Parametric

Bernstein/Bézier

Curves and Tensor Product Surfaces

Brian A. Barsky

Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, California 94720
U.S.A.

ACM/SIGGRAPH'90 Course 25
Unifying Parametric and Implicit Surface Representations for Computer Graphics

Dallas, Texas
7 August 1990

Parametric
Bernstein/Bézier
Curves and Tensor Product Surfaces

Brian A. Barsky†

Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, California 94720
U.S.A.

University of California at Berkeley Technical Report No. UCB/CSD 90/571
May 1990

This tutorial describes parametric Bernstein/Bézier curves and parametric tensor-product Bernstein/Bézier surfaces. The parametric representation is described, the Bézier curve representation is explained, and the mathematics are presented. The key properties of Bézier curves are discussed.

The single Bézier curve is extended to a composite Bézier curve using parametric continuity. Then the more general geometric continuity is defined, first for order two (G^2), and then for arbitrary order n (G^n). Composite Bézier curves are stitched together with G^1 and G^2 continuity using constraints on the control vertices and using geometric constructions.

The subdivision of Bézier curves is then derived along with a discussion of the associated geometric construction, the deCasteljau Algorithm, and flatness testing.

Then, the Bézier curve is generalized to a tensor-product surface. Finally, the rational Bézier curve and rational tensor-product surface are discussed.

† Supported in part by a National Science Foundation Presidential Young Investigator Award (number CCR-8451997).

1. Parametric Representation

The traditional explicit functional form (such as $y = f(x)$) has various drawbacks. For instance, the representation of a multiple-valued curve (such as a circle) requires splitting the curve into various segments. And this splitting of the curve might have to be recomputed if the curve were rotated. Furthermore, infinite values would arise in the representation of vertical tangents. Some of these problems could be addressed using implicit functions (having the form $f(x,y) = 0$). However, this has shortcomings in the evaluation of a particular point, the calculation of derivatives, and the specification of a portion of a shape (such as a semi-circle).

To address these problems, our piecewise polynomials will be formulated using a *parametric representation*. In this form, each coordinate is represented by its own separate, independent function. Continuing with the example of a circle, this could be easily represented as $(r\cos\theta, r\sin\theta)$. However, the parametric representation is not a panacea; although it addresses the problems outlined above, it does introduce an additional level of complexity. For example, derivatives cease to be *scalar-valued*, but become *vector-valued*. That is to say that the n^{th} derivative is a vector whose components are the n^{th} derivative of each coordinate with respect to the parameter. This means that the derivative information now includes direction in addition to magnitude. This can introduce subtleties even in simple situations. For example, a curve could have a continuous unit tangent vector or slope and yet lack a continuous first derivative due to a jump in the magnitude of the first derivative. This simple idea shows the distinction between *parametric continuity* and *geometric continuity*;[1,4,7,8,9,10,11,22,23,29,30,32,34,35] this is covered in more detail in Section 5.

Intuitively, the parametrization can be thought of as a description in terms of time. The vector-valued representation provides the position at a given instant in time. As time passes, the path is traced out. Using this metaphor, it is easy to imagine different parametrizations that trace out the same path. The identical path can be traced out with different velocities. Whether a particle moves with uniform speed or alternatively accelerates and decelerates, the very same path can be traversed. This illustrates that many parametrizations can specify the same curve. Thus, one should distinguish between a *parametrization* and a *curve*. Consequently, the very same curve can be *reparametrized* such that the parametrization changes but the shape of the final curve does not. These ideas are at the heart of the study of geometric continuity. Another complexity introduced by the parametric representation is that there are now two separate spaces with which to deal. The curve or surface itself exists in a *geometric space*. However, there is also a parameter space which is one-dimensional for a curve and two-dimensional for a surface. Another way to interpret this parametric representation is that we are defining a mapping which distorts the parameter space into the corresponding shape in geometric space. Imagine taking an infinitely-stretchable rectangular sheet of rubber (which is the parameter space) and bending and twisting it to form a surface in three-space.

More precisely, a *parametric* function defines a mapping from a *domain parameter space*, into geometric or *Euclidean* space. The definition of a curve involves functions of a single parameter, whereas for a surface it uses functions of a pair of parameters. Specifically, in the case of curves, the parametric function defines a mapping from u into Euclidean two-space as $\mathbf{Q}(u) = [x(u), y(u)]$ or into Euclidean three-space as $\mathbf{Q}(u) = [x(u), y(u), z(u)]$. This function can be used to define a curve by letting u range over some interval $[u_0, u_f]$ of the u axis. For a surface, the parametric function is a mapping from u,v into three-space as $\mathbf{Q}(u,v) = [x(u,v), y(u,v), z(u,v)]$. A surface is then defined using this function by letting u and v range over some rectangle $[u_0, u_f] \times [v_0, v_f]$ in the u,v plane. Note that the use of **boldface** is to demonstrate that the function is vector-valued.

In the case of curves, if the domain parameter is thought of as time, the parametric function is used to locate the position of the particle in space at a given instant. As time passes, the particle sweeps out a path, thereby tracing the curve. A parametric function therefore defines more than just a path; there is also information about the direction and speed of the particle as it moves along the path.

2. Explanation of Bézier Curves

Bézier curves and surfaces,[2, 11, 12, 13, 14, 15, 16, 18, 26, 28] named for Pierre Bézier, form the nucleus of *Systeme Unisurf* at Renault. The Bézier curve is specified by a set of points, called *control vertices*, which are connected in a sequence to form an open or closed *control polygon* (Figure 1). The resulting curve begins at the first control vertex and ends at the last control vertex but does not necessarily interpolate any of the interior vertices. The curve is tangent to the first and last polygon edge. The shape of the resulting curve mimics that of the control polygon, but in a smoother fashion.

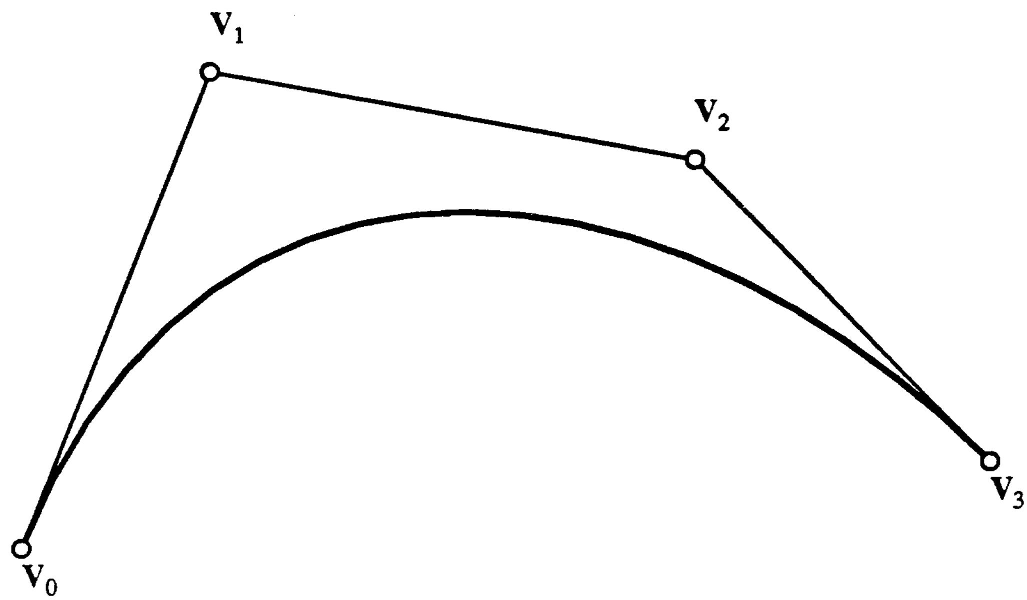

Figure 1: *Bézier curve and its control polygon.*

The Bézier curve can be expressed mathematically as a weighted average of these control vertices. A particular point on the curve corresponds to a specific set of weights applied to these control vertices. As the values of these weights are varied, the curve is then traced out. Each weighting factor is a function of a parameter. Thus, the connection between the value of the parameter and a point on a curve is established by evaluating each of these weighting functions at the particular value of the parameter and then computing the corresponding weighted average. Curves of different shapes can be generated by using different positions of the control vertices. This weighted average can also be regarded as a linear combination where the control vertices are the combination coefficients. In this interpretation, the weighting factors play the role of basis vectors. For this reason, these weighting factors are usually referred to as *basis functions*. Like weights, these basis functions are nonnegative and sum to one. The idea, then, is that as the value of the parameter is varied, the basis functions attain various values that alter the weighting of the control vertices thereby producing a set of points to form the final curve.

