

symSpline: Symmetric Two-Handed Spline Manipulation

Celine Latulipe, Stephen Mann, Charles L. A. Clarke and Craig S. Kaplan
David R. Cheriton School of Computer Science
University of Waterloo
{clatulip, smann, claclark, cskaplan}@uwaterloo.ca

ABSTRACT

We introduce symSpline: a symmetric, dual-mouse technique for the manipulation of spline curves. In symSpline, two cursors control the positions of the ends of the tangent to an edit point. By moving the tangent with both mice, the tangent and the edit point can be translated while the curvature of the spline is adjusted simultaneously, according to the length and angle of the tangent. We compare the symSpline technique to two asymmetric dual-mouse spline manipulation techniques and to a standard single-mouse technique. In a spline matching experiment, symSpline outperformed the two asymmetric dual-mouse techniques and all three dual-mouse techniques proved to be faster than the single-mouse technique. Additionally, symSpline was the technique most preferred by test participants.

Keywords

two-handed interaction, splines, spline manipulation, symmetric interaction

Introduction

Drawing a desired spline curve is a difficult task that requires considerable skill. There are a number of problems with traditional methods of spline manipulation that make these methods challenging for novice users and inefficient for expert users. The first problem is related to the variety of ways a user can manipulate a spline: by moving control points, by moving edit points, by adjusting the slope of a tangent to an edit point, or by adjusting the length of a tangent to an edit point. Deciding which manipulation to perform first is difficult: should the edit point be translated, and then the slope of the tangent adjusted? Or should it be done in the opposite order? This thought process is time consuming for the novice user. The second problem is that the result of the chosen manipulation is often far from what is expected by the novice user. The third problem with traditional one-handed spline manipulation is that it is time-multiplexed. Users can only perform one type of spline manipulation at a time and some type of mode switch is required to change the editing tool.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2006, April 22-28, 2006, Montréal, Québec, Canada.
Copyright 2006 ACM 1-59593-178-3/06/0004...\$5.00.

The mode switch is both distracting and time-consuming, whether achieved by pressing keyboard modifier keys or by clicking on a toolbar button or selecting a menu option. In this paper, we introduce a bimanual (two-handed) symmetric interaction technique, symSpline, which allows users to explore and adjust curves in a fast and intuitive manner.

A dual-mouse system allows for a space-multiplexed style of manipulation where the spatial input from the two mice can be used for more than one type of spline editing at any given time. One possible implementation of such a system would be to let one mouse control the translation of a spline edit point, while the other mouse controls the tangent to that point. This approach could follow the guidelines set out by Guiard in his seminal work on asymmetric interaction [?]. Guiard's guidelines have been used extensively, as many tasks can be characterized as asymmetric: the non-dominant hand sets the frame of reference in which the dominant hand does the detail work.

We developed the symSpline method because we believe that there is a natural symmetry to be exploited in curve editing. Following Guiard, we characterize interaction as *symmetric* whenever the two hands work together at similar levels of spatial and temporal resolution. We have created a symmetric curve manipulation technique where the two mice control the ends of the tangent to the selected edit point. The symmetry of this design allows a user to perform point translation, tangent scaling and tangent rotation all in a single, symmetric gesture (see Figure ??). This means that novice users need not spend time deciding what type of manipulation to perform and expert users can avoid the cost of switching modes. Thus, the symSpline technique affords easy exploration of the effects of the different types of manipulation. By testing the symSpline technique with a group of novice participants, we show that it is significantly more efficient and more natural than previous approaches, at least in the spline matching task we tested.

To understand how symSpline fits into current graphics and user interface research, we describe some related interfaces in the following section. The Experimental Design section describes the experiment used to test symSpline against both two-handed asymmetric techniques and a traditional one-handed technique and includes results and discussion. A

Figure 1: symSpline uses two cursors to control the ends of the tangent to an edit point. By translating the whole tangent, the edit point is translated and the curve changes appropriately.

summary of the main contributions and a description of the planned follow-up work conclude the paper.

Background

Guiard characterizes asymmetric interactions as those in which the non-dominant hand sets the frame of reference in which the dominant hand works [?]. Additionally, the action of the non-dominant hand precedes that of the dominant hand. The level of detail, both spatial and temporal, at which the dominant hand works is finer than that of the non-dominant hand. Handwriting is an example of an asymmetric two-handed activity: the non-dominant hand positions and holds the paper while the dominant hand writes. Many everyday activities can be characterized as asymmetric, explaining the importance of Guiard's work.

Early psychology results in the area of two-handed cooperation [?, ?] led many interface researchers to believe that the two hands cannot work together at a similar level of spatial or temporal detail because of interlimb motor interference. However, recent findings have shown that the two hands can in fact work together symmetrically [?, ?, ?]. These results suggest that problems of interlimb interference are associated with the planning and perceptual organization of the task being performed and may have nothing to do with motor capabilities. Latulipe discusses the implications of these results [?] and argues that two-handed symmetric interaction in the computer interface has great potential, provided the task being performed perceptually unifies the action of the two hands.

