(1)

Geometric Continuity with Interpolating Bézier Curves

Extended Summary → Preliminary Report

Alain Fournier

Computer Systems Research Institute
Department of Computer Science, University of Toronto
Toronto, Ontario
M5S 1A4

Brian A. Barsky

Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, California 94720

ABSTRACT

The Bézier formulation for parametric curves has many qualities, among them the intuitive relationship between the shape of the *control polygon* and the shape of the curve, and the ease of computation and subdivision. Other formulations, however, have become more popular because they offer local control, or because they are interpolating, or even more recently because they provide the added flexibility of *shape parameters*.

We present here techniques to use the Bézier formulation to interpolate the two-dimensional points given by a user with cubic piecewise Bézier curves, while maintaining up to $G^{[2]}$ continuity, and to interactively manipulate the *bias* and *tension* of each span, with geometric entities clearly related to the curve, while preserving the degree of geometric continuity prescribed by the user.

RESUME

La mèthode de Bézier pour dèfinir des courbes paramètriques a de nombreuses qualités, parmi lesquelles la relation intuitive entre la forme de la courbe et la forme du *polygone de controle*, et la facilité avec laquelle les courbes sont calculées et subdivisées. D'autres méthodes, cependant, sont devenus plus populaires parce qu'elles permettent le controle local, parce qu'elles interpolent, ou bien plus récemment parce qu'elles permettent de plus la possibilité de *paramètres de formes*.

Nous prèsentons ici des techniques pour utiliser la mèthode de Bézier pour interpoler les points en deux dimensions donnés par l'utilisateur avec des arcs de courbes de Bézier, tout en maintenant la continuité gèmètrique jusqu'à $G^{[2]}$. Le système permet aussi à l'utilisateur de manipuler de façon interactive par l'intermédaire d'objects géomètriques intuitivement reliés aux propriétés dsirées le *biais* et la *tension* de la courbe obtenue.

KEYWORDS: geometric modelling, Bézier curves, geometric continuity, font design systems.

1. Bézier Curves

The use of piecewise *spans* of parametric polynomial curves joined together under some continuity constraints to design curves (and surfaces) goes back to Coons and Bézier. Later, Riesenfeld used B-splines for the same purpose. Each of the various formulations used have qualities and drawbacks, but behind all is the desire to provide the user (the designer) with a more direct and more intuitive control of the shape of the curves.

Bézier curves use a simple and efficient formulation where the curve is defined solely in terms of a set of points called *control vertices* connected in a sequence to form a *control polygon* (Figure 1). The curve mimics the overall shape of the control polygon, but interpolates only the first and last vertices of the control polygon. The curve is defined by a polynomial whose degree is equal to the

(3)

geometric characteristics of the Bézier formulation.

3. Parametric and Geometric Continuity in Bézier Curves

The relationships between the parametric derivatives curves and the control vertices are simple and well known [Faux79]. For example, for cubic curves, and the first two derivatives, assuming the spans defined above have the control vertices $\{W_i\}$ and $\{V_i\}$ (Figure 3):

$$\mathbf{R}^{(1)}(t_1)=3(W_3-W_2)$$

$$\mathbf{Q}^{(1)}(u_0)=3(V_1-V_0)$$

R $$\mathbf{P}^{(2)}(t_1)=6(W_1-2W_2+W_3)$$

$$\mathbf{Q}^{(2)}(u_0)=6(V_0-2V_1+V_2)$$

To ensure geometric continuity ($G^{[2]}$) the preceding relationships together with the beta constraints given above give the following relationships for the vertices of span **Q** as a function of the vertices of span **R**:

$$V_0=W_3$$

$$V_1=(1+\beta_1)W_3-\beta_1 W_2$$

check $\Rightarrow$ $$V_2=\beta_1^2 W_1-(2\beta_1^2+2\beta_1+\frac{\beta_2}{2})W_2+(\beta_1^2+2\beta_1+1+\frac{\beta_2}{2})W_3$$

In other words, only V_3 can be freely chosen if we insist on $G^{[2]}$ geometric continuity **once** β^1 and β^2 are chosen. The crucial point is that the relaxation of the continuity constraints gives us two more degrees of freedom. In particular, we can adjust β_1 and β_2 to be able to ensure $G^{[2]}$ on both sides of the span. The usefulness of the Bézier formulation is in the fact that the control vertices and the shape parameters are directly and simply related. [Rams85] discusses the issues of parametrizations, geometric continuity and geometric constructions in Bézier formulations.