Consider a control polygon $[\mathbf{V}_0, \cdots, \mathbf{V}_i, \cdots, \mathbf{V}_d]$. Consequently, a Bézier curve of degree d, denoted by $\mathbf{Q}_d(u)$ where $u \in [0,1]$, is defined by:

$$\mathbf{Q}_d(u) = \sum_{i=0}^{d} \mathbf{V}_i B_{i,d}(u), \quad u \in [0,1] \tag{1}$$

where $B_{i,d}(u)$ is the i^{th} *Bernstein* polynomial of degree d

$$B_{i,d}(u) = \binom{d}{i} u^i (1-u)^{d-i}, \quad i=0, \cdots, d. \tag{2}$$

In the case of $d=3$, the cubic basis functions are:

$$B_{0,3}(u) = (1-u)^3 = -u^3+3u^2-3u+1 \quad (3.1)$$

$$B_{1,3}(u) = 3u(1-u)^2 = 3u^3-6u^2+3u \quad (3.2)$$

$$B_{2,3}(u) = 3u^2(1-u) = -3u^3+3u^2 \quad (3.3)$$

$$B_{3,3}(u) = u^3 \quad (3.4)$$

The Bernstein polynomials, $B_{i,d}(u)$, $u \in [0,1]$, are plotted for $d = 1, \ldots, 6$ in Figure 2.

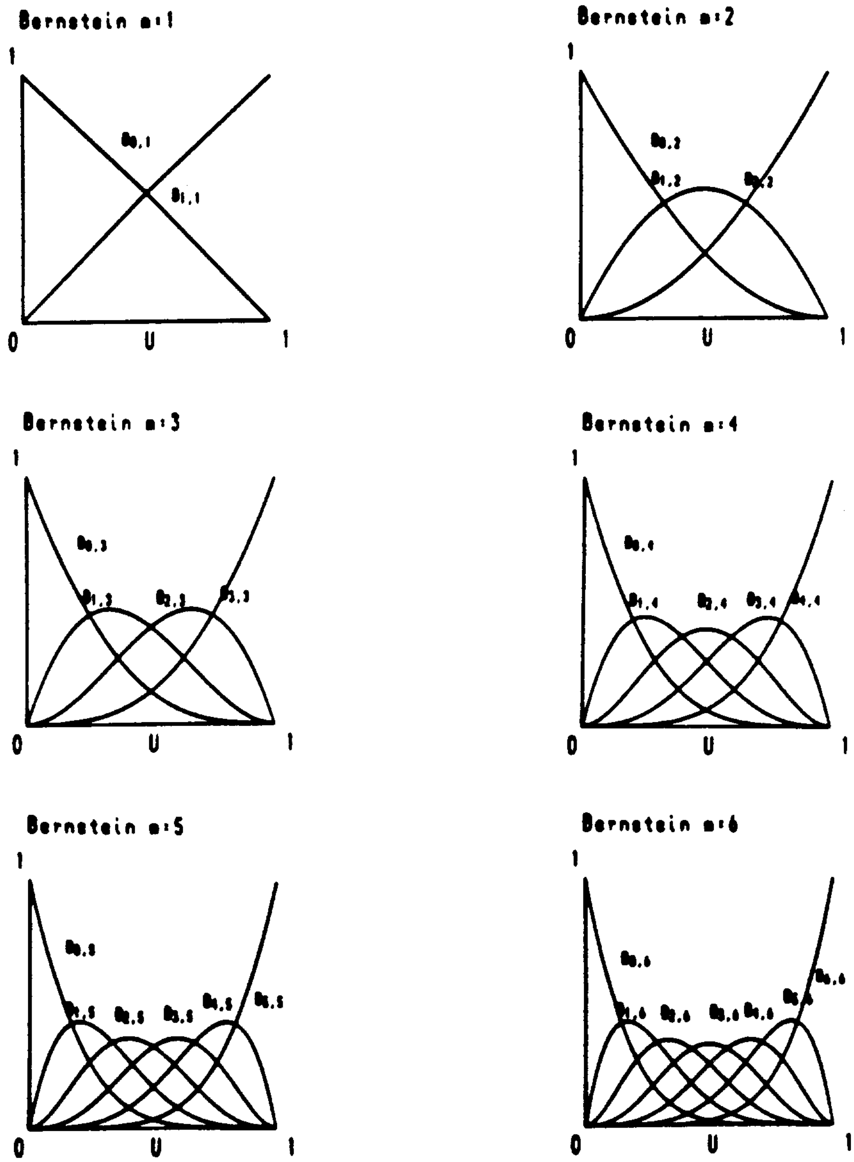

Figure 2: *The Bernstein polynomials for degree 1 through 6.*

These Bernstein polynomials are the polynomials that are seen in the binomial distribution as well as in the proof of the Weierstrass Theorem. For this reason, Bézier curves are sometimes referred to as *Bernstein-Bézier curves*. Because of the interpretation of these basis functions as the binomial distribution, there are some interesting probabilistic interpretations of readily apparent.

The equations for a Bézier curve can be recast in matrix notation. From equation (1), the curve can be expressed as

$$\mathbf{Q}_d(u) = [B_{0,d}(u)\, B_{1,d}(u) \cdots B_{d,d}(u)] \begin{bmatrix} \mathbf{V}_0 \\ \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_d \end{bmatrix} \tag{4}$$

From equation (2), the polynomials can be written as:

$$[B_{0,d}(u)\, B_{1,d}(u) \cdots B_{d,d}(u)] = [u^d\; u^{d-1} \cdots 1][B] \tag{5}$$

where $[B]$ is the matrix of coefficients of the Bernstein polynomials. Substituting equation (5) into equation (4), the curve can be rewritten in the following matrix form:

$$\mathbf{Q}_d(u) = [u^d\; u^{d-1} \cdots 1][B] \begin{bmatrix} \mathbf{V}_0 \\ \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_d \end{bmatrix} \tag{6}$$

In the case of d=3, this is the cubic Bézier curve, where the matrix of coefficients is given by

$$B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

3. Properties of Bézier Curves

The relationships between the parametric derivatives at u=0 and u=1 and the control vertices are simply expressed; specifically,

(i) **Position:** $\mathbf{Q}_d(u)$ interpolates $\mathbf{V}_0$ at u=0, and $\mathbf{V}_d$ at u=1:

$$\mathbf{Q}_d(0) = \mathbf{V}_0 \tag{8.1}$$

$$\mathbf{Q}_d(1) = \mathbf{V}_d. \tag{8.2}$$

(ii) **First Derivatives:** The initial first derivative vector is in the direction of the vector from $\mathbf{V}_0$ to $\mathbf{V}_1$, and the final first derivative vector is in the direction of the vector from $\mathbf{V}_{d-1}$ to $\mathbf{V}_d$. More precisely, the initial and final first derivative vectors are:

$$\mathbf{Q}_d^{(1)}(0) = d(\mathbf{V}_1 - \mathbf{V}_0) \tag{9.1}$$

$$\mathbf{Q}_d^{(1)}(1) = d(\mathbf{V}_d - \mathbf{V}_{d-1}). \tag{9.2}$$

(iii) **Second Derivatives:** The initial second derivative vector depends only on $\mathbf{V}_0$, $\mathbf{V}_1$, and $\mathbf{V}_2$, and the final second derivative vector depends only on $\mathbf{V}_{d-2}$, $\mathbf{V}_{d-1}$, and $\mathbf{V}_d$; specifically,

$$\mathbf{Q}_d^{(2)}(0) = d(d-1)(\mathbf{V}_0 - 2\mathbf{V}_1 + \mathbf{V}_2) \tag{10.1}$$

$$\mathbf{Q}_d^{(2)}(1) = d(d-1)(\mathbf{V}_{d-2} - 2\mathbf{V}_{d-1} + \mathbf{V}_d). \tag{10.2}$$

The degree of a Bézier curve is equal to the number of edges in the control polygon, that is, one less than the number of control vertices. In this manner, the curve is a single polynomial of this degree. Note that this is not a piecewise representation. Since this is simply a polynomial, it is C^∞ continuous.

Furthermore, this approach has global, not local, control. A curve representation with local control has the effect of a control vertex restricted to a small predetermined region of the curve or surface.

