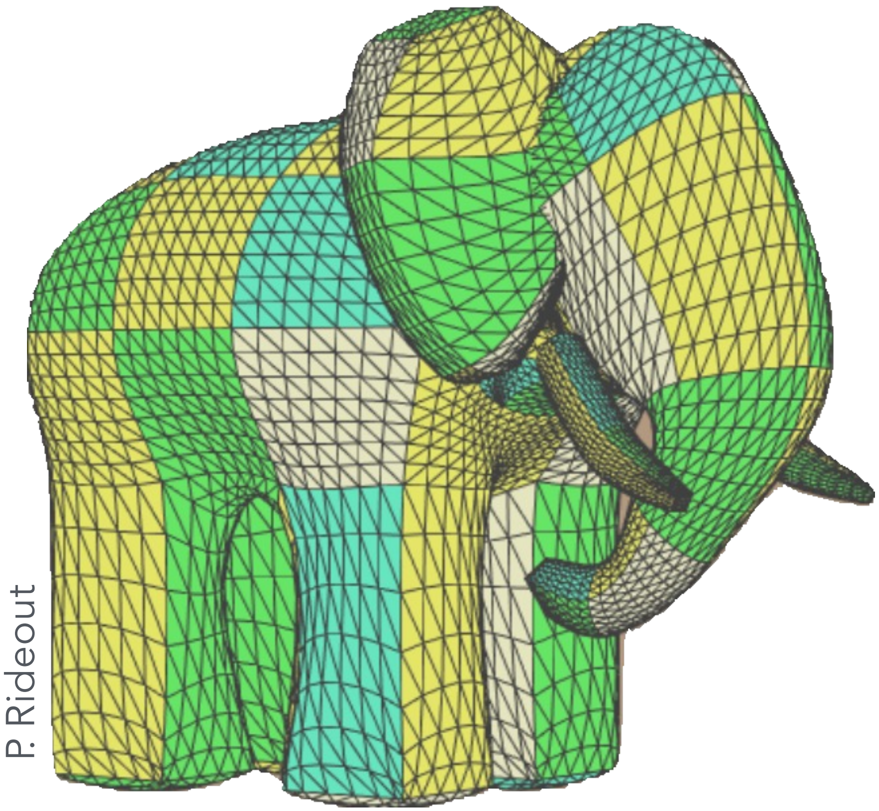
Geometry Processing

Today

- Curves
  - Bezier curves
  - De Casteljau's algorithm
  - B-splines, etc.
- Surfaces
  - Bezier surfaces
  - Subdivision surfaces (triangles & quads)

