중점 기울기 핸들에 의한 베지에 곡선 제어

주 우 석† · 장 혁 수†† · 임 태 식†††

요 약

단순하면서도 다양한 곡선과 곡면 모습을 생성할 수 있기 때문에 캐드나 그래픽 패키지에서 빈번히 사용되는 스플라인 중 하나가 베지에 곡선이다. 곡선 설계의 편의성 면에서 볼 때 이 곡선이 지닌 가장 큰 장점은 부드러움을 유지하면서도 곡선의 양 끝점에서 사용자가 제시한 기울기 방향을 따라 간다는 점이다. 본 논문은 확장된 베지에 곡선의 공식을 유도함으로써, 곡선의 중점 부근에서도 사용자가 제시한 기울기 방향을 따라갈 수 있도록 하였다. 이는 사용자가 제시한 기울기 벡터를 분석하고 그 결과를 베지에 기반함수 행렬에 매개변수의 형태로 반영함으로써 가능하다. 결과적으로 사용자로서는 양 끝점을 포함하여 중점에 또 하나의 기울기 핸들을 사용하여 추가적으로 더욱 정밀한 베지에 곡선을 생성할 수 있게 된다.

Control of Bezier Curve Shapes by Midpoint Slope Handle

Wouseok Jou† · Hyuk-soo Jang†† · Tae-shik Lim†††

ABSTRACT

One of the most frequently used spline scheme in CAD and graphics package is Bezier curve. Although simple and easy to implement, it supports diverse kinds of curves and surfaces. In view of the design convenience, the main advantage of the Bezier curves is that they observe user-specified slope conditions at both endpoints while maintaining smoothness. This paper expands the advantage by deriving equations for generalized Bezier curves, and applying the equation to observe additional slope condition at midpoint. This is possible by decomposing and analyzing the user-specified midpoint slope, and reflecting the result back into the Bezier basis matrix in parametric form. Consequently, users can control the curve shapes not only by the endpoint slope handles but also at the midpoint slope handle, which helps them to be able to apply more accurate control over the conventional Bezier curve shapes.

1. INTRODUCTION

Bezier was the originator of an early CAD system, UNISURF used by Renault car company. The theory of Bezier curve is elegant and has geometric interpretation, making it one of the most popular spline forms. A Bezier curve $p(t)$ of degree n can be defined in terms of a set of control points p_i $(i=0,1,2,...,n)$ and is given by

$$p(t) = \sum_{i=0}^{n} p_i B_i^n(t) \tag{1}$$

with $B_i^n(t) = C_{n,i}\ t^i\ (1-t)^{n-i}$ where $C_{n,i}$ is a binomial coefficient $C_{n,i} = n!\ /\ i!(n-i)!$

Bezier derived Eq. (1) by purely geometric analysis. But later it was known that before Bezier derived the equation, Bernstein has already derived it by algebraic method. The curve scheme can be characterized by the following property:

i) In general, the curve passes through only the first and last control point. It does not pass through the rest

† 정 회 원 : 명지대학교 컴퓨터공학부 교수
†† 정 회 원 : 명지대학교 정보통신공학부 교수
††† 정 회 원 : 명지대학교 대학원 컴퓨터공학부
논문접수 : 2000년 6월 10일, 심사완료 : 2000년 8월 28일

of the control points. Hence it belongs to the class of approximating spline.

ii) The curve always resides inside the control polygon formed by connecting control points. In other words, the curve satisfies convex hull property.

iii) The curve is invariant under an affine transformation. Therefore applying linear transform to the control points and rebuilding the curve has exactly the same effect as the application of transform to every point on the curve.

iv) The slope of the curve at the first and last control point coincides with the control polygon. More specifically, the first derivative of an order n Bezier curve evaluated at the control points is $n(p_1 - p_0)$ and $n(p_n - p_{n-1})$ respectively. For instance, the tangent of cubic Bezier at the first point is $3(p_1 - p_0)$.

