Acknowledgments

- Sources: Figures from Angel and Shreiner 6th edition, unless otherwise noted
- Some slides courtesy of V. Zordan
Hermite curves
Hermite Curves

Interpolate endpoints and match derivatives

this gives $C^1$ continuity
Hermite Curves Basis

\[ p(u) = h_{00}(u)p(0) + h_{01}(u)p(1) + h_{10}(u)p'(0) + h_{11}(u)p'(1) \]

\[ h_{00}(u) = 2u^3 - 3u^2 + 1 \]
\[ h_{01}(u) = -2u^3 + 3u^2 \]
\[ h_{10}(u) = u^3 - 2u^2 + u \]
\[ h_{11}(u) = u^3 - u^2 \]

Wikimedia Commons
Geometric Continuity
Bezier curves
Bezier blending functions
Bernstein Polynomials

• The blending functions are a special case of the Bernstein polynomials

\[ b_{kd}(u) = \frac{d!}{k!(d-k)!} u^k (1-u)^{d-k} \]

• These polynomials give the blending polynomials for any degree Bezier form
  - All zeros at 0 and 1
  - For any degree they all sum to 1
  - They are all between 0 and 1 inside (0,1)
Bezier Curves
Beziers Curves

- curve lies in the convex hull of the data
Bezierser Patches

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

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

Patch lies in convex hull
Bezier surface patches
Cubic B-Splines
Cubic B-Splines
Spline blending functions

\[ b_0(u) = \frac{1}{6} (1 - u)^3 \]

\[ b_1(u) = \frac{1}{6} (4 - 6u^2 + 3u^3) \]

\[ b_2(u) = \frac{1}{6} (1 + 3u + 3u^2 - 3u^3) \]

\[ b_3(u) = \frac{1}{6} u^3 \]
Spline properties

Basis functions

convexity
General Splines

- Defined recursively by *Cox-de Boor recursion formula*

\[
b_{j,0}(t) = \begin{cases} 
1 & \text{if } t_j \leq t \\
0 & \text{otherwise}
\end{cases}
\]

\[
b_{j,n}(t) := \frac{t - t_j}{t_{j+n} - t_j} b_{j,n-1}(t) + \frac{t_{j+n+1} - t}{t_{j+n+1} - t_{j+1}} b_{j+1,n-1}(t)
\]
Curve and Surface Rendering
Sampling

- Sample the curve and render resulting flat polygons

- Evaluate the curve at several sample points (n mults)
  \[ p(u) = c_0 + u(c_1 + u(c_2 + u(\ldots + c_n u))) \]

- Use a divided difference table (O(n) additions, no mults)
Recursive Subdivision
Recursive Subdivision

- work with convex hull, does not require evaluating the polynomial
- Bezier curves most convenient -- other curves can be transformed to Bezier
- same approach for surfaces
Project ideas
Rendering

Ambient occlusion rendering

NPR

high dynamic range rendering

Other possibilities: motion blur, water rendering, atmospheric scattering, advanced shadow techniques - shadow volumes and soft shadows, volumetric effects, displacement mapping, volume rendering in OpenGL, other ideas from book Real-Time Rendering, Shader in GLSL
Modeling

• procedural content generation
• implicit surface tessellation
• subdivision surfaces
• mesh generation
• mesh cutting
• interactive mesh deformation
• terrain modeling

Parish, Mueller

Molino et al, 2003
Character Animation

- rigging and skinning
- keyframe animation
- control algorithms

Treuille, A. Lee, Y. Popović, Z.

Macchietto et al. 2009

Popovic
Simulation

- basic water simulator
- rigid body dynamics - integration, collisions
- cloth simulation
- fracture simulation
- particle system
- physically-based sound

Losasso et al., 2008

Guendelman et al., 2003

Selle et al., 2009
Project proposal

• Pre-proposal: Monday, Feb. 20. A short paragraph with the main idea. I will give feedback and any changes should be agreed by Wed.

• Proposal due Monday, Feb. 27

• 2-3 pages

• list group (1-2 people)

• background on the problem

• concrete project goals