Bézier Surfaces

Extend Bézier curves to surfaces



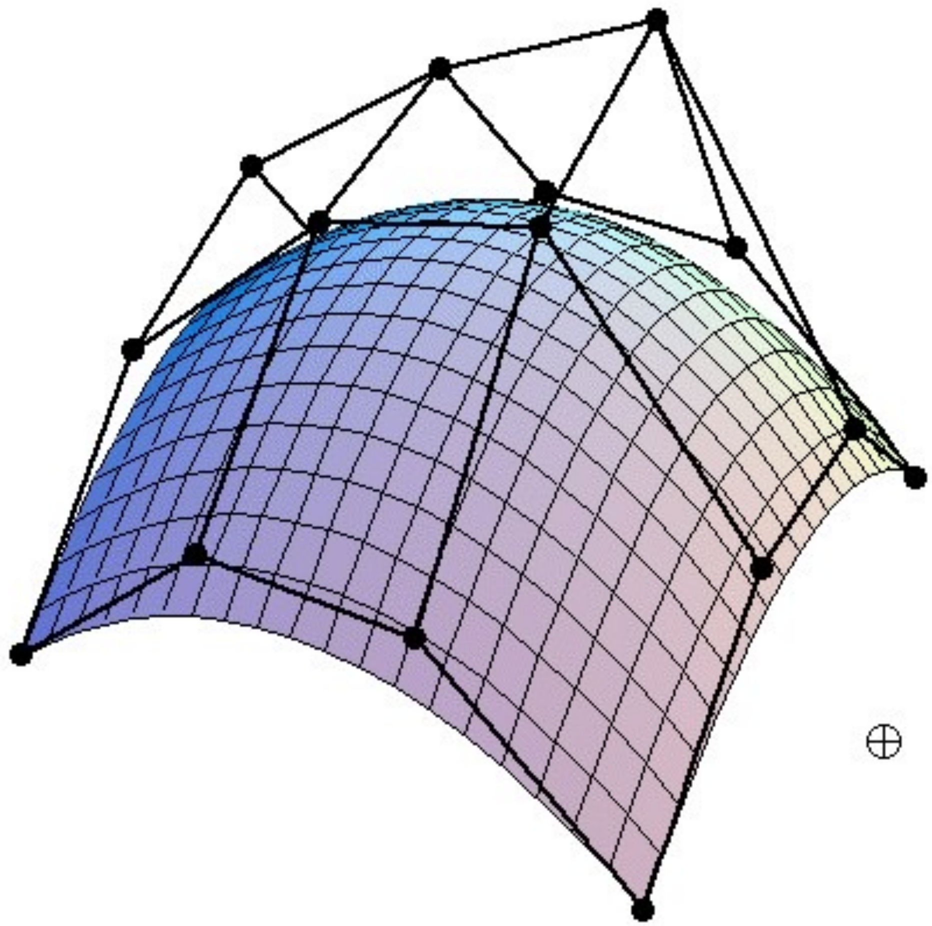
P. Rideout

Ed Catmull's "Gumbo" model

renderspirit.com

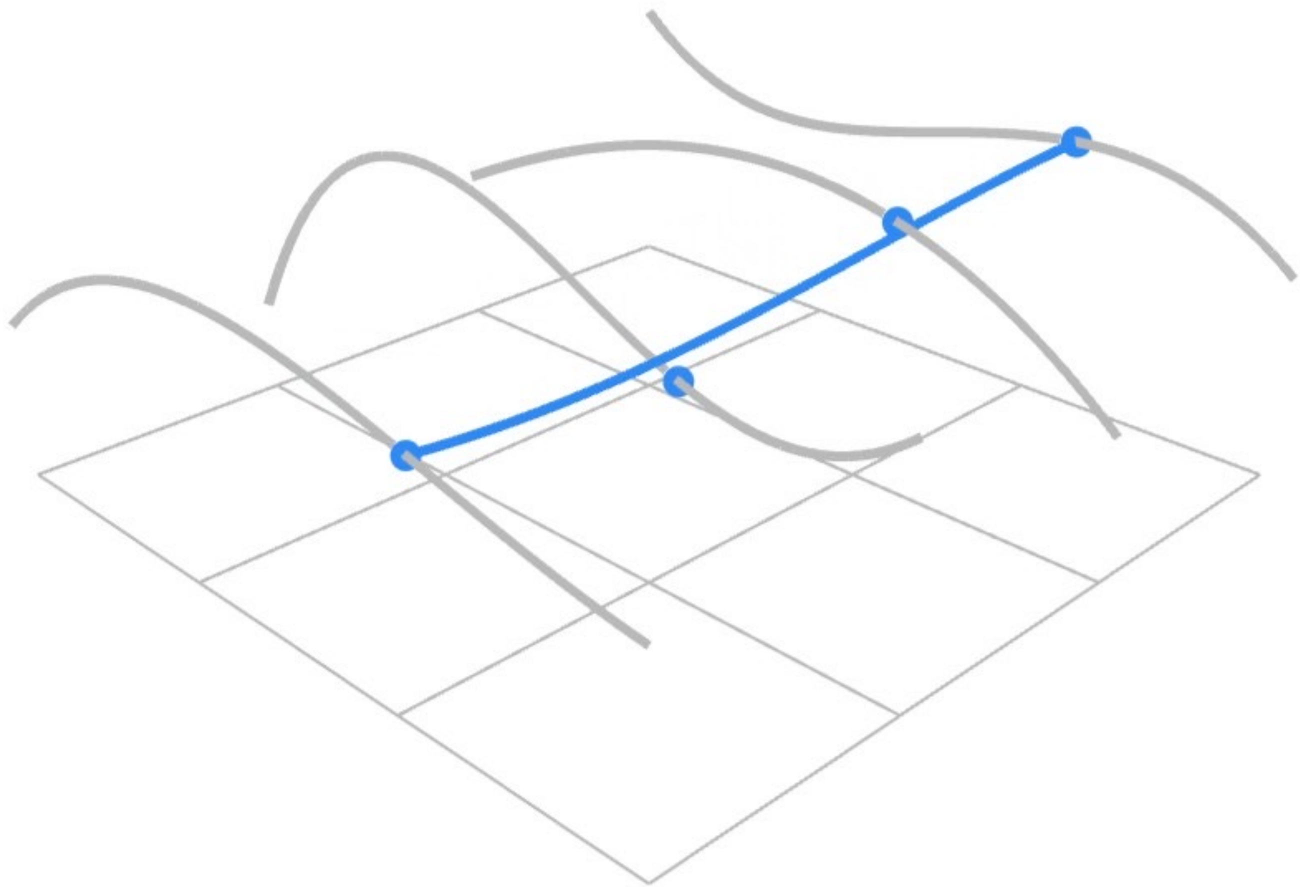
Utah Teapot

Bicubic Bézier Surface Patch



Bezier surface and 4 x 4 array of control points

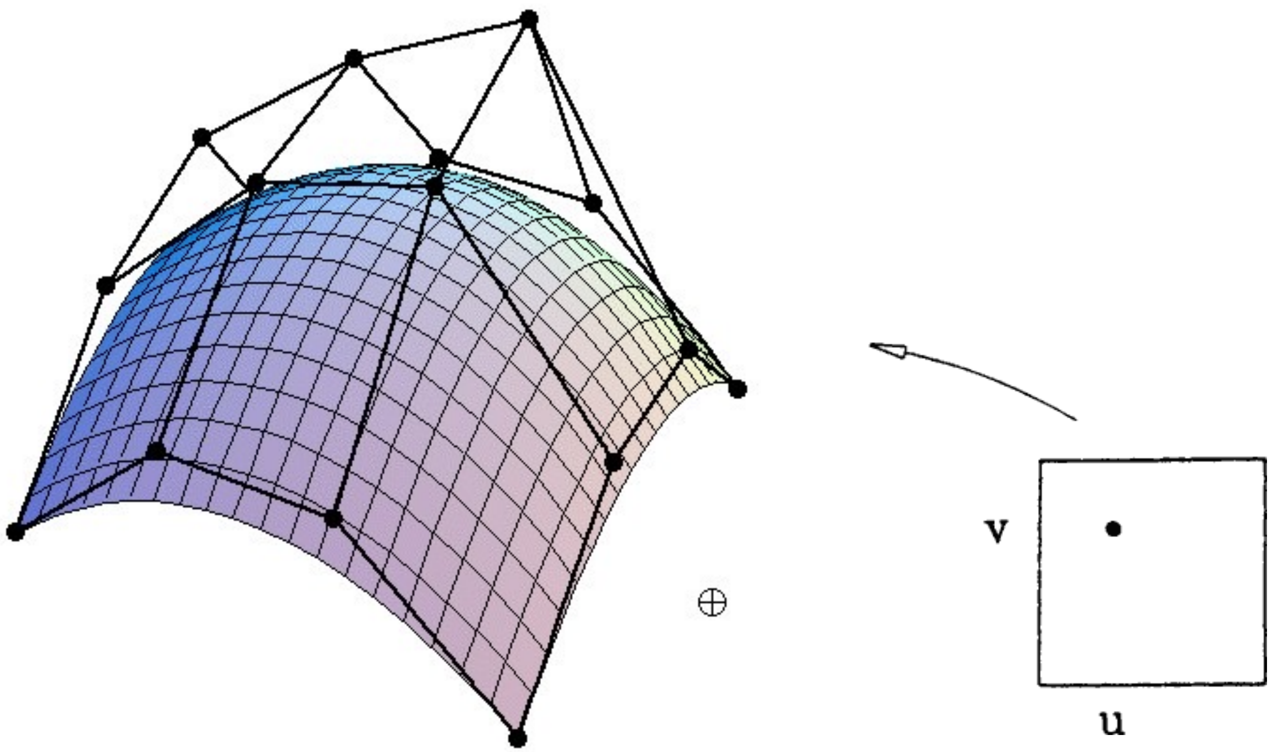
Visualizing Bicubic Bézier Surface Patch



Evaluating Bézier Surfaces

Evaluating Surface Position For Parameters (u,v)

For bi-cubic Bezier surface patch,
  Input: 4x4 control points
  Output is 2D surface parameterized by (u,v) in $[0,1]^2$

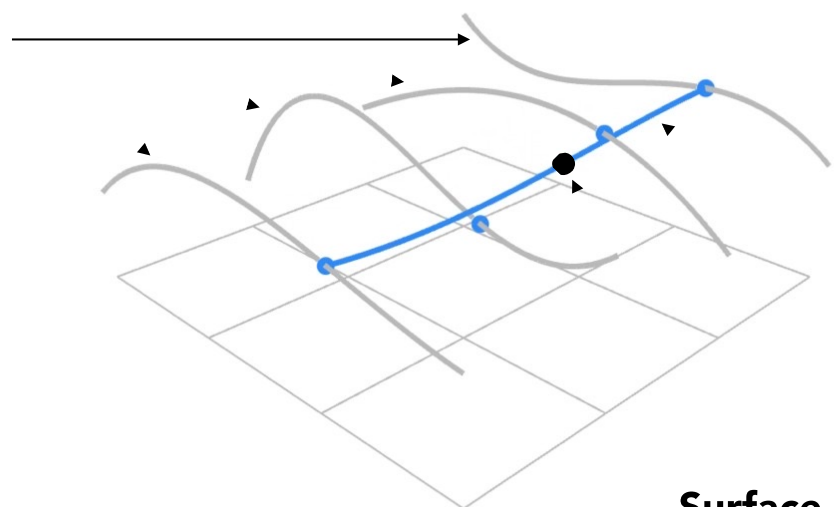


Method: Separable 1D de Casteljau Algorithm

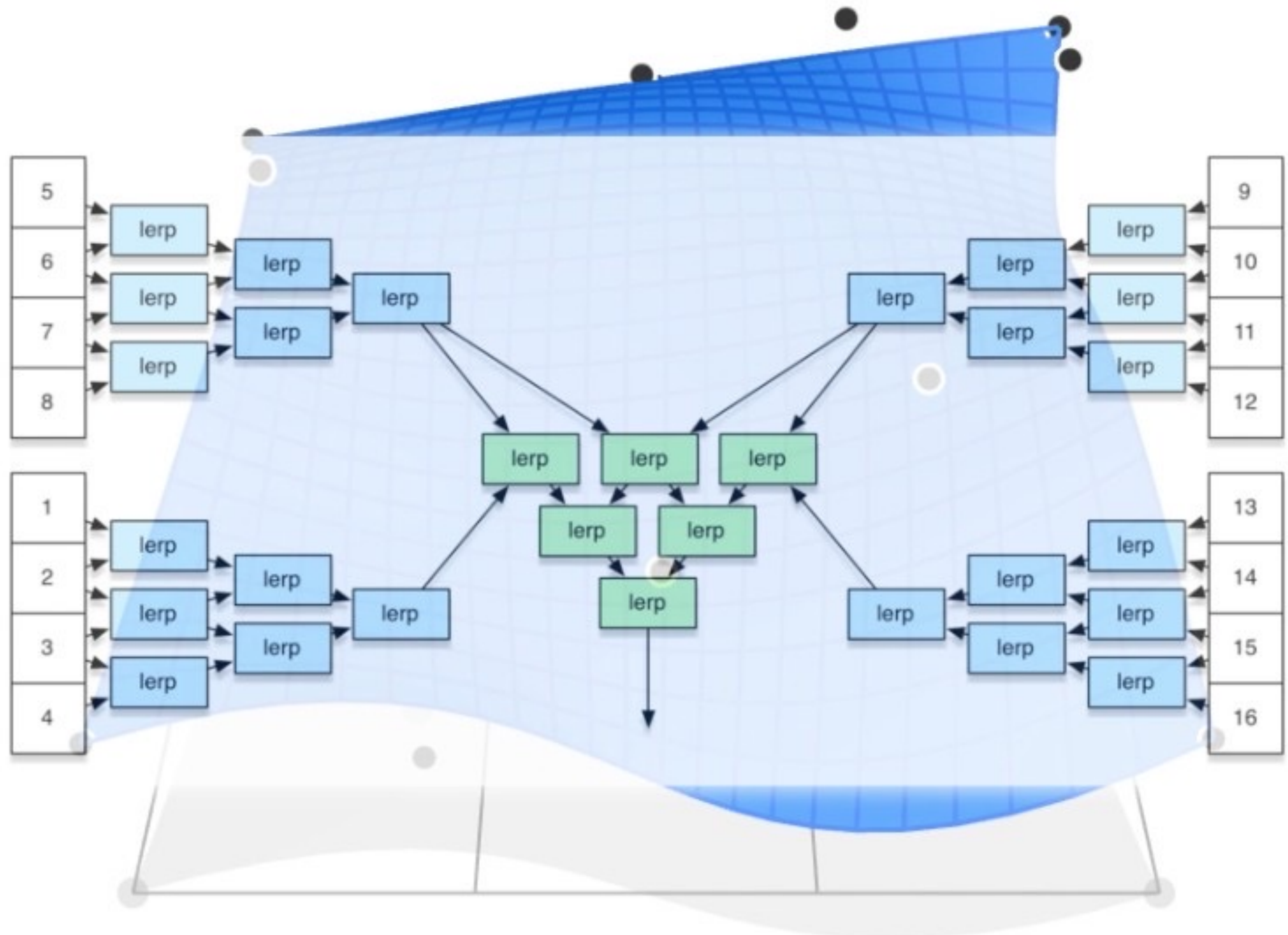
Goal: Evaluate surface position corresponding to (u,v)

(u,v)-separable application of de Casteljau algorithm

- Use de Casteljau to evaluate point u on each of the 4 Bezier curves in u. This gives 4 control points for the "moving" Bezier curve
- Use 1D de Casteljau to evaluate point v on the "moving" curve