Among Bezier curves, cubic Bezier is most widely used. With third order, the curve location can be evaluated in a relatively quick time and the resulting locality is fairly acceptable. An alternative convention for specifying a Bezier curve is the matrix convention. With the convention, the basis function $B_i^n(t)$ in Eq. (1) can be decomposed into a row matrix and a basis matrix which shows the characteristics of the curve. For instance, the cubic Bezier can be represented as follows:

$$p(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{2}$$

In this paper, we generalize the cubic Bezier basis matrix in Eq. (2) such that the property iv) can be relaxed controlled. We will represent the basis matrix in a parametric form with the parameter being the tangential magnitude at endpoints. Furthermore, we will provide a method to determine the parameter value by assigning a tangent vector near middle of the curve. The remainder of this paper is organized as follows: Section 2 provides the review of the previous research related to the manipulation of Bezier curve shape. Section 3 explains how the generalized Bezier basis can be derived and how it can be used in connection with midpoint slope handle. Section 4 estimates and compares our method with the original Bezier curve scheme. Finally, we provide our concluding remarks in Section 5.

2. PREVIOUS WORKS

Cubic spline interpolation was introduced into the literature of computer-aided geometric design by J. Ferguson [1], while the mathematical theory was studied in approximation theory [2, 3]. Classification and matrix representation of the cubic spline appears in [4]. Detailed definition and property of Bezier curve can be found in [5-7]. Control of curve shapes by the concept of bias and tension can be found in [8-10]. In this context, the bias parameter controls the curve behavior near the control points. By tension, the curve between control points can be controlled to be flat or reflexed. The importance of this approach is that the shape parameter is incorporated into corresponding basis matrix.

Given control points and corresponding parameter values, the curve that passes through the control points are known as Catmull-Rom spline or Cardinal Spline [7, 11]. Given two control points and corresponding tangent vectors, the cubic curve that passes through the control points are known as Hermite interpolation spline [5, 6]. The curve was generated by extrapolating two additional control points to guide the curve shape. As can be verified, Hermite spline collapses to ordinary Bezier curve by simple algebraic manipulation [6].

Control of Bezier curve shape by assigning weight values to control points can be found in [12, 13]. The primary purpose of this approach is to assure smooth joint connectivity between piecewise Bezier curves, but only at the expense of the curve being rational Bezier function. As for the joint connectivity, Faux and Pratt [14] pointed out that the co-linearity of control points near joint is too severe in practical CAD applications, and proposed an algorithm to relax the condition for application to Bezier surface patch. Previous researches that manipulate Bezier curve shape has been performed mainly in the context of the control point constraints, and joint connectivity between piecewise Bezier curves. In this paper, we propose a new method to control the curve shape by specifying a tangent near middle of the control points.

3. PROPOSED BASIS AND MIDPOINT HANDLE

3.1 Derivation of Generalized Bezier Basis

In an attempt to allow more degrees of freedom on the curve shape, we roughly represent cubic Bezier curves in the following form where all the elements of the basis matrix are set to unknown parameters a_{ij}.

$$p(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{3}$$

Starting from the generalized basis matrix, we determine relations between a_{ij}'s by selectively applying Bezier constraints as follows:

i) Symmetry Constraints

We restrict the shape of the basis functions $B_1^3(t)$ and $B_2^3(t)$ to be symmetric with respect to the center of the parameter space. The basis functions $B_0^3(t)$ and $B_3^3(t)$ are also restricted to be symmetric. The basis function of a given control point has the effect of attracting the curve shape toward the control point itself. By restricting this way, we can guarantee that the curve shape is always symmetric if control points are symmetric. Applying these constraints to Eq. (3), we have

$$\begin{aligned} &a_{13} = -a_{12}, \quad a_{23} = 3a_{12} + a_{22}, \\ &a_{33} = -3a_{12} - 2a_{22} - a_{32}, \\ &a_{43} = a_{12} + a_{22} + a_{32} + a_{42} \end{aligned} \tag{4}$$

$$\begin{aligned} &a_{14} = -a_{11}, \quad a_{24} = 3a_{11} + a_{21}, \\ &a_{34} = -3a_{11} - 2a_{21} - a_{31}, \\ &a_{44} = a_{11} + a_{21} + a_{31} + a_{41} \end{aligned} \tag{5}$$

ii) Endpoint Positional Constraints

We let the curve to pass through both endpoints exactly. For this purpose, value of the basis function $B_0^3(t)$ is set to be 1 at control point p_0. Values of the other basis functions are set to zero at the point. Similarly, for the control point p_3, the basis function $B_3^3(t)$ evaluated at t = 1 is set to 1, and others are set to 0. We apply these constraints to Eq. (3). If the resulting equation is combined with Eq. (4) and (5), we get

$$\begin{aligned} &a_{13} = -a_{12}, \quad a_{23} = 3a_{12} + a_{22}, \\ &a_{33} = -2a_{12} - a_{22}, \\ &a_{43} = 0, \quad a_{32} = -a_{12} - a_{22}, \quad a_{42} = 0 \end{aligned} \tag{6}$$

$$\begin{aligned} &a_{14} = -a_{11}, \quad a_{24} = 3a_{11} + a_{21}, \\ &a_{34} = -2a_{11} - a_{21} + 1, \\ &a_{44} = 0, \quad a_{31} = -a_{11} - a_{21} - 1, \quad a_{41} = 1 \end{aligned} \tag{7}$$

