

Comparative Evaluation of Desktop and Cave Environments for Learning Hypercube Rotations

Prabhat, David H. Laidlaw
Department of Computer Science
prabhat@cs.brown.edu, dhl@cs.brown.edu

Thomas F. Banchoff
Department of Mathematics
Thomas_Banchoff@Brown.Edu

Cullen D. Jackson
Department of Psychology
Cullen_Jackson@Brown.Edu

Brown University, Providence, RI 02912

ABSTRACT

We compare user performance in a Cave and on a desktop for rotating a cube and a hypercube using interactive controls. Initial results indicate that users are faster on the desktop but more accurate in the Cave. Responses to a post-experiment questionnaire indicate that people learn more about the geometry of the hypercube in the Cave. We also present the notion of using a geometric control space to perform rotations. In particular, we develop the concept of using a triangle for performing 3D rotations and a tetrahedral control space for performing 4D rotations. We implement these on a desktop and a Cave system and use them to evaluate the environments.

KEYWORDS: virtual reality, user evaluation, immersion, hypercube, learning task

INTRODUCTION

While virtual reality is compelling and very effective for demos, its relative strengths and weaknesses, compared with traditional desktop user interfaces, have not been well established for specific applications. Ware, *et al.*, have measured the utility of using head-tracking and stereo for understanding 3D graphs [8], work that we believe can be generalized to other domains. In a similar spirit, we have measured accuracy and speed of users in manipulating the orientation of a cube in 3D and a hypercube in 4D and also quantified the effectiveness of the learning process involved.

Our motivations in choosing these tasks were twofold. First, we were interested in teaching about 4D orientations and rotations and wanted to understand the relative merits of the two user interaction environments for this specific task. Second, we felt that performance on these tasks would generalize to other unfamiliar tasks that users need to learn within

interactive environments – our users were not familiar with 4D interactions before the experiments

Interactive computer graphics has been used to visualize higher dimensional mathematical objects for quite some time. Unfortunately, simple interactions like rotations, which are natural in the 3D world, become inherently complex in higher dimensions. In this work we develop the notion of using a geometric control space for performing such rotations. Building on the analogy of using a triangle to perform 3D rotations, we develop a tetrahedral control space to perform 4D rotations.

In the following sections, we describe the hypothesis for the user study, the theory of the control space and details of the implementation on both systems. After that we describe the experimental design. Finally, we discuss the results and issues.

Related Work

Several researchers have studied different interaction techniques for performing 3D rotations on a desktop. Hinkley, *et al.* conduct a study on comparing performance of 3D input devices vs. mouse-driven interaction techniques for an orientation matching task[3]. Poupyrev, *et al.* use non-isomorphic techniques to design effective spatial user interfaces[6]. Our work focuses on 4D rotations, an unfamiliar task, as a way to compare a Cave to a desktop interaction environment.

Both Ware, *et al.*[8], and Pausch, *et al.*[5], compare different computer graphics viewing/interaction paradigms, i.e., with and without stereo or head tracking, for performance of specific tasks. Our work is in this spirit, but compares environments using tasks at a level closer to the application and with different user interfaces in the different environments. The interfaces differ in an attempt to make them more appropriate for the environments in an attempt to avoid biasing results with an inappropriate user interface.

Kaufmann, *et al.* use VR for mathematics and geometry education [4]. They have conducted an informal pilot study to evaluate the efficiency and value of the Construct3D system,

but there has been no formal comparison with a desktop system.

There are also applications for viewing and interacting with 4-dimensional geometry. Geomview [1], for example, uses a combination of mouse button mappings to rotate 4D objects. Our emphasis goes beyond developing interaction methods for viewing 4D objects to evaluating different interaction environments.

USER STUDY

Hypothesis

We pose the experimental hypothesis formally as follows :

- Users will perform 4D rotations faster in an immersive medium (Cave [2]) than in a non-immersive medium (desktop).
- Users will perform 4D rotations more accurately in the Cave.
- They will learn more about the hypercube and its 4D rotations in the Cave.

3D Rotation Tasks

The 3D rotations tasks essentially ask the users to rotate a cube so that it matches up with another cube. Initially there is a training phase in which the users are shown how moving a point inside the triangle causes the cube to rotate. In the first set of tasks, the users are asked to repeat these rotations. They are encouraged to understand the rotations as they interact with the triangle and are given feedback on their accuracy and speed.

After the training is finished, a second set of tasks is shown for which the users are timed. Fig.2 shows the experimental setup for 3D rotation tasks. A target orientation of the cube is displayed on the right side. The cube which they control is displayed on the left side. They are then instructed to move the point in the triangle so that both the views match up. They are also told to be as quick and accurate as possible.

