Surface Representations and Geometric Modeling

Prof. Dr. Markus Gross
Subdivision Surfaces

- Generalization of spline curves / surfaces
  - Arbitrary control meshes
  - Successive refinement (subdivision)
  - Converges to smooth limit surface
  - Connection between splines and meshes
Subdivision Surfaces

• Generalization of spline curves / surfaces
  – Arbitrary control meshes
  – Successive refinement (subdivision)
  – Converges to smooth limit surface
  – Connection between splines and meshes
Example: Geri’s Game (Pixar)

- Subdivision used for
  - Geri’s hands and head
  - Clothing
  - Tie and shoes
Example: Geri’s Game  (Pixar)

Woody’s hand (NURBS)  Geri’s hand (subdivision)
Example: Geri’s Game (Pixar)

- Sharp and semi-sharp features
Example: Dassault’s CATIA

- Traditional CAD system based on splines
- Latest version also includes subdivision surfaces
Subdivision Curves

Given a control polygon...

...find a smooth curve related to that polygon.
Subdivision Curve Types

- Approximating
- Interpolating
- Corner Cutting
Approximating
Approximating

Splitting step: split each edge in two
Approximating

Averaging step: relocate each (original) vertex according to some (simple) rule...
Approximating

Start over ...
Approximating

...splitting...
Approximating

...averaging...
Approximating

...and so on...
Approximating

If the rule is designed carefully...

... the control polygons will converge to a smooth limit curve!
Equivalent to ...  

- Insert *single* new point at mid-edge  
- *Filter* entire set of points.

Catmull-Clark rule: Filter with (1/8, 6/8, 1/8)
Corner Cutting

- Subdivision rule:
  - Insert *two* new vertices at $\frac{1}{4}$ and $\frac{3}{4}$ of each edge
  - *Remove* the old vertices
  - Connect the new vertices
B-Spline Curves

- Piecewise polynomial of degree $n$

B-spline curve

$$s(u) = \sum_{i=0}^{k} d_i N_i^n(u)$$

control points

parameter value

basis functions
B-Spline Curves

- Distinguish between odd and even points

- Linear B-spline
  - Odd coefficients \((1/2, 1/2)\)
  - Even coefficient \((1)\)
B-Spline Curves

- **Quadratic B-Spline (Chaikin)**
  - Odd coefficients \((\frac{1}{4}, \frac{3}{4})\)
  - Even coefficients \((\frac{3}{4}, \frac{1}{4})\)

- **Cubic B-Spline (Catmull-Clark)**
  - Odd coefficients \((\frac{4}{8}, \frac{4}{8})\)
  - Even coefficients \((\frac{1}{8}, \frac{6}{8}, \frac{1}{8})\)
Cubic B-Spline
Cubic B-Spline
Cubic B-Spline

odd

even

\[
\begin{align*}
\frac{4}{8} & \quad \frac{4}{8} & \frac{1}{8} & \quad \frac{6}{8} & \quad \frac{1}{8} \\
\end{align*}
\]
Cubic B-Spline
Cubic B-Spline

odd

\[
\begin{array}{cc}
\frac{4}{8} & \frac{4}{8} \\
\end{array}
\]

even

\[
\begin{array}{cc}
\frac{1}{8} & \frac{6}{8} & \frac{1}{8} \\
\end{array}
\]

\[
\begin{array}{c}
\frac{1}{8} \\
\frac{1}{8} \\
\frac{6}{8} \\
\end{array}
\]
Cubic B-Spline

odd

\[
\begin{align*}
\frac{4}{8} & \quad \frac{4}{8} \\
\frac{1}{8} & \quad \frac{6}{8} & \quad \frac{1}{8}
\end{align*}
\]

even

\[
\begin{align*}
\frac{4}{8} & \quad \frac{4}{8} & \quad \frac{4}{8}
\end{align*}
\]
Cubic B-Spline

\[
\begin{align*}
\frac{4}{8} & \quad \frac{4}{8} \\
\text{odd} & \quad \text{even}
\end{align*}
\]
Cubic B-Spline

odd

even

\[
\frac{4}{8}, \frac{4}{8}, \frac{1}{8}, \frac{6}{8}, \frac{1}{8}, \frac{4}{8}
\]
Cubic B-Spline
Cubic B-Spline

odd

even

\[
\begin{array}{cccc}
\frac{4}{8} & \frac{4}{8} & \frac{1}{8} & \frac{6}{8} & \frac{1}{8} & \frac{4}{8} \\
\end{array}
\]
Cubic B-Spline

odd

\[ \frac{4}{8}, \frac{4}{8} \]

\[ \frac{1}{8}, \frac{6}{8}, \frac{1}{8} \]

even

Even

\[ \frac{1}{8} \]

\[ \frac{6}{8} \]
Cubic B-Spline

odd

even
Cubic B-Spline

odd

even
Cubic B-Spline

odd

Even

\[ \frac{4}{8} \quad \frac{4}{8} \quad \frac{1}{8} \quad \frac{6}{8} \quad \frac{1}{8} \]
B-Spline Curves