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 control polygon that generates the exact same curve with a (degenerate) polynomial of higher order (Figure 3). If we have the control polygon $[\mathbf{V}_0, \cdots, \mathbf{V}_i, \cdots, \mathbf{V}_d]$, the following formula gives the control polygon $[\mathbf{W}_0, \cdots, \mathbf{W}_i, \cdots, \mathbf{W}_{d+1}]$:

$$\mathbf{W}_i = \left(\frac{i}{d+1}\right)\mathbf{V}_{i-1} + \left(1 - \frac{i}{d+1}\right)\mathbf{V}_i \qquad i = 0, \cdots, d+1 \tag{11}$$

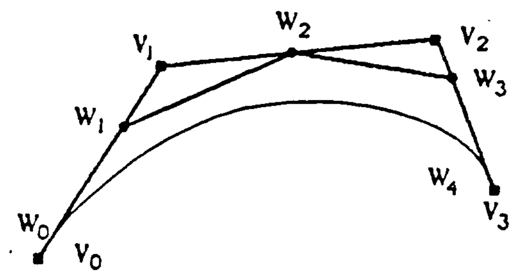

Figure 3: *Raising the degree of a Bézier curve from cubic to quartic.*

Bézier curves have the *variation-diminishing* property.[33,41] Intuitively, a curve satisfying this property wiggles less than the underlying data. To be more precise, consider an ordered sequence of straight line segments connecting the data points. A curve is said to be variation diminishing if there does not exist any line that can be drawn that would intersect the curve more often than it would intersect the line segments connecting the data points. Although this property fails for interpolation schemes, it is achieved for approximation schemes such as the Bézier, B-spline, and Beta-spline representations. It is interesting to note that there is no analogous variation-diminishing property for surfaces.

Another important property is the *convex hull* property which provides a region in which the curve or surface must lie. Intuitively, the convex hull of a set of points is the region that would be enclosed by an infinitely stretchable rubber material wrapped around the points and pulled taut. In two dimensions, this could be thought of as an elastic rubber band enclosing an area. In three dimensions, the convex hull of a control graph is obtained by imagining a rubber membrane stretched taut around the graph; the volume within the membrane is the convex hull. The Bézier surface will lie inside the convex hull of its control graph.

The convex hull property can be exploited in many ways. It allows bounding regions for the curve or surface to be easily calculated, a fact that greatly increases the efficiency of many algorithms. This property can also be used in conjunction with subdivision to develop a fast occlusion algorithm. If two control polygons or graphs do not occlude one another, neither can the curves or surfaces they define. If they do interfere, subdivision can be used as a means of resolving the interference.

4. Composite Bézier Curves Using Parametric Continuity

Although the form for the single Bézier curve described in Section 1 is simple and C^∞ continuous, the lack of local control and the connection of the degree to the number of control vertices are problematic. These impediments can be circumvented through the use of a piecewise version of the Bézier curve, although this is at the expense of a reduction in the level of continuity achieved. The composite (or piecewise) Bézier curve strings together a sequence of Bézier curves, each with its own control polygon, thereby reducing the degree and establishing local control. However, to achieve a given level of continuity requires the application of constraints to the positions of the control vertices. This represents a departure from the idea that the control vertices could be placed in any position desired.

The simplest continuity constraint is positional continuity which would require that the last vertex of one control polygon be in the same position as the first vertex of the succeeding control polygon. In this case, moving any interior control vertex of a control polygon would affect just that one Bézier curve while moving this common control vertex would result in the modification of two Bézier curve segments. If *first order parametric continuity* (C^1) is required in addition, then the last edge of the previous control polygon must be collinear with the first edge of the next one and these two edges must also be of equal length (Figure 4). *Parametric second derivative vector continuity* (C^2) at a joint involves constraints on the common control vertex and on the two control vertices on either side of the joint. Consequently, for cubics, maintenance of C^2 continuity when moving one control vertex requires repositioning some control vertices associated with several neighbouring segments. Note that it is sufficient to maintain C^2 continuity by modifying control vertices on only one adjacent segment if higher degree curve segments were used. Since C^2 continuity at a joint affects the common control vertex and the two control vertices on either side of the joint, this continuity could be ensured by defining each curve segment with six control vertices, that is, by using fifth degree curve segments.

More generally, since C^n continuity at a joint affects $n+1$ control vertices per segment including the common control vertex on either side of the joint, this continuity could be ensured by defining each curve segment with $2(n+1)$ control vertices, that is, by using degree $2n+1$ curve segments. However, it is possible to maintain C^n continuity with a composite Bézier curve of degree $n+1$ by adjusting the control vertices of only $n+2$ segments.

To see why this is the case, first recall that degree $n+1$ B-spline curves have this local control property. Then note that it must be possible to represent a composite Bézier curve of degree $n+1$ as a single knot degree $n+1$ B-spline curve since these curves are both C^n piecewise polynomial curves. The effect of moving a control vertex along with the neighbouring vertices that would need to be adjusted in the Bézier representation so as to maintain C^n continuity can be achieved by moving a B-spline control vertex. Finally, the resulting B-spline curve can be converted back to a Bézier representation using knot insertion.[11]

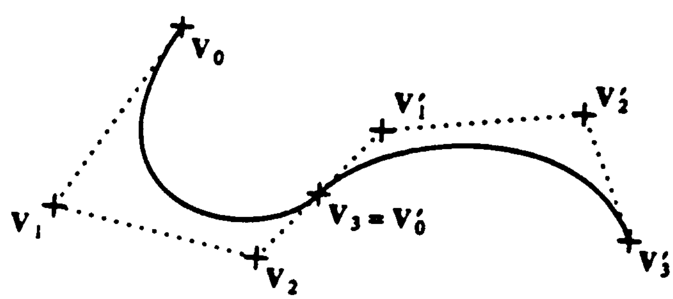

Figure 4: *A composite cubic Bézier curve.*

5. Geometric Continuity

As was mentioned in Section 1, there are many subtleties involved in establishing continuity constraints for parametric splines. Traditionally, the continuity constraints that have been used have been the same as for nonparametric splines, except simply transposed into a parametric form. In other words, it has been the derivatives that have been constrained to be continuous. For the parametric representation, the derivatives in question are parametric derivative vectors, where each component is the derivative of a coordinate with respect to the parameter. Because this form of derivative is fundamentally different than the scalar-valued functional derivatives to which we are accustomed, our intuition regarding the associated geometry often fails. As an example, it is possible to parametrize a piecewise representation of a circle to have a discontinuity in the second derivative vector, although the curvature (and curvature vector) would be continuous. This leads to the idea that it would be of interest to constrain the unit tangent and curvature vectors to be continuous rather than the first and second derivative vectors. We have named this form of continuity as *geometric continuity*. In the case of constraining the unit tangent and curvature vectors to be continuous, we refer to this as G^2 continuity. We call the traditional method of maintaining continuous parametric derivative vectors as *parametric continuity* to distinguish this from the geometric continuity approach.[4,11]

In addition to being a more appropriate measure of continuity for parametric curves, geometric continuity has the advantage that it is a more relaxed form of continuity. The continuity constraints that it generates are generalizations of the continuity constraints for parametric continuity. These more general geometric continuity constraints liberate some degrees of freedom, which are called *shape parameters*, that can be captured to provide further control of shape.

Having defined geometric continuity of order two (G^2), it is of interest to investigate generalizing to higher order. How can we define geometric continuity of order n, for an arbitrary n? The key to our answer to this is the observation, which was made in Section 1, that many different parametrizations can describe the same curve.[11] Two regular† C^n parametrizations are said to be *equivalent* if there exists a regular C^n function that is regular and onto and that when composed with one of the parametrizations yields the other one. Intuitively, equivalent parametrizations trace out the same set of points in the same order. Thus, it is possible to alter the parametrization of a curve without changing its shape; this is referred to as *reparametrization*.

