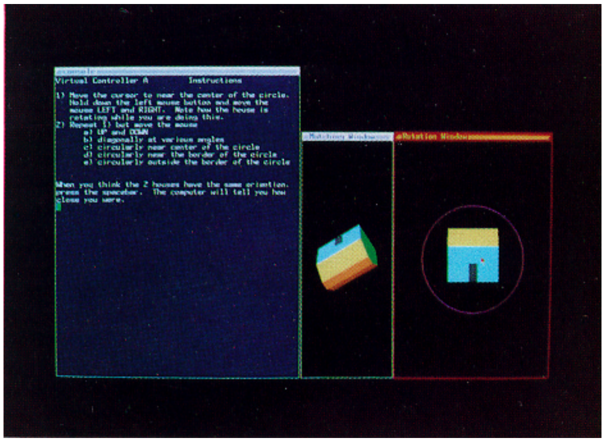
After pressing the spacebar to indicate a match, subjects were given feedback on the accuracy of the match for each trial. This feedback was provided to subjects to illustrate the desired quality of exactness in house positioning to help subjects achieve optimal performance. Accuracy was obtained by comparing the 3x3 rotation matrices of the two houses. The accuracy measure was calculated as the sum of the differences between the corresponding elements in the rotation matrices squared. From the subject's perspective, accuracy was rated as "Excellent****" (squared error from 0 to 0.02), "Good Match***" (squared error of 0.02 to 0.035), or "Not good enough, try harder next time**" (squared error greater than 0.035). The squared error of 0.02 and 0.035 corresponds to a rotation mismatch of 5.7 and 7.6 degrees respectively.

3.1.4. Design

Each subject performed using all four controllers, using a within subject design. Order of controllers was counterbalanced according to a Latin-square design. For each controller, there were nine different non-upright house positions to be matched. Three of the nine orientations required only simple rotations about the x-, y- or z-axis. The other six orientations were more complex, requiring coupled axes of rotation using the full range of axes manipulation. Each orientation was presented three times for a total of 27 trials per controller. Orientations were presented randomly and sampled without replacement, with the constraint that simple orientations were presented first, followed by complex ones.

3.1.5. Procedure

All instructions for the experiment were provided on-line. At the start of the session, subjects were given a general description of the experimental procedure. Specific instructions for using the first controller were then presented, followed by three minutes of practice. During these three minutes, subjects could attempt to match as many orientations as possible, and performance feedback was provided. Figure 4 shows a photograph of the actual experimental screen with instructions on the left, the house orientation to be matched in the middle and the house to be rotated in the right window. Each subject was then given two practice trials (not timed) and then 27 timed trials consisting of 9 different orientations each repeated three times. At the end of each of the block of 27 trials, the subject was given a break. The same procedure was then repeated for the remaining three controllers. The entire experimental session lasted approximately 1 1/2 hours.



```
Virtual Controller A          Instructions

1) Move the cursor to near the center of the circle.
   Hold down the left mouse button and move the
   mouse LEFT and RIGHT.  Note how the house is
   rotating while you are doing this.
2) Repeat 1) but move the mouse
      a) UP and DOWN
      b) diagonally at various angles
      c) circularly near center of the circle
      d) circularly near the border of the circle
      e) circularly outside the border of the circle


When you think the 2 houses have the same orientaion.
press the spacebar.  The computer will tell you how
close you were.
```

Figure 4. Photo of actual experimental screen of the IRIS. Instructions for each controller are presented on the left, the house orientation to be matched in the middle window, and the house to be rotated in the right window.

3.2. Results and Discussion

Figure 5 shows the average time and standard deviations in seconds to complete rotations for simple versus complex orientations collapsed across all subjects. The results show an interesting interaction between type of controller and complexity of the matching task. In performing simple, single-axis tasks, the conventional slider and the overlapping sliders produced significantly faster performance ($p<0.001$). However for complex rotations, the Continuous XY with additional Z, and Virtual Sphere controllers were clearly faster ($p<0.001$). The variance in speed of performance remained relatively constant across controllers for both simple and complex tasks, larger for complex rotations and smaller for simple rotations.

