B-Spline and Subdivision Surfaces

Slides from Mirela Ben-Chen

Images from: 3drender.com & sunflow.sourceforge.net
Surface Models

• B-Spline surfaces
  – NURBS surfaces

• Subdivision surfaces
  – Theory
  – Zoo
Reminder: B-Spline Curves

*B-Spline Curve*

Decouple number of control points from degree of curve

“Glue” a few degree $p$ Bézier curves, with continuity conditions

[Applet (Curve)](javascript:openApplet('Curve'))
Reminder: B-Spline Curves

**B-Spline Curve**

**Building blocks:**

- $n + 1$ control points $P_i$
- Knot vector $U = \{ u_0, u_1, \ldots, u_m \}$
- The degree $p$
- $m, n, p$ satisfy $m = n + p - 1$

$$C(t) = \sum_{i=0}^{n} N_{i,p}(t)P_i$$

$$N_{i,0}(u) = \begin{cases} 1 & u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

**Applet (Basis functions)**
B-Spline Surfaces

A collection of Bezier patches, with continuity conditions

Decoupling the degree and the number of control points
B-Spline Surfaces

*B-Spline surface* - tensor product surface of B-Spline curves

\[
S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) P_{ij}
\]

Building blocks:

- Control net, \( m + 1 \) rows, \( n + 1 \) columns: \( P_{ij} \)
- Knot vectors \( U = \{ u_0, u_1, \ldots, u_h \} \), \( V = \{ v_0, v_1, \ldots, v_k \} \)
- The degrees \( p \) and \( q \) for the \( u \) and \( v \) directions
Basis Functions

Cubic $\times$ Quadratic basis functions:
Properties

• Non negativity

\[ N_{i,p}(u)N_{j,q}(v) \geq 0, \quad \text{for all } i, j, p, q, u, v \]

• Partition of unity

\[
\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v) = 1, \quad \text{for all } (u, v) \in [0,1] \times [0,1]
\]

• Affine invariance

\[
AS_{\{p_{ij}\}}(u,v) + B = S_{\{AP_{ij} + B\}}(u,v)
\]

for all \( A \in R^{3x3}, B \in R^{3}, (u,v) \in [0,1] \times [0,1] \)
Properties

• If \( n = p, \ m = q, \ U = \{ 0, \ldots, 0, 1, \ldots, 1 \} \) and \( V = \{ 0, \ldots, 0, 1, \ldots, 1 \} \) then

\[
N_{i,p}(u)N_{j,q}(v) = B^n_i(u)B^m_j(v)
\]

and \( S(u,v) \) is a Bézier surface

• \( S(u,v) \) is \( C^{p-k} \) continuous in the \( u \) direction at a \( u \) knot of multiplicity \( k \), and similar for \( v \) direction
Properties

• Compact support

\[ N_{i,p}(u)N_{j,q}(v) = 0, \quad \text{for all } (u,v) \notin [u_i,u_{i+p+1}] \times [v_j,v_{j+q+1}] \]

• Local modification scheme
  – Moving \( P_{ij} \) affects the surface only in the rectangle \([u_i,u_{i+p+1}] \times [v_j,v_{j+q+1}]\)
Properties

• Local definition
  – In any rectangle \([u_{i_0}, u_{i_0+1}] \times [v_{j_0}, v_{j_0+1}]\) the only non-zero basis functions are
    \[N_{i,p}(u)N_{j,q}(v), \quad \text{for } i_0 - p \leq i \leq i_0 \quad \text{and} \quad j_0 - q \leq j \leq j_0\]

• Strong convex hull property
  – If \((u,v) \in [u_{i_0}, u_{i_0+1}] \times [v_{j_0}, v_{j_0+1}]\) then \(S(u,v)\) is in the convex hull of the control points \(P_{ij} \quad i_0 - p \leq i \leq i_0 \quad \text{and} \quad j_0 - q \leq j \leq j_0\)
Reminder: NURBS Curves

