CS-C3100 Computer Graphics

4.2 Subdivision Surfaces

Representing Surfaces

- Triangle meshes ✔
  - Surface analogue of polylines, this is what GPUs draw
- Tensor Product Splines (previous video)
  - Surface analogue of spline curves
- **Subdivision surfaces (this video)**
- Implicit surfaces
  - f(x,y,z)=0
- Procedural
  - e.g. surfaces of revolution, generalized cylinder
- From volume data (medical images, etc.)

In This Video

- Subdivision for curves and surfaces
- Loop subdivision for triangle meshes

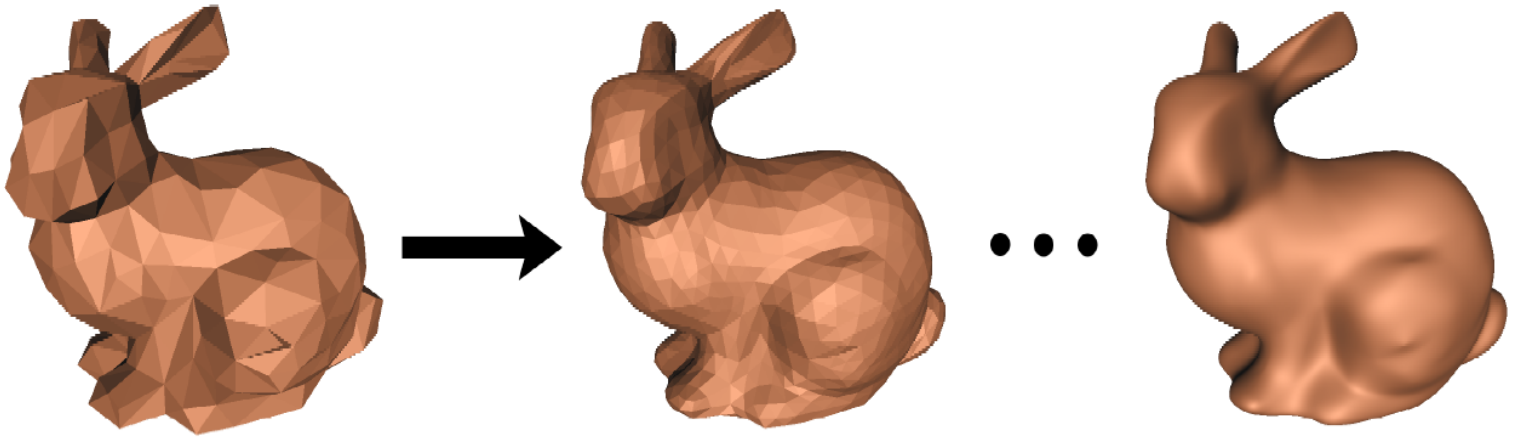
Issues with Spline Patches

- Regular connectivity only (square patches)
- Hard to ensure smoothness across boundaries

- Subdivision surfaces turn meshes into smooth surfaces through a recursive subdivision rule
  - Can use regular mesh modelling tools
  - Intuitive control
  - Smoothness built in (with some caveats)

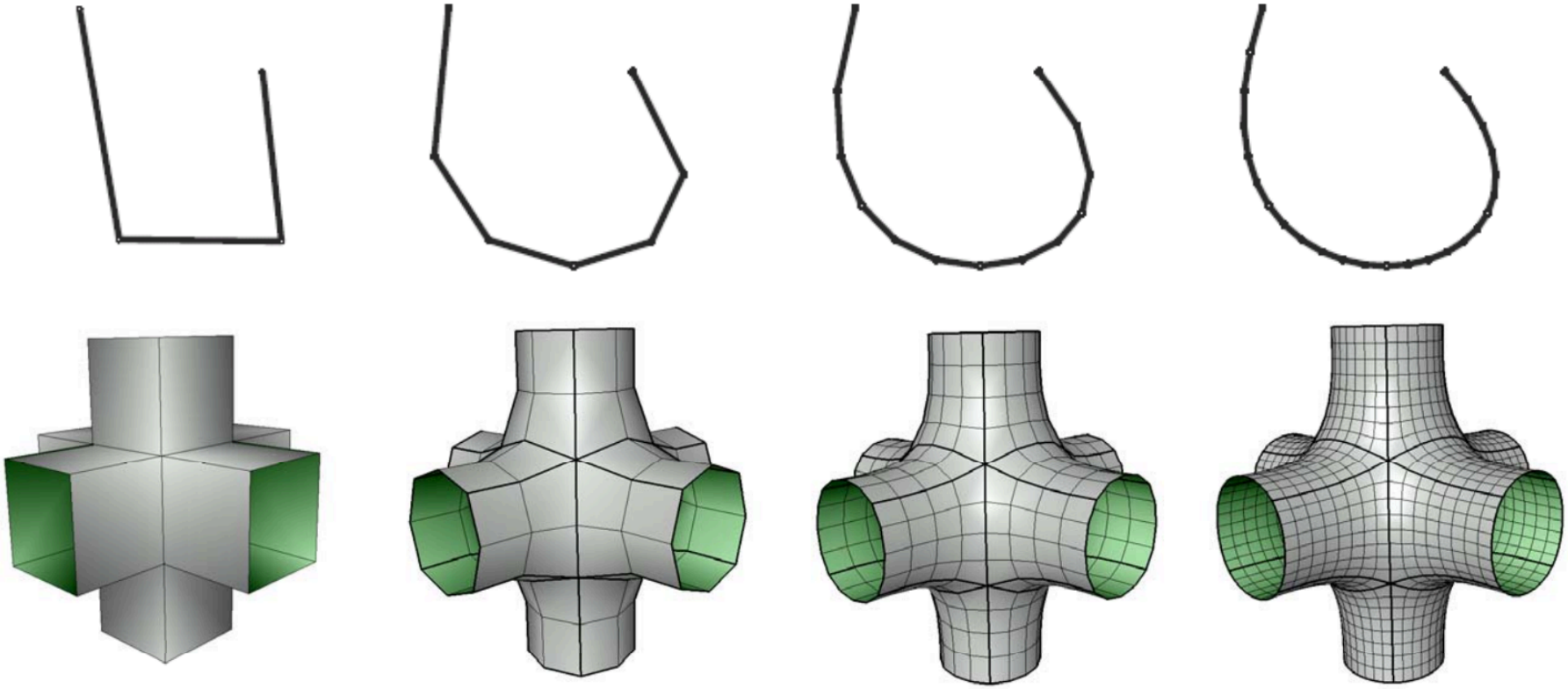
Subdivision Surfaces

- Start with polygonal mesh
- Subdivide into larger number of polygons, smooth result after each subdivision
  - Lots of ways to do this.
- The limit surface is smooth!

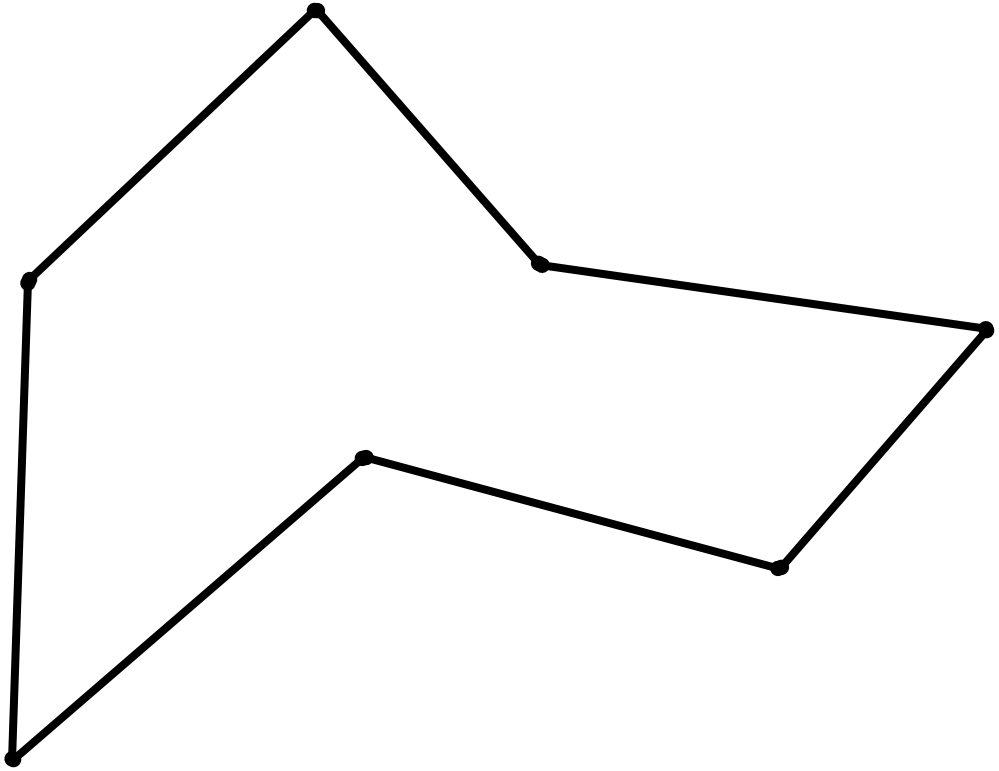


Subdividing curves vs. Surfaces

Image from Mirela Ben-Chen's Geometry Processing Algorithms class (Stanford)