4D Rotation Tasks

In the 4D rotation tasks, the control is a tetrahedron and the object manipulated is a hypercube. The task sequence is exactly the same as earlier: during the training phase, a point moves on the edges and faces of the tetrahedron and the hypercube rotates accordingly. In the first set of tasks, the users are instructed to repeat these rotations. They are encouraged to understand the rotations as they interact with the tetrahedron and are given feedback on their accuracy and speed.

After the training is over, a second set of tasks is shown for which the users are timed. Fig.3 shows the experimental setup for 4D rotation tasks. A target orientation of the hypercube is displayed on the right side. The hypercube which they control is displayed on the left side. As usual, the users are instructed to be as quick and accurate as possible.

THEORY OF CONTROL SPACE

3D Rotations

Suppose we are interested in looking at a cube placed so that the center of the cube is at the center, C , of a sphere (see Fig.1). Consider an octant (1/8th) of a sphere. You can view the object by choosing a point on the surface of the octant and

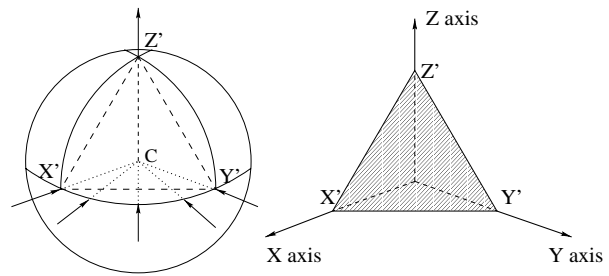


Figure 1: Control Space

looking inwards in the direction of the radius. If you were on the point X' the view would be as if you were looking down the X -axis. Similarly at the point Y' the view would correspond to the Y -axis and Z' would correspond to the Z -axis. As your chosen point moves on the sphere, you see different views of the cube, so it appears to rotate. Because of the symmetry of the cube, the octant permits all possible views.

Now suppose that instead of moving on the octant surface we move on the triangle $X'Y'Z'$. Each radius, starting from the center C , would hit the triangle first before hitting the octant. Therefore we can get exactly the same effect by moving a point on the triangle as moving on a spherical surface. Moving on the triangle will give us rotations in 3D.

If we move along the edge $X'Y'$, this corresponds to a rotation in the XY plane with the z coordinate held fixed. Choosing a point P in the interior of the triangle $X'Y'Z'$ gives a projection into the plane in 3-space perpendicular to the vector P .

4D Rotations

In 3D we had a triangle where each vertex represented an axis (X , Y or Z). When we add one more dimension, we'll have one more vertex corresponding to the W -axis; the shape corresponding to the triangle is a tetrahedron.

As in the 3D case, each vertex of the tetrahedron corresponds to the view looking down that axis; for example, point X' corresponds to the view down the X -axis, Y' corresponds to the view down Y -axis and so on.

If we move along the edge $X'Y'$ in the tetrahedron, then, as in the 3D case, this corresponds to a rotation in the XY plane, this time with both the z and the w coordinates being held fixed. Choosing a point P in the interior of the triangle $X'Y'Z'$ gives a projection into the hyperplane in 4-space perpendicular to the vector P .

USER INTERFACE

3D rotations on the desktop

For 3D rotations of a cube, we display the flat 2D projections of the cube in a Cave-like room, textured with walls. The triangle is drawn in the foreground and the user is able to move a point on the triangle by dragging on it with the left mouse button. Releasing the mouse button at any point keeps the point where it was last dragged.

The user begins the interaction by pressing the spacebar.