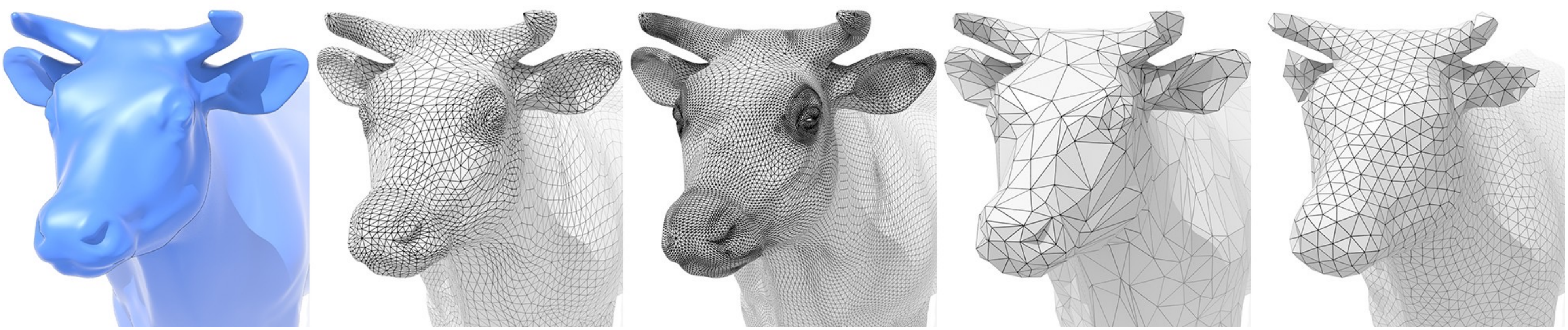


Method: Separable 1D de Casteljau Algorithm



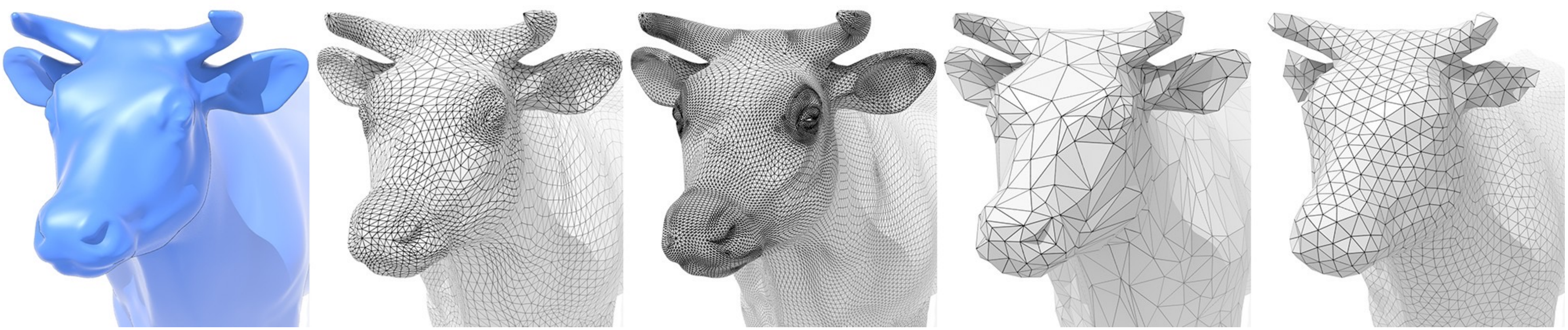
Mesh Operations: Geometry Processing

- Mesh subdivision
- Mesh simplification
- Mesh regularization

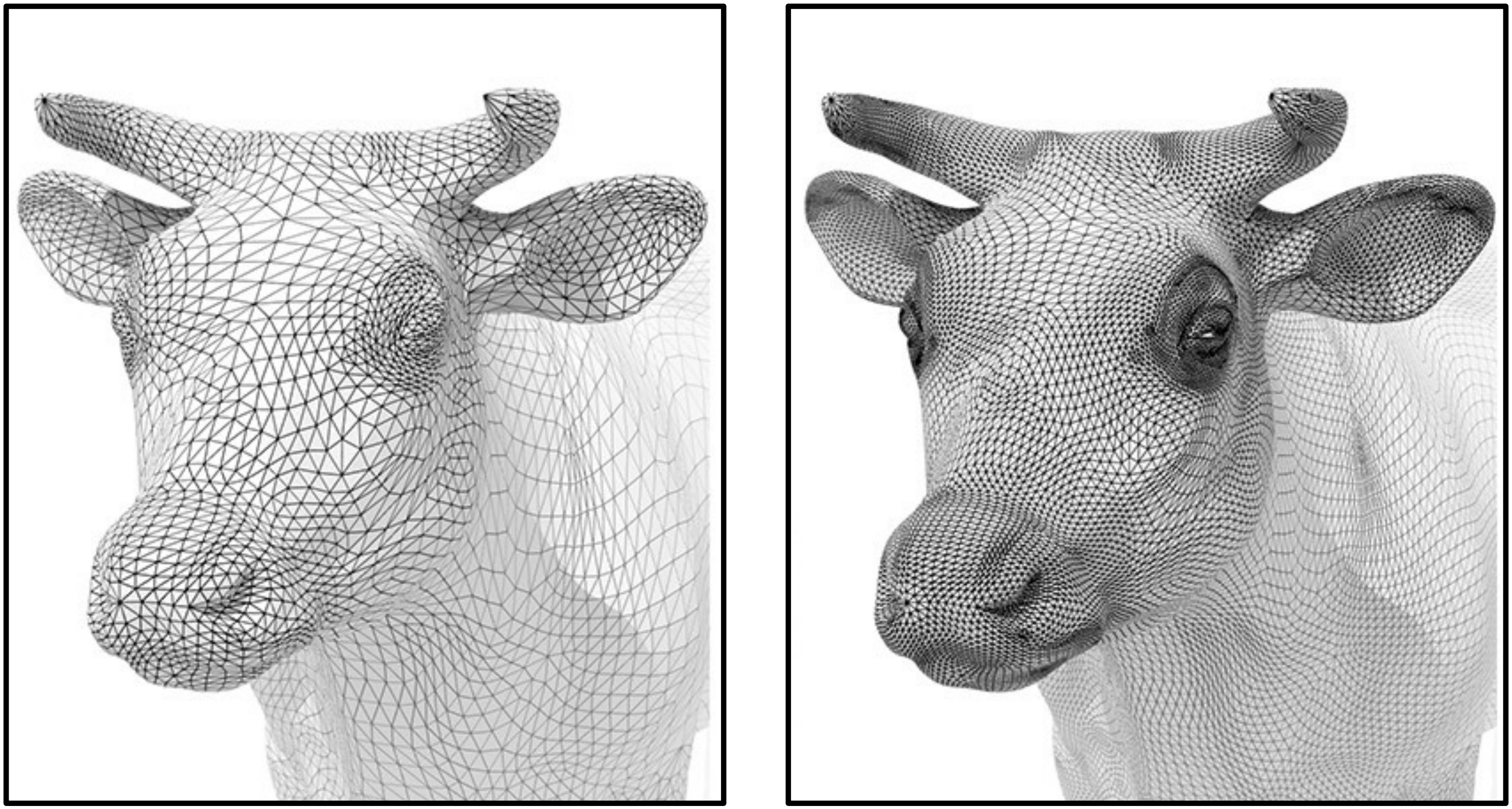


Mesh Operations: Geometry Processing

- Mesh subdivision
- Mesh simplification
- Mesh regularization

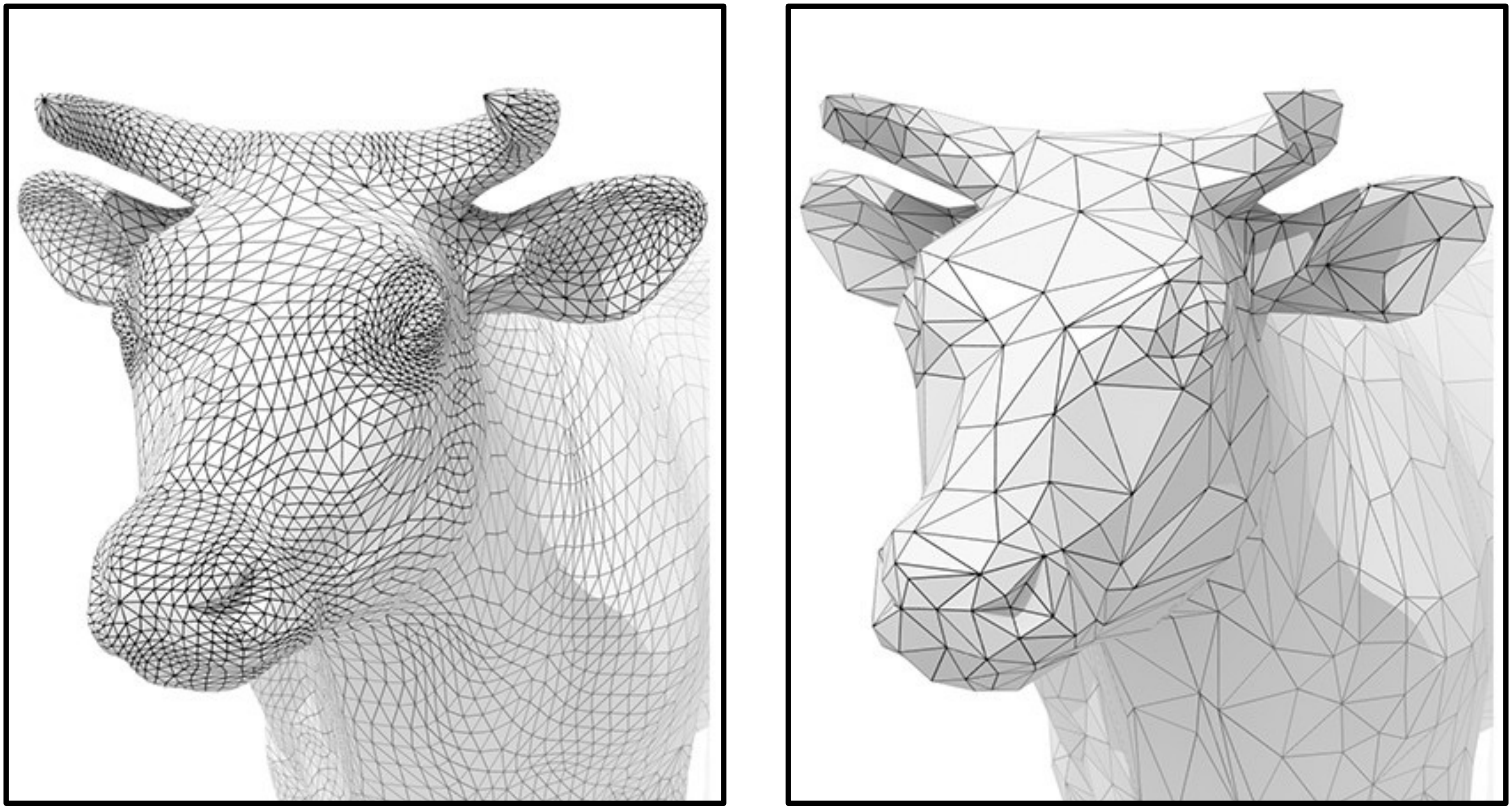


Mesh Subdivision (upsampling)



