Comparative Evaluation of Desktop and Cave Environments for Learning Hypercube Rotations

Prabhat, David H. Laidlaw, Thomas F. Banchoff and Cullen D. Jackson

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-05-09
June 2005

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 for learning about the geometry of the hypercube and how it can be oriented in 4-space. This task was chosen because it is difficult to teach and yet has a geometric character that we expected would be more effectively addressed in an immersive environment like a Cave. Our results indicate that users are faster on the desktop but more accurate in the Cave. Responses to a post-experiment questionnaire show that people learn more about the geometry of the hypercube in the Cave.

Hypercube orientation changes are performed interactively by moving an indicator within a tetrahedral control space. We train users to use this tetrahedral control space for 4D orientations by first teaching them about an analogous, but simpler, triangular control space for 3D orientations. We implement these on a desktop and in a Cave system and use them to evaluate user performance in 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 demonstrated that head-tracking and stereo permit understanding of larger 3D graphs than static or moving monocular images [11]. We believe that this work can be generalized to other domains. In a similar spirit, we have measured the accuracy and speed of users in manipulating the orientation of a a hypercube in 4D and a cube in 3D and also evaluated how well users understand the hypercube and its orientations after the experiment.

Our motivations in choosing these tasks were twofold. First, we were interested in teaching about 4D orientations and the hypercube. We hypothesized that a Cave would make an effective learning environment and wanted to understand its merits relative to a standard desktop computer. Second, we felt that performance on these tasks would generalize to other unfamiliar geometric 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[2, 4, 5]. Manipulating these objects, however, becomes more difficult because simple interactions like rotating an object, which are familiar in our 3D world, become more complex and less familiar in higher dimensions. In this work we use the notion of a geometric control space for performing such rotations. For specifying and changing the orientation of a hypercube in 4-space, we use a tetrahedral control space. To teach about this control space for 4D rotations, we have also developed a simpler triangle control space to specify the orientation of a cube in 3D.

Our experimental approach was to develop different applications in each of the the two interaction environments, a Cave and a desktop computer. Because the concept of the control space is central to the learning process we were evaluating, users manipulated rotations using it in both environments. However, the interaction techniques for manipulating it were tailored to each environment. For example, in the Cave a 3D tracker was used for maneuvering through the space while on the desktop a mouse was used. By using appropriate interactions for each environment, we hoped to avoid biasing the results in favor of either.

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. We then describe the experimental design and discuss the results and issues.

RELATED WORK

Several researchers have studied different interaction techniques for performing 3D rotations on a desktop. Hinckley, *et al.* conduct a study on comparing performance of 3D input devices vs. mouse-driven interaction techniques for an orientation matching task[6]. Poupyrev, *et al.* use non-isomorphic techniques to design effective spatial user interfaces[9]. Previous results seem to indicate that 3D interaction techniques lead to faster performance than 2D interaction techniques for 3D rotation tasks. We would like to mention here that for 3D rotation tasks we are essentially using a 2D interaction technique (triangular control) both in the Cave and desktop. Moreover, 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.*[11], and Pausch, *et al.*[8], compare different computer graphics 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 to avoid biasing results with an inappropriate user interface.

Kaufmann, *et al.* use VR for mathematics and geometry education [7]. 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. We present an alternative method for specifying orientations in 4D and also have an additional evaluative component in our work.

USER STUDY

In this section we present our initial hypothesis and motivation for it, then describe the two types of tasks that users perform: 3D rotations of a cube and 4D rotations of a hypercube. The 3D rotations are training for understanding, by analogy, how the 4D rotations work.

Hypothesis

In comparing the Cave and desktop environments, we hypothesized that users would perform 4D rotations faster and more accurately and would learn more about the hypercube and its 4D rotations. We theorized that the immersive nature of the Cave [3] would provide a more intuitive learning environment better matched to the geometric nature of the problem.

3D Rotation Tasks

Users first performed a 3D orientation task analogous to the later 4D rotation task. They rotated a cube so that its orientation became identical to that of another cube. Initially, there is a training phase in which the users are shown how moving a point inside a triangle causes the cube to rotate. This is described in more detail below. In the first set of tasks, the users are asked to repeat these rotations. They are encouraged to understand the rotations at the same time as performing them as quickly and accurately as possible. They press a button to indicate completion of each task and are then given feedback on their accuracy and speed. After the training is finished, a second set of tasks is performed.