Incorporating Eq. (4)-(7) into Eq. (3), the basis matrix of Eq. (3) can now be represented as follows: Here, each element of the basis matrix can be represented with only 4 parameters a_{11}, a_{12}, a_{21}, and a_{22}:

$$\begin{bmatrix} a_{11} & a_{12} & -a_{12} & -a_{11} \\ a_{21} & a_{22} & 3a_{12} + a_{22} & 3a_{11} + a_{21} \\ -a_{11} - a_{21} - 1 & -a_{12} - a_{22} & -2a_{12} - a_{22} & -2a_{11} - a_{21} + 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

iii) Endpoint Tangential Constraints

Although our current basis is represented in more a general form, ordinary Bezier basis still also meets the requirements of the constraints i) and ii). The most crucial difference between the ordinary Bezier basis and our generalized Bezier basis comes from the tangential endpoints constraints. Ordinary Bezier curves have strict requirement that tangential magnitude must be 3 at the endpoints. Here we will deviate from the magnitude regulation, and let our generalized Bezier basis to be flexible in the selection of the tangential magnitude.

Differentiating the curve represented by the matrix (8) with respect to the parameter t, we get the following basis matrix:

$$\begin{bmatrix} 3a_{11} & 3a_{12} & -3a_{12} & -3a_{11} \\ 2a_{21} & 2a_{22} & 6a_{12}+2a_{22} & 6a_{11}+2a_{21} \\ -a_{11}-a_{21}-1 & -a_{12}-a_{22} & -2a_{11}-a_{22} & -2a_{11}-a_{21}+1 \end{bmatrix} \quad (9)$$

The slopes of the curve represented by the basis matrix (9) can be evaluated at starting and ending control points, yielding

$$p'(0)=(-a_{11}-a_{21}-1)p_0+(-a_{12}-a_{22})p_1+(-2a_{12}-a_{22})p_2+(-2a_{11}-a_{21}+1)p_3 \quad (10)$$

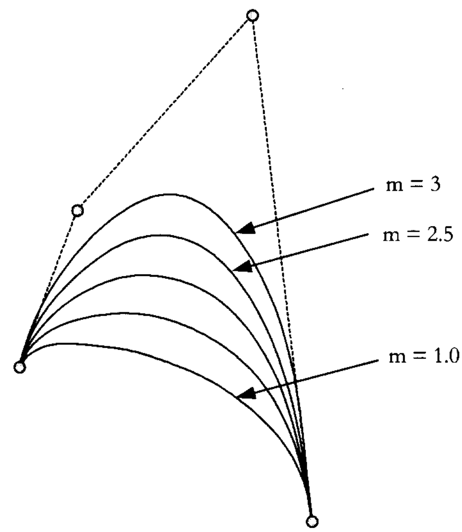
$$p'(1)=(2a_{11}+a_{21}-1)p_0+(2a_{12}+a_{22})p_1+(a_{12}+a_{22})p_2+(a_{11}+a_{21}+1)p_3 \quad (11)$$

It means that the endpoint slope can be represented as a function of the basis parameters and all of the control points. Linear combination of the basis parameters are multiplied to each control point, and the result is summed up. In terms of the starting point slope in Eq. (10), Bezier basis is a special case that the coefficients of the control points p_2, p_3 are all zero such that the effects due to the control points are nullified. Moreover, Bezier basis is also a special case that the coefficients of p_0 and p_1 equals -3 and 3 respectively, thus yielding the starting point slope of $3(p_1-p_0)$.

