INTERPOLATING SPLINES FOR KEYFRAME ANIMATION

Doris H. U. Kochanek

French Animation Studio, P-36
National Film Board of Canada
P.O. Box 6100, Station A
Montreal, P.Q. H3C 3H5
(514) 333-3434

Richard H. Bartels

Computer Graphics Laboratory
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
(519) 886-1351

ABSTRACT

This paper presents a new method for representing interpolating splines for use in a keyframe animation system. Three control parameters allow the animator to change the tension, bias and continuity of the default interpolating spline. Each of the three parameters can be used for either local or global control. Due to their high degree of flexibility this class of splines has potential applications in other areas, such as CAD/CAM, modeling of dynamic phenomena, and robotics.

1. Summary

Automatic inbetweening based on keyframes drawn by the animator is one of the oldest techniques used in computer animation [Burtnyk & Wein 71]. The intermediate drawings required to move from one keyframe to the next are generated by interpolating between the given keys. The straightforward linear interpolation algorithm used in many inbetweening systems produces some undesirable side effects which give the animation a mechanical look, often referred to as the computer signature. The most objectionable characteristic of this type of animation is the lack of smoothness in the motion. The keyframes are clearly visible because of sudden changes in the direction and speed of motion at each key position. Another common problem is the distortion which occurs whenever the movement has a rotational component.

In view of the serious drawbacks of the simple linear interpolation technique, a number of different methods which produce smoother motion have been published. These techniques include P-curves [Baecker 69], skeletons [Burtnyk & Wein 76], action overlap [Tuori 77], and moving point constraints [Reeves 81]. All of these techniques require the animator to specify additional information other than just the keyframes; a completely automatic technique which uses only the keyframes drawn by the animator is described in [Kochanek 82]. The system described there is based on fitting a set of interpolating splines through the key positions, thus ameliorating the discontinuity problems produced by linear interpolation.

The splines used in that system work well for most animated sequences. However, a standardized smooth motion through a given set of keys does not always produce the effect desired by the animator. In certain cases the animator may want the motion to follow a wider, more exaggerated curve, while in other cases he may want the motion path to be much tighter, maybe almost linear. Even continuity in the direction and speed of motion is not necessarily desirable at all times. Animating a bouncing ball, for example, actually requires the introduction of a discontinuity in the motion at the point of impact. The research described in this paper replaces the standardized interpolating spline used in [Kochanek 82] by a highly flexible class of splines which interpolate the key positions but vary in several control parameters.

These three parameters, tension, bias, and continuity, allow the animator to fine-tune the animated sequence by changing certain characteristics of the standardized interpolating spline either locally (applying only in the vicinity of a specific keyframe), or globally (applying to the entire motion sequence). The research presented here is based in part on the concepts of tension and bias in approximating splines presented in [Barsky 81], [Barsky & Beatty 83], and [Bartels et al. 83]. An excellent introduction to the theory of interpolating and approximating splines for computer animation can be found in [Smith 83].

Given a sequence of key positions, we can interpolate them with a piecewise cubic polynomial (a spline) by choosing two constraints for each interpolation interval. Together with the interpolation requirement (the curve must pass through both of the keys which define the interval), this gives a total of four constraints which uniquely determine the four coefficients of the desired cubic polynomial for each interval.

First and second derivative continuity are a possible choice for these constraints. However, this technique is computationally complex and not very flexible. Instead, we define the tangent vectors at each key position directly, based on the geometry of the surrounding keys. This approach allows us to provide first derivative continuity where desired, yet leaves enough flexibility to generate more than one standard curve. The resulting splines do not have

second derivative continuity. This does not prove to be a significant problem because the curves are never actually seen in graphic form, but only indirectly through the dynamic behaviour of an object. In the time dimension discontinuities of the second derivative are almost impossible to detect.