Fig. 2 shows the desktop experimental setup for 3D rotation tasks and Fig. 4 shows a photograph of the Cave setup. 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 of the views match up.

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 not only to be quick and accurate, but also to understand the rotations as they interact with the tetrahedron. They are given feedback on their accuracy and speed after each task is completed. Following the training, another set of timed tasks is performed.

Fig. 3 shows the experimental setup for 4D hypercube rotation tasks on the desktop. Fig. 5 shows a photograph of the Cave setup. A target orientation of the hypercube is displayed on the right side. The hypercube, which they control, is displayed on the left side.

THEORY OF CONTROL SPACES

3D Rotations

Users were able to perform 3D rotations of a cube by interactively specifying certain orientations of the cube. A point within an equilateral triangle controls the orientation as follows. Imagine a cube that can pivot about its center and that has a square stick-like handle protruding from its front face. The end of the stick is constrained to move only through one octant of the sphere that it could otherwise sweep out. Fig. 1 shows the sphere, with the octant represented by the points X', Y', and Z'. Those same vertices also define a triangle, and that triangle is the control space for the cube.

The handle is constrained so that it does not rotate, removing one possible degree of freedom for rotations of the cube. This is analogous to a limitation in the 4D rotations. The limitation simplifies the learning process in a way that we felt was appropriate for teaching purposes. This type of rotation would correspond to a rotation of the projected image. The constraint is implemented as if the two side faces of the square handle remained always perpendicular to the ground. The handle is further constrained to avoid the straight up position, where there would otherwise be a singularity.

With the control point within the triangle at the lower left corner, we see the face of the cube with the handle head-on. Moving the control point to the right corner rotates the cube about the vertical axis, orienting the face with the handle toward the right. Moving the control point toward the top

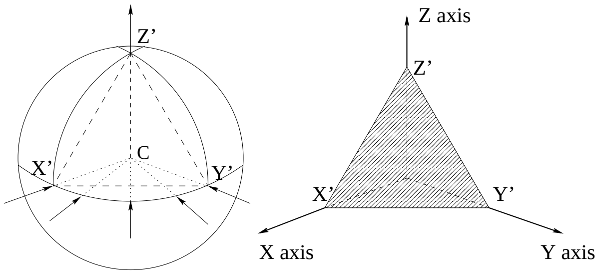

Figure 1: 3D control space for a cube.

vertex of the triangle rotates the face with the handle so that it becomes the top of the cube.

It should be noted that no visible representation of the handle is present – it is described here only to aid in the explanation of the control space.

4D Rotations

The 4D rotation controls are analogous to the 3D rotation controls. In the 3D case, the control handle determined the projection direction from the 3D to the 2D image plane. The handle now determines the projection direction from 4D to 3D. As in the 3D rotation case, the orientation of the projection of the hypercube is not manipulable. It stays "upright" in the same way that the orientation of the cube projection stays upright.

The control space through which the end of the handle moves is now a tetrahedron. As a cube can be constructed from six square faces, a hypercube can be constructed from eight cubes. The handle is attached to one of these 3D cubes. The vertices of the tetrahedral control space correspond to orientations where the 3D cube of the hypercube is pointing, respectively, forward, right, up, and into the fourth dimension. These four positions correspond to the vertices and are not particularly interesting because they look like cubes. This is analogous to the 3D case where the vertices correspond to views that show only a single square face. Intermediate positions of the control point, however, show orientations that illustrate more of the structure of the hypercube, and it is from these views and the way that they are connected, that users get an understanding of the hypercube and of 4D rotations.

One additional constraint is imposed on the 4D orientations that can be reached. The control point is constrained to lie on the surface of the tetrahedron. This simplifies the possible projections while still including a significant contribution from the fourth dimension. Again, this simplification helps in the teaching process.

USER INTERFACE

3D rotations on the desktop

For 3D rotations of a cube, we display the flat 2D projections hanging within a Cave-like room with textured walls (see Fig. 2). To obtain the flat projections, we use parallel projection from 3D to 2D. As the cube rotates in 3D, the shape of its projection changes.

