CS559 – Lecture 23
Approximating Curves

These are course notes (not used as slides)
Written by Mike Gleicher, Oct. 2005
Updates Oct 2007

Note: in lecture, I went in a bit of a different order

Approximating Curves

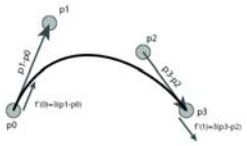
- Interpolation isn't the only way to describe a curve
- Give points that "influence" a curve
- Why?
  - Better control of what happens in between points
- 2 important cases for computer graphics
  - Bezier
  - B-Spline

Bezier Segments

- Curve is made of many segments
  - Nomenclature issue
- Each segment is a polynomial
  - Of any degree
  - 3 is most common in computer graphics

Bezier Segments

- A segment of degree d has (d+1) control points
- A segment interpolates its first and last controls
  - With u= 0, 1 respectively
- The first derivative at the beginning (end) is proportional to the vector between the first 2 (last 2) points – scaled by the degree of the curve



Bezier Segments (2)

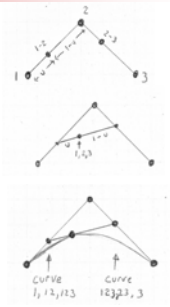
- The nth derivative depends on the first (or last) n points
- Cubics are similar to Hermites
  - All points in space (not derivative amounts)
  - Scaling factors
- Pieces connected by placing points correctly
  - C(0) by matching endpoints
  - C(1) by aligning end vectors
  - G(1) by end-vectors being co-linear

Properties of Bezier Curves

- Simple mathematical form for basis functions
- Good algorithms for computation
  - Subdivision procedure
  - De Casteljau algorithm
  - Divide and conquer because…
- Convex Hull Properties
- Variation Diminishing
- Symmetric
- Affine invariant
  - NOT perspective invariant

De Casteljau Algorithm

- Evaluate curve at u
  - Divide line segments
  - U of the way
- Can use to subdivide curves
- Repeated linear interpolation for ANY degree!



Decastlejau to Bernstein

- Apply geometric construction to derive equations
- Different groups came at this differently
- Algebraic vs. Subdivision

$$p_{12} = (1-u)P_1 + uP_2 \qquad p_{23} = (1-u)P_2 + uP_3$$

$$p_{123} = (1-u)P_{12} + uP_{23}$$

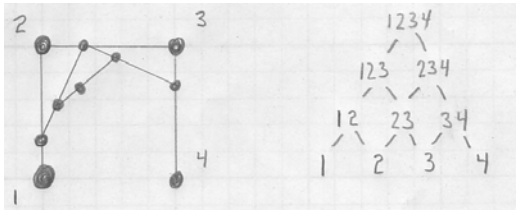
$$f(u) = (1-u)\left((1-u)P_1 + uP_2\right) + u\left((1-u)P_2 + uP_3\right)$$

$$= \underbrace{(1-u)^2}_{b_1} P_1 + \underbrace{2u(1-u)}_{b_2} P_2 + \underbrace{u^2}_{b_3} P_3$$

Derivatives of the Bezier Curve

- F(u)=(1-u)^2 P1 + 2u (1-u) P2 + u^2 P3
- F'(u) = (-2-2u) P1 + (2-4u) P2 + 2u P3
- F'(0) = 2 (P2-P1)
- F'(1) = 2 (P3-P2)

Higher Order Beziers



- Successive Linear interpolation (to find point)
  - Pyramid
- Use points to divide curve

General Bezier Blending Functions

$$b_{k,n}(u) = C(n,k)\ u^k\ (1-u)^{(n-k)}$$

$$C(n,k) = \frac{n!}{k!\ (n-k)!}.$$

- Blending functions = $b_i(u)\ p_i$
- Need to index on n as well (number of basis functions)
- Bernstein Basis Polynomials