1

# Subdivision Surfaces

Adam Finkelstein
Princeton University
COS 426, Fall 2001

2

## Course Syllabus

I. Image processing
II. Rendering
III. Modeling
IV. Animation

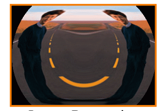
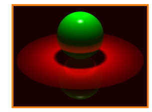
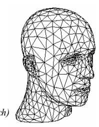
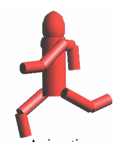


Image Processing
*(Rusty Coleman, CS426, Fall99)*


Rendering
*(Michael Bostock, CS426, Fall99)*


Modeling
*(Dennis Zorin, CalTech)*


Animation
*(Angel, Plate 1)*

3

## Course Syllabus

I. Image processing
II. Rendering
**III. Modeling**
IV. Animation

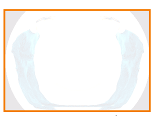
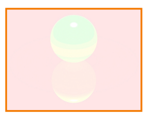
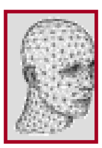
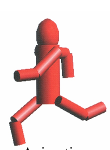


Image Processing
*(Rusty Coleman, CS426, Fall99)*


Rendering
*(Michael Bostock, CS426, Fall99)*

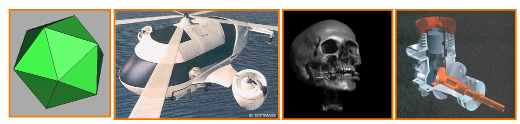

**Modeling**
***(Dennis Zorin, CalTech)***


Animation
*(Angel, Plate 1)*

4

## 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

5

## 3D Object Representations

- Raw data
  - Voxels
  - Point cloud
  - Range image
  - Polygons
- Surfaces
  - Mesh
  - Subdivision
  - Parametric
  - Implicit
- Solids
  - Octree
  - BSP tree
  - CSG
  - Sweep
- High-level structures
  - Scene graph
  - Skeleton
  - Application specific

6

## Equivalence of Representations

- Thesis:
  - Each fundamental representation has enough expressive power to model the shape of any geometric object
  - It is possible to perform all geometric operations with any fundamental representation!
- Analogous to Turing-Equivalence:
  - All computers today are turing-equivalent, but we still have many different processors

## Computational Differences

7

- Efficiency
  - Combinatorial complexity (e.g. O( n log n ) )
  - Space/time trade-offs (e.g. z-buffer)
  - Numerical accuracy/stability (degree of polynomial)
- Simplicity
  - Ease of acquisition
  - Hardware acceleration
  - Software creation and maintenance
- Usability
  - Designer interface vs. computational engine

## 3D Object Representations

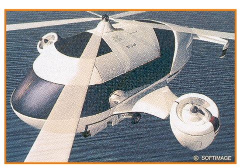
8

- Raw data
  - Voxels
  - Point cloud
  - Range image
  - Polygons
- Solids
  - Octree
  - BSP tree
  - CSG
  - Sweep
- **Surfaces**
  - **Mesh**
  - **Subdivision**
  - Parametric
  - Implicit
- High-level structures
  - Scene graph
  - Skeleton
  - Application specific

## Surfaces

9

- 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

10

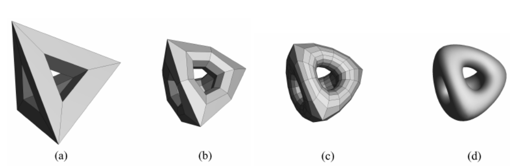
- How do you make a smooth curve?

Zorin & Schroeder
SIGGRAPH 99
Course Notes

## Subdivision Surfaces

11

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

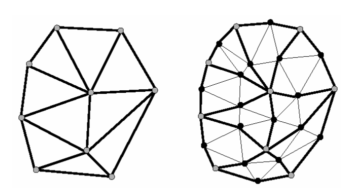



Zorin & Schroeder
SIGGRAPH 99
Course Notes

## Key Questions

12

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

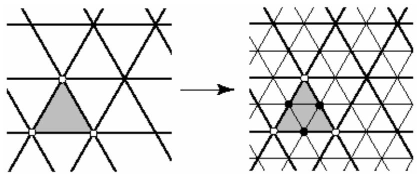


Zorin & Schroeder
SIGGRAPH 99
Course Notes

13

## Loop Subdivision Scheme

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

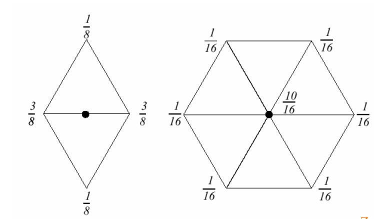


Zorin & Schroeder
SIGGRAPH 99
Course Notes

14

## Loop Subdivision Scheme

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



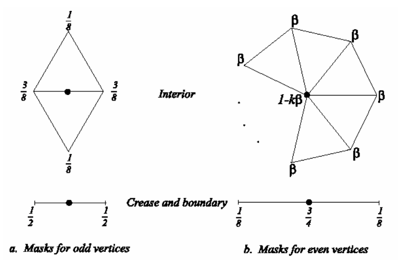

What if vertex does not have degree 6?

