



Subdivision Surfaces

COS 426

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
 - Application specific

3D Object Representations

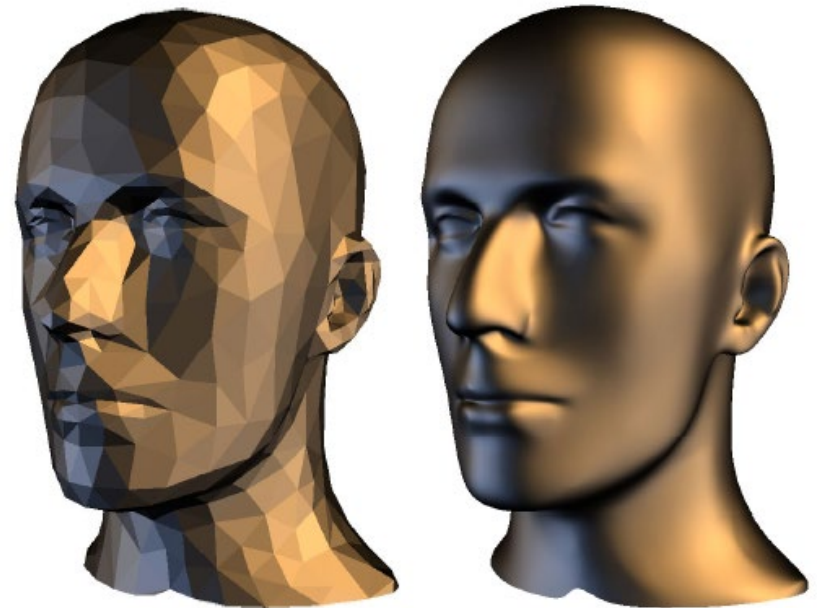


- Raw data
 - o Voxels
 - o Point cloud
 - o Range image
 - o Polygons
- **Surfaces**
 - o Mesh
 - **Subdivision**
 - o Parametric
 - o Implicit
- Solids
 - o Octree
 - o BSP tree
 - o CSG
 - o Sweep
- High-level structures
 - o Scene graph
 - o Application specific

Subdivision Surfaces



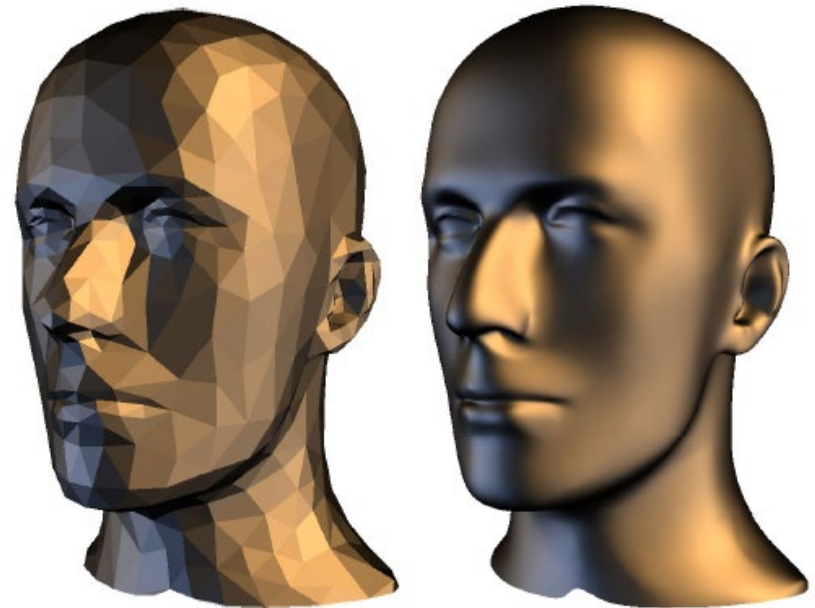
- What makes a good surface representation?
 - o Accurate
 - o Concise
 - o Intuitive specification
 - o Local support
 - o Affine invariant
 - o Arbitrary topology
 - o Guaranteed continuity
 - o Natural parameterization
 - o Efficient display
 - o Efficient intersections



Subdivision Surfaces



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





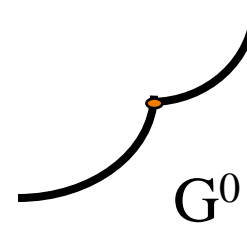
Continuity

- A curve / surface with G^k continuity has a continuous k -th derivative

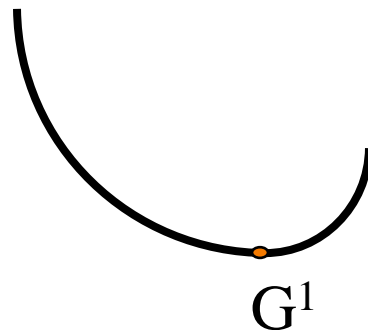
No continuity: “ G^{-1} ”



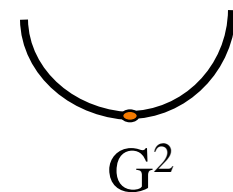
G^0



G^1

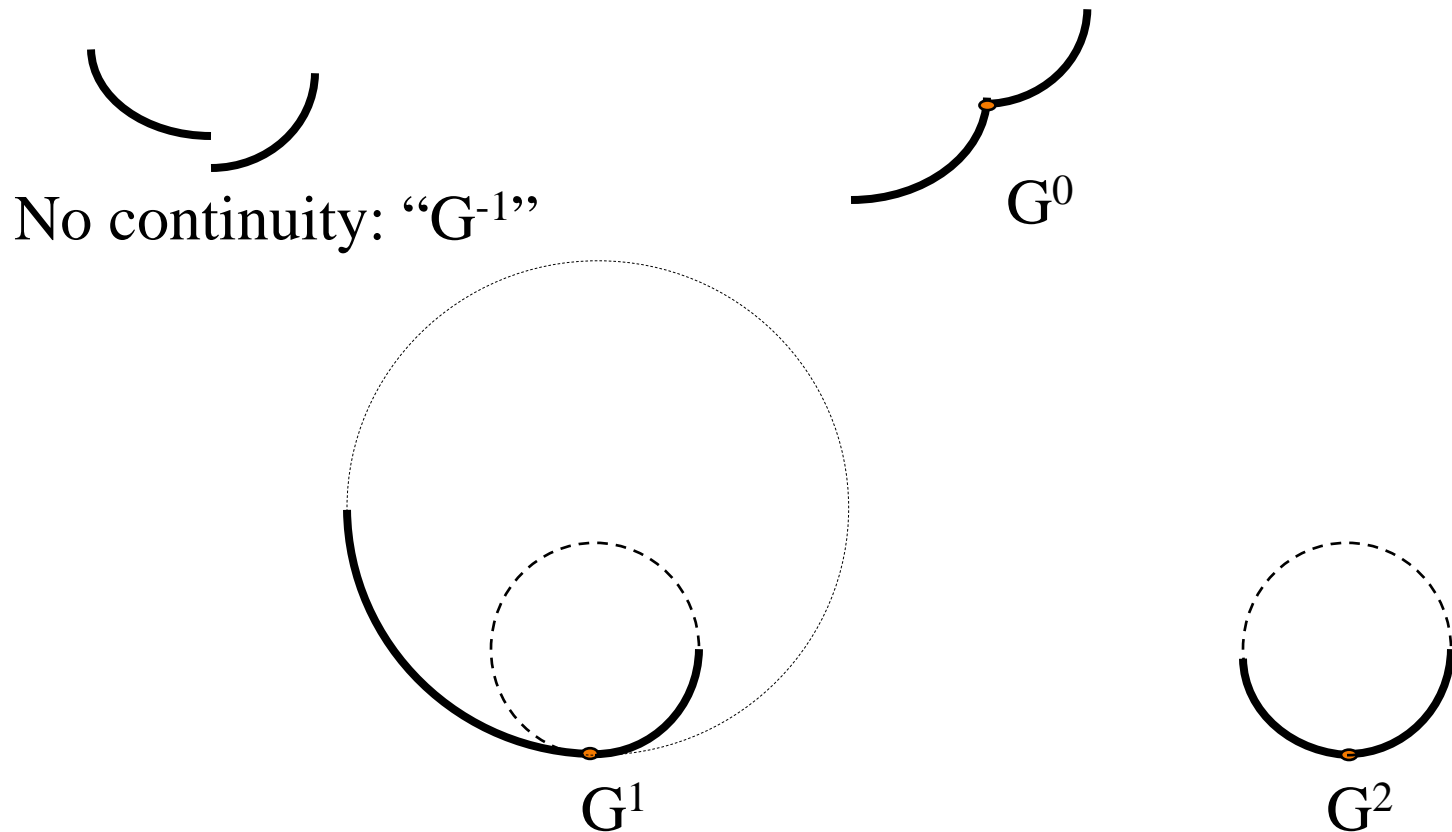


G^2



Continuity

- A curve / surface with G^k continuity has a continuous k -th derivative





Parametric Continuity

- Curve $x = f(t)$, $y = g(t)$ is C^k continuous iff

$$\frac{d^k f}{dt^k} \text{ and } \frac{d^k g}{dt^k} \text{ are continuous}$$

- For surfaces: $x = f(s,t)$, $y = g(s,t)$, $z = h(s,t)$

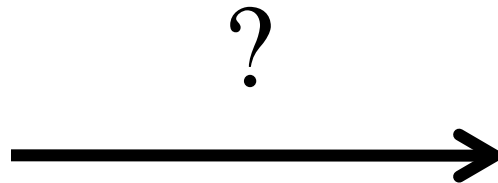
$$C^k \text{ requires continuous } \frac{\partial^k f}{\partial s^k}, \frac{\partial^k f}{\partial t^k}, \frac{\partial^k g}{\partial s^k}, \frac{\partial^k g}{\partial t^k}, \frac{\partial^k h}{\partial s^k}, \frac{\partial^k h}{\partial t^k}$$