• B-spline curves cannot represent exactly circles and ellipses

• Generalize to *rational polynomials*

\[
C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i}
\]
Reminder: NURBS Curves

A weight per control point allows to change the influence of a point on the curve, without moving the point.
NURBS Surfaces

Add a weight for every control point of a B-spline surface, and normalize

\[ S(u, v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{ij}P_{ij}}{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{ij}} \]

Is not a tensor product patch
NURBS Surface Example

Control net

\[ U = V = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\} \]

\[ w_{ij}(\bullet) = 10, \ w_{ij}(\circ) = 1 \]
NURBS Surfaces
Problems with NURBS

• A single NURBS patch is either a topological disk, a tube or a torus

• Must use many NURBS patches to model complex geometry

• When deforming a surface made of NURBS patches, cracks arise at the seams
Subdivision

“Subdivision defines a smooth curve or surface as the limit of a sequence of successive refinements”
Subdivision Rules

• How the connectivity changes

• How the geometry changes
  – Old points
  – New points
Design Goals for Subd Rules

• Efficiency
• Compact support
• Local definition
• Affine invariance
• Simplicity
• Smoothness

Same properties NURBS have, but will work for any topology
An Example – Loop Scheme

• Connectivity

• Geometry

• Analysis?
  – Does it converge?
  – Is the limit surface smooth?
  – Any problems at extraordinary (valence != 6) vertices?

{\[ \beta = \frac{1}{k} \left( \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{k} \right)^2 \right) \]
Parameterization of Subd Surfaces

• B-spline curves and surfaces are *parameterized*

\[ S(t) : [t_0, t_m] \rightarrow \mathbb{R}^2 \]

\[ S(u,v) : [u_0, u_h] \times [v_0, v_k] \rightarrow \mathbb{R}^3 \]

• To analyze subd schemes, we need a similar parameterization

• Which domain to use? A planar rectangle cannot work

• Solution: Use initial control mesh as the domain
Parameterization of Subd Surfaces

• Apply subd rules to initial mesh, *without updating the geometry*

• Use resulting polyhedron as the domain
Smoothness of Surfaces

$C^1$ continuity

A surface $f: |K| \rightarrow R^3$ is $C^1$ continuous if for every point $x \in |K|$ there exists a regular parameterization $\pi : D \rightarrow f(U_x)$, over a unit disk $D$ in the plane, where $U_x$ is the neighborhood in $|K|$ of $x$.

A regular parameterization $\pi$ is one that is continuously differentiable, one-to-one, and has a Jacobi matrix of maximum rank.
Subdivision Properties

• How can we prove properties of a subdivision scheme?

• Express the subdivision as a local matrix operation

• Prove properties (convergence, continuity, affine invariance, etc.) using eigen-analysis.
Subdivision Matrix

- Look at the local neighborhood of an extraordinary vertex $v$

- Let $U^j$ be the set of vertices in the 2-ring neighborhood of $v$ after $j$ Loop subdivision steps

- Let $p_i^j$ be the corresponding control points

- We can compute $p_i^{j+1}$ using only $p_i^j$
Subdivision Matrix

• An extraordinary vertex of degree $k$ has $\hat{k} = 3k + 1$ vertices in its 2-ring neighborhood.

\[
\begin{pmatrix}
 p_0^{j+1} \\
 \vdots \\
 p_{3k}^{j+1}
\end{pmatrix}_{\hat{k} \times 3}
 = S_{\hat{k} \times \hat{k}}
\begin{pmatrix}
 p_0^j \\
 \vdots \\
 p_{3k}^j
\end{pmatrix}_{\hat{k} \times 3}
\]

\[
p^{j+1} = Sp^j
\]

• $S$ depends on $k$
Subdivision Matrix

• Assume $S$ has a full set of eigenvectors $\{\varphi_i\}$ with corresponding real eigenvalues $\{\lambda_i\}$, arranged in non-increasing order.