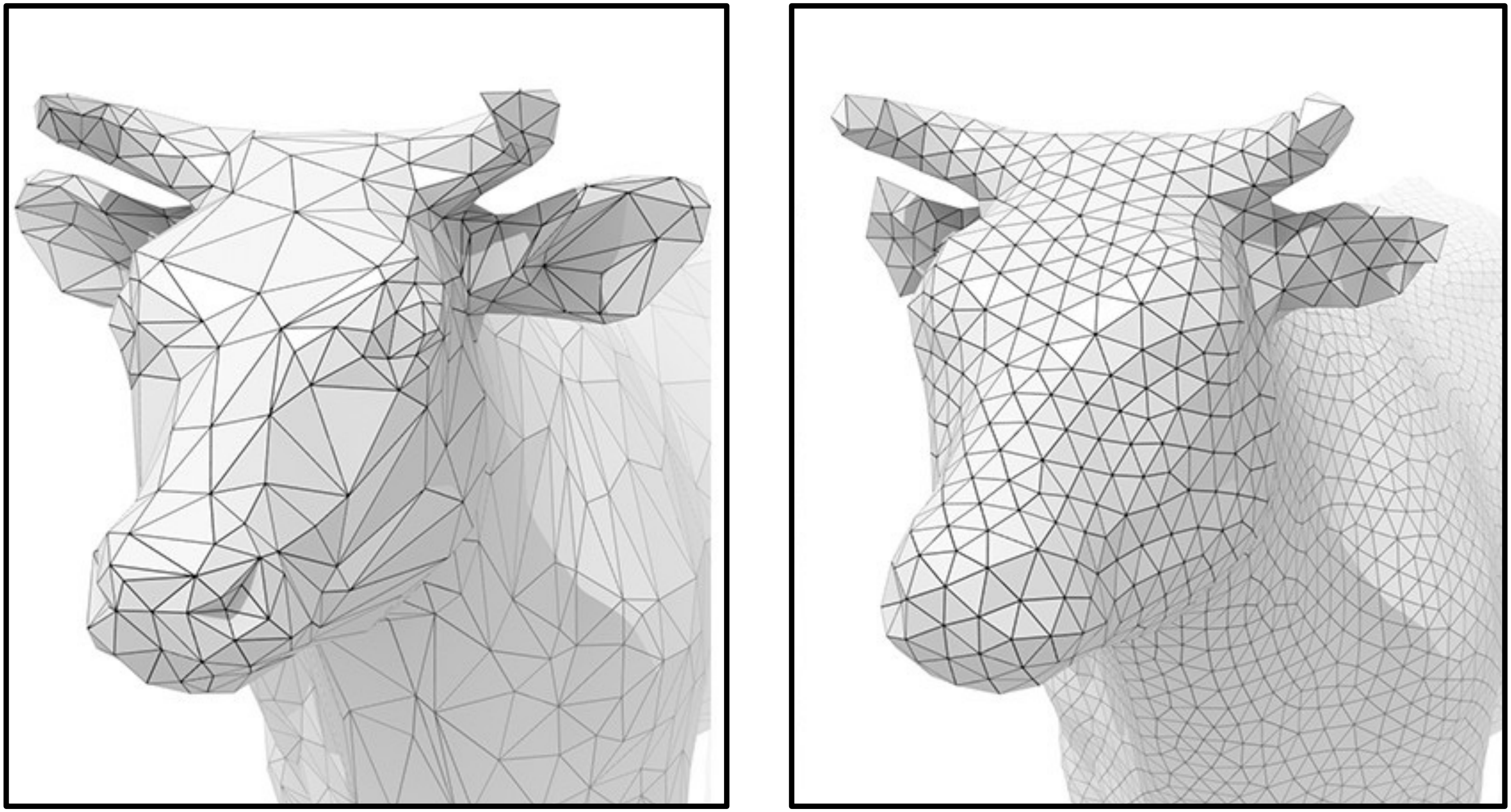
Increase resolution

Mesh Simplification (downsampling)



Decrease resolution; try to preserve shape/appearance

Mesh Regularization (same #triangles)



Modify sample distribution to **improve quality**

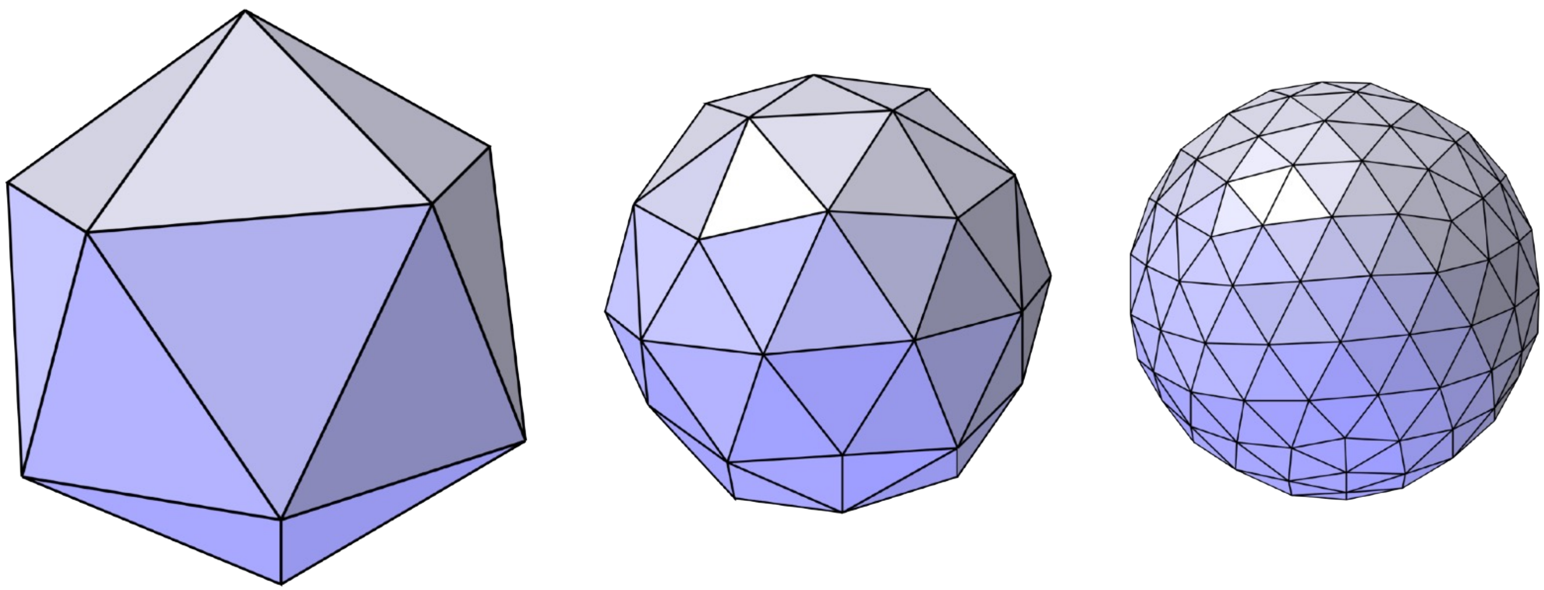
Subdivision

Loop Subdivision

Common subdivision rule for triangle meshes

First, create more triangles (vertices)

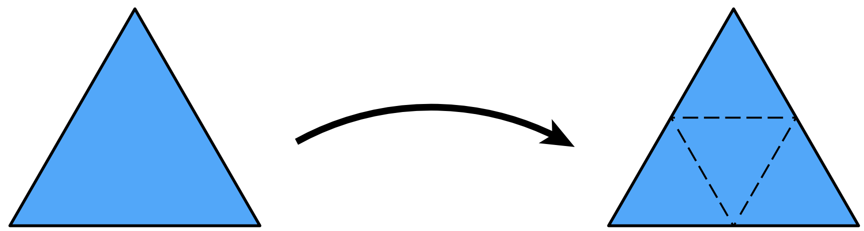
Second, tune their positions



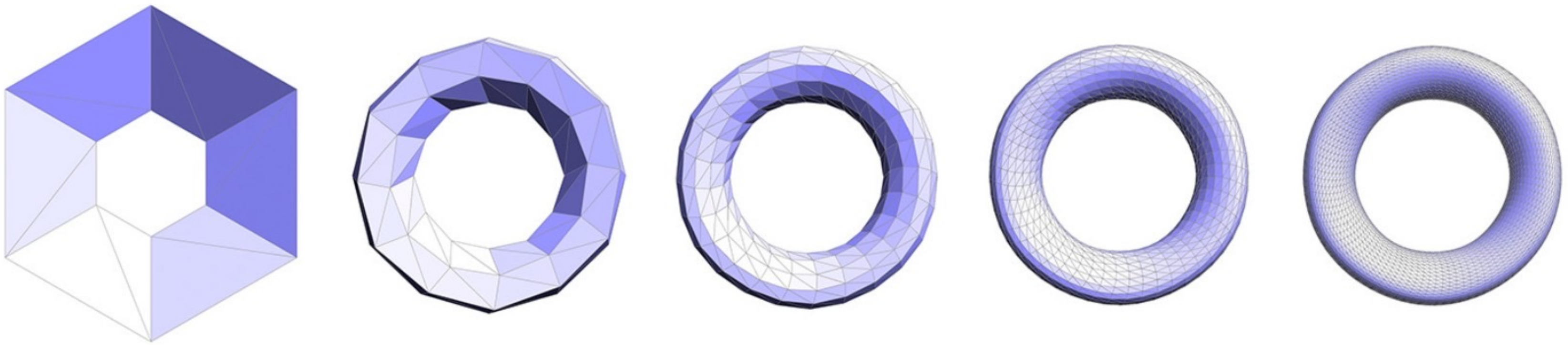
Simon Fuhrman

Loop Subdivision

- Split each triangle into four

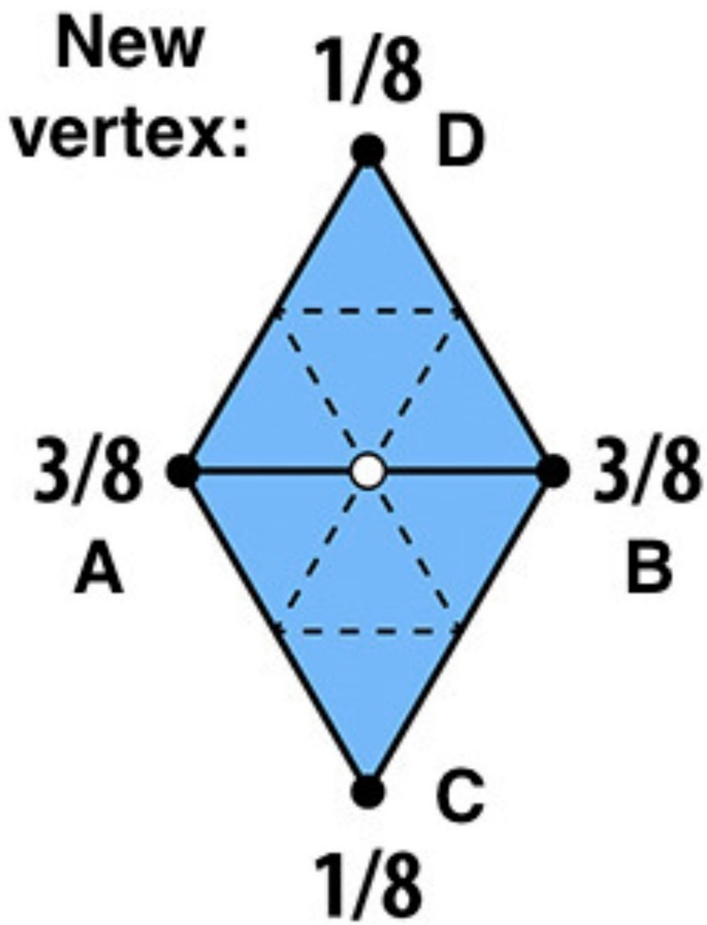


- Assign new vertex positions according to weights
  - New / old vertices updated differently



