Curves & Surfaces

Introductions – Who are you?

• name
• year/degree
• graphics background (if any)
• research/job interests, future plans
• something fun, interesting, or unusual about yourself
• your favorite thing about programming
Last Time?

- Adjacency Data Structures
  - Geometric & topologic information
  - Dynamic allocation
  - Efficiency of access
- Mesh Simplification
  - edge collapse/vertex split
  - geomorphs
  - progressive transmission
  - view-dependent refinement

Progressive Meshes

- Mesh Simplification
  - vertex split / edge collapse
  - geometry & discrete/scalar attributes
  - priority queue
- Level of Detail
  - geomorphs
- Progressive Transmission
- Mesh Compression
- Selective Refinement
  - view dependent
Selective Refinement

Figure 10: Selective refinement of a terrain mesh taking into account view frustum, silhouette regions, and projected screen size of faces (7,438 faces).

Preserving Discontinuity Curves

Figure 12: Approximations of a mesh $\mathcal{M}$ using (b–c) the PM representation, and (d–f) the MRA scheme of Eck et al. [7]. As demonstrated, MRA cannot recover $\mathcal{M}$ exactly, cannot deal effectively with surface crossings, and produces approximating meshes of inferior quality.
Other Simplification Strategies

• Remove a vertex & surrounding triangles, re-triangulate the hole

• Merge Nearby Vertices
  – will likely change the topology…

Other Simplification Strategies


When to Preserve Topology?

Figure 3: On the left is a regular grid of 100 closely spaced cubes. In the middle, an approximation built using only edge contractions demonstrates unacceptable fragmentation. On the right, the result of using more general pair contractions to achieve aggregation is an approximation much closer to the original.

Quadric Error Simplification

- Contract (merge) vertices $v_i$ and $v_j$ if:
  - $(v_i, v_j)$ is an edge, or
  - $\|v_i - v_j\| < t$, where $t$ is a threshold parameter

- Track cumulative error by summing 4x4 quadric error matrices after each operation:
  $$\Delta(v) = \sum_{p:\text{plane}(v)} (v^p)(p^v) = \sum_{p:\text{plane}(v)} \vec{v}(pp^v) = v^2 \left( \sum_{p:\text{plane}(v)} K_p \right)$$
  Garland & Heckbert, "Surface Simplification Using Quadric Error Metrics" SIGGRAPH 1997

Judging Element Quality

- How “equilateral” are the elements?
  - For Triangles?
    - Ratio of shortest to longest edge
    - Ratio of area to perimeter$^2$
    - Smallest angle
    - Ratio of area to area of smallest circumscribed circle
  - For Tetrahedra?
    - Ratio of volume$^2$ to surface area$^3$
    - Smallest solid angle
    - Ratio of volume to volume of smallest circumscribed sphere
Reading for Today

- "Teddy: A Sketching Interface for 3D Freeform Design", Igarashi et al., SIGGRAPH 1999
- Post a comment or question on the LMS discussion by 10am on Tuesday

Today

- Limitations of Polygonal Models
  - Interpolating Color & Normals in OpenGL
  - Some Modeling Tools & Definitions
- What's a Spline?
  - Interpolation Curves vs. Approximation Curves
  - Linear Interpolation
- Bézier Spline
- BSpline (NURBS)
- Extending to Surfaces – Tensor Product
Today

- Limitations of Polygonal Models
  - Interpolating Color & Normals in OpenGL
  - Some Modeling Tools & Definitions
- What's a Spline?
  - Interpolation Curves vs. Approximation Curves
  - Linear Interpolation
- Bézier Spline
- BSpline (NURBS)
- Extending to Surfaces – Tensor Product

Limitations of Polygonal Meshes

- Planar facets (& silhouettes)
- Fixed resolution
- Deformation is difficult
- No natural parameterization (for texture mapping)
It’s easy in OpenGL to specify different colors and/or normals at the vertices of triangles:

Why is this useful?