• We can express $p^0$ in terms of the eigenvectors of $S$:

$$ (p^0)_{k \times 3} = \sum_i (\varphi_i)_{k \times 1} (a_i)_{1 \times 3} $$

• Where:

$$ a_i = \varphi_i^T p^0 $$
Subdivision Matrix

• Now we can express $p^j$ using only $a_i$ and the eigenvalues and eigenvectors of $S$:

$$p^j = S^j \sum_i \varphi_i a_i$$

$$= \sum_i S^j \varphi_i a_i$$

$$= \sum_i (\lambda_i)^j \varphi_i a_i$$

• We used the fact: $S^j \varphi_i = (\lambda_i)^j \varphi_i$
Subdivision Properties

- **Convergence:** We need $|\lambda_i| \leq 1$ for all $i$ (and only one eigenvalue can be exactly 1)

- **Affine invariance:** If for all $A, B$:

$$S \cdot (p^j \cdot A_{3x3} + 1_{kx1} \cdot B_{1x3}) = S \cdot (p^j) \cdot A_{3x3} + 1_{kx1} \cdot B_{1x3}$$

$$S \cdot (p^j) \cdot A + S \cdot 1 \cdot B = S \cdot (p^j) \cdot A + 1 \cdot B$$

For affine invariance we need $S \cdot 1 = 1$

$\rightarrow \lambda_0 = 1$
Subdivision Properties

• **Limit position:**

\[ p^\infty = \lim_{{j \to \infty}} p^j = \lim_{{j \to \infty}} \sum_{{i}} (\lambda_i)^j \varphi_i a_i \]

Since \(|\lambda_i| < 1\) for all \(i > 0\), we have:

\[ p^\infty = (\lambda_0)^j \varphi_0 a_0 = 1 \cdot a_0 \]

In the limit the 2-neighborhood of \(v\) is mapped to the same position: \(a_0\)

We can compute the limit positions without recursively applying the subdivision
Subdivision Properties

- **Limit tangent plane:**

  Since scheme is translation invariant, fix \( a_0 = 0 \)

  Assume \( \lambda = \lambda_1 = \lambda_2 > \lambda_3 \)

  \[
  \frac{p^j}{\lambda^j} = \varphi_1a_1 + \varphi_2a_2 + \sum_{i \geq 3} \left(\frac{\lambda_i}{\lambda}\right)^j \varphi_i a_i
  \]

  For large enough \( j \), the 2-neighborhood of \( v \) is mapped to linear combinations of \( a_1 \) and \( a_2 \)

  \( a_1 \) and \( a_2 \) span the tangent plane of the limit surface at \( v \)
Piecewise Smooth Surfaces

• So far, only considered closed smooth surfaces
• Surfaces have boundaries and creases
• A subdivision scheme should have rules for all the following cases:

- Interior
- Smooth boundary
- Convex boundary
- Concave boundary
- Crease
Subdivision Zoo

• Can be classified according to:

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

• Many more...
Terminology

• *Regular* vertices
  – Tri meshes
    • In the interior - degree 6
    • On the boundary – degree 4
  – Quad meshes
    • In the interior - degree 4
    • On the boundary – degree 3

• *Extraordinary* vertices – all the rest
Terminology

- **Odd** vertices – new vertices at current subdivision level
- **Even** vertices – vertices inherited from previous level
- **Face** vertices – odd vertices inserted in a face
- **Edge** vertices – odd vertices inserted on an edge
Boundaries and Creases

• Special subdivision rules will be given for each scheme for the boundary vertices

• The boundary curve of the limit surface:
  – Should not depend on interior control vertices
    • In case two surfaces will be merged along the boundary
  – Should be $C^1$ or $C^2$

• Use boundary rules for edges tagged as creases
Loop Scheme

• Possible choices for $\beta$
  
  – Original (by Loop):
    
    $$\beta = \frac{1}{k} \left( \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{k} \right)^2 \right)$$