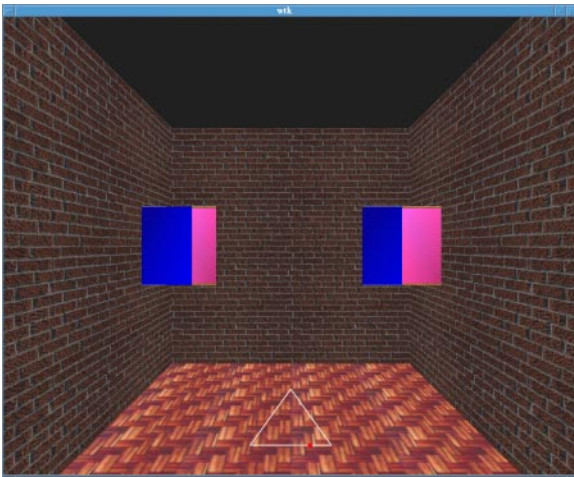


Figure 2: Snapshot of desktop version for cube rotations

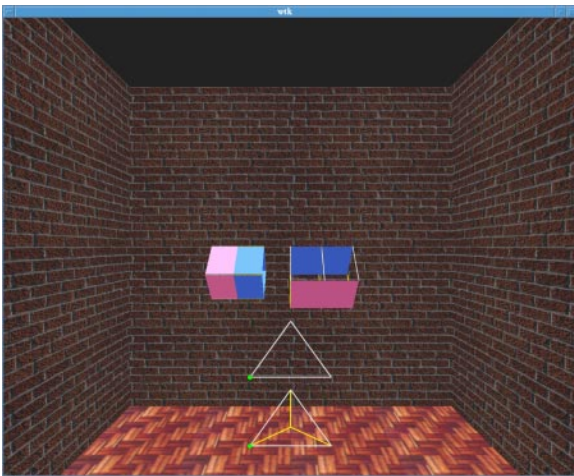


Figure 3: Snapshot of desktop version for hypercube rotations

When the user is finished with the rotation, he presses the spacebar again. A new task is loaded and the user presses the spacebar again to start.

4D rotations on the desktop

For 4D rotations of a hypercube, we display the 3D projections in the same room. There are 2 issues that arise :

- We need a 2D representation of a tetrahedron
- We now need to match 3D objects

We will discuss these issues and our proposed solution next :

We represent the four different faces of the tetrahedron with two triangles drawn in the foreground. One of the triangles represents the base and the other shows the top-down view of the tetrahedron with the three side faces. The problem with this representation is the discontinuity in moving from the base edges to the base or the side. The way we tackle this problem is to display the point on both the base triangle and the bottom triangle, and giving the user the option to

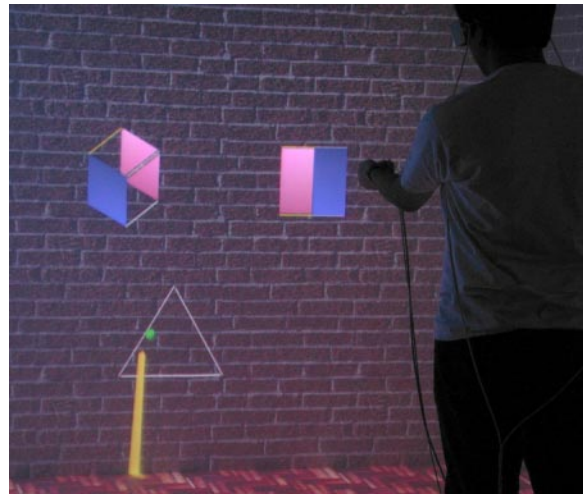


Figure 4: User performing cube rotations in the Cave

move on the face that he wants. We discuss the problems with alternate solutions in the Issues Section.

The hypercubes project onto 3D objects displayed in the scene. The user now has to match 3D objects as opposed to flat projections. Placing two 3D objects offset from each other in space means that they will not appear to be the same at any viewpoint. This problem can be solved only if they are placed together, centered in the same point in space, but that would create far too many occlusions for the purpose of our experiment.

So we decide to offset the hypercubes both horizontally and in depth. The reason for offsetting in depth is because very often the users want to move to a position from which they can have a side-on view of the objects. If the hypercubes are lined up with each other in depth, apart from the effects of perspective, the user's view of the further off hypercube will be occluded. This is a drawback in this experimental setup, but it is inherent to the nature of the task.

On the desktop implementation, we allow the user to rotate in a horizontal arc around the hypercubes until he touches the walls of the room. We also allow the user to rock the objects slightly. He can press the left and right arrow keys on the keyboard to move left or right in the arc. The up arrow key can be used to rock the objects. All the other interactions, namely the mouse-dragging and the spacebar start-stop sequences, are the same as before.

3D rotations in the Cave

We texture the walls of the cave with a brick texture. Having a texture on the walls reduces perceptual ambiguities because the focal distance of the virtual wall matches the focal distance of the real projection surface. It also gives a nearby background against which the foreground objects can be more easily segmented. The regular pattern helps with stereo fusion.

The flat projections of the cube and the triangle float in front of the user. The user interacts with them with the help of a wand. A virtual stick emerges from the wand and is oriented

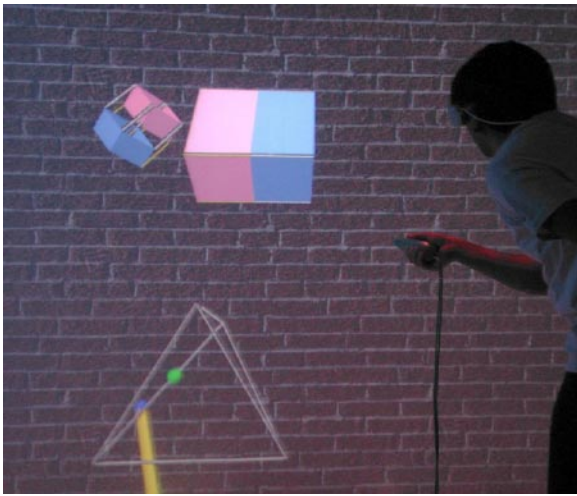


Figure 5: User performing hypercube rotations in the Cave

in the direction of the wand. Users have commented that they found this to be a fairly natural form of interaction. The wand has three buttons on it. We decided to use the right button to indicate a start-stop sequence similar to the spacebar in the desktop version. The left wand button works similar to the left mouse button in the desktop, if you keep it pressed down, it drags the point on the triangle. If you release the button, the point remains where it was last dragged.