Color & Normal Interpolation

What is Gouraud Shading?

Instead of shading with the normal of the triangle, we’ll shade the vertices with the average normal and interpolate the shaded color across each face.

This gives the illusion of a smooth surface with smoothly varying normals.

How do we compute Average Normals? Is it expensive??
Phong Normal Interpolation \textit{(Not Phong Shading)}

- \textit{Interpolate the average vertex normals} across the face and compute \textit{per-pixel shading}
  - Normals should be re-normalized (ensure length=1)

- Before shaders, per-pixel shading was not possible in hardware (Gouraud shading is actually a decent substitute!)

Gouraud Shading/Phong Normal Interpolation

- Not always good enough
  - Still low, fixed resolution (missing fine details)
  - Still have polygonal silhouettes
  - Intersection depth is planar (e.g. ray tracing visualization)
  - Collisions problems for simulation
  - Solid Texturing problems
  - ...
Some Non-Polygonal Modeling Tools

- Extrusion
- Surface of Revolution
- Spline Surfaces/Patches
- Quadrics and other implicit polynomials

Continuity definitions:

- \(C^0\) continuous
  - curve/surface has no breaks/gaps/holes
- \(G^1\) continuous
  - tangent at joint has same direction
- \(C^1\) continuous
  - curve/surface derivative is continuous
  - tangent at joint has same direction and magnitude
- \(C^n\) continuous
  - curve/surface through \(n^{th}\) derivative is continuous
  - important for shading

“Shape Optimization Using Reflection Lines”, Tosun et al., 2007
Today

• Limitations of Polygonal Models
  – Interpolating Color & Normals in OpenGL
  – Some Modeling Tools & Definitions

• What's a Spline?
  – Interpolation Curves vs. Approximation Curves
  – Linear Interpolation

• Bézier Spline
• BSpline (NURBS)
• Extending to Surfaces – Tensor Product
Definition: What's a Spline?

• Smooth curve defined by some control points
• Moving the control points changes the curve

Interpolation  Bézier (approximation)

BSpline (approximation)

Interpolation Curves / Splines

The ducks and spline are used to make tighter curves

www.abm.org
Interpolation Curves

- Curve is constrained to pass through all control points
- Given points $P_0$, $P_1$, ... $P_n$, find lowest degree polynomial which passes through the points

\[
x(t) = a_{n-1}t^{n-1} + \ldots + a_2t^2 + a_1t + a_0 \\
y(t) = b_{n-1}t^{n-1} + \ldots + b_2t^2 + b_1t + b_0
\]

Linear Interpolation

- Simplest "curve" between two points

\[
Q(t) = \begin{pmatrix} Q_x(t) \\ Q_y(t) \\ Q_z(t) \end{pmatrix} = \begin{pmatrix} P_0 \\ P_1 \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} t \\ 1 \end{pmatrix}
\]

$Q(t) = GBT(t) = \text{Geometry } G \cdot \text{Spline Basis } B \cdot \text{Power Basis } T(t)$
Interpolation vs. Approximation Curves

Interpolation curve must pass through control points
Approximation curve is influenced by control points

- Interpolation Curve – over constrained → lots of (undesirable?) oscillations
- Approximation Curve – more reasonable?
Questions?

Today

• Limitations of Polygonal Models
  – Interpolating Color & Normals in OpenGL
  – Some Modeling Tools & Definitions
• What's a Spline?
  – Interpolation Curves vs. Approximation Curves
  – Linear Interpolation
• Bézier Spline
• BSpline (NURBS)
• Extending to Surfaces – Tensor Product
Cubic Bézier Curve

- 4 control points
- Curve passes through first & last control point
- Curve is tangent at $P_1$ to $(P_2 - P_1)$ and at $P_4$ to $(P_4 - P_3)$

A Bézier curve is bounded by the convex hull of its control points.

