Tech-note: Dynamic Dragging for Input of 3D Trajectories

Daniel F. Keefe* Robert C. Zeleznik† David H. Laidlaw‡

Department of Computer Science
Brown University

Abstract

We present Dynamic Dragging, a virtual reality (VR) technique for input of smooth 3D trajectories with varying curvature. Users "drag" a virtual pen behind a hand-held tracked stylus to sweep out curving 3D paths in the air. Previous explorations of dragging-style input have established its utility for producing controlled, smooth inputs relative to freehand alternatives. However, a limitation of previous techniques is the reliance on a fixed-length drag line, biasing input toward trajectories of a particular curvature range. Dynamic Dragging explores the design space of techniques utilizing an adaptive drag line that adjusts length dynamically based on the local properties of the input, such as curvature and drawing speed. Finding the right mapping from these local properties to drag line length proves to be critical and challenging. Three potential mappings have been explored, and results of informal evaluations are reported. Initial findings indicate that Dynamic Dragging makes input of many styles of 3D curves easier than traditional drag-style input, allowing drag techniques to approach the flexibility for varied input of more sophisticated and much harder to learn techniques, such as two-handed tape drawing.

Keywords: 3D drawing, direction-guided input, tape drawing.

Index Terms: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality

1 Introduction

Many virtual reality applications rely upon input from users' hands as recorded by 3-DOF or 6-DOF tracking technologies. Hands are used to point, grasp, gesture, and sweep out complex 3D trajectories in VR. Sweeping 3D input, the focus of this paper, has application in 3D selection and path planning techniques for science [8], immersive CAD tools [2], and artistic interfaces for virtual sculpting and scientific illustration [4, 5, 6, 7].

While direct, sweeping input from the hand is a seemingly intuitive way to specify complex 3D shapes and trajectories, unfortunately, it also tends to be quite challenging to control. A primary difficulty is reliably controlling one's hand as it moves through the air without the aid of a surface to push against. Human muscular jitter combined with tracking errors and perceptual challenges, such as accurate judgments of depth in VR, have traditionally limited this style of input to applications where sketchy, gestural, or slightly imprecise input is acceptable. To move beyond these limitations, researchers have proposed new interactions [5] and input filtering techniques [3]. In this technote, we aim to refine one of these, the "drag drawing mode" of the Drawing on Air system [7], to more appropriately handle input of trajectories with wide variation in curvature.

*e-mail: dfk@cs.brown.edu
†e-mail: bcz@cs.brown.edu
‡e-mail: dhl@cs.brown.edu

IEEE Symposium on 3D User Interfaces 2008
8-9 March, Reno, Nevada, USA

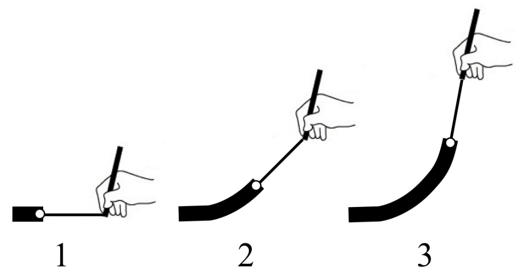

Figure 1: In dragging-style input, the stylus held by the user is attached by a drag line (the thin black line) to a virtual pen (the small white circle). The curve input to the computer (the thick black line) is swept out by the virtual pen as it is towed around behind the stylus. This level of indirection in the input technique has the advantage of acting as a user-guided filtering mechanism, smoothing the input and making it easier to draw accurate 3D curves.

Our proposed technique builds considerably upon previous work. In particular, it has been shown that dragging a virtual brush behind a stylus manipulated by the user is useful, as compared to freehand input strategies, for producing smooth, controlled input, both in 2D [1] and 3D [7]. One limitation of these techniques is that the length of the drag line used biases the technique toward input of trajectories within a certain range of curvature. We examine a class of techniques that address this limitation through dynamic adjustment of the drag line length during input. The intent is to adaptively tune the input technique to best match the local curvature of the trajectory as it is input.

The design of such a technique raises several interesting questions: What properties of the input processed thus far may be used to predict the likely future input? How are these properties mapped to the drag line length parameter? Can the fluidity of this sweeping input style be maintained even as the technique adjusts dynamically? What visual feedback is most appropriate in helping to guide the user?

This technote makes two main contributions. First, we motivate and describe the design space alluded to by the questions above. Second, we present our initial investigations into this design space together with informal evaluation of several alternative designs. We begin with a discussion of the most pertinent related work.

