Investigating the Effectiveness of Direct Manipulation of 3D B-Spline Curves Using the Shape-Matching Paradigm

Stanley B. Jang, Kellogg S. Booth, David R. Forsey, Peter Graf*
Department of Computer Science
University of British Columbia, Vancouver, British Columbia, V6T 1Z2
604-822-2218
e-mail sjang@cs.ubc.ca

ABSTRACT

B-splines are one of many spline formulations for representing smooth curves. These formulations are found in a variety of applications, including interactive curve design. Previous research has shown that the B-spline is an effective formulation for this setting. However, a possible drawback for the novice user in using the B-spline is the fact that its control vertices may lie far away from the curve, making its manipulation unintuitive. This problem is compounded in three dimensions. A direct manipulation technique, allowing a curve to be manipulated with points that lie on the curve itself, offers an alternative to control vertex manipulation. An experiment was conducted to compare the interactive design of 3D curves using control vertex manipulation of B-spline curves and a particular type of direct manipulation of B-spline curves. The results of the experiment revealed that direct manipulation was significantly faster than control vertex manipulation, without sacrificing accuracy in the shape of the final 3D curve. A general testbed designed for this investigation and related studies of 3D interaction techniques was used to conduct the experiment.

RÉSUMÉ

La B-spline est une des nombreuses formulations de spline pour représenter des courbes. Ces formulations se retrouvent dans une varité d'applications, incluant le design intéractif de courbe. Certains résultats ont démontré que la B-spline est une formulation efficace pour cet emploi. Cependant, un problème pour un usager novice réside dans le fait que les points de contrôle peuvent etre très éloignés de la courbe, rendant sa manipulation peu intuitive. Ce problème est accru en trois dimensions. Une technique de manipulation directe, permettant de manipuler la courbe avec des points résidant sur la courbe elle-même, offre une alternative à la manipulation des points de contrôle. Une expérience fut conduite pour comparer le design intéractif de courbes en 3D. Elle compare entre la manipulation des points de contrôle et un type particulier de manipulation directe. Les résultats de l'expérience ont démontré que la manipulation directe est plus rapide que la manipulation des points de contrôle, sans pour autant sacrifier la précision dans la forme finale de la courbe. Pour réaliser l'expérience, un testbed général fut conçu à partir d'études reliées aux techniques intéractives.

1. B-SPLINE CURVE DESIGN

With ever-improving computer technology fuelling the emergence of 3D interactive computer applications, there is a growing need to study and improve the interface between the human user and the computer. These changes often involve many compromises because although the needs and requirements of 2D computer interaction are well known, the needs and requirements of 3D interaction are still being discovered.

To study the different interaction techniques for 3D applications, an expandable testbed was written. The testbed allows the user to select a variety of options affecting the interaction technique, the environment and the rendering used in the display during the execution of simple 3D tasks. The performance in completing one of these tasks can then be assessed based on a number of metrics. In fact, the testbed is capable of conducting formal experiments using any of the tasks

*Department of Psychology, UBC

and options.

One of the tasks included in the testbed is the matching of spline curves. This task is similar to those used in shape-matching experiments at the University of Waterloo to compare the usefulness of different spline formulations for interactive design. Examples of such applications are font outlines, motion paths for animation cameras, and automotive and aeronautical CAD. Of course, in the actual applications, shapes are not copied or matched, but are created. However, to conduct a formal experiment comparing two techniques, subjects have to be given a concise description of the task so that their performance can be judged for correctness. Because of the difficulty in specifying the target shapes for the subjects to create, one is lead to consider the alternative task of shape-matching rather than shape-creation. The use of a shape-matching task as a standin for a shape-creation task is termed the shape-matching paradigm. It is described in a series of papers that discuss various applications of the shape-matching paradigm for the study of 2D curve manipulation techniques [4, 5 and 10].