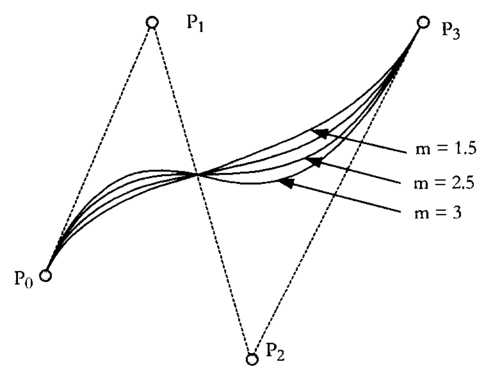
By postulating this starting point slope as $m(p_1-p_0)$, we do generalize Bezier basis. It is possible by setting the coefficient of p_1 as m, and solving the other coefficient as a function of m. Here, the parameter m means the magnitude of the slope vector at the starting control point. The direction remains the same as the ordinary Bezier curves. Nevertheless, now the curve shape gets greater flexibility by the ability to control the slope magnitude. If we apply the parameter m at starting control points, the slope at the ending control points is automatically assigned the value $m(p_3-p_2)$ by the intrinsic relationship between Eq. (10) and (11). Consequently, our generalized Bezier basis can be represented in terms of the single, tangential-magnitude parameter m as follows:

$$\begin{bmatrix} 2-m & m & -m & m-2 \\ 2m-3 & -2m & m & 3-m \\ -m & m & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (12)$$

The curves produced by the variation of m value is shown in (Fig. 1) and (Fig. 2). Even if all the control points remain fixed, the curve shapes can be controlled to fit into the control polygon either tightly or loosely.



(Fig. 1) Variation of the Parameter m



(Fig. 2) Variation of the Parameter m

Here, the case $m = 3$ corresponds to the ordinary Bezier curve. In other words, our generalized Bezier basis forms a set of curves which is a superset of the curves drawn by conventional Bezier curve scheme. Notice that, as the tangential magnitude m gets smaller, the curve deviates earlier from the edges near endpoints. But still, the direction of the curve strictly follows the endpoint edges.

3.2 Midpoint Slope Handle

The basis matrix (12) can be used to control the curve shape by adjusting the magnitude of endpoint slope, namely the parameter m. If it's larger, the curve will follows the edges $p_0 p_1$ and $p_2 p_3$ for the longer times. The entire curve shape is changed as a result. However, the limitation of this type of control is that graphical users can specify the curve behavior only near the endpoints. Often times, however, the users want to define the curve shape by specifying a curve behavior in between the control polygon.

Thinking backward, we claim that the user-specified midpoint behavior can be used to decide the value of the endpoint slope parameter m. In deriving the basis matrix (12) from Eq. (10) and (11), we deliberately removed the effect of $(p_3 - p_2)$ at the starting control point and $(p_1 - p_0)$ at the ending control point. However, the effect of these vectors are usually non-zero in between the two control points. In this paper, we concentrate on the curve behavior near midpoint, where the normalized parameter t has value of .5. At this point, the users are allowed to specify a slope vector, which in turn can be used to determine the endpoint slopes.

We differentiate the curve represented by the basis matrix (12). Rearranging the resultant equation, we get the slope vector S at midpoint as follows:

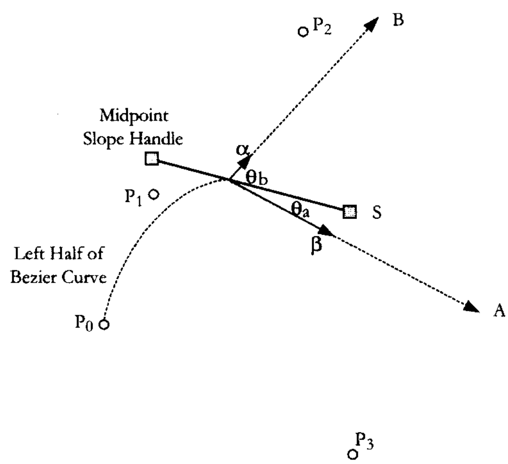
$$S = p'(.5) = \left(\frac{3}{2} - \frac{m}{4}\right)(p_3 - p_0) + \left(\frac{m}{4}\right)(p_2 - p_1)$$
$$= \left(\frac{3}{2} - \frac{m}{4}\right) A + \left(\frac{m}{4}\right) B = \alpha + \beta \qquad (13)$$