Loop Subdivision — Update

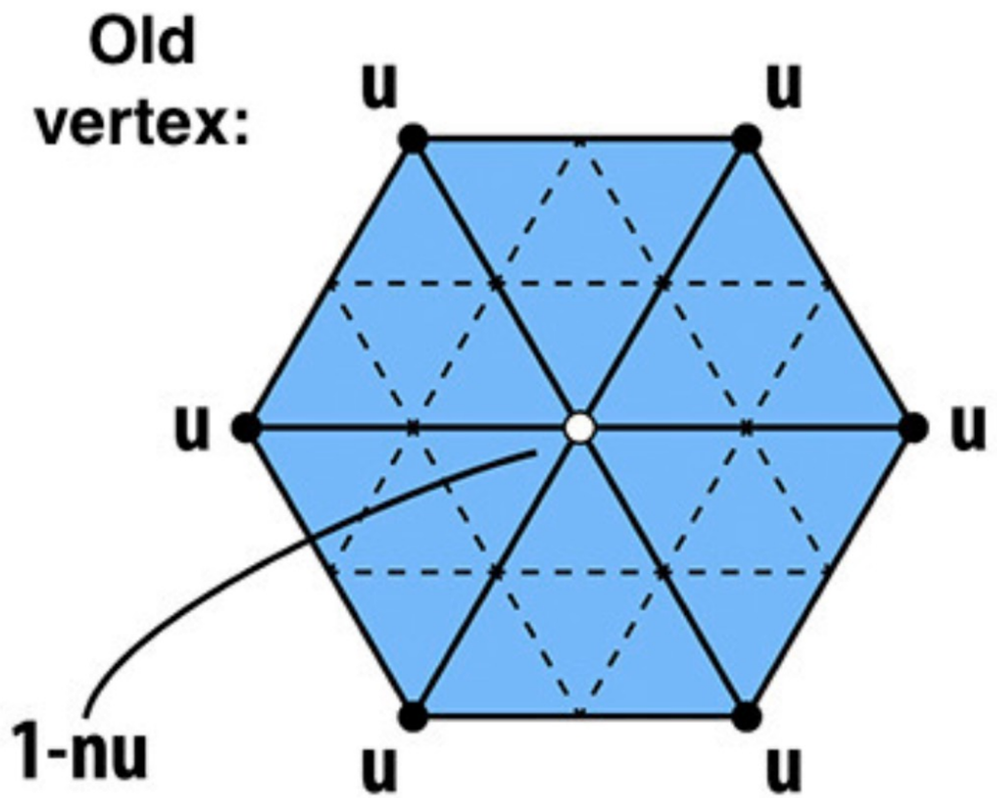
For new vertices:



Update to:
3/8 * (A + B) + 1/8 * (C + D)

Loop Subdivision — Update

For old vertices (e.g. degree 6 vertices here):

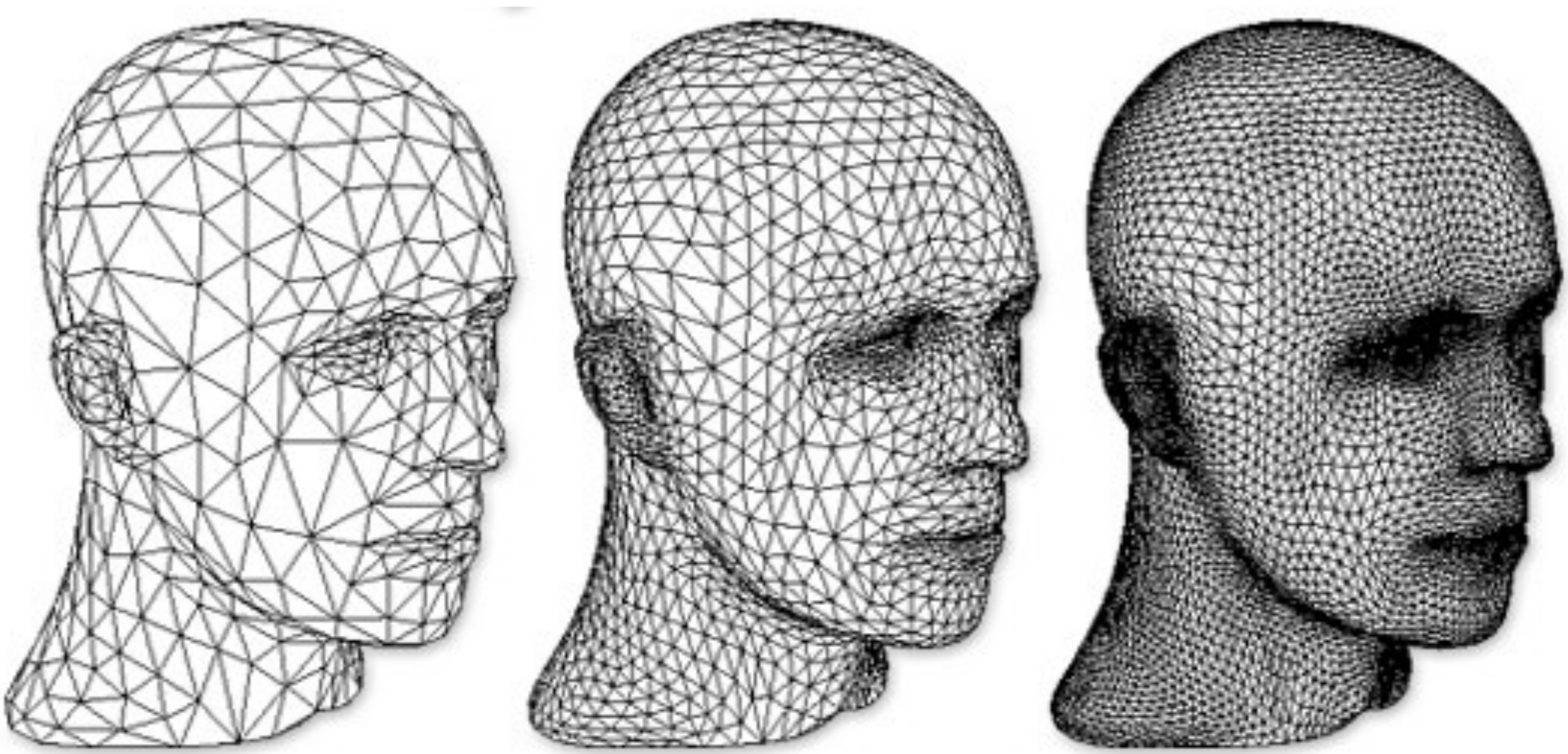


Update to:
(1 - n*u) * original_position +
u * neighbor_position_sum

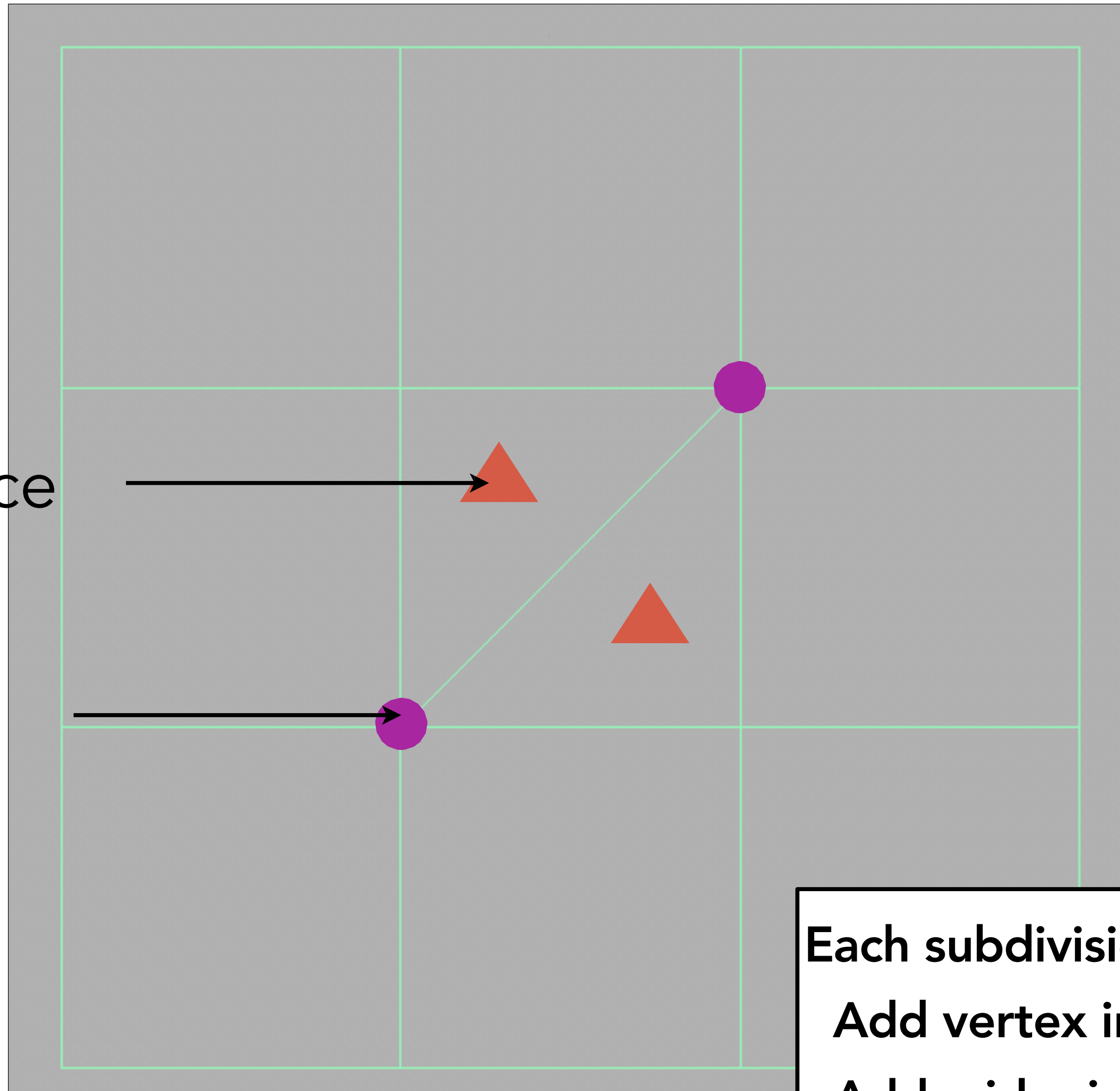
n: vertex degree

u: 3/16 if n=3, 3/(8n) otherwise

Loop Subdivision Results



Catmull-Clark Subdivision (General Mesh)