• Subdivision rules for control polygon
  \[ d^0 \rightarrow d^1 = Sd^0 \rightarrow \ldots \rightarrow d^j = Sd^{j-1} = S^j d^0 \]

• Mask of size \( n \) yields \( C^{n-1} \) curve
Interpolating (4-point Scheme)

- Keep old vertices
- Generate new vertices by fitting cubic curve to old vertices
- $C^1$ continuous limit curve

![Diagram](enter image description here)

\[ f(x) = ax^3 + bx^2 + cx + d \]

\[ f(j) = \mathbf{p}_{i+j}, \quad j = 0, \ldots, 3 \]

\[ q_i = f(3/2) = \frac{1}{16}(-\mathbf{p}_i + 9\mathbf{p}_{i+1} + 9\mathbf{p}_{i+2} - \mathbf{p}_{i+3}) \]
Interpolating
Interpolating
Interpolating
Interpolating
Interpolating demo
Subdivision Surfaces

- No regular structure as for curves
  - Arbitrary number of edge-neighbors
  - Different subdivision rules for each valence
Classic Subdivision Operators

• Classification of subdivision schemes

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primal</td>
<td>Faces are split into sub-faces</td>
</tr>
<tr>
<td>Dual</td>
<td>Vertices are split into multiple vertices</td>
</tr>
<tr>
<td>Approximating</td>
<td>Control points are not interpolated</td>
</tr>
<tr>
<td>Interpolating</td>
<td>Control points are interpolated</td>
</tr>
</tbody>
</table>
### Classic Subdivision Operators

- Classification of subdivision schemes

<table>
<thead>
<tr>
<th></th>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangles</td>
<td>Rectangles</td>
<td></td>
</tr>
<tr>
<td>Approximating</td>
<td>Loop</td>
<td>Catmull-Clark</td>
</tr>
<tr>
<td>Interpolating</td>
<td>Butterfly</td>
<td>Kobbelt</td>
</tr>
</tbody>
</table>
## Classic Subdivision Operators

### Classification of subdivision schemes

<table>
<thead>
<tr>
<th>Approximating</th>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangles</td>
<td>Rectangles</td>
<td></td>
</tr>
<tr>
<td>Loop</td>
<td>Catmull-Clark</td>
<td>Doo-Sabin Midedge</td>
</tr>
<tr>
<td>Butterfly</td>
<td>Kobbelt</td>
<td></td>
</tr>
</tbody>
</table>
Doo-Sabin Subdivision

- Generalization of \textit{bi-quadratic} B-Splines
- Dual, approximating subdivision scheme
- Applied to \textit{polygonal} meshes
- Generates $G^1$ \textit{continuous} limit surfaces:
  - $C^0$ for the set of finite extraordinary points
  - Center of irregular polygons after 1 subdivision step
  - $C^1$ continuous everywhere else
Doo-Sabin Subdivision

\[ V_2 = \frac{1}{n} \sum_{j=1}^{n} d_j \]

\[ E_i = \frac{1}{2} (d_1 + d_{2i}) \]

\[ d'_{1,j} = \frac{1}{4} (d_1 + E_j + E_{j-1} + V_j) \]
Doo-Sabin Subdivision
Classic Subdivision Operators

- Classification of subdivision schemes

<table>
<thead>
<tr>
<th></th>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Triangles</td>
<td>Rectangles</td>
</tr>
<tr>
<td>Approximating</td>
<td>Loop</td>
<td>Catmull-Clark</td>
</tr>
<tr>
<td>Interpolating</td>
<td>Butterfly</td>
<td>Kobbelt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Doo-Sabin Midedge</td>
</tr>
</tbody>
</table>
Catmull-Clark Subdivision

- Generalization of bi-cubic B-Splines
- Primal, approximation subdivision scheme
- Applied to polygonal meshes
- Generates $G^2$ continuous limit surfaces:
  - $C^1$ for the set of finite extraordinary points
    - Vertices with valence $\neq 4$
  - $C^2$ continuous everywhere else
Catmull-Clark Subdivision

\[ V_2 = \frac{1}{n} \times \sum_{j=1}^{n} d_j \]

\[ E_i = \frac{1}{4} \left( d_1 + d_{2i} + V_i + V_{i+1} \right) \]

\[ d'_1 = \frac{(n-3)}{n} d_1 + \frac{2}{n} R + \frac{1}{n} S \]

\[ R = \frac{1}{m} \sum_{i=1}^{m} E_i \]

\[ S = \frac{1}{m} \sum_{i=1}^{m} V_i \]
Catmull-Clark Subdivision
## Classic Subdivision Operators