It means that the user-specified midpoint slope in the left hand side of the equation can be decomposed by two vectors α and β, whose direction is toward $(p_3 - p_0)$ and $(p_2 - p_1)$ respectively. From the specified slope, we solve this equation for the value of the unknown parameter m. Taking the ratio of the first and second term of Eq. (13), we get

$$\frac{|\alpha|}{|\beta|} = \frac{\left(\frac{3}{2} - \frac{m}{4}\right)|A|}{\left(\frac{m}{4}\right)|B|} = \left(\frac{6}{m} - 1\right)\frac{|A|}{|B|} \qquad (14)$$

Solving this equation, the unknown parameter m can be expressed as follows:

$$m = 6 \Big/ \left(\frac{|\alpha|}{|\beta|} \cdot \frac{|B|}{|A|} + 1\right) \qquad (15)$$



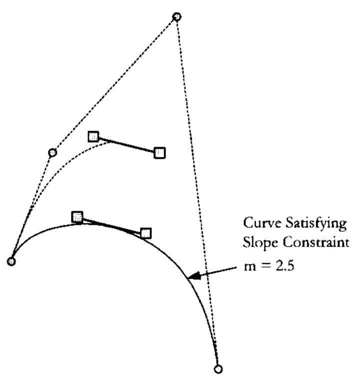
(Fig. 3) Midpoint Slope Handle

Once the user specifies the midpoint slope as in (Fig. 3), the absolute value of A, B can be calculated by estimating the vectors $(p_3 - p_0)$ and $(p_2 - p_1)$ respectively. Let us denote the angle formed by A vector and the midpoint slope vector as θ_a, and the angle formed by B and the midpoint slope vector as θ_b. By the geometry and the sinusoidal rule of trigonometric functions, we have $|\alpha| / |\beta| = \sin\theta_a / \sin\theta_b$. The angles can be calculated by the general definition of inner product, namely $p \cdot q = |p||q| \cos\theta$.

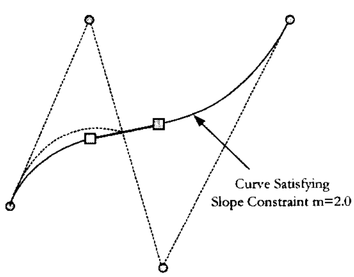
Inserting these relations into Eq. (15), we have the final relation between the slope vector and the tangential magnitude parameter m in the basis matrix (12).

$$m = 6 / \left(\frac{|\sin(\cos^{-1}(B \cdot S / |B||S|))|}{|\sin(\cos^{-1}(A \cdot S / |A||S|))|} \cdot \frac{|B|}{|A|} + 1 \right) \quad (16)$$

In fact, the selection of the midpoint tangent drives the production of the curve with the corresponding tangent as shown in (Fig. 4) and (Fig. 5). As a result, the midpoint position of the original Bezier curve may not necessarily be maintained. What we have designated with midpoint tangent handle is that we wish to draw a curve with that tangent at midpoint, not that we wish to keep the designated midpoint position.



(Fig. 4) Midpoint Handle Example 1



(Fig. 5) Midpoint Handle Example 2

The initial midpoint tangent handle can be placed anywhere near middle as far as the user interface is concerned. Usually, this operation can be facilitated by drawing the left half of the original Bezier curve and positioning the handle at the right end, as shown in the upper curve of (Fig. 4). Right after the slope constraint is designated there, the curve shape and the midpoint position changes to satisfy the given constraint. Therefore, the position of the midpoint used in designating the slope handle does not affect the final curve shape.

4. EVALUATION AND COMPARISON

4.1 Convexity

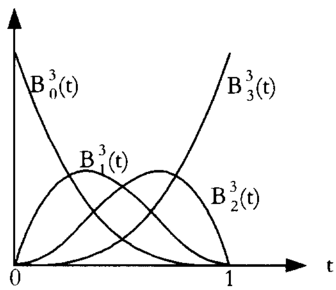
In order to obtain the convexity property, our generalized Bezier basis must obey two conditions.

i) Sum of the weighting functions must be 1 for all values of t:

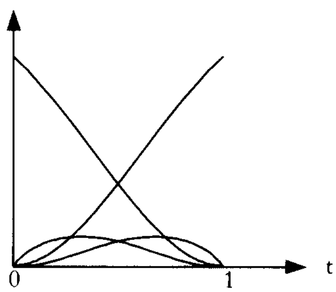
In the basis matrix of Eq. (12), the sum of the 1st row, which is the coefficient of t^3 equals zero. Similarly, the coefficient of t^2 and t equals zero, respectively. This is true for any t value. Only the sum of the last row remains, and it always equals 1. Hence our generalized Bezier basis satisfies this property.