Zorin & Schroeder
SIGGRAPH 99
Course Notes

15

## Loop Subdivision Scheme

- Rules for *extraordinary vertices* and *boundaries*:




Zorin & Schroeder
SIGGRAPH 99
Course Notes

16

## Loop

- How to choose β?
  - Analyze properties of limit surface
  - Interested in continuity of surface and smoothness
  - Involves calculating eigenvalues of matrices
    - » Original Loop

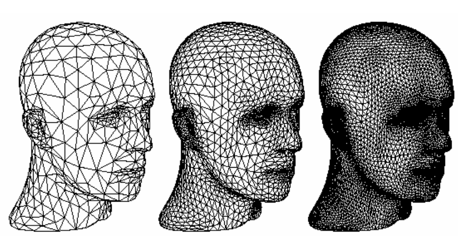
$$\beta = \frac{1}{n}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n}\right)^2\right)$$

    - » Warren

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

17

## Loop Subdivision Scheme



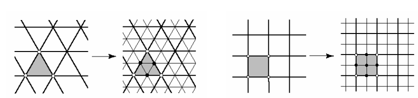
Limit surface has provable smoothness properties!

Zorin & Schroeder
SIGGRAPH 99
Course Notes

18

## Subdivision Schemes

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



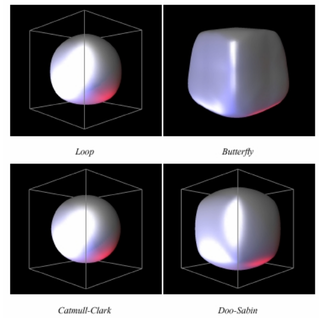

| Face split | | |
|---|---|---|
| | *Triangular meshes* | *Quad. meshes* |
| *Approximating* | Loop ($C^2$) | Catmull-Clark ($C^2$) |
| *Interpolating* | Mod. Butterfly ($C^1$) | Kobbelt ($C^1$) |

| Vertex split |
|---|
| Doo-Sabin, Midedge ($C^1$) |
| Biquartic ($C^2$) |

Zorin & Schroeder, SIGGRAPH 99 , Course Notes

## Subdivision Schemes

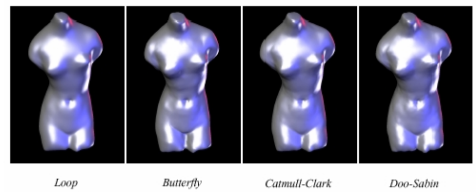
19




Zorin & Schroeder
SIGGRAPH 99
Course Notes

## Subdivision Schemes

20

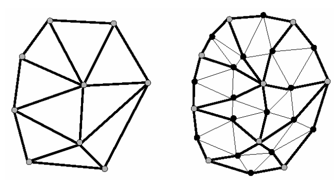



Zorin & Schroeder
SIGGRAPH 99
Course Notes

## Key Questions

21

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

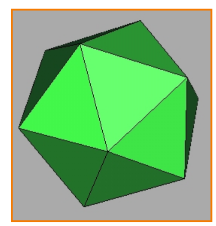


Zorin & Schroeder
SIGGRAPH 99
Course Notes

## Polygon Meshes

22

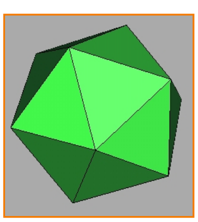
- Mesh Representations
  - Independent faces
  - Vertex and face tables
  - Adjacency lists
  - Winged-Edge

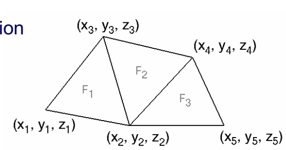


## Independent Faces

23

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



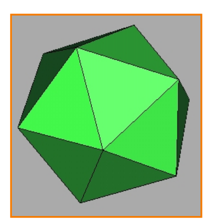


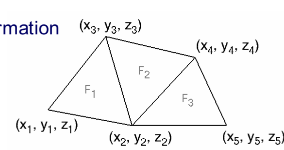

| FACE TABLE | |
|---|---|
| $F_1$ | $(x_1, y_1, z_1)$ $(x_2, y_2, z_2)$ $(x_3, y_3, z_3)$ |
| $F_2$ | $(x_2, y_2, z_2)$ $(x_4, y_4, z_4)$ $(x_3, y_3, z_3)$ |
| $F_3$ | $(x_2, y_2, z_2)$ $(x_5, y_5, z_5)$ $(x_4, y_4, z_4)$ |