From this observation, we are now ready to define geometric continuity for arbitrary order n. Two regular C^n parametrizations, denoted $\mathbf{q}(u)$ and $\mathbf{r}(t)$, meet with n^{th} order geometric continuity, denoted G^n, if it is possible to reparametrize one of the parametrizations such that it would meet the other parametrization with C^n continuity. Although this does provide a definition of geometric continuity of arbitrary order, it is not very practical because it still leaves open the question of how to determine whether or not such a reparametrization exists. Based on this definition of geometric continuity of arbitrary order n and using the idea of composition for the equivalent parametrizations yields equations for the first n derivatives in terms of two different parameters. Performing the prescribed differentiation requires invoking the chain rule at each level of differentiation. The resulting equations involve the first n derivatives of one parameter with respect to the other. Denoting the j^{th} such derivative at the joint by β_j yields the so-called *Beta-constraints*. These β's are the same β's that appear as shape parameters in the Beta-spline. The Beta-constraints provide a set of necessary and sufficient conditions that two parametrizations meet with G^n continuity; specifically, two parametrizations meet with G^n continuity if and only if there exist numbers $\beta_1, \cdots, \beta_n$ that satisfy the Beta-constraints‡.

As an example of the form of the Beta-constraints, the constraints for G^4 continuity are

$$\mathbf{r}^{(1)}(0) = \beta_1 \mathbf{q}^{(1)}(1) \tag{12.1}$$

$$\mathbf{r}^{(2)}(0) = \beta_1^2 \mathbf{q}^{(2)}(1) + \beta_2 \mathbf{q}^{(1)}(1) \tag{12.2}$$

$$\mathbf{r}^{(3)}(0) = \beta_1^3 \mathbf{q}^{(3)}(1) + 3\beta_1\beta_2 \mathbf{q}^{(2)}(1) + \beta_3 \mathbf{q}^{(1)}(1) \tag{12.3}$$

$$\mathbf{r}^{(4)}(0) = \beta_1^4 \mathbf{q}^{(4)}(1) + 6\beta_1^2\beta_2 \mathbf{q}^{(3)}(1) + (4\beta_1\beta_3 + 3\beta_2^2)\mathbf{q}^{(2)}(1) + \beta_4 \mathbf{q}^{(1)}(1). \tag{12.4}$$

where β_2, β_3, and β_4 are arbitrary, but β_1 is constrained to be positive. The Beta-constraints can be used to

† A parametrization is *regular* if its first derivative vector never vanishes.

‡ To maintain the orientation preserving property, β_1 is constrained to be positive.

form a composite Bézier curve that is smooth, as is discussed in the following section.

6. Geometric Continuity for Composite Bézier Curves

Consider the goal of stitching Bézier curves together with G^1 and G^2 continuity.[8,9] The problem we now wish to address is to maintain some geometric continuity at the joints; specifically,

Given: The shape parameters β_1 and β_2, and the control polygon $[\mathbf{V}_0, \cdots, \mathbf{V}_i, \cdots, \mathbf{V}_d]$ defining the parametrization

$$\mathbf{Q}_d(u) = \sum_{i=0}^{d} \mathbf{V}_i B_{i,d}(u), \quad u \in [0,1], \tag{13}$$

find: constraints on the control polygon $\mathbf{W}_0, \cdots, \mathbf{W}_d$ defining the parametrization

$$\mathbf{R}_d(t) = \sum_{j=0}^{d} \mathbf{W}_j B_{j,d}(t), \quad t \in [0,1] \tag{14}$$

such that: $\mathbf{Q}_d$ and $\mathbf{R}_d$ meet with G^1 (or G^2) continuity at $\mathbf{Q}_d(1)$ with respect to β_1 (and β_2) (see Figure 5).

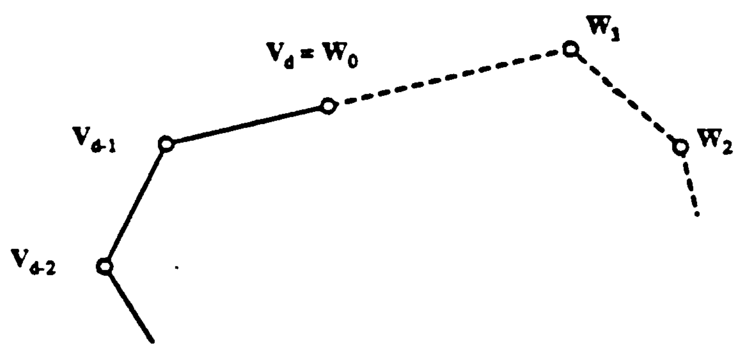

Figure 5: *Situation for stitching two Bézier curves* $\mathbf{Q}_d$ *and* $\mathbf{R}_d$ *together with* G^n *continuity.*

Since a Bézier curve interpolates its first and last control vertices, we can guarantee C^0 (and hence G^0) continuity by setting $\mathbf{W}_0 = \mathbf{V}_d$, as shown in Figure 5. To achieve G^1 continuity for a given $\beta_1>0$, we can find $\mathbf{W}_1$ by recalling Equation (12.1) and using Equation (9) to yield

$$d(\mathbf{W}_1 - \mathbf{W}_0) = d\beta_1(\mathbf{V}_d - \mathbf{V}_{d-1}), \quad \beta_1>0. \tag{15}$$

Simplification and rearrangement yields

$$\mathbf{W}_1 = \mathbf{W}_0 + \beta_1(\mathbf{V}_d - \mathbf{V}_{d-1}), \quad \beta_1>0, \tag{16}$$

and since $\mathbf{W}_0 = \mathbf{V}_d$,

$$\mathbf{W}_1 = \mathbf{V}_d + \beta_1(\mathbf{V}_d - \mathbf{V}_{d-1}), \quad \beta_1>0. \tag{17}$$

Geometrically, Equation (17) states that $\mathbf{W}_1$ must lie on the ray starting at $\mathbf{V}_d$ (= $\mathbf{W}_0$), extending in the direction of the vector from $\mathbf{V}_{d-1}$ to $\mathbf{V}_d$. The length of the segment $\mathbf{W}_0\mathbf{W}_1$ relative to the length of $\mathbf{V}_{d-1}\mathbf{V}_d$ is given by the parameter β_1. Thus, given $\mathbf{V}_{d-1}$, $\mathbf{V}_d$ and $\beta_1>0$, the control vertices $\mathbf{W}_0$ and $\mathbf{W}_1$ can be determined geometrically as shown in Figure 6, or algorithmically using the following construction:

$$(1)\ \mathbf{W}_0 \Leftarrow \mathbf{V}_d \tag{18.1}$$

$$(2)\ \mathbf{W}_1 \Leftarrow \mathbf{W}_0 + \beta_1(\mathbf{V}_d - \mathbf{V}_{d-1}) \tag{18.2}$$

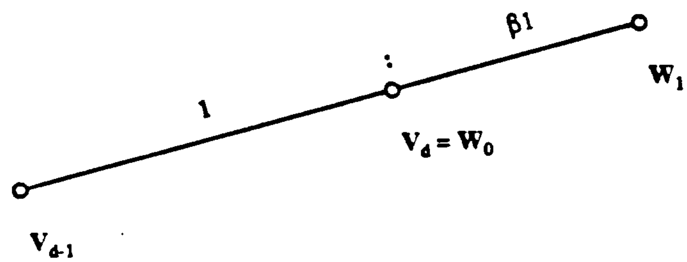

Figure 6: *The construction of* $\mathbf{W}_0$ *and* $\mathbf{W}_1$ *to achieve* G^1 *continuity for a given* β1.

Once $\mathbf{W}_0$ and $\mathbf{W}_1$ have been constrained subject to G^1 continuity, the control vertex $\mathbf{W}_2$ can be constrained to guarantee G^2 continuity for a given β2 by recalling Equation (12.2) and using Equation (10) to yield

$$d(d-1)(\mathbf{W}_0 - 2\mathbf{W}_1 + \mathbf{W}_2) = \beta_1^2\, d(d-1)(\mathbf{V}_{d-2} - 2\mathbf{V}_{d-1} + \mathbf{V}_d) + \beta_2\, d(\mathbf{V}_d - \mathbf{V}_{d-1}). \tag{19}$$

Solving for $\mathbf{W}_2$ yields

$$\mathbf{W}_2 = 2\mathbf{W}_1 - \mathbf{W}_0 + \beta_1^2(\mathbf{V}_{d-2} - 2\mathbf{V}_{d-1} + \mathbf{V}_d) + \frac{\beta_2(\mathbf{V}_d - \mathbf{V}_{d-1})}{d-1} \tag{20}$$

Substituting $\mathbf{V}_d$ for $\mathbf{W}_0$ and Equation (17) for $\mathbf{W}_1$, and rearranging yields

$$\mathbf{W}_2 = \beta_1^2\mathbf{V}_{d-2} - (2\beta_1^2 + 2\beta_1 + \frac{\beta_2}{d-1})\mathbf{V}_{d-1} + (\beta_1^2 + 2\beta_1 + \frac{\beta_2}{d-1} + 1)\mathbf{V}_d \tag{21}$$

Rather than the algebraic approach given above for the determination of $\mathbf{W}_2$, a more geometric approach was developed by Farin[24] and later improved upon by Boehm.[17] For our purposes, it is most convenient to think of the approach of Farin and Boehm as a convenient factorization of Equation (21), each term of which has a well-defined geometric interpretation.