ii) Weighting functions must always be non-negative for all values of t:

In order for the weighting functions $B_i^3(t)$ in the basis matrix (12) to satisfy this property, the range of the parameter m must be limited to $0 \le m \le 3$. The range can be calculated by factoring inequalities of all of the weighting functions. Notice that when our generalized Bezier has m value of 3, it falls into the ordinary Bezier. Therefore, if we want our generalized Bezier basis to satisfy the convexity, the maximum achievable tangential magnitude at endpoints is that of ordinary Bezier basis, namely $m = 3$. In order to keep convexity, this restriction must be incorporated in designating the midpoint slope handle. (Fig. 7) demonstrates an example of the generalized Bezier basis when $m = 1$. Although the shape of each basis function is different from (Fig. 6), the property i) and ii) are satisfied, and convexity is preserved.



(Fig. 6) Conventional Bezier Basis

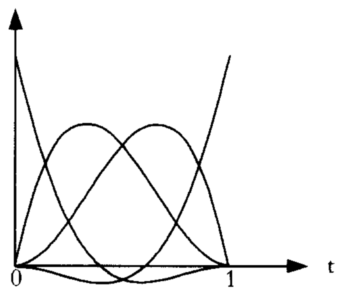


(Fig. 7) Generalized Bezier Basis 1

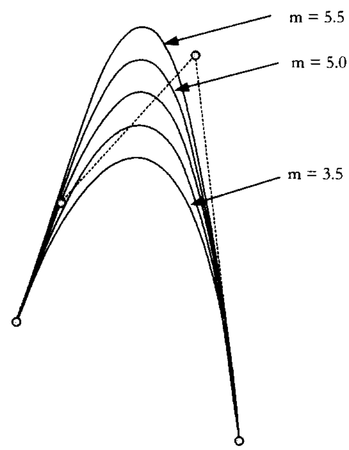
Sometimes, the convexity property is too stern a requirement for the users to follow. Our generalized Bezier basis provides the users with the freedom to violate the condition ii). They can increase their design flexibility by simply setting the parameter m to be outside the range, yielding negative weighting functions on some interval. A force repelling to a given control point can be exerted. As a result, the curve shape gets farther away from the control point and it may cross the control polygon boundaries. (Fig. 8) shows an example of the convexity violation with the parameter m = 4.5, and (Fig. 7) shows a shape of the basis function in such a case. As can be seen readily, some of the basis functions does not satisfy the property ii) and the negative basis function appears. However, the property i) is still satisfied since we incorporated that condition into the derivation of the generalized Bezier basis.

4.2 Conversion to Ordinary Bezier Curves

Our generalized Bezier curves can readily be



(Fig. 8) Generalized Bezier Basis 2



(Fig. 9) Violation of Convexity

converted to the ordinary Bezier curves. By simply moving the control point locations, the conversion can be accomplished. In order to equate Eq. (12) with Eq. (2), the terms containing the magnitude parameter m must melt down into the positional matrix [p_0 p_1 p_2 p_3]. Rearranging and modifying the resulting equations, we get the following vector relation:

$$p_1' = p_0 + \left(\frac{m}{3}\right)(p_1 - p_0)$$

$$p_2' = p_3 + \left(\frac{-m}{3}\right)(p_3 - p_2) \qquad (17)$$

By extending the position of the control points p_1, p_2 into p_1', p_2' along the corresponding edges of the control polygon, the conversion is done. In other words, if we extend the control points and draw the ordinary Bezier curve, the result is the same as the generalized Bezier curve with the original control points.

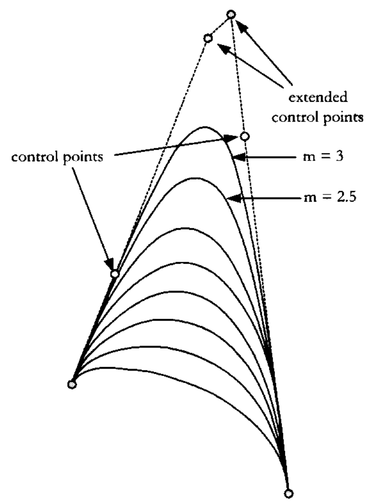
4.3 Joint Conditions

The continuity condition when two piecewise Bezier curves are joined together requires that the endpoint slope must be continuous. This condition assures visual smoothness near the joint when two Bezier curves meet. Say we want to join our cubic Bezier with an arbitrary cubic Bezier which has control points p_4 to p_7. Then the continuity condition states that the control point p_4 must lie on the line connecting p_2 to p_3. That is, the line connecting p_2, p_3, and p_4 must be co-linear.