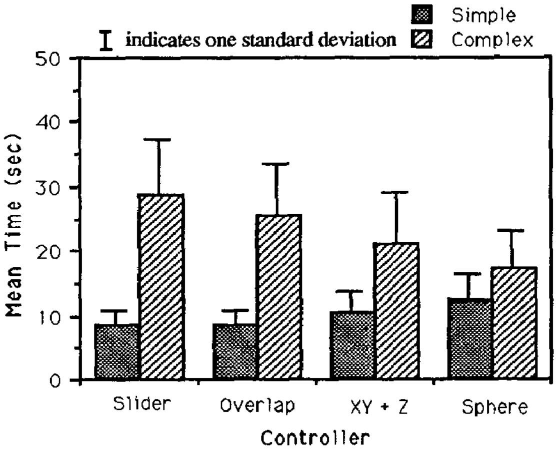


Figure 5. Mean time to complete simple and complex rotations.

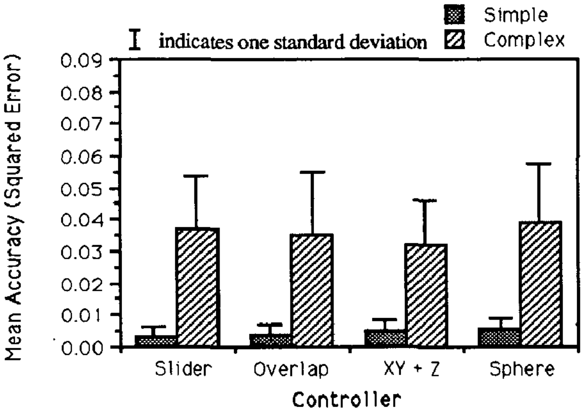


Figure 6. Mean accuracy for simple and complex rotations.

As a result of observing subjects performing single-axis rotation with the slider controllers, it was clear that when subjects selected the correct slider, the time to complete the match was short. However, subjects would often begin by selecting the wrong slider and then spend their time correcting the error. However, for the continuous controllers (XY + Z and the Virtual Sphere), initial movement was almost always in the correct direction, but the extra degrees of freedom made single axis rotation more difficult, so more time was needed to compensate for small deviations from rotation about that axis. This suggests that allowing the user to work with independent axes of control may be best when precise rotation is required around one axis. The real word situations in which such rotations may be required, however, seem limited.

When subjects performed complex rotations, the Virtual Sphere was clearly superior in terms of speed. On the basis of these data, we can expect an average savings of almost twelve seconds for a single, complex rotation task by using the Virtual Sphere as compared to conventional slider controllers. Furthermore, most subjects commented that they preferred the Virtual Sphere of the four controllers that they used, while two subjects preferred the Continuous XY with Additional Z controller. Subjects remarked that the Virtual Sphere seemed "more natural" and that they felt like they were actually rotating the object directly, rather than manipulating a controller which in turn rotated the object. It seems that the use of continuous control is one important aspect in the design of virtual controllers for this kind of task.

A further point of interest is that the overlapping sliders, while not producing performance as fast as the continuous controllers, did give a shorter mean task completion time than the traditional slider approach. This performance difference is probably due to the increased S-R compatibility of this controller versus the traditional slider controller.

Subjects performing simple rotations were significantly less accurate using the continuous controllers compared to the two slider controllers ($p<0.05$). These results are shown in Figure 6. However, the actual magnitude of these differences was small (at most a squared deviation of 0.003) . There were no significant differences in accuracy for the complex rotations. Again, variances across controllers were fairly constant, both simple and complex rotations indicated the same trends.

The data suggest that if the task to be performed is extremely simple, and if it is important that the rotation be accurate, then sliders may be most suitable. However, given any increase in the complexity of the task, controllers designed based on the principles of direct manipulation produce faster and just as accurate performance.