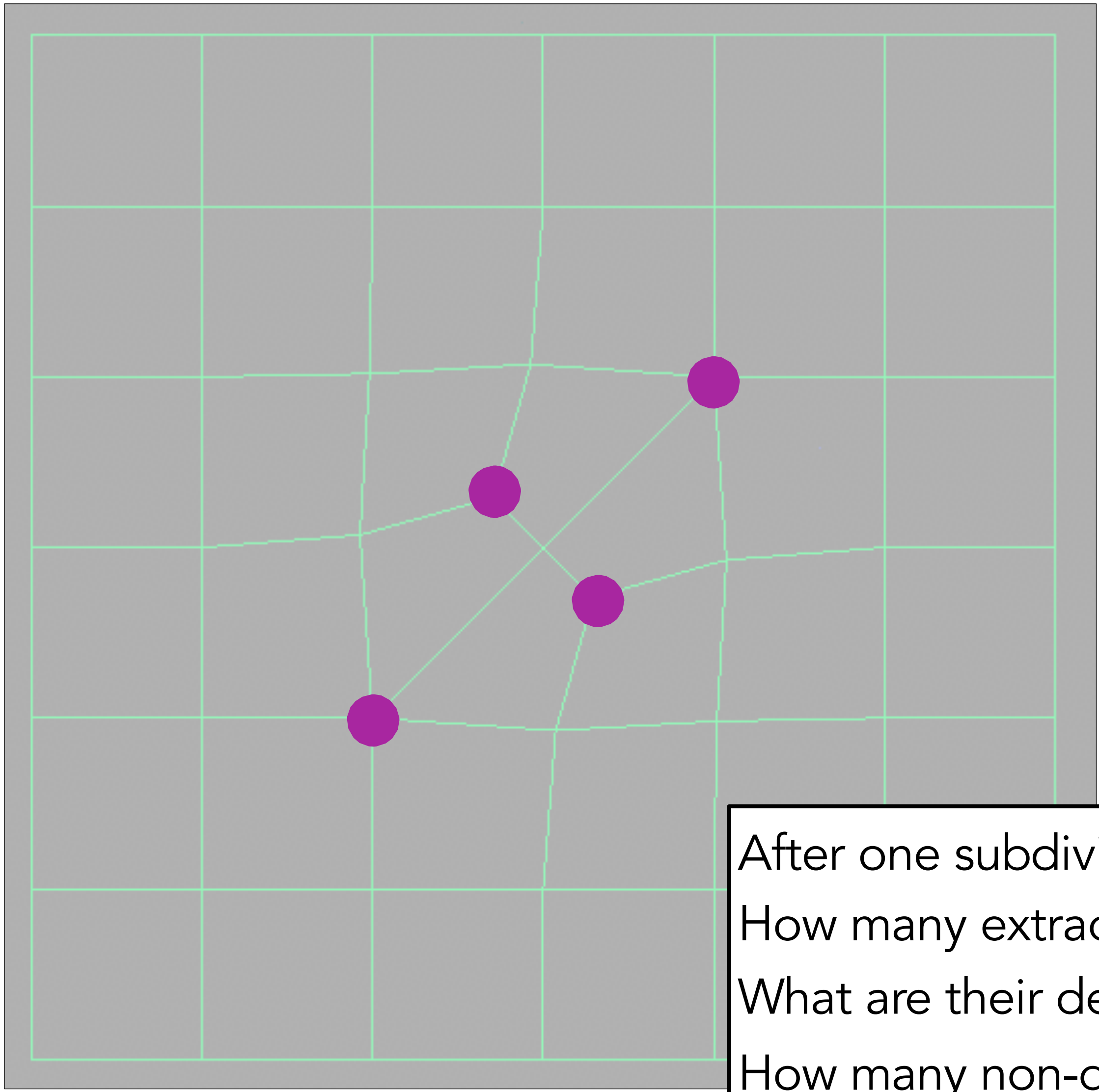


Non-quad face

Extraordinary vertex (degree != 4)

Each subdivision step:
- **Add vertex in each face**
- **Add midpoint on each edge**
- **Connect all new vertices**

Catmull-Clark Subdivision (General Mesh)



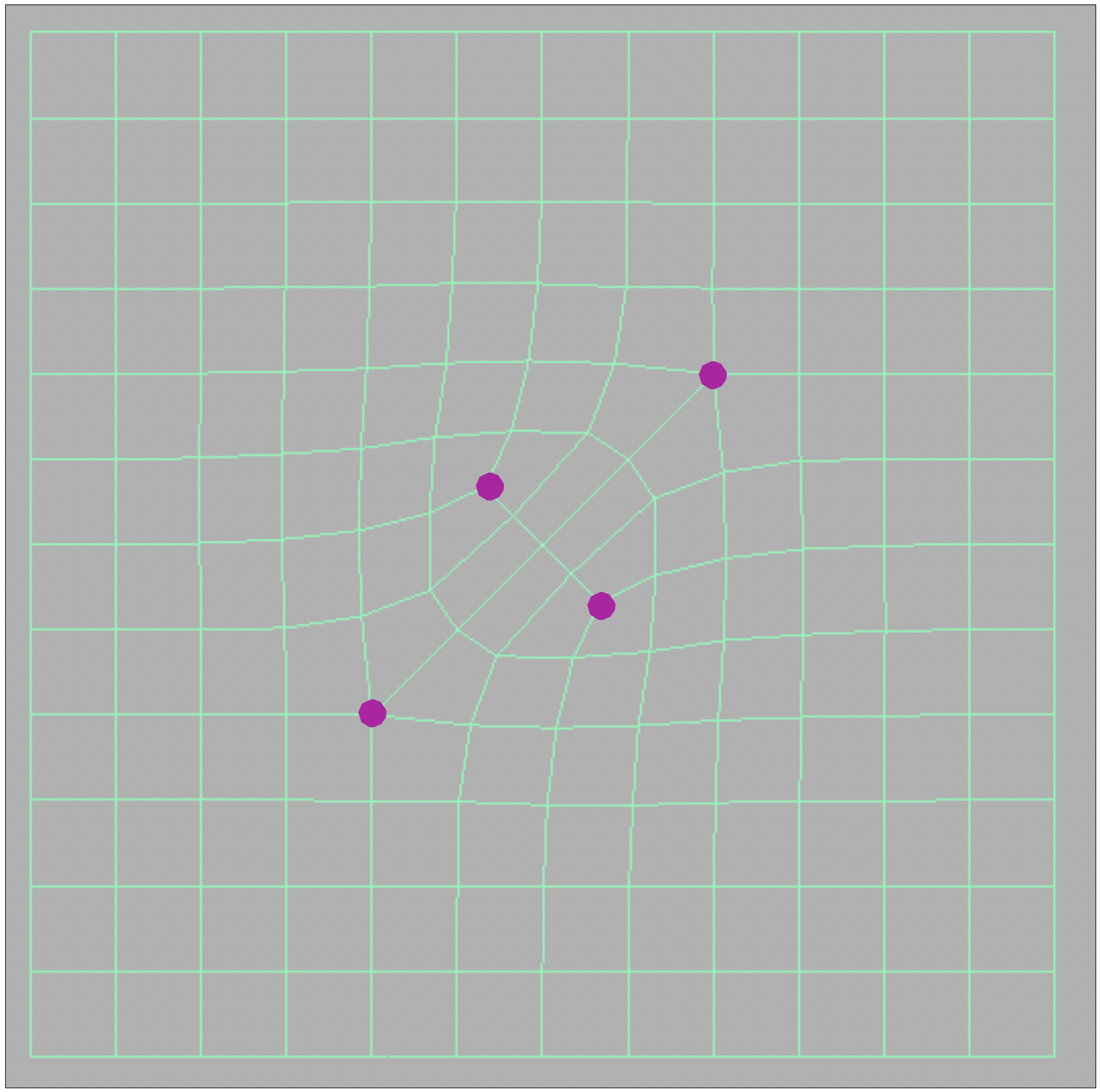
After one subdivision:

How many extraordinary vertices?

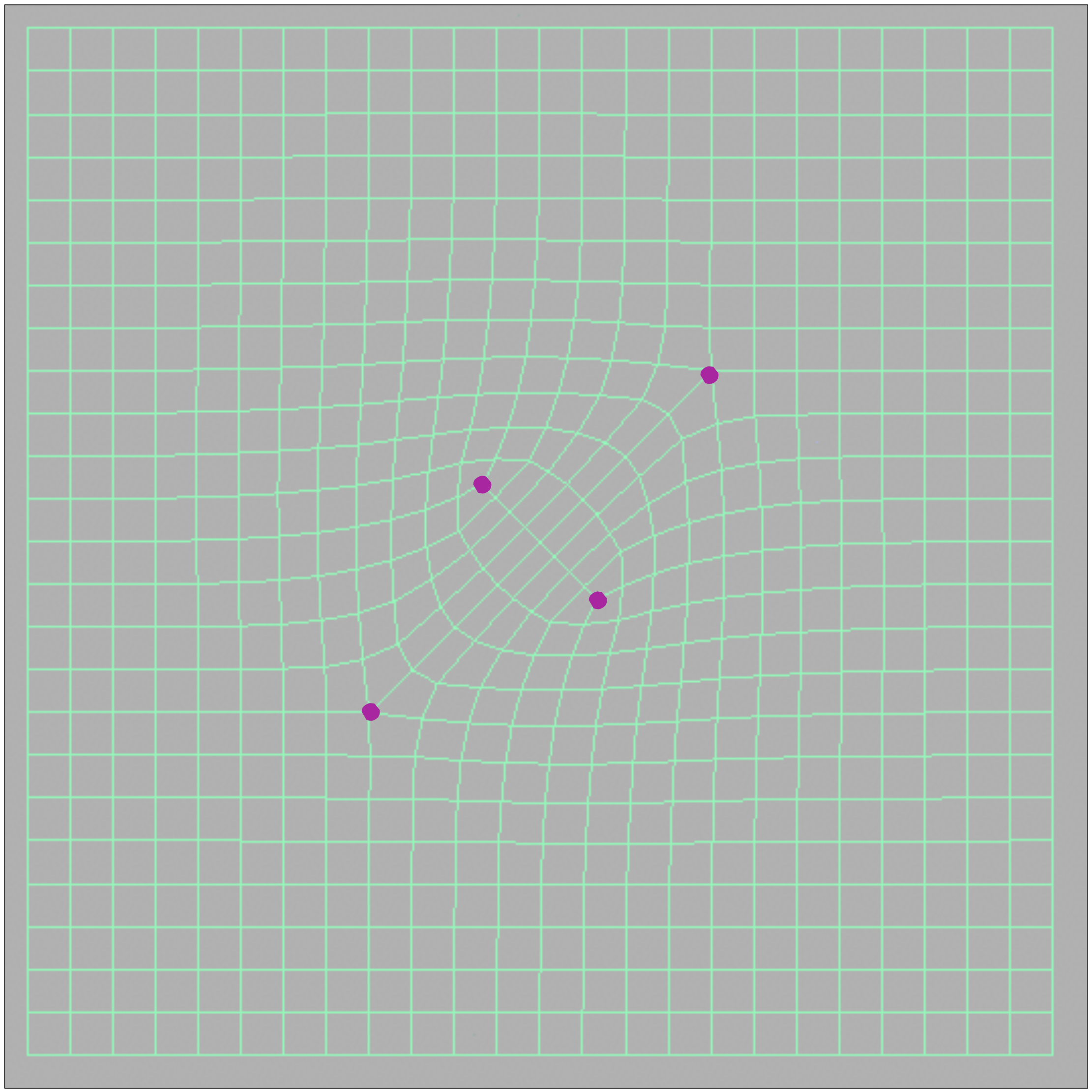
What are their degrees?