The Farin-Boehm construction takes as input the control polygon $[\mathbf{V}_{d-2}, \mathbf{V}_{d-1}, \mathbf{V}_d]$ and the shape parameters $\beta_1 > 0$ and β2, and produces as output the control vertices $\mathbf{W}_0$, $\mathbf{W}_1$, and $\mathbf{W}_2$ such that the curves meet with G^2 continuity with respect to β1 and β2. The construction may be stated as:

$$(1)\ \gamma \Leftarrow \frac{(d-1)(1+\beta_1)}{\beta_2 + \beta_1(d-1)(1+\beta_1)} \tag{22.1}$$

$$(2)\ \mathbf{W}_0 \Leftarrow \mathbf{V}_d \tag{22.2}$$

$$(3)\ \mathbf{W}_1 \Leftarrow \mathbf{W}_0 + \beta_1(\mathbf{V}_d - \mathbf{V}_{d-1}) \tag{22.3}$$

$$(4)\ \mathbf{T} \Leftarrow \mathbf{V}_{d-1} + \beta_1^2\gamma(\mathbf{V}_{d-1} - \mathbf{V}_{d-2}) \tag{22.4}$$

$$(5)\ \mathbf{W}_2 \Leftarrow \mathbf{W}_1 + \frac{1}{\gamma}(\mathbf{W}_1 - \mathbf{T}) \tag{22.5}$$

The geometric interpretation of this construction is shown in Figure 7.

In other words, only $\mathbf{W}_3$ can be freely chosen if we insist on G^2 geometric continuity *once* β1 and β2 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 β1 and β2 to be able to ensure G^2 on both sides of the span.

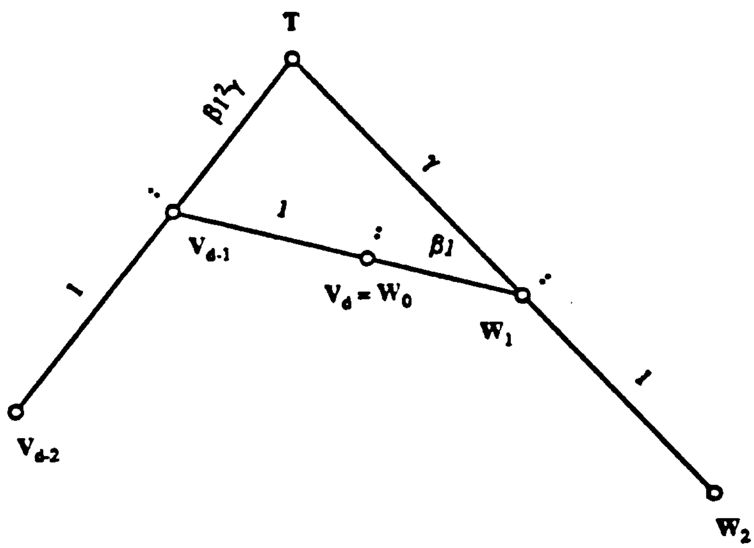

Figure 7: *The Farin-Boehm construction.*

7. Subdivision of Bézier Curves

One of the most important attributes of the Bézier curve is the ease with which it can be *subdivided.* By introducing new control vertices, subdivision splits the curve into two pieces, each of which has its own defining control polygon. In specific cases, the positions of the new control vertices can be determined explicitly from the positions of the original control vertices. More generally, a set of intermediate control vertices is introduced.

To be more precise, a Bézier curve $Q_d(u)$ of degree d can be rewritten as:[3]

$$Q_d(u) = \sum_{i=0}^{d-1} [(1-u)V_i + uV_{i+1}]B_{i,d-1}(u) \tag{23}$$

Equation (23) can be rewritten as

$$Q_d(u) = \sum_{i=0}^{d-1} V_i^{[1]}(u)B_{i,d-1}(u) \tag{24}$$

where

$$V_i^{[1]}(u) = (1-u)V_i + uV_{i+1}. \tag{25}$$

Repeating this process recursively k times yields

$$Q_d(u) = \sum_{i=0}^{d-k} V_i^{[k]}(u)B_{i,d-k}(u) \tag{26}$$

where

$$V_i^{[k]}(u) = \begin{cases} (1-u)V_i^{[k-1]}(u) + uV_{i+1}^{[k-1]}(u), & k=1,\ldots,d \\ V_i & k=0 \end{cases} \tag{27}$$

In the case of $k=d$, equation (26) becomes

$$Q_d(u) = V_0^{[d]}(u)B_{0,0}(u) \tag{28}$$

which is

$$Q_d(u) = V_0^{[d]}(u). \tag{29}$$

Thus, for a given parametric value u^*, the point on the curve at u^* is $V_0^{[d]}(u^*)$, as defined in equation (27). This yields another way to compute a point on the curve; simply recursively compute this vertex using a ratio equal to the parametric value of the desired point. This idea can be used to geometrically construct a Bézier curve. To compute the point $Q_d(u^*)$, each edge of the control polygon is divided in the ratio $u^*:1-u^*$. For example, for midpoint subdivision each edge is simply divided at its parametric midpoint. Then, these new vertices are connected in succession. This forms a sequence of edges where the number of edges here is one less than in the original polygon (that is, $d-1$ edges). Each of these new edges is then subdivided in the same ratio and these new points are connected again. At each stage of this process, there is one less edge than there was at the preceding stage. This process is continued, and after $d-1$ iterations, a single edge results. Then, in the d^{th} iteration, this edge is subdivided again in the same ratio. This point is then the common vertex of the two new Bézier curves. The remaining vertices of each of the two new Bézier curves are a subset of those found in this development. Figure 8 illustrates the computation of a point on a cubic Bézier curve. This same process can be performed for various values of the parameter u, and then these points can be connected to generate a piecewise linear approximation to the curve.

In addition, $V_0^{[d]}(u^*)$ is the common vertex between two subdivided curves, each with its own control polygon. This is illustrated in Figure 8 for degree d=3.

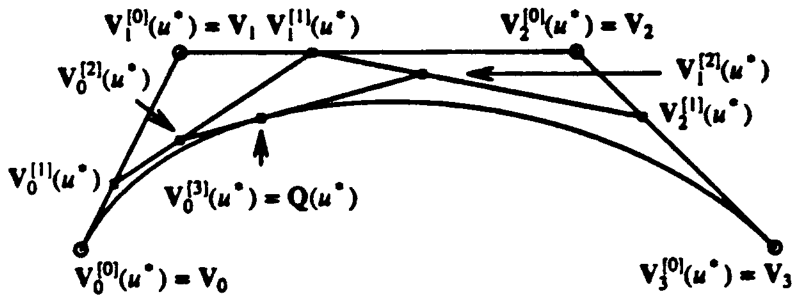

Figure 8: $V_0^{[3]}(u^*)$ *is the common vertex between the two subdivided curves for degree* d=3.

Because Bézier curves interpolate their endpoints, this new point just found will lie on the Bézier curve. For this reason, this geometric construction is often given as a method to compute a point on the Bézier curve. This approach is sometimes referred to as the *deCasteljau Algorithm*.[19, 20]

Other points on the curve can be found in one of two ways. One possibility is simply to rerun this subdivision at a different value of the parameter u. Thus, subdividing for a sequence of different values of the parameter u will generate a sequence of points on the curve. The other possibility is to then treat each one of the new Bézier curves as a starting point for subdivision and simply recursively subdivide on each side. This latter approach is frequently referred to as *recursive subdivision*. Associated with recursive subdivision is some criterion for termination. One such criterion is *flatness*. In this approach, subdivision of a particular Bézier curve segment stops when that segment is deemed "flat." At that point, the curve segment can be approximated either by its control polygon or by the straight line segment connecting the first and last control vertex for that curve segment.

