Spline Curves

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

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

Color Interpolation
- Interpolate colors of the 3 vertices
- Linear interpolation, barycentric coordinates

glShadeModel (GL_SMOOTH);
- From OpenGL Reference Manual:
  - Smooth shading, the default, causes the computed colors of vertices to be interpolated as the primitive is rasterized, typically assigning different colors to each resulting pixel fragment.
  - Flat shading selects the computed color of just one vertex and assigns it to all the pixel fragments generated by rasterizing a single primitive.
  - In either case, the computed color of a vertex is the result of lighting if lighting is enabled, or it is the current color at the time the vertex was specified if lighting is disabled.

Normal Interpolation

glBegin(GL_TRIANGLES);
glNormal3f(...);
glVertex3f(...);
glNormal3f(...);
glVertex3f(...);
glNormal3f(...);
glVertex3f(...);
glEnd();
### 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.

- How do we compute Average Normals? Is it expensive??

### Today

- Interpolating Color & Normals in OpenGL
- Limitations of Polygonal Models
- Some Modeling Tools & Definitions
- What’s a Spline?
- Linear Interpolation
- Interpolation Curves vs. Approximation Curves
- Bézier Spline
- BSpline (NURBS)

### Limitations of Polygonal Meshes

- Planar facets (& silhouettes)
- Fixed resolution
- Deformation is difficult
- No natural parameterization (for texture mapping)

### Gouraud 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
- \(C^2\) 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
Questions?

Today

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

Definition: What's a Spline?

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

Interpolation Curves / Splines

Interpolation Curves

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

Interpolation formula:

$$x(t) = a_{n-1}t^{n-1} + \ldots + a_1t + a_0$$
$$y(t) = b_{n-1}t^{n-1} + \ldots + b_1t + b_0$$

Linear Interpolation

• Simplest "curve" between two points

Linear interpolation formula:

$$Q(t) = (1 - t)P_0 + tP_1$$

Spline Basis Functions

$$Q(t) = \begin{pmatrix} Q_1(t) \\ Q_2(t) \\ Q_3(t) \end{pmatrix} = \begin{pmatrix} (P_0) & (P_1) \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} t \\ 1 \end{pmatrix}$$

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

Interpolation curve must pass through control points

Approximation curve is influenced by control points

Interpolation vs. Approximation Curves

Interpolation Curve – over constrained → lots of (undesirable?) oscillations

Approximation Curve – more reasonable?

Questions?

Today

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

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

\[ Q(t) = (1-t)^3P_1 + 3(1-t)^2tP_2 + 3(1-t)t^2P_3 + t^3P_4 \]

Bernstein Polynomials

\[ B_i(t) = \binom{n}{i} (1-t)^{n-i} t^i \]

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

- Interpolating Color & Normals in OpenGL
- Limitations of Polygonal Models
- Some Modeling Tools & Definitions
- What's a Spline?
- Linear Interpolation
- Interpolation Curves vs. Approximation Curves
- Bézier Spline
- BSpline (NURBS)
**Cubic BSplines**

- ≥ 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.

**Connecting Cubic BSpline Curves**

- Can be chained together
- Better control locally (windowing)

**BSpline Curve Control Points**

- What's the relationship between
  - the number of control points, and
  - the number of cubic BSpline subcurves?

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

Bézier

BSpline

Converting between Bézier & BSpline

• Using the basis functions:

\[ \mathbf{B}_{\text{Bézier}} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \]

\[ \mathbf{B}_{\text{BSpline}} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \]

\[ \mathbf{Q}(t) = \mathbf{G} \cdot \mathbf{B}_{\text{BSpline}}(t) = \mathbf{G} \cdot \mathbf{B}_{\text{Bézier}} \cdot \mathbf{T}(t) \]

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:

\[ P_0, P'_1, P'_2, P'_3 \]

\[ P'_3, P'_4, P'_5, P_6 \]

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

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)
Questions?

Readings for Today *(pick one)*
- "Free-form deformation of solid geometric models", Sederberg & Parry, SIGGRAPH 1986
- "Teddy: A Sketching Interface for 3D Freefrom Design", Igarashi et al., SIGGRAPH 1999

Readings for Friday (1/25) *(pick one)*
- DeRose, Kass, & Truong, "Subdivision Surfaces in Character Animation", SIGGRAPH 1998
- Post a comment or question on the LMS discussion by 10am on Friday