4D rotations in the Cave

For the hypercube, we maintain the same environment as before. Now we have two 3D objects and, for reasons previously mentioned, we offset them horizontally and in depth. The wand-stick interaction remains the same as before. The major difference is that our control space is now a 3D tetrahedron rather than a flat triangle. So now we require a scheme for projecting the end-point of the stick on the surface of the tetrahedron. We tried to implement a naive projection technique by which the end-point would be projected to the closest point on the face along the line, joining it and the center of the tetrahedron. But it turns out the movement along edges becomes very tough as you need to move on a very fine arc to project onto the edges. For the purpose of our experiment we know that movement along edges is very important, so we chose to implement a projection technique which favors edges. Essentially the idea is that the end-point of the stick is projected perpendicular to the face, and when it does not find the face, it sticks to the closest edge. Since there are a greater number of points that map onto edges as compared to the naive radial projection approach, the user finds it easier to stick to the edges. We have found this technique to be very effective for our set of tasks.

EXPERIMENTAL DESIGN

Variables

The choice of environment (Cave vs. desktop) is the control variable. There are many characteristics inherent to the two environments which will thus be included in the comparison and which cannot be separated out. We argue that we would

like each environment be the best it can, and then make the comparison.

- Tetrahedral control space and interaction

The tetrahedral control space is not represented in exactly the same way on the desktop and the Cave. While it is a 3D object in the Cave, it is represented as two flat triangles on the desktop. We could opt for drawing a 3D tetrahedron on the desktop, but for reasons mentioned previously, we have to avoid that.

The same issue holds true for the interaction. While the user's hand movement is constrained to a flat surface (mouse movements) on the desktop, he is free to move in space in the Cave. He uses mouse dragging on the desktop, where as in the Cave he uses a wand-stick interaction.

We understand that this introduces extra variables in the study, but we would not like to change the natural way of interaction on the desktop or the Cave. We take advantage of the 3D possibilities of the Cave and the familiarity of the desktop and try to see which one is better.

- Size of objects

The objects on the desktop subtend a certain FOV on the eye of the user. To subtend exactly the same FOV in the Cave, the objects would have to be very small relative to the size of the Cave environment. On the other hand, increasing the size of the objects on the desktop would mean that they would overlap with each other. So we let the program maintain the same size of object internally. When it displays them on the desktop, it uses the desktop parameters (window size and viewpoint). When it displays them in the Cave it uses the Cave parameters (wall size and viewpoint). Since the viewpoint is set to be the same in both the environments, size remains the only variable that we do not account for.

Again: we reason that we should show a size, which is natural for the environment that we are working in. On the desktop the objects appear to be the right size, as compared to the size of the screen. In the Cave the objects appear to be of the right size as compared to the walls and the user.

Subject Pool

The users for this study come from a pool of undergraduate students who are taking a course in Honors Linear Algebra. All of the students are freshmen. Since the participation is restricted to the course students and is voluntary, we expect the users to be motivated and interested in the mathematical visualization. We pay the users \$10 for their participation.

A total of 18 students participated in the user study. 10 of them were randomly assigned to do the desktop version and 8 were assigned to do the Cave version. There was one female student in each of the two groups.

Pre-experiment Questionnaire

In the pre-experiment questionnaire, we are interested in knowing about the background of the user. We ask questions about his experience in 2D and 3D interactions. We ask him if he has worked on a VR system before. We also try to find out about his exposure to four-dimensional objects and four-dimensional visualization programs. He is asked if he has

seen a demo/video of a hypercube, read about, or interacted with it.

Data acquired

During the experiment, we time the user for the duration they take to complete the task. The time starts when they press the spacebar (desktop) or the right wand button (Cave). The time stops when the user presses the spacebar or the right wand button again.

We also note the accuracies of the user when he finishes the task. We find the distance between barycentric coordinates of the user's point and barycentric coordinates of the point corresponding to the right match. Then we take the ratio of this distance, to the distance between the barycentric coordinates of the points, corresponding to the maximum error he could make. It is important to note that we are plotting the Error Rate in the graphs in this paper. We define $ErrorRate = 1 - Accuracy$.

