NON-OVERSHOOTING HERMITE CUBIC SPLINES FOR KEYFRAME INTERPOLATION

Paul Heckbert

3-D Technical Memo 10
7 February 1985

Three Dimensional Animation Systems Group
Computer Graphics Laboratory
New York Institute of Technology

Abstract

A technique for limiting the knot slopes of hermite cubic splines in order to eliminate overshoot is proposed. It is proven that constraining each knot's slope to lie between 0 and three times the slope to the knot on either side forces all extrema to occur at knots. This allows overshoot to be eliminated without sacrificing slope continuity. The technique has applications in keyframe interpolation of motion parameters.

1. The Problem

Hermite cubic splines are often used for keyframe interpolation of motion parameters. Our *BBOP* program, for instance, uses local, independent non-uniform hermite cubics for each of the nine motion parameters (move, scale, and rot in x, y, and z) at each joint. Each parameter's spline is a univariate function $x(t)$ with first derivative (C^1) continuity but no guarantees of geometric continuity (see [Bohm, Farin, and Kahmann] for a survey of splines).

Users often complain of the overshoot which spline interpolation causes. Sudden changes in velocity force the curve to overshoot an extremum (a maximum or minimum) of the function as shown in figure 1. Animators often intend their key frames to be motion extremes, but the animation program usually ignores this implicit information. Is it possible to constrain a spline so that it doesn't have undesirable overshoots?

2. Previous Solutions

There are several existing partial solutions to the overshoot problem:

linear interpolation
Use of linear rather than cubic interpolation is an obvious but brute force way to eliminate overshoot. Continuity of velocity is sacrificed.

tense splines
A *tension* parameter can be used to interpolate between linear and cubic functions yielding a compromise shape. The proposed methods have ranged from global tension for 1-D (univariate) splines employing hyperbolic functions [Cline] to local tension for 3-D cubic splines [Kochanek and Bartels], [Duff]. Overshoot can be reduced by turning up the tension, but the resulting curves are not as smooth as one would like.

B-splines
The cubic B-spline is an approximating rather than interpolating spline which enjoys the *convex hull property*: the curve stays within the convex hull of the four local knots. The curve will therefore not overshoot, but unfortunately it does not interpolate the knots, so it is of little value for keyframe animation.

function editing
Rather than attempt to build intelligence into the interpolator by teaching it to guess which keys are extremes, perhaps it is best to leave the decision up to the animator. A function editor lets the user edit and redraw curves interactively, perhaps even allowing local control of interpolation function, tension, bias, and slope. These capabilities greatly enhance the flexibility of an animation system, but since function editors are regularly used to correct for parameter overshoot, it would be wise to fix the problem at its source, in the interpolator. This could reduce the tedium of function editing.

different slope formulas
Several formulas for automatic calculation of slopes for hermite splines have been proposed. One of these (figure 2a) generates the Catmull-Rom spline [Smith]. Some slope formulas cause more overshoot than others, as shown in figure 2. Figure 2c illustrates that *clamping* all knot slopes to 0 results in a non-overshooting, C^1 continuous spline. Unfortunately, this spline does not preserve straight lines, as seen toward the right side of that figure. It does prove, however, that linear interpolation is not the only dependable technique for eliminating overshoot. The technique described in the following section loosens this spline's constraints as much as possible by allowing small slopes.

Figure 2: Three hermite knot slope formulas.

3. Slope Constraints

The new method constrains the slope of the curve at each knot in order to eliminate overshoot without sacrificing continuity. Obviously, knots which are local maxima or minima of the knot sequence must be constrained to slope 0 to eliminate overshoot. Knot B in figure 2a shows that this constraint is not sufficient in all situations. It is clear that the slopes at knots A and C in that figure must be limited also. How do we find the constraints in general?