The experiment presented here is an extension of the work performed at Waterloo using the shape-matching paradigm. This study deals with 3D spline curves. Two techniques were investigated for 3D B-spline manipulation. B-splines are approximating splines, meaning that their control vertices do not in general lie on the curve. Compared to an interpolating spline in which control vertices do lie on the curve, control vertex manipulation of a B-spline may be less intuitive. It is not always obvious which part of the curve a given control vertex affects, especially for 3D curves represented by a 2D projection on the display screen. Instead of using control vertices, a different set of manipulation points can be created, all of which do lie on the curve. Using these points, the B-spline behaves much like an interpolating spline. 3D curve manipulation using these points may be more intuitive than using the control vertices. The 3D shape-matching experiment discussed in the next section formally compares these two manipulation techniques.

A spline is a class of mathematical formulations for representing curves. A spline curve is made up of a number of *curve segments* each of which is represented by a set of polynomials (often cubic). The coefficients of the polynomials are determined by a basis matrix (a set of blending functions) and a vector of geometric constraints (called a geometry vector). These constraints are often 3D points called *control vertices*. The blending functions are unique for a given spline formulation. The equation for a curve segment $Q(t)$ is $Q(t) = T \cdot M \cdot G$ where T is a vector of polynomials in a parameter t, M is the basis matrix and G is the geometry vector. Expanding this product for the case of cubic polynomials gives:

$$
\begin{aligned}
Q(t) &= [x(t)y(t)z(t)] \\
&= [t^3 t^2 t 1] \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{pmatrix}
\end{aligned}
$$

The product $T \cdot M$ gives the set of blending functions for the spline which can be written explicitly as the vector $[b_1(t), b_2(t), b_3(t), b_4(t)]$. Each of the elements G_i is a vector describing the 3D location of a control vertex.

Explicitly multiplying $T \cdot M$ and then using the summation formula for matrix multiplication yields an alternative representation for the curve as a weighted sum of basis functions

$$Q(t) = \sum_{i=0}^{4} b_i(t) \cdot G_i$$

where the vector-valued $Gi = (x_i, y_i, z_i)$ are the 3D weights for the basis functions b_i.

Direct manipulation is a term coined by Shneiderman [11] almost a decade ago. He used it to describe computer interfaces that were easy to learn and master, were intuitive for the novice and experienced computer user, and were enjoyable to use. More recently, the term direct manipulation has been used to describe the technique of manipulating a spline curve by picking any point on the curve, dragging it to a new location, and having the curve pass through the new point [2]. In this paper, direct manipulation refers to a similar technique where a finite number of points on the curve are chosen for manipulation. The points on the curve which have been chosen for the direct manipulation points are the joints of the curve. These points on the curve are determined by only three control vertices because they lie at the ends of curve segments. The following equation computes the location of a direct manipulation point (D_i) from the corresponding B-spline control vertices (G_{i-1}, G_i, G_{i+1}).

$$D_i = \tfrac{1}{6}G_i + \tfrac{4}{6}G_i + \tfrac{1}{6}G_i$$

These coefficients are determined from the basis functions. A similar direct manipulation formulation for surfaces was introduced by Forsey [8]. For background information on splines, refer to an introductory text on splines [1]. For further details on direct manipulation of B-splines, refer to Bartels and Beatty's conference paper on direct manipulation [2] or Jang's Masters thesis [9].

2. AN EXPERIMENT TO COMPARE DIRECT MANIPULATION WITH CONTROLLED VERTEX MANIPULATION

A curve-matching experiment comparing the effectiveness of direct manipulation of 3D B-spline curves with that of control vertex manipulation was conducted to test the hypothesis that direct manipulation is superior for 3D curves. Control vertex manipulation is indirect because these points may potentially lie very far away from the spline curve itself. When working with 3D curves represented on a 2D display, this indirection could be a major barrier to overcome in successfully manipulating the curves because the 2D projection sometimes makes it difficult to detect the relationship between the 3D control vertices and the segments of the 3D curve. The experiment environment was similar to that used in previous shape-matching experiments. The subject was isolated in a dimly lit room with a Silicon Graphics Iris 4D workstation, model 240VGX. Although it was connected to a network, the subject was the sole user during the experiment. The mouse was fixed directly in front of the subject and the keyboard was placed between the mouse and the monitor. Sixteen subjects took part in the experiment, twelve of whom were male. Nearly every subject had computer and mouse experience.

