Subdivision Surfaces

Adam Finkelstein
Princeton University
COS 426, Spring 2005

Modeling

- How do we ...
  - Represent 3D objects in a computer?
  - Construct 3D representations quickly/easily?
  - Manipulate 3D representations efficiently?

Different representations for different types of objects

3D Object Representations

- Raw data
  - Voxels
  - Point cloud
  - Range image
  - Polymers

- Surfaces
  - Mesh
  - Subdivision
  - Parametric
  - Implicit

- High-level structures
  - Scene graph
  - Skeleton
  - Application specific

- Solids
  - Octree
  - BSP tree
  - CSG
  - Sweep

- Surfaces
  - Mesh
  - Subdivision
  - Parametric
  - Implicit

- High-level structures
  - Scene graph
  - Skeleton
  - Application specific

Surfaces

- What makes a good surface representation?
  - Accurate
  - Concise
  - Intuitive specification
  - Local support
  - Affine invariant
  - Arbitrary topology
  - Guaranteed continuity
  - Natural parameterization
  - Efficient display
  - Efficient intersections

H&B Figure 10.46
Subdivision

• How do you make a smooth curve?

Subdivision Surfaces

• Coarse mesh & subdivision rule
  - Define smooth surface as limit of sequence of refinements

Key Questions

• How refine mesh?
  - Aim for properties like smoothness

• How store mesh?
  - Aim for efficiency for implementing subdivision rules

Loop Subdivision Scheme

• How refine mesh?
  - Refine each triangle into 4 triangles by splitting each edge and connecting new vertices
  - Need rules for "even / odd" (white / black) vertices

• How position new vertices?
  - Choose locations for new vertices as weighted average of original vertices in local neighborhood

• Rules for extraordinary vertices and boundaries:

  a. Masks for odd vertices
  b. Masks for even vertices

What if odd vertex only touches one triangle?
What if even vertex does not have degree 6?
**Loop**

- How to choose $\beta$?
  - Analyze properties of limit surface
  - Interested in continuity of surface and smoothness
  - Involves calculating eigenvalues of matrices

  » Original Loop
  $\beta = \frac{1}{n} \left( \frac{3}{n} - \left( \frac{3}{n} + \frac{1}{2} \cos \frac{3\pi}{n} \right)^2 \right)$

  » Warren
  $\beta = \begin{cases} 
  \frac{1}{8n} & n > 3 \\
  \frac{1}{16} & n = 3 
  \end{cases}$

**Subdivision Schemes**

- There are different subdivision schemes
  - Different methods for refining topology
  - Different rules for positioning vertices
    - Interpolating versus approximating

<table>
<thead>
<tr>
<th>Face split for triangles</th>
<th>Face split for quads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face split</td>
<td>Face split</td>
</tr>
<tr>
<td>Face split</td>
<td>Face split</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Triangular mesh</th>
<th>Quad mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approximating</td>
<td>Loop $n^2$</td>
</tr>
<tr>
<td>Interpolating</td>
<td>Mod. butterfly $n^3$</td>
</tr>
</tbody>
</table>

**Subdivision Surfaces**

- Properties:
  - Accurate
  - Concise
  - Intuitive specification
  - Local support
  - Affine invariant
  - Arbitrary topology
  - Guaranteed continuity
  - Natural parameterization
  - Efficient display
  - Efficient intersections
Key Questions

- How refine mesh?
  - Aim for properties like smoothness
- How store mesh?
  - Aim for efficiency for implementing subdivision rules

Polygon Meshes

- Mesh Representations
  - Independent faces
  - Vertex and face tables
  - Adjacency lists
  - Winged-Edge

Independent Faces

- Each face lists vertex coordinates
  - Redundant vertices
  - No topology information

Vertex and Face Tables

- Each face lists vertex references
  - Shared vertices
  - Still no topology information

Adjacency Lists

- Store all vertex, edge, and face adjacencies
  - Efficient topology traversal
  - Extra storage

Partial Adjacency Lists

- Can we store only some adjacency relationships and derive others?
Winged Edge

- Adjacency encoded in edges
  - All adjacencies in $O(1)$ time
  - Little extra storage (fixed records)
  - Arbitrary polygons

![Winged Edge Diagram]

Triangle Meshes

- Relevant properties:
  - Exactly 3 vertices per face
  - Any number of faces per vertex

- Useful adjacency structure for Loop subdivision:
  - Do not represent edges explicitly
  - Faces store refs to vertices and neighboring faces
  - Vertices store refs to adjacent faces and vertices

![Triangle Meshes Diagram]

Assignment 3

- Interactive editing of subdivision surfaces
  - Loop subdivision scheme
  - Partial adjacency list mesh representation
  - Interactive vertex dragging

![Assignment 3 Diagram 1]

- Store hierarchy of meshes
  - Full triangle mesh at every level
  - Vertices store references to counterparts one level up and one level down
  - Enables efficient re-positioning of mesh vertices after interactive dragging

![Assignment 3 Diagram 2]
Summary

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

• Difficulties:
  o Intuitive specification
  o Parameterization
  o Intersections