Another important piece of information that we record is the user's viewpoint position when he decides that the hypercubes are matched. We do not plan to conduct an extensive analysis of this data, but it should be possible to investigate if users prefer to move around in a certain region of the Cave or prefer certain viewpoints to perform the matching task. This would also require collecting the entire stream of head-movement data which we are not doing right now.

Post-experiment Questionnaire

The post-experiment questionnaire is very important for this user study. Apart from the times and accuracy that we obtain from the actual experimental tasks, we would like to know if the user understood or learned something about a hypercube and its 4D rotations. It is very difficult to formulate questions that definitively estimate the extent to which the person learned the object or the rotations. Nevertheless we have formulated a number of questions which ask about the person's knowledge of hypercubes. Some examples of each type of questions are as follows:

- How many vertices does a hypercube have ?
- How many edges does a hypercube have ?
- Draw three different hypercubes.
- Complete the following sketch of a hypercube.
- Find the vertex opposite vertex A in the following hypercube.
- Find the face opposite face F in the following hypercube.
- The following starting and destination orientations lie on an edge of the tetrahedron. Identify the intermediate orientation that the hypercube would pass through while rotating from one to the other.

The questions range from simple questions about the number of edges and vertices of a hypercube to more complex ones involving filling in the missing lines of partially drawn hypercubes. More visual questions on drawing three different hypercubes help us to find if the experiment left an impression on the user's mind and if he remembers the shapes. Questions on finding the opposite vertex and opposite face have more to do with testing if the user understands the geometry of the hypercube. The last set of questions ask the user to identify the intermediate shape that the hypercube would

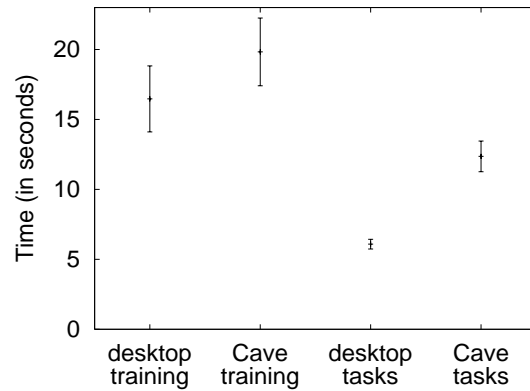


Figure 8: Time comparison for cube rotations

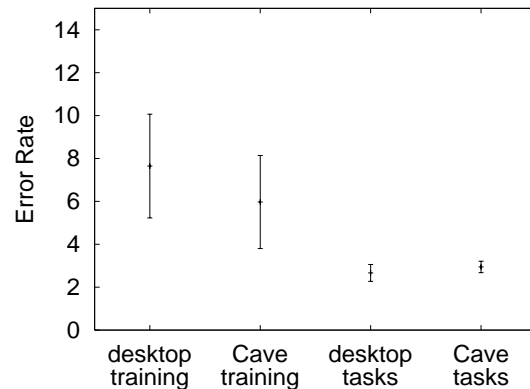


Figure 9: Accuracy comparison for cube rotations

pass through while moving along an edge. We feel that this set of questions would force the users to mentally go through the rotation and visualize how one shape would transform into the other.

RESULTS AND DISCUSSIONS

Statistics were computed based on averaging the results (time, accuracy) for a given task for all the users. Out of the 24 cube rotations tasks, the initial 9 tasks are for training. Out of the 31 hypercube tasks, the initial 10 tasks are for training. The training tasks were not used for calculation of statistical significance. The number of training tasks was determined by a pilot run of two users in each environment. They were presented with a minimal set of representative tasks, and we noted the number of trials that they took to reach a desirable level of efficiency. The number of actual tasks were determined by splitting the triangle and tetrahedron into regions. We then choose starting and ending regions based on a distance criterion, and then we randomly choose a point within the selected regions. It must be mentioned that all tasks are not equally hard, and that the level of difficulty increases towards the end of the experiment.

Rotation tasks

We see that for the cube tasks, the times are statistically significant in favor of the desktop (see Table 1 and Fig. 8). This might be because people are very familiar with using a mouse on the desktop. The desktop interaction involves dragging,