- Often easier to check / prove things about than G^k continuity



Subdivision

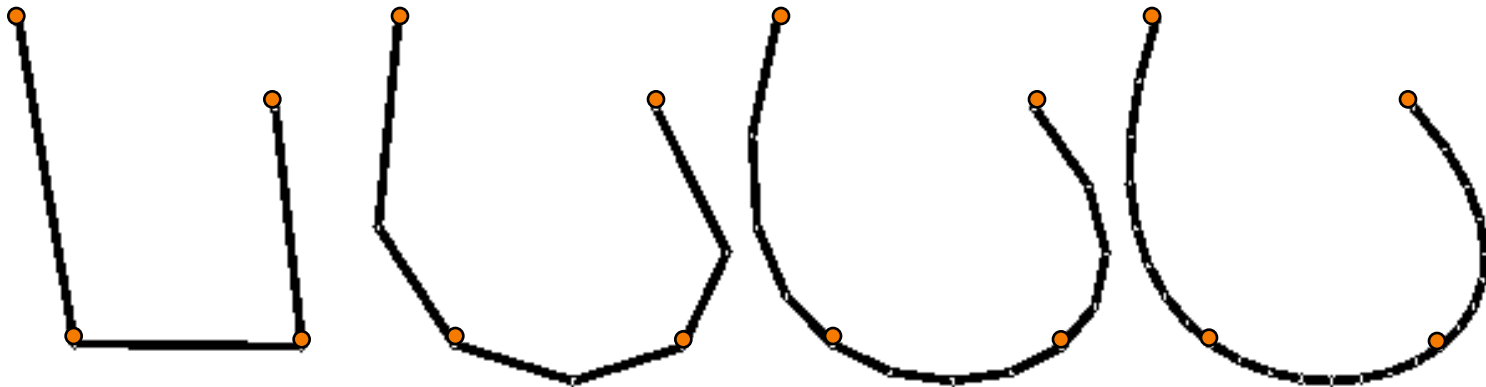
- How do you make a curve with guaranteed continuity?



Subdivision

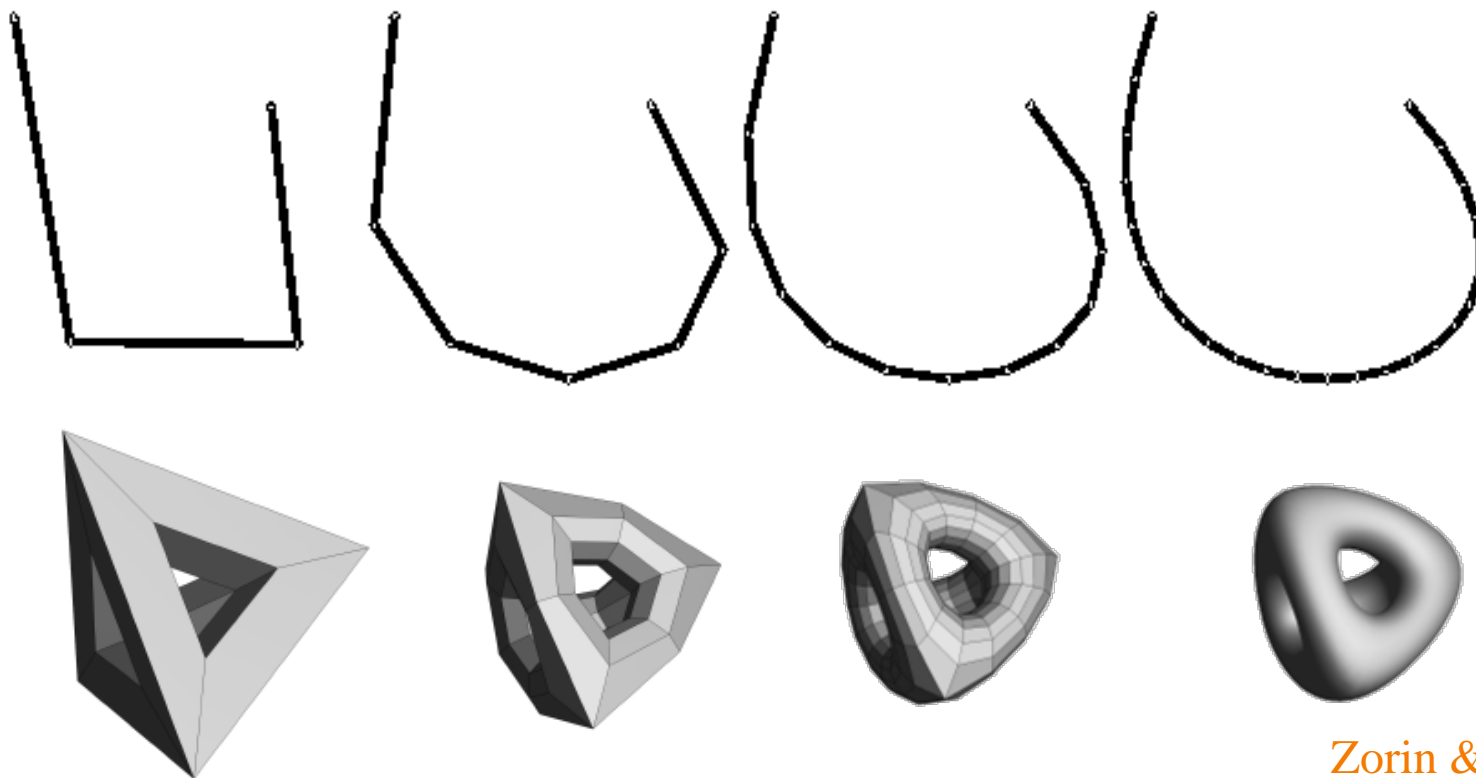


- How do you make a curve with guaranteed continuity? ...



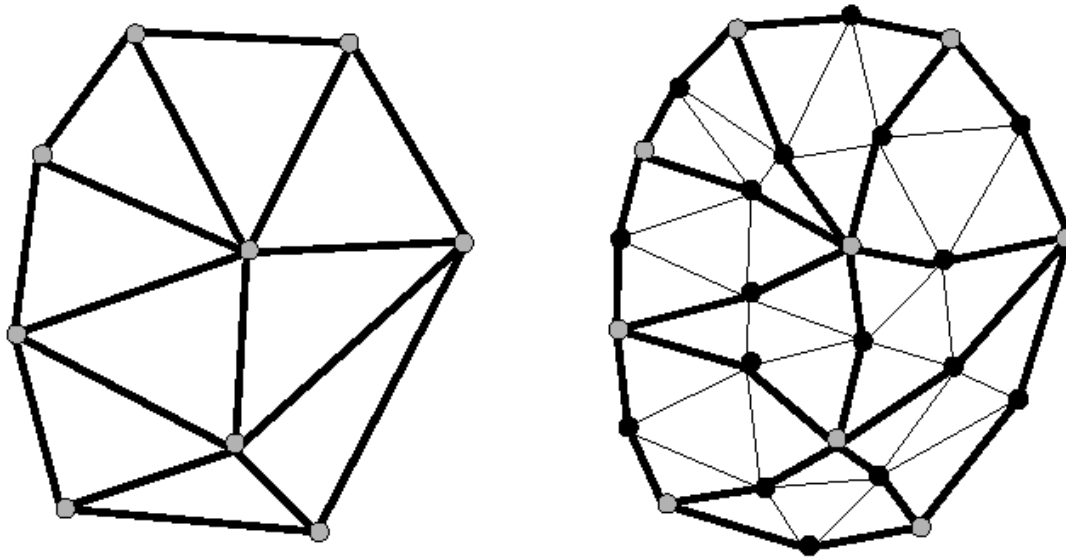
Subdivision

- How do you make a surface with guaranteed continuity?