Two-handed geometric transformations

Previously studied techniques for two-handed geometric transformations have often followed the Guiard guidelines. The prototype rotation and scale tools described by Fitzmaurice et al. used two bricks to control transformations [?]. Although the two devices were symmetric in functional capability, the techniques used for rotation and scale were asymmetric, requiring that one brick act as an anchor or point of rotation, while the brick controlled by the dominant hand adjusts angle or stretch. The authors did not indicate whether they allowed users to translate while rotating or scaling. Shaw's THRED [?] system also follows Guiard's guidelines for asymmetric interactions.

Many commercial graphics applications use a single mouse with modifier keys on the keyboard to allow users to switch modes during complex operations such as spline manipulation. This approach is the typical time-multiplexed input technique: a single mouse can do only one spatial task at a time, and so the modifier keys serve the purpose of specifying the temporal divisions between separate spatial tasks. An even less efficient alternative is the use of toolbar buttons

to switch modes. This is less attractive because two spatial sub-tasks are required to switch modes: the user has to move the cursor to the button, click and then move the cursor back to the object being manipulated. The benefit of the keyboard approach is that with single-mouse systems, the user's non-dominant hand generally rests on or near the keyboard, which makes the mode switch relatively fast compared to pressing toolbar buttons.

Symmetric interaction

A few studies have shown the potential benefits of symmetric interaction techniques. Casalta et al. found that symmetric techniques performed well for rectangle creation [?] and Leganchuk et al. found symmetric techniques performed well for area sweeping tasks [?]. Balakrishnan and Hinckley [?] studied a symmetric target-following task in which users had to track two different targets around the screen using two cursors. They measured both the level of parallelism and the level of symmetric interaction exhibited by participants. They found that reducing the amount of visual integration between the two tracked objects led to degradations both in the symmetry of action and in the level of parallelism. They concluded that objects being manipulated in a symmetric interaction should be visually connected. The amount of symmetric interaction was not affected by dividing attention between the targets (*i.e.*, separating them further on screen) or by increasing the task speed. However, these two factors did decrease the amount of parallelism exhibited by users. In our symSpline technique, both hands are manipulating the same object (the tangent) and so Balakrishnan and Hinckley's visual integration recommendation is met.

One notable example of research that takes advantage of symmetric interaction is the symmetric deformation technique found in the Twister implementation for 3-D object manipulation by Llamas et al. [?]. They describe the ability to stretch and deform an object using symmetric interaction with two trackers. However, the emphasis in that research is on the mathematical models used for deformation of 3-D objects, rather than on the symmetry of the interaction.

Kurtenbach's T3 system [?] includes many techniques that follow Guiard's guidelines for asymmetric interaction, as well as a symmetric shape-drawing and manipulation technique called 'two-handed stretchies'. However, they did not describe this interaction in detail or perform any evaluation of the symmetric interaction.

Latulipe et al. evaluated symmetric shape-manipulation techniques for object placement, involving simultaneous rotation and translation [?] and for image alignment, involving simultaneous rotation, scale and translation [?]. In both studies

they found that the symmetric techniques significantly outperformed the asymmetric and single mouse techniques.

Two-handed spline manipulation

Direct manipulation of splines is not new. The technique was first introduced in 1989 by Bartels and Beatty [?]. The technique was refined and tested to include the ability to adjust the frequency of a spline segment by increasing the length of the tangent at the selected point [?, ?]. Since that time, numerous researchers have used the method and applied it to 3-D surfaces and to specific modelling problems. Finkelstein and Salesin [?] made important contributions by addressing issues of locality of control and preservation of detail, while allowing direct manipulation of splines. These techniques were asymmetric and made use of a single mouse, sometimes with the addition of keyboard controls. Fowler made use of a dataglove for direct manipulation of splines, which allowed 3D control of tensor product surfaces [?]. Although this was an important contribution, the manipulation was asymmetric and required a specialized input device.

Figure 2: An example from the spline manipulation of Owen et al. The participants had to manipulate the spline by manipulating the pink (square) and blue (circle) control vertices. Shapes used for illustration purposes only.

Owen et al. used spline manipulation as a test case for examining the differences in performance between bimanual and unimanual interaction techniques [?]. They were also interested in the difference between integrated and separate workspaces for the two hands in bimanual interaction. Their experiment did not involve direct manipulation of splines; participants were asked to match splines by moving control vertices shown as pink and blue squares in Figure ?? . In addition, they used asymmetric input devices: a stylus in the dominant hand and a puck in the non-dominant hand. They found the two-handed techniques to be faster than the unimanual technique. They found no difference in performance related to whether the devices used separate workspaces (each using a specified half of a drawing tablet), or an integrated workspace (sharing the whole drawing tablet). The two-handed interaction technique was symmetric in that the task for each hand was to drag a control point around. However, the spline was not visually integrated, and so the user would consider each hand to be performing a separate task of moving a control point.