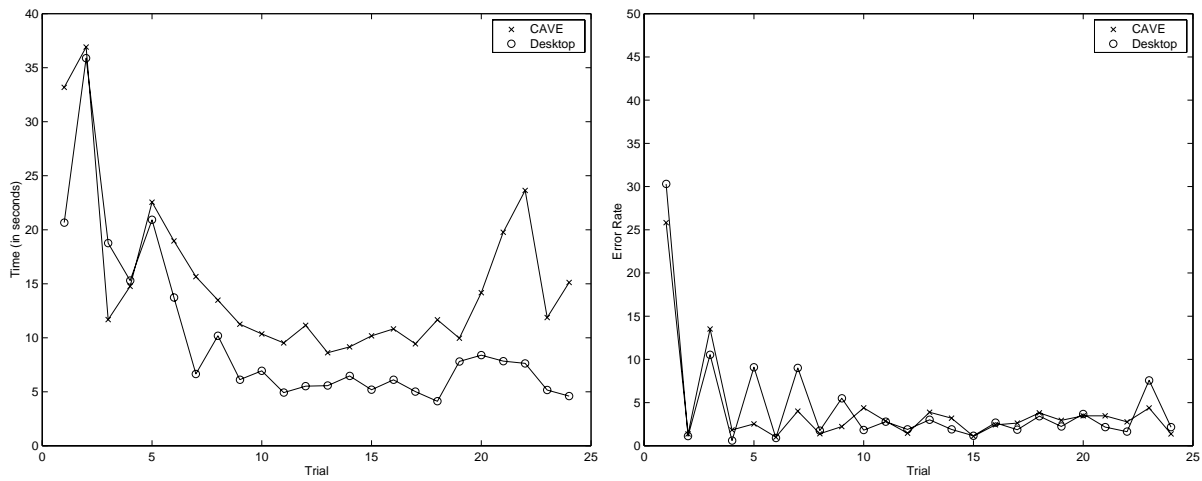


Figure 6: Time and accuracy plots for cube rotations

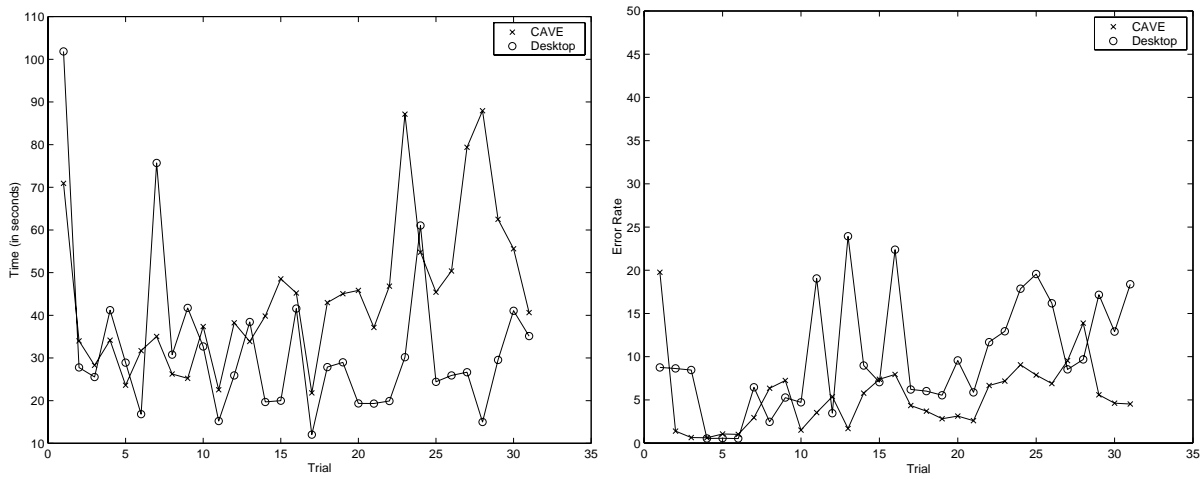


Figure 7: Time and accuracy plots for hypercube rotations

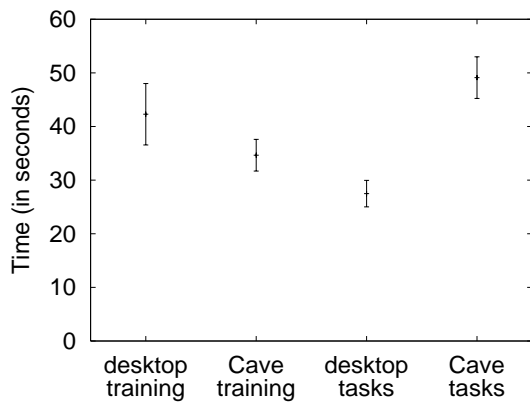


Figure 10: Time comparison for hypercube rotations

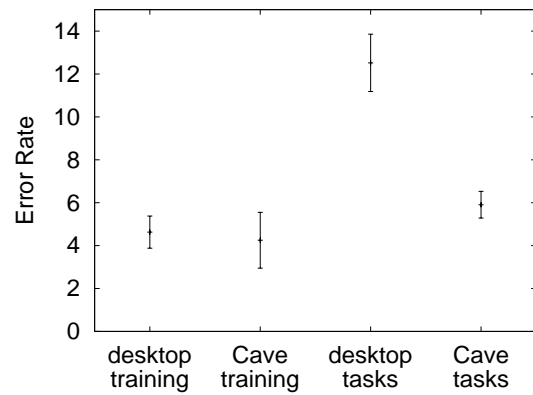


Figure 11: Accuracy comparison for hypercube rotations

| Comparison | <i>F</i> | <i>p</i> |
|------------------------------|----------|----------|
| Cube Tasks (Times) | 29.7 | < 0.001 |
| Cube Tasks (Accuracies) | 0.3426 | > 0.5 |
| Hypercube Tasks (Times) | 22.11 | < 0.001 |
| Hypercube Tasks (Accuracies) | 20.12 | < 0.001 |