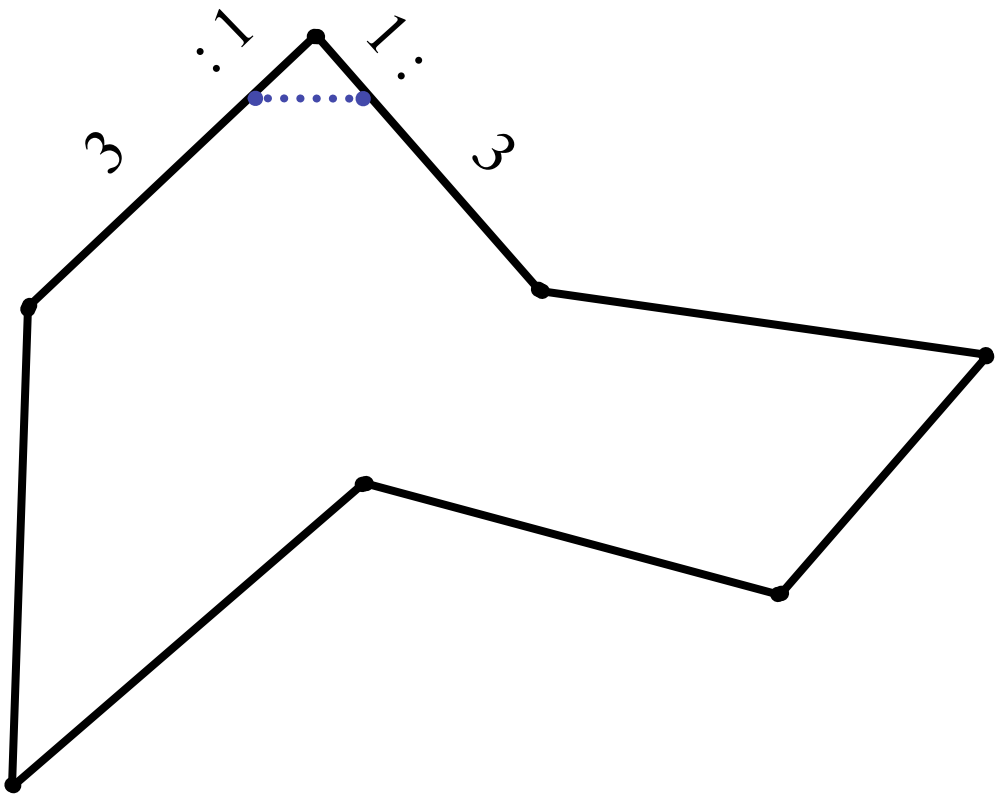


Corner Cutting



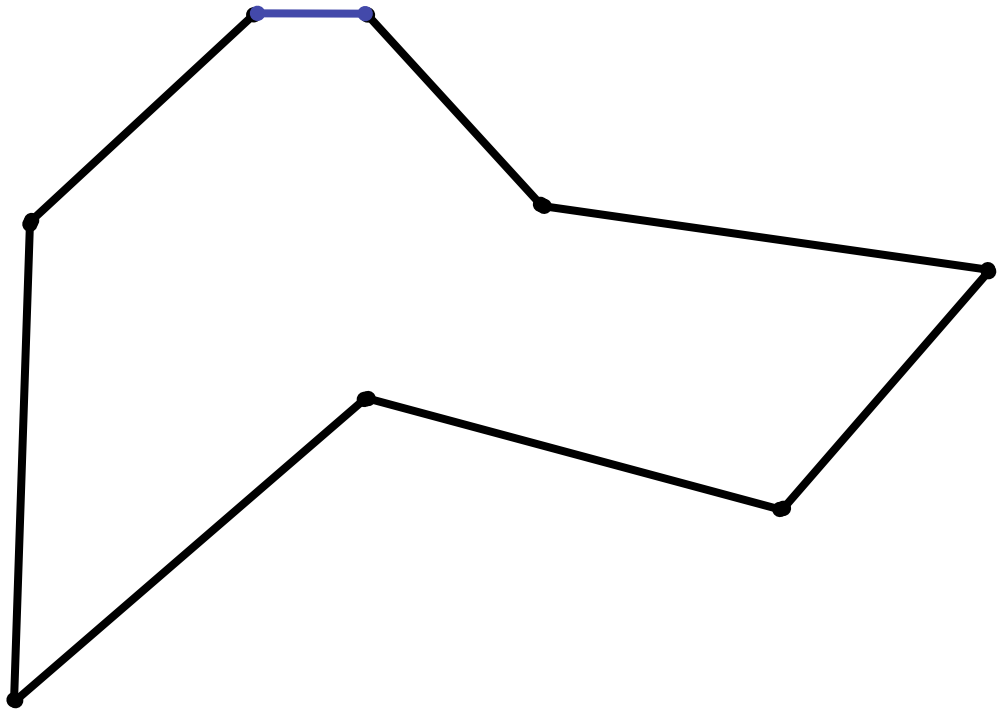
Slide by Adi Levin

Corner Cutting

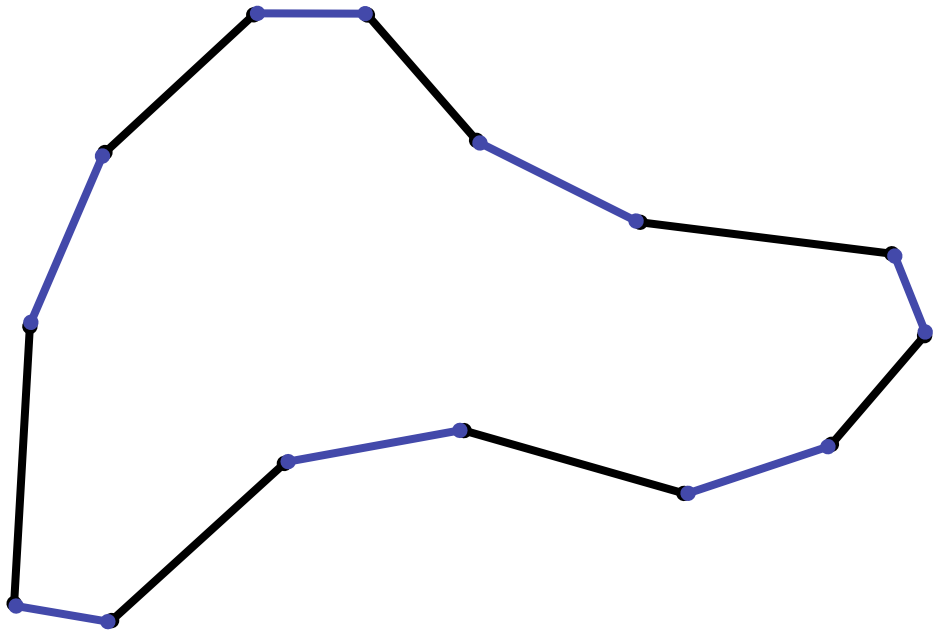


Slide by Adi Levin

Corner Cutting

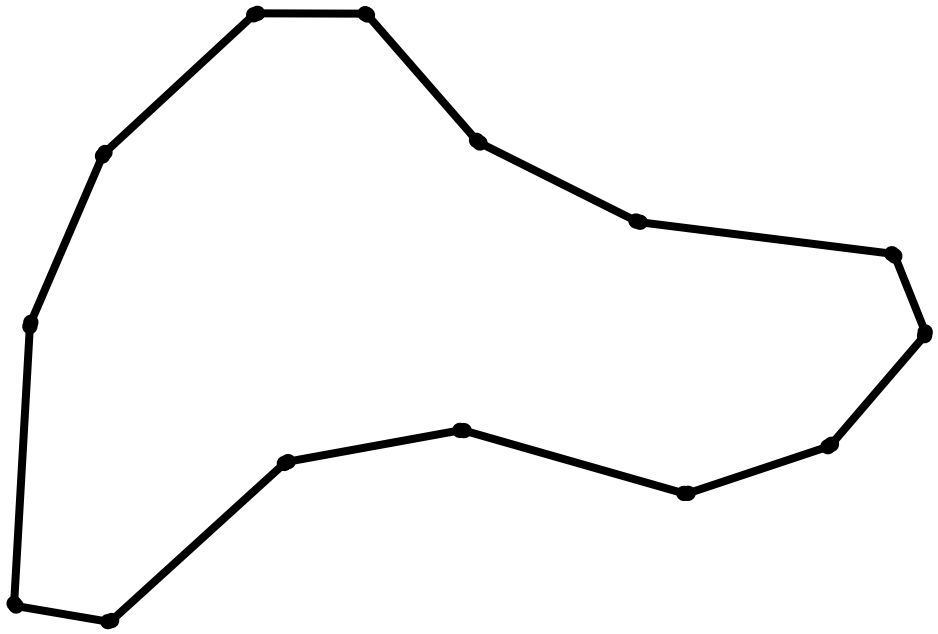


Corner Cutting



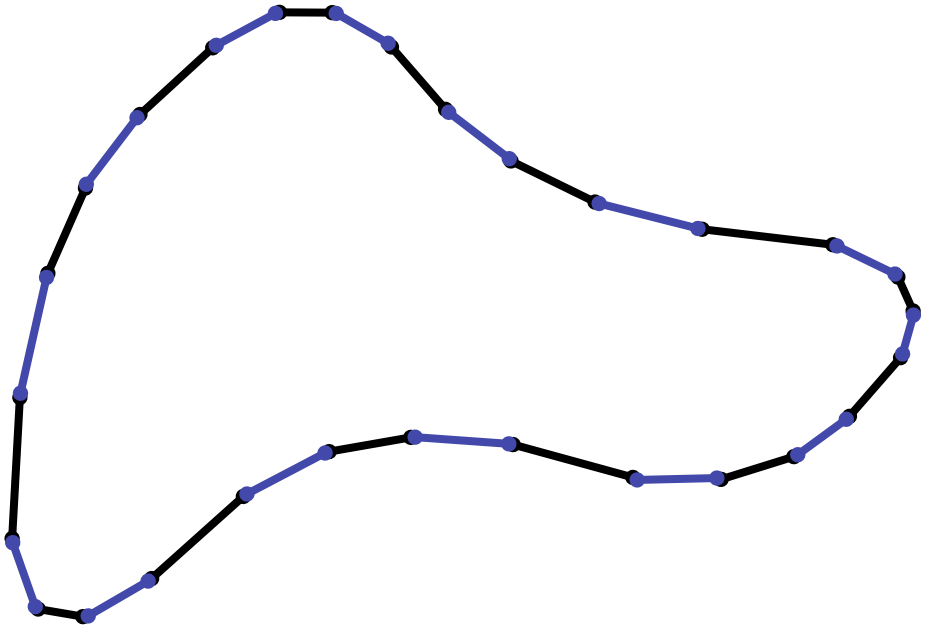
Slide by Adi Levin

Corner Cutting



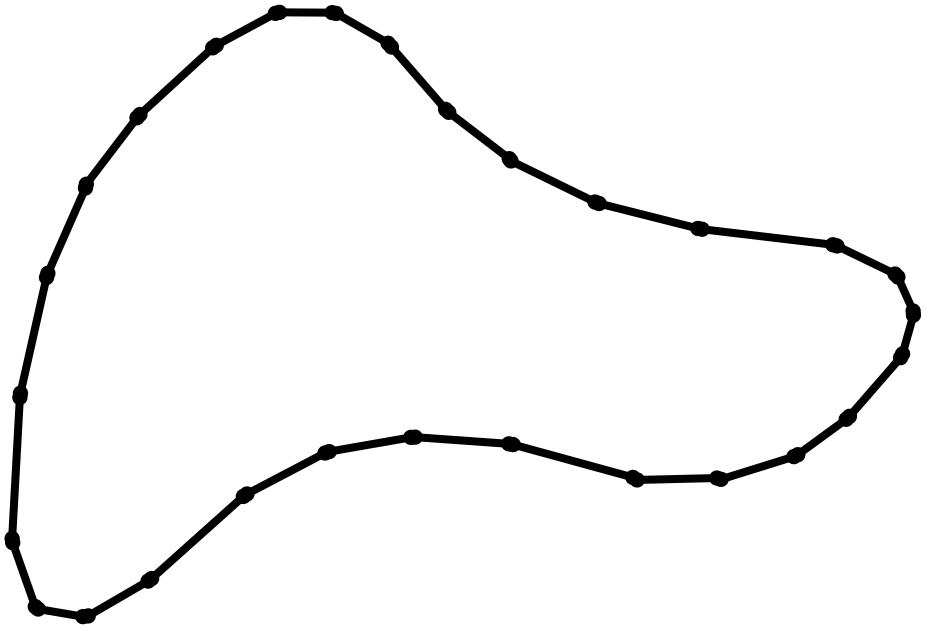
Slide by Adi Levin

Corner Cutting



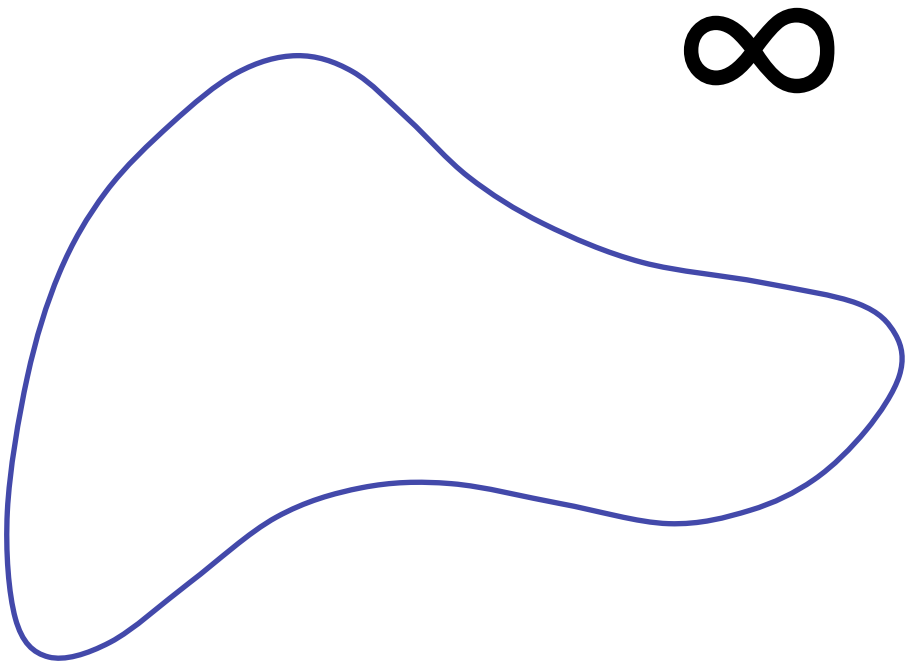
