Martin Roche and Wan Xing Li

A Note on Real Time Parametric Cubic Segment Curve Generation

ABSTRACT *As coordinate values are determined, we wish to add them to an array and apply an interpolation procedure to the new coordinate values of the array. That is, as the sequence is being increased, it will be interpolated by a cubic fit. We will exhibit two procedures which limit the cubic construction to be one segment behind the last segment of the sequence. That is the input coordinate values are not the end points for the cubic segment being constructed. Another procedure will include this last input coordinate value as end coordinate values for this last cubic segment being generated. The method of parabolic blending for the curve and surface interpolation originally conceived by A. W. Overhauser [1] is applied as well as two procedures employing three points and one vector. As will be seen, these methods lend themselves to real time curve generation.*

1. Introduction

During testing and sometimes during a general running operation, it is desirable to be able to display a smooth curve (By a smooth curve we mean at a given input defining point p_k, the tangent vector for the cubic segment defined between the points p_{k-1} and p_k has the same direction as the tangent vector for the cubic segment defined between the points p_k and p_{k+1} when each is evaluated at p_k.) that is an interpolation of data as the data is being generated. This differs from the usual problem of interpolating a given string $p_1, \ldots, p_n$ of points since all the points are not given at the time the interpolation is being determined. With our problem we are given the i-th point and want an interpolation curve for the $p_1, \ldots, p_i$ points while the point p_{i+1} is being computed.

In the classical problem of interpolation, spline (or B-spline) functions [2] are quite often used for the interpolation [3]. The point information is available for the classical problem and this is required in applying classical spline theory. Since all points are not available during real time processing, the use of the classical spline (or B-spline) algorithms would be very time consuming for our applications. That is the number of functional operations(multiplications and additions, etc.) rule out consideration of this technique.

Typical of a procedure that might be examined for real time applications is that of Akima [4] which relies on local construction procedures. This method has been discounted because of the number of operations required to compute the coefficients for a cubic segment.

The parabolic blending (or Overhauser) procedure [1] which limits the required information needed to generate a cubic segment is attractive [Fig. 1]. The procedure will be seen to limit the arithmetical operations and to minimize the

Martin Roche and Wan Xing Li, Robotics Assembly Group, National Bureau of Standards, Gaithersburg, Md. 20899.

computational speed. This procedure has in the past been used in generating curves for CAD applications. For these applications the defining points are known before one generates the curve. But this curve type is such that it can lend itself to our application of generation of a curve as the point information is supplied.

Another attractive procedure is the three points and a vector cubic segment definition method. For the three points the vector could be associated with the first point and a cubic segment would be defined between the first and second given points [Fig. 2], or the vector could be associated with the second point and the cubic segment would be defined between the second and third points [Fig. 3]. Both of these cases are considered below. Coefficients for both of these cases can be determined with a minimum number of operations.

It should be noted that in the construction of the curves of Figures 1, 2, and 3, the same input coordinate values were used for all three figures.

2. The Information Needed To Generate A Cubic By The Different Procedures

Three cubic curve interpolation schemes being discussed are:

(i) The parabolic blending (Overhauser curve) procedure;

(ii) The three points and vector definition procedure with the vector attached to the first point;

(iii) The three points and vector definition procedure with the vector attached to the second point.

If the interpolation curve between points p_i and p_{i+1} is desired, then the points p_{i-1} and p_{i+2}, would also have to be known to apply procedure (i). But to apply procedure (ii) we would need the additional points p_{i+2} and a vector which is used to define the tangent vector of the cubic at the point p_i. To apply procedure (iii) we would need the additional point p_{i-1} and a vector V which would be used to define the tangent vector at point p_i. This implies that the curve segment being generated would lag behind the input interval information by only one interval for procedures (i) or (ii), whereas with procedure (iii) it would not lag at all. We mention in passing that for the Akima procedure the points p_{i-2}, p_{i-1}, p_{i+2} and p_{i+3} would have to be known. Hence if this procedure was applied, curve segment generation would be lagging by two points. In addition, the procedure is computationally intense relative to (i),(ii), or (iii). It is for this combination of factors that (i), (ii), or (iii) would be a better choice.