Recently, there have been two applications of two-handed constraint-based direct manipulation of splines, both targeted at the automotive design field. The first and most notable was the work on Digital Tape Drawing by Balakrishnan et al. [?]. In real tape drawing, the user's two hands asymmetrically apply electrical tape to a large wall display to create curves. To edit a curve, the tape is unstuck and reapplied at a differ-

ent angle. The digital tape drawing system used two trackers and a large digital wall display. The method was not designed to follow Guiard's guidelines for asymmetric interaction; instead, it mimicked the way traditional tape artists worked with the real tape. This work showed the benefit of using both hands for the creation of digital splines as well as the importance of carefully studying the real-life interaction technique before implementing a digital version.

Inspired by the digital tape drawing interface, Bae et al. used two graspable handles to do fine-tuning of spline curves drawn on a large scale wall display [?]. Their interaction technique was asymmetric and follows Guiard's guidelines. They used specialized circular devices and a language of gestures to specify spline manipulation operations.

Both of these large scale wall techniques have demonstrated the benefits of using two hands for spline manipulation, but have an audience limited to automotive design engineers. Our work aims to scale this idea down to the desktop, with a regular display and with regular mice, rather than specialized devices. Two-handed spline manipulation could then be used by a more general audience including artists, animators and everyday computer users. Finally, our method differs from the one used by Owen et al. in that we allow direct manipulation of the spline curves and provide full visual integration between the cursors controlled by each hand.

The symSpline technique

Our method of spline manipulation requires two standard USB mice and two cursors. The availability of spatial information from two mice allows multiple spatial tasks to be completed at once (*i.e.*, space-multiplexed input). By using the two mice to move the ends of the tangent to a point on a curve, symSpline combines the four degrees of freedom available to simultaneously translate, stretch and rotate the tangent.

The symSpline technique affords curve scaling in a manner that mimics the way people use their hands symmetrically to stretch objects in real life. Additionally, the tangent in symSpline is pushed around the screen by the two cursors similar to the way a bike is steered by the two hands holding the ends of the handlebar. This steering metaphor is effective because it mimics the way people naturally move large objects in space.

The symSpline technique is implemented inside of a prototype two-handed graphics program. The curves that can be manipulated with symSpline are endpoint-interpolating cubic B-splines. The splines can only be manipulated directly, not by moving control points. The algorithm used for the constraint-based manipulation of the spline curves comes from the thesis work of Fowler [?]. Fowler's algorithm takes a change in a point on the curve or the tangent to the curve and makes the minimum adjustment to the control points to achieve this change.

Interaction

In symSpline, the user starts by selecting a point P on a curve. After selecting P , the ends of the tangent to P are locked to the two cursors and P is locked to the midpoint of the tangent. By moving the tangent around with two mice, three separate effects are available:

- Point translation - as P moves, the spline segment moves.
- Tangent rotation - as the angle of the tangent changes, the spline adjusts such that the first derivative at P matches the slope of the tangent.
- Tangent scale - as the tangent lengthens, the spline segment gets wider, as the tangent shortens, the spline segment becomes more narrow.

Figure ?? shows the first two effects in serial: a spline that is relatively straight is made into a curve by moving the edit point up, and then it is skewed by adjusting the angle of the tangent. Figure ?? shows the last effect. A curve of medium width can be widened by pulling the tangent ends apart or narrowed by pressing the tangent ends together.

Figure 3: A straight curve is adjusted by moving the edit point up and then rotating the tangent to skew the curve.

Figure 4: A medium curve is made wide or narrow by stretching or shrinking the tangent.

Because we use cubic B-splines, the effects above can be applied to the middle of a straight curve to twist it into an ‘S-curve’ in a symmetric motion (see Figure ??). The nature of the task and the symmetric design of symSpline allow all three effects to be combined in a single two-handed symmetric motion. The three effects do not have to be serialized; indeed, we expect expert symSpline users to combine the effects simultaneously as illustrated in Figure ??.

Figure 5: Three tangent movement effects applied on a straight curve to twist it into an ‘S’ shape. Previous curve state is shown at each stage for illustration purposes.

Advantages of symSpline

In the experiment described in this paper, we tested four techniques: symSpline, a dual-mouse left-asymmetric technique, a dual-mouse right-asymmetric technique and a single-mouse technique. The advantage of symSpline and the dual-mouse asymmetric techniques over the single-mouse technique is the ability to explore many different curves. Once a point has been selected, the two-handed techniques allow a user

to move the point freely while simultaneously exploring the tangent adjustments, without requiring a mode switch. Owen et al. described this exploration ability as the ‘expressiveness’ available in two-handed input [?]. In contrast, the single-mouse approach requires a mode switch between the translation of the point and the adjustment of the tangent, which makes the process of exploring the possible point locations and tangent adjustments tedious, and is likely to result in fewer possibilities being explored and a less satisfactory final curve.