Such a termination criterion requires some kind of *flatness test*. The flatness test should be some computation performed on the vertices themselves so as to avoid a calculation of a point on the curve. Many different tests are possible. However, it is a challenge to develop a test that is both computationally efficient and provides a correct conclusion over a wide range of cases. Given a particular test, it is interesting to construct counter-examples for which the test would yield a misleading answer. This cannot be solved simply by developing more elaborate tests, since it would defeat the purpose of subdivision if the amount of computation required to perform the flatness test equaled or exceeded that required to perform another level of subdivision.

8. Tensor Product Bézier Surfaces

A tensor-product Bézier surface, denoted by $\mathbf{Q}_{d,e}(u,v)$, is a generalization of a Bézier curve, and can be defined as

$$\mathbf{Q}_{d,e}(u,v) = \sum_{i=0}^{d}\sum_{j=0}^{e} \mathbf{V}_{ij}\, B_{i,d}(u)\, B_{j,e}(v) \tag{30}$$

The $B_{i,d}(u)$ and $B_{j,e}(v)$ are the Bernstein basis functions, defined in the u and v parametric directions, respectively, and have degree d in the u direction and e in the v direction. The $\mathbf{V}_{ij}$ (x_{ij}, y_{ij}, z_{ij}) are the three-dimensional control vertices organized in a two-dimensional $m+1$ by $n+1$ control graph with a rectangular topology as shown in Figure 9. Note that the connectivity of the graph is implicitly determined by considering the graph as a two-dimensional array of vertices. Two vertices share an edge if and only if they are adjacent in the array.

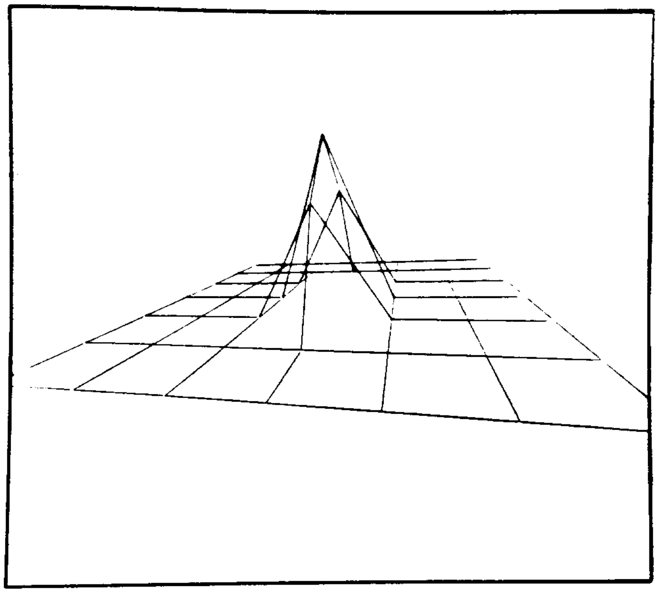

Figure 9: *A control graph for surface specification.*

Given the control graph $[\mathbf{V}_{00}, \mathbf{V}_{01}, \cdots \mathbf{V}_{de}]$, a point on the surface is a weighted average of these control vertices:

$$\mathbf{Q}_{d,e}(u,v) = \sum_{i=0}^{d}\sum_{j=0}^{e} \mathbf{V}_{ij} B_{i,d}(u) B_{j,e}(v) \tag{31}$$

or, in matrix form

$$\mathbf{Q}_{d,e}(u,v) = [B_{0,d}(u)\, B_{1,d}(u) \cdots B_{d,d}(u)][\mathbf{V}] \begin{bmatrix} B_{0,e}(v) \\ B_{1,e}(v) \\ \vdots \\ B_{e,e}(v) \end{bmatrix} \tag{32}$$

where

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{00} & . & . & . & \mathbf{V}_{0e} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{V}_{d0} & . & . & . & \mathbf{V}_{de} \end{bmatrix} \tag{33}$$

Figure 10 shows a Bézier surface where $d=e=6$.

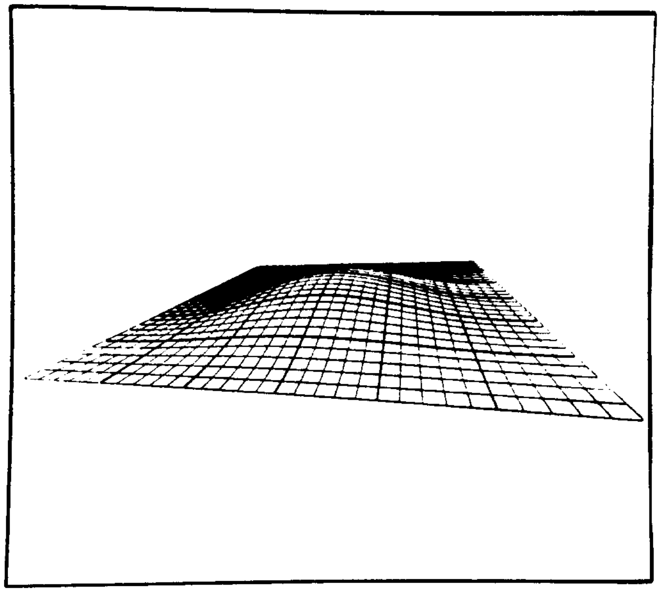

Figure 10: *The bivariate sixth degree Bézier surface defined by the control graph in Figure 9.*

9. Rational Bézier Curves and Surfaces

The *rational* form has the advantage that it can represent conic[21,31,40] curves and quadric surfaces[31] as well as free-form curves and surfaces. Examples of conic curves include circles, ellipses, parabolas and hyperbolas while examples of quadric surfaces include spheres, ellipsoids, cylinders, cones, paraboloids, hyperboloids, and hyperbolic paraboloids.

A further advantage is that a rational formulation is invariant under projective transformation (in addition to affine transformation, as is the case for the integral counterpart). Since perspective is a projection, this property can be exploited to generate a perspective projection of a rational curve or surface without resorting to applying this projection for every point to be displayed, as is the case for the integral version. Additionally, there are weights which can be used to control shape in a manner similar to shape parameters.

A rational Bézier curve[25] can be viewed as an integral Bézier curve in a vector space whose dimension is one higher than that of the space of the rational Bézier curve. This "next higher" dimensional space is referred to as the *homogeneous coordinate space*. For details on homogeneous coordinates, the reader is referred to.[27,36,37,38,39]

Denoting the dimension of the space of the rational spline by N, then in this scheme, the "extra" coordinate is used as a denominator for the first N coordinates. When each coordinate is a polynomial, the $N+1$ coordinates taken together can be interpreted as N rational polynomial coordinates each sharing the same denominator. A rational spline in $\mathbf{R}^N$ is then the projection of an integral spline in the corresponding homogeneous coordinate space, $\mathbf{R}^{N+1}$.

For illustrative purposes, consider a rational curve in the plane, that is, $N=2$. This rational curve in two dimensions is defined in a three-dimensional space represented by homogeneous coordinates. The third coordinate is the *weight* and is assumed to be positive. There is a distinct weight associated with each vertex. The three-dimensional curve can be projected to two dimensions yielding a *rational* curve.[5,32]

An illustration of how a rational curve is the projection of an integral curve from a vector space of one higher dimension is provided in Figure 11. The solid curve in $\mathbf{R}^2$ is the projection of the dashed curve in $\mathbf{R}^3$.