How many non-quad faces?

Catmull-Clark Subdivision (General Mesh)

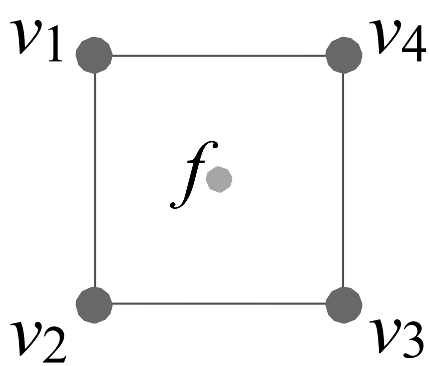


Catmull-Clark Subdivision (General Mesh)



FYI: Catmull-Clark Vertex Update Rules (Quad Mesh)

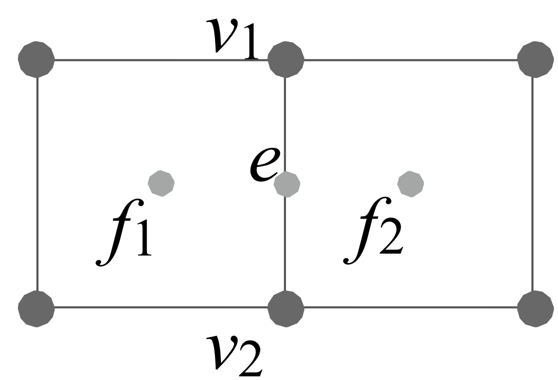
Face point

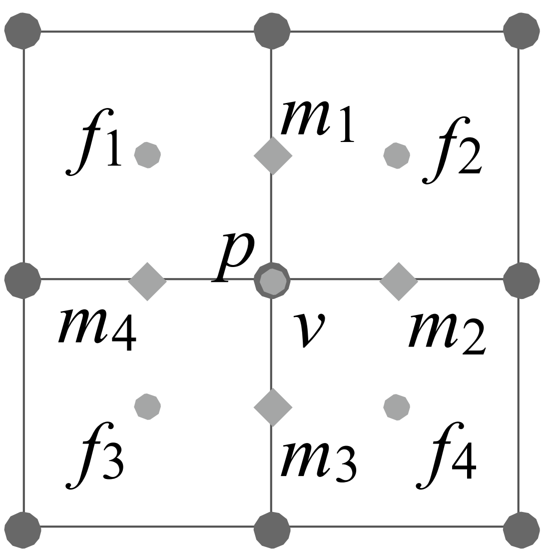


$$f = \frac{v_1 + v_2 + v_3 + v_4}{4}$$

$$e = \frac{v_1 + v_2 + f_1 + f_2}{4}$$

Edge point





Vertex point

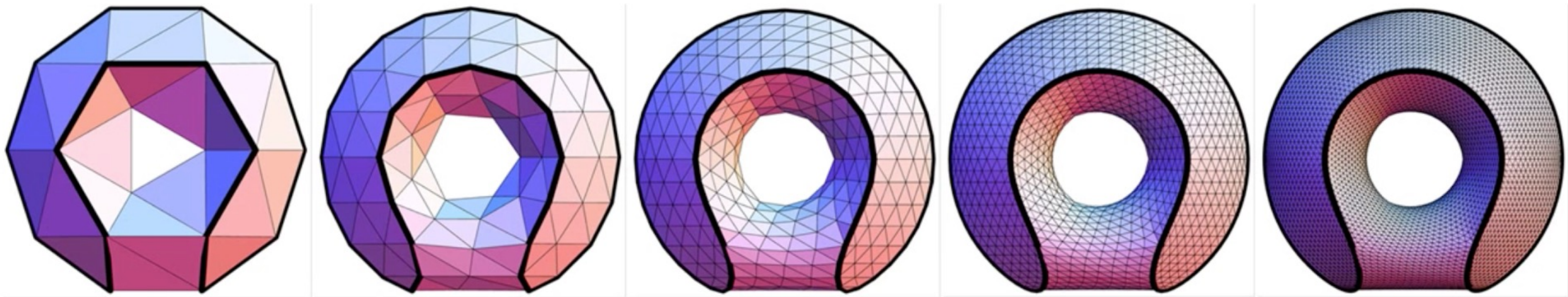
$$v = \frac{f_1 + f_2 + f_3 + f_4 + 2(m_1 + m_2 + m_3 + m_4) + 4p}{16}$$

m midpoint of edge

p old "vertex point"

Convergence: Overall Shape and Creases

Loop with Sharp Creases



Catmull-Clark with Sharp Creases

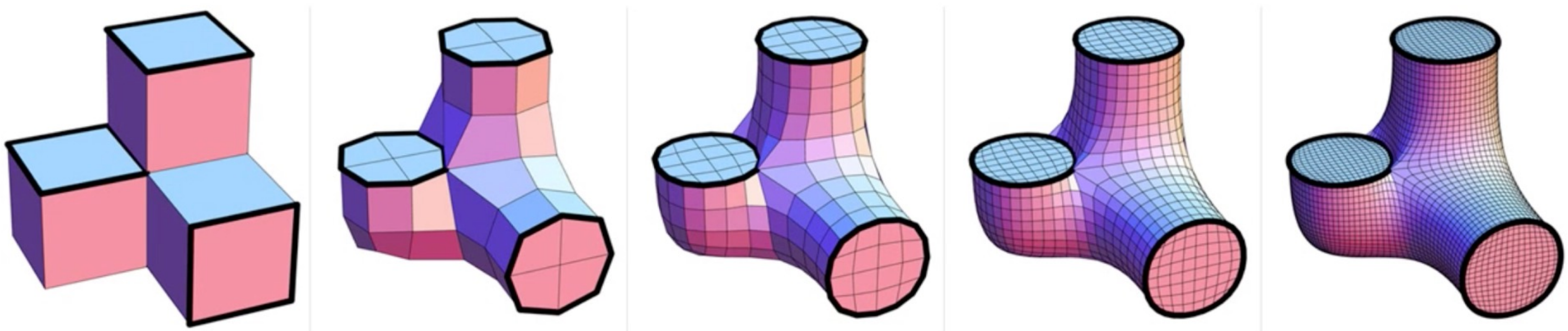


Figure from: Hakenberg et al. Volume Enclosed by Subdivision Surfaces with Sharp Creases

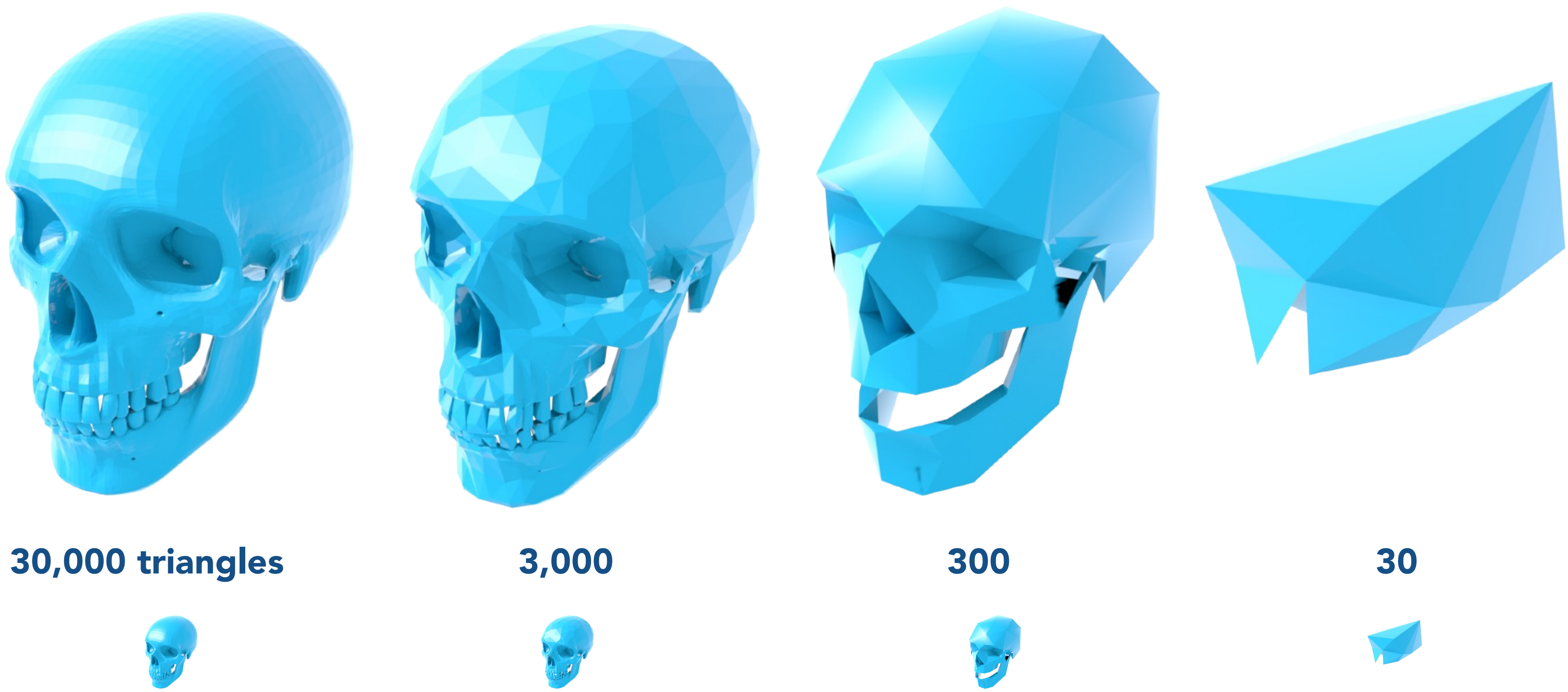
Subdivision in Action (Pixar's "Geri's Game")