- Classification of subdivision schemes

<table>
<thead>
<tr>
<th>Approximating</th>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangles</td>
<td>Loop</td>
<td>Doo-Sabin</td>
</tr>
<tr>
<td>Rectangles</td>
<td>Catmull-Clark</td>
<td>Midedge</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interpolating</th>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Butterfly</td>
<td>Kobbelt</td>
<td></td>
</tr>
</tbody>
</table>
Loop Subdivision

- Generalization of *box splines*
- Primal, approximating subdivision scheme
- Applied to *triangle* meshes
- Generates $G^2$ *continuous* limit surfaces:
  - $C^1$ for the set of finite extraordinary points
    - Vertices with valence $\neq 6$
  - $C^2$ continuous everywhere else
Loop Subdivision

\[ E_i = \frac{3}{8} (d_1 + d_i) + \frac{1}{8} (d_{i-1} + d_{i+1}) \]

\[ d'_1 = \alpha_n d_1 + \frac{1 - \alpha_n}{n} \sum_{j=2}^{n+1} d_j \]

\[ \alpha_n = \frac{3}{8} + \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{n} \right)^2 \]
Loop Subdivision
### Classic Subdivision Operators

- Classification of subdivision schemes

<table>
<thead>
<tr>
<th></th>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approximating</td>
<td>Triangles</td>
<td>Rectangles</td>
</tr>
<tr>
<td></td>
<td>Loop</td>
<td>Catmull-Clark</td>
</tr>
<tr>
<td>Interpolating</td>
<td>Butterfly</td>
<td>Kobbelt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Doo-Sabin Midedge</td>
</tr>
</tbody>
</table>
Butterfly Subdivision

- Primal, interpolating scheme
- Applied to *triangle* meshes
- Generates $G^1$ continuous limit surfaces:
  - $C^0$ for the set of finite extraordinary points
    - Vertices of valence $= 3$ or $> 7$
  - $C^1$ continuous everywhere else
Butterfly Subdivision

\[ E_1 = \frac{1}{2} (d_1 + d_2) + \omega (d_3 + d_4) - \frac{\omega}{2} (d_5 + d_6 + d_7 + d_8) \]

\[ d'_i = d_i \]
Butterfly Subdivision
Remark

- Different masks apply on the boundary
- Example: Loop

![Diagram](image-url)
Comparison

Doo-Sabin

Catmull-Clark

Loop

Butterfly
Comparison

Loop
Butterfly
Catmull-Clark
Doo-Sabin
Catmull-Clark
Doo-Sabin
Comparison

Loop
Butterfly
Catmull-Clark
Doo-Sabin

Initial mesh
Loop
Catmull-Clark
Catmull-Clark after triangulation
Comparison

• Continuity of a scheme determines the quality
• Approximating schemes give best results
• Approximating schemes shrink meshes
• Best schemes: Catmull-Clark & Loop
  – Loop is applied on triangle meshes
  – Catmull-Clark preserves symmetry better
Analysis of Subdivision

• Invariant neighborhoods
  – How many control-points affect a small neighborhood around a point?

• Subdivision scheme can be analyzed by looking at a local subdivision matrix
Local Subdivision Matrix

- Example: Cubic B-Splines

\[
\begin{pmatrix}
  p_{-2}^{j+1} \\
p_{-1}^{j+1} \\
p_0^{j+1} \\
p_1^{j+1} \\
p_2^{j+1}
\end{pmatrix}
= \frac{1}{8}
\begin{pmatrix}
  1 & 6 & 1 & 0 & 0 \\
  0 & 4 & 4 & 0 & 0 \\
  0 & 1 & 6 & 1 & 0 \\
  0 & 0 & 4 & 4 & 0 \\
  0 & 0 & 1 & 6 & 1
\end{pmatrix}
\begin{pmatrix}
p_{-2}^j \\
p_{-1}^j \\
p_0^j \\
p_1^j \\
p_2^j
\end{pmatrix}
\]

- Invariant neighborhood size: 5
Analysis of Subdivision

- Analysis via eigen-decomposition of matrix $S$
  - Compute the eigenvalues
    $$\{\lambda_0, \lambda_1, \ldots, \lambda_{n-1}\}$$
  - and eigenvectors
    $$X = \{x_0, x_1, \ldots, x_{n-1}\}$$
  - Let $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{n-1}$ be real and $X$ a complete set of eigenvectors
Analysis of Subdivision

