Haptics-based 3D B-spline Curve Sketch Using Multi-resolution and Two-handed Techniques

Zhan Gao, Ian Gibson

Department of Mechanical Engineering, the University of Hong Kong, Hong Kong, China
gaozhan@hkusua.hku.hk igibson@hku.hk

Abstract. : The ability to specify 3D curves is important in 3D modelling and animation systems. Effective techniques for specifying such curves using 3D input devices are desirable, but existing methods are typically based on 2D input and require the user to edit the curve from several viewpoints. We present a novel method for specifying 3D curves with 3D input with haptic feedback from a PHANToM device. The user first draws the curve in the 3D space with the haptic aids from PHANToM and visual aids of projected shadows. Then the user can manipulate and modify these curves with two-handed interface. User's left and right hands conduct different tasks according to specific features of both hands. Integrated operations of user's two hands and haptic feedback from PHANToM help user to anticipate the way curve deforms.

1 Introduction

In CAD/CAM design, parametric curves have become the standard for curve representation. Parametric curves have the advantage of compact representation and higher order continuity. Specifying 3D B-spline curves is one of the most important tasks that a 3D user interface must support. Most current methods processing sketches are based on 2D sketches. Freehand drawings on different views are automatically processed to generate reasonable 3D B-spline curves. The technique we present is used for specifying and manipulating 3D curves with 3D input and haptic feedback from PHANToM device.[1] Through a two-handed interface and multi-resolution curve editing techniques, integrated operations from the user's two hands and haptic feedback from the PHANToM help the user to anticipate the way the curve deforms.

2 Multi-resolution B-spline curve and wavelet filter banks

Several editing operations like fee form deformation (FFD), direct manipulation and finite element method are employed to generate or edit B-spline curves. However, even with these operations, it is sometimes difficult to achieve more complex editing and deformation. One difficulty is to arbitrarily choose the editing level of resolution. When details are to be added into a curve, the knot vector of the curve is usually refined and more control points are generated. However, after the refinement, the curve will not be edited globally any more.

Finkelstein suggested multi-resolution curve representation based on B-spline wavelets to support a variety of editing operations. [2] With multi-resolution editing, the curve may be smoothed and the overall form of the curve may be changed while preserving its details. Therefore, the curve may be edited at any continuous level of detail.

Let coefficients C^j be the control points of a B-spline curve at a resolution scale of j. A lower resolution version of curve C^{j-1} could be decomposed from C^j: $C^{j-1} = A^j \cdot C^j$ The lost information in the process of decomposition is stored in another matrix called D^{j-1} obtained as: $D^{j-1} = B^j \cdot C^j$. This process of obtaining C^{j-1} and D^{j-1} is called wavelet decomposition. The original coefficients C^j can be reconstructed from the lower scale coefficients C^{j-1} and the detail matrix D^{j-1}: $C^j = P^j \cdot C^{j-1} + Q^j \cdot D^{j-1}$. The process of recovering C^j from C^{j-1} and D^{j-1} is called wavelet reconstruction. The matrices A^j and B^j are called analysis filters. The matrices P^j and P^j are called synthesis filters.

With these transformations we can decompose and synthesis e the endpoint interpolating B-spline curves at different resolution scales. [2]

Sweep editing and fractional editing of B-spline curves have been introduced by Finkelstein and Salesin.[3] The capabilities of wavelet transformation to filter out detail features and recover detail features back provide an easy way for editing curves at any resolution. Suppose we have a curve with control point C^n and all of its low-resolution and detail parts $C^0, \cdots, C^{n-1}$ and $D^0, \cdots, D^{n-1}$. If we modify some low-resolution version C^j and then add back in the detail $D^j, D^{j+1}, \cdots, D^{n-1}$, we will have changed the overall sweep of the curve. If ΔC^j is denoted as the change to the control points at level j, then the total change at a higher level n is given by: $\Delta C^n = P^n P^{n-1} \cdots P^{j+1} \Delta C^j$. Changing a control point at a lower level will result in a change in a larger portion of the higher resolution curve. At the lowest level, when j=0, the entire curve is affected; at the highest level, when $j = n$, only the narrow portion is influenced.

3 Two-handed curve editing

3.1 Editing a desired portion

Interpolating the portions affected at levels j and j+1, curves can be affected at the fractional level of $m+1$, where $0 < m < 1$. As m changes from 0 to 1, the portion affected becomes smaller because we are editing the curve at a higher resolution level. The basic idea of fractional editing is to add the wavelet coefficients in a linear interpolation manner during the synthesis process, which is:

$$C^{j+m} = P^{j+1} \cdot C^{j} + mQ^{j+1} \cdot D^{j}$$

Fractional editing provides an efficient way for continuous editing at adjacent multiresolution levels.

The effect of editing a curve often depends on the parameterization of the curve. Dragging different points along the curve will not necessarily affect constant-length portions of the curve. For example, the influenced portion where control points are sparse would be longer than that where control points are dense. However, with multiresolution editing method, we can compensate for this defect. Let h be the desired length of editable portion of the curve. The system computes the suitable editing level l that will affect a portion of the curve of about h in length, centred at the point $g^{n}(t_0)$ being dragged.