Slide by Adi Levin

Corner Cutting



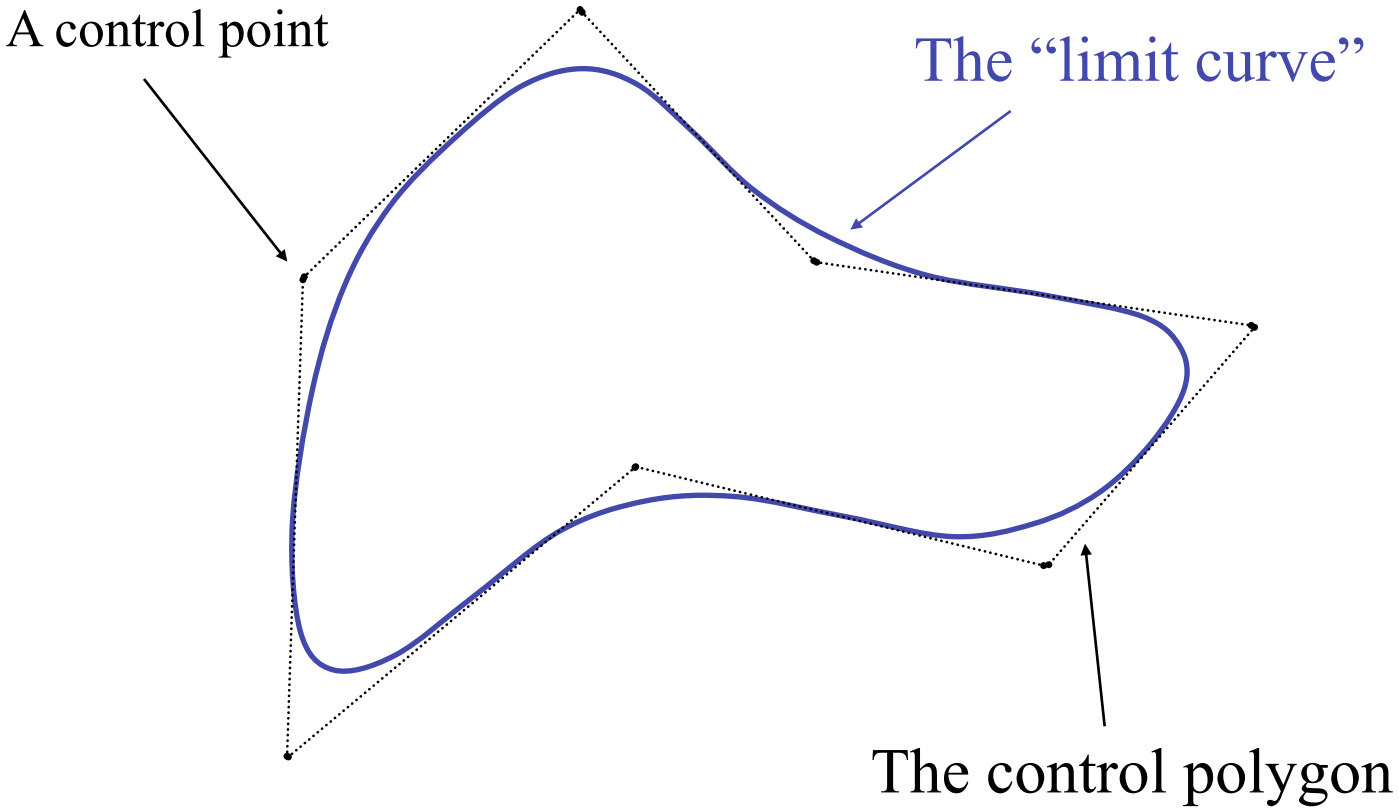
Slide by Adi Levin

Corner Cutting



Slide by Adi Levin

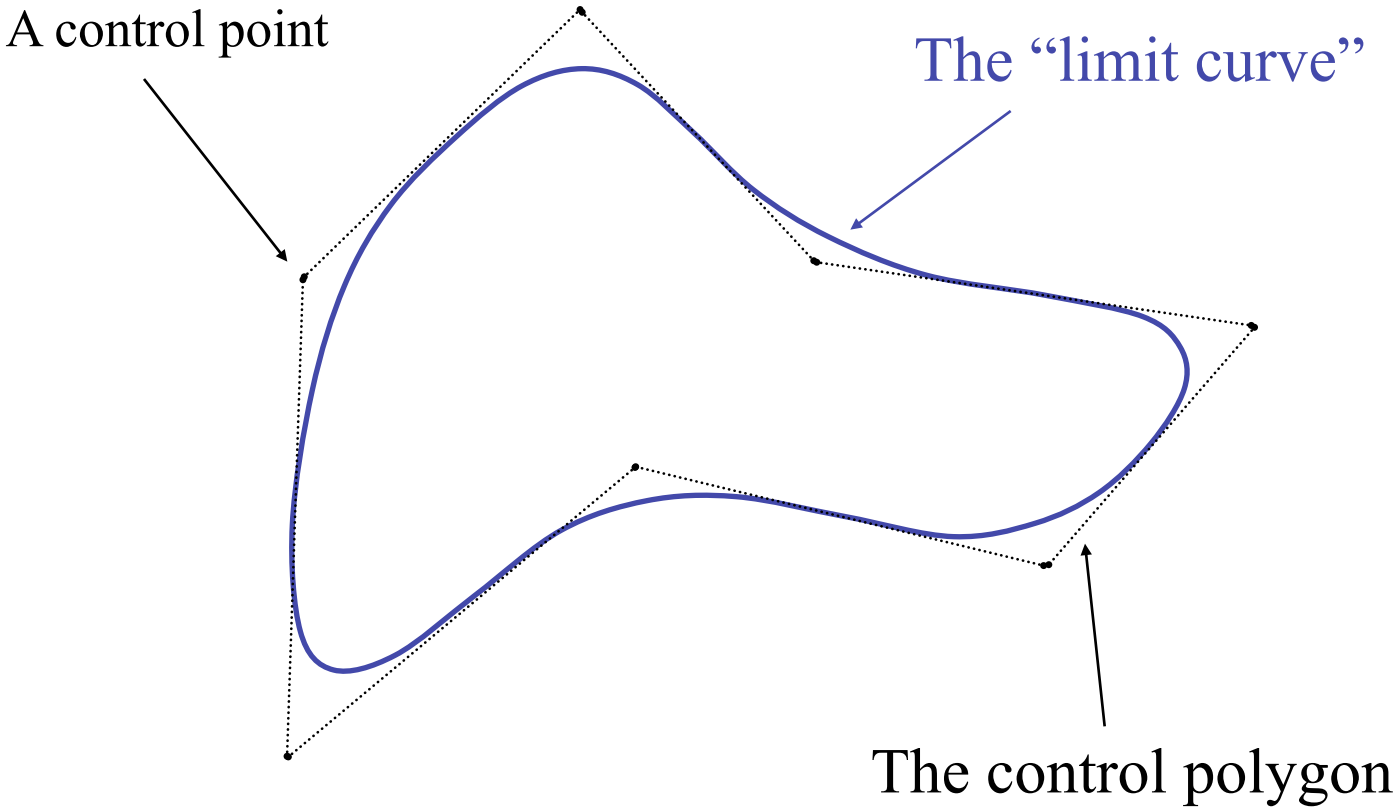
Corner Cutting



Slide by Adi Levin

Corner Cutting

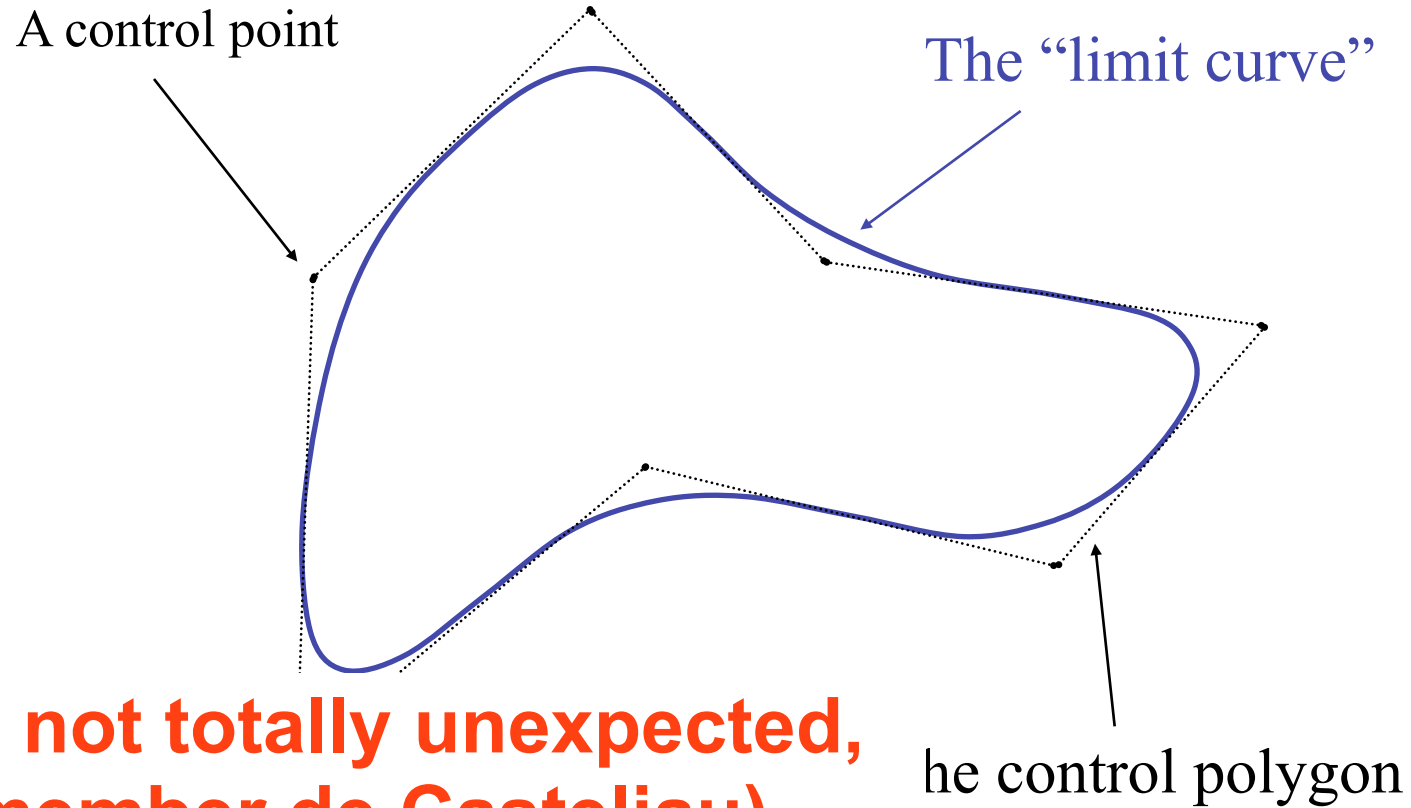
It turns out corner cutting (Chaikin's Algorithm) produces a quadratic B-Spline curve! (Magic!)



Slide by Adi Levin

Corner Cutting

It turns out corner cutting (Chaikin's Algorithm) produces a quadratic B-Spline curve! (Magic!)