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

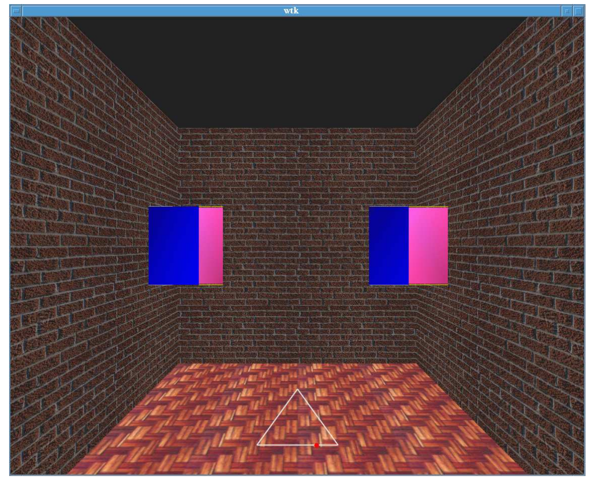

Figure 2: Snapshot of desktop version for cube rotations

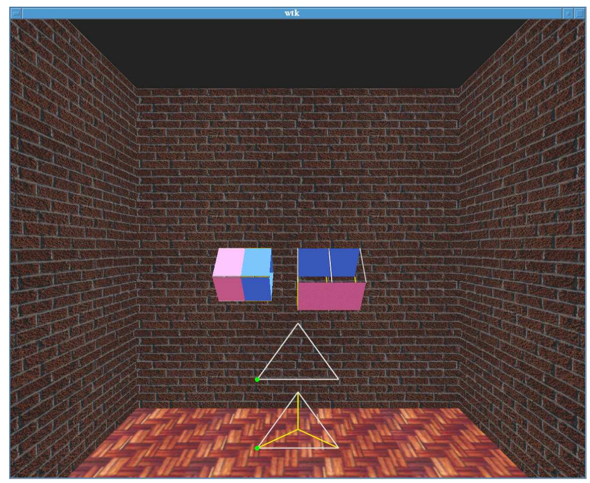

Figure 3: Snapshot of desktop version for hypercube rotations

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. 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 virtual room. There are 2 issues that arise: first, we need a 2D representation of a tetrahedron, and second, we now need to allow users to match orientations of 3D objects.

We represent the four different faces of the tetrahedron with two triangles drawn in the foreground (see Fig. 3). 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

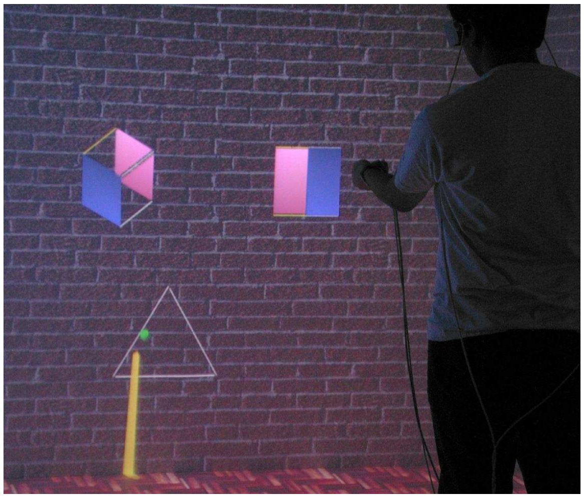

Figure 4: User performing cube rotations in the Cave

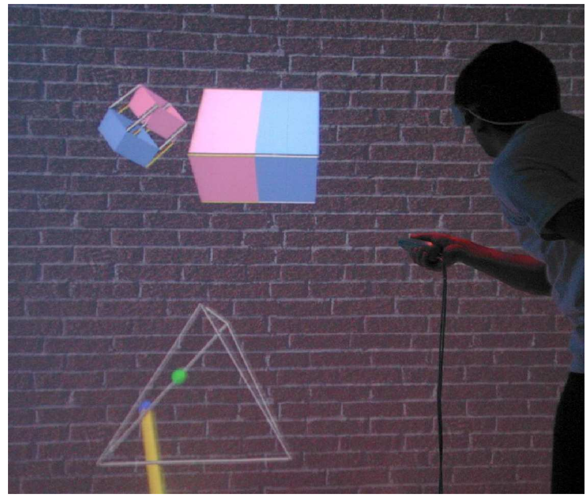

Figure 5: User performing hypercube rotations in the Cave

triangle and the bottom triangle, and giving the user the option to move on the face that he wants. We discuss this issue further later.