Subdivision Surfaces

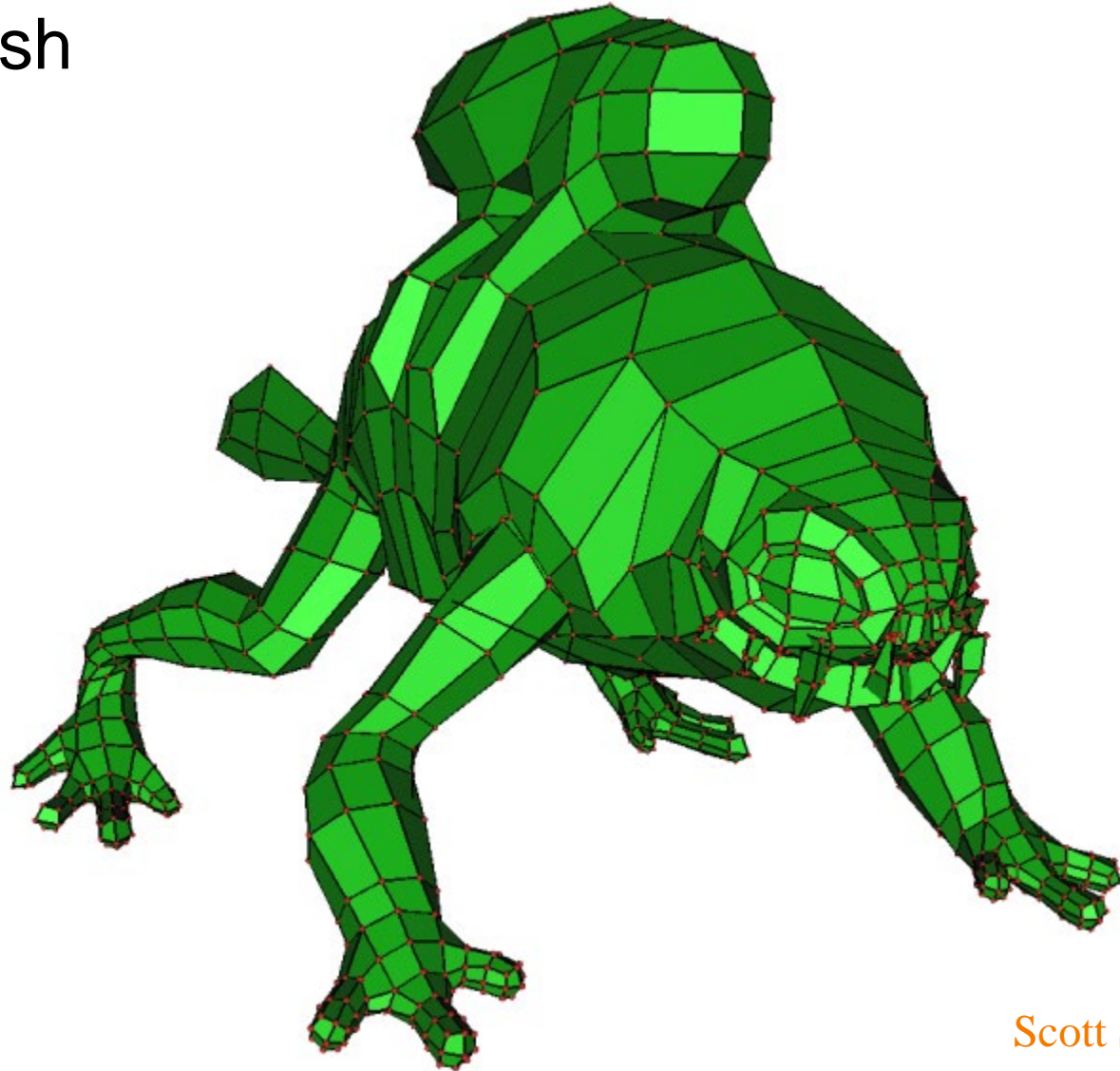
- Repeated application of
 - Topology refinement (splitting faces)
 - Geometry refinement (weighted averaging)