Figure 3: Canonical hermite cubic

It helps to abstract the situation by considering the interval [0,1] and the *canonical hermite cubic*: any cubic which passes through (0,0) and (1,1) as in figure 3. Any function $y=f(x)$ through these two points can be scaled and offset to pass through an arbitrary pair of points (x_0,y_0) and (x_1,y_1) with the transformation:

$$y' = g(x') = y_0+(y_1-y_0)f\left(\frac{x'-x_0}{x_1-x_0}\right)$$

We will call the slopes at 0 and 1 p and q. We expect to find that to guarantee no overshoot p must be between 0 and some maximum slope $p_{\max}$ which is a function of q, and likewise q must be between 0 and some maximum $q_{\max}$ which is a function of p. Figure 4 shows examples of non-overshooting splines with maximum slope at q: $q=q_{\max}(p)$ which demonstrate that $q_{\max}(p)$ is not a constant function.

Figure 4: Non-overshooting hermites which maximize $f'(1)=q$.

We would like to constrain the slopes as little as possible so that one of the slope formulas mentioned earlier can be used at most knots, and only knots near sharp velocity changes will be affected.

However, if a knot's slope constraints are dependent on its neighboring knot's slopes, which are in turn constrained by their neighbor's slopes, etc, the spline is no longer local. Therefore, p_{max} must be independent of q and q_{max} independent of p. To ensure locality, the constraints must depend on the width Δx and height Δy of the interval only (which are both 1 for the canonical interval). By symmetry, any non-overshooting function $f(x)$ can be mapped into a non-overshooting function $1-f(1-x)$ by rotating it 180 degrees about (1/2,1/2). Therefore, the constraints at the two ends of the interval must be the same: $p_{max}=q_{max}$. Each knot must satisfy constraints for the intervals on both sides. A knot which is a local maximum, for instance, must have a zero or positive slope according to its left constraint but it must have a zero or negative slope according to its right constraint. The only solution is a slope of 0 with no slack.

In order to find p_{max} we will imagine the set of all canonical hermite cubics as a function space with two degrees of freedom. Given the cubic

$$f(x) = a+bx+cx^2+dx^3$$

We can eliminate two degrees of freedom with the conditions:

$$f(0)=a=0$$
$$f(1)=b+c+d=1 \quad \Rightarrow \quad b=1-c-d$$

The knot slopes are:

$$\begin{aligned} p &= f'(0) = b = 1-c-d \\ q &= f'(1) = b+2c+3d = 1+c+2d \end{aligned} \qquad \text{or conversely} \qquad \begin{aligned} c &= 3-2p-q \\ d &= p+q-2 \end{aligned}$$

Each ordered pair (c,d) or (p,q) defines a hermite cubic function. Either function space may be used; the latter is more convenient, however. We wish to find the locus of points in pq space defining non-overshooting hermite cubics. A *non-overshooting function* over an interval is defined to be one with no extrema within the interval (except perhaps at the knots). Extrema are located by finding the roots of the derivative function:

$$f'(x) = b+2cx+3cx^2 = 0$$

$$\text{roots are: } x = \frac{-c\pm\sqrt{\xi}}{3d}$$

$$\text{where } \xi = \textit{discriminant} = c^2-3bd = p^2+pq+q^2-6p-6q+9$$

Note that $\xi(p,q)$ is symmetric in p and q. The locus $\xi=0$ defines an ellipse in pq space inside of which lie all canonical hermite cubics which have no roots in their derivatives and hence no extrema (figure 5).

Figure 5: The ellipse $\xi(p,q)=0$ of hermites with a saddle point.

Figure 6: The derivatives of canonical hermites with no roots in [0,1]; four cases.

Except in the degenerate case $d=0$, the derivative function is a parabola. There are four cases in which it has no roots in the interval [0,1], as shown in figure 6. The locus of points satisfying each case defines a region in *pq* function space. The union of these regions will be the set of all canonical hermite cubics without internal extrema.

Case 1: concave up, no roots

Concave up implies $d>0$ and no roots implies $\xi<0$, so this is the locus of points inside the ellipse.

