Approximate Catmull-Clark Patches

Scott Schaefer
Charles Loop
Approximate Catmull-Clark Patches

Scott Schaefer
Charles Loop

Catmull-Clark Surface

ACC-Patches
Polygon Models

- Prevalent in game industry
- Very fast to render
- Not smooth (faceted)
- Complicated LOD management
- High-resolution models require lots of band-width and computational resources
Goal: Fast Smooth Surfaces

- Eliminate faceting artifacts
- Animate low-res representation
- Let GPU worry about LOD
DirectX 10 Pipeline

Input Assembler

Vertex Shader

Geometry Shader

Rasterizer

Interpolator

Pixel Shader

Output Merger

Merger

Depth/Stencil

Render Target

Tessellator

Hull Shader

Domain Shader

Texture

Stream Output
Tessellator Unit
Tessellator Unit
Domain Shader

- Tessellation factor per edge
- Called for each vertex of the sample pattern
- Early form in the XBox 360

```c
DS_OUT DS(float2 uv : BARYCENTRIC, int patchInd : INDEX)
{
    DS_OUT Out = (DS_OUT)0;

    // fetch data for patch #patchInd

    // evaluate patch at uv

    return Out;
}
```
Subdivision Surfaces

- Used in movie and game industries
- Supported by most 3D modeling software
Subdivision Surfaces

- Set of rules $S$ that recursively act on a shape $p^0$
  \[ p^{k+1} = S \, p^k \]
- Arbitrary topology surfaces
- Smooth everywhere
Catmull-Clark Surfaces

\[
\begin{align*}
\frac{1+3(v + v^3 - v^3)}{6} & \quad \frac{4 - 6v^2 + 3v^3}{6} \\
\frac{(1 - v)^3}{6} & \quad \frac{4 - 6u^2 + 3u^3}{6} & \quad \frac{1+3(u + u^2 - u^3)}{6} & \quad \frac{u^3}{6}
\end{align*}
\]
Catmull-Clark Surfaces
Catmull-Clark Surfaces
Catmull-Clark Surfaces
Catmull-Clark Surfaces
Catmull-Clark Surfaces
Catmull-Clark Surfaces
Catmull-Clark Surfaces
Stam’s Exact Evaluation Algorithm

- Subdivide until $x$ is in ordinary region

$$S^i P$$

$S^i P$

![Diagram of a grid with a point $x$.]
Stam’s Exact Evaluation Algorithm

- Subdivide until $x$ is in ordinary region $S^i P$
Stam’s Exact Evaluation Algorithm

- Subdivide until $x$ is in ordinary region

$S^i P$
Stam’s Exact Evaluation Algorithm

- Subdivide until $x$ is in ordinary region $S^i P$
- Extract B-spline control points and evaluate at $x$
Stam’s Exact Evaluation Algorithm

- Subdivide until $x$ is in ordinary region

$$V^i \Lambda^{-1} V P$$

- Extract B-spline control points and evaluate at $x$
Performance Issues

- Limits # extraordinary verts
Performance Issues

- Limits # extraordinary verts
Performance Issues

- Limits # extraordinary verts
- Lots of shader constants

\[ V \Lambda^i (V^{-1}P) \]

<table>
<thead>
<tr>
<th>Valence</th>
<th>Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>882</td>
</tr>
<tr>
<td>4</td>
<td>1040</td>
</tr>
<tr>
<td>5</td>
<td>1206</td>
</tr>
<tr>
<td>6</td>
<td>1380</td>
</tr>
<tr>
<td>7</td>
<td>1562</td>
</tr>
<tr>
<td>8</td>
<td>1752</td>
</tr>
</tbody>
</table>
Solution: Polynomial Patches

- Replace extraordinary patches with polynomials
  - Geometry patch (degree 3x3)
  - 2 Tangent patches (degree 3x2)
- Based on conversion from B-spline to Bezier form
Solution: Polynomial Patches

- Replace extraordinary patches with polynomials
  - **Geometry patch** (degree 3x3)
  - 2 Tangent patches (degree 3x2)
- Based on conversion from B-spline to Bezier form
Solution: Polynomial Patches

- Replace extraordinary patches with polynomials
  - Geometry patch (degree 3x3)
  - 2 Tangent patches (degree 3x2)
- Based on conversion from B-spline to Bezier form
Solution: Polynomial Patches

- Replace extraordinary patches with polynomials
  - Geometry patch (degree 3x3)
  - 2 Tangent patches (degree 3x2)
- Based on conversion from B-spline to Bezier form
ACC Geometry Patches

- Use knot-insertion rules from ordinary case
- At corners, use limit masks
ACC Geometry Patches

- Use knot-insertion rules from ordinary case
- At corners, use limit masks
ACC Geometry Patches

- Use knot-insertion rules from ordinary case
- At corners, use limit masks
ACC Geometry Patches

- Use knot-insertion rules from ordinary case
- At corners, use limit masks

\[
\frac{1}{n+5} \quad \frac{1}{2(n+5)}
\]

\[
\frac{n}{n+5} \quad \frac{2}{n+5}
\]
ACC Geometry Patches

- Use knot-insertion rules from ordinary case
- At corners, use limit masks
ACC Geometry Patches

■ Use knot-insertion rules from ordinary case
■ At corners, use limit masks
■ Smooth everywhere except edges touching extraordinary vertices
ACC Tangent Patches

- Continuous normal field needed for displacement mapping
- Create two tangent patches \((u/v)\) for each position patch
ACC Tangent Patches

- Continuous normal field needed for displacement mapping
- Create two tangent patches \((u/v)\) for each position patch
ACC Tangent Patches

- Continuous normal field needed for displacement mapping
- Create two tangent patches \((u/v)\) for each position patch
ACC Tangent Patches

- Continuous normal field needed for displacement mapping
- Create two tangent patches \((u/v)\) for each position patch
ACC Tangent Patches

- Continuous normal field needed for displacement mapping
- Create two tangent patches \((u/v)\) for each position patch
$v(t) \times u(t) \propto \hat{v}(t) \times u(t)$
ACC Tangent Patches

\[ \nu_{10} = \frac{2}{3} c_0 \nu_{10} - \frac{1}{3} c_1 \nu_{00} \]

\[ \nu_{20} = \frac{1}{3} c_0 \nu_{20} - \frac{2}{3} c_1 \nu_{10} \]

\[ c_i = \cos \left( \frac{2\pi}{n_i} \right) \]
Examples
Examples

ACC Geometry Patches
Examples

ACC Geo/Tan Patches
Examples

Catmull-Clark Surface
Examples
Examples

ACC Geometry Patches
Examples

ACC Geo/Tan Patches
Examples

Catmull-Clark Surface
Examples
Examples

ACC Geometry Patches
Examples

ACC Geo/Tan Patches
Examples

Catmull-Clark Surface
Examples

ACC Geo/Tan

Catmull-Clark
Examples
Examples

ACC Geometry Patches
Examples

ACC Geo/Tan Patches
Examples

Catmull-Clark Surface
Comparison

PN Triangles

PCCM

ACC Patches

Normal Error

Geometry Error

Better

Worse
Comparison

- PN Triangles
- PCCM
- ACC Patches

Normal Error

Geometry Error

Better

Worse
Comparison

PN Triangles | PCCM | ACC Patches

Normal Error

Geometry Error

Better  Worse
Comparison

PN Triangles  PCCM  ACC Patches

Normal Error

Geometry Error

Better  Worse
Conclusions

- Creates visually smooth surfaces
- Suitable for displacement/normal mapping
- Handles any number of extraordinary vertices
- Simple to evaluate

- Demo part of DirectX March 2008 SDK