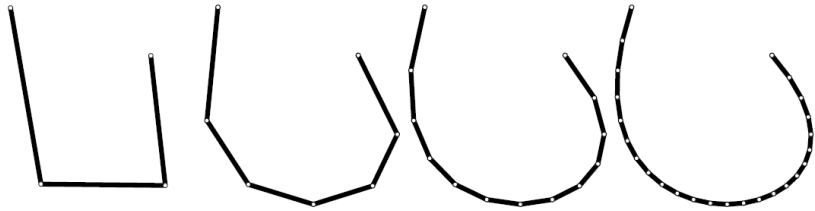


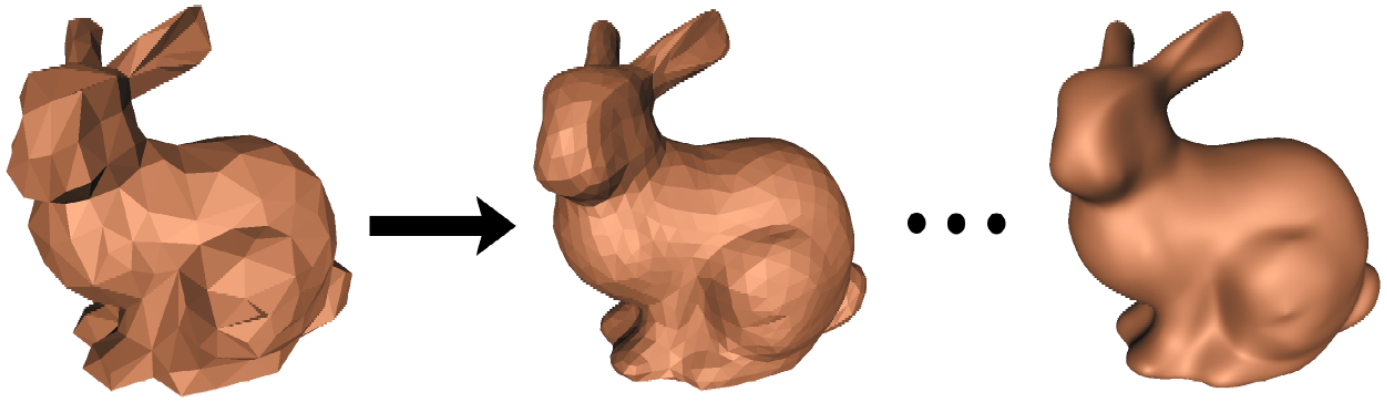
(Well, not totally unexpected, remember de Casteljau)

Slide by Adi Levin

Subdivision Curves and Surfaces

- Idea: cut corners to smooth
- Add points and compute weighted average of neighbors
- Same for surfaces
  - Special case for irregular vertices
    - vertex with more or less than 6 neighbors in a triangle mesh

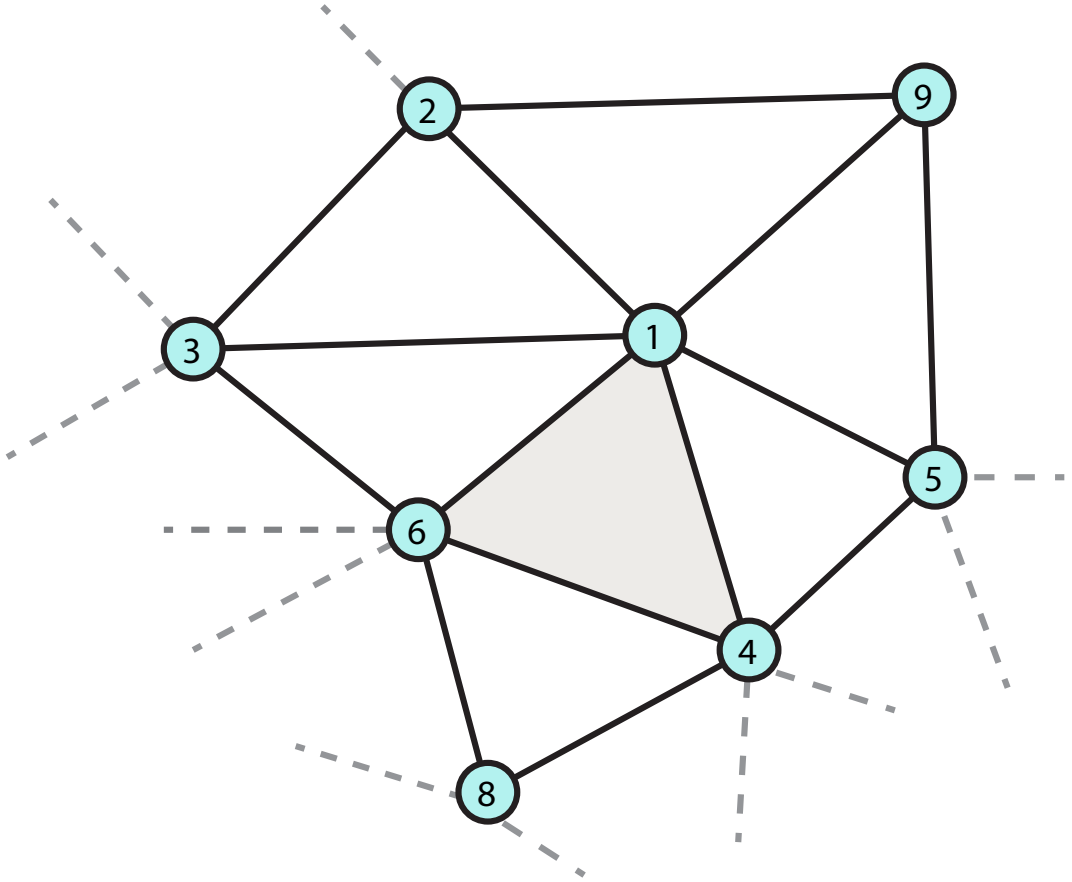




Warren et al.