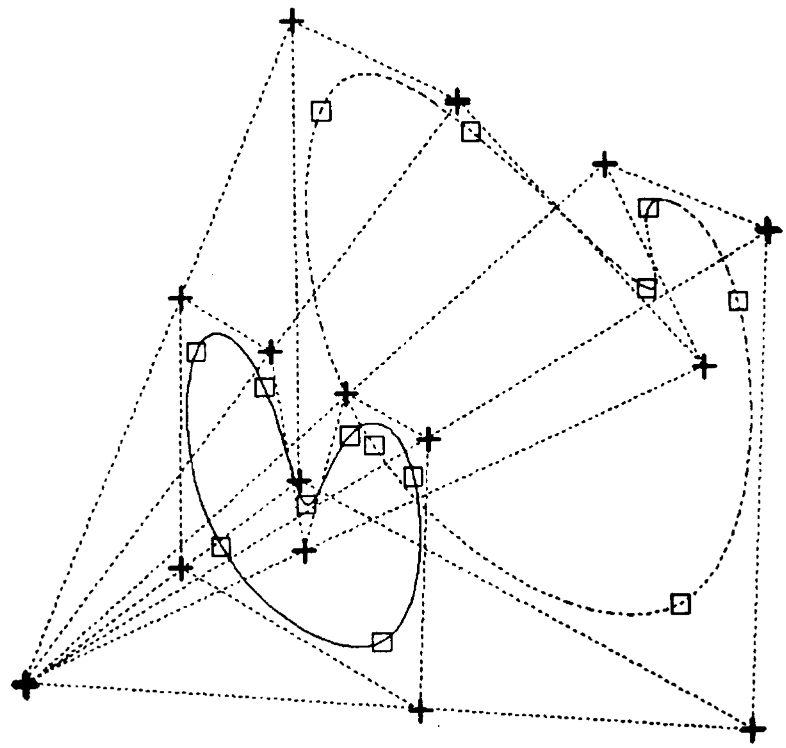

Figure 11: *Rational curve is the projection of an integral curve.*

The term "rational" refers to the ratio which characterizes this approach. More specifically, let w_i, for $i = 0, \cdots, d$, be the $d+1$ weights corresponding to the control vertices $\mathbf{V}_i$. Then, a rational Bézier curve of degree d is given by:

$$\mathbf{Q}_d(u) = \frac{\sum_{i=0}^{d} w_i \mathbf{V}_i B_{i,d}(u)}{\sum_{r=0}^{d} w_r B_{r,d}(u)} \tag{34}$$

The rational Bézier curve defined in equation (34) can be rewritten in a more familiar form, that is, as a linear combination of basis functions which are now *rational* basis functions. Rearranging equation (34) yields

$$\mathbf{Q}_d(u) = \sum_{i=0}^{d} \mathbf{V}_i \left[\frac{w_i B_{i,d}(u)}{\sum_{r=0}^{d} w_r B_{r,d}(u)} \right] \tag{35}$$

Denoting the term in brackets by $R_{i,d}(u)$,

$$R_{i,d}(u) = \frac{w_i B_{i,d}(u)}{\sum_{r=0}^{d} w_r B_{r,d}(u)} \tag{36}$$

and replacing this in equation (35) results in

$$\mathbf{Q}_d(u) = \sum_{i=0}^{d} \mathbf{V}_i \, R_{i,d}(u) \tag{37}$$

Rewriting the rational Bézier curve in the form given by equation (37) reveals a curve formulation that is indistinguishable from the integral form except that the basis functions are themselves rational. This immediately implies that virtually all the results for integral curves will carry over to the corresponding rational curve provided that the rational basis functions satisfy the same properties as their integral brethren. In addition, note that the integral Bézier curve is always available as a special case by easily arranging that the denominator be unity.

The rational Bézier curve can be generalized to a tensor-product surface. For degree d in the u parametric direction and degree e in the v parametric direction, the tensor-product rational Bézier surface, denoted by $\mathbf{Q}_{d,e}(u,v)$ where $u \in [0,1]$ and $v \in [0,1]$, can be defined as:

$$\mathbf{Q}_{d,e}(u,v) = \frac{\sum_{i=0}^{d}\sum_{j=0}^{e} w_{ij}\, \mathbf{V}_{ij}\, B_{i,d}(u)\, B_{j,e}(v)}{\sum_{r=0}^{d}\sum_{s=0}^{e} w_{rs}\, B_{r,d}(u)\, B_{s,e}(v)} \tag{38}$$

Analogous to the case for curves, each three-dimensional control vertex is represented as the homogeneous control vertex $\mathbf{V}_{ij}^{w}\ (w_{ij}x_{ij}, w_{ij}y_{ij}, w_{ij}z_{ij})$ in four-dimensional space and is associated with a weight w_{ij}, for $i = 0, \cdots, d$ and $j = 0, \cdots, e$.

Following a similar derivation to that presented for the curve case, the above expression (38) for the rational Beta-spline surface can be rearranged as follows

$$\mathbf{Q}_{d,e}(u,v) = \sum_{i=0}^{d}\sum_{j=0}^{e} \mathbf{V}_{ij} \left[\frac{w_{ij}\, B_{i,d}(u)\, B_{j,e}(v)}{\sum_{r=0}^{d}\sum_{s=0}^{e} w_{rs}\, B_{r,d}(u)\, B_{s,e}(v)} \right] \tag{39}$$

and the term in brackets could be denoted by $R_{i,d;j,e}(u,v)$

$$R_{i,d;j,e}(u,v) = \frac{w_{ij}\, B_{i,d}(u)\, B_{j,e}(v)}{\sum_{r=0}^{d}\sum_{s=0}^{e} w_{rs}\, B_{r,d}(u)\, B_{s,e}(v)} \tag{40}$$

and replacing this in equation (39) yields

$$\mathbf{Q}_{d,e}(u,v) = \sum_{i=0}^{d}\sum_{j=0}^{e} \mathbf{V}_{ij}\, R_{i,d;j,e}(u,v) \tag{41}$$

Analogous to the curve case, the expression (41) yields a surface formulation that resembles the integral form with the exception that the bivariate basis functions $R_{i,d;j,e}(u,v)$ are rational. These rational bivariate basis functions possess properties similar to those of the analogous univariate versions.

In summary, the rational Bézier form provides a unified representation for free-form curves and surfaces along with conic sections and quadric surfaces, is invariant under projective transformation, and possesses weights which can be used to control shape in a manner similar to shape parameters.

10. Conclusion

Parametric Bernstein/Bézier curves and parametric tensor-product Bernstein/Bézier surfaces have been described. This curve and surface representation uses Bernstein polynomials as basis functions. The parametric form represents each coordinate by its own separate, independent function. The Bézier curve representation was explained, and the mathematics presented.

The key properties of Bézier curves were discussed, including the relationships between the parametric derivatives at each end of the curve and the control vertices, degree-raising, the local control property, the variation-diminishing property, and the convex hull property.

The single Bézier curve was extended to a composite Bézier curve using parametric continuity. A discussion of the more general geometric continuity then ensued. Geometric continuity of order two (G^2) constrains the unit tangent and curvature vectors to be continuous. We then generalized geometric continuity to higher order, providing a reparametrization definition of geometric continuity of order n, denoted G^n, for an arbitrary n. From this, equations which are called the Beta-constraints were derived; these provide a set of necessary and sufficient conditions that two parametrizations meet with G^n continuity. The Beta-constraints were used to form a composite Bézier curve that is geometrically continuous. Constraints on the control vertices were derived such that two Bézier parametrizations would meet with geometric continuity and the construction for geometrically continuous Bézier curves was provided.

The subdivision of Bézier curves was then derived. A recursive expression was derived for the control vertices that define each of the two new pieces. This expression was interpreted as a geometric construction for a Bézier curve, and reference was made to the deCasteljau Algorithm. The determination of a set of points on the curve using recursive subdivision with an associated flatness test was also discussed.

Then, the Bézier curve was generalized to a tensor-product surface, and both a summation and matrix formulation for a point on the surface was given. Finally, the rational form was presented. The rational Bézier curve and rational tensor-product surface were discussed and mathematical expressions for both were provided.

Acknowledgements

This work was supported in part by a National Science Foundation Presidential Young Investigator Award (number CCR-8451997). This paper draws heavily from material in.[2,3,5,6,7,8,9,11,28,32]

References

[1] Brian A. Barsky, *The Beta-spline: A Local Representation Based on Shape Parameters and Fundamental Geometric Measures*, Ph.D. Thesis, University of Utah, Salt Lake City, Utah, December, 1981.

[2] Brian A. Barsky, "A Description and Evaluation of Various 3-D Models," *IEEE Computer Graphics and Applications*, Vol. 4, No. 1, January, 1984, pp. 38-52.Earlier version published in *Proceedings of InterGraphics '83*, Japan Management Association, Tokyo, 11-14 April 1983, pp. (B2-5) 1 to 21 and reprinted in *Computer Graphics -- Theory and Applications*, edited by Tosiyasu L. Kunii, Springer-Verlag, Tokyo, 1983, pp. 75-95.

[3] Brian A. Barsky, *Arbitrary Subdivision of Bézier Curves*, Technical Report No. UCB/CSD 85/265, Computer Science Division, Electrical Engineering and Computer Sciences Department, University of California, Berkeley, California, USA., November, 1985.

[4] Brian A. Barsky, *Computer Graphics and Geometric Modeling Using Beta-splines*, Springer-Verlag, Heidelberg, 1988.