During each trial, the subject was presented with two 3D B- spline curves, one blue and one red, drawn inside the same shaded 3D bounding box that filled the entire screen. All of these objects were drawn using a perspective projection. The blue curve remained fixed throughout the trial and was called the target curve. The shape of the red curve could be changed, and was therefore called the controlled curve. The controlled curve could be manipulated by moving any of its three control points, represented as small black cubes. If the trial was part of the direct manipulation session of the experiment, the control points would all lie on the controlled curve. If it was a control vertex session, they would usually not. The task during each trial was to modify the controlled curve so that it matched the target curve as closely as possible in a short amount of time. Because the curves were three dimensional, it was necessary to view and manipulate the curves in different orientations to complete the match.

To move a control point, the subject would place the cursor over the desired point, press and hold down the left mouse button, and then drag the cursor, moving the control point along with it. To change the orientation of the curves, the subject was given the ability to rotate the curves by ninety degrees to the left, right, up or down using the middle mouse button. Depending on which region of the screen the cursor was in when the button was released, the curves would rotate in one of the four directions. The rotation was not instantaneous, but was animated over one second. This helped the subject visualize the shape of the curves by allowing him to see the curves in motion. The right mouse button was used to end the trial.

During the course of a trial, the subject was required to do the following: select a START button on the screen which starts the timer and displays the curves for matching; match the curves; stop the timer by picking the appropriate item from a pop-up menu (quit or cancel); and, finally, give a subjective rating of the match. Subjective ratings were implemented in the original version of the experiment software and although rating data was collected in this experiment, no analysis was performed on it. The subject was allowed to pause as long as desired between trials. Figure 1 illustrates the execution of a trial.

Both the target and the controlled curves were defined by B-splines with nine control vertices. The position of each control vertex for the target curve was randomly generated. There were twenty target curves used in the experiment. After the set of target curves was created, the curves were screened for their level of difficulty, but none were rejected. The four practice target curves were not randomly generated, but were carefully chosen so that the initial practice trial was straightforward, with each successive practice trial gradually becoming more difficult.

One difficulty in earlier 2D shape-matching experiments was the matching of the endpoints of the target curves. To simplify the matching of the 3D curves for this experiment, it was decided that the endpoints of each controlled curve would be permanently attached to the target curve so that only the middle section needed to be manipulated. To achieve this, the three control vertices at either end of the controlled curve were assigned to the corresponding control vertices in the target curve, and thus were not modifiable by the subject during the trial. With this constraint in place, each trial involved matching the three middle control points of the controlled curve to those of the target curve, leaving the other six control points unchanged. Only the three manipulated control points of the controlled curve were displayed, so the manipulation had to be performed using only visual information about the shape of the target curve and the controlled curve, not information about the position of the control points of the target curve.

The experiment was divided into two independent sessions for each subject. One session dealt with B-spline curves using standard control vertex manipulation while the other dealt with B-spline curves using direct manipulation. The order in which the two sessions were performed was random, with half of the subjects doing the sessions in each order. Each session comprised an on-line tutorial (which was optional for the second session), twenty-four trials that were separated into four initial practice trials and twenty recorded trials, and a short subjective rating session. At the end of each session, subjects were given a comments form to fill out. Refer to Jang's Master's thesis [9] for further details on the experiment environment and procedure.

3. EXPERIMENTAL RESULTS

For this curve-matching experiment, the following three hypotheses were investigated:

H1. Direct manipulation is better than control vertex manipulation for 3D curve-matching. H2. Learning has a major effect on the performance of this task. H3. Direct manipulation can be learned faster than control vertex manipulation.

Results from the experiment were collected from statistical analyses of the time and error data and subject comments forms. Hypothesis 1 was verified. Direct manipulation was faster than control vertex manipulation with similar levels of accuracy. Hypothesis 2 was also verified, and was in fact the strongest effect in the curve-matching experiment. Subjects improved with each trial using both techniques. There was no statistical evidence for Hypothesis 3. Subjects improved in match time at similar rates using both techniques with similar levels of accuracy.