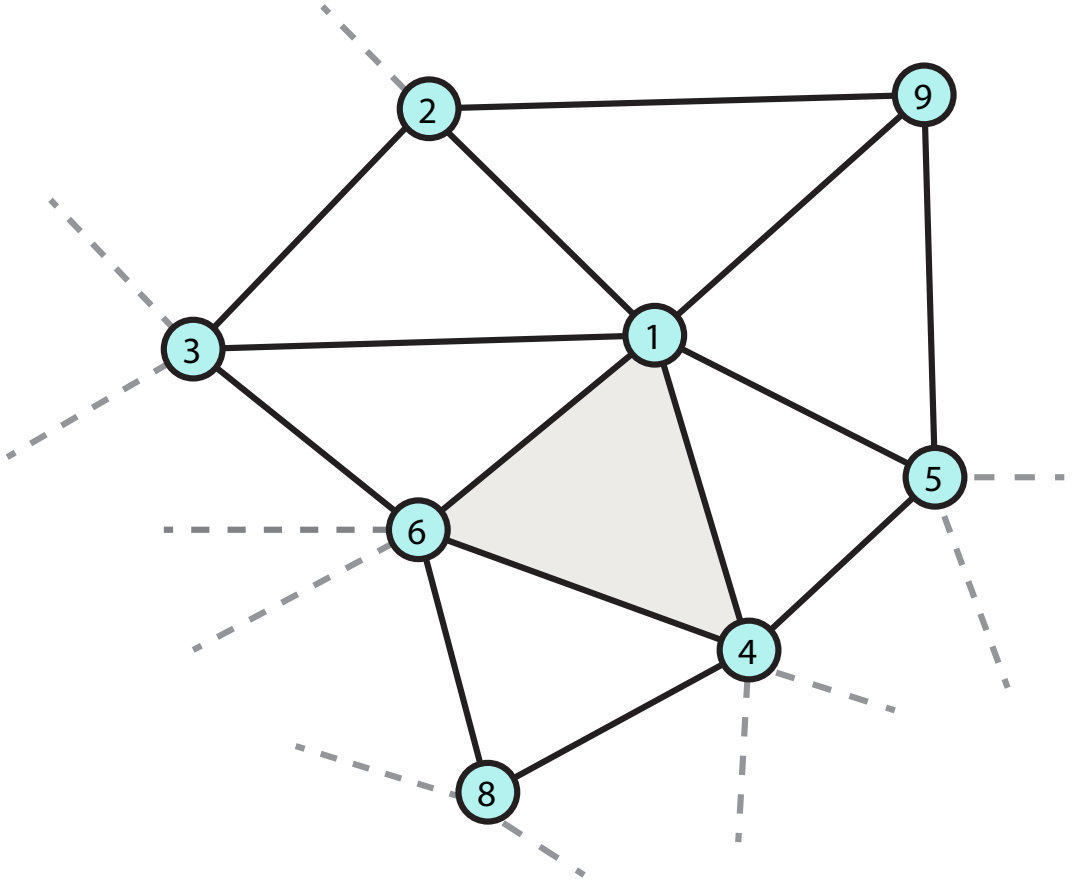
Assignment 2: Loop Subdivision

- (After Charles Loop, now with NVIDIA Research)



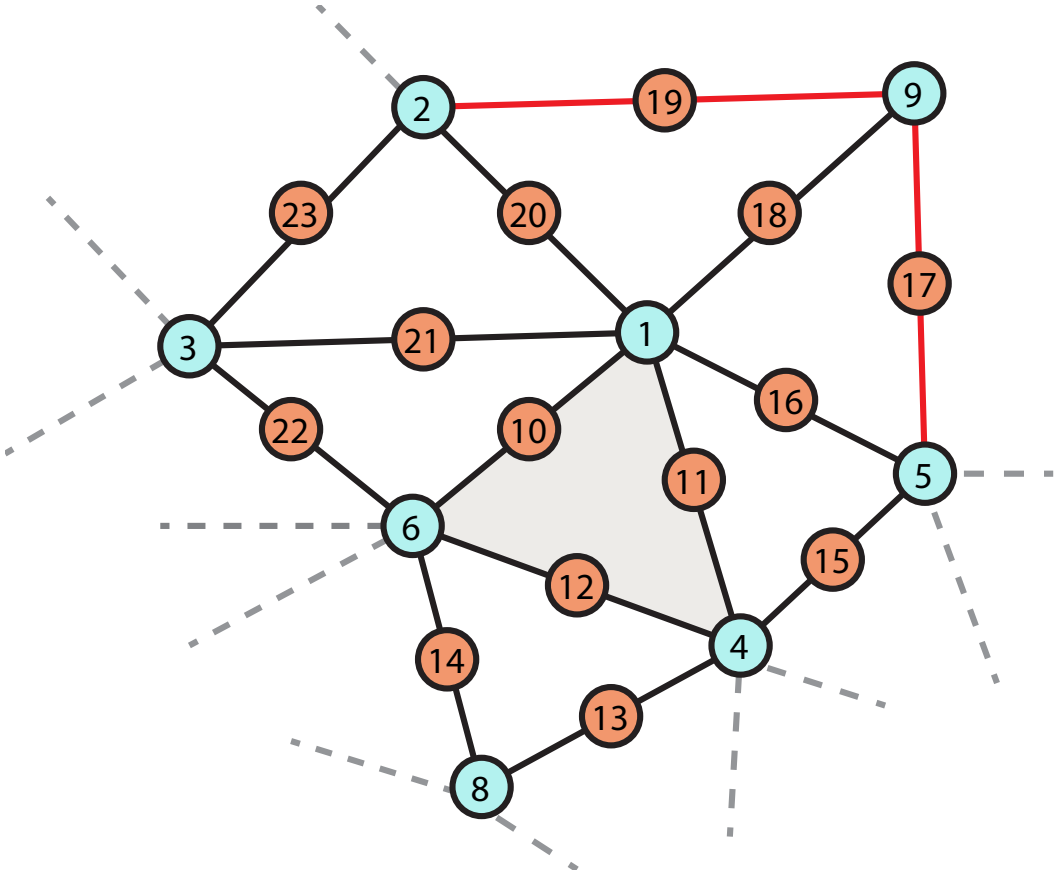
Assignment 2: Loop Subdivision

- The input mesh



Assignment 2: Loop Subdivision

- First step: each input edge gets a new vertex



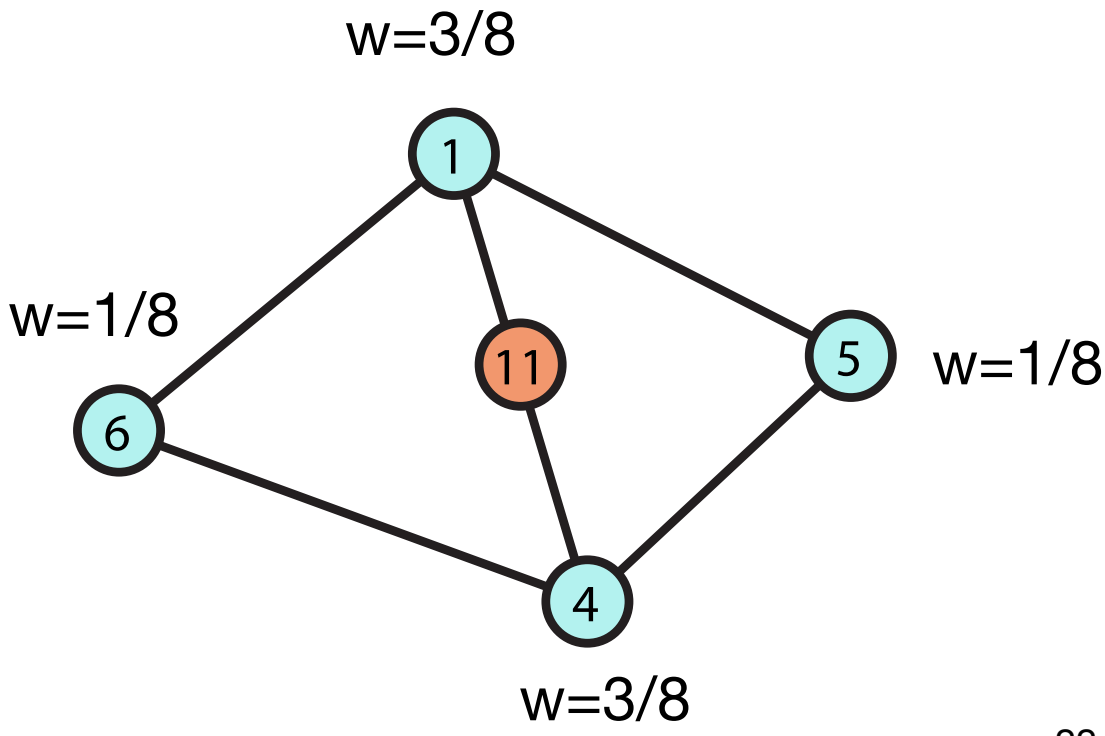
Vertex from input mesh

Newly-introduced vertex

Assignment 2: Loop Subdivision

- Then compute positions for new vertices
  - Weighted combinations of the connected *old* vertices
  - Need adjacency information (details in handout)

$$p_{11} = \frac{1}{8}\, p_5 + \frac{1}{8}\, p_6 + \frac{3}{8}\, p_1 + \frac{3}{8}\, p_4$$



Vertex from input mesh

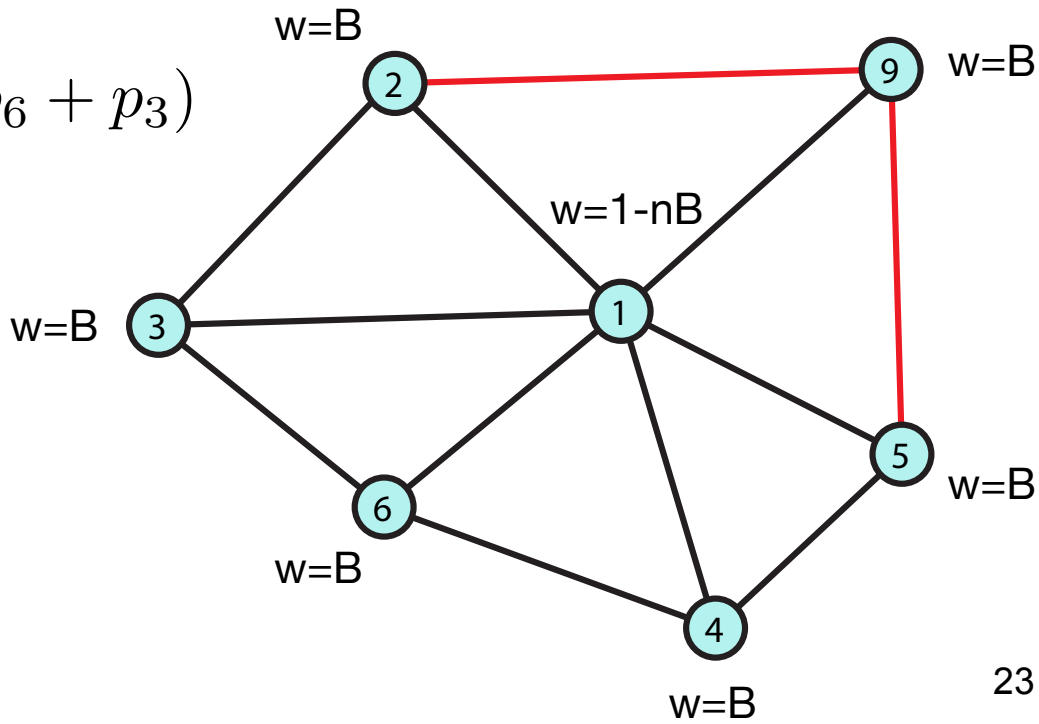
Newly-introduced vertex

Assignment 2: Loop Subdivision

- Then compute new positions for all *old* vertices!
  - Again weighted combinations of nearby *old* vertices
  - Need to find all n connected vertices