The benefit of the symSpline technique over the dual-mouse asymmetric techniques is that the user does not have to remember which hand is performing which task. The tangent stretching between the two hands visually integrates the tasks of each hand into a single, perceptually unified task. The symmetry allows the user to concentrate on moving the tangent around in space to achieve the desired curve, rather than trying to remember to translate the point with one hand and adjust the tangent with the other.

Dual cursor issues

There are some issues that can arise with dual-mouse, dual-cursor interaction techniques. The first issue is divided attention: if the user has to keep track of two separate cursors on screen, she must decide which cursor to pay attention to at any given time, and may end up serially switching attention back and forth between the two. This issue was studied by Balakrishnan and Hinckley, who recommended that the objects controlled by the two hands be visually integrated [?]. Because the symSpline technique provides visual integration between the two cursors through the tangent, divided attention issues do not arise. The second dual cursor issue is the relative position of the cursors on screen, with respect to the user’s hands. If the cursor controlled by the left hand moves to the right of the cursor controlled by the right hand, the user can become confused because humans generally expect things controlled by their right hand to be to the right of things controlled by their left hand. This problem is well-known in psychology as a stimulus-response incompatibility [?, ?]. The stimulus of moving the right mouse is not compatible with the response of the movement of the cursor farthest to the left on screen. This issue can arise in dual-mouse spline manipulations including symSpline and the asymmetric dual-mouse techniques that we test in our experiment. It is most likely to occur in the splines that are oriented vertically. Because the dual-mouse spline manipulation techniques provide visual integration between the two cursors, and the responses are continuous, rather than discrete, we expected the users to be able to handle the stimulus-response incompatibility without large degradation in performance. We felt that the users would learn to handle the reverse mapping by considering the tangent to be ‘flipped over’. However, a small degradation in performance seemed likely.

Figure 6: Practice splines. The control polygon of each spline is shown here, but was not displayed to participants.

(1) (2) (3) (4) (5) (6) (7)

Figure 7: Splines used in timed trials.

Experimental design

We conducted an experiment to test the symSpline technique against other spline manipulation techniques. We felt it was important to test symSpline against both the traditional one-handed technique and two-handed asymmetric techniques. The methods we tested are summarized in Table ?? . The experimental task involved matching two spline curves, similar to the task recommended by Bartels et al. for splines experimentation [?]. Matching was performed by translating a predetermined edit point on the movable spline and adjusting the tangent until the movable spline matched the fixed target spline. One difference between Bartel’s spline matching task and ours is that the two splines share the same screen space rather than being in two separate side-by-side windows. We felt this would allow closer matching, and prevent fatigue from switching focus between two different screens. This experimental setup is similar to the one used by Owen et al. [?]. The one-handed technique chosen was based on the technique used in current graphics software such as Adobe Illustrator, where the cursor can be used either to move the edit point or adjust the tangent. In the asymmetric-left technique, the left mouse controlled point translation and the right mouse controlled tangent adjustment. The asymmetric-right technique reversed the task assignment.

Technique	Point Translation	Tangent Adjustment
Single Mouse	Select point and drag with main button pressed	Select end of tangent and drag with main button pressed
Asymmetric Left	Drag left mouse with main button pressed	Drag right mouse with main button pressed
Asymmetric Right	Drag right mouse with main button pressed	Drag left mouse with main button pressed
symSpline	Unified manipulation: drag both mice with main buttons pressed	

Table 1: Techniques tested in experiment. ‘Main’ button refers to the left button on the right mouse and the right button on the left mouse.

Asymmetric techniques

We have included two asymmetric techniques in our experiment because it was unclear in the pilot testing stage which of

the two asymmetric techniques was more natural. Of the two sub-tasks (point translation and tangent adjustment), the tangent adjustment sub-task seems to require more fine-grained control. This indicates that the asymmetric-left technique, where the left hand controls the translation of the point and the right hand controls the adjustment of the tangent would be most natural for our right-handed participants. Additionally, since pilot participants appeared to move the point first and adjust the tangent second when using the asymmetric techniques, having the left hand do the task that comes first fits with Guiard’s guidelines. However, most pilot participants complained that this technique was difficult to work with and that they preferred the opposite technique, where the right hand translates the point and the left hand adjusts the tangent. The completion times for these pilot test participants were also better for the asymmetric-right than for the asymmetric-left. Because there was no clear winner between the two dual-mouse asymmetric techniques, we decided to include both of them in our final experiment. The fact that there is no obvious way to divide the task asymmetrically suggests that the task be naturally symmetric.