The majority of the analysis looked at the time and error data from the experiment. Two types of analysis were used, the t-test and the analysis of variance (ANOVA). Literature on these tools can be found in any introductory statistics text [7]. An α value of 0.05 or lower was used to decide if a result was *significant*. When reporting significant results using the t-test, the two means will be given followed by their respective standard deviations in parentheses.

The experiment results can be broken up into four blocks upon which the analysis was based. These included control vertex sessions performed as first sessions, control vertex sessions performed as second sessions, direct manipulation sessions performed as first sessions, and direct manipulation sessions performed as second sessions. These blocks are later referred to as CV1, CV2, DM1, and DM2 respectively. Each of these blocks consisted of twenty trial results from eight subjects. For a given subject, the twenty trials involved matching a fixed set of twenty target curves given in a random permutation. Table 1 summarizes the results by blocks with the median times and median errors.

Two measures were used to gauge a subject's performance for a trial, the time to complete the trial and the error in the match. The error was the sum of the Euclidean distances between corresponding control vertices on the two curves. This metric is based on the mathematics behind the splines as opposed to one based on what is viewed by the subject on the monitor. Poor matches are represented by a large error and good matches are represented by an error value close to zero. A perfect match, although not necessarily achievable due to the discrete computations in hardware, has an error of zero. These distances were transformed from units in the control vertices' world coordinates to screen pixels by an approximate scale factor to make the error values more meaningful.

During the analyses, the raw data was averaged in one of two ways depending on whether a trial analysis or a subject analysis was being performed. In a subject analysis, the average for each subject over all trials in a block was calculated and that data was used in the analysis. Because there were eight subjects in a block, the subject analysis dealt with eight values per cell. For a trial analysis, averages over all eight subjects for each of the twenty trials were used. Trials are labelled by their chronological order for each subject, taking into account the practice trials. For an item analysis, averages over all eight subjects for trials containing each target curve were used. There were twenty different target curves so twenty values for each block were used in this analysis. In most cases, grouping the data in each of these ways produced similar results unless there were outliers in the data. Thus, comparing these results helped reveal possible outliers.

There are a number of techniques for averaging a set of data. The mean is the most common measure. However, it is not very useful when the data contains outliers (extreme data points). The mean should especially be avoided when the data is bounded on one side, as is the case with time data, because outliers are certain to bias the mean in only one direction. Two other averaging measures, the median and the mean of the log transform of the data, are less affected by outliers. For this experiment, the median was chosen over the mean of the log transform because transforming data is more complicated when reporting results of means and confidence intervals and the median analysis did provide a satisfactory analysis.

Testing Hypothesis 1 - Analysis of Spline Manipulation Technique

To determine which technique performed best in this experiment, the t-test was used to analyze the time and error data. Because of the possible learning factor involved in the experiment, the analysis compared CV1 and DM1 and ignored CV2 and DM2. A trial analysis of time was significant with $\alpha = 0.01$ and a t statistic of 2.99 ($t_{.005,38} = 2.71$). The mean values for CV1 and DM1 were 161 (std dev 21.7) sec. and 140 (21.0) sec., respectively. A subject analysis was also performed, but because of large variances between subject scores, not enough data was available to produce a significant result. A trial analysis of the error generated a significant result with $\alpha = 0.002$ and t statistic 3.51 ($t_{.001,38} = 3.32$) and mean errors of 13.0 (3.36) and 9.5 (2.8) pixels for CV1 and DM1. So, direct manipulation was superior for subjects with no prior experience in performing this task, both in terms of the time to achieve a match and the accuracy of the match.

An alternative question is: which technique is better for experienced subjects? For this analysis, experienced is defined as having previously matched curves with either technique, so a comparison between CV2 and DM2 was made. A trial analysis of time was significant with $\alpha = 0.01$ and t statistic 2.92 ($t_{.005,38} = 2.71$) with respective mean times of 125 (16.7) sec. and 110 (15.0) sec. Analysis of the error did not produce significant results. Again, direct manipulation proved to be the better technique, but only in terms of the time required to achieve a match.