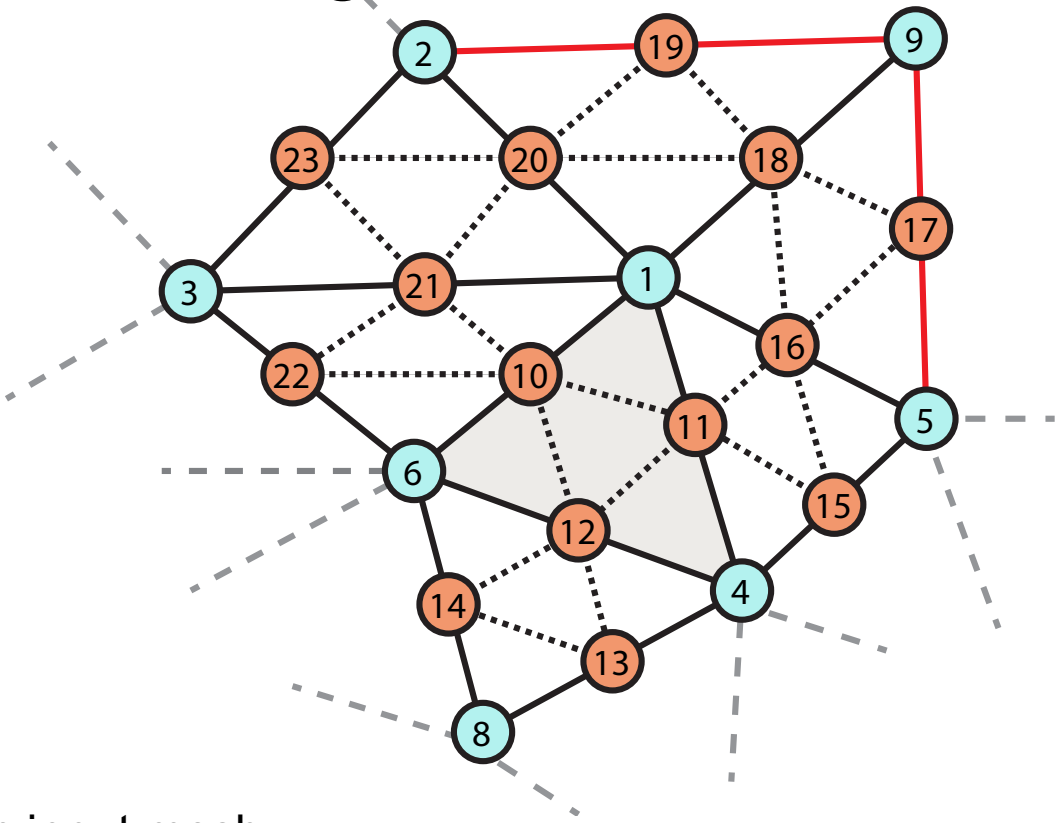
$$p_1' = (1 - nB)\, p_1 + B(p_2 + p_9 + p_5 + p_4 + p_6 + p_3)$$

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



Assignment 2: Loop Subdivision

- Finally, split all input triangles into four: remove old triangle, insert 4 new triangles