Mathematically, this means that the incoming direction of the first curve at the control point p_3 must be the same as the outgoing direction of the second curve. The magnitude of the tangent does not matter. Only the direction of the tangent matters. As long as the outgoing tangent is a scalar multiple of the incoming tangent, the visual continuity - namely G^1 continuity - is maintained. From Eq. (17), we see that the tangential direction of our generalized Bezier curve is the same as the original Bezier curve. The only difference is in the tangential magnitude. Therefore, we can conclude that our generalized Bezier curve still satisfies the joint condition without posing any problem.

4.4 Design Interface Considerations

When graphic users want to design a new curve, they first visualize imaginary curve shape in their mind. If they decide to use cubic Bezier curve, the only template they can rely on is the control polygon. By the property of Bezier curves, edges $p_0 p_1$ and $p_2 p_3$ can be a good template to define the curve shape near beginning and ending. But, it's hard for them to predict the curve behavior near the middle of the curve, since Bezier curve belongs to the class of approximating spline. In other words, control points p_1, p_2 are seldom hit by the curve. In our midpoint Bezier handle approach, they first define edges $p_0 p_1$ and $p_2 p_3$ to guide the curve shape near endpoints. However, the position of p_1 and p_2 need not be fixed yet. One thing they should keep in mind is that the positions should better be extended along the edges as in (Fig. 10), so that our generalized Bezier basis could make room for more diversified shape with the restricted parameter range of $0 \le m \le 3$.

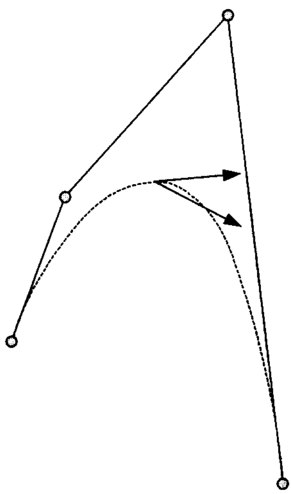


(Fig. 10) Extension of Control Points

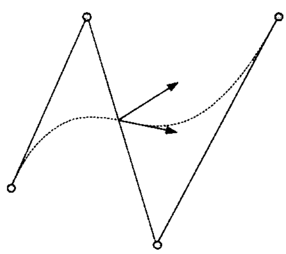
The other thing they should keep in mind is the slope of $p_1 p_2$. As can be seen in Eq. (13), this slope together with the slope of $p_0 p_3$ composes the midpoint slope vector. If one slope is quite different from the other, flexibility in specifying midpoint slope will increase.

In (Fig. 11) and (Fig. 12), the relative span of the control angle is larger than that of (Fig. 13). In practice, whenever the users draw control polygon as in (Fig. 13), they almost decided the direction of the midpoint slope. On the contrary, if they draw them as in (Fig. 11) and (Fig. 12), they are expecting a wide range of angle variation. Hence, this angular restriction is quite natural in the user's point of view.

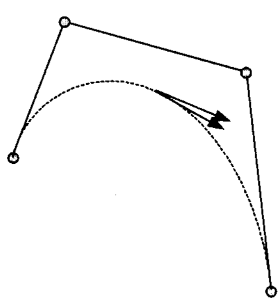
Once the two control points are positioned this way, we get additional template to define the curve shape at the midpoint. Overall, we could control Bezier curve shape by three handles with the proposed method. Two handles to guide the tangential direction of both endpoints of the curve, and one handle to guide the midpoint slope. In practice, the midpoint slope is incorporated in the form of the tangential magnitude at the endpoints.



(Fig. 11) Span Example 1



(Fig. 12) Span Example 2



(Fig. 13) Span Example 3

5. CONCLUDING REMARKS

We have proposed a new method to control Bezier curve shape. Besides the conventional endpoint tangents, the curve can now also be controlled by the midpoint tangent handle. First, we derived a generalized Bezier basis as a function of the tangential magnitude parameter m, relaxing the constraints of the ordinary Bezier curve. Second, we proved how the user-specified midpoint tangent can be incorporated into the tangential parameter. The vector was decomposed and interpreted into directional vectors, which were directly fed into determining the tangential parameter. Accommodation of the midpoint tangent did not affect the direction of the endpoint tangents. It affects only the magnitude. Consequently, we could control the Bezier curve shape with three tangential handles. In using the handles, the specified control points remains fixed and untouched. In terms of convenience, adding a new handle means adding additional flexibility in the design process. Researches on the extension of our method to parametric surface patch should follow in near future.