2 Related Work

Our technique builds most directly upon the Drawing on Air tools described by Keefe et al. [7] which, in turn, are inspired by "tape drawing", the two-handed drawing technique used by car designers [1]. Drawing on Air contains two input modes: one-handed drag mode and two-handed tape mode. Both modes make possible controlled styles of input by putting the user in direct control of the drawing *direction* (the tangent) of the trajectory recorded by the computer. By specifying the drawing direction and then advancing along that direction, as opposed to directly inputting position by moving the hand freely in the air, these techniques establish a constraint that acts as a user-guided filter, making the input smooth and more accurate than freehand alternatives. A 2D illustration of the drag technique is shown in Figure 1.

One important reason for having two input modes in these systems is that two-handed techniques based on tape drawing take time to master. While artists and designers are likely to be willing to invest the required training time, less frequent users, scientists and doctors for example, may be less willing. The one-handed drag mode has many of the same input filtering qualities of the two-handed tape mode, and, since it relies on just one hand, it is easier and more natural to learn.

In this work, we address a current limitation of the one-handed approach as compared to the two-handed, the inability to adjust the technique while drawing to handle complex curving trajectories. In the two-handed approach, the tasks of setting the drawing direction and advancing along it are separated, one assigned to each hand, making it far more flexible with regard to adapting to different input shapes. We seek to capture this flexibility in a one-handed technique that may be adopted by users with little or no previous training.

3 Dynamic Dragging

Our approach is to dynamically adjust the length of the drag line to an appropriate value based on an estimation of the input the user will perform next. In particular, the length of the drag line is selected based on the expected curvature of the drawing trajectory. In general, short drag lines are appropriate for specifying tight curves, and long drag lines are appropriate for straight and wide curves. When inputting a curve of varying curvature, the effectiveness of the drag technique is likely to be significantly enhanced if the length of the drag line adjusts dynamically as warranted by changes in curvature.

The sections below present details of the design tradeoffs discovered while investigating this technique. Important to consider are strategies for adjusting drag line length, maintaining a fluid style of interaction despite dynamic changes in the technique, and providing appropriate visual feedback to the user.

3.1 Adjusting the Drag Line

The most critical design decision in this work is determining the best mapping from the input encountered thus far to the drag line length. This requires forming a prediction of the curvature of the input to be encountered next.

3.1.1 Strategy 1: Local Curvature

The most straightforward approach is to predict future curvature using the most recent curvature, which can be computed from the portion of the trajectory already entered. Care must be taken in specifying an appropriate sampling window and weighting for calculating the local curvature. A small window leads to a responsive technique, but too responsive can lead to a lack of control as changes in the drag line length appear confusing and noisy.

A successful approach as evaluated by our pilot users is to calculate local curvature at equally spaced samples along the most recently input 2 cm of the curve and calculate a weighted average of these values. [1] Our implementation uses a Gaussian falloff for the average, giving the highest weight to the most recent samples. Given this estimate of the current desired input curvature, we set the desired drag line length d_1 to the radius of curvature (the reciprocal of curvature K).

$$d_1 = \frac{1}{K} \tag{1}$$

[1] 2 cm is appropriate for our desktop VR system, which has a working volume limited by the range of the Phantom tracking device used. The appropriate value should change with the scale of the VR form factor and extents of typical input on other systems.

In practice, d_1 should also be clamped to a minimum and maximum to avoid extreme values.

3.1.2 Strategy 2: Drawing Speed

An alternative design drives the change in drag line length by the speed of drawing. The theoretical basis for this comes from the neuroscience literature, where the Two-Thirds Power Law describes a relationship between drawing speed and the curvature of the drawing trajectory. In simplified terms, drawing is slower in regions of higher curvature. [9]

Following this experimentally derived relationship, a second strategy for calculating a desired drag line length may be expressed as

$$d_2 = k||\vec{v}||, \tag{2}$$

where $||\vec{v}||$ is the current drawing speed, calculated as a weighted average from recent samples, and k is a hand-tuned gain factor ($k = 1.2$ in our implementation).

3.1.3 Strategy 3: Combination

Informal evaluation (see Section 4) indicates that both strategies have merit and neither significantly outperforms the other in all cases. As such, our current practice is to average the results of the two strategies,

$$d_3 = \frac{d_1 + d_2}{2}, \tag{3}$$

so as to incorporate both recent curvature and drawing speed in the calculation of a desired drag line length.

3.2 Achieving Fluid Interaction

Given a desired length for the drag line, there remains a question of how to update the technique to use this new length in a non-disruptive way. If the length changes too drastically from one frame to the next, then the effect can become distracting. The design principle employed in our solution is to change the drag line length each frame in proportion to the amount of change introduced into the system by the user's drawing movements. The user expects the display to update as he moves, and the technique masks the change in the the drag line length in a larger change resulting from the user's movement of the stylus. From one frame to the next, drag line length is constrained to change no more than seventy-five percent of the arclength of the curve segment drawn between the two frames. Over several frames, the changes accumulate and the drag line length approaches the desired value.