4. The Interactive System

The user first interactively inputs a set of two-dimensional points. These are the points that will be interpolated by the system, and the points the user will see most of the time and use to modify the basic shape of the curve. The system then draws a smooth interpolating curve through the points. The default method is to fit the cubic interpolating version of Catmull-Rom splines [Catm74]. In order to accomplish this while interpolating the end points, more information (two values per end point) has to be provided. One solution is to ask the user for "phantom" vertices. But it is better to have the system provide "reasonable" extra vertices on its own. A good solution when the intended curves have definite symmetries is to take the mirror image of the third point through the perpendicular bisector to the first segment (and similarly at the end; see Figure 4). The curve obtained is only $C^{[1]}$ continuous at the joints. More continuity could be provided (actually the curve could be made $G^{[2]}$ continuous everywhere) but that would be, in general, at the price of undesirable variations between the joints. Note also that $C^{[1]}$ continuous means $G^{[1]}$ continuous with $\beta_1=1.0$.

The next step is to create the extra control vertices that define Bézier spans identical with the spans already computed. It is a straighforward change of basis. To simplify the terminology, we will call ***points*** the points originally defined by the user. Note that they are also joints between the spans. We will call ***control vertices*** the added vertices of the Bézier control polygons. Of course the joints are also control vertices. We will mention them explicitly if need be (Figure 5).

Now the user has available two kinds of control: shape and shape parameters. Shape is controlled through the points (joints) and shape parameters through the control vertices. The system maintains the continuity required by default or as specified by the user. We will illustrate some possibilities.

The original points given can be manipulated to change the overall shape of the curve. These changes can be made while respecting the local continuity that the user wants respected. Note that these changes are only local, since they affect only a small number of spans. Four spans are affected for $C^{[2]}$ continuity, two for $C^{[1]}$ or $G^{[2]}$ continuity. Of course the user can choose to split the curve at this joint. While acting on the original points, in general the control vertices are not even displayed. They are automatically updated by the system to maintain the required constraints.

The other group of options is to manipulate the control vertices to influence the shape parameters of the curve at a joint. In this

(2)

number of edges in the control polygon (that is the number of control vertices minus one). It follows immediately from this definition that the formulation has *global* not *local* control; that is the motion of a control vertex affects the shape of the entire curve. Likewise the curve is infinitely differentiable by virtue of being a polynomial. A Bézier curve $\mathbf{Q}(u)$ defined by a set of control vertices $\{\mathbf{V}_i\}$ is given by:

$$\mathbf{Q}(u) = \sum_{i=0}^{d} \mathbf{V}_i \mathbf{B}_i(u)$$

where d is the degree of the curve and B_i are the binomial coefficients:

$$B_i(u) = \binom{n}{i} u^i (1-u)^{n-i} \quad , i = 0,\dots,d$$

The Bézier formulation has numerous advantages: the shape is better related to the control points than in others, there is an easy geometric construction for the curves, and splitting a curve into two spans is also geometrically easy.

It is frequently desirable to decouple the number of control vertices from the degree of the curve, and to have local control as well. In the Bézier formulation it is easy to raise the degree of the curve, by creating a new set of vertices that generate the exact same curve with a (degenerate) polynomial of higher order (Figure 2). If we have the set $\{\mathbf{V}_i\}$ with $i = 0,\dots\dots,d$, the following formula gives the set $\{\mathbf{W}_i\}$ of control vertices with $i = 0,\dots\dots,d+1$:

$$\mathbf{W}_i = \left(\frac{i}{d+1}\right)\mathbf{V}_{i-1} + \left(1 - \frac{i}{d+1}\right)\mathbf{V}_i \qquad i = 0,\dots\dots,d+1$$

To obtain local control, we use a piecewise representation of the curve. The entire curve is composed of *curve segments*, each of which is a Bézier polynomial. The common point between them is called a *joint*. The problem we face now is to maintain some amount of *continuity* at the joints.

2. Parametric and Geometric Continuity

The smoothness of a piecewise curve has traditionally been measured by maintaining *parametric continuity* at the joints; that is, a continuity in the parametric derivatives on both sides of the joint. Parametric continuity is usually noted $C^{[n]}$ where n is the order of the parametric derivative which is equal on both sides of the joint. Recent work has shown that parametric continuity is overly restrictive; more relaxed constraints of *geometric continuity* have recently been proposed [Bars81, Bars83, DeRo85]. The key insight is that the traditional measure of continuity is affected by reparametrization. Geometric continuity provides a metric which is independant of the parametrization. The use of geometric continuity entails the use of *shape parameters* which provide further control of shape above and beyond that of control vertices. β_1 is known as *bias* and measures the relative influence of the tangent direction at a joint on each span. β_2 is called *tension* and controls the sharpness or flatness of the curve. These parameters appear in the equations of geometric continuity which are hence referred to as *beta constraints*. When two shape parameters are used, the continuity is denoted $G^{[2]}$ continuity. The reader is referred to [DeRo85] for a generalization of geometric continuity to any order $G^{[n]}$ and for the principle to construct the beta constraints of any order based on the chain rule of calculus. There are domains of application where parametric continuity is important, since the parametrization is directly related to the speed along the curve. But in other applications, only the shape is important, and this is where geometric continuity is the relevant concept.

