Beziers and Spline Curves and Surfaces

Objectives

• Introduce Bezier curves and surfaces
• Derive required matrices
• Introduce B-spline and compare to standard cubic Bezier
Bezier’ s Idea

• In graphics and CAD, do not usually have derivative data
• Bezier suggested using same 4 data points as with cubic interpolating curve to approximate derivatives in Hermite form

Approximating Derivatives

\[ p_1 \text{ located at } u = 1/3 \]
\[ p_2 \text{ located at } u = 2/3 \]
\[ p'(0) \approx \frac{p_1 - p_0}{1/3} \]
\[ p'(1) \approx \frac{p_3 - p_2}{1/3} \]
\[ \text{slope } p'(0) \]
\[ \text{slope } p'(1) \]
Equations

Interpolating conditions same
\[ p(0) = p_0 = c_0 \]
\[ p(1) = p_3 = c_0 + c_1 + c_2 + c_3 \]

Approximating derivative conditions
\[ p'(0) = 3(p_1 - p_0) = c_0 \]
\[ p'(1) = 3(p_3 - p_2) = c_1 + 2c_2 + 3c_3 \]

Solve four linear equations for \( c = M_B p \)

Beziers Matrix

\[
M_B = \begin{bmatrix}
1 & 0 & 0 & 0 \\
-3 & 3 & 0 & 0 \\
3 & -6 & 3 & 0 \\
-1 & 3 & -3 & 1
\end{bmatrix}
\]

\[ p(u) = u^T M_B p = b(u)^T p \]

blending functions
Blending Functions

\[ b(u) = \begin{bmatrix} (1-u)^3 \\ 3u(1-u)^2 \\ 2u^2(1-u) \\ u^3 \end{bmatrix} \]

Note that all zeros are at 0 and 1 \( \Rightarrow \) functions smooth over \((0,1)\)

Bernstein Polynomials

- Blending functions are special case of Bernstein polynomials
  \[ b_{kd}(u) = \frac{d!}{k!(d-k)!} u^k (1-u)^{d-k} \]
- These polynomials give blending polynomials for any degree Bezier form
  - All zeros at 0 and 1
  - For any degree all sum to 1
  - All between 0 and 1 inside \((0,1)\)
Convex Hull Property

- Properties of Bernstein polynomials ensure that all Bezier curves lie in convex hull of their control points
- Even though do not interpolate all data, cannot be too far away

Beziers curve

\[ p_0, p_1, p_2, p_3 \]

Convex hull

Beziers Patches

Using same data array \( P = [p_{ij}] \) as with interpolating form

\[
p(u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} b_i(u) b_j(v) p_{ij} = u^T M_B P M_B^T v
\]

Patch lies in convex hull

\[ P_{00}, P_{03}, P_{30}, P_{33} \]
Analysis

- Although Bezier form much better than interpolating form, derivatives not continuous at join points
- Can do better?
  - Go to higher order Bezier
    - More work
    - Derivative continuity still only approximate
    - Supported by OpenGL
  - Apply different conditions
    - Tricky without letting order increase

B-Splines

- Basis splines: use data at \( p_i = [p_{i-2} \ p_{i-1} \ p_i \ p_{i+1}]^T \) to define curve only between \( p_{i-1} \) and \( p_i \)
- Allow to apply more continuity conditions to each segment
- For cubics, can have continuity of function, 1st and 2nd derivatives at join points
- Cost =
  - 3x for curves
  - 9x for surfaces
Cubic B-spline

\[ p(u) = u^T M_s \mathbf{p} = b(u)^T \mathbf{p} \]

\[
M_s = \begin{bmatrix}
1 & 4 & 1 & 0 \\
-3 & 0 & 3 & 0 \\
3 & -6 & 3 & 0 \\
-1 & 3 & -3 & 1
\end{bmatrix}
\]

Blending Functions

\[ b(u) = \frac{1}{6} \begin{bmatrix}
(1-u)^3 \\
4 - 6u^2 + 3u^3 \\
1 + 3u + 3u^2 - 3u^2 \\
u^3
\end{bmatrix} \]

convex hull property
B-Spline Patches

\[ p(u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} b_i(u) b_j(v) p_{ij} = u^T M_S P M_S^T v \]

defined over only 1/9 of region

Splines and Basis

• If examine cubic B-spline from perspective of each control (data) point, each interior point contributes (through blending functions) to four segments
• can rewrite \( p(u) \) in terms of data points as

\[ p(u) = \sum B_i(u) p_i \]

defining basis functions \( \{B_i(u)\} \)
Basis Functions

In terms of the blending polynomials

\[ B_i(u) = \begin{cases} 
0 & u < i - 2 \\
 b_0(u + 2) & i - 2 \leq u < i - 1 \\
 b_1(u + 1) & i - 1 \leq u < i \\
 b_2(u) & i \leq u < i + 1 \\
 b_3(u - 1) & i + 1 \leq u < i + 2 \\
0 & u \geq i + 2 
\end{cases} \]

Generalizing Splines

• Can extend to splines of any degree
• Data and conditions to not have to be given at equally spaced values (knots)
  - Nonuniform and uniform splines
  - Can have repeated knots
    - Can force spline to interpolate points
• Cox-deBoor recursion gives method of evaluation
NURBS

- Nonuniform Rational B-Spline curves and surfaces add 4th variable \( w \) to \( x,y,z \)
  - Can interpret as weight to give more importance to some control data
  - Can also interpret as moving to homogeneous coordinate
- Requires perspective division
  - NURBS act correctly for perspective viewing
- Quadrics = special case of NURBS