- Invariance under affine transformations
  \[ \text{transform} (\text{subdivide}(P)) = \text{subdivide} (\text{transform}(P)) \]
Analysis of Subdivision

- Invariance under affine transformations
  - \( \text{transform}(\text{subdivide}(P)) = \text{subdivide}(\text{transform}(P)) \)

- Rules have to be affine combinations
  - Even and odd weights each sum to 1

\[
\sum_j S_{2i,j} = \sum_j S_{2i+1,j} = 1
\]
Analysis of Subdivision

- Invariance under reversion of point ordering
- Subdivision rules (matrix rows) have to be symmetric
Analysis of Subdivision

Conclusion: 1 has to be eigenvector of $S$ with eigenvalue $\lambda_0=1$
Limit Behavior - Position

- Any vector is linear combination of eigenvectors:

\[ p = \sum_{i=0}^{n-1} a_i x_i \quad a_i = x_i^T p \]

- Apply subdivision matrix:

\[ S p^0 = S \sum_{i=0}^{n-1} a_i x_i = \sum_{i=0}^{n-1} a_i S x_i = \sum_{i=0}^{n-1} a_i \lambda_i x_i \]
Limit Behavior - Position

- For convergence we need $1 = \lambda_0 > \lambda_1 \geq \cdots \geq \lambda_{n-1}$

- Limit vector:

$$p^\infty = \lim_{j \to \infty} S^j p^0 = \lim_{j \to \infty} \sum_{i=0}^{n-1} a_i \lambda_i^j x_i = a_0 \cdot 1$$

$$p_i^\infty = a_0 = \bar{x}_0^T p^j \quad \text{independent of } j!$$
Limit Behavior - Tangent

- Set origin at $a_0$:
  \[ p^j = \sum_{i=1}^{n-1} a_i \lambda_i^j x_i \]

- Divide by $\lambda_1^j$
  \[
  \frac{1}{\lambda_1^j} p^j = a_1 x_1 + \sum_{i=2}^{n-1} a_i \left( \frac{\lambda_i}{\lambda_1} \right)^j x_i
  
  \]

- Limit tangent given by:
  \[ t^\infty_i = a_1 = \tilde{x}_1^T p^j \]
Limit Behavior - Tangent

- **Curves:**
  - All eigenvalues of $S$ except $\lambda_0=1$ should be less than $\lambda_1$ to ensure existence of a tangent, i.e.

$$1 = \lambda_0 > \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_{n-1}$$

- **Surfaces:**
  - Tangents determined by $\lambda_1$ and $\lambda_2$

$$1 = \lambda_0 > \lambda_1 = \lambda_2 > \lambda_3 \geq \cdots \geq \lambda_{n-1}$$
Example: Cubic Splines

- Subdivision matrix & rules

\[
S = \frac{1}{8} \begin{pmatrix}
1 & 6 & 1 & 0 & 0 \\
0 & 4 & 4 & 0 & 0 \\
0 & 1 & 6 & 1 & 0 \\
0 & 0 & 4 & 4 & 0 \\
0 & 0 & 1 & 6 & 1
\end{pmatrix}
\]

\[
p_{2i}^{j+1} = \frac{1}{8}p_{i-1}^j + \frac{6}{8}p_i^j + \frac{1}{8}p_{i+1}^j
\]

\[
p_{2i+1}^{j+1} = \frac{1}{2}p_i^j + \frac{1}{2}p_{i+1}^j
\]

- Eigenvalues

\[
(\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4, ) = \left(1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right)
\]
Example: Cubic Splines

- Eigenvectors

\[
X = \begin{pmatrix}
1 & -1 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 \\
1 & -1 & -1 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 \\
\end{pmatrix}
\quad
X^{-1} = \begin{pmatrix}
0 & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} & 0 \\
0 & -1 & 0 & 1 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\end{pmatrix}
\]

- Limit position and tangent

\[
p_i^\infty = \ddot{x}_0 p^i = \frac{1}{6} \left( p_{i-1}^j + 4p_i^j + p_{i+1}^j \right)
\]

\[
t_i^\infty = \ddot{x}_1 p^i = p_{i+1}^j - p_i^j
\]
Properties of Subdivision

• Flexible modeling
  – Handle surfaces of arbitrary topology
  – Provably smooth limit surfaces
  – Intuitive control point interaction

• Scalability
  – Level-of-detail rendering
  – Adaptive approximation

• Usability
  – Compact representation
  – Simple and efficient code