CS559: Computer Graphics

Lecture 25: Shape Modeling and Blending

Li Zhang

Spring 2010
Quadtree Idea
Bilinear Bezier Patch

- Define a surface that passes through a, b, c, d?

\[ e = (1 - u)a + ub, \]
\[ f = (1 - u)c + ud. \]

\[ p(u, v) = (1 - v)e + vf \]
\[ = (1 - u)(1 - v)a + u(1 - v)b + (1 - u)vc + uvd. \]
Biquadratic Bezier Patch

- Define a surface that passes a 3x3 control lattice.

\[
p(u,0) = (1-u)^2 p_{00} + 2(1-u)u p_{10} + u^2 p_{20}
\]

\[
p(u,1) = (1-u)^2 p_{01} + 2(1-u)u p_{11} + u^2 p_{21}
\]

\[
p(u,2) = (1-u)^2 p_{02} + 2(1-u)u p_{12} + u^2 p_{22}
\]

\[
p(u,v) = (1-v)^2 p(u,0) + 2(1-v)v p(u,1) + v^2 p(u,2)
\]
Different degree in different directions

$p(u, v)$
Subdivision Curves: Approximating

Initial (Control) Curve: \( P_0 = \{p_0^0, \ldots, p_{n-1}^0\} \),

For each iteration \( k+1 \), add two vertices between: \( p_i^k \) and \( p_{i+1}^k \)

\[
\begin{align*}
p_{2i}^{k+1} &= \frac{3}{4} p_i^k + \frac{1}{4} p_{i+1}^k, \\
p_{2i+1}^{k+1} &= \frac{1}{4} p_i^k + \frac{3}{4} p_{i+1}^k.
\end{align*}
\]

Approximating: Limit curve is very smooth (C2), but does not pass through control points
Subdivision Curves: Interpolating

Initial (Control) Curve: \( P_0 = \{ p_0^0, \ldots, p_{n-1}^0 \} \),

For each iteration \( k+1 \), add two vertices between: \( p_i^k \) and \( p_{i+1}^k \)

\[
\begin{align*}
p_{2i}^{k+1} &= p_i^k, \\
p_{2i+1}^{k+1} &= \left( \frac{1}{2} + w \right) (p_i^k + p_{i+1}^k) - w (p_{i-1}^k + p_{i+2}^k).
\end{align*}
\]

**Interpolating**: for \( 0 < w < 1/8 \), limit curve is C1, and passes through control points
Subdivision Surfaces

Extend subdivision idea from curves to surfaces

RTR, 3e, figure 13.32
Loop Subdivision

\[ p^{k+1} = (1 - \eta \beta) p^k + \beta (p_0^k + \cdots + p_{n-1}^k), \]

\[ p_i^{k+1} = \frac{3p_i^k + 3p_i^{k+1} + p_{i-1}^k + p_{i+1}^k}{8}, \quad i = 0 \ldots n - 1. \]

\[ \beta(n) = \frac{3}{n(n+2)}. \]

\[ \beta(n) = \frac{1}{n} \left( \frac{5}{8} - \frac{(3 + 2 \cos(2\pi/n))^2}{64} \right). \]
Sqrt(3) subdivision

$\beta(n) = \frac{4 - 2 \cos(2\pi/n)}{9n}$
Basic Steps of Subdivision Surfaces

Loop Subdivision

Sqrt(3) Subdivision
Catmull-Clark Subdivision

- Work for arbitrary polygons, not limited to triangles
- Used by Pixar in Geri’s game, toy story2 and all subsequent features
Catmull-Clark Subdivision

Regular vertices: valence = 4

After first insertion, we only have quads in the mesh
Catmull-Clark Subdivision

For each face, add a new vertex at its centroid

For each edge, add an new vertex

\[ e_{j}^{k+1} = \frac{v_{j}^{k} + e_{j}^{k} + f_{j-1}^{k+1} + f_{j}^{k+1}}{4} \]

For each old vertex, update

\[ v_{j}^{k+1} = \frac{n-2}{n} v_{j}^{k} + \frac{1}{n^2} \sum_{j=0}^{n-1} e_{j}^{k} + \frac{1}{n^2} \sum_{j=0}^{n-1} f_{j}^{k+1} \]
Example

Academy Award for Technical Achievement in 2006.
Standard subdivision is not enough
Standard subdivision
Crease: a smooth curve on the surface, where the continuity across the curve is $C^0$.

