Subdivision Surfaces

COS 526: Advanced Computer Graphics

Slide credits: Denis Zorin, Peter Schröder, Ravi Ramamoorthi
Video: Geri’s Game
Subdivision Surfaces

- Coarse mesh & subdivision rule
  - Smooth surface = limit of sequence of refinements
Key Questions

- How to refine mesh? (“Topology”)
- Where to place new vertices? (“Geometry”)

[Zorin & Schröder]
Loop Subdivision Scheme

- How refine mesh?
  - Refine each triangle into 4 triangles by splitting each edge and connecting new vertices

[Zorin & Schröder]
Loop Subdivision Scheme

- Where to place new vertices?
  - Choose locations for new vertices as weighted average of original vertices in local neighborhood

[Zorin & Schröder]
Loop Subdivision Scheme

- Where to place new vertices?
  - Rules for \textit{extraordinary vertices} and \textit{boundaries}:

\begin{itemize}
  \item \textbf{Interior}
  \item \textbf{Crease and boundary}
\end{itemize}

\textbf{a. Masks for odd vertices}

\textbf{b. Masks for even vertices}

[Zorin & Schröder]
Choose $\beta$ by analyzing continuity of limit surface

- **Original Loop**

$$\beta = \frac{1}{n} \left( \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{n} \right)^2 \right)$$

- **Warren**

$$\beta = \begin{cases} 
\frac{3}{8n} & n > 3 \\
\frac{3}{16} & n = 3 
\end{cases}$$
Butterfly Subdivision

- Interpolating subdivision: larger neighborhood
Modified Butterfly Subdivision

Need special weights near extraordinary vertices

- For $n = 3$, weights are $\frac{5}{12}, -\frac{1}{12}, -\frac{1}{12}$
- For $n = 4$, weights are $\frac{3}{8}, 0, -\frac{1}{8}, 0$
- For $n \geq 5$, weights are

$$\frac{1}{n}\left(\frac{1}{4} + \cos\left(\frac{2\pi j}{n}\right) + \frac{1}{2} \cos\left(\frac{4\pi j}{n}\right)\right), \quad j = 0..n - 1$$

- Weight of extraordinary vertex = $1 - \sum$ other weights
A Variety of Subdivision Schemes

- Triangles vs. Quads
- Interpolating vs. approximating

<table>
<thead>
<tr>
<th>Face split</th>
<th>Triangular meshes</th>
<th>Quad. meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Approximating</strong></td>
<td>Loop ($C^2$)</td>
<td>Catmull-Clark ($C^2$)</td>
</tr>
<tr>
<td><strong>Interpolating</strong></td>
<td>Mod. Butterfly ($C^1$)</td>
<td>Kobbelt ($C^1$)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vertex split</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Doo-Sabin, Midedge ($C^1$)</td>
<td></td>
</tr>
<tr>
<td>Biquartic ($C^2$)</td>
<td></td>
</tr>
</tbody>
</table>

[Zorin & Schröder]
More Exotic Methods

• Kobbelt’s subdivision:
More Exotic Methods

- Kobbel’s subdivision:

  - Number of faces triples per iteration: gives finer control over polygon count
Subdivision Schemes

Loop
Butterfly
Catmull-Clark
Doo-Sabin

[Zorin & Schröder]
Subdivision Schemes

Loop
Butterfly
Catmull-Clark
Doo-Sabin

[Zorin & Schröder]
Analyzing Subdivision Schemes

- Limit surface has provable smoothness properties
Analyzing Subdivision Schemes

• Start with curves: 4-point interpolating scheme

(old points left where they are)
4-Point Scheme

• What is the support?

Step $i$:

$\mathbf{v}_{-2} \quad \mathbf{v}_{-1} \quad \mathbf{v}_0 \quad \mathbf{v}_1 \quad \mathbf{v}_2$

Step $i+1$:

$\mathbf{v}_{-2} \quad \mathbf{v}_{-1} \quad \mathbf{v}_0 \quad \mathbf{v}_1 \quad \mathbf{v}_2$

So, 5 new points depend on 5 old points
Subdivision Matrix

- How are vertices in neighborhood refined? (with vertex renumbering like in last slide)

\[
\begin{pmatrix}
V_{-2}^{(i+1)} \\
V_{-1}^{(i+1)} \\
V_0^{(i+1)} \\
V_1^{(i+1)} \\
V_2^{(i+1)}
\end{pmatrix}
= 
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
-\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \\
0 & 0 & 0 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
V_{-2}^{(i)} \\
V_{-1}^{(i)} \\
V_0^{(i)} \\
V_1^{(i)} \\
V_2^{(i)}
\end{pmatrix}
\]
Subdivision Matrix

- How are vertices in neighborhood refined? (with vertex renumbering like in last slide)

\[ \vec{V}^{(i+1)} = S \vec{V}^{(i)} \]