For geometric continuity up to $G^{[3]}$, the necessary relationships are, given that the span $\mathbf{R}(t)$ has a common joint with the span $\mathbf{Q}(u)$ for the values of the parameters t_1 and u_0:

$$\mathbf{R}^{(0)}(t_1) = \mathbf{Q}^{(0)}(u_0)$$

$$\mathbf{R}^{(1)}(t_1) = \beta_1 \mathbf{Q}^{(1)}(u_0)$$

$$\mathbf{R}^{(2)}(t_1) = \beta_1^2 \mathbf{Q}^{(2)}(u_0) + \beta_2 \mathbf{Q}^{(1)}(u_0)$$

$$\mathbf{R}^{(3)}(t_1) = \beta_1^3 \mathbf{Q}^{(3)}(u_0) + 3\beta_1\beta_2 \mathbf{Q}^{(2)}(u_0) + \beta_3 \mathbf{Q}^{(1)}(u_0)$$

The formulations to which geometric continuity has been applied so far, besides being somewhat expensive to compute, suffer from another problem: the extra freedom given by the new parameters β_1 and β_2 is not intuitively related to some geometric entity to be manipulated by the user, but is controlled by setting some *dials* provided by the program [Koch84]. This is counter to the general philosophy of parametric curves. We propose here to remedy this by the use of the case, the vertices are visible, but move only according to the continuity constraints required. For instance, to manipulate the bias at a joint, while respecting $G^{[1]}$ continuity, the user only has to slide the control vertex along the line defined by the tangent. The program automatically maintains the vertex on this line (Figure 6). To adjust the tension, while for example keeping tangent continuity at each end of the span, the user grabs the segment defined by the two middle control vertices, and moves it closer to or farther from the segment defined by the two original points (Figure 7).

The fact that the shape and the shape parameters are locally controlled, and that they are controlled through geometric entities visibly related to their effects is a great help in the interaction. The problem with local control of shape parameters is the often undesirable behaviour of the curves between the joints. By having a "visible" control of shape, the user is better able to avoid these problems, and to understand them if they occur.

5. Conclusions

Bézier curves are easier to compute and to subdivide than most formulations. They also allow easy control over geometric continuity by the manipulation of the control vertices. The scheme we presented here exploits these properties to give the user explicit and intuitive control over bias and tension with interpolating curves.

The scheme is currently limited to planar curves. There are many areas of applications where good shape control in the plane is useful. One of the most important ones is font design. Some systems have already been implemented using parametric curves [Plas83, Knut79, Pavl84]. It is our intention to implement a prototypical system to apply the techniques described in this paper to that particular task.

Acknowledgements

We gratefully acknowledge the support of the Natural Sciences and Engineering Research Council. Mark Corbould has implemented the prototype system, and contributed many ideas, and Avi Naiman provided some of software for the interface and wise advice.

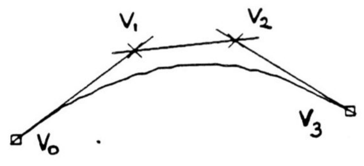

Figure 1. Bézier curve and its control polygon.

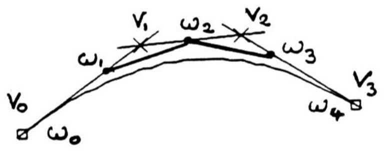

Figure 2. Raising the degree of a Bézier curve from cubic to quartic.

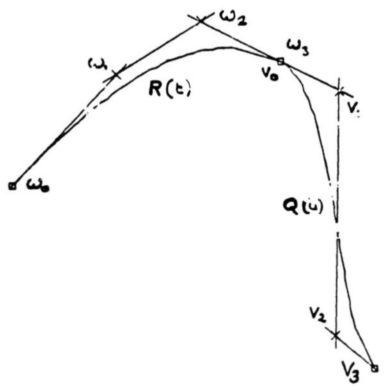

Figure 3.Continuity constraints at the joints.