A corner is a vertex where three or more creases come together

A dart is a vertex where a crease ends and smoothly blends into the surface.
Semisharpness

Derose et al, Subdivision Surfaces in Character Animation, SIGGRAPH 1998
Semisharpness

Derose et al, Subdivision Surfaces in Character Animation, SIGGRAPH 1998
Piecewise smooth subdivision
Geri’s game

- http://www.youtube.com/watch?v=QC-KHaSh0rl
Procedural Shape Modeling

Simple procedure
Procedural Terrain Modeling

- Has a gross structure
- Also with some randomness
- Want a height map $z = h(x,y)$

F.K. Musgrave
1D case

Start with a single horizontal line segment. 
Repeat for a sufficiently large number of times
{
  Find the midpoint of the line segment.
  Displace the midpoint in Y by a random amount.
  Recursively apply this operation for the resulting two segments
    with reduced range for the random numbers (by a factor 0<f<1).
}
1D case

Start with a single horizontal line segment. Repeat for a sufficiently large number of times
{  
  Find the midpoint of the line segment.
  Displace the midpoint in Y by a random amount.
  Recursively apply this operation for the resulting two segments with reduced range for the random numbers (by a factor 0<f<1).
}
2D case

- Subdivide and Displace
2D case

- Subdivide and displace
  - Seed corners with values
  - Perturb midpoint randomly
  - Recurse using a smaller window
  - In 2D, best to use “diamond-square” recursion
2D case

F.K. Musgrave
Texture mapping

F.K. Musgrave
Adding water

• Use an elevation threshold ($z < z_{\text{water}}$)
Fractal Plants (L-Systems)

• Uses “production rules” applied to a seed “axiom”

• Example:
  Axiom: B
  Rule: B->ABB
L-system Example

- **Axiom**: B
- **Replacement Rule**: B → B Y B A
- **4th Iteration**:

![Diagram showing the growth of a tree structure through iterations based on the L-system rules.](image)
L-systems example: Koch snowflake

How to make the flake look less regular?
L-systems example: Serpinski Triangle
Procedural Trees and Bushes

• Define a branch structure
• Define a leaf
Algorithmic Plants

• excellent web resource with free online book: http://algorithmicbotany.org/

• Numerous papers by Przemyslaw Prusinkiewicz and colleagues
Procedural Trees from PovTree

http://propro.ru/go/Wshop/povtree/tutorial.html
Interactive Fractal Tree Design
Popular Modeling Techniques

• Polygon meshes
  – Surface representation, Parametric representation
• Prototype instancing
  – Swipe Shape
• Volume enumeration schemes
  – Volume, Parametric or Implicit
• Parametric curves and surfaces
  – Surface, Parametric
• Subdivision curves and surfaces
• Procedural models
Project 3

• Technical Challenge 1: sky box/sky dome
  – Render the environment within a box or hemisphere with sky texture
  – Google “sky box texture”, you can get many images for this.

http://mpan3.homeip.net/earth
Project 3

• Technical Challenge: sky box/sky dome
  – Render the environment within a box or hemisphere with sky texture
  – Google “sky box texture”, you can get many images for this.

http://skymatter.thegamecreators.com/?f=pack1
Cross-tree using alpha

Side view

Top View

```c
glEnable(GL_ALPHA_TEST);
glAlphaFunc(GL_GREATER, 0.05);
```
Recap on Texture mapping

- Computing Perspective Correct Texture Coord
- Sampling Texture Values (Mipmap, ...)
- OpenGL single texture mapping
- Projector Texture Mapping
- Bump Map
- Displacement Map
- 3D Texture
- Environment Map
- Multi-texturing
- Shadow map
Blending