Stollnitz suggests a method to estimate level l. [4] For any integer-level resolution j, let $h^{j}(t_0)$ denote the length of $g^{n}(t_0)$ affected by editing the curve at the point $g^{n}(t_0)$. Next, let j_- be the largest value of j with $h^{j_-}(t_0) \geq h$ and j_+ be the smallest value of j with $h^{j_+}(t_0) \leq h$. We use linear interpolation to find a suitable fractional editing level l that lies between j_- and j_+:

$$l := \frac{h - h^{j_+}}{h^{j_-} - h^{j_+}} j_- + \frac{h^{j_-} - h}{h^{j_-} - h^{j_+}} j_+$$

With l in terms of an integer level j and a fractional offset m, we can edit the curve changing only the desired portion.

3.2 Two-handed interface for curve editing

Guiard's three principles serve as guidelines for two-handed interface design. [5] They explain how our two hands work in harmony. The first principle is right-to-left reference: Motion of the right hand typically finds spatial reference in the results of motion of the left hand (for right-handed people). The second one is asymmetric scales of motion: The right and left hands are involved in asymmetric scales of motion. The right hand specializes in rapid, small-scale movements; the left, in slower, larger-scale movements. The last principle is left-hand precedence: The movement of left hand precedes the right hand.

Guiard's three principles generalize the human habit of working with two hands, with which a good two-handed human computer interface should accord. Under the framework of the principles, different tasks are designated to the preferred hand and the nonpreferred hand respectively. In our project, the PHANToM device is for the user's preferred hand and a Polhemus tracker (6DOF input) [0] for the non-preferred hand. The preferred hand holds the stylus of the PHANToM to perform the actual curve creating and editing tasks. Tasks of the nonpreferred hand include moving and rotating models. Using this method the non-preferred hand together with the preferred hand form a natural anticipation of where the curve is to be dragged and how long the affected portion is.

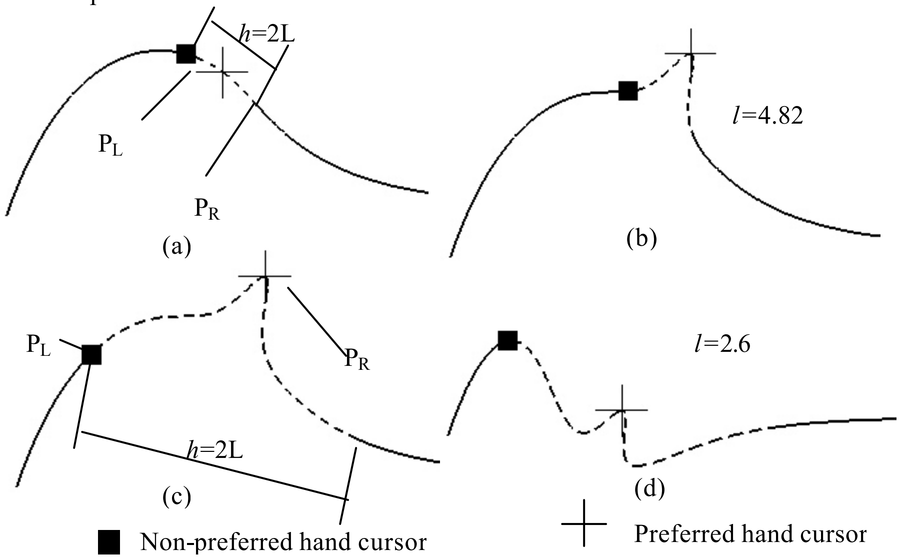


Fig. 1. Two-handed multi-resolution curve editing. (a) A curve at scale 5 with 35 control points. (b) The curve edited at resolution scale of 4.82. The affected portion is narrow. (c) Wider affected portion is expect ed by user. (d) The curve edited at resolution scale of 2.67.

We assume that before the actual editing operation the user has finished his mental work of how and where the deformation should be. According to the third principle of Guiard, the non-preferred hand initiates the editing operation by selecting a curve at point P_L. Then, the non-preferred hand may reorient the selected curve for the convenience of preferred hand to perform further operation. Next, the user selects a dragging point P_R on the curve with the preferred hand. The system then calculates the curve length between point P_L and P_R as L. Since point P_R is chosen as the dragging point, the user usually expects deformation at the dragging point P_R is the largest; therefore, it is reasonable to make point P_R as the midpoint of the affected portion and P_L as one of the endpoints of the affected portion. Let 2L be the desired length *h* we can find the fractional editing level *l*. Finally, the user directly manipulates the curve at the level *l*. In this way, the deformation of the portion close to the non-preferred hand point P_L is limited and constrained.

4 Haptic rendering

In this project, magnet model is chosen for haptically rendering static B-spline curves. We break the problem of rendering a B-spline curve into several phases. Curve proximity testing determines when the user's haptic interface point (HIP) is near enough to a curve to potentially contact it. When the HIP enters the boundary box of a curve, the curve is made active and the closest point on the curve is tracked along with user's PHANToM movement. Contact occurs as the distance between HIP and the closest point on curve falls below a certain threshold. An attraction force is computed using Hooke's law $f = kd$, where d is the distance between HIP and the closest point on curve. The direction of the attraction force is always from the HIP toward the closest point. With the magnetic curve model, exploring curves feels like touching magnetic curves with an iron cursor.