Cubic Bézier Curve

- de Casteljau's algorithm for constructing Bézier curves
Cubic Bézier Curve

\[ Q(t) = (1 - t)^3 P_1 + 3t(1 - t)^2 P_2 + 3t^2(1 - t) P_3 + t^3 P_4 \]

Bernstein Polynomials

\[ B_{Bezier} = \begin{pmatrix}
-1 & 3 & -3 & 1 \\
3 & -6 & 3 & 0 \\
-3 & 3 & 0 & 0 \\
1 & 0 & 0 & 0 \\
\end{pmatrix} \]

Connecting Cubic Bézier Curves

Asymmetric: Curve goes through some control points but misses others

- How can we guarantee \( C^0 \) continuity?
- How can we guarantee \( G^1 \) continuity?
- How can we guarantee \( C^1 \) continuity?
- Can’t guarantee higher \( C^2 \) or higher continuity
Connecting Cubic Bézier Curves

- Where is this curve
  - \( C^0 \) continuous?
  - \( G^1 \) continuous?
  - \( C^1 \) continuous?

- What’s the relationship between:
  - the # of control points, and
  - the # of cubic Bézier subcurves?

Higher-Order Bézier Curves

- > 4 control points
- Bernstein Polynomials as the basis functions

\[
B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, \quad 0 \leq i \leq n
\]

- Every control point affects the entire curve
  - Not simply a local effect
  - More difficult to control for modeling
Questions?

Today

• Limitations of Polygonal Models
  – Interpolating Color & Normals in OpenGL
  – Some Modeling Tools & Definitions
• What's a Spline?
  – Interpolation Curves vs. Approximation Curves
  – Linear Interpolation
• Bézier Spline
• BSpline (NURBS)
• Extending to Surfaces – Tensor Product
Cubic BSplines

- $\geq 4$ control points
- Locally cubic
- Curve is not constrained to pass through any control points

A BSpline curve is also bounded by the convex hull of its control points.

Cubic BSplines

- Iterative method for constructing BSplines

Shirley, Fundamentals of Computer Graphics
Cubic BSplines

\[ Q(t) = \frac{(1-t)^3}{6} P_{i-3} + \frac{3t^3 - 6t^2 + 4}{6} P_{i-2} + \frac{-3t^3 + 3t^2 + t}{6} P_{i-1} + \frac{t^3}{6} P_i \]

\[ Q(t) = GBT(t) \quad B_{B-Spline} = \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \]

Connecting Cubic BSpline Curves

- Can be chained together
- Better control locally (windowing)
Connecting Cubic BSpline Curves

- What’s the relationship between
  - the # of control points, and
  - the # of cubic BSpline subcurves?

BSpline Curve Control Points

- Default BSpline
- BSpline with Discontinuity
- BSpline which passes through end points

Repeat interior control point
Repeat end points
Bézier is not the same as BSpline

- Relationship to the control points is different
Converting between Bézier & BSpline

• Using the basis functions:

\[
B_{\text{Bez}} = \begin{pmatrix}
-1 & 3 & -3 & 1 \\
3 & -6 & 3 & 0 \\
-3 & 3 & 0 & 0 \\
1 & 0 & 0 & 0
\end{pmatrix}
\]

\[
B_{\text{Spline}} = \frac{1}{6} \begin{pmatrix}
-1 & 3 & -3 & 1 \\
3 & -6 & 0 & 4 \\
-3 & 3 & 3 & 1 \\
1 & 0 & 0 & 0
\end{pmatrix}
\]

\[Q(t) = GBT(t) = \text{Geometry } G \cdot \text{Spline Basis } B \cdot \text{Power Basis } T(t)\]
NURBS (generalized BSplines)

- BSpline: uniform cubic BSpline

- NURBS: Non-Uniform Rational BSpline
  - non-uniform = different spacing between the blending functions, a.k.a. knots
  - rational = ratio of polynomials (instead of cubic)

Neat Bezier Spline Trick

- A Bezier curve with 4 control points:
  - $P_0$ $P_1$ $P_2$ $P_3$

- Can be split into 2 new Bezier curves:
  - $P_0$ $P'_1$ $P'_2$ $P'_3$
  - $P'_3$ $P'_4$ $P'_5$ $P_3$