3.3 Providing Visual Feedback

Visual feedback in the form of a virtual representation for the drag line displayed on-screen is imperative for the success of the technique, allowing users to work deliberately lining up the guideline before advancing to achieve accurate input [7]. Worried that a guideline that changes length might confuse users, we explored two visual feedback options. In the first, the guideline drawn on screen is the length of drag line and changes dynamically during drawing. In the second, the exact same approach to updating the drag line length is used in the underlying system, but the guideline is drawn on the screen with a constant, relatively long length. Feedback from user testing of these two configurations in reported in the next section.

4 Evaluation

Three members of our lab compared Dynamic Dragging to the Drawing on Air drag mode technique [7]. The VR setup is pictured in Figure 2. A stereoscopic, head-tracked desktop display was used in conjunction with a SensAble Phantom device for precise 3D tracking of the hand. The force feedback feature of the

Phantom was used only minimally. A small viscous force was applied to provide slight resistance to movement of the stylus through the air.

4.1 Tasks

Participants were asked to complete two tasks multiple times with each technique. Task 1 is tracing a 3D curve displayed in the virtual environment. The curve is a repeating sine wave with varying frequency and amplitude that is bent slightly into a 3D shape. Task 2 is motivated by a 3D selection problem posed in a brain visualization application developed in our lab. The task is selecting a subset of neural fiber tracts situated within a larger set of fibers by drawing an enclosing curve. A set of red and green tubes oriented in space as seen in Figure 3 were used to approximate the neural data. Participants were asked to draw curves around the green tubes only, as if to select them via a lasso technique. The input required to complete both tasks demands accurately describing regions of both high and low curvature within a single curve.

4.2 Results

All participants immediately confirmed the difficulty of producing input trajectories of high curvature with the standard drag technique in both tasks. Difficulty was encountered in regions requiring direction reversal at the minima and maxima of the sine wave. In the selection task, working in tight spaces was difficult. A tightly curving, slalom-like trajectory was required in one instance. This provided the most obvious example of differences between the two techniques. In each of these instances, the drag line used in the standard technique was too long, requiring a large amount of sideways movement as compared to forward drawing movement in order to produce a tight curve. The curves produced tend to look jaggy as drawing must stop to allow for sideways reorienting of the drag line before each slight forward advance.

In some instances Dynamic Dragging led to a drag line length that was too short. This was encountered only during the tracing task. In the straighter regions of the sine wave shape, when the wave has both a high amplitude and frequency, if the drag line is too short, then the smoothing and filtering benefits of the technique are reduced and the trajectory produced tends toward looking wobbly, as if drawn by a freehand technique. We did not notice an instance of the drag line becoming too short in the selection task, one reason may be that when tracing, participants tended to move at a constant, very slow speed. In contrast, this extremely deliberate drawing was only necessary in difficult areas in the selection task, which tended to be of high curvature, so short drag line lengths worked well in these areas.

All participants reported that Dynamic Dragging was faster and easier than the standard technique, although they had difficulty establishing a consistent preference for whether curvature, speed, or a combination of both should be used to control the drag line length. Drawing speed seemed to be a less appropriate control than curvature for the tracing task. Thus, the right method to use may be task dependent.

While participants preferred the Dynamic Dragging method for the tracing task, it is unclear whether tracing performance was actually better with Dynamic Dragging or whether the areas of difficulty simply switched from regions of high curvature to regions of low curvature. Quantitative analysis and evaluation in other contexts will be useful in the future.

With respect to visual feedback, we were surprised that users did not report being bothered by guideline length changes. While the changing itself did not seem to be a problem, two participants complained that the guideline is too hard to see when it is small. The mixed mode, where a dynamic line is used in the underlying calculation but a constant-length line is drawn on screen, was found to be confusing.

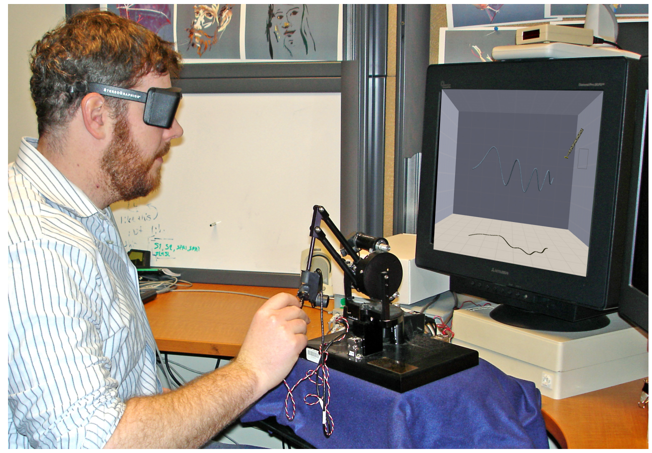

Figure 2: The virtual reality setup used to evaluate Dynamic Dragging. Task 1 is displayed on the screen.