4. EXPERIMENT 2

In experiment 1 the Virtual Sphere produced the best user performance of the four controllers in complex rotations. It seemed of interest to know how the Virtual Controllers would perform relative to a similar controller developed by Evans et al. [4]. This further experiment was prompted by some experts in the area claiming that the two controllers were very similar. However, it was our opinion that several differences existed between these two controllers, both in terms of technical implementation and in visual presentation style.

Technically, the Evans et al. technique is a combination of the "two-axis trackball" and the "stirrer" techniques described in their paper. Their implementation recognizes straight line (continuous rotation in x and y) and circular (rotation in z) gestures. To detect the different motions, a "stirring angle" is calculated based on the change in movement of the last three positions of the input device. This value is then compared to a threshold to decide whether the movement is in a "relatively" straight-line or not. Unfortunately, the threshold is dependent on two interrelated variables: the speed with which each individual user likes to draw the circle and the frequency of taking a reading from the input device. If the sampling rate is too fast or the user prefers to draw the circle slowly, the three readings would tend to indicate that a straight line is drawn. Thus, threshold adjustments may be needed for different systems and different users with this technique[1]. The Virtual Sphere, on the other hand, allows rotation about an arbitrary axis in 3-space. The direction and amount of rotation is based only on the last two locations of the input device, and no user dependent adjustment is necessary.

The two techniques also have different visual presentations. With the Evans et al. technique, the location of the cursor which is controlled by the input device is not important; the user can ignore the cursor and just concentrate on the object being rotated. With the Virtual Sphere, the cursor must stay inside the "circle" to control rotation about all three axes. This technique works best when the circle is surrounding the object being rotated, so as to take advantage of the direct manipulation quality that the controller affords. With respect to the cursor, the Evans et al device is a relative controller whereas the Virtual Sphere is an absolute controller. Our Virtual Controller provides the user with some additional visual guidance as to where to concentrate their manipulation movements.

To implement the Evans et al. technique, we invited one of the co-authors, Peter Tanner, to help us reproduce the "feel" of their original implementation. The following adjustments were made to deal with the sampling problem mentioned above:

- Cursor movement is only recognized if the change is greater than a 3 pixels radius.
- The largest stirring angle (rotation in z) is limited to approximately 33 degrees per screen update.
- The stirring angle is scaled proportional to the amount of cursor movement.

The stirring threshold was set to approximately 13 degrees so that an angular change in movement of less than 13 degrees is considered movement along a straight line.

Quantatively, a 360 degrees of rotation of an object requires about 3200 degrees of quick small circular motion or 1100 degrees of quick large circular motion. For the same rotation in x-y, the implementation required about 2.5 times the movement distance as the Virtual Sphere controller.

[1] The Evans et al paper suggested that it is possible to perform rotations in x, y and z together, by reducing the x and y rotations when the stirring motion is large, and reducing z rotation when stirring motion is small. However, Tanner informed us [personal communication] that their implementation *did* use an angular threshold to decide whether to perform rotation in x-y or in z.

4.1. Method

The method for this experiment was identical to that used in the previous experiment 1, with the following exceptions:

- Six different, right-handed, male subjects were used instead of twelve. Again, all were familiar with the mouse while only two of the six had used any 3-D computer graphics systems.
- Only two controllers were used in this experiment. Half the subject used the Virtual Sphere first, while the other half used the Evans et al. controller first.
- The entire session lasted about 45 minutes.
- An IRIS 2400 Turbo with a mechanical mouse was used in this experiment. The mechanical mouse provided about the same controller-display ratio as the optical mouse used in experiment 1.

4.2. Results and Discussion

Figure 7 shows the average time in seconds to complete rotations for simple versus complex orientations collapsed across subjects. Figure 8 shows the mean accuracy scores for both simple and complex rotations. The results under all conditions show the Virtual Sphere and the Evans et al. technique to be similar. Statistical tests showed there were no significant differences between the two controllers at the 0.05 level.