Table 1: Comparison of desktop and Cave environments for accuracy and time using ANOVA.

which is also a very common mouse operation. On the other hand, using the wand-stick interaction in the Cave was new to all the users. Although they became comfortable with it, we need to observe that their hand motion is not constrained to a flat surface.

It is interesting to note that the accuracies for the cube tasks are not statistically significant (see Fig. 9) even though the physical representation of the tetrahedron on the desktop and Cave are not of the same size. This is very interesting vis-a-vis the accuracies for the hypercube tasks. Even though the physical size of the tetrahedron remains the same as earlier in both the environments, the accuracies become statistically significant in favor of the Cave (see Fig. 11). We believe that immersion is responsible for the differences rather than limitations of the interface implementation. The users are better able to visually estimate the similarity of two 3D objects in the Cave relative to the desktop. Although the desktop users have the functionality to rotate around and rock the objects slightly, apparently that is not sufficient to estimate whether the objects are the same.

For the hypercube tasks, the times are statistically significant in favor of the desktop (see Fig. 10). Most probably this is because the desktop user is still moving on a flat surface, and the only new complexity is in the time it takes for him to jump from the base to the side faces or vice versa; the Cave user has to learn a new interaction scheme altogether. He has to move on the surface of a 3D object. Even though the user became comfortable with the interaction, expecting them to acquire a level of proficiency comparable to the desktop in the duration of the user study is unrealistic. Moreover from the personal observation of the users while doing the tasks, it was apparent that in case the user did not know where the answer was, he found it faster to traverse the whole space on the desktop. In the Cave the user was forced to be systematic in his exploration, owing to the nature of the control space interaction.

Post-experiment Questionnaire Responses

We observe that the main source of difference in the totals of the post-experiment questionnaire is due to the questions on diagram completion and number of edges and vertices (see Table 2). Both these types of questions require the user to think about the geometry of the hypercube and to draw or analyze what they remember. (We took special care in the evaluation of the drawing questions not to penalize the users who had poor drawing skills. As long as they had the conceptual figure correctly drawn, we gave them most of the points.)

It is interesting to note that there was not much of a difference among the totals for the 4D rotation questions. This is despite the fact that many of the desktop users (in a

| | Desktop (mean) | std dev | Cave (mean) | std dev |
|----------------------------------|-------------------|------------|----------------|------------|
| Understanding of 3D rotations | 1.56 | 0.53 | 1.43 | 0.53 |
| 4D rotations | 3.72 | 0.87 | 3.00 | 0.58 |

Table 3: Self Rating on Scale of 1-7

post-experiment interview) admitted to guessing the answers rather than thinking about the rotations. Having more than 2 choices for the correct answer, would have reduced the effects of guessing.

The users were also asked to rate themselves on their understanding of the cube and hypercube rotations. They were asked to respond on a scale of 1 - 7, in which 1 means that they understood everything perfectly and 7 means that they did not understand anything. The ratings are presented in Table 3. Even though the ratings are not statistically significant, they do indicate that the Cave users were slightly more confident about their understanding of the hypercube rotations. This is also supported by subjective opinions expressed by the users in a post-experiment interview. The Cave users 'felt' that they understood relationships between movements on edges and the corresponding rotations better than the desktop users. Two Cave users are known to have remarked that they knew which edge/face the correct orientation would lie on, and that the only problem was in getting the hypercubes to match up precisely.

Almost all the users expressed the desire to remain with the experiment for a longer period of time so that they could better understand the underlying mathematics. As indicated before, it subjectively appears that the desktop users seem to be slightly worse off in understanding the relationship between the control space and the rotations. We think that apart from the immersion, one important reason could have been that the users in the Cave were naturally constrained to be systematic in their exploration of the control space whereas the desktop users had more freedom in moving on the tetrahedron.

Issues

There are a number of issues which might have affected the results of the study.