Trial setup

The experiment session was a within-participants design, consisting of four blocks of trials, one for each technique described above. In the first block there were six practice trials and in each of the following three blocks, there were three practice trials (see Figure ?? for pictures of the practice splines). The increased practice trials in the first block were used to help the participants become accustomed to the splines. Each set of practice trials was followed by 28 randomized trials, consisting of 14 different splines, each seen twice (see Figure ??). Thus, each participant performed a total of 15 practice matches and $14 * 2 * 4 = 112$ timed matches during the experiment. The order of the four blocks was randomized for each participant.

The fourteen splines shown in Figure ?? were chosen to test the techniques. These splines have four main characteristics, being either wide, medium or narrow in the shape of the curve, being either centered or skewed in the alignment of the curve, and having either one or two bumps (we will refer to a spline with two bumps as an S-curve). We wanted this variety for two reasons. First, using a variety of different splines would help to prevent motor learning effects from the participants repeating the same motion. Second, we felt that

skewed splines and S-curve splines would be harder to match and would therefore allow us to see how the techniques differ when the task is more challenging. We considered adding more complex splines, but we found during pre-trials that the addition of more control points changed the nature of the task and was too difficult and time-consuming for novice participants. In addition, we wanted to limit the splines to those that could be matched (from a straight line) with a single motion.

Our participant pool consisted of thirty-four undergraduate students from a variety of departments (Arts, Accounting, Health, Science, Biochemistry, Mathematics). Eighteen of the students were female, sixteen were male. All participants self-identified as right-handed.

Because we expected our participants to have no familiarity with manipulating spline curves, we gave them some instruction before the experiment. We developed an online presentation that showed examples of spline curves and how they could be changed by moving an edit point, adjusting the angle of a tangent and adjusting the length of a tangent. The presentation also gave the participants general information on how the experiment would proceed.

Each experiment session lasted roughly 40 minutes, including the time to sign consent forms, view the online instructions and ask any questions. A few participants took as little as 30 minutes and one participant took almost 55 minutes to complete the experiment. The participants were paid \$10 for their participation.

Figure ?? shows the process of a trial. At the start of each trial, one or two small hollow circles appeared, and the participant had to move their cursor(s) into the circle(s) (Figure ??(a)). When the cursor(s) were inside the circle(s), four evenly-spaced beeps were played and the trial began at the end of the fourth beep. At this point, the blue movable spline and the green target spline both appeared on screen (Figure ??(b)). The two splines had their end points clamped together. The blue spline started off as a straight line stretching between the two endpoints of the green target spline. The spline endpoints were located where the start circles had been located, thus the participant's cursors always started at the ends of the splines.

The spline matching task

Because splines are relatively specialized, and our test participants were not familiar with how to manipulate them, we needed to make the spline matching task as simple as possible, while still testing the effectiveness of the interaction techniques. Specifically, we did not want the timing results to be affected by the level of splines knowledge and experience. To prevent this type of variance, we limited the participants to manipulating a single point at the center of the movable spline. In this way, the participants did not have to make a value judgment about what part of the spline to select for manipulation.

There were two other simplifications to the spline matching task. First, the end points of the two splines were clamped together and the participant could not adjust the end points of the movable spline. Second, the splines used for the experiment were simple. All of the target splines were single-segment cubic B-splines. We used dual-segment cubic B-splines for the movable splines, because single-segment splines proved difficult for pilot participants to match. These factors combined to make the task easy enough that participants with no prior splines experience could quickly master the task.

The participant's task was to acquire the highlighted edit point in the middle of the blue spline (Figure ??(c)) and then manipulate the blue spline until it matched (*i.e.*, was on top of) the green target spline (Figure ??(d)). A spline match was obtained when the maximum of the Euclidean distances from each point on the movable spline to the nearest point on the target spline was under ten pixels. Thus the match was geometric and not parametric.

The edit point was always acquired by clicking on it with the main mouse button (left button on right mouse or right button on left mouse). Once the point was acquired, the tangent appeared, and the cursor(s) warped appropriately. All manipulations required the user to hold down the main mouse button(s) and drag to move the tangent and/or edit point. In the single-mouse technique, the user could either click down on the edit point and drag to translate, or click down on the highlighted end of the tangent and drag to adjust the angle and length of the tangent.

Hypotheses

We predicted the following results:

- H1: The symSpline technique would be faster than the other three techniques.
- H2: The dual-mouse asymmetric techniques would be faster than the single-mouse technique, but slower than the symSpline technique.
- H3: The symSpline technique would be the most preferred technique in the participants' post-experiment evaluation.

Results and discussion

The thirty-four participants completed 112 timed trials each, for a total of 3808 trials. All three hypotheses are strongly supported by the results of the experiment.

Outlier Removal

Out of the 3808 trials, 134 trials (3.2 percent) were removed as outliers, being greater than three standard deviations from the mean trial completion time. These outliers were not correlated with any of the four conditions or with any of the participants or spline IDs. Thus, the remainder of the analysis was performed using 3674 trials.