  – Or (by Warren):
    
    $$\beta = \begin{cases} 
    \frac{3}{16} & n = 3 \\
    \frac{3}{8n} & n > 3 
    \end{cases}$$
Loop Scheme

• Limits for interior vertex \( v \)
  (using control points from \( \text{any} \) subd level \( j \))
  
  – Position

  \[ p^\infty = S\chi p^j \quad ; \quad \chi = \frac{1}{3/8\beta + k} \]

  – Tangents

  \[
  t_1 = \sum_{i=0}^{k-1} \cos \frac{2\pi i}{k} p_i^j \quad ; \quad t_2 = \sum_{i=0}^{k-1} \sin \frac{2\pi i}{k} p_i^j
  \]

  Where \( v_i^j \) for \( i = \{0,..,k-1\} \) are the one ring neighbors of the vertex \( v \) at subd level \( j \), and \( p_i^j \) are the corresponding control points
Modified Butterfly Scheme

- Interpolating scheme
  - Even vertices don’t move

Mask for interior odd vertices with regular neighbors

Mask for crease and boundary vertices

a. Masks for odd vertices

b. Mask for odd vertices adjacent to an extraordinary vertex
Catmull-Clark Quad Scheme

Mask for a face vertex

\[ \begin{array}{|c|c|c|c|}
\hline
\frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} \\
\hline
\end{array} \]

Mask for an edge vertex

\[ \begin{array}{|c|c|c|}
\hline
\frac{1}{16} & \frac{1}{16} \\
\frac{3}{8} & \frac{3}{8} \\
\frac{1}{16} & \frac{1}{16} \\
\hline
\end{array} \]

Mask for a boundary odd vertex

\[ \begin{array}{|c|c|}
\hline
\frac{1}{2} & \frac{1}{2} \\
\hline
\end{array} \]

Interior

\[ \begin{array}{|c|c|c|c|}
\hline
\beta & \frac{\gamma}{k} \\
\frac{\beta}{k} & \frac{\gamma}{k} \\
\frac{1-\beta-\gamma}{k} & \frac{\beta}{k} \\
\frac{\beta}{k} & \frac{\gamma}{k} \\
\hline
\end{array} \]

\[ \beta = \frac{3}{2k}; \gamma = \frac{1}{4k} \]

Crease and boundary

\[ \begin{array}{|c|c|c|}
\hline
\frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\
\hline
\end{array} \]

Can be modified to work on general polygons
Kobbelt Scheme

Main observation - to compute a face control point:

- compute all *edge control points*
- compute *face control points* using the *edge rule* applied to edge control points on same level
Scheme Comparison

- Subdividing a cube
  - Loop result is asymmetric, because cube was triangulated first
  - Both Loop and Catmull-Clark are better than Butterfly ($C^2$ vs. $C^1$)
  - Interpolation vs. smoothness
Scheme Comparison

- Subdividing a tetrahedron
  - Same insights
  - Severe shrinking for approximating schemes
Scheme Comparison

- Spot the difference?
- For smooth meshes with uniform triangle size, different schemes provide very similar results
- Beware of interpolating schemes for control polygons with sharp features

Loop  Butterfly  Catmull-Clark
So Who Wins?

• Loop and Catmull-Clark best when interpolation is not required
• Loop best for triangular meshes
• Catmull-Clark best for quad meshes
  – Don’t triangulate and then use Catmull-Clark
The Dark Side of Subd Surfaces

• Problems with curvature continuity
  – Requires either very large support, or forces 0 curvature at extraordinary vertices
  – Generates ripples near vertices of large degree
The Dark Side of Subd Surfaces

• Decreased smoothness with degree
  – For large degrees, third eigenvalue approaches second and first eigenvalues
  – Generates creases

• Can fix by modifying scheme
  – Creates more ripples