The representation of the tetrahedron in the Cave and desktop is not exactly the same. We could have displayed the tetrahedron as a 3D object on the desktop, but then we would be able to see only three faces at one point in time. In order to get to the other face we would need to rotate the tetrahedron. The user is in a continuous process of building a mental map of how positions on the tetrahedron map onto orientations. Asking the user to rotate the tetrahedron will imply that he rotate his mental map of orientations too, which will complicate the task. Other options that we looked into were unfolding the faces of the tetrahedron into a star-shaped object. The problem with that is when the user moves on any edge, the point appears to move on another edge in the opposite direction. In the worst case on the vertex, the point will appear separately on three different triangles. This can be visually confusing.

Our implementation solves these problems to some extent, but it is relatively easier to move quickly on all the four faces

| <i>Questions on</i> | <i>max</i> <i>score</i> | <i>Desktop</i> <i>(mean)</i> | <i>std.dev</i> | <i>Cave</i> <i>(mean)</i> | <i>std.dev</i> |
|---------------------------------|----------------------------|---------------------------------|----------------|------------------------------|----------------|
| 4D rotations | 30 | 24.17 | 6.25 | 24.64 | 3.66 |
| Marking opposite faces/vertices | 32 | 29.78 | 3.53 | 31.43 | 1.51 |
| Completing diagrams | 26 | 17.44 | 3.28 | 22.64 | 2.23 |
| Number of edges/vertices | 12 | 4.44 | 4.67 | 9.71 | 3.90 |
| Total Score | 100 | 75.61 | | 88.28 | |

Table 2: Statistics computed for sections of the post-experiment questionnaire.

of the tetrahedron on the desktop. This helps the user to be faster, but the lack of constraints seems to affect the accuracy with which the user performs the task and the spatial map of orientation correspondences that he develops for himself. Having exactly the same interface in both environments would be the ideal solution. We are contemplating the idea of using a physical mockup or prop of a tetrahedron for performing the rotations.

Assuming that the background of the users is similar, or that they are at the same point on the learning curve before the experiment, is a tricky assumption. Based on what exposure they might have had to the subject (hypercubes) the experiment might have clarified some things for them to different extents. Although all the users indicated that they had either heard about a hypercube and seen a picture/demo before the experiment, none of them had interacted with a hypercube before. We have reason to believe that responses to the post-experiment questionnaire might have been biased by their previous experience. For example, some users drew hypercubes in orientations which were not possible to attain in our experimental setup. It might have been instructive to perform standard mental rotation tests [7] on our users before the experiment. Their score on the test might have helped us to split the users evenly across the two environments.

The uniform motivation of the subjects could have been an issue. There was a user in the desktop experiment who was finishing the tasks too quickly without bothering for accuracy. The Cave users could have been more motivated because of the excitement of working in an immersive environment for the first time.

CONCLUSIONS

We have presented a comparative evaluation of desktop and Cave environments for learning hypercube rotations. Users describe a preference for interacting with and learning about the geometry of higher dimensional objects in an immersive environment, suggesting that they find it more engaging. Quantitative results show that, although people perform faster on the desktop, they are more accurate in the Cave. Cave users appear to learn more about the geometry of the hypercube. There has been some question about the value of immersive environments such as the Cave. We believe that this work supports a more general hypothesis that the Cave is more effective than a desktop for applications requiring relatively complex spatial understanding.

ACKNOWLEDGMENT

Thanks to Daniel Acevedo, Çağatay Demiralp and Daniel Keefe for their help with technical issues in the Cave. Also, thanks to Tim Miller and Andrew Forsberg for helpful dis-

cussions. This work was partially supported by NSF(CCR-0086065) and used Cave facilities made available through a grant from NSF(9724347).

REFERENCES

1. Nina Amenta, Stuart Levy, Tamara Munzner, and Mark Phillips. Geomview: A system for geometric visualization. In *Symposium on Computational Geometry*, pages C12–C13, 1995.
2. Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 135–142, August 1993.
3. Ken Hinckley, Joe Tullio, Randy F. Pausch, Dennis Proffitt, and Neal F. Kassell. Usability analysis of 3d rotation techniques. In *ACM Symposium on User Interface Software and Technology*, pages 1–10, 1997.
4. Hannes Kaufmann, Dieter Schmalstieg, and Michael Wagner. Construct3d: A virtual reality application for mathematics and geometry education.
5. Randy Pausch, Dennis Proffitt, and George Williams. Quantifying immersion in virtual reality. *Computer Graphics (SIGGRAPH'97 proceedings)*, 31(Annual Conference Series):13–18, 1997.
6. Ivan Poupyrev, Suzanne Weghorst, and Sidney Fels. Non-isomorphic 3d rotational techniques. In *CHI*, pages 540–547, 2000.
7. Roger N. Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171, 1971.
8. Colin Ware and Glenn Franck. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Transactions on Graphics*, 15(2):121–140, 1996.