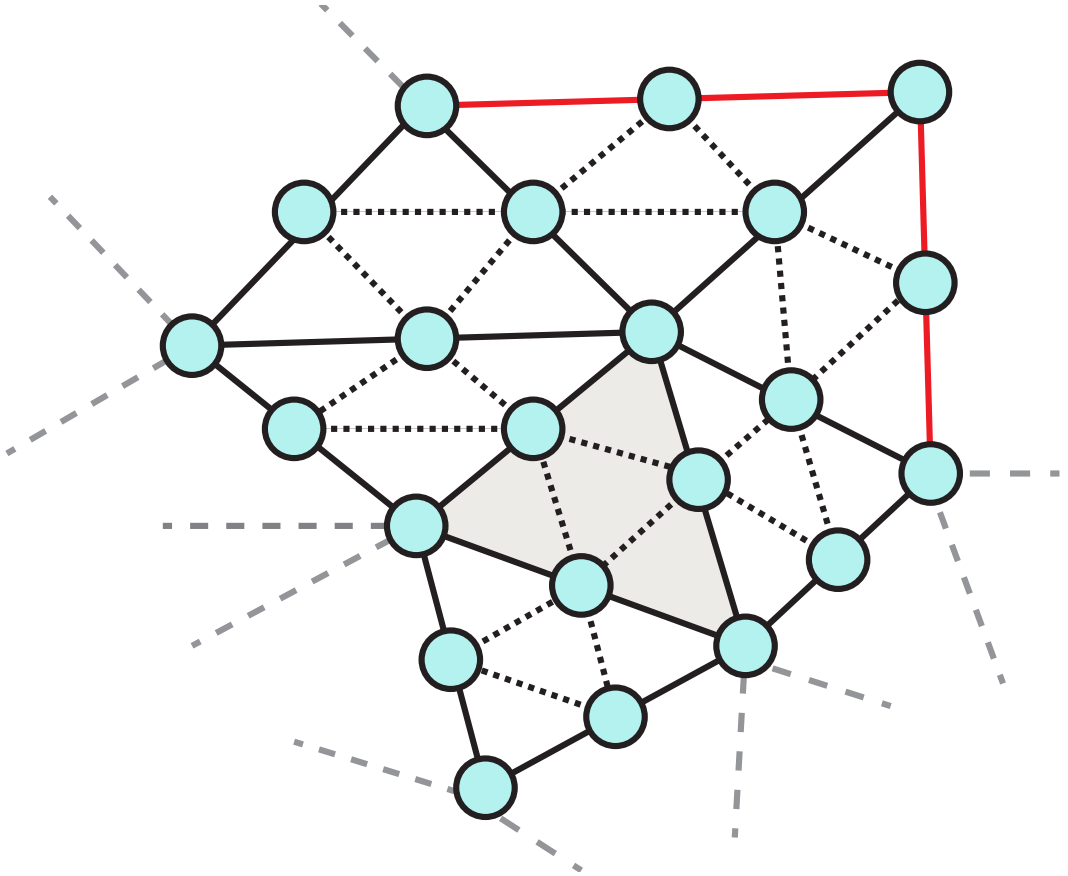


Vertex from input mesh

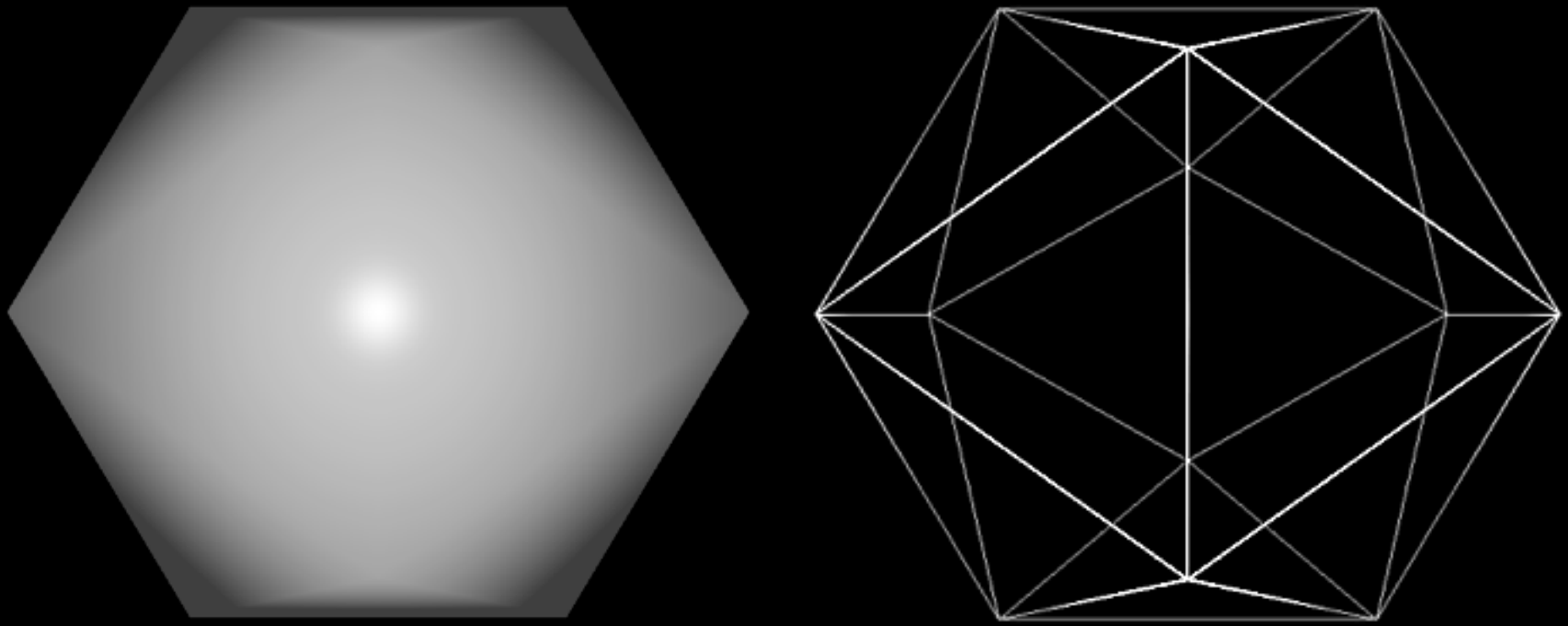
Newly-introduced vertex

Assignment 2: Loop Subdivision

- This becomes the new base mesh for next round

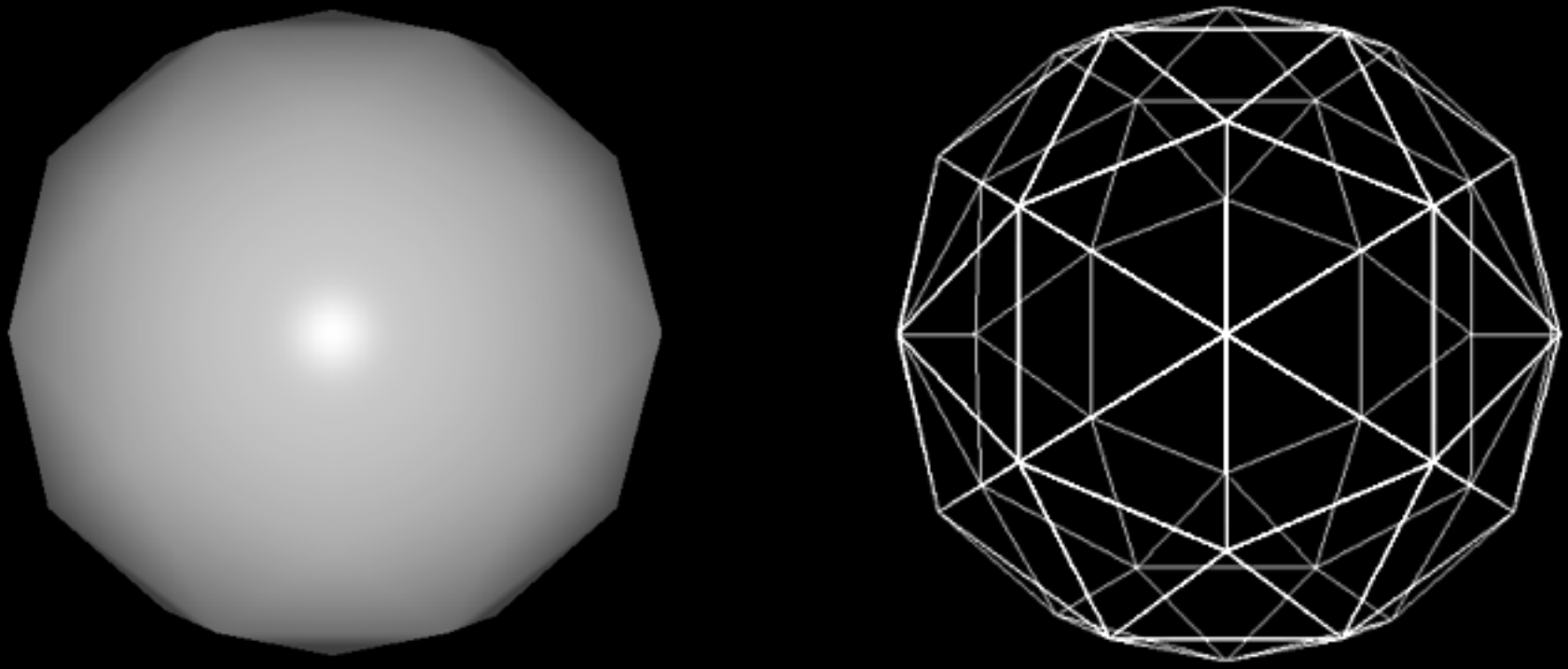


Assignment 2: Loop Subdivision



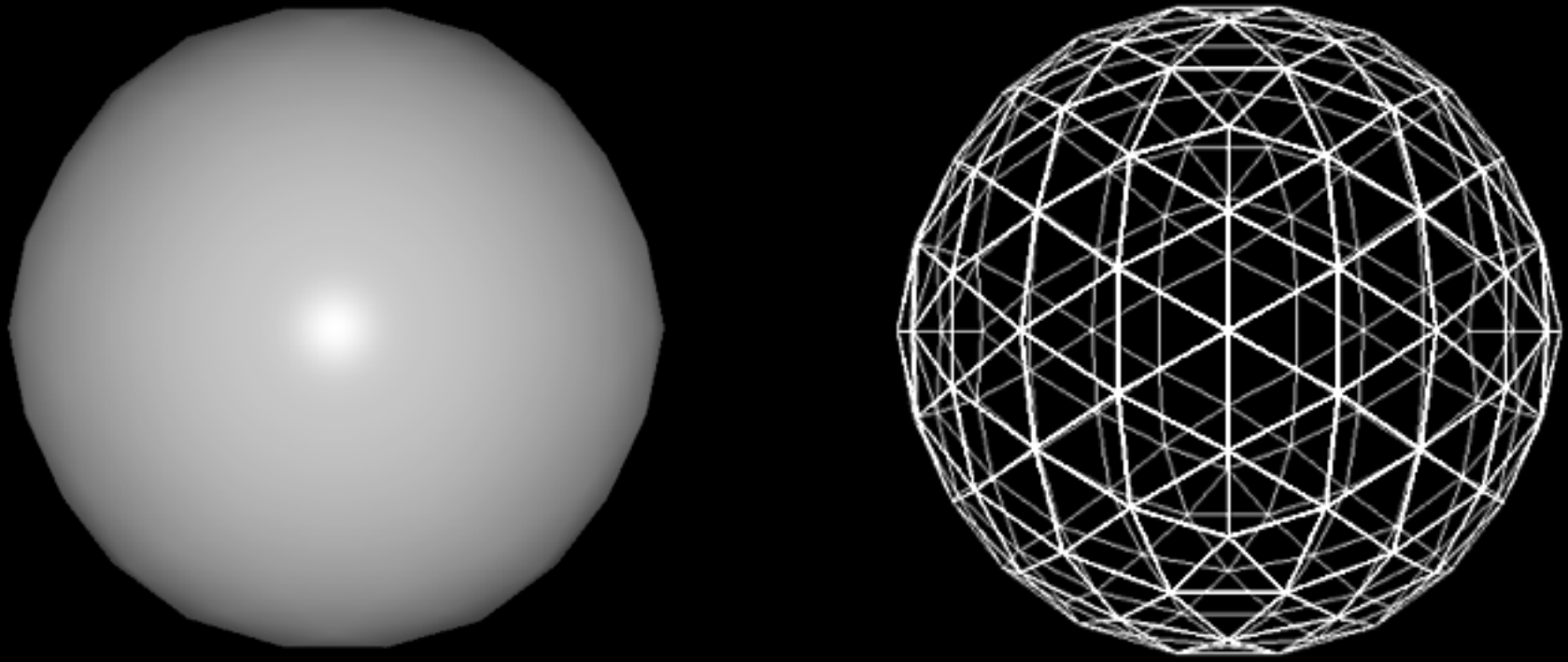
Input mesh

Assignment 2: Loop Subdivision



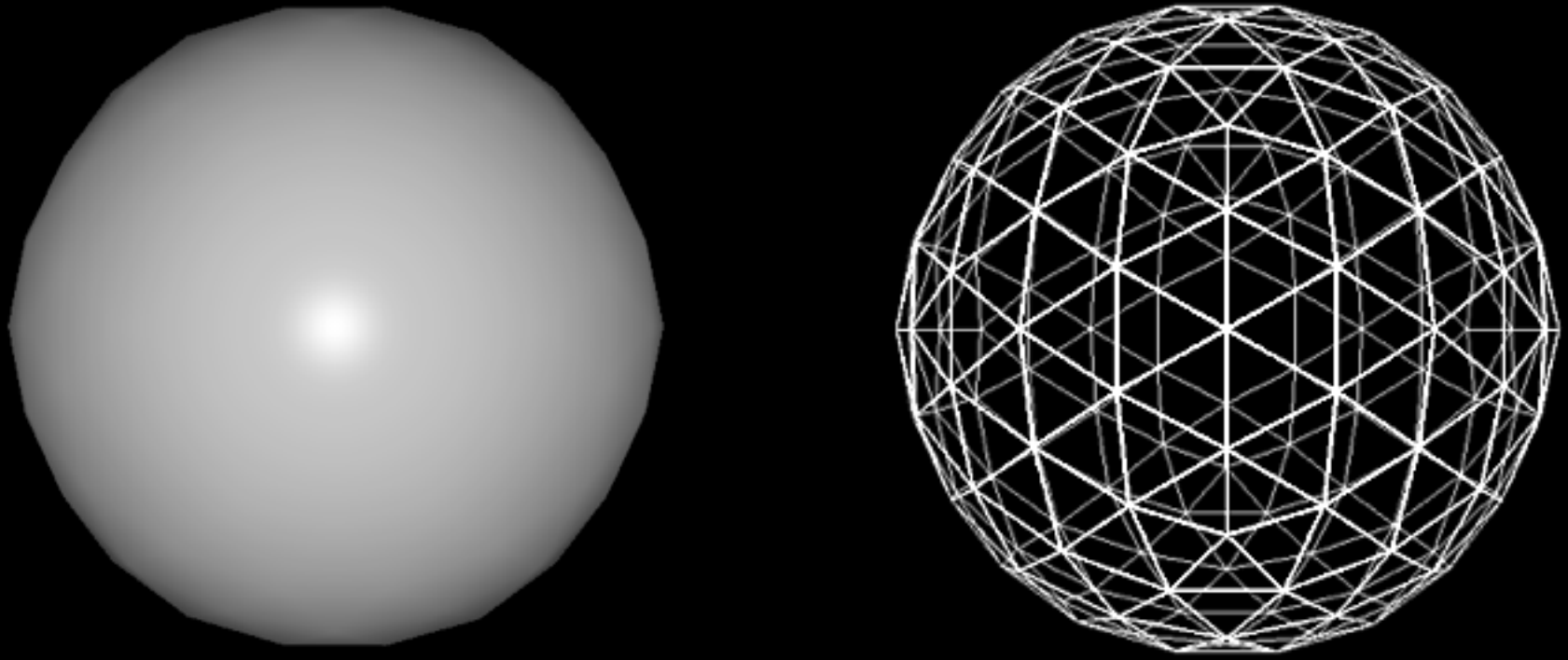
1st level of subdivision

Assignment 2: Loop Subdivision



2nd level of subdivision

Assignment 2: Loop Subdivision

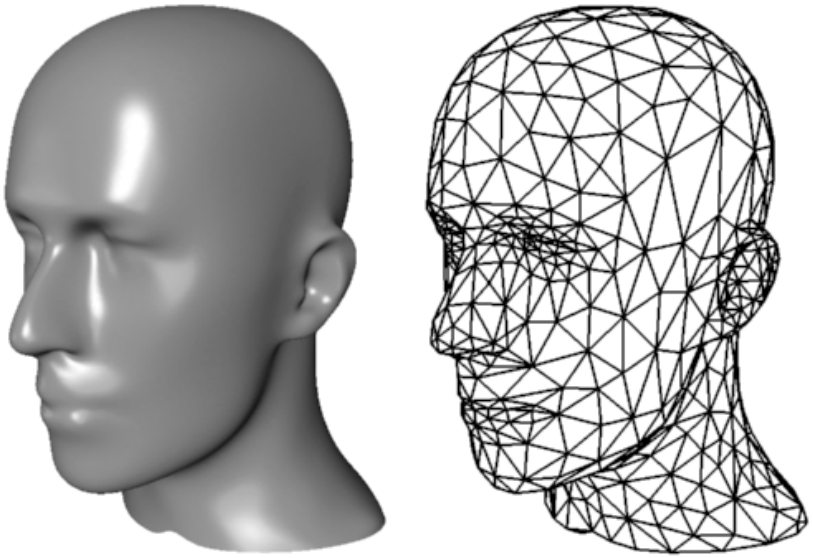


2nd level of subdivision

etc.