A Bézier curve is bounded by the convex hull of its control points.
Today

- Limitations of Polygonal Models
  - Interpolating Color & Normals in OpenGL
  - Some Modeling Tools & Definitions
- What's a Spline?
  - Interpolation Curves vs. Approximation Curves
  - Linear Interpolation
- Bézier Spline
- BSpline (NURBS)
- Extending to Surfaces – Tensor Product

Spline Surface via Tensor Product

- Of two vectors:
  \[
  \begin{bmatrix}
  a_1 & a_2 & a_3 \\
  a_2 & a_3 \\
  \end{bmatrix} \otimes 
  \begin{bmatrix}
  b_1 & b_2 & b_3 & b_4 \\
  b_2 & b_3 & b_4 \\
  \end{bmatrix} = 
  \begin{bmatrix}
  a_1 b_1 & a_2 b_1 & a_3 b_1 \\
  a_1 b_2 & a_2 b_2 & a_3 b_2 \\
  a_1 b_3 & a_2 b_3 & a_3 b_3 \\
  a_1 b_4 & a_2 b_4 & a_3 b_4 \\
  \end{bmatrix}
  \]

- Similarly, we can define a surface as the tensor product of two curves....

Farin, Curves and Surfaces for Computer Aided Geometric Design
Bilinear Patch

Bi-lerp a (typically non-planar) quadrilateral

\[
L(P_1, P_2, \alpha) \equiv (1 - \alpha)P_1 + \alpha P_2
\]

\[
Q(s, t) = L(L(P_1, P_2, t), L(P_3, P_4, t), s)
\]

Bilinear Patch

- Smooth version of quadrilateral with non-planar vertices...

- But will this help us model smooth surfaces?
- Do we have control of the derivative at the edges?
Ruled Surfaces in Art & Architecture

Chiras Julia
Astri Isabella
Matiss Shteinerts

Antoni Gaudi
Children’s School
Barcelona

Bicubic Bezier Patch

Notation: \( \mathbf{CB}(P_1, P_2, P_3, P_4, \alpha) \) is Bézier curve with control points \( P_i \) evaluated at \( \alpha \)

Define “Tensor-product” Bézier surface

\[
Q(s, t) = \mathbf{CB}( C\mathbf{B}(P_{00}, P_{01}, P_{02}, P_{03}, t), \\
C\mathbf{B}(P_{10}, P_{11}, P_{12}, P_{13}, t), \\
C\mathbf{B}(P_{20}, P_{21}, P_{22}, P_{23}, t), \\
C\mathbf{B}(P_{30}, P_{31}, P_{32}, P_{33}, t), \\
s)
\]
Editing Bicubic Bezier Patches

Curve Basis Functions

Surface Basis Functions

Bicubic Bezier Patch Tessellation

- Given 16 control points and a tessellation resolution, we can create a triangle mesh

resolution: 5x5 vertices
resolution: 11x11 vertices
resolution: 41x41 vertices
Modeling with Bicubic Bezier Patches

• Original Teapot specified with Bezier Patches

• But it’s not "watertight": it has intersecting surfaces at spout & handle, no bottom, a hole at the spout tip, a gap between lid & base

Trimming Curves for Patches

Shirley, Fundamentals of Computer Graphics
Spline-Based Modeling Headaches

irregular sampling

“pinched” surfaces

seams & holes

Questions?

• Bezier Patches?

or

• Triangle Mesh?

Henrik Wann Jensen
Readings for Friday *(pick one)*

- Hoppe et al., “Piecewise Smooth Surface Reconstruction”
  SIGGRAPH 1994

- DeRose, Kass, & Truong, "Subdivision Surfaces in Character Animation",
  SIGGRAPH 1998

*Post a comment or question on the LMS discussion by 10am on Tuesday*

---

Homework 1:

- Questions/Comments?