The high degree of flexibility in our approach stems from the fact that any piecewise cubic polynomial interpolating the key positions can be generated by choosing appropriate tangent vectors. A single keyframe may contain several thousand points, making it infeasible to ask the animator to specify the tangent vectors directly. Instead the system defines the vectors based on the surrounding key positions and provides several control parameters which supply the desired flexibility. For keyframe n the tangent vector for a point P is calculated as a linear combination of the *source chord* vector SC and the *destination chord* vector DC These two chords are usually not co-linear. Thus they form the basis of a two-dimensional vector space, and any vector in the same plane can be generated from a linear combination of these two chords. A default curve is defined which is based on tangent vectors produced by simply averaging the two adjacent chord vectors. This default case, even though expressed in terms of different basis functions, is exactly the Catmull-Rom spline described in [Smith 83].

The first control parameter is *tension* which is used to increase or decrease the tightness of the curve. The tension parameter modifies the length of the tangent vectors, without changing their direction. Increasing the tension shortens the length of the tangent vectors, thus producing a tighter curve. Varying the tension equally for all key positions generates the entire class of cardinal splines of which the Catmull-Rom spline is one particular example [Smith 83].

The second control parameter is *bias* which is used to assign different weighting factors to the two chords when averaging them to find the tangent vector. In the extreme case the tangent vector is determined entirely by one of the chords, whereas the default is an equal weighting factor for both. The bias control is very useful for producing the traditional animation effects of overshoot and follow through for an action.

The third control parameter is *continuity*. By default the splines have tangent vector continuity, i.e. the tangent vector when approaching a key position is equal to the tangent vector when leaving the same key position. By changing the continuity parameter the animator can force two different tangent vectors for approaching and leaving a key position, thus deliberately introducing kinks into the curve.

The three control parameters add a degree of flexibility to the curve which has previously been found only in approximating Beta-splines. The class of interpolating splines described in this way was developed for keyframe animation, but their potential range of application is far more general, including areas such as modeling of dynamic phenomena, tool path definition in computer-aided manufacturing, and the description of complex motion sequences in robotics.

2. Bibliography

[Baecker 69]

R. Baecker, "Interactive Computer-Mediated Animation", PhD Thesis, MIT, Project MAC Technical Report MAC-TR-61, 1969.

[Barsky 81]

B. Barsky, "The Beta-spline: A Local Representation Based on Shape Parameters and Fundamental Geometric Measures", Ph. D. dissertation, Department of Computer Science, University of Utah, December, 1981.

[Barsky & Beatty 83]

B. Barsky and J. Beatty, "Local Control of Bias and Tension in Beta-Splines", *Computer Graphics (SIGGRAPH '83)*, **17 (3)**, pp. 193-218, July, 1983.

[Bartels et al. 83]

R. Bartels, J. Beatty, and B. Barsky, "An Introduction to the Use of Splines in Computer Graphics", Department of Computer Science, University of Waterloo, TR CS-83-09, August 1983.

[Burtnyk & Wein 71]

N. Burtnyk and M. Wein, "Computer Generated Key Frame Animation", *Journal of the SMPTE 80*, pp. 149-153, March, 1971.

[Burtnyk & Wein 76]

N. Burtnyk and M. Wein, "Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation", Communications of the ACM **19 (10)** pp. 564-569, October, 1976.

[Kochanek 82]

D. Kochanek, "A Computer System for Smooth Keyframe Animation", MMath Thesis, Department of Computer Science, University of Waterloo, Technical Report CS-82-42, December, 1982.

[Reeves 81]

W. Reeves, "Inbetweening for Computer Animation Utilizing Moving Point Constraints", *Computer Graphics (SIGGRAPH '81)*, **15 (3)**, pp. 263-269, August, 1981.

[Smith 83]

A. Smith, "Spline Tutorial Notes – Technical Memo No. 77", *SIGGRAPH '83 Tutorial Notes: Introduction to Computer Animation*, pp. 64-75, July, 1983.

[Tuori 77]

M. Tuori, "Tools and Techniques for Computer-aided Animation", MSc Thesis, Department of Computer Science, University of Toronto, 1977.