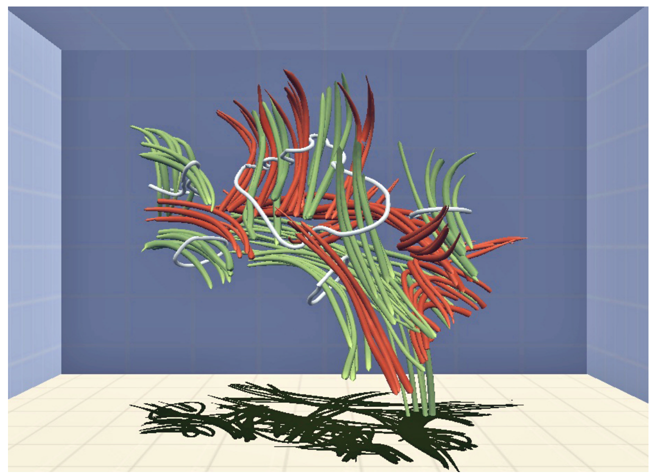

Figure 3: A simulated neural fiber lasso-selection task requires input of an accurate 3D curve with regions of both high and low curvature. The white selection curves were input with Dynamic Drawing.

5 Conclusions and Future Work

Dynamic Dragging would benefit from additional study of drag line length controls. Using drawing speed and local curvature to control length was useful for instances tested in our evaluation. Alternative factors may also be useful. 3D position and direction of drawing affect the difficulty of this style of 3D input [5, 10]. These parameters may also be relevant. Further, the relative importance of parameters appears to shift based on task, as seen when comparing tracing to selection. In the future, it will be useful to identify important factors for selection tasks in visualization applications as compared to curve input tasks in free-form modeling or CAD applications.

Dynamic Drawing takes a step forward in making drag-style input easier for a wide range of input trajectories. The new technique maintains the ease of use and input filtering qualities of the original, but approaches the flexibility of alternative input strategies, such as tape drawing, which are much harder to learn. As such, Dynamic Drawing may be useful in a variety of VR applications where accurate 3D sweeping input is desired with minimal user training.

Acknowledgements

The authors wish to thank the members of the Brown Scientific Visualization Research Lab. This work was partially supported by the W. M. Keck Foundation, NSF (ITR 04-27374, DBI 05-52051, IOS-0723392) and NIH (EB4155A1).

References

[1] R. Balakrishnan, G. Fitzmaurice, G. Kurtenbach, and W. Buxton. Digital tape drawing. In *Proceedings of UIST '99*, pages 161–169, 1999.

[2] J. Butterworth, A. Davidson, S. Hench, and M. T. Olano. 3DM: A three dimensional modeler using a head-mounted display. In *Symposium on Interactive 3D Graphics*, pages 135–138, 1992.

[3] M. Fiorentino, G. Monno, P. A. Renzulli, and A. E. Uva. 3D sketch stroke segmentation and fitting in virtual reality. In *Proceedings of GRAPHICON*, pages 188–191, 2003.

[4] A. D. Gregory, S. A. Ehmann, and M. C. Lin. inTouch: Interactive multiresolution modeling and 3D painting with a haptic interface. In *Proceedings of IEEE VR 2000*, pages 45–52, 2000.

[5] D. F. Keefe. *Interactive 3D Drawing for Free-Form Modeling in Scientific Visualization and Art: Tools, Methodologies, and Theoretical Foundations*. PhD thesis, Brown University, 2007.

[6] D. F. Keefe, D. A. Feliz, T. Moscovich, D. H. Laidlaw, and J. J. LaViola Jr. CavePainting: A fully immersive 3D artistic medium and interactive experience. In *Proceedings of I3D 2001*, pages 85–93, 2001.

[7] D. F. Keefe, R. C. Zeleznik, and D. H. Laidlaw. Drawing on air: Input techniques for controlled 3D line illustration. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1067–1081, 2007.

[8] J. S. Sobel, A. S. Forsberg, D. H. Laidlaw, R. C. Zeleznik, D. F. Keefe, I. Pivkin, G. E. Karniadakis, P. Richardson, and S. Swartz. Particle flurries: Synoptic 3D pulsatile flow visualization. *IEEE Computer Graphics and Applications*, 24(2):76–85, March/April 2004.

[9] P. Viviani and C. Terzuolo. Trajectory determines movement dynamics. *Neuroscience*, 7(2):431–437, 1982.

[10] S. Zhai. *Human Performance in Six Degree of Freedom Input Control*. PhD thesis, University of Toronto, 1995.