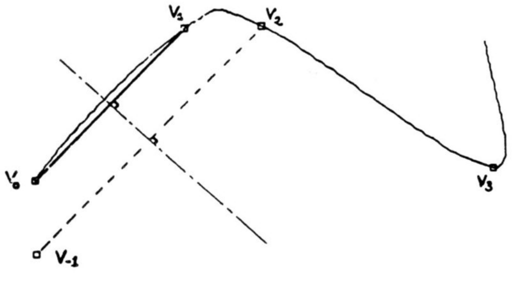

Figure 4.Creating the phantom vertex by symmetry.

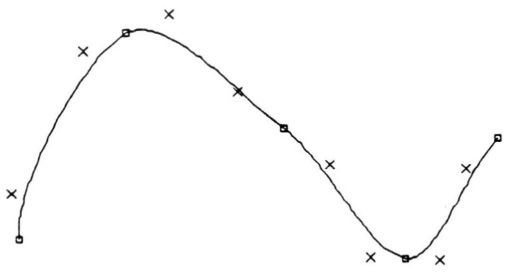
Figure 5. Interpolated points and control vertices.

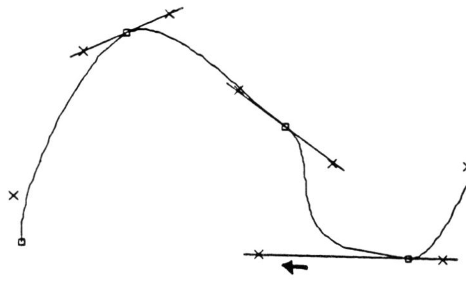
Figure 6. Controlling the bias at a joint.

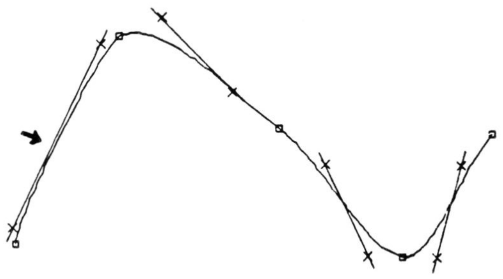
Figure 7. Controlling the tension.

References

[Bars81]
Barsky, B. A., *The Beta-Spline: A Local Representation Based on Shape Parameters and Fundamental Geometric Measures*, Ph. D. Thesis, Department of Computer Science, University of Utah, Salt Lake City, Utah, December 1981.

[Bars83]
Barsky, B. A. and J. C. Beatty, "Local Control of Bias and Tension in Beta-Splines", ACM Transactions on Graphics, Volume 2, Number 2, April 1983.

[Catm74]
Catmull, E. E. and Rom, R. J., "A Class of Local Interpolating Splines", *Computer Aided Geometric Design*, Barnhill, R. E. and Riesenfeld, R. F., Eds, Academic Press, (1974), 317-326.

[DeRo85]
DeRose, T. D. and Barsky, B. A., "An Intuitive Approach to Geometric Continuity for Parametric Curves and Surfaces", ~~to appear~~ m ~~in~~ *Proceedings of Graphics Interface* '85, May 1985. add refer -

[Knut79]
Knuth, D. E., *Metafont: A System for Alphabet Design*, Technical Report STAN-CS-79-762, Department of Computer Science, Stanford University, Stanford, California, September 1979.

[Koch84]
Kochanek, D. H. and R. H. Bartels, "Interpolating Splines with Local Tension, Continuity, and Bias Control", SIGGRAPH 1984 Proceedings, published as *Computer Graphics*, Volume 18, Number 3, July 1984, 33-41. ? no ital

[DeRo84]
DeRose, T. D. and B. A. Barsky, "Geometric Continuity and Shape Parameters for Catmull-Rom Splines", in *Proceedings of Graphics Interface 1984*, Ottawa, Ontario, May 1984, 57-62.

[Faux79]
Faux, I. D. and M. J. Pratt, *Computational Geometry for Design and Manufacture*, John Wiley and Sons, 1979.

[Pavl83]
Pavlidis, T., "Curve Fitting with Conic Splines", ACM Transactions on Graphics, Volume 2, Number 1, January 1983, 1-31.

[Plas83]
Plass, M. and Stone, M., "Curve-Fitting with Piecewise Parametric Cubics", SIGGRAPH 1983 Proceedings, also published as Computer Graphics, Volume 17, Number 3, July 1983, 229-239.

[Rams85]
Ramshaw, L., "A Euclidean View of Joints between Bézier Curves", to appear in ACM Transactions on Graphics.