Testing Hypothesis 2 - Analysis of Learning Effects

To study the effects of learning, t-tests were performed on CV1 vs. CV2, DM1 vs. DM2, the first ten trials of CV1 vs. the last ten trials of CV1, and similarly for CV2, DM1, and DM2. Also, graphs of the data with regression analyses were produced. For CV1 vs. CV2, the trial analysis of time was significant with $\alpha = 0.001$ and t statistic 5.71 ($t_{.0005,38} = 3.57$) with mean times of 161 (21.7) sec. and 125 (16.7) sec. A subject analysis of time was significant with $\alpha = 0.05$ and t statistic 2.47 ($t_{.025,14} = 2.15$). A trial analysis of error was significant with $\alpha = 0.001$ and t statistic 4.53 ($t_{.0005,38} = 3.57$), and mean errors of 13.0 (3.36) and 8.9 (2.10) pixels. For DM1 vs. DM2, a trial analysis of time was significant with $\alpha = 0.001$ and t statistic 5.11 ($t_{.0005,38} = 3.57$) with mean times of 140 (21.0) sec. and 110 (15.0) sec. A trial analysis of error showed no significant results.

Similar analysis was performed on the first half of a block vs. the second half of the block. None of the blocks produced significant results except for CV1. For that analysis, significant results were found for both the trial analysis of time and of error for $\alpha = 0.05$ and t statistics 2.38 and 2.21 repspectively ($t_{.025,18} = 2.10$). The mean times for the first and second half were 172 (22.7)sec. and 151 (14.1) sec., while the mean errors were 14.6 (3.17) pixels and 11.5 (2.80) pixels.

These results show that the learning factor is present across sessions and may in fact be stronger than the technique factor in Hypothesis 1. The differences in mean times for CV1 vs. CV2 and DM1 vs. DM2 were almost double those in the technique analysis. Within the blocks, significant effects in learning were only found in CV1.

The equation for a regression line provides information about the amount of learning and the level of difficulty of the task. A large y-intercept indicates a difficult task requiring more time to complete or a higher level of error. A large negative slope represents rapid learning or improvement. Figures 2 and 3 show the regression lines for the mean trial times and mean trial errors respectively. To analyze the significance of these lines, hypothesis tests were performed to determine if the null hypothesis of $slope$=0 could be rejected. Only the median trial error for CV1 produced a significant result, with = 0.05 and a 95confidence interval for the slope of (-0.47,-0.05). Interestingly, the lines for CV1, CV2 and DM1 in Figure 2 have similar slopes leading one to hypothesize that rates of learning in those blocks were equivalent. The regression line for DM2 has essentially zero slope meaning that there was very little learning or improvement. The y-intercepts for both control vertex blocks were larger than the direct manipulation blocks, showing that using control vertices is more difficult to start with. Looking at Figure 3, the error was fairly constant in CV2, DM1 and DM2. In CV1, errors were very high initially, but improved to the levels of the other blocks.

Testing Hypothesis 3 - Interaction between Learning and Technique

A two factor ANOVA was performed to study the interaction between the technique factor and the learning factor. For the trial analysis of time, both the technique and the learning factor were significant for $\alpha = 0.01$, with the result for the learning factor being much stronger. For the subject analysis of time, only the learning factor was significant at that level of α. In both analyses, there was no evidence of interaction. So, for the time data, the two factors were independent. Learning affected the performance in using the two techniques in similar ways and vice versa. In contrast, for the trial and subject analysis of error, the technique and learning were not significant, but the interaction factor was. Learning affected the error of one technique, control vertex manipulation, more than it did direct manipulation. In fact, in comparing the plots in Figure 3, the median error for each trial did not change appreciably from DM1 to DM2.

Comments from Subjects

Subjects were given three questions in which they were asked for a preference between the two curve manipulation techniques. The questions and the results are summarized below:

Q1. In which session did you find the control of the curves easiest?
Q2. In which session did you think that you were the most successful in your final matches?
Q3. Which session did you enjoy the most?

For the first and third question, direct manipulation was the overwhelming favourite. The score for the second question was almost even because most subjects were equally successful with all of their matches. The difficulty of the trial, whether due to the technique or not, was reflected in the match time not the match quality.