Note that Figure 7 and 8 also show the result for the Virtual Sphere from experiment 1. Some performance variations between the experiments using the same controller are to be expected, see Figure 7, and these differences are relatively small. However, Figure 8 shows noticeably different standard deviations for the Virtual Sphere between the two experiments, larger in the second than the first. This may be a result of using different subjects who used only two controllers in experiment 2, compared with four in experiment 1, or because of using two different Iris machines with two different types of mice.

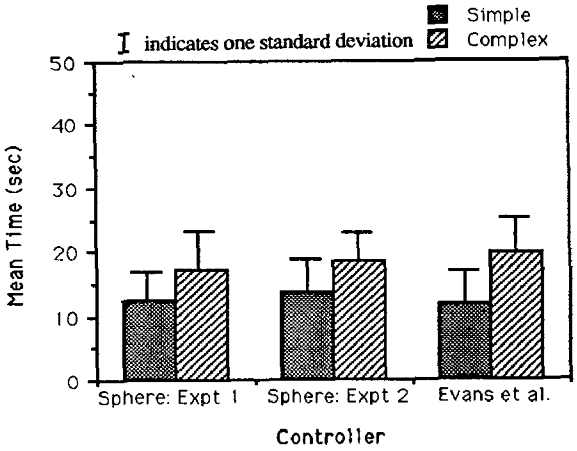


Figure 7. Mean time to complete simple and complex rotations.

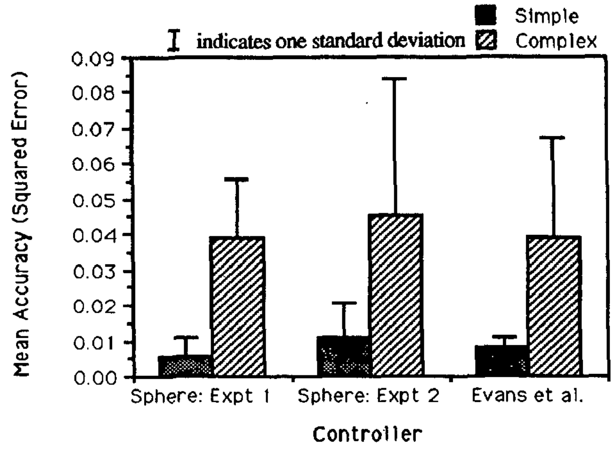


Figure 8. Mean accuracy for simple and complex rotations.

Comments from the subjects indicated that the majority (5 out of 6) preferred the Virtual Sphere over the Evans et al. controller. They commented that the Virtual Sphere felt more "natural", even though only two subjects were explicit about comparing the controller to manipulating a sphere. The one subject that preferred the Evans et al. controller indicated he liked it because he did not have to watch the cursor, only the object being manipulated. However, all the subjects said that they had difficulty in making fine rotation in z, since this required quick but short circular motions. Also a large rotation in z requires a lot of circular motion since the controller has a built-in maximum rotation speed. Large circles were also said to be less effective because often the stirring threshold was not reached, and resulted in x-y rotations.

5. CONCLUSIONS

The data reported in the first study support the use of continuous-axes controllers for complex multi-axis object manipulations. Observation of the subjects confirmed that moving between axes is cumbersome with the sliders, since there is no inherent direct manipulation capability. However, the slider controllers are just as good for simple single axis rotation, where the axes are already constrained to only one axis movement at a time, a situation which simplifies the user's control options. This would indicate that some constraint mechanism should be provided to limit the axis of rotation for more continuous-type controllers, if they are to be used in a real system.

In both of our experiments, the new controllers have a "one-to-one" controller to display (C-D) ratio. This created the impression that when using the Virtual Sphere controller, subjects thought they could actually grab the corners of the house and move it. A smaller C-D ratio might have made fine adjustments easier. However, subjects had more difficulty in judging orientation accurately than in performing fine mouse movements. Nevertheless, it would be useful to test the effects of different C-D ratios or dynamic ratios which would vary with the speed of motion. It might also be worthwhile to re-test our subjects to examine any performance changes now they know the conceptual model behind the controllers.

