

A Haptic Interface for Editing Space Trajectories

Diana Garroway and Vincent Hayward

Haptics Laboratory, McGill University, Montréal, Québec, Canada

Abstract

Animators working with 3D models and data typically are still bound to 2D interaction. We present a new multi-modal interface for editing 3D motion data using a haptic device. In the basic mode, the haptic device guides the hand of the user along a trajectory that was previously recorded or specified. Any significant deviation from the initial trajectory instantly results in permanent changes made to this trajectory. This interface provides a simple, intuitive method for the user to actually experience movement other than visually for purposes of creation.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism-Animation

1. Introduction

In 3D computer animation, editing space-time trajectories, a four dimensional object, is essential to obtaining a precisely controlled animation. Unfortunately, the process is generally mathematically oriented (e.g. coordinate frames, control points), tedious and non-intuitive. We propose an intuitive editing method that brings the animator closer to the animation by allowing him or her to feel the virtual curves with a haptic device. Our interface abstracts the mathematics of curve editing and allows the animator to concentrate solely on the shape of the curve and on its effects on the resulting animation. This method is easy to learn and provides the animator with direct and immediate control over the resulting animation trajectory.

Common interfaces for editing motion trajectories for 3D animation involves editing 2D projections of the curve using control points (key frames). The user may control the speed of the transition between key frames and precisely refine the position of the keys, therefore editing the animation trajectory. Other trajectory editing tools have been developed based on many different mathematical properties, but most are based on the same principle of 2D projections. A method using haptic feedback to edit a character animation was presented in [DH00] by editing bundles of trajectories mapped to a haptic vector field. Our work differs in that we use friction to introduce plasticity (i.e. shape memory) in a single motion trajectory. Haptic feedback has been used in many other applications [HACH*04].

2. System Overview

The curve editing system comprises a haptic device, haptic control software and animation software. Figure 1 shows the interaction between these components.

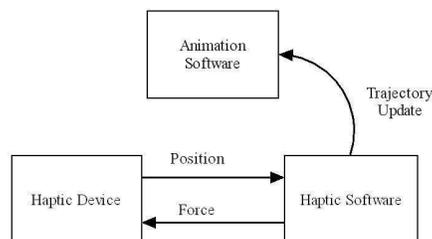


Figure 1: System illustration.

We use a commercial haptic device, the Freedom-6S™, produced by MPB Technologies (Montréal, Canada). This device has capabilities to provide 6 DOF in both the sensed position and the rendered force. It's design is similar to an arm-with a shoulder, elbow and wrist joint and the user holds a pen-like object that attaches to the wrist joint. It is used in a fashion similar to holding a magic wand. Our method, however, does not depend on this device, and other haptic devices could be substituted. We use Softimage's XSI software as our animation software.

3. Trajectory Editing

In our system, when the user selects an object in the scene that has motion data, she or he is then presented with the option to edit the trajectory using the haptic device. If this option is selected, the motion trajectory is generated in the graphic scene as a NURBS curve, and at the same time, is rendered by the haptic device. The handle of the haptic device is then pulled onto the curve where it is free to explore the rendered curve. Picturing a bead on a wire is a good metaphor for the haptic interaction. The handle of the device can be thought of as the bead and the curve represents the wire. The bead is constrained to the wire, and with minimal force can explore the length of the wire. However, if the bead is pushed or pulled to hard, the wire will follow the bead, changing its shape, and will remain in the deformed position after the force has been released. The bead can now slide along the new curve of the wire. The device works in this manner to edit the NURBS curve. In our current implementation, the arc length of the curve is mapped to time in the animation. Thus as the user explores the haptic curve, he is also changing the animation time-line and viewing the edits in real-time.

3.1. Haptic Model

The user is constrained to the curve by a spring and damper force that increases proportional to the distance that the handle of the device deviates off of the curve. This force is given by Hooke's Law $f = Kx$ where x is the vector representing the position of the handle, with respect to the closest point on the curve. This is a common method used in haptic rendering.

The editing of the NURBS curve is triggered by changing a single point on the curve. This point, w , changes as the handle moves and is always the closest point on the curve to the handle position. We also consider w to be the position of the object that is being animated at time t . We use a basic friction model to control translations to the position of w and the transition from navigating to editing the curve [HA00]. The basic friction model has two states: pre-sliding adhesion and sliding. If the handle position x is exactly located on the curve, then no resistance is given and this represents the relaxed state. However, if x not on the curve, and yet is within a distance d_{\max} of the curve, then we are in the pre-sliding adhesion state. This is where we are building up tension, but are not yet actually moving w . Now we transition to the sliding state if $|x|$ exceeds d_{\max} , or back into the relaxed state if x returns to the curve. This elasto-plastic behavior is illustrated in Figure 2. There are many methods for editing NURBS. When editing, we have chosen to constrain the curve to pass through our new point w_t and to affect the minimum possible number of control points. We have used the GNU nurbs++ library to implement our curve calculations. The curve is plastic and after being changed, the device handle is then constrained to the new curve.

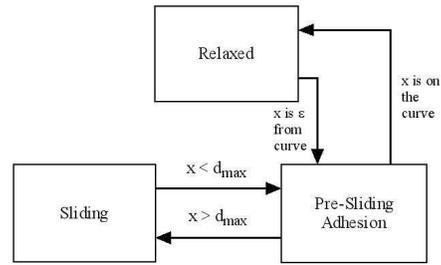


Figure 2: The transition between the states of the friction model for trajectory editing.

The position of w in 3D is governed by the update equation from time $t - 1$ to time t

$$w_t = \begin{cases} x_t - \frac{x_t - w_{t-1}}{|x_t - w_{t-1}|} d_{\max} & \text{if } |x_t - w_{t-1}| > d_{\max}, \\ w_{t-1} & \text{otherwise.} \end{cases} \quad (1)$$

3.2. Conclusions and Future Research

Our one-to-one mapping and intuitive “virtual yet physical” haptic model of the animation trajectories affords easy editing of motion curves. The friction model is easy to learn and provides us with a straightforward metaphor for editing 3D trajectories. Overall, although our system is still in its infant stages, we think haptic interaction has great potential to be used in the digital content creation industry and especially for animation editing. Our future direction will be to improve our selection of editing tools and then look at extending our NURBS curve editing system to include surfaces. This would allow the animator to make local deformations to NURBS surfaces in a very intuitive manner, enabling future interaction possibilities for specifying and editing shape animation.

References

- [DH00] DONALD B. R., HENLE F.: Using haptic vector fields for animation motion control. In *Proceedings of the 1997 IEEE international conference on Robotics and Automation* (2000), pp. 3435–3442. 1
- [HA00] HAYWARD V., ARMSTRONG B.: A new computational model of friction applied to haptic rendering. In *Experimental Robotics VI* (2000), Springer-Verlag, pp. 403–412. 2
- [HACH*04] HAYWARD V., ASTLEY O. R., CRUZ-HERNANDEZ M., GRANT D., ROBLES-DE-LA-TORRE G.: Haptic interfaces and devices. *Sensor Review* 24, 1 (Feb 2004), 16–29(14). 1