

MA 323 Geometric Modelling

Course Notes: Day 16

Interpolating Bezier Curves & Best Fit Bezier Curves

David L. Finn

Today, we want to talk about some issues in designing with Bezier curves. The issues we want to consider concern the use of control points versus points on the curve. We have said earlier that control points are better to use than points on the curve for stability reasons, but what happens when you are constructing the curve from a sampling method or scanning method. You get points on the model, and want to construct a model from the given points. You are not given the control points, but must construct them. Rather than artistically trying to match the curve by placing control points, it is useful to automatically generate the control points to get an approximate curve and then tweak the control points as desired.

The use of control points to describe the curve in general gives the outliers that form a basic shape. The Bezier curve formed by the control points follows the control polyline generally. However, the exact way that the curve follows the control polyline is not easy to predict intuitively, as you saw from yesterday's interactive exercises. It is as much an art form than a science. After some practice, you may should be able to choose the control points to duplicate an unknown Bezier curve, by using pattern matching skills developed from practice. This is useful for approximating a random curve as you develop the skills to know how to place the control points to get a desired behavior.

16.1 Interpolation for Bezier curves:

Recall the generic interpolation problem is given $n + 1$ points $a_0, a_1, a_2, \dots, a_n$ to find a curve (for us a Bezier curve) that passes through the given points. If we are looking for a Bezier curve, this is relatively easy to do. We need to choose $n + 1$ values of the parameter t for the curve, and then solve a system of $n + 1$ equations to determine the control points.

Given the information $(t_0, a_0), (t_1, a_1), (t_2, a_2), \dots, (t_n, a_n)$ where each a_i is a point the the t_i are numbers satisfying $0 \leq t_0 < t_1 < t_2 < \dots < t_n \leq 1$. Find the Bezier curve $c(t)$ that satisfies $c(t_i) = a_i$.

To find the curve, we want to solve the system of equations

$$a_j = \sum_{i=0}^n \beta_i^n(t_j) p_i$$

for p_i the control points. This can done easily be done as a set of $(n+1) \times (n+1)$ systems of linear equations. There is one system of linear equations for each coordinate function needed to describe the points.

This can be recast as a system of matrix equations as

$$A = BP$$

viewing A as matrix of points to be interpolated, B as the matrix of the control functions and P as the matrix of control points. The sizes of the for planar points are $2 \times (n+1)$ for A , $(n+1) \times (n+1)$ for B and $2 \times (n+1)$ for P . The points are expressed as row vectors, and the B matrix is then

$$\begin{bmatrix} \beta_0^n(t_0) & \beta_1^n(t_0) & \cdots & \beta_n^n(t_0) \\ \beta_0^n(t_1) & \beta_1^n(t_1) & \cdots & \beta_n^n(t_1) \\ \vdots & \vdots & \ddots & \vdots \\ \beta_0^n(t_n) & \beta_1^n(t_n) & \cdots & \beta_n^n(t_n) \end{bmatrix}.$$

This can be solved by linear algebra easily. The advantage of using Bezier curves and the Bernstein polynomials over the monomial basis $[1, t, t^2, \dots, t^n]$ is numerical stability. It can be shown that the matrix of Bernstein polynomials has better numerical stability for inverses. In addition, small errors in the location of the control points do not change the curve as much as small changes in the interpolated points or the coefficients in the monomial basis.

16.2 Best Fit Bezier Curves

The matrix formulation and the use of linear algebra to interpolate also allows us to use the method of least squares to find a Bezier curve that passes near to a set of points. This is another useful method for creating a Bezier curve from a list of data points, and then once the control points are obtained, you can move the control points of the Bezier curve as desired to obtain a better or more useful model. Once the problem has been cast in a system of linear equations, it is just a matter of using least squares instead to solve instead of Gaussian elimination.

16.3 Choosing the Parameter Values for Interpolation and Data Fitting

Recall, the art form in interpolation is in choosing the values of the parameters t_i . The difference between the art form of choosing the control points and the art form of choosing the parameters is that there is some reasonable methods for choosing the values of the parameters.

Uniform spacing: A standard choice is to select $t_i = i/n$. This sets the parameters equally spaced. The advantage in this method is that it is quick and easy. The disadvantage is that it does not take into account anything about the points a_i .

Chord length spacing. An alternative to uniform spacing is to compute the distance between consecutive points on the curve, and determine the values of the parameter with respect to these distance. For instance, let $d_i = \|a_{i+1} - a_i\|$, and $s_0 = 0$ and define $s_i = \sum_{j=1}^i d_j$ where $i = 1, 2, \dots, n$. Then, we set $t_i = s_i/s_n$.

Exercises

1. Given the points $a_0 = [1, 0]$, $a_1 = [2, 1]$, $a_2 = [3, 2]$, $a_3 = [0, 2]$. Find the Bezier curve that interpolates these points using
 - (a) uniform spacing
 - (b) chord length spacing
2. Given the points $a_0 = [1, 0]$, $a_1 = [2, 0]$, $a_2 = [2, 1]$, $a_3 = [3, 3]$, $a_4 = [0, 3]$, $a_5 = [0, 2]$. Find the cubic Bezier curve that best fits these points using
 - (a) uniform spacing
 - (b) chord length spacing
3. How close to a circle is a Bezier curve that interpolates $n + 1$ points uniformly spaced points on a circle? Use $a_i = [\cos(2\pi i/n), \sin(2\pi i/n)]$ and uniform spacing for the parameter. As n increases do you get close to a circle? Where are the control points?
4. Play with the applets for interpolating Bezier curves and best fit Bezier curves.