The fact that the results in experiment 2 showed no significant difference between Evans et al.'s technique and the Virtual Sphere in itself *is* significant. It would be tempting to regard these controllers as competitors, where one controller could be chosen and then used by all users. However as we mentioned before, these two techniques differ both in implementation and more importantly, in their visual presentation. Note also that both techniques deal with only one aspect of rotation manipulation. We have ignored the processes involved with the user having to actually select the object as well as selecting it's centre of rotation. Designing these additional performance features and integrating them successfully into the entire interface design is an important and critical next step. The results indicate that either of the techniques would perform better relative to other existing techniques for 3-D interface rotations. The ultimate decision should be based on user's preference, and on which technique fits in better with the entire interface design for the broadest range of different user tasks.

The Evans et al. paper presented a catalogue of interesting techniques, but gave no supporting behavioural data comparing these techniques, or to other existing techniques that were common at the time. While some of the techniques described were novel and appeared to be fairly powerful, they are not in common usage. For example, we are not aware of any commercial system that makes use of their technique that we replicated in this experiment. One of the driving forces in the current work was not just to introduce another interaction technique, but to provide objective comparative user data which allows the reader to quantatively access the relative strengths and weaknesses of the different controllers.

We plan to further develop these techniques and to explore their use in a range of more complex tasks. More complex and diverse tasks may further indicate where the advantages for different controllers exist. Furthermore, the "complex" rotations subjects performed in these studies may be viewed as relatively simple when compared to the kinds of tasks that may be required in real-world settings. Users need to rotate objects in the context of other objects as well as to perform translation and sizing operations in the 3-D graphics environment.

Our virtual, alternative continuous-axes controllers did not require the use of special purpose 3-D control devices, nor did they require the use of a multi-button mouse. The performance value of the continuous controllers, (Continuous XY with additional Z and the Virtual Sphere), lies both in their intuitive easy-to-learn features and their direct manipulation capabilities. These controllers are worthy of further experimental validation and refinement for use in designing interfaces by extending the user interface principle of *WYSIWYG*, What You See is What You Get, to *WYDIWYS*, What You Do Is What You See!

Acknowledgements

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada and with a grant and internship program from Apple Computer Inc. It was undertaken at Apple Computer Inc., Cupertino, and at the Dynamic Graphics Project and the Department of Landscape Architecture, both at the University of Toronto. We are especially indebted to Prof. John Danahy and his students at the Department of Landscape Architecture, as well as Jim Batson in the Advance Technology Group (Apple Computer Inc.) for the use of their IRIS' on short notice.

We would like to thank Peter Tanner for helping us reproduce their interaction technique for the experiment. We also thank Bill Buxton, Eric Hulteen, and Bill Gaver for suggestions and comments on this paper.

APPENDIX A: The Implementation of the Virtual Sphere Controller

A.1. Rotation of a 3-D Trackball

On a 3-D trackball (see Figure 9), if one touches the ball at point **P**, and rotates it in a tangential direction $\vec{d}$, the axis of positive rotation $\vec{a}$, can be computed by the cross product:

$$\vec{a} = \overrightarrow{OP} \times \vec{d}$$

where **O** is the centre of the trackball, and $\overrightarrow{OP}$ is a vector from the point **O** to **P**.

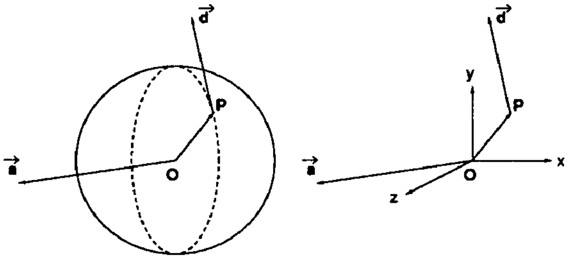


Figure 9. Rotation of a 3-D trackball.

A.2. Emulation of a 3-D Trackball Using a 2-D Control Device

The computation of the corresponding axis of rotation using a 2-D control device is done in three steps as shown in Figures 10, 11 and 12.