After \( n \) rounds:

\[ \vec{V}^{(n)} = S^n \vec{V}^{(0)} \]
Convergence Criterion

Expand in eigenvectors of $S$:

\[ \vec{V}^{(n)} = S^n \vec{V}^{(0)} \]

\[
S = \sum_{i=0}^{4} \lambda_i e_i e_i^T
\]

\[
\vec{V}^{(0)} = \sum_{i=0}^{4} a_i e_i
\]

\[
\vec{V}^{(n)} = \sum_{i=0}^{4} a_i \lambda_i^n e_i
\]

Criterion I: $|\lambda_i| \leq 1$
Convergence Criterion

• What if all eigenvalues of $S$ are $< 1$?
  – All points converge to 0 with repeated subdivision

Criterion II: $\lambda_0 = 1$
Translation Invariance

• For any translation $t$, want:

\[
\begin{bmatrix}
V^{(i+1)}_{-2} + t \\
V^{(i+1)}_{-1} + t \\
V^{(i+1)}_0 + t \\
V^{(i+1)}_1 + t \\
V^{(i+1)}_2 + t
\end{bmatrix} = S
\begin{bmatrix}
V^{(i)}_{-2} + t \\
V^{(i)}_{-1} + t \\
V^{(i)}_0 + t \\
V^{(i)}_1 + t \\
V^{(i)}_2 + t
\end{bmatrix}
\]

\[
\vec{V}^{(i+1)} + t\vec{1} = S(\vec{V}^{(i)} + t\vec{1})
\]

\[
S\vec{1} = \vec{1}
\]

Criterion III: $e_0 = 1$, all other $|\lambda_i| < 1$
Smoothness Criterion

- Plug back in:
  \[
  \vec{V}^{(n)} = a_0 \vec{e}_0 + \sum_{i=1}^{4} a_i \lambda_i^n \vec{e}_i
  \]
- Dominated by largest \( \lambda_i \)

- Case 1: \( |\lambda_1| > |\lambda_2| \)

- \( \vec{V}^{(n)} = a_0 \vec{e}_0 + a_1 \lambda_1^n \vec{e}_1 + (small) \)
  - Group of 5 points gets shorter
  - All points approach multiples of \( \vec{e}_1 \to \) on a straight line
  - Smooth!
Case 2: $|\lambda_1| = |\lambda_2|$

- Points can be anywhere in space spanned by $e_1, e_2$
- No longer have smoothness guarantee

Criterion IV: Smooth iff $\lambda_0 = 1 > |\lambda_1| > |\lambda_i|$
Continuity and Smoothness

• So, what about 4-point scheme?
  – Eigenvalues = $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{8}$
  – $e_0 = 1$
  – Stable ✓
  – Translation invariant ✓
  – Smooth ✓
2-Point Scheme

- In contrast, consider 2-point interpolating scheme

\[
\begin{array}{ccc}
\frac{1}{2} & \frac{1}{2} & 0 \\
0 & 1 & 0 \\
0 & \frac{1}{2} & \frac{1}{2}
\end{array}
\]

- Support = 3
Continuity of 2-Point Scheme

- Eigenvalues = $1, \frac{1}{2}, \frac{1}{2}$
- $e_0 = 1$
- Stable ✓
- Translation invariant ✓
- Smooth X
  - Not smooth; in fact, this is piecewise linear
Now consider 2-point *approximating* scheme

- Support = 2
- Subdivision matrix = 
  
  \[
  \begin{pmatrix}
  \frac{3}{4} & \frac{1}{4} \\
  \frac{1}{4} & \frac{3}{4}
  \end{pmatrix}
  \]
Continuity of 2-Point Approximating Scheme

- Eigenvalues = $1, \frac{1}{2}$
- $e_0 = 1$
- Stable ✓
- Translation invariant ✓
- Smooth ✓
  - This is equivalent to 2$\text{nd}$ order B spline
For Surfaces…

• Similar analysis: determine support, construct subdivision matrix, find eigenstuff
  – Caveat 1: separate analysis for each vertex valence
  – Caveat 2: consider more than 1 subdominant eigenvalue

Reif’s smoothness condition: $\lambda_0 = 1 > |\lambda_1| \geq |\lambda_2| > |\lambda_i|$

• Points lie in subspace spanned by $e_1$ and $e_2$
  – If $|\lambda_1| \neq |\lambda_2|$, neighborhood stretched when subdivided, but remains 2-manifold
Fun with Subdivision Methods

Behavior of surfaces depends on eigenvalues

(Real) Complex Degenerate

(recall that symmetric matrices have real eigenvalues)
Summary

• Advantages:
  – Simple method for describing complex, smooth surfaces
  – Relatively easy to implement
  – Arbitrary topology
  – Local support
  – Guaranteed continuity
  – Multiresolution

• Difficulties:
  – Intuitive specification
  – Parameterization
  – Intersections