## Vertex and Face Tables

24

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



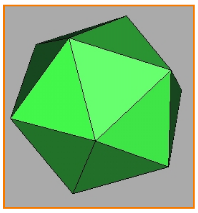


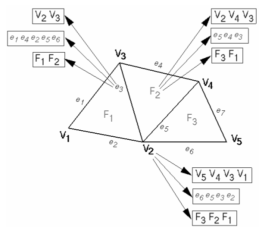

| VERTEX TABLE | | | |
|---|---|---|---|
| $V_1$ | $X_1$ | $Y_1$ | $Z_1$ |
| $V_2$ | $X_2$ | $Y_2$ | $Z_2$ |
| $V_3$ | $X_3$ | $Y_3$ | $Z_3$ |
| $V_4$ | $X_4$ | $Y_4$ | $Z_4$ |
| $V_5$ | $X_5$ | $Y_5$ | $Z_5$ |

| FACE TABLE | | | |
|---|---|---|---|
| $F_1$ | $V_1$ | $V_2$ | $V_3$ |
| $F_2$ | $V_2$ | $V_4$ | $V_3$ |
| $F_3$ | $V_2$ | $V_5$ | $V_4$ |

## Adjacency Lists
25

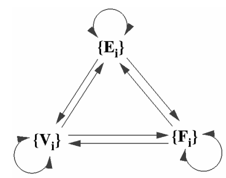
- Store all vertex, edge, and face adjacencies
  - Efficient topology traversal
  - Extra storage

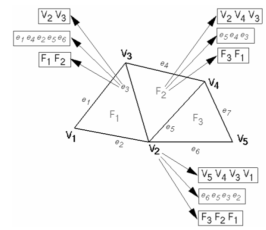





## Partial Adjacency Lists
26

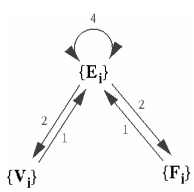
- Can we store only some adjacency relationships and derive others?

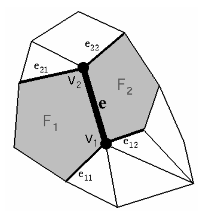



## Winged Edge
27

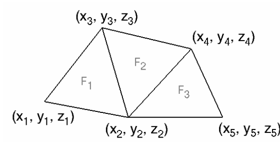
- Adjacency encoded in edges
  - All adjacencies in O(1) time
  - Little extra storage (fixed records)
  - Arbitrary polygons




## Winged Edge
28

- Example:




VERTEX TABLE

| | | | | |
|---|---|---|---|---|
| V1 | X1 | Y1 | Z1 | e1 |
| V2 | X2 | Y2 | Z2 | e6 |
| V3 | X3 | Y3 | Z3 | e3 |
| V4 | X4 | Y4 | Z4 | e5 |
| V5 | X5 | Y5 | Z5 | e6 |

EDGE TABLE

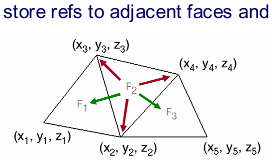
| | | | | | 11 | 12 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|
| e1 | V1 | V3 | | F1 | e2 | e2 | e4 | e3 |
| e2 | V1 | V2 | F1 | | e1 | e1 | e3 | e6 |
| e3 | V2 | V3 | F1 | F2 | e2 | e5 | e1 | e4 |
| e4 | V3 | V4 | | F2 | e1 | e3 | e7 | e5 |
| e5 | V2 | V4 | F2 | F3 | e3 | e6 | e4 | e7 |
| e6 | V2 | V5 | F3 | | e5 | e2 | e7 | e7 |
| e7 | V4 | V5 | | F3 | e4 | e5 | e6 | e6 |

FACE TABLE

| | |
|---|---|
| F1 | e1 |
| F2 | e3 |
| F3 | e5 |

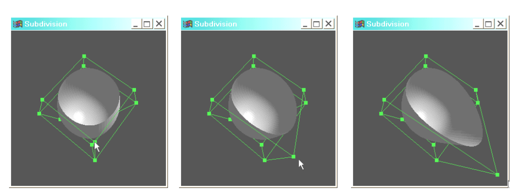
## Triangle Meshes
29

- 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




## Assignment 4
30

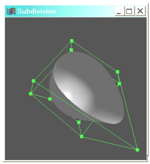
- Interactive editing of subdivision surfaces
  - Loop subdivision scheme
  - Partial adjacency list mesh representation
  - Interactive vertex dragging

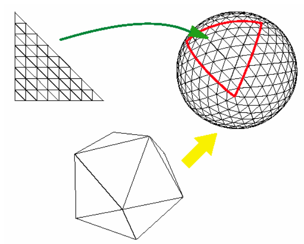


31

## Assignment 4

- Edit coarse mesh while display subdivided mesh

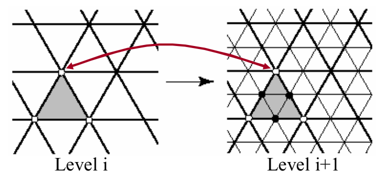




32

## Assignment 4

- 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




33

## Subdivision Surfaces

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

Pixar

34

## Summary

- Advantages:
  - Simple method for describing complex surfaces
  - Relatively easy to implement
  - Arbitrary topology
  - Local support
  - Guaranteed continuity
  - Multiresolution
- Difficulties:
  - Intuitive specification
  - Parameterization
  - Intersections