Billboard
Blending in OpenGL

Source --- In coming fragment color: \((Rs, Gs, Bs, As)\)

\(*\)

Modulate Source color by \((Sr, Sg, Sb, Sa)\)

\(+\)

Destination --- Current Frame buffer color: \((Rd, Gd, Bd, Ad)\)

\(*\)

Modulate Destination color by \((Dr, Dg, Db, Da)\)

\[\text{Final Color: ( ?, ?, ?, ?)}\]

Example:

```c
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
```
## Blending in OpenGL

<table>
<thead>
<tr>
<th>Source --- In coming fragment color:</th>
<th>(Rs, Gs, Bs, As)</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Modulate Source color by</td>
<td>(Sr, Sg, Sb, Sa)</td>
</tr>
<tr>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Destination --- Current Frame buffer color:</td>
<td>(Rd, Gd, Bd, Ad)</td>
</tr>
<tr>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Modulate Destination color by</td>
<td>(Dr, Dg, Db, Da)</td>
</tr>
<tr>
<td>Final Color:</td>
<td>( ?, ?, ?, ?, ?)</td>
</tr>
</tbody>
</table>

Example:
```c
glEnable(GL_BLEND);
glBlendColor(0.3,0.4,0.5,0.6);
glBlendFunc(GL_CONSTANT_COLOR, GL_ONE_MINUS_CONSTANT_COLOR);
```
Other choices

Table 6-1: Source and Destination Blending Factors

<table>
<thead>
<tr>
<th>Constant</th>
<th>Relevant Factor</th>
<th>Computed Blend Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL_ZERO</td>
<td>source or destination</td>
<td>(0, 0, 0, 0)</td>
</tr>
<tr>
<td>GL_ONE</td>
<td>source or destination</td>
<td>(1, 1, 1, 1)</td>
</tr>
<tr>
<td>GL_DST_COLOR</td>
<td>source</td>
<td>(Rd, Gd, Bd, Ad)</td>
</tr>
<tr>
<td>GL_SRC_COLOR</td>
<td>destination</td>
<td>(Rs, Gs, Bs, As)</td>
</tr>
<tr>
<td>GL_ONE_MINUS_DST_COLOR</td>
<td>source</td>
<td>(1, 1, 1, 1)-(Rd, Gd, Bd, Ad)</td>
</tr>
<tr>
<td>GL_ONE_MINUS_SRC_COLOR</td>
<td>destination</td>
<td>(1, 1, 1, 1)-(Rs, Gs, Bs, As)</td>
</tr>
<tr>
<td>GL_SRC_ALPHA</td>
<td>source or destination</td>
<td>(As, As, As, As)</td>
</tr>
<tr>
<td>GL_ONE_MINUS_SRC_ALPHA</td>
<td>source or destination</td>
<td>(1, 1, 1, 1)-(As, As, As, As)</td>
</tr>
<tr>
<td>GL_DST_ALPHA</td>
<td>source or destination</td>
<td>(Ad, Ad, Ad, Ad)</td>
</tr>
<tr>
<td>GL_ONE_MINUS_DST_ALPHA</td>
<td>source or destination</td>
<td>(1, 1, 1, 1)-(Ad, Ad, Ad, Ad)</td>
</tr>
<tr>
<td>GL_SRC_ALPHA_SATURATE</td>
<td>source</td>
<td>(f, f, f, 1); f=min(As, 1-Ad)</td>
</tr>
</tbody>
</table>
Blending depends on Order

• In 3D, the problem is tricky
  – If an opaque obj is in front of a translucent obj
    • Draw opaque
  – If a translucent obj is in front of an opaque
    • Blend
Blending depends on Order

• Solution
  – A-buffer
  – BSP tree
Blending depends on Order

- Hack
  - Draw opaque ones first
  - Freeze depth map, glDepthMask(GL_FALSE);
  - Draw transparent ones
    - If behind the depth map, hide
    - If in front of depth map, blend

- When will it introduce artifacts?