https://vimeo.com/168651722

Mesh Simplification

Mesh Simplification

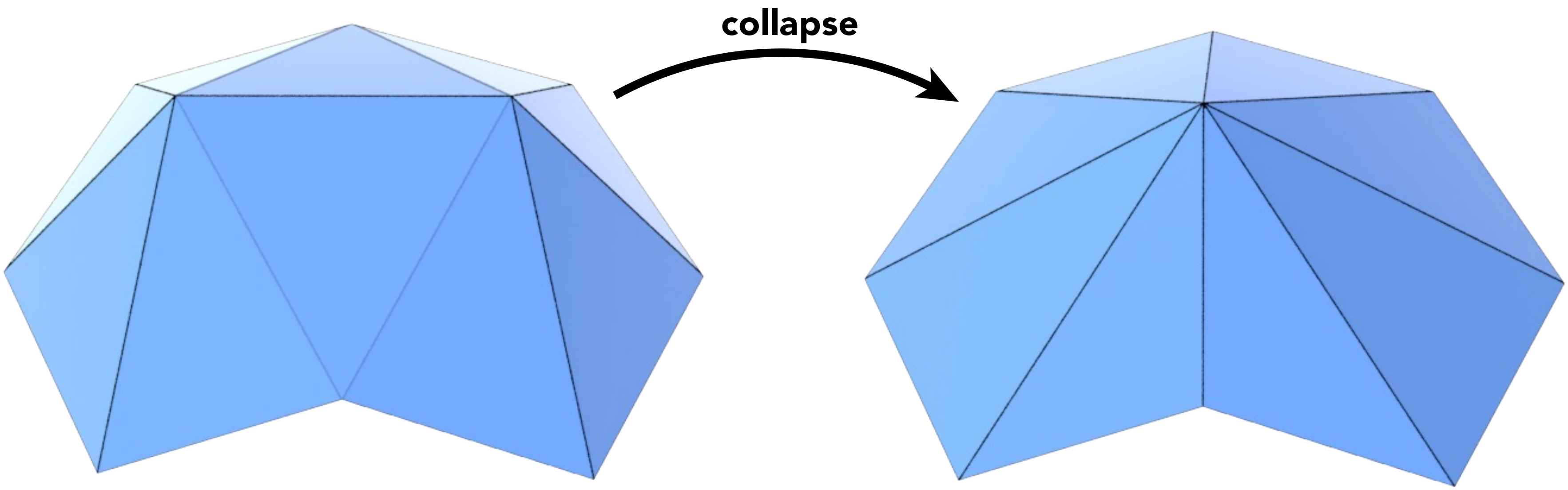
Goal: reduce number of mesh elements while maintaining the overall shape



How to compute?

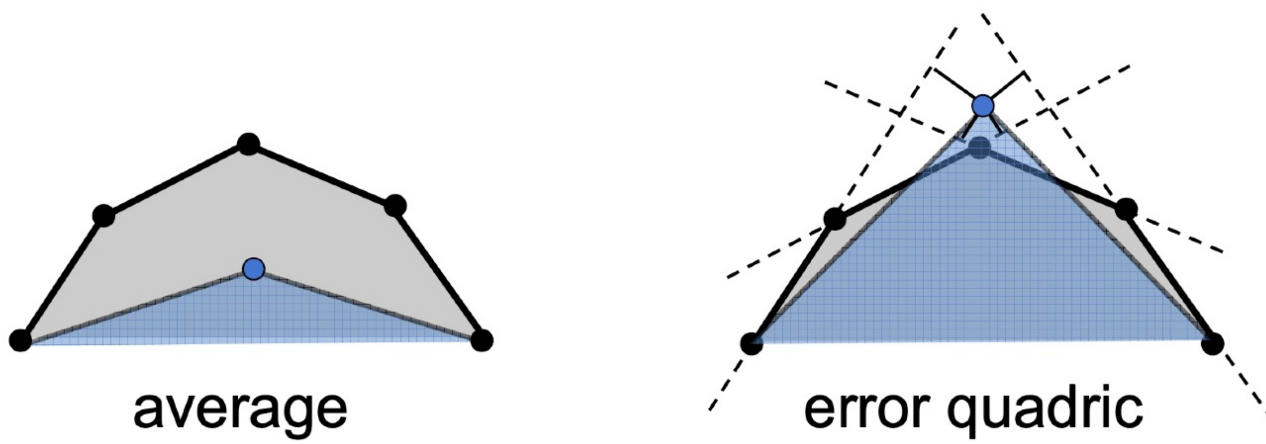
Collapsing An Edge

- Suppose we simplify a mesh using edge collapsing



Quadric Error Metrics

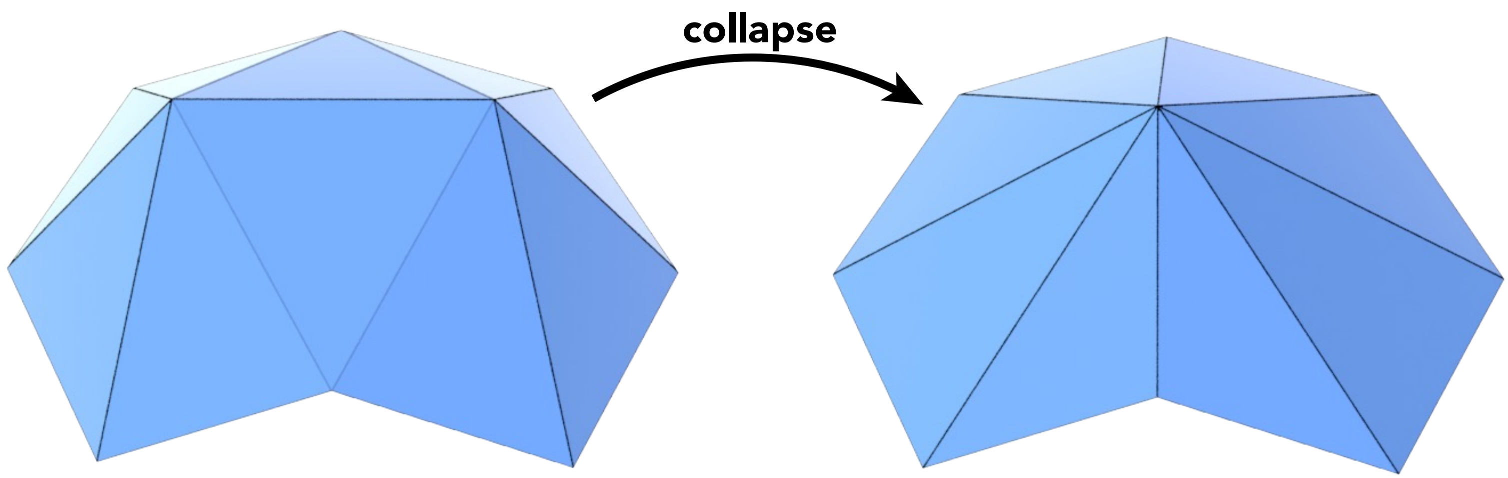
- How much geometric error is introduced by simplification?
- Not a good idea to perform local averaging of vertices
- Quadric error: new vertex should minimize its sum of square distance (L2 distance) to previously related triangle planes!



http://graphics.stanford.edu/courses/cs468-10-fall/LectureSlides/08_Simplification.pdf

Quadric Error of Edge Collapse

- How much does it cost to collapse an edge?
- Idea: compute edge midpoint, measure quadric error



- Better idea: choose point that minimizes quadric error
- More details: Garland & Heckbert 1997.

Simplification via Quadric Error

Iteratively collapse edges

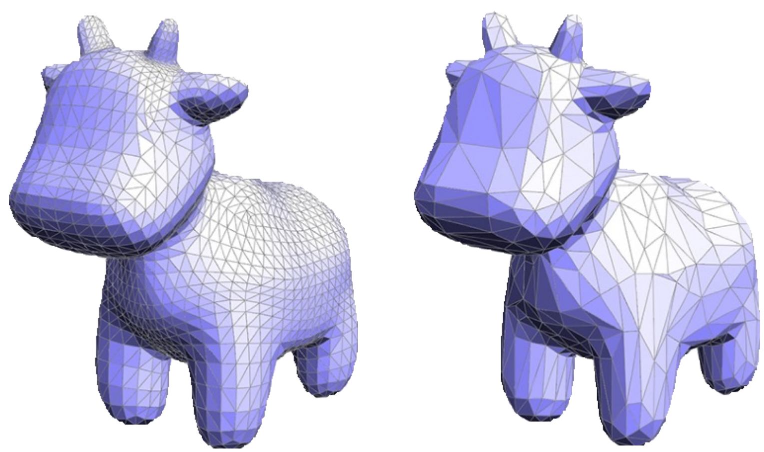
Which edges? Assign score with quadric error metric*

- approximate distance to surface as sum of distances to planes containing triangles
- iteratively collapse edge with smallest score
- greedy algorithm… great results!

* (Garland & Heckbert 1997)

Quadric Error Mesh Simplification

5,804 **994** **532** **248** **64**



Garland and Heckbert '97

Thank you!

(And thank Prof. Lingqi Yan, Prof. Ravi Ramamoorthi and Prof. Ren Ng for many of the slides!)