In general, trial performance was a trade-off between match time and accuracy. In most cases, the trials were not extremely demanding, so a fairly constant, high level of accuracy was maintained. Any effects due to learning or difficulty of the target curve were observed in the trial time alone. For some subjects, however, having little curve-matching experience and faced with the less intuitive control vertex manipulation, early trials in CV1 did prove to be extremely demanding. In these cases subjects may have been forced to compromise the desired level of accuracy in favour of a reasonable match time.

4. A TESTBED FOR 3D INTERACTION EXPERIMENTS

The software used for the direct manipulation experiment is actually a subset of a larger program with an abundant list of options including depth cues and parameters to customize the 3D task being studied. This source code is a re-working of Ruest's experiment software [10]. The modification of his software to work with 3D curves was fairly easy because the Silicon Graphics library routines (GL) work in a 3D world by default. However, the revision of the user interface to handle 3D curve-matching generated many questions to which there were many possible answers. For example, how should 3D movement of the control points be handled using the mouse as the input device? Instead of just offering one interface to the curve-matching task, a number of distinct interfaces were implemented. In doing so, their effectiveness for this particular 3D task could be tested first hand. Some of these interfaces were well- suited for the curve-matching task, while others appeared to be more effective for other 3D tasks.

The effort in modifying the source code to work with 3D splines is a good example of the difficulty in implementing a 3D application using an input device originally designed for 2D applications. Although mathematically 3D splines are not much more complicated than 2D splines, the interface allowing the curves to be manipulated requires much work. The interface is more elaborate for two reasons. First, in 2D there are but two degrees of translation and only one degree of rotation, while in 3D there are three degrees of translation and three degrees of rotation. Second, the mouse and the CRT display, the standard input and output devices for a workstation, are both inherently 2D devices. This hardware is fine for 2D spline manipulation, but not immediately adequate for 3D splines. The testbed can be used to explore interaction techniques using these devices for 3D tasks.

Besides the different 3D interface options, the environment and the specific task have a number of options to change their properties. Many aspects of the program can be changed either from a pop-up menu, command-line arguments, or a parameter file, making the software quite versatile. For example, the thickness of the curves or the speed of the mouse can be changed from a menu rather than updating a constant in the source code and re-compiling. With such versatility, the software is very useful for the design of future experiments. A researcher can easily compare a variety of techniques and environments without rewriting parts of the source code. In fact, an objective comparison is possible because match evaluation routines and other software for running an actual experiment are already available. For the same reason, when the design of the experiment has been completed, preparing the software for the formal experiment requires minimal work.

5. FUTURE WORK

Three major contributions were made in this research. First, a prototype testbed for the study of 3D interaction techniques was written. Using this software, a number of basic 3D tasks can be performed using various combinations of the interaction techniques. Second, additional experience was gained in the design of a 3D curve-matching experiment using experiment software similar to that used in previous 2D work. Finally, an experiment was conducted to compare direct manipulation of B-spline curves with control vertex manipulation of B-spline curves. The results provided evidence that direct manipulation is the better technique.

This experiment was the first of the curve-matching experiments dealing with 3D splines. At the same time, attention has shifted towards the basic 3D interaction techniques used as building blocks for the higher-level spline manipulation. Further research in this area includes both enhancements to the testbed software and additional experiments. The software will be expanded to use other input devices and run different 3D tasks. It will also be generalized to simplify the implementation of such modifications. The list of future experiments to be performed is extensive, involving both curve-matching and other 3D tasks. Some of the more immediate studies proposed are:

- Running a similar experiment with "shadows" of the control points and curves orthogonally projected onto each of the four side walls of the box enclosing the curves. Besides providing depth information, the shadows of the control points can be picked and dragged along the wall, forcing the actual control point to move in response. Using this interface, the subject could conceivably complete the entire match in 3D without changing the orientation of the curves. Essentially, five views would be displayed on the screen at all times, four of which would be at reduced resolution as shadows.
- Representing the curves as smooth, generalized cylinders and applying specular reflection to them. Specular reflection should provide a very good curvature cue, again possibly reducing the need to change the orientation of the curves.
- Using the ADL-1 Head Tracker to change the view of the curves. Changing the view with head movement should be more natural than changing the orientation with the mouse. The subject's hand that is used to control the mouse is relieved of a task, leaving it with the sole job of manipulating control points, possibly reducing some confusion. Unfortunately, the head tracker does not allow more than about forty-five degrees of rotation in any direction. It is not known whether this amount is adequate for matching the curves in depth. The following related experiment should be conducted as a preliminary study.
- Investigate how the amount of rotation allowed by the ADL-1 Head Tracker affects the accuracy in depth of a simple task. This task could be to place a point at the mid-point of a line segment (not actually drawn) connecting two fixed points, or to place a point at the center of a cube. In this experiment, subjects would perform the task given varying amounts of rotation. As an initial prediction, the amount of rotation allowed should correlate positively with the accuracy in depth. Deering has investigated similar tasks and found that a head-tracked display provides very good depth cues [6].
- Using a Spaceball or other 6-D input device to change the orientation of curves and/or manipulation of control points [12]. As with the head tracker, the mouse is freed of the task of changing views. Instead, the subject would use his left hand to control the Spaceball. Capable of three degrees of rotation, the Spaceball should be very intuitive to use for this application. These experiments might involve both mixed-device strategies (mouse and Spaceball) and single-device strategies (mouse or Spaceball alone).

All of these extensions are easy to include in the existing testbed, although it is anticipated that a re-design of the testbed will at some point be required after more experience is gained with it.

6. ACKNOWLEDGEMENTS

The research reported here was supported in part by the Natural Sciences and Engineering Research Council of Canada under various operating and strategic grants and by the University and Province of British Columbia. Paul Ruest wrote the 2D version of the shape-matching software and Richard Bartels provided comments on the extensions to 3D shape-matching and the direct manipulation formulation for B-splines.

7. REFERENCES

1. Bartels, Richard H., John C. Beatty, and Brian A. Barsky, An Introduction to Splines for Use in Computer Graphics & Geometric Modelling, Morgan Kaufmann, Inc., 1987.
2. Bartels, Richard H., and John C. Beatty, A Technique for the Direct Manipulation of Spline Curves, Graphics Interface Conference Proceedings, 1989, p. 33-39.
3. Bartels, Richard H., John C. Beatty, Kellogg S. Booth, Eric G. Bosch, and Pierre Jolicoeur, Experimental Comparison of Spline Bases Using the Shape-Matching Paradigm, ACM Transactions on Graphics, 1993. To appear.
4. Bosch, Eric G., Workstation-Based Shape Matching Experiments, Masters Thesis, University of Waterloo, 1987.
5. Bosch, Eric G., Richard H. Bartels, Kellogg S. Booth, Pierre Jolicoeur, Workstation-based Shape Matching Experiments, 2nd IEEE Conference on Computer Workstations, 1988, p. 132-141.
6. Deering, Michael, High Resolution Virtual Reality, Computer Graphics, SIGGRAPH Conference Proceedings, 1992, Vol.26, Num. 2, p. 195-202.
7. Devore, Jay L., Probability and Statistics for Engineering and the Sciences, Brooks/Cole, 1982.
8. Forsey, David R., Richard H. Bartels, Hierarchical B-Spline Refinement, Computer Graphics, SIGGRAPH Conference Proceedings, 1988, Vol.22, Num. 4, p. 205-212.
9. Jang, Stanley, 3D Interaction Studies Using the Shape-Matching Paradigm, Masters Thesis, University of British Columbia, 1992.
10. Ruest, Paul, An evaluation of tension within an extensible spline testing facility, Masters Thesis, University of Waterloo, 1989.
11. Shneiderman, Ben, Direct Manipulation: A Step Beyond Programming Languages, IEEE Computer, 1983, Vol. 16, Num. 8, p. 57-69.
12. Thalmann, Daniel, Andre LeBlanc, Prem Kalra, Nadia Magnenat Thalmann, Sculpting with the "Ball and Mouse" Metaphor, Graphics Interface Conference Proceedings, 1991, p. 152-159.