Subdivision Surfaces – Examples



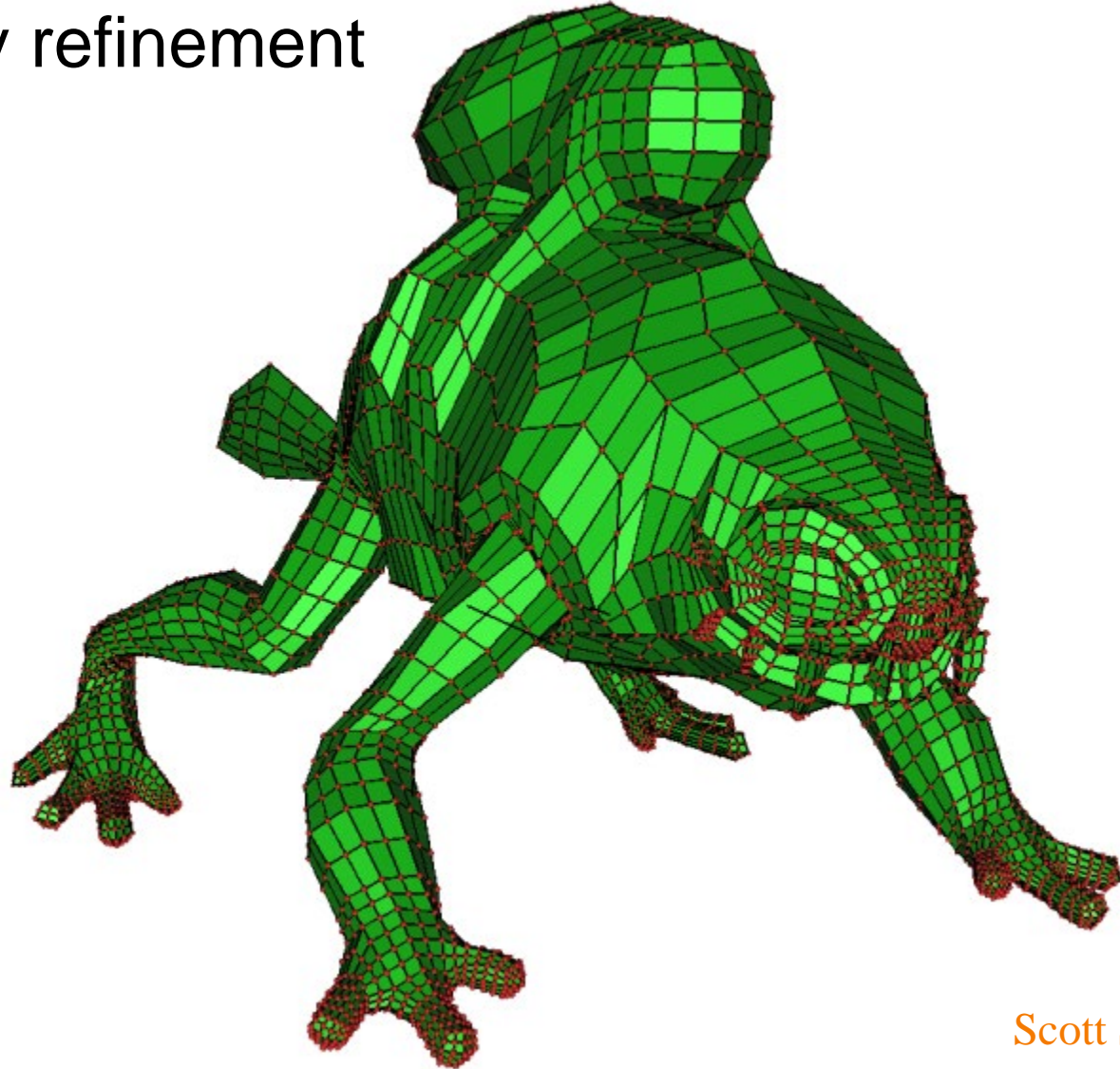
- Base mesh



Subdivision Surfaces – Examples



- Topology refinement

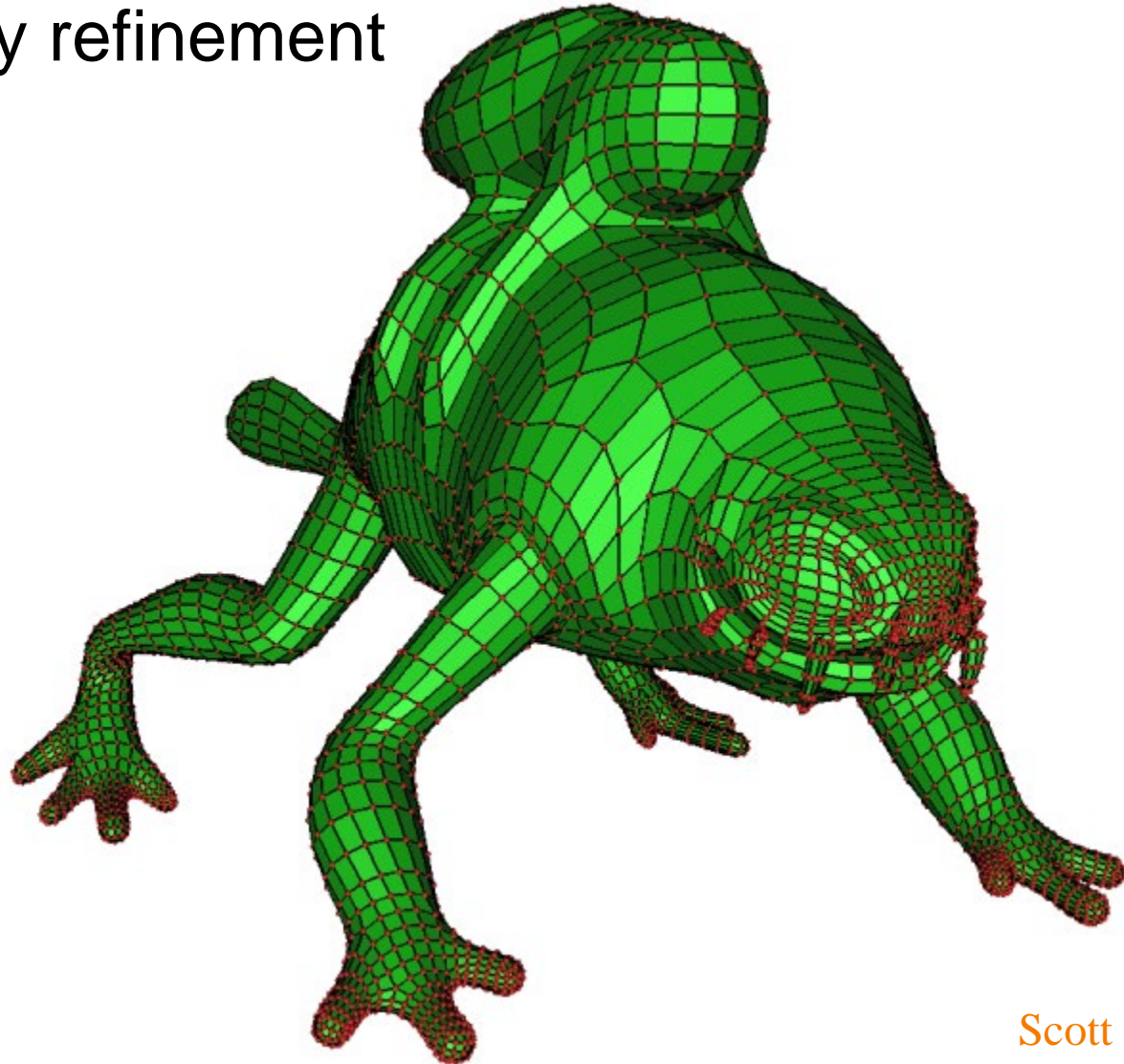


Scott Schaefer

Subdivision Surfaces – Examples



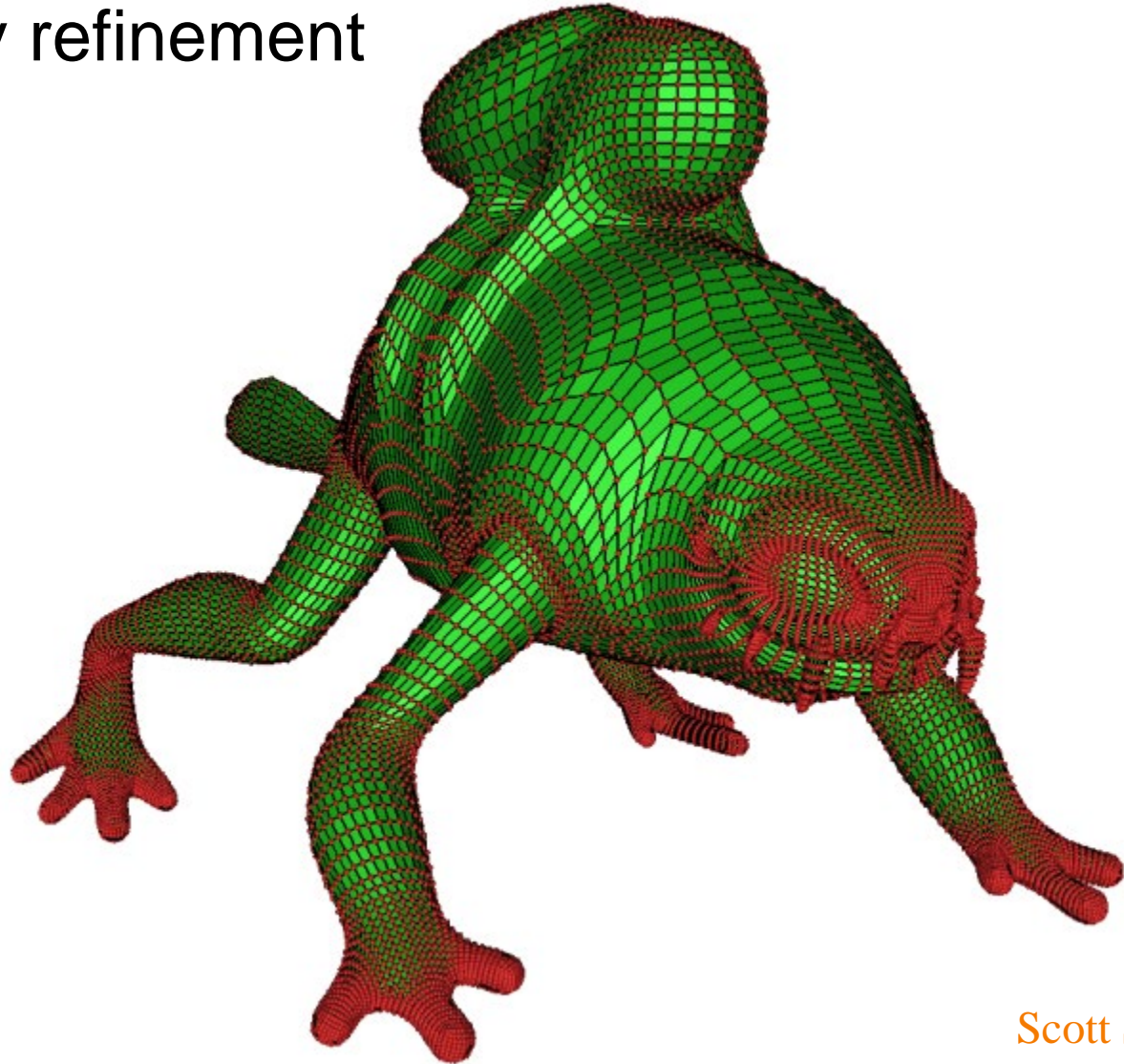
- Geometry refinement



Subdivision Surfaces – Examples



- Topology refinement

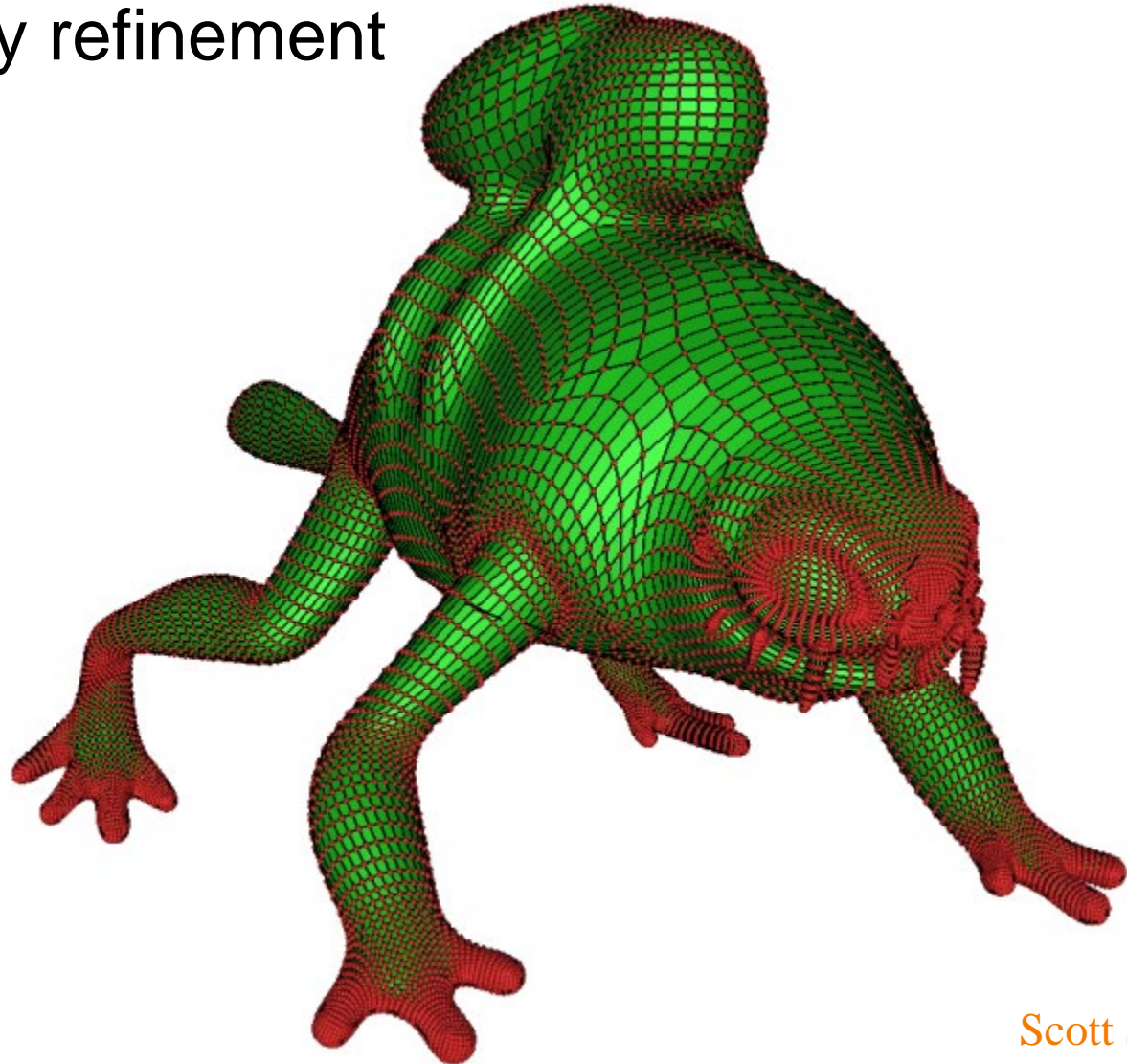


Scott Schaefer

Subdivision Surfaces – Examples