The 3D projections of the hypercubes are displayed in the room as a pair of cubes connected to each other by lines. As the manipulable hypercube rotates in 4D, the shape of its 3D projection changes. We offset the hypercubes both horizontally and in depth. This matches the display within the Cave, which we explain when we describe 4D rotations in the Cave.

The user now has to match 3D objects. On the desktop implementation, we allow the user to rotate the view of the room 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. The mouse-dragging and the spacebar start-stop sequences, are the same as for the cube.

3D rotations in the Cave

We texture the walls of the Cave with a brick texture. Having a texture on the walls appears to improve the immersive experience. It gives a nearby background against which the foreground objects can be more easily segmented and the texture 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 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. We offset them both horizontally and in depth so that they lie on a diagonal within the Cave. This permits users to move around and view them side by side or one in front of the other while staying within the constraints of our Cave, which has no back wall.

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. The end of the virtual stick is projected to the nearest point on the surface of the tetrahedron.

EXPERIMENTAL DESIGN

Subject Pool

The users for this study came from a pool of undergraduate students who were taking a course in Honors Linear Algebra. All of the students were freshmen. Since participation was restricted to the course students and was voluntary, we expected the users to be motivated and interested in the mathematical visualization. We paid the users $10 for their participation.

A total of 18 students participated in the user study. Of these, 10 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 queried the background of each subject. We asked questions about experience in 2D and 3D interactions, about previous work in VR systems, and about prior exposure to four-dimensional objects and four-dimensional visualization programs.

Based on background surveys taken before each subject was run, we found that the Cave subjects were not well matched with desktop subjects. Accordingly, one subject was moved from the desktop group to the Cave group. This unbalanced the number of subjects, but balanced the backgrounds of the two groups.

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 record 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. The control space has unit-length sides. It is important to note that we are plotting the $\text{Error} = (1 - \text{Accuracy}) * 100$ in the graphs in this paper.

Post-experiment Questionnaire

The post-experiment questionnaire attempts to measure a subject's understanding or learning about the hypercube and its 4D rotations. It is 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 are:

- 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 to vertex A in the following hypercube.
- Find the face opposite to 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 out 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 asks the user to identify the intermediate shape that the hypercube would 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 time or accuracy measurements for a given task for all the users. Subjects performed 24 cube rotation tasks. A pilot study with two users showed that 9 tasks were sufficient for subjects to become familiar with the task and perform consistently with respect to accuracy and speed. A second set of 9 task iterations involved orientations that were constrained to the edges of the triangle control space. The final 6 task iterations involved orientations that lay within the triangle. These two different types of orientations have different difficulties and are analyzed separately.

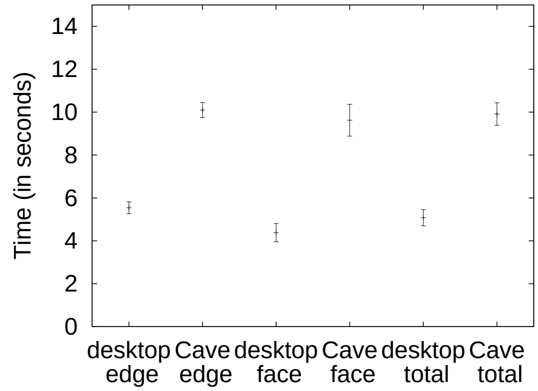

Figure 6: Time comparison for cube rotations. All desktop vs. Cave differences are significant.

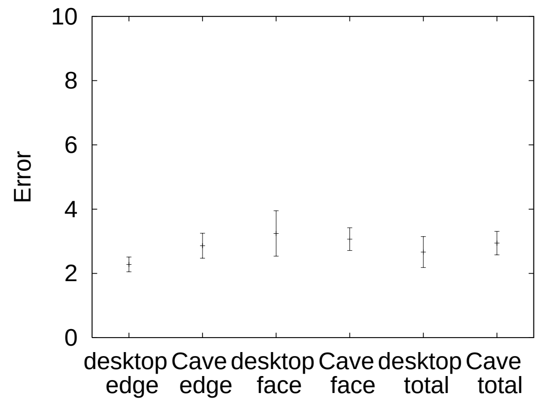

Figure 7: Error comparison for cube rotations. No desktop vs. Cave differences are significant.

Out of the 31 hypercube tasks, the initial 10 tasks were for training. Twelve tasks with orientations restricted to the edges followed, and an additional 9 tasks with orientations in the faces finished the set.