Step 1:

Figure 10 shows the top hemisphere of the 3-D trackball conceptually being flattened into-a disk. Let **O'** be the centre of the disk. Let **P'** be the starting point where the 2-D control device is first moved, and $\vec{d'}$ be the direction of movement. If **P'** = **O'** and $\vec{d'}$ makes angle τ' with the x-axis, the axis of rotation is on the x-y plane perpendicular to $\vec{d'}$ and can be obtained from equation (1):

$$\vec{a}(x,y,z) = [-\sin(\tau')\ \cos(\tau')\ 0] \tag{1}$$

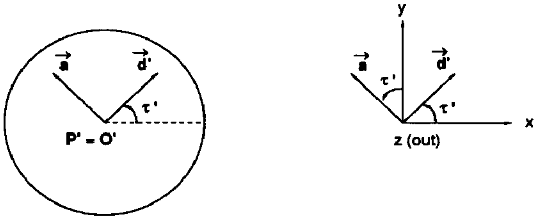


Figure 10. Movement of a 2–D device where P' = O'.

Step 2:

If **P'** is on the positive x-axis along the line $\overline{\mathbf{O'R'}}$ as show in Figure 11, the axis of rotation is that obtained from equation (1) but rotated by $\omega = \mathbf{f}\left(\frac{|\overrightarrow{\mathbf{O'P'}}|}{|\overline{\mathbf{O'R'}}|}\right)$ degrees about the y-axis. Namely,

$$\vec{a}\,(x,y,z) = \left[-\sin(\tau')\ \cos(\tau')\ 0\right] \cdot \begin{bmatrix} \cos f(\omega) & 0 & -\sin f(\omega) \\ 0 & 1 & 0 \\ \sin f(\omega) & 0 & \cos f(\omega) \end{bmatrix} \qquad (2)$$

where $|\overrightarrow{\mathbf{O'P'}}|$ and $|\overline{\mathbf{O'R'}}|$ are the length of the (2-D) vector $\overrightarrow{\mathbf{O'P'}}$ and the line $\overline{\mathbf{O'R'}}$ respectively, and **f**(x) can be any monotonically-increasing function with conditions:

$$\mathbf{f}(x) = \begin{cases} 0^\circ & \text{if} \quad x \le 0 \\ 90^\circ & \text{if} \quad x \ge 1 \end{cases}$$

The function **f**(x) describes how the hemisphere is distorted into the flat disk. The Virtual Sphere controller in the experiments used **f**(x) = x, with the above constraints. Note that if $|\overrightarrow{\mathbf{O'P'}}| = 0$, equation (2) is the same as equation (1). If $|\overrightarrow{\mathbf{O'P'}}| = |\overline{\mathbf{O'R'}}|$, then the axis of rotation is on the y-z plane.

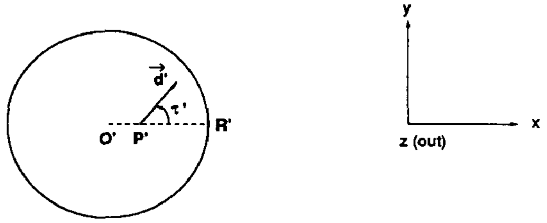


Figure 11. Movement of the 2-D device where **P'** is on $\overline{\mathbf{O'R'}}$.

Step 3:

In the general case (see Figure 12),

$\overrightarrow{\mathbf{O'P'}}$ makes angle θ' with the x-axis, and
$\vec{d}$ " " θ'+τ' " ".

