More Splines, Bezier Splines

CS116B

Chris Pollett

Feb 9, 2005.

Outline

- Cardinal Splines
- Kochanek-Bartels Splines
- Bezier Spline Curves
- Bezier Surfaces

Cardinal Splines

- Similar to Hermite Splines except now we don't explicitly give the value of the derivatives at the endpoints.
- Instead, we calculate it from the coordinates of the two adjacent control points.
- That is:

  $P(0)= p_k$

  $P(1) = p_{k+1}$

  $P'(0) = 1/2(1-t)(p_{k+1} - p_{k-1})$

  $P'(1) = 1/2(1-t)(p_{k+2} - p_k)$
- Here t is called the **tension** and is a fixed constant. When t=0, Cardinal Splines called **Catmull-Rom splines** or **Overhauser splines.**
- Same idea as with Hermite splines can be used to find matrix and generating polynomials.

Kochanek-Bartels Splines

- These are a generalization of Cardinal Splines to allow for discontinuous derivatives across boundaries.
- Conditions this time are:

$$P(0) = p_k$$

$$P(1) = p_{k+1}$$

$$P'(0)_{in} = 1/2(1-t)((1+b)(1-c)(p_{k} - p_{k-1}) + (1-b)(1+c)(p_{k+1} - p_k))$$

$$P'(1)_{out} = 1/2(1-t)((1+b)(1+c)(p_{k+1} - p_k) + (1-b)(1-c)(p_{k+2} - p_{k+1}))$$

- b is called the bias and adjusts the curvature at each end.
- c controls the continuity. If nonzero, curve will be discontinuous.

Bezier Spline Curves

- Another kind of spline curve.
- Bezier curves have several useful properties and are widely implemented in graphics systems.
- They work with arbitrarily many control points but most graphics packages limit the number to four.

Bezier Curve Equations

- Assume have n+1 control points $p_k=(x_k,y_k,z_k)$.
- The Bezier curve is given by:

  $P(u) = \sum^n_{k=0} p_k BEZ_{k,n}(u)$ for u in [0,1]

- Here $Bez_{k,n}(u)=C(n,k)u^k(1-u)^{n-k}$ where C(n,k) is n!/(k!n-k!)
- Since C(n,k) = (n-k+1)/k*C(n,k-1) can show:

  $BEZ_{k,n}(u)=(1-u)BEZ_{k,n-1}(u) + uBEZ_{k-1,n-1}(u)$ where

  $BEZ_{k,k}(u)=u^k$, and $BEZ_{0,k}(u)=(1-u)^{k.}$

Properties of Bezier Curves

- The curve connects with the first and last point. i.e., $\mathbf{P}(0)=\mathbf{p_0}$ and $\mathbf{P}(1)=\mathbf{p_n}$.
- The value of the derivatives at the endpoints can be calculated from the control points.

  $\mathbf{P}'(0) = -n\mathbf{p_0} + n\mathbf{p_1}$

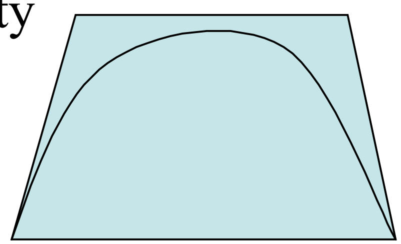
  $\mathbf{P}'(1) = -n\mathbf{p_{n-1}} + n\mathbf{p_n}$

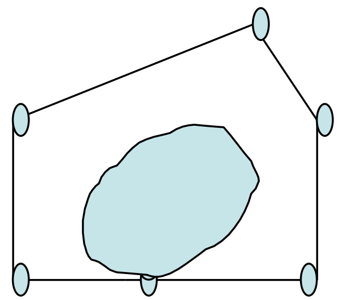
  That is the slope of curve is along the between last two pairs of points.
- The second derivative can be calculated using:

  $\mathbf{P}''(0)=n(n-1)[(\mathbf{p_2}-\mathbf{p_1}) - (\mathbf{p_1}-\mathbf{p_0})]$ and $\mathbf{P}''(1) = n(n-1)[(\mathbf{p_{n-2}}-\mathbf{p_{n-1}}) - (\mathbf{p_{n-1}}-\mathbf{p_n})]$
- Lastly, $\sum^{n}_{k=0} BEZ_{k,n}(u)=1$. So contained with hull of points.

Design Techniques for Bezier Curves

Example of convex hull property



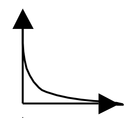


- If make first and last control points the same get a closed curve
- Can generate complicated curves by piecing together several degree four curves and matching at the endpoints.
- Note it is easy to make two adjacent curves C1 and C2 first derivative continous (C^1): one puts $\mathbf{p_{n-}}\ \mathbf{p_{n-1}}$ of C1 and p_1- p_0 of C2 all on the same line.
- Can use our equation for the second derivative to obtain even C^2 continuity.

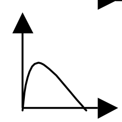
Cubic Bezier Curves

- If have four control point case get a cubic curve.
- The four blending functions are:

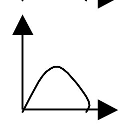
$BEZ_{0,3} = (1\text{-}u)^3$



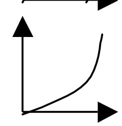
$BEZ_{1,3} = u*(1\text{-}u)^2$



$BEZ_{2,3} = u^2*(1\text{-}u)$



$BEZ_{3,3} = u^3$



Bezier Matrix

$P(u) = [\ u^3\ u^2\ u^1\ 1]M_{Bez}[p_0\ p_1\ p_2\ p_3]$

Where

$$M_{Bez} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & 6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Bezier Surfaces

As with other Spline can generalize Bezier Curves to Bezier Surfaces:

$$P(u,v) = \sum^{m}_{j=0} \sum^{n}_{k=0} p_{j,k} BEZ_{j,m}(v) BEZ_{k,n}(u)$$