Results	CV1	CV2	DM1	DM2
Median Time (sec) / Std. Dev.	161 / 21.7	125 / 16.7	140 / 21.0	110 / 15.0
Median Error (pixels) / Std. Dev.	13.0 / 3.4	8.9 / 2.1	9.5 / 2.8	10.3/ 2.5

Table 1: Time and error results: medians determined for each of the 20 trials, then means of medians computed.

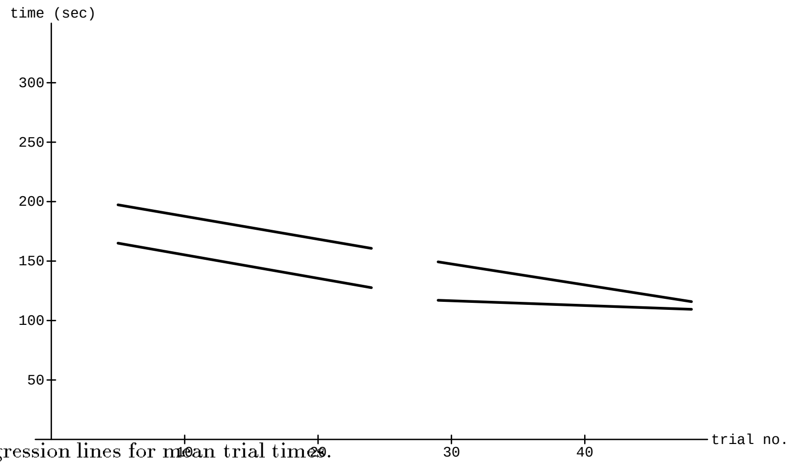

Figure 1: Regression lines for mean trial times.
($CV1: y = 207 - 1.93x; CV2: y = 200 - 1.76x; DM1: y = 175 - 1.97x; DM2: y = 129 - 0.40x$)

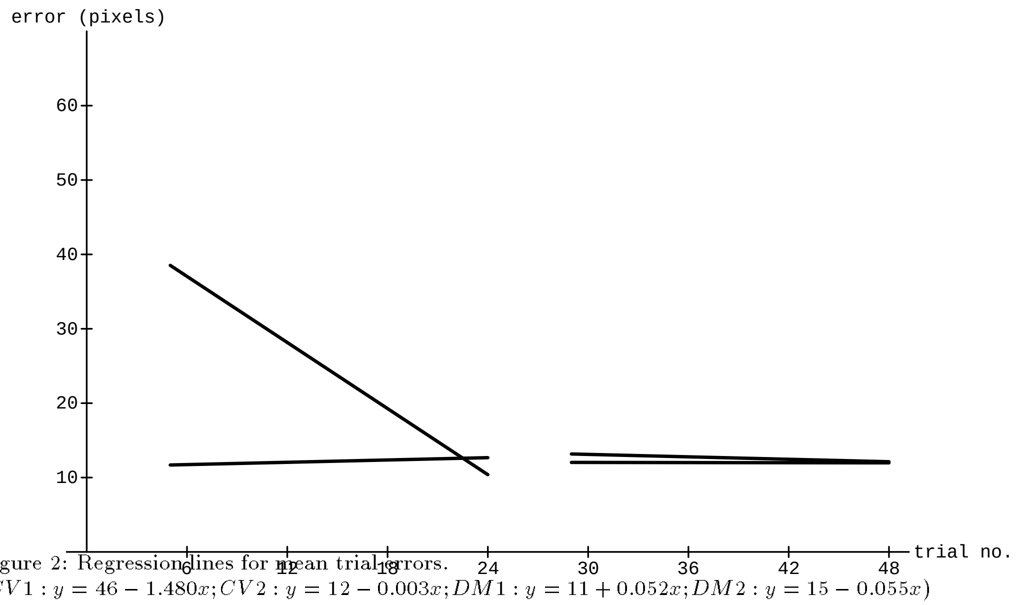

Figure 2: Regression lines for mean trial errors.
($CV1: y = 46 - 1.480x; CV2: y = 12 - 0.003x; DM1: y = 11 + 0.052x; DM2: y = 15 - 0.055x$)