For the rest of the paper, we refer to tasks with the final orientation constrained to the edges as 'edge' tasks; and tasks with the final orientation on faces as 'face' tasks.

Rotation tasks

We see that for the cube edge and face tasks, the times are statistically significant in favor of the desktop (see Table 1 and Fig. 6). This might be because people are very familiar with using a mouse on the desktop. The desktop interaction involves dragging, 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 edge and face tasks are not statistically significant (see Fig. 7) even though the physical representation of the tetrahedron on the desktop and Cave are not of the same size. This is very

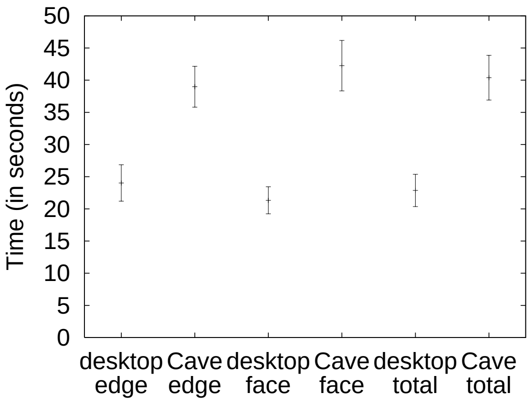

Figure 8: Time comparison for hypercube rotations. All desktop vs. Cave differences are significant.

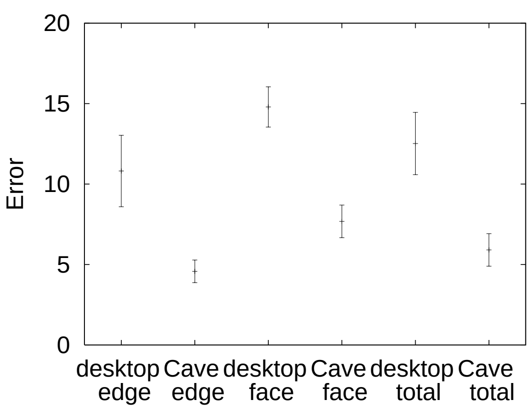

Figure 9: Error comparison for hypercube rotations. All desktop vs. Cave differences are significant.

Comparison	F	p
Cube Edge Tasks (Times)	107.2	< 0.001
Cube Face Tasks (Times)	26.55	< 0.001
Cube Tasks (Times)	96.85	< 0.001
Cube Edge Tasks (Accuracies)	1.751	> 0.2
Cube Face Tasks (Accuracies)	0.031	> 0.8
Cube Tasks (Accuracies)	0.342	> 0.5
Hypercube Edge Tasks (Times)	16.76	< 0.001
Hypercube Face Tasks (Times)	23.54	< 0.001
Hypercube Tasks (Times)	40.34	< 0.001
Hypercube Edge Tasks (Accuracies)	8.726	< 0.008
Hypercube Face Tasks (Accuracies)	18.84	< 0.001
Hypercube Tasks (Accuracies)	20.12	< 0.001

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

interesting vis-a-vis the accuracies for the hypercube edge and face tasks : even though the physical size of the tetrahedron remains the same as earlier in both of the environments, the accuracies become statistically significant in favor of the Cave (see Fig. 9). 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 than they are on the desktop. Although the desktop users have the functionality to rotate around and rock the objects slightly, that is, apparently, not as effective for estimating whether the objects are the same.

For the hypercube edge and face tasks, the times are statistically significant in favor of the desktop (see Fig. 8). 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, by contrast, 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 personal observation of the users while doing the tasks, it was apparent that in cases where the user did not know where the answer was, he found it faster to traverse the whole space on the desktop.

Cave and desktop results for the hypercube rotation tasks seem to indicate a classic speed-accuracy tradeoff. Even though both of the user groups are instructed to be as quick and accurate as possible, desktop users perform the tasks faster at the cost of accuracy. We speculate 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.