- Geometry refinement

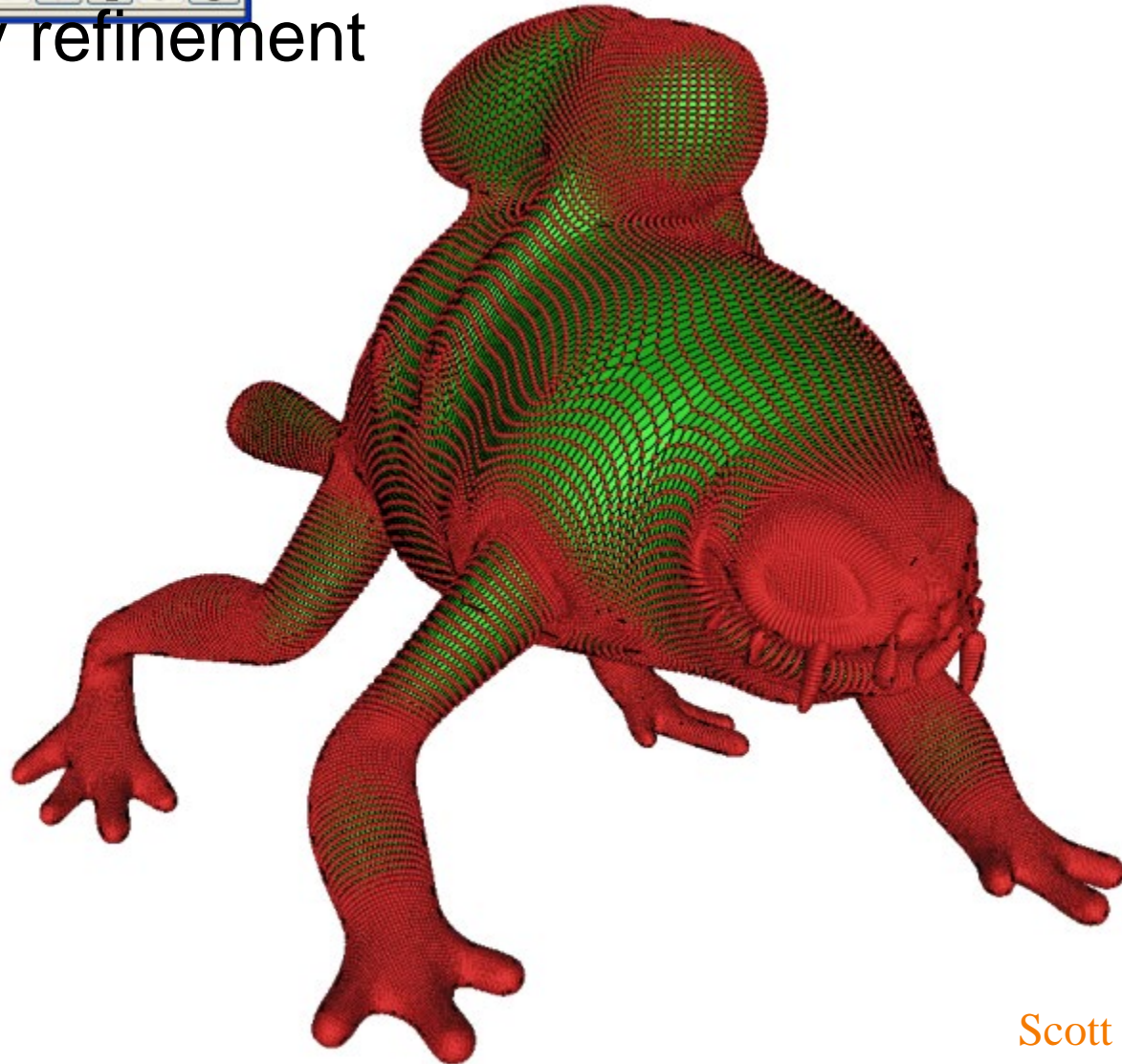


Scott Schaefer

Subdivision Surfaces – Examples



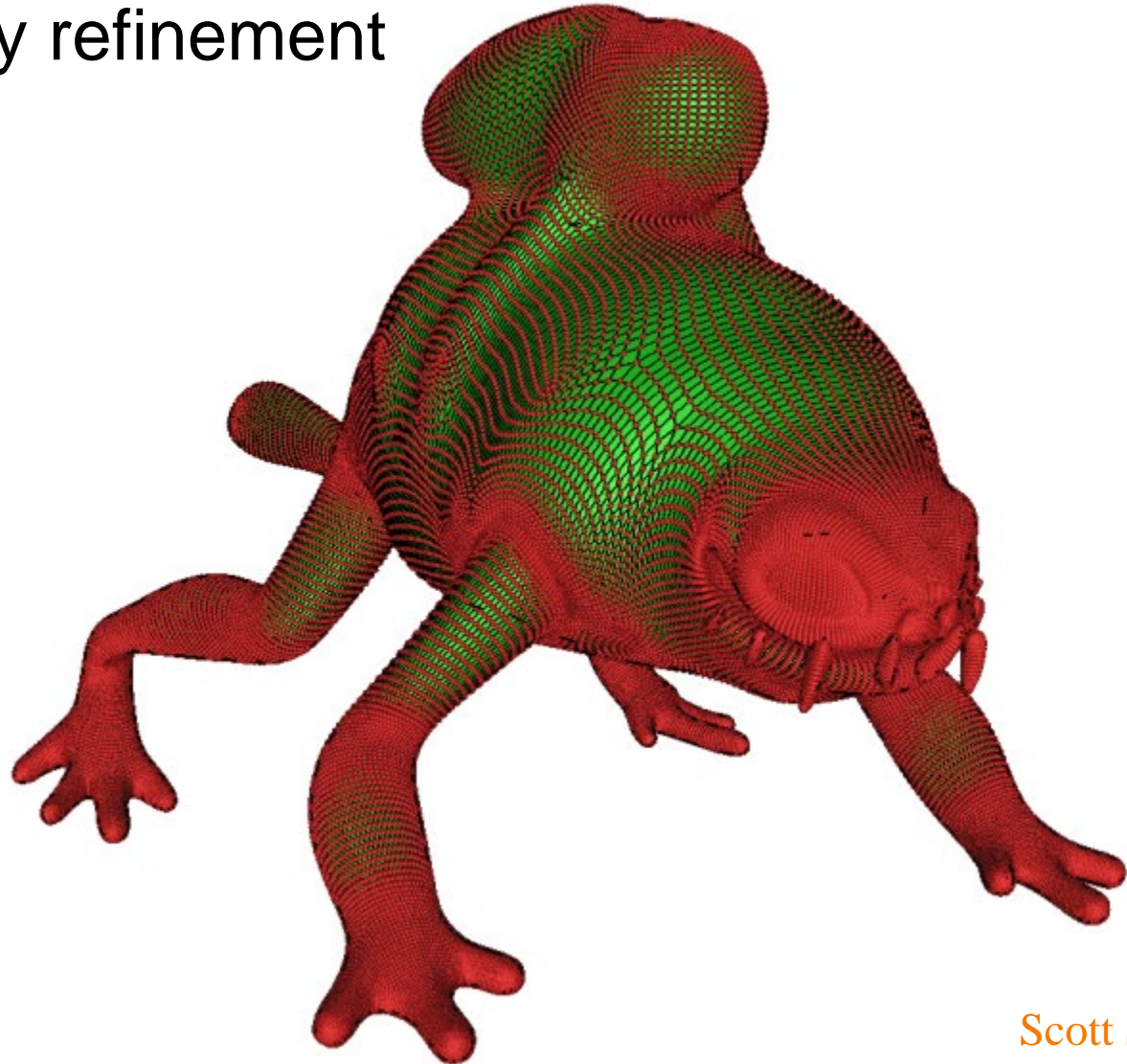
- Topology  refinement



Subdivision Surfaces – Examples



- Geometry refinement

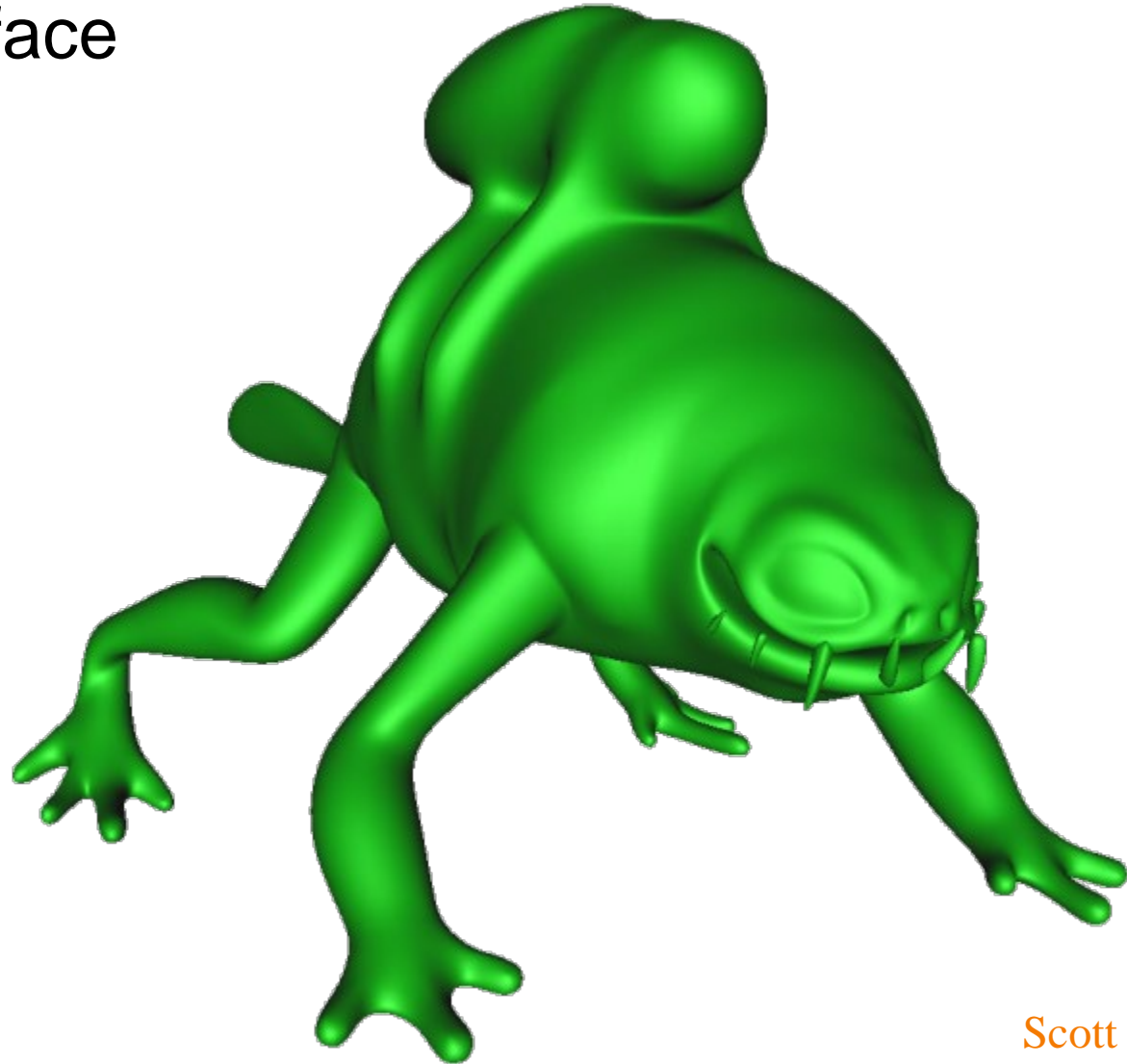


Scott Schaefer

Subdivision Surfaces – Examples



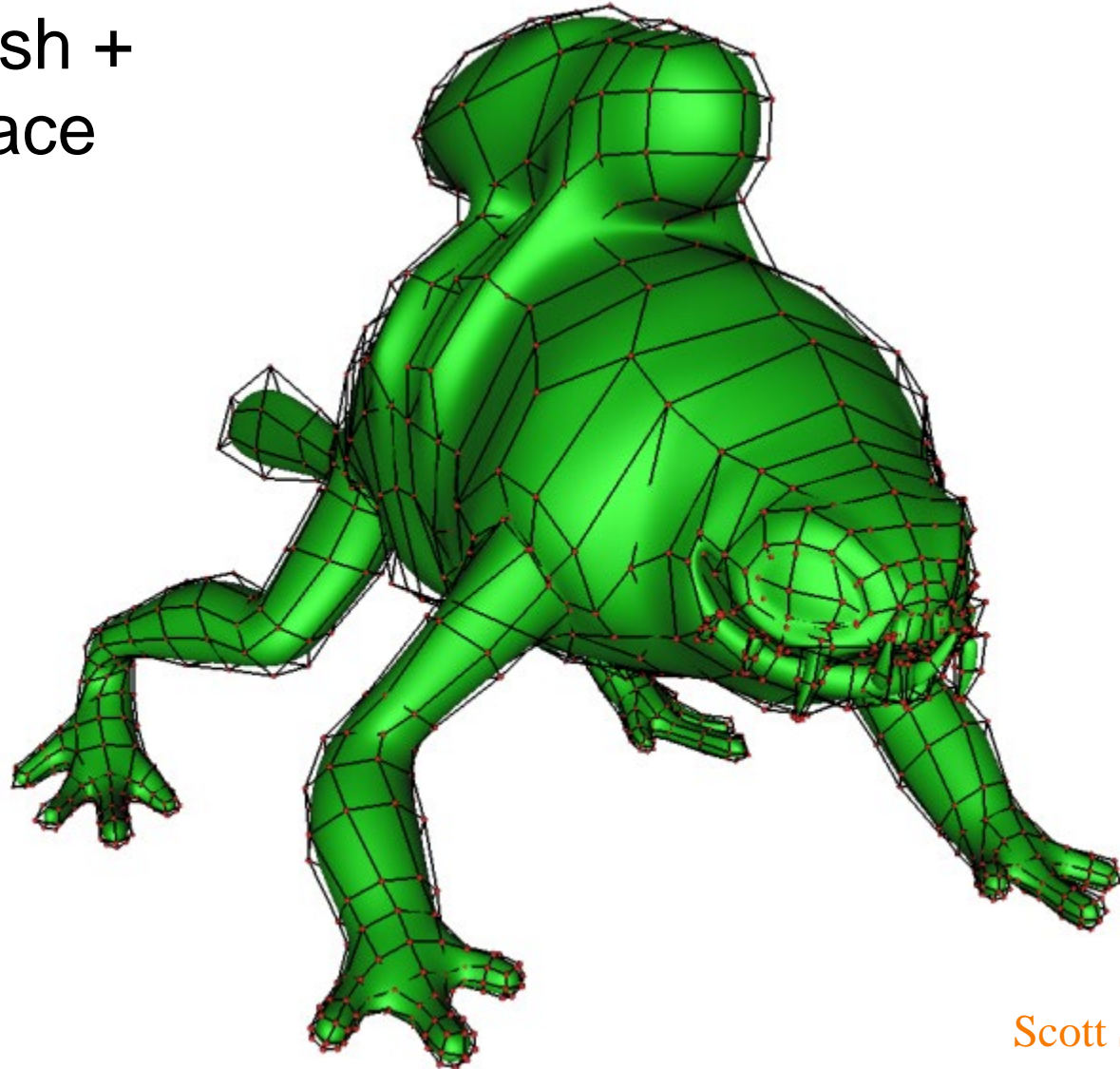
- Limit surface



Subdivision Surfaces – Examples

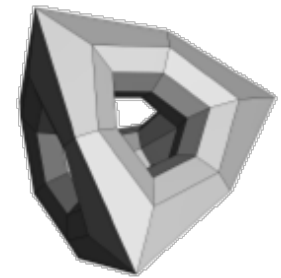
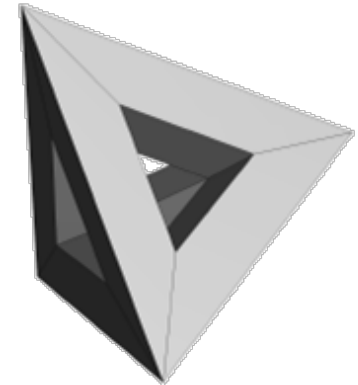


- Base mesh + limit surface



Design of Subdivision Rules

- What types of input?
 - Quad meshes, triangle meshes, etc.
- How to refine topology?
 - Simple implementations
- How to refine geometry?
 - Smoothness guarantees in limit surface
 - » Continuity (C^0 , C^1 , C^2 , ...?)
 - Provable relationships between limit surface and original control mesh
 - » Interpolation of vertices?