3. Defining Cubic Curve Segments by Parabolic Blending

As the name suggests, parabolic blending involves the concept of using blending functions [5,6] to blend two parametrically defined second degree polynomials into a cubic polynomial over an interval. In particular, for four given points p_1, p_2, p_3 and p_4, a second degree polynomial, p , will be defined by the points p_1, p_2 and p_3 and another second degree polynomial, q , will be defined by the points p_2, p_3, and p_4. We wish to select blending functions B1 and B2 such that: (i) B1 = B1(t) and B2 = B2(t) are functions of the same parameter; (ii) the resultant vector function C(t) = B1 p + B2 q has vector equal to p_2 for t= 0 and vector value equal to p_3 for t= 1 [Fig. 4]. Overhauser [1] selected B1= (1 - t) and B2 = t. Since p and q are second degree vector polynomials, we can write

$$(1)\qquad p = p(r) = (r^2, r, 1)\,B$$
$$\text{and}\qquad q = q(s) = (s^2, s, 1)\,D$$

where both B and D are 3x3 matrices. Writing the parameters r and s as linear functions of t (i.e. r = a t + b and s = c t + d), the resulting vector curve C is a parametric cubic:

$$C(t) = (t^3, t^2, t, 1)\,A\ ,$$

where A is a 4 x 4 matrix.

If we select the r values and the s values such that

$$p(0) = p_1\ ,\ p(1/2) = p_2 \text{ and } p(1) = p_3$$
$$q(0) = p_2\ ,\ q(1/2) = p_3 \text{ and } q(1) = p_4$$

we get from equations (1) that r and s relate to t by the equations

$$(2)\qquad r = 0.5(t+1) \quad\text{and}\quad s = 0.5\,t\ .$$

Using equations (1) we have

$$\begin{bmatrix} p(0) \\ p(1/2) \\ p(1) \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1/4 & 1/2 & 1 \\ 1 & 1 & 1 \end{bmatrix} B = VB$$

Since the inverse of V is

$$\begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

we have

$$B = \begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

and similarly for the matrix , D , of equation (1) in terms of p_2, p_3, and p_4 we have

$$D = \begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

Hence,

(3) $C(t) = (1 - t)\, p + t\, q = (1-t)(r^2,r,1)\ B + t\ (s^2,s,1)\ D$. Substituting equation (2) into (3) we arrive at

$$(4) \qquad C(t) = [t^3\ t^2\ t\ 1]\ M \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix},$$

where

$$M = \begin{bmatrix} -1/2 & 3/2 & -3/2 & 1/2 \\ 1 & -5/2 & 2 & -1/2 \\ -1/2 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

or

$$(4') \qquad C(t) = a\ t^3 + b\ t^2 + c\ t + d \quad ,$$

where

$$a = -1/2\, p_1 + 3/2\, p_2 - 3/2\, p_3 + 1/2\, p_4$$
$$b = p_1 + -5/2\, p_2 + 2\, p_3 + -1/2\, p_4$$
$$c = -1/2\, p_1 + 1/2\, p_3$$
$$d = p_2$$

(where details of the manipulations can be found in [1],[3],or[7]). Note: One must keep in mind that each p_i is a vector, say $p_i = (x_i, y_i, z_i, w_i)$, so that the coefficients determined for the cubics are vectors [we are working with parametric equations].