Flavors of Subdivision Surfaces

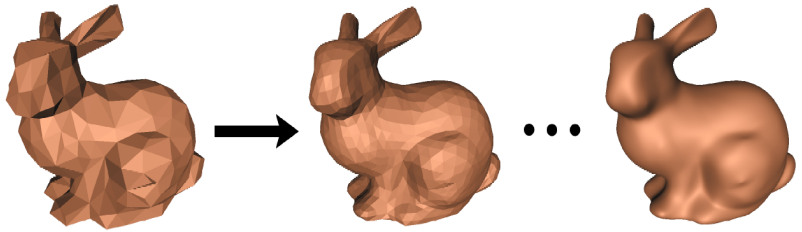
- Catmull-Clark
  - Quads and triangles
  - Generalizes bicubics to arbitrary topology!
- **Loop**, Butterfly
  - Triangles
- Doo-Sabin, sqrt(3), biquartic...
  - and a whole host of others
- Used **everywhere** in movie and game modeling!
- See http://www.cs.nyu.edu/~dzorin/sig00course/



Leif Kobbelt

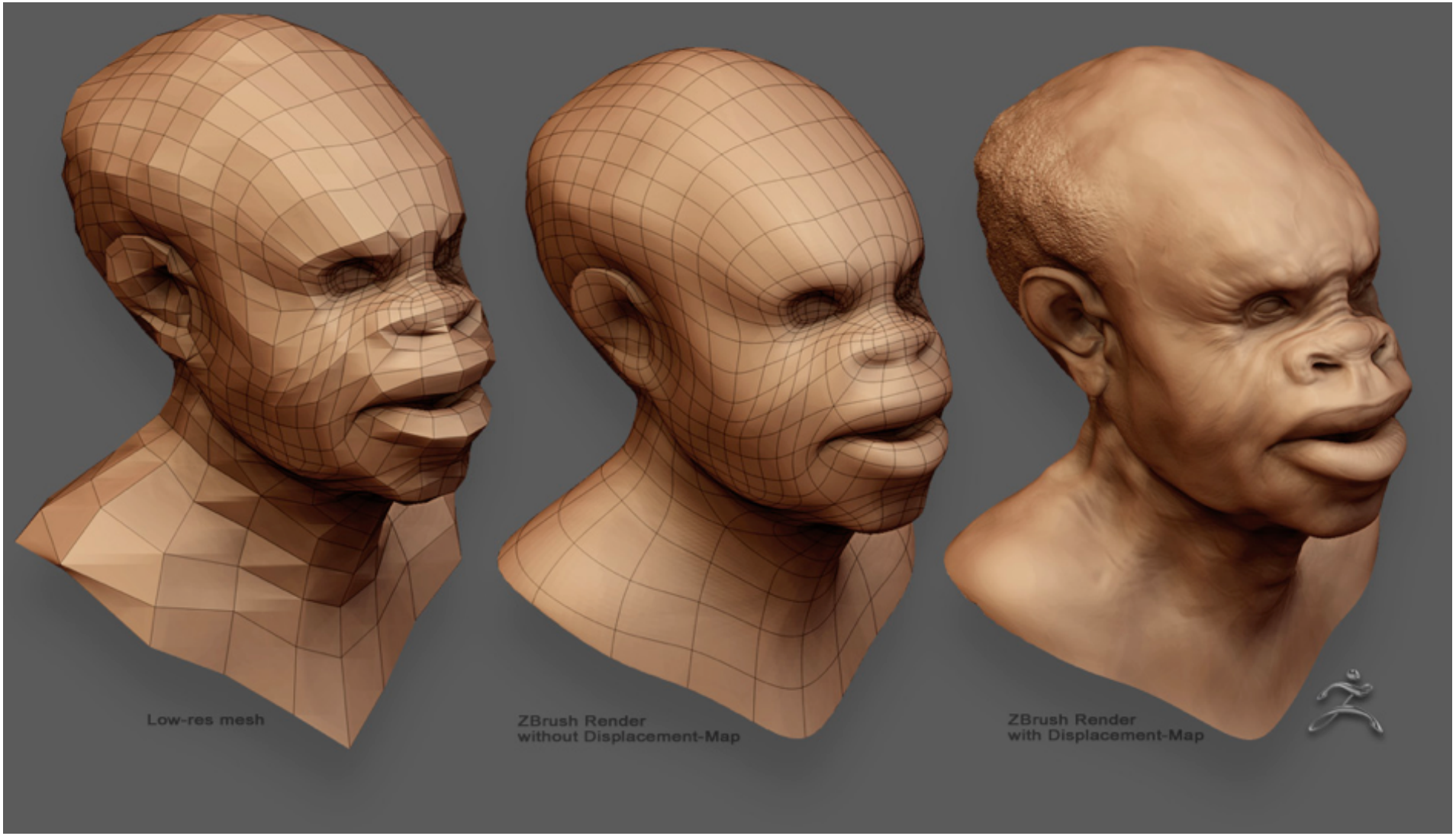
Subdivision Curves and Surfaces

- Advantages
  - Arbitrary topology
  - Smooth at boundaries
  - Level of detail, scalable
  - Simple representation
  - Numerical stability, well-behaved meshes
  - Code simplicity
- Little disadvantage:
  - Procedural definition
  - Not parametric, not implicit
  - Tricky at special vertices



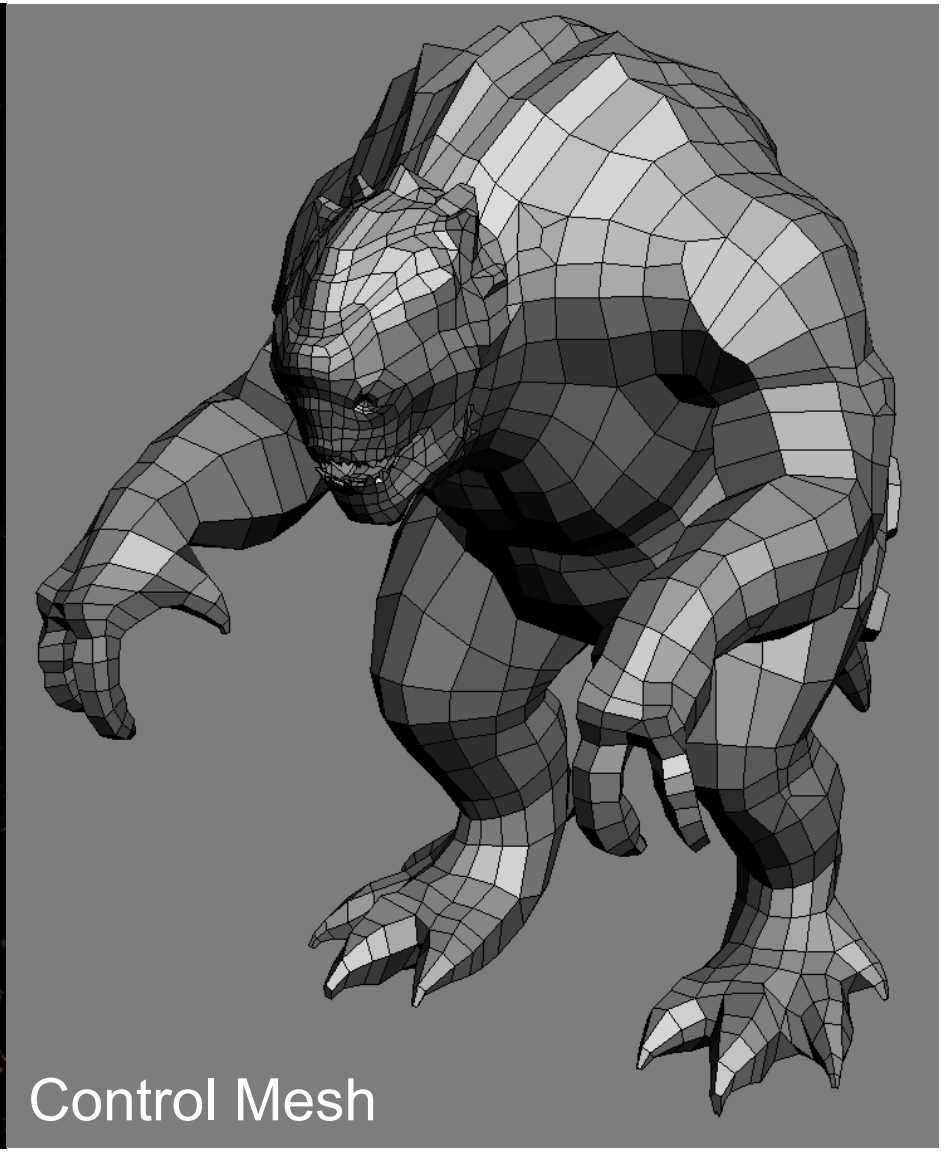
Warren et al.

Subdivision + Displacement



Subdivision + Displacement

Epic Games



ZBrush Modeling Session

- http://www.youtube.com/watch?v=i8lIvWN8ZmQ