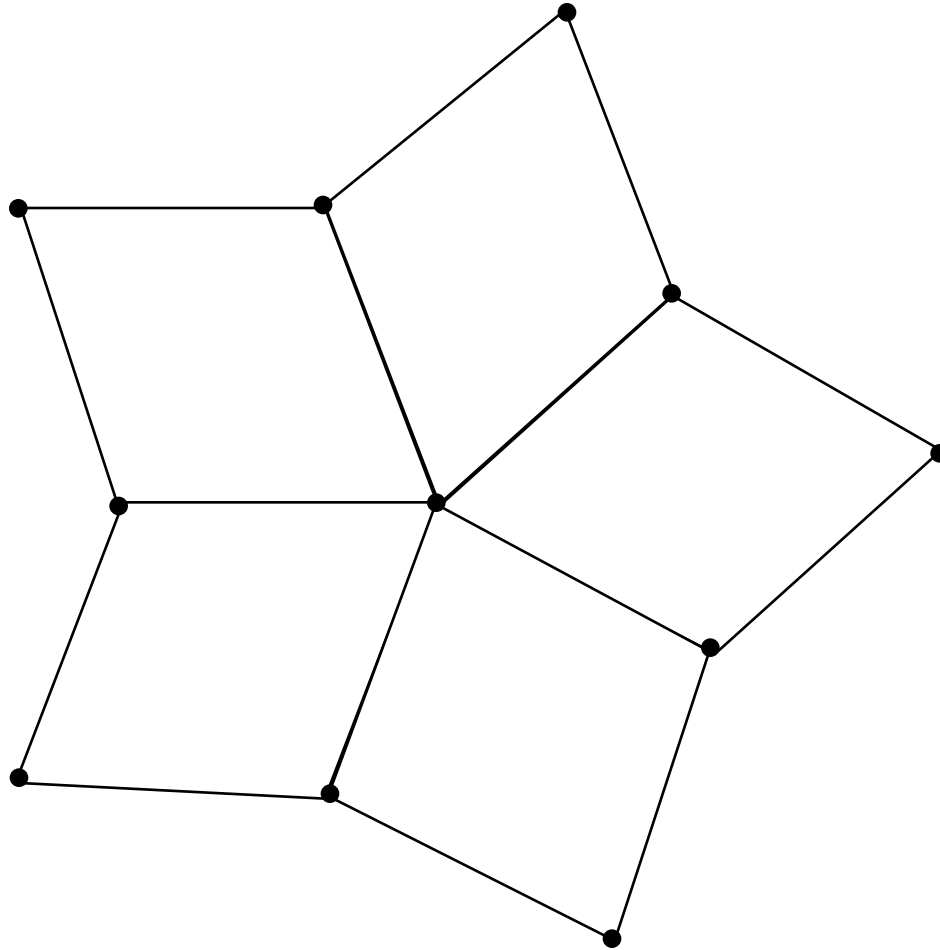


Linear Subdivision

- Type of input
 - Quad mesh -- four-sided polygons (*quads*)
 - Any number of quads may touch each vertex
- Topology refinement rule
 - Split every quad into four at midpoints
- Geometry refinement rule
 - Average vertex positions

This is a simple example to demonstrate how subdivision schemes work

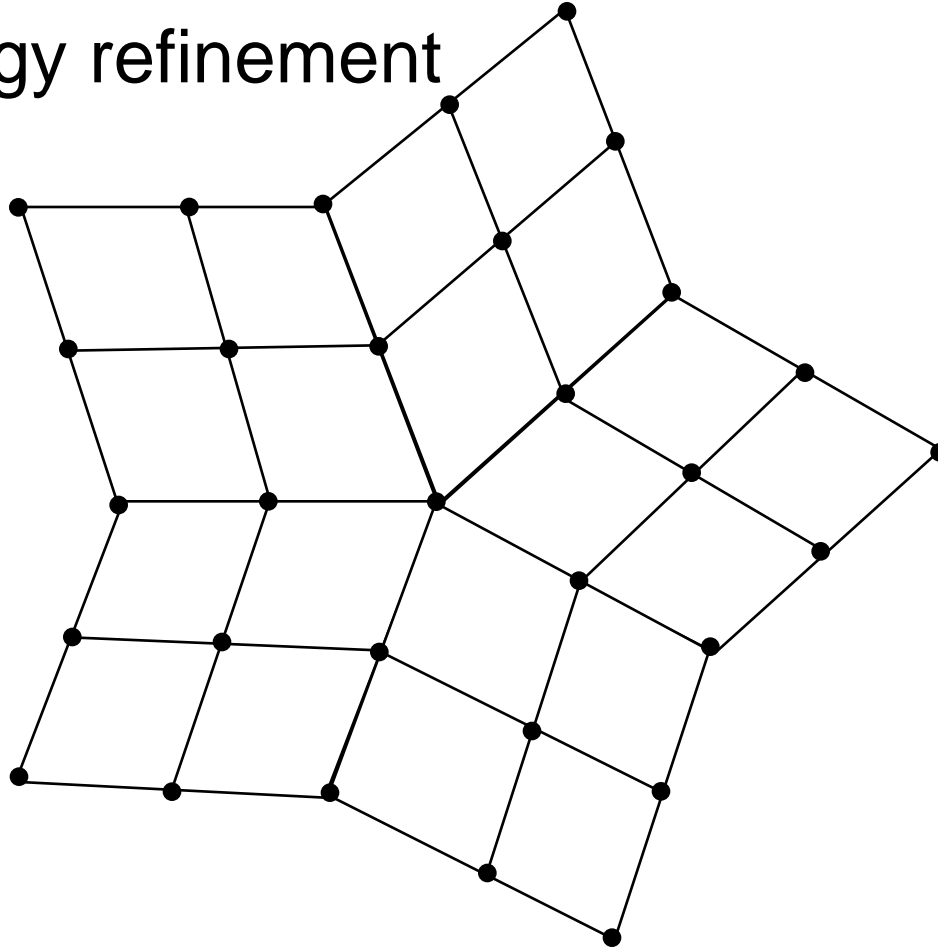
Linear Subdivision



Linear Subdivision

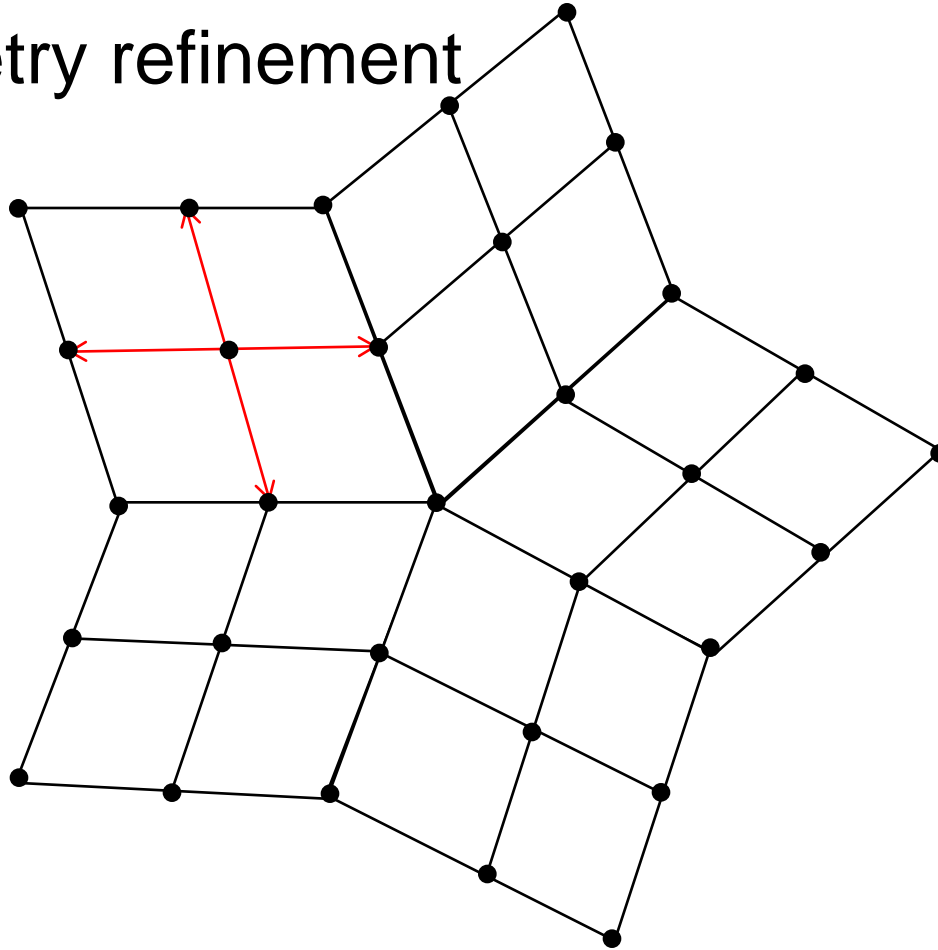


- Topology refinement



Linear Subdivision

- Geometry refinement





Linear Subdivision

LinearSubivision (F_0, V_0, k)

for $i = 1 \dots k$ levels

$(F_i, V_i) = \text{RefineTopology}(F_{i-1}, V_{i-1})$

$\text{RefineGeometry}(F_i, V_i)$

return (F_k, V_k)



Linear Subdivision

RefineTopology (F, V)

$newV = V$

$newF = \{ \}$

for each face F_i

 Insert new vertex c at centroid of F_i into $newV$

 for $j = 1$ to 4

 Insert/lookup in $newV$ new vertex e_j at
 centroid of each edge ($F_{i,j}, F_{i,j+1}$)

 for $j = 1$ to 4

 Insert new face ($F_{i,j}, e_j, c, e_{j-1}$) into $newF$

return ($newF, newV$)



Linear Subdivision

RefineGeometry(F , V)