The deformation of curve is computed by ACIS [7] with least elastic energy and multi-resolution method. Haptic feedback during curve deformation makes the process more realistic. Force generated due to curve deformation has to be obtained to output to users. The curve is simplified as zero-weight to make the simulation more simple and processing less time consuming. The curve is discretized as joint points connected by a chain of springs and dampers. This representation is computationally simple and can be simulated using only basic physical equations.

The motion equation of all joint points using a discrete simulation of Lagrangian dynamics is $D\dot{d} + Kd = f_d$, where f_d is the external force, d is the displacement of a joint point, D is coefficient of damping and K is coefficient of elasticity. The force at every joint point is the sum of all possible external forces: $f_d = \sum f_{ext}$. The internal forces are generated by the connecting springs and dampers. The force from spring is computed with force $f = Kd$ while the force of damper as $f = D\dot{d}$. The rest length of each spring is determined upon initialization of deformation.

By giving each dragging point a displacement based on the user's actions, the curve deforms according to the physical principle of least strain energy. The strain energy of a curve is

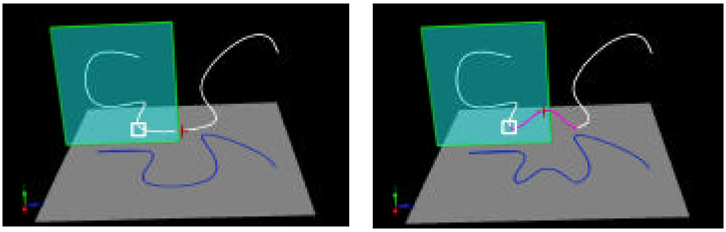
$$E = \int_{\Omega} a(s - s^0)^2 + b(k - k^0)^2 + g(t - t^0)^2 \, da$$

where a, b, g are the amount of resistance of the curve to stretching, bending, and twisting respectively.[8] The system constantly evolves the curve (governed by equations of least strain energy) in response to the user's operation. The system updates the shape of the curve at the rate of 30Hz. Throughout the entire editing process, the discrete model of joint points are constrained to the B-spline curve.

While the graphic display of the curve deformations requires an update rate of only 30Hz, the stable and smooth haptic display requires an update rate of at least 1000Hz. However, this discrepancy can be solved by interpolating the haptics status between two graphic deformation states using the necessary high frequency and delaying the entire curve deformation process, both graphically and haptically, by one graphic time step of about 1/30 seconds.

5 Implementation

The input hardware of this system includes a Phantom device and a Polhemus tracker. ACIS from Spatial is chosen as the geometry engine due to its excellent solid modelling and deformable modelling capabilities.[0] HOOPS from TechSoft is an optional graphics environment bundled with ACIS.



(a) (b)

Fig. 2. A cyan half-transparent vertical plane is seen as the virtual sketch board. The virtual sketch board can be manipulated using the PHANToM or Polhemus tracker. Curves can be drawn on the sketch board. A red 3D cross cursor is controlled by PHANToM device with the preferred hand, while a white box by Polhemus tracker with the non-preferred hand. (a) The user anticipates the desired portion through two hands interaction; (b) The desired span of curve is deformed.

6 Conclusion

We have presented a haptics-based interface and a 3D curve modelling system that facilitates the direct manipulation of multi-resolution based on wavelet theory and two-handed techniques. A magnet model and a discretized spring-damp model have been employed to haptically render static curves and curve deformations respectively. The 3D haptics-based interface is more intuitive and natural than conventional 2D mouse-based interfaces. Our system offers users a set of interaction toolkits, supporting point and tangent manipulation via haptic feedback devices. Users can manipulate and modify these curves with a specially designed two-handed interface. Through integrated operations of the user's two hands, users can anticipate the way a curve deforms.

Reference:

1. *PHANToM™*, SensAble Technologies, Inc., http://www.sensable.com
2. Ranga Narayanaswami and Ji Yan: Multiresolution modeling techniques in CAD. Computer-Aided Design, Volume 35, Issue 3, (2003) 225-240
3. Adam Finkelstein, David H. Salesin: Multiresolution curves. Proceedings of the 21st annual conference on Computer graphics and interactive techniques. (1994) 261-268
4. Eric J. Stollnitz, Tony D. DeRose, David H. Salesin: Wavelets for computer graphics. San Francisco, Calif. Morgan Kaufmann Publishers, (1996)
5. Guiard, Y.: Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. Journal of Motor Behavior, (1987) 19(4): 486-517.
6. *Polhemus FASTRAK*, Polhemus Inc. USA, http://www.polhemus.com/
7. *3D ACIS Modeler*, Spatial, Inc., http://www.spatial.com/
8. Demetri Terzopoulos, John Platt, Alan Barr, Kurt Fleischer: Elastically deformable models. Proceedings of the 14th annual conference on Computer graphics and interactive techniques. (1987) 205-214