Case 2: concave up, roots to left

Concave up implies $d>0$ and roots to the left of 0 implies $\xi>0$ and $x_+ = \dfrac{-c+\sqrt{\xi}}{3d} < 0$ which means $2p+q<3$ and $p>0$.

Case 3: concave up, roots to right

Concave up implies $d>0$ and roots to the right of 1 implies $\xi>0$ and $x_- = \dfrac{-c-\sqrt{\xi}}{3d} > 1$ which means $p+2q<3$ and $q>0$.

Case 4:: concave down, roots straddle [0,1]

Concave down implies $d<0$ and straddling roots implies $\xi>0$ and $x_- = \dfrac{-c+\sqrt{\xi}}{3d} < 0$ which means $p>0$ and $x_+ = \dfrac{-c-\sqrt{\xi}}{3d} > 1$ which means $q>0$.

The four regions are shown in figure 7. From this diagram it is clear that any canonical hermite cubic within the square $0<p,q<3$ will have no extrema within the interval [0,1]. Generalizing this to an arbitrary interval, constraining $f'(x_0)$ and $f'(x_1)$ to between 0 and $3\dfrac{\Delta y}{\Delta x}=3\dfrac{y_1-y_0}{x_1-x_0}$ guarantees that the spline will not overshoot.

4. Applying the Constraints

Adding these slope constraints to a keyframe interpolator is quite simple. When slopes are calculated at each key, they must be checked against the maximum slope constraints $3\frac{\Delta y}{\Delta x}$ for the intervals on the left and right. If the desired slope is within the constraint, no change is necessary, but if not then the slope must be clamped to the constraint value. The method works for uniform or non-uniform knot spacing. Figure 8 illustrates the effectiveness of the constraints in eliminating overshoot.

Non-overshooting splines are also useful when interpolating a parameter with a bounded interval, such as color coordinates which are constrained to lie between 0 and 255. If all of the knots are within these bounds, the spline will be also; it obeys a global convex hull property.

Figure 8: Slope formula 2b with and without constraints. Constrained intervals shown bold.
Min and max slopes are indicated on both sides of each knot.

5. Further Work

In some cases overshoot is desirable, so ideally the user should be able to turn off the constraints described above or to use them locally, perhaps as part of a function editor.

It has been suggested that a serious flaw of splines is their tendency to generate extraneous inflection points [Cline]. Figure 9 is a *pq* function space diagram showing which hermite cubics have no inflection point within the interval [0,1]. Given a criterion for deciding which inflection points are ''extraneous'', it might be possible to eliminate them using slope constraints.

Figure 9: The *pq* space regions corresponding to hermites
with no inflection point in the interval [0,1].

6. References

Bohm, Wolfgang, Gerald Farin, and Jurgen Kahmann, ‘‘A Survey of Curve and Surface Methods in CAGD’’, *Computer Aided Geometric Design,* vol. 1, 1984, pp. 1-60.

Cline, A. K., ‘‘Scalar- and Planar-Valued Curve Fitting Using Splines Under Tension’’, *C.A.C.M.,* vol. 17, no. 4, April 1974, pp. 218-223.

Duff, Tom, ‘‘Families of Local Matrix Splines’’, Technical Memo #104, Lucasfilm, Jan. 1984, reprinted in *SIGGRAPH ’84 The Mathematics of Computer Graphics seminar notes,* July 1984.

Kochanek, Doris H. U., and Richard H. Bartels, ‘‘Interpolating Splines with Local Tension, Continuity, and Bias Control’’, SIGGRAPH ’84 Proceedings, *Computer Graphics,* vol. 18, no. 3, July 1984, pp. 33-41.

Smith, Alvy Ray, ‘‘Spline Tutorial Notes’’, Technical Memo No. 77, Lucasfilm, May 1983, reprinted in *SIGGRAPH ’83 Introduction to Computer Animation seminar notes,* July 1983, pp. 64-75.