[5] Brian A. Barsky, "Introducing the Rational Beta-spline," pp. 16-27 in *Proceedings of the Third International Conference on Engineering Graphics and Descriptive Geometry, Vol. 1*, Vienna, 11-16 July 1988.

[6] Brian A. Barsky, "An Intuitive Description of Parametric Splines in Computer Graphics," pp. 252-266 in *Proceedings of Graphics Interface '90*, Canadian Man-Computer Communications Society, Halifax, 14-18 May 1990. Invited Keynote Address.

[7] Brian A. Barsky and Tony D. DeRose, *Geometric Continuity of Parametric Curves*, Technical Report No. UCB/CSD 84/205, Computer Science Division, Electrical Engineering and Computer Sciences Department, University of California, Berkeley, California, USA, October, 1984.

[8] Brian A. Barsky and Tony D. DeRose, "Geometric Continuity of Parametric Curves: Three Equivalent Characterizations," *IEEE Computer Graphics and Applications,* Vol. 9, No. 6, November, 1989, pp. 60-68.

[9] Brian A. Barsky and Tony D. DeRose, "Geometric Continuity of Parametric Curves: Constructions of Geometrically Continuous Splines," *IEEE Computer Graphics and Applications,* Vol. 10, No. 1, January, 1990, pp. 60-68.

[10] Brian A. Barsky and Tony D. DeRose, "Deriving the Beta-constraints for Geometric Continuity of Parametric Curves." Submitted for publication.

[11] Richard H. Bartels, John C. Beatty, and Brian A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling,* Morgan Kaufmann Publishers, Inc., San Mateo, California, 1987.

[12] Pierre E. Bézier, *Emploi des machines à commande numérique,* Masson et Cie., Paris, 1970. Translated by Forrest, A. Robin and Pankhurst, Anne F. as *Numerical Control -- Mathematics and Applications,* John Wiley and Sons, Ltd., London, 1972.

[13] Pierre E. Bézier, "Mathematical and Practical Possibilities of UNISURF," pp. 127-152 in *Computer Aided Geometric Design,* ed. Barnhill, Robert E. and Riesenfeld, Richard F., Academic Press, New York, 1974.

[14] Pierre E. Bézier, *Essai de définition numérique des courbes et des surfaces expérimentales,* Ph.D. Thesis, l'Université Pierre et Marie Curie, Paris, February, 1977.

[15] Pierre E. Bézier, *The Mathematical Basis of the UNISURF CAD System,* Butterworths Scientific Ltd., Guildford, Surrey, England, May 1986.

[16] Pierre E. Bézier, *Mathématiques et CAO 4: Courbes et surfaces,* Editions Hermès, Paris, 1987.

[17] Wolfgang Boehm, "Curvature Continuous Curves and Surfaces," *Computer Aided Geometric Design,* Vol. 2, No. 4, December, 1985, pp. 313-323.

[18] Wolfgang Boehm, Gerald Farin, and Jurgen Kahmann, "A Survey of Curve and Surface Methods in CAGD," *Computer Aided Geometric Design,* Vol. 1, No. 1, July, 1984, pp. 1-60.

[19] Paul de Faget de Casteljau, *Mathématiques et CAO 2: Formes à poles,* Editions Hermès, Paris, 1985.

[20] Paul de Faget de Casteljau, *Mathematics and CAD Volume 2: Shape Mathematics and CAD,* Kogan Page Ltd., London, 1986.

[21] J. L. Coolidge, *A History of the Conic Sections and Quadric Surfaces,* Oxford University Press, 1945.

[22] Anthony D. DeRose, *Geometric Continuity: A Parametrization Independent Measure of Continuity for Computer Aided Geometric Design,* Ph.D. Thesis, University of California, Berkeley, California, August, 1985. Also available as Technical Report No. 86-09-04, Department of Computer Science, FR-35, University of Washington, Seattle, WA 98195, and as Technical Report No. UCB/CSD 86/255, Computer Science Division, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley.

[23] Tony D. DeRose and Brian A. Barsky, "An Intuitive Approach to Geometric Continuity for Parametric Curves and Surfaces," pp. 343-351 in *Proceedings of Graphics Interface '85,* Montreal, 27-31 May 1985. Extended abstract in *Proceedings of the International Conference on Computational Geometry and Computer-Aided Design,* New Orleans (5-8 June 1985), pp. 71-75. Revised version published in *Computer-Generated Images -- The State of the Art,* edited by Nadia Magnenat-Thalmann and Daniel Thalmann, Springer-Verlag, 1985, pp. 159-175.

[24] Gerald Farin, "Visually C2 Cubic Splines," *Computer-Aided Design,* Vol. 14, No. 3, May, 1982, pp. 137-139.

[25] Gerald Farin, "Algorithms for Rational Bézier Curves," *Computer-Aided Design,* Vol. 15, No. 2,

March, 1983, pp. 73-77.

[26] Gerald Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, Boston, 1988.

[27] Ivor D. Faux and Michael J. Pratt, *Computational Geometry for Design and Manufacture*, Ellis Horwood Ltd., Chichester, 1979.

[28] Alain Fournier and Brian A. Barsky, "Geometric Continuity with Interpolating Bézier Curves (Extended Summary)," pp. 337-341 in *Proceedings of Graphics Interface '85*, Montreal, 27-31 May 1985. Revised version published in *Computer-Generated Images -- The State of the Art*, edited by Nadia Magnenat-Thalmann and Daniel Thalmann, Springer-Verlag, 1985, pp. 153-158.

[29] Ronald N. Goldman and Brian A. Barsky, "On β-continuous Functions and Their Application to the Construction of Geometrically Continuous Curves and Surfaces," pp. 299-311 in *Mathematical Methods in Computer Aided Geometric Design*, ed. Lyche, Tom and Schumaker, Larry L., Academic Press, Boston, 1989. Conference held 16-22 June 1988 in Oslo.

[30] Ronald N. Goldman and Brian A. Barsky, "β-continuity and Its Application to Rational Beta-splines," pp. 5-11 in *Proceedings of the Computer Graphics '89 Conference*, Smolenice, Czechoslovakia, 15-18 May 1989. Also Tech. Report No. UCB/CSD 88/442, Computer Science Division, Electrical Engineering and Computer Sciences Department, University of California, Berkeley, California, USA, August, 1988.

[31] Christoph M. Hoffmann, *Geometric and Solid Modeling*, Morgan-Kaufmann Publishers, Inc., San Mateo, California, 1989.

[32] Michael E. Hohmeyer and Brian A. Barsky, "Rational Continuity: Parametric, Geometric, and Frenet Frame Continuity of Rational Curves," *ACM Transactions on Graphics*, Vol. 8, No. 4, October, 1989, pp. 335-359.Special issue on Computer-Aided Geometric Design and Geometric Modeling.

[33] Jeffrey M. Lane and Richard F. Riesenfeld, "Bounds on a Polynomial," *BIT*, Vol. 21, No. 1, 1981, pp. 112-117.

[34] Dinesh Manocha and Brian A. Barsky, "Basis Functions for Rational Continuity," in *Proceedings of CG International '90*, Singapore, 26-30 June 1990. To appear.

[35] Dinesh Manocha and Brian A. Barsky, "Varying the Shape Parameters of Rational Continuity," in *Proceedings of IMACS'91*, Dublin, 22-26 July 1991. To appear.

[36] E. A. Maxwell, *Methods of Plane Projective Geometry Based on the Use of General Homogeneous Coordinates*, Cambridge University Press, Cambridge, England, 1946.

[37] E. A. Maxwell, *Co-ordinate Geometry with Vectors and Tensors*, Oxford University Press, 1958.

[38] E. A. Maxwell, *General Homogeneous Coordinate in the Space of Three Dimensions*, Cambridge University Press, London, 1951, 1959.

[39] Michael A. Penna and Richard R. Patterson, *Projective Geometry and its Applications to Computer Graphics*, Prentice-Hall, Englewood Cliffs, 1986.

[40] G. Salmon, *A Treatise on Conic Sections*, Longmans, Green, & Co., London, 1879. Sixth edition. Reprinted by Dover Publications Inc., New York.

[41] Isaac J. Schoenberg, "On Spline Functions," pp. 255-291 in *Inequalities*, ed. Shisha, Oved, Academic Press, New York, 1967. With supplement by T. N. E. Greville. Also Math. Res. Center Report 625, University of Wisconsin, Madison, May 1966.