Since Figure 12 is just Figure 11 rotated by θ' degrees about the z-axis, the axis of rotation is that obtained from equation (2) excepted rotated by θ' degrees about z. Namely,

$$\vec{a}\,(x,y,z) = \left[-\sin(\tau')\ \cos(\tau')\ 0\right] \cdot \begin{bmatrix} \cos f(\omega) & 0 & -\sin f(\omega) \\ 0 & 1 & 0 \\ \sin f(\omega) & 0 & \cos f(\omega) \end{bmatrix} \cdot \begin{bmatrix} \cos\theta' & \sin\theta' & 0 \\ -\sin\theta' & \cos\theta' & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

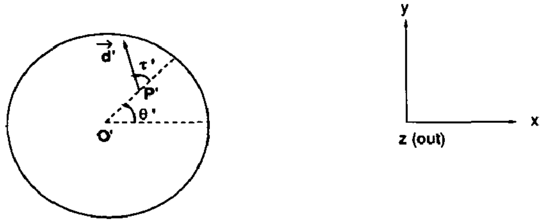


Figure 12. Movement of the 2-D device where **P'** is arbitrarily located.

Once the axis of rotation is obtained from equation (3), the rotation matrix **R** can be computed by:

$$\mathbf{R}_{\vec{a}}\,(\varphi) = \begin{bmatrix} ta_x^2+c & ta_xa_y+sa_z & ta_xa_z-sa_y \\ ta_xa_y-sa_z & ta_y^2+c & ta_ya_z+sa_x \\ ta_xa_z+sa_y & ta_ya_z-sa_x & ta_z^2+c \end{bmatrix} \qquad (4)$$

where a_x, a_y and a_z are the components of $\vec{a}$, s = sinφ, c = cosφ, and t = 1-cosφ, and φ is the amount of rotation about $\vec{a}$ [7].

The angle of rotation φ can simply be the distance of cursor movement times a suitable scaling factor. However, we decided to model the rolling of the sphere more precisely. We scaled the amount of rotation such that:

1) a full sweep of the mouse across the circle (passing through **O'**) produces 180 degrees of rotation;
2) a full circle around the edge (or outside) the circle produces 360 degrees of rotation.

The following formula for φ in degrees (obtained empirically) was used in the experiment, and provides a good approximation to the two desirable properties described above:

$$\varphi = 90^\circ * \frac{|\vec{d}|}{|\overline{\mathbf{O'R'}}|} \left\{ 1 - (1 - \frac{0.2}{\pi}) \frac{\omega}{90^\circ} (1 - |\cos\tau'|) \right\} \qquad (5)$$

References

1. Bier, Eric A. Skitters and Jacks: Interactive 3-D Positioning Tools. In *Proceedings 1986 Workshop on Interactive 3-D Graphics* (Chapel Hill, North Carolina, October 1986), 183-196.

2. Buxton, William. There's More to Interaction than Meets the Eye: Some Issues in Manual Input. In *User Centred System Design: New Perspectives on Human-computer Interaction*, D. A. Norman, & S. W. Draper Eds. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986. pp. 319-337.

3. Chen, Michael. A Technique for Specifying Rotations in Three Dimensions Using a 2D Input Device. In *Proceedings IEEE Montech'87 – Compint'87* (Montréal, Québec, November 1987), 118-120.

4. Evans, Kenneth B, Tanner, Peter P. & Wein, Marceli. Tablet Based Valuators that Provides One, Two or Three Degrees of Freedom. Proceedings of SIGGRAPH'81 (Dallas, Texas, August 1981). In *Computer Graphics* 15, 3 (August 1981), 91-97.

5. Mountford, S. Joy, Spires, Shannon & Korner, Kim. Visage: A Three-Dimensional Graphics Editor – Evaluation and Review. MCC Technical Report #HI-105-86-P, Microelectronics and Computer Technology Corporation, Austin, Texas, 1986. Data presented at 31st Annual Meeting of *Human Factors Society* in New York, October 19-22, 1987.

6. Nielson, Gregory M, Olsen, Dan R. Jr. Direct Manipulation Techniques for 3-D Objects Using 2D Locator Devices. In *Proceedings 1986 Workshop on Interactive 3-D Graphics* (Chapel Hill, North Carolina, October 1986), 175-182.

7. Pique, Michael E. Semantics of Interactive Rotations. In *Proceedings 1986 Workshop on Interactive 3-D Graphics* (Chapel Hill, North Carolina, October 1986), 259-269.