REFERENCES

[1] J. Ferguson, "Multivariable Curve Interpolation," J. ACM, Vol.2, No.2, pp.221-228, 1964

[2] C. de Booc, "Bicubic Spline Interpolation," J. Math. Phys., Vol.41, pp.212-218, 1962

[3] L. Schumaker, "On Shape Preserving Quadratic Spline Interpolation," SIAM J. Numer. Anal., Vol.20, pp.854-864, 1983

[4] A. R. Smith, "Spline Tutorial Notes," ACM SIGGRAPH 88 Course Notes : Introduction to Computer Animation, pp.131-142, 1988

[5] D. F. Rogers, and J. A. Adams, *Mathematical Elements for Computer Graphics*, McGraw-Hill, New York, New York, 1976

[6] G. Farin, *Curves and Surfaces for Computer Aided*

Geometric Design, Academic Press, Tempe, Arizona, 1990

[7] A. Watt, and M. Watt, *Advanced Animation and Rendering Techniques*, Addison-Wesley, New York, New York, 1992

[8] B. A. Barsky, and J. C. Beatty, "Local Control of Bias and Tension in Beta Spline," Computer Graphics, Vol.17, No.3, pp.193-218, 1988

[9] D. H. Kochanek, and R. H. Bartels, "Interpolating Splines with Local Tension, Continuity, and Bias Control," Computer Graphics, Vol.18, No.3, pp.33-41, 1994

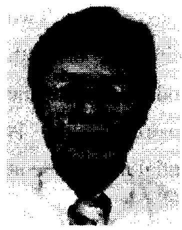
[10] T. A. Foley, "Local Control of Interval Tension Using weighted Spline," Computer Aided Geometric Design, Vol.3, pp.281-294, 1986

[11] T. D. Derose, and B. A. Barsky, "Geometric Continuity, Shape Parameters, and Geometric Constructions for Catmull Rom Splines," ACM Trans. Graphics, Vol.7, No.1, pp.1-41, 1988

[12] T. Saito, and M. Hosaka, "On the extended rational quadratic Bezier curve and its use to curve-fitting methods," Trans. IPS Japan, Vol.31, No.1, pp.33-41, 1990

[13] S. Jiang, M. Xu, H. Anzai, and A. Tamura, "Generation of Rational Cubic Bezier with Given Tangent Vectors," IEICE Trans. Inf. & Syst., Vol.79, No.9, 1996

[14] I. D. Faux, and M. J. Pratt, *Computational Geometry for Design and Manufacture*, Ellis Horwood, Chichester, UK, 1979

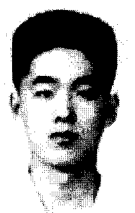


주 우 석

e-mail : red@mju.ac.kr
1976년~1983년 서울대학교 공과대학 전자공학과
1983년~1985년 한국 IBM, 데이콤 정보통신 연구소
1985년~1987년 Univ. of Florida 컴퓨터공학 석사
1987년~1991년 Univ. of Florida 컴퓨터공학 박사
1992년~현재 명지대학교 컴퓨터공학부 부교수
관심분야 : 컴퓨터그래픽스, 인터넷, 데이터베이스

장 혁 수

e-mail : jang-h@mju.ac.kr
1975년~1983년 서울대학교 공과대학 산업공학과
1984년~1986년 Ohio State Univ. 산업시스템공학 석사
1986년~1990년 Ohio State Univ. 전산학 박사
1990년~1992년 경북대 전자공학과 교수
1998년 Ohio State Univ. 전산학과 교환교수
1992년~현재 명지대 전자·정보통신 공학부 부교수
관심분야 : 인터넷, 컴퓨터통신, 컴퓨터구조



임 태 식

e-mail : pinksy@mju.ac.kr
1992년~1999년 명지대학교 공과대학 컴퓨터공학과
1999년~현재 명지대학교 공과대학 컴퓨터공학부석사과정
관심분야 : 컴퓨터그래픽스, 멀티미디어, 데이터베이스