$newV = 0 * V$

wt = array of 0 whose size is number of vertices

$newF = F$

for each face F_i

$cent$ = centroid for F_i

$newV[F_i] += cent$ // syntax: repeat for all vtx indices in F_i

$wt[F_i] += 1$ // syntax: repeat for all vtx indices in F_i

for each vertex $newV[i]$

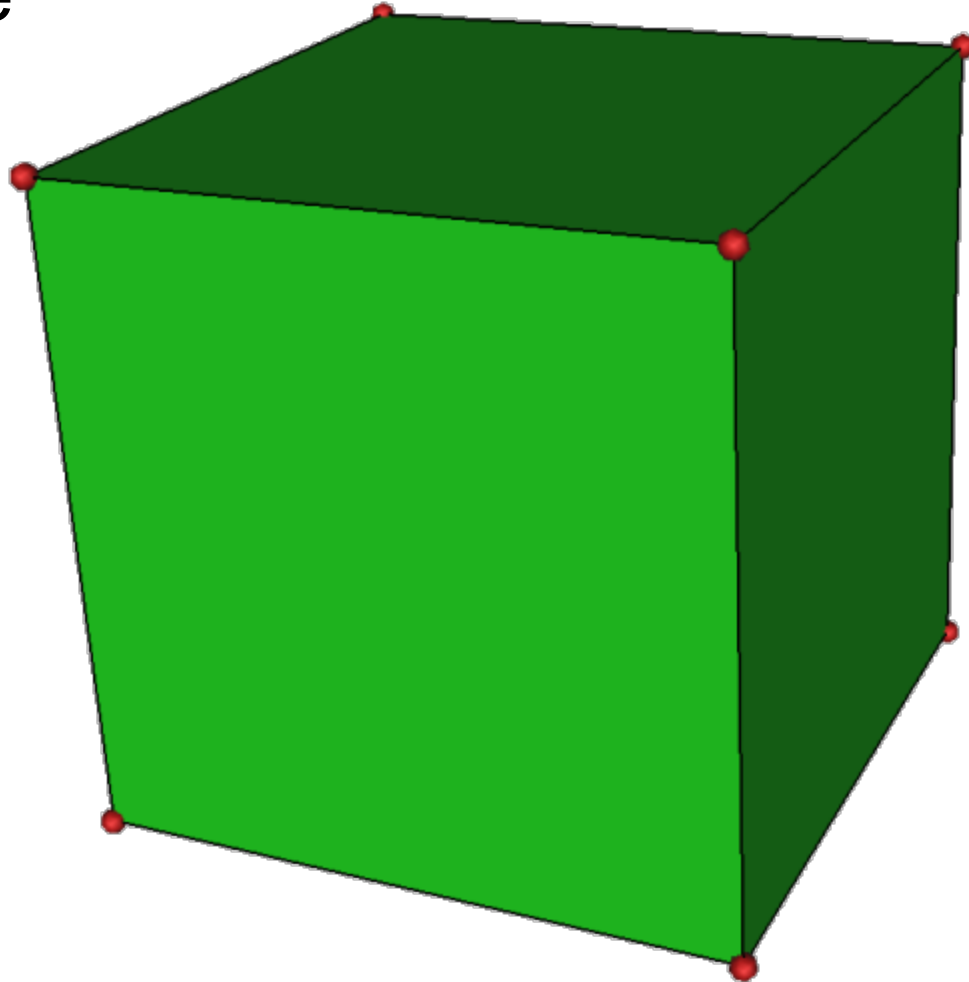
$newV[i] /= wt[i]$

return ($newF$, $newV$)

Linear Subdivision



- Example

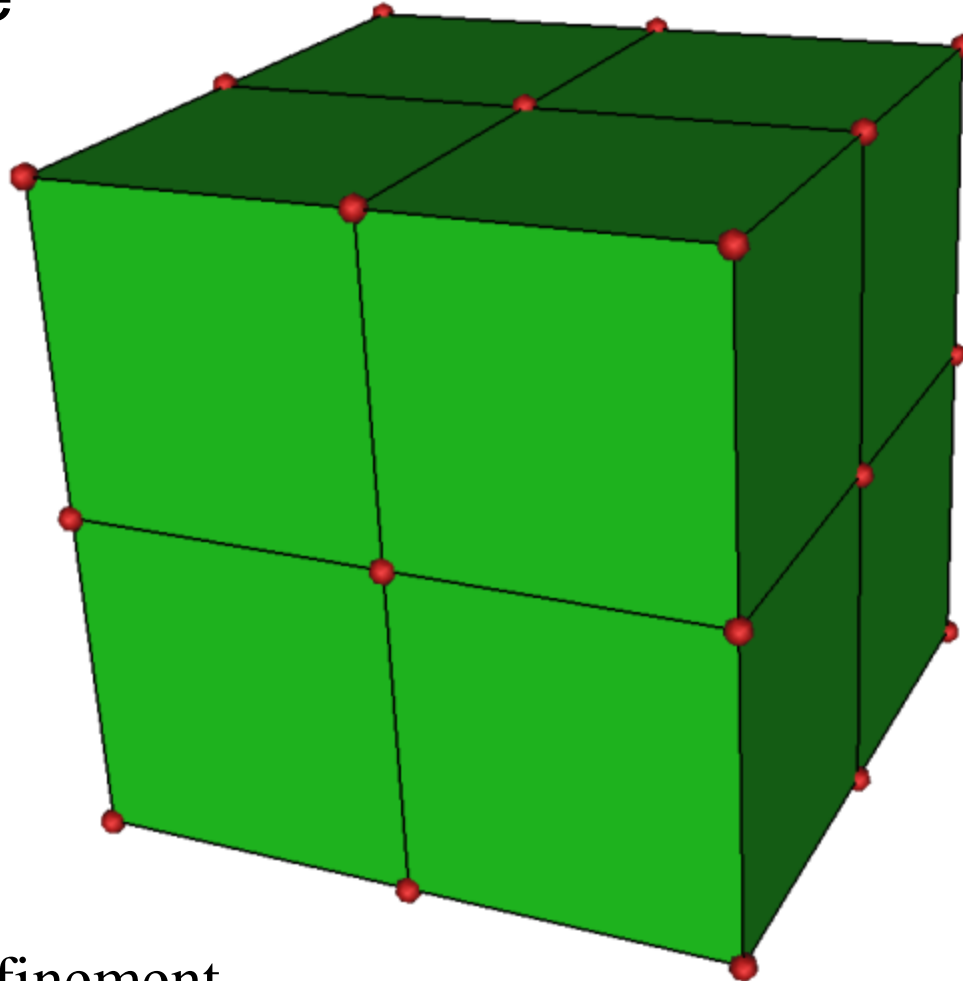


Input mesh

Scott Schaefer

Linear Subdivision

- Example



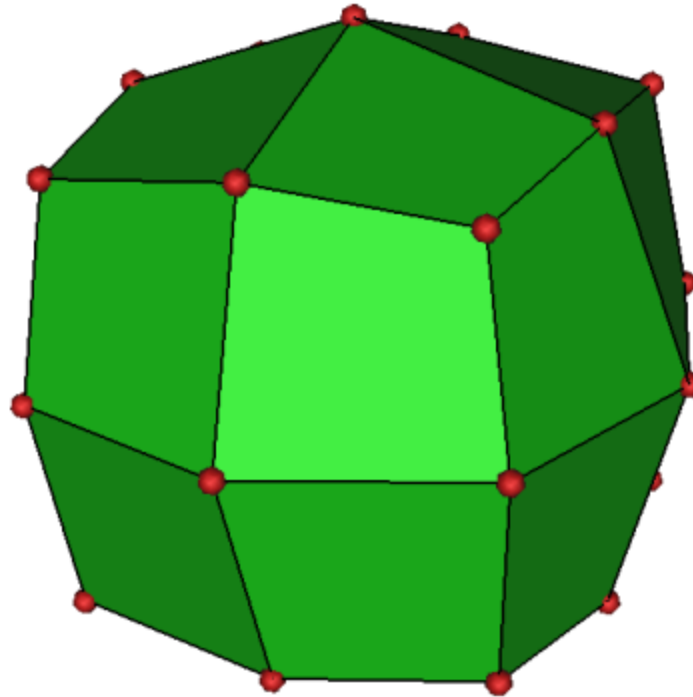
Topology refinement

Scott Schaefer

Linear Subdivision



- Example



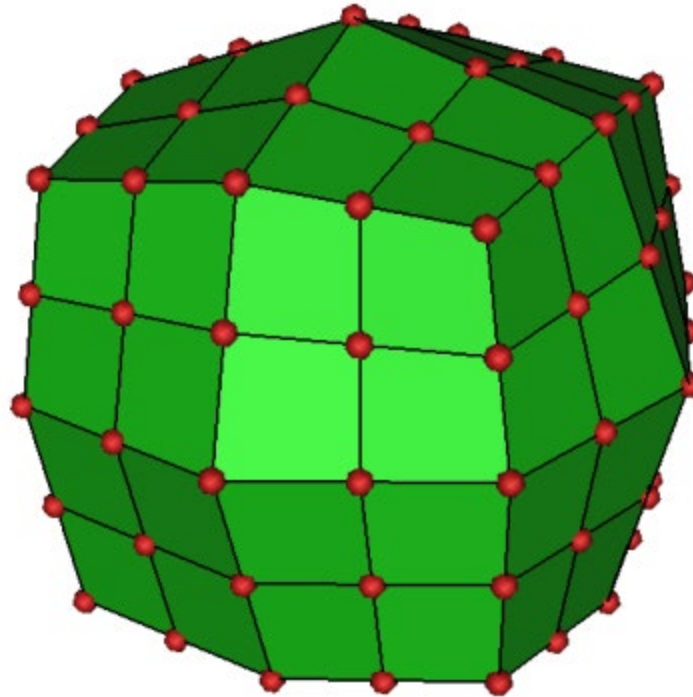
Geometry refinement

Scott Schaefer

Linear Subdivision



- Example