In generating a complete curve, each curve segment is defined by the use of equation (4). Special consideration is given to the first and the last segment. If p_1, p_2, and p_3 are the first three points generated, then to generate the cubic between p_1 and p_2, set $p_0 = p_1$ and use equation(4). If p_{n-1} is the last point to be generated, then set $p_n = p_{n-1}$ and apply equation (4) to the points p_{n-3}, p_{n-2}, p_{n-1} and p_n. It is worth noting that for a sequence of distinct points (p_i not equal to p_{i+1} for any i) that first derivative continuity at the boundary of two adjacent cubic curve segments is maintained, i.e, a smooth curve is generated (see [7] for details).

4. Cubic Segments Defined By Three Points and A Vector

Given three points P_0, P_1 and P_2, and a vector V, a cubic segment between points P_0 and P_1 can be defined [Fig. 5]. Let C(t) be the cubic polynomial such that

$$C'(0) = V$$

and assume a parameterization such that

$$C(0) = P_0;$$
$$C(1) = P_1;$$
$$C(2) = P_2.$$

If

$$C(t) = at^3 + bt^2 + ct + d$$

is the cubic, then

$$(5) \quad \begin{cases} d = P_0; \\ c = V; \\ C(1) = a + b + c + d = P_1; \\ C(2) = 8a + 4b + 2c + d = P_2. \end{cases}$$

Letting $q_1 = P_1 - (c + d)$ and) $q_2 = P_2 - (2c + d)$ the last two equations can be written as

$$\begin{bmatrix} 1 & 1 \\ 8 & 4 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

Note, the inverse of the 2x2 matrix on the left side of the above expression is

$$\begin{bmatrix} -1 & 1/4 \\ 2 & -1/4 \end{bmatrix}$$

Hence,

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -1 & 1/4 \\ 2 & -1/4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

and

$$(6) \qquad a = -q_1 + 0.25 q_2$$
$$(7) \qquad b = 2q_1 - 0.25 q_2.$$

The coefficients a, b, c, d are seen to be easily determined by employing system (5) and equations (6) and (7).

To define a complete curve using these cubic curve segments the following steps are required:

a. For the first cubic segment the following three steps are required:
(i) The first three points p_0, p_1 and p_2 of a sequence are given;
(ii) We assume at point p_0 the cubic vector segment $C_0(t)$ to be defined has all zero second derivative components. If

$$C_0(t) = at^3 + bt^2 + ct + d$$

is the cubic, then by our assumption b = 0 (the zero vector), and we also have $C_0(0) = p_0 = d$. Then by solving the system

$$C_0(1) = a+c+d=p_1$$

$$C_0(2) = 8a+2c+d=p_2$$

we obtain the coefficient vectors a and c. As this is similar to the above we have skipped the details. Hence the cubic between the points p_0 and p_1 is determined.
(iii) Using the cubic coefficients of (ii), points between p_0 and p_1 is computed.

b. If a cubic segment has been defined between the points p_{i-2} and p_{i-1}, the following steps are taken to determine the cubic for the segment between p_{i-1} and p_i:
(i) p_{i+1} will be given;
(ii) The vector V is determined by the cubic that was defined between p_{i-2} and p_{i-1}. In particular V is set equal to the tangent vector of that cubic evaluated at the point p_{i-1} ;
(iii) The three points p_{i-1},p_i and p_{i+1} and the vector V determine the cubic vector coefficiens as outlined above;
(iv) Points between p_{i-1} and p_i are computed.

c If a cubic segment has been defined between the points p_{n-2} and p_{n-1}, and p_n is to be the last input point, then either of the following methods could be used to determine a cubic segment between p_{n-1} and p_n:
(i) Use the cubic vector segment that was defined between the points p_{n-1} and p_n and compute points with the parameter values varying from 1 (one) to 2 (two). [This Procedure was used in generating Figure 2.]
(ii) A vector V is determined by the cubic vector that was defined between p'_{n-2} and p_{n-1}. In particular V is set equal to the tangent vector of that cubic vector evaluated at the point p_{n-1}. Then the points p_{n-2}, p_{n-1} and p_n and the vector V determine a cubic vector by the procedure to be discussed in section 5,below.