(a) (b) (c) (d)

Figure 8: Trial procedure for symSpline experiment.

Condition Presentation and Learning Effects

Our choice of a within-participants design made it necessary to check that there were no confounding effects due to the order of presentation of the techniques. After aggregating the results for each participant for each of the four conditions, an ANOVA of average trial completion time was performed, using condition and block number as variables. The condition variable was significant ($F_{3,120} = 12.1, p \leq 0.0001$). The block number variable was not significant ($F_{3,135} = 0.59, p = 0.62$), showing that there were no significant learning effects. In addition, there was no interaction effect for condition * block number ($F_{9,120} = 0.51, p = 0.8673$). This shows that the random assignment of condition presentation order was sufficient to prevent order effects, and so the block number variable could be removed from the analysis.

General Results

Figure 9: Boxplots of trial completion time for the four different techniques. The mean trial completion time is shown by the black dash inside of each box.

Source	DF	F Ratio	Prob > F
Technique	3	98.46	< .0001
Alignment	1	111.27	< .0001
ID	6	46.85	< .0001
Alignment*Technique	3	8.86	< .0001
ID*Technique	18	13.90	< .0001
Alignment*ID	6	16.01	< .0001
Technique*Alignment*ID	18	1.04	.4059

Table 2: Repeated Measures Analysis of Variance for completion time against technique, splineID and alignment, with interactions. DF Denominator is 1806 for all variables (1895 data points included).

After removing outliers and ensuring that learning and condition order were not significant factors, we analyzed the aggregate data for all participants for each of the four conditions and each of the 14 splines (effectively splitting our original data set in half, since each participant saw each spline twice for each of the four techniques). The hypothesis that the symSpline method would be faster than the other three techniques is supported by the aggregate participant results as summarized in Figure ?? and the repeated measures ANOVA shown in Table ?. Tukey HSD pairwise tests also indicate that the symSpline technique is significantly different from the other techniques. The mean completion time

of the symSpline technique is more than 2.5 seconds faster than the mean completion time of the single-mouse technique. The low variance shown by the symSpline boxplot in Figure ?? suggests that the symSpline technique is more natural for novice users than the other three techniques.

Our second hypothesis, that the dual-mouse asymmetric techniques would be faster than the single-mouse technique, is also supported. The two asymmetric techniques generated faster completion times than the single-mouse technique, but were not as fast as the symSpline technique. However, the difference between the two asymmetric techniques appears to be minimal and is not statistically significant in Tukey HSD pair-wise tests, which supports our argument that the task is naturally symmetric.

Variable Analysis

To analyze the effects of the different splines and the orientation of the splines, we aggregated the data for each participant by splineID x spline alignment x technique and analyzed this data with a repeated measures ANOVA, shown in Table ?. Tukey HSD Post Hoc tests confirm that the four techniques are statistically different from one another at the $\alpha = .05$ level, except for the two asymmetric techniques, which are not significantly different. We expected little difference in performance between the asymmetric left and asymmetric-right techniques. These tests, combined with the low completion time mean, show that symSpline is significantly faster than the other techniques in our experiment.

Spline Alignment Effects The splineID variable is a unique identifier assigned to each of the seven horizontal splines and to each of the seven vertical splines in the data set, see Figure ?. It makes sense for the ID to have a significant effect, as the different splines in the dataset would have different difficulty levels, depending on various aspects of the spline’s shape. In addition, there is an interaction between splineID and technique. This makes sense, as we expected the difference between the symSpline technique and the other techniques to be more pronounced for splines that are more difficult to match. The validity of this explanation is examined in the following section. We were interested in the alignment characteristic because we anticipated that the vertically-aligned splines would be more challenging, especially for the dual-cursor techniques. There is a main effect of alignment and there are also alignment * splineID and alignment * technique interaction effects.

Spline-dependent results

Our choice of the fourteen splines displayed in Figure ?? allowed us to analyze the trial completion times with respect to various aspects of the splines themselves. We consider two

different variables in combination with spline alignment: the number of bumps on the spline (1 or 2) and the shape of the spline (symmetric or skewed). We were interested in the number of bumps and the shape of the spline because we anticipated that S-curve splines and skewed splines (shown in Figure ??) would be more challenging to match and we were interested in seeing if the symmetric technique performed well with more difficult spline-matching tasks.

Single-bump versus S-curve Splines To examine the differences between single-bump splines and S-curve splines, we selected two single-bump splines (one vertical and one horizontal) to compare with the vertical and horizontal S-curve splines. Splines labeled 4 were chosen to compare with the splines labeled 7 (see Figure ?? because similar tangents would be required).

Figure 10: A single-bump spline and an S-curve spline.

Figure 11: Mean trial completion times for single-bump versus S-curve splines for each of the four techniques.