Post-experiment Questionnaire Responses

The post-experiment questionnaire results are statistically significant in favor of the Cave users. We observe that the main source of difference in the totals is due to the questions on diagram completion and counting the number of edges and vertices (see Table 2). Both these types of questions re-

Questions on	*max score*	*Desktop mean*	*Desktop std.dev*	*Cave mean*	*Cave std.dev*	*Significance t-test*
4D rotations	30	24.17	6.25	24.64	3.66	No (p=0.861)
Marking opposite faces/vertices	32	29.78	3.53	31.43	1.51	No (p=0.268)
Completing diagrams	26	17.44	3.28	22.64	2.23	Yes (p=0.003)
Number of edges/vertices	12	4.44	4.67	9.71	3.90	Yes (p=0.031)
Total Score	100	75.61	11.86	88.28	4.17	Yes (p=0.018)

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

Self Rating on	*Desktop mean*	*Desktop std.dev*	*Cave mean*	*Cave std.dev*	*t-test*
3D rotations	1.56	0.53	1.43	0.53	p=0.642
4D rotations	3.72	0.87	3.00	0.58	p=0.079

Table 3: Self Rating on Scale of 1-7

quire 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 at 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 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. The ratings are not statistically significant either way, which means that both Cave and Desktop users rated themselves equally on their understanding of the material. However, in post-experiment interviews, 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 of 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. We also obtained satisfaction ratings from the users. The ratings were on a scale of 1 - 7, in which 1 indicated that they were perfectly comfortable with the experiment and 7 meant that they disliked the experiment. The mean rating for Cave users was 1.86 and 2.33 for desktop users. The ratings were not statistically significant (t = 0.403) either way which indicates that both Cave and desktop users were equally comfortable with the experiment. We find it encouraging that novice users indicated a high level of comfort with the Cave interface.

Issues

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

Experimental Design

- Tetrahedral control space and interaction
  The representations of the tetrahedron in the Cave and desktop are not exactly the same. We could have displayed the tetrahedron as a 3D object on the desktop, but then we would have been able to see only three faces at any one point in time. In order to get to the other face, we would have needed 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 would have demanded that he rotate his mental map of orientations too, which would have complicated the task. Other options that we looked into were unfolding the faces of the tetrahedron into a star-shaped object. The problem with this, is that 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. As this can be visually confusing, we find this representation undesirable.
  The interaction technique is different for the two environments: 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. 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.
  We understand that these design decisions introduce 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 objects, relative to the environment parameters. Thus, 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 of the environments, size remains the only variable that we do not account for. Again, and as with our choices of control space and mode of interaction, 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, relative to the size of the screen. In the Cave, the objects appear to be the right size relative to the walls and the user.

- Timing Mode
  The timing data that is currently acquired from the experiment accounts only for the task interaction time. After pressing the Stop button for the last task, and before pressing the Start button for the new task, which has already been displayed, the user might decide to take some time to plan out a path. It might be interesting to record whether or not users think for different periods of time in the two environments. In practice, however, we have not seen users stop in between interactions. This behaviour suggests that users chose not to plan out a strategy beforehand. Instead, they relied on visual feedback during the interactions, to complete the tasks.

User Background

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 debatable. 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 of 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 [10] 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 striving 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. Quantitative results show that, although people perform faster on the desktop, they are more accurate in the Cave. More importantly, Cave users appear to learn more about the geometry of the hypercube and its rotations in 4-space. We believe that this work supports a more general hypothesis that the Cave is more effective than a desktop for applications requiring the learning of relatively complex geometric concepts or tasks.

ACKNOWLEDGMENTS

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. D. Banks. Interactive manipulation and display of two-dimensional surfaces in four-dimensional space, 1992.
3. 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.
4. C. Gunn. Discrete groups and visualization of three-dimensional manifolds. *Computer Graphics (SIGGRAPH'93 proceedings)*, (Annual Conference Series):255–262, 1993.
5. A. J. Hanson and R. A. Cross. Interactive visualization methods for four dimensions. In *IEEE Visualization '93 Proceedings*, pages 196–203. IEEE Computer Society, 1993.
6. 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.
7. Hannes Kaufmann, Dieter Schmalstieg, and Michael Wagner. Construct3d: A virtual reality application for mathematics and geometry education.
8. Randy Pausch, Dennis Proffitt, and George Williams. Quantifying immersion in virtual reality. *Computer Graphics (SIGGRAPH'97 proceedings)*, 31(Annual Conference Series):13–18, 1997.
9. Ivan Poupyrev, Suzanne Weghorst, and Sidney Fels. Non-isomorphic 3d rotational techniques. In *CHI*, pages 540–547, 2000.
10. Roger N. Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171, 1971.
11. 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.