5. A Second Procedure Using Three Points and A Vector

Suppose that a cubic vector polynomial has been defined between points P_{i-1} and P_i. Now given P_{i+1}, consider the generation of a cubic vector polynomial between P_i and P_{i+1}. One way of accomplishing this would be to use the points P_{i-1}, P_i, and P_{i+1} and the slope vector for the point P_i obtained from the cubic defined between P_{i-1} and P_i.

As in the case of the preceeding section, one is given three points P_{-1}, P_0, and P_1, and a vector V and a cubic segment is to be defined between points P_0 and P_1 [Fig. 6]. Let C(t) be the cubic polynomial such that

$$C'(0) = V$$

and assume a parameterization such that

$$C(-1) = P_{-1} ;$$

$$C(0) = P_0 ;$$

$$C(1) = P_1 .$$

If

$$C(t) = at^3 + bt^2 + ct + d$$

is the cubic, then

$$(8) \quad \begin{cases} d = P_0 ; \\ c = V ; \\ C(-1) = -a + b + -c + d = P_{-1} ; \\ C(1) = a + b + c + d = P_1 . \end{cases}$$

Letting $q_1 = P_{-1} + (c - d)$ and $q_2 = P_1 - (c + d)$ the last two equations of system (8) can be written as

$$\begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

Note, the inverse of the 2x2 matrix on the left side of the above expression is

$$\begin{bmatrix} -1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$$

Hence,

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

and

$$(9) \quad a = (-q_1 + q_2)/2$$

$$(10) \quad b = (q_1 + q_2)/2.$$

The coefficients a, b, c, d are seen to be easily determined by employing system (8) and equations (9) and (10).

To define a complete curve using these cubic curve segments the steps required are similar to those outlined in section 4.

6. Vector Considerations For The Three Points And Vector Construction

Because we are constructing parametric cubic segments using a tangent vector as part of its definition during the construction technique, caution must be used [Chapter 6,4]. If the vector magnitude is greater than the distance between which the curve segment is being defined, the cubic could show a whip like effect[Fig. 7, 8, and 9].

In our implementation we resolved this dilemma by allowing for the scaling of each vector component
by S / TV, where:
S = a scale factor;
TV = the absolute value of the component of the vector that is largest in magnitude.
A more sophisticated procedure could be implemented by using the factor (S * DI)/TV, where S and TV are as above and
DI = the absolute value of the component that is the largest in magnitude of the vector which is the difference of the two points between which the cubic vector is to be defined.
By using this last procedure and restricting S to values between 0 and 1, the vector magnitude will always be less than the magnitude of the distance between which the curve segment is being defined. (DI could also have been selected as the distance between the two points.)

7. Comparing The Required Number of Operations to Compute A Cubic Curve Segment

A criterion for suitability of a curve type is the number of computations required to compute points along a given curve segment. To determine the points, the cubic vector coefficients are computed to determine the expression

$$C(t) = a\,t^3 + b\,t^2 + c\,t + d\ ;$$

then this expression is used to compute coordinate values along the curve. In each of the procedures discussed, the difference in computation time will be related to the determination of the coefficients. With this in mind one needs only to consider the operations involved in defining the vector coefficients to make a comparison of the different procedures. In comparing the number of operations performed to define vector coefficients for the above three cases, one needs only to consider one vector component. That is, one would solely need to determine the number of operations of the first vector component of each vector coefficient. With this in mind:

(i) One observes that for each cubic component for the Overhauser parabolic blending case there are nine multiplications and seven additions required to compute the coefficients.

(ii) For the three points and a vector case discussed in sectoin 4, the coefficients for one component of a cubic segment are determined by six multiplications; the sum of the additions and subtractions will be eight (these numbers include the computations to compute the vector V). One could also include the multiplication of the scaling mentioned in section 6. This would bring the number of multiplications to seven or eight depending upon how you wished to include the S term being divided by the TV term.