When the aggregated trial completion times are separated by splines with one bump versus S-curves (see Figure ??), salient differences appear. The difference between the symSpline technique and the single-mouse technique is especially large for S-curve splines. The symSpline technique is actually faster with S-curve splines than it is with single-bump splines, which is not the case for the two asymmetric techniques. We are unsure of the reason for this effect. To analyze the significance of these differences, a repeated measures ANOVA was done using trial data for the four splines aggregated over the 34 participants. The results of this analysis are shown in Table ?. There is an interaction effect of Bumps * Technique, showing that the difference between single-bump and S-curve splines for the four techniques is significant. A Tukey HSD Post Hoc test shows that the symSpline technique is significantly different from the other techniques for S-curve splines at the $\alpha = .05$ level.

Skewed versus Symmetric Splines To examine the differences between skewed splines and symmetric splines, we selected splines labeled 2 and 5 to represent symmetric splines and splines labeled 4 and 6 to represent skewed splines (see Figure ??).

When the mean trial completion time results are separated by symmetric splines versus skewed splines, there are no significant differences across the four techniques. Although skewed splines take longer to match than symmetric splines, this difference does not vary with the technique being used. Thus, our conjecture that the harder to match splines would benefit

Source	DF	F Ratio	Prob > F
Technique	3	89.85	< .0001
Alignment	1	.83	.3626
Bumps	1	76.30	< .0001
Alignment*Technique	3	7.80	< .0001
Bumps*Technique	3	44.37	< .0001
Alignment*Bumps	1	2.05	.1521
Technique*Alignment*Bumps	3	0.44	.7210

Table 3: Repeated Measures Analysis of Variance for completion time against technique, alignment, and bumps with interactions. DF Denominator is 492 for all variables (541 data points included).

most from the symSpline technique is not supported by this data. A repeated measures ANOVA for the four spline sets in this analysis has an $F_{3,1034} = 0.37, p = 0.7753$ for technique * skew, showing that there is no significant difference for this interaction.

Subjective Results

Our third hypothesis, that the symSpline method would be preferred by participants, is also supported. After completing the experiment, we asked users to rate the four methods in terms of usage preference (Table ??). Specifically, we asked participants the question:

If you had a job where you had to manipulate curves like those in the experiment, which method would you most like to use, and least like to use?

The participants were asked this question without being informed of their relative performance with the various techniques. Out of the 34 undergraduate participants, 22 preferred the symSpline method, while only 3 listed symSpline as their least preferred method. The single-mouse technique was preferred by 3 participants and least preferred by 14 participants. Another interesting result here is that of the two asymmetric techniques, the asymmetric right was more preferred than the asymmetric left, and the asymmetric left was the least preferred by many participants. This is what we expected after our pilot testing, although it seems to violate Guiard's model of asymmetric interaction.

Technique	Most Preferred	Least Preferred
Single Mouse	3	14
Asym Left	4	14
Asym Right	6	4
symSpline	22	3

Table 4: Number of participants who declared each technique as their most preferred and least preferred.

Conclusions and future work

Our experiments demonstrate that symSpline is an effective technique for the manipulation of spline curves, at least in the simplified setting we tested. New symSpline users are able to manipulate and match splines faster than with any of the other techniques tested. While the symSpline technique wasn't significantly better at manipulating the more difficult skewed splines, it was significantly faster for manipulating the more complicated S-curves. This suggests that the symSpline technique is likely to fare well in a more general setting. The participants were generally enthusiastic about the symSpline technique.

The major contributions of this work are twofold. First, we have presented and tested a new and effective method for spline manipulation for novices performing simple spline manipulation. Second, we have presented an example of the usefulness of symmetric two-handed interaction, which supports our earlier work [?] in arguing that symmetric interaction deserves more attention in HCI research.

This research is situated in a larger context of work that argues for the introduction of a second spatial input device to the standard computing desktop. There are pragmatic issues related to the large base of installed applications that use a single cursor. In such an environment, adding an application that differs in its basic interaction and hardware requirements is difficult. However, as part of our larger research agenda, we are working on other dual-cursor applications, as well as a dual-cursor window manager that supports legacy single-cursor applications. Given this larger context, we feel that the results presented here are not simply theoretical.

In the future, we wish to investigate how experts perform with symSpline in a task that is less constrained than the experimental task described here. We also plan to extend symSpline into a "symSurface" tool for symmetric surface editing. Finally, we are interested in investigating other domains where dual-mouse symmetric interaction could allow easy exploration of data, similar to the way symSpline allows easy exploration of spline curves.

Acknowledgments

We would like to acknowledge Kevin Moule and other members of the Computer Graphics Lab for their support and encouragement. Ravin Balakrishnan provided valuable feedback on this paper. Finally, we would like to thank the CHI reviewers and our statistical 'shepherd', Michel Beaudouin-Lafon, for many valuable comments and suggestions.

REFERENCES

1. Seok-Hyung Bae, Takahiro Kobayashi, Ryugo Kijima, and Won-Sup Kim. Tangible NURBS-curve manipulation techniques using graspable handles on a large display. In *UIST 2004 Proceedings*, 2004.
2. Ravin Balakrishnan, George Fitzmaurice, Gordon

- Kurtenbach, and William Buxton. Digital tape drawing. In *UIST'99 Proceedings*, 1999.
3. Ravin Balakrishnan and Ken Hinckley. Symmetric bimanual interaction. In *CHI 2000 Proceedings*, 2000.
4. Richard Bartels and John Beatty. A technique for the direct manipulation of spline curves. In *Graphics Interface 1989 Proceedings*, 1989.
5. Richard Bartels, John Beatty, Kellogg Booth, Eric Bosch, and Pierre Jolicoeur. Experimental comparison of splines using the shape-matching paradigm. *ACM Transactions on Graphics*, 12(3):179–208, 1993.
6. Didier Casalta, Yves Guiard, and Michel Beaudouin-Lafon. Evaluating two-handed input techniques: Rectangle editing and navigation. *ACM CHI'99 Conference (Extended Abstracts)*, pages 236–237, 1999.
7. Jorn Diedrichsen, Richard B. Ivry, Eliot Hazeltine, Steven Kennerley, and Asher Cohen. Bimanual interference associated with the selection of target locations. *Journal of Experimental Psychology: Human Perception and Performance*, 29(1):64–77, 2003.
8. Adam Finkelstein and David H. Salesin. Multiresolution curves. *Computer Graphics*, 28(Annual Conference Series):261–268, 1994.
9. P.M. Fitts and R.L. Deininger. S-R compatibility: Correspondance among paired elements within stimulus and response codes. *Journal of Experimental Psychology: Human Perception and Performance*, 48:483–492, 1954.
10. George Fitzmaurice, Hiroshi Ishii, and William Buxton. Bricks: Laying the foundations for graspable user interfaces. In *CHI'95 Proceedings*, 1995.
11. Barry Fowler. Geometric techniques for interactive curve design. Master's thesis, University of Waterloo, 1990.
12. Barry Fowler. Geometric manipulation of tensor product surfaces. In *SI3D '92: Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 101–108, New York, NY, USA, 1992. ACM Press.
13. Barry Fowler and Richard Bartels. Constraint-based curve manipulation. *IEEE Computer Graphics and Applications*, 13(5):43–49, 1993.
14. Elizabeth A. Franz, Howard N. Zelaznik, Stephan Swinnen, and Charles Walter. Spatial conceptual influences on the coordination of bimanual actions: When a dual task becomes a single task. *Journal of Motor Behaviour*, 33(1):103–112, 2001.

15. Yves Guiard. Asymmetric division of labour in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behaviour*, pages 486–517, 1987.
16. Stanley Jang, Kellogg S. Booth, David R. Forsey, and Peter Graf. Investigating the effectiveness of direct manipulation of 3D B-spline curves using the shape-matching paradigm. In *Graphics Interface 1993 Proceedings*, 1993.
17. J.A.S. Kelso, Dan L. Southard, and David Goodman. On the coordination of two-handed movements. *Journal of Experimental Psychology: Human Perception and Performance*, 5(2):229–238, 1979.
18. Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. The design of a GUI paradigm based on tablets, two-hands and transparency. In *CHI'97 Proceedings*, 1997.
19. Celine Latulipe. Symmetric interaction in the user interface. In *Companion Proceedings of UIST 2004*, pages 63–66, 2004.
20. Celine Latulipe. Simultaneous rotation and translation. In *Proceedings of HCI 2005, Volume 2*, 2005.
21. Celine Latulipe, Craig S. Kaplan, and Charles L.A. Clarke. Unimanual and bimanual image alignment: An evaluation of mouse-based techniques. In *Proceedings of UIST 2005*, 2005.
22. Andrea Leganchuk, Shumin Zhai, and William Buxton. Manual and cognitive benefits of two-handed input. *ACM Transactions on Computer-Human Interaction*, 5(4):326–359, 1998.
23. Ignacio Llamas, Byungmoon Kim, Joshua Gargus, Jarek Rossignac, and Chris D. Shaw. Twister: a space-warp operator for the two-handed editing of 3D shapes. *ACM Trans. Graph.*, 22(3):663–668, 2003.
24. Franz Mechsner, Dirk Kerzel, Gunther Knoblich, and Wolfgang Prinz. Perceptual basis of bimanual coordination. *Nature*, 414:69–73, 2001.
25. R. Nicoletti and C. Umilta. Right-left prevalence in spatial compatibility. *Perception and Psychophysics*, 35:333–345, 1984.
26. Russell Owen, Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. When it gets more difficult, use both hands - exploring bimanual curve manipulation. In *GI 2005 Proceedings*, 2005.
27. Chris Shaw and Mark Green. THRED: A two-handed design system. *Multimedia Systems*, 5:126–139, 1997.