(iii) For the three points and a vector case discussed in sectoin 5, the coefficients for one component of a cubic segment are determined by four multiplications and divisions; the sum of the additions and subtractions will be eight (these numbers include the computations to compute the vector V). As mentioned in (ii) the multiplications of the scaling mentioned in section 6 could also be included bringing the number of multiplications to five or six depending upon how you wished to include the S term being divided by the TV term.

(iv) If one had considered the Akima [4] procedure, one would have found that for each cubic component the sum of the required multiplications and divisions would have been twelve and the sum of the additions and subtractions would have been fourteen.

Continued on Page 174

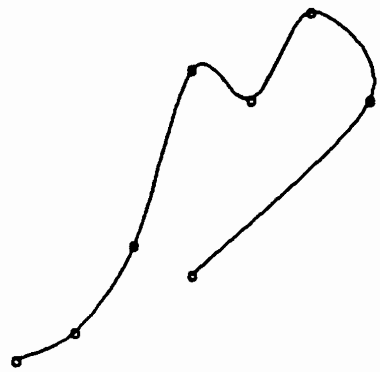

Figure 1 *A curve generated using the parabolic blending procedure.*

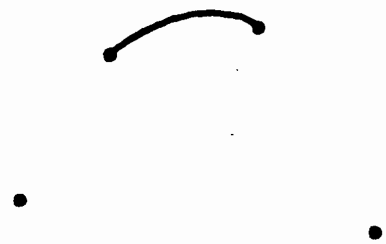

Figure 4 *Four points defining a parabolic blend between the second and third points are being displayed. Also displayed is the cubic segment defined by these points.*

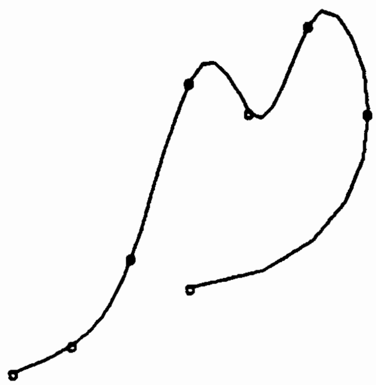

Figure 2 *A curve generated using three points and a vector procedure. The vector defines the tangent value at the first point.*

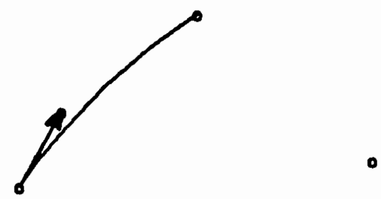

Figure 5 *Three points and a vector defining a cubic curve segment between the first and second points as outlined in Section 4 are being displayed. Also displayed is the cubic segment defined by the points and the vector.*

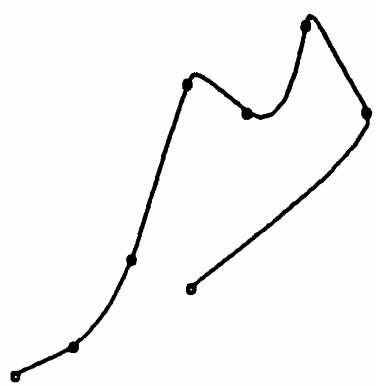

Figure 3 *A curve generated using three points and a vector procedure. The vector defines the tangent value at the second point.*

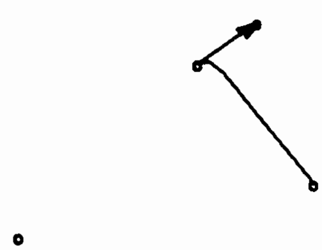

Figure 6 *Three points and a vector defining a cubic curve segment between the second and third points as outlined in Section 5 are being displayed. Also displayed is the cubic segment defined by the points and the vector.*

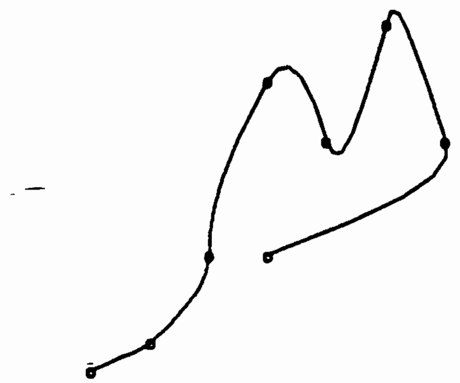

Figure 7 *An example using the procedure of Section 5 and multiplying the components of the defining vector V by S/TV, where S is equal to 15 is displayed.*

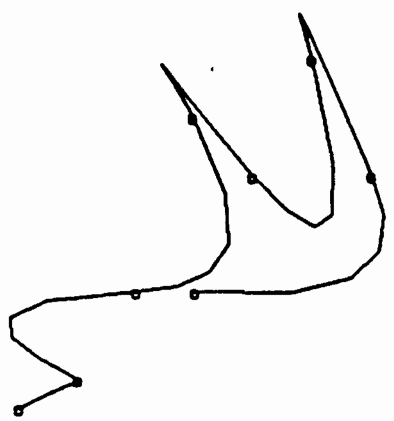

Figure 8 *An example using the procedure of Section 5 and multiplying the components of the defining vector V by S/TV, where S is equal to 3S is displayed. (The same input points as in Figure 7 are used.)*

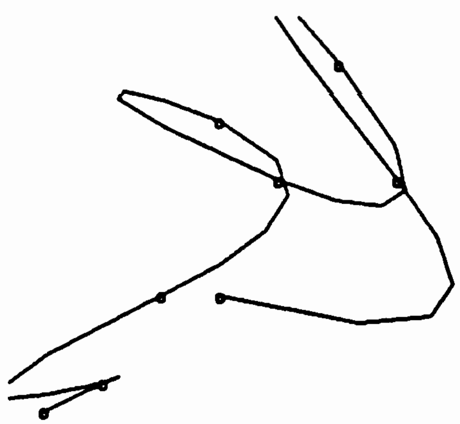

Figure 9 *An example using the procedure of Section 5 and multiplying the components of the defining vector V by S/TV, where S is equal to 50 is displayed. (The same input points as in Figure 7 are used.)*

If speed is the sole criterion for selection, then one of the three points and a vector procedure or something similar to them would be selected. One will note that when using the second three points and a vector procedure outlined above, at times a natural looking curve is not generated because of the lack of control of the slope vector at the point P_2 of section 5.With the other mentioned procedures the last point supplied is used as part of the procedure to define the slope vector value for the preceeding point. The curve generated by the Overhauser parabolic blending case has more aesthetic appeal, but has a slightly greater computational time. The proper choice of a curve type will depend on one's application.

References

1. Overhauser, A., "Analytic Definition of Curves and Surfaces by Parabolic Blending," Technical Report No. SL68-40, Ford Motor Company Scientific Laboratry, May 8, 1968 .
2. deBoor, C., *A Practical Guide to Splines,* Springer-Verlag New York Inc., New York, N. Y. , 1978.
3. Rogers,D., F.,and Adams, J., A., *Mathematical Elements for Computer Graphics* , McGraw-Hill Book Company,New York,1976.
4. Akima, H., " A New Method of Interpolation and Smooth Curve Fitting Based On Local Procedures," Journal of the Association for Computing Machines, Volume 17, Number 4, October 1970, pp 589-602.
5. Mortenson, M. E., *Geometric Modeling,* John Wiley and Sons, Inc., Somerset, New Jersey, 1985.
6. Forrest, A.R., "Curve and Surfaces for Computer Aided Design," University of Cambridge, PhD Thesis, July 1968.
7. Brewer, J.A.,"Three Dimensional Design by Graphical Man-Computer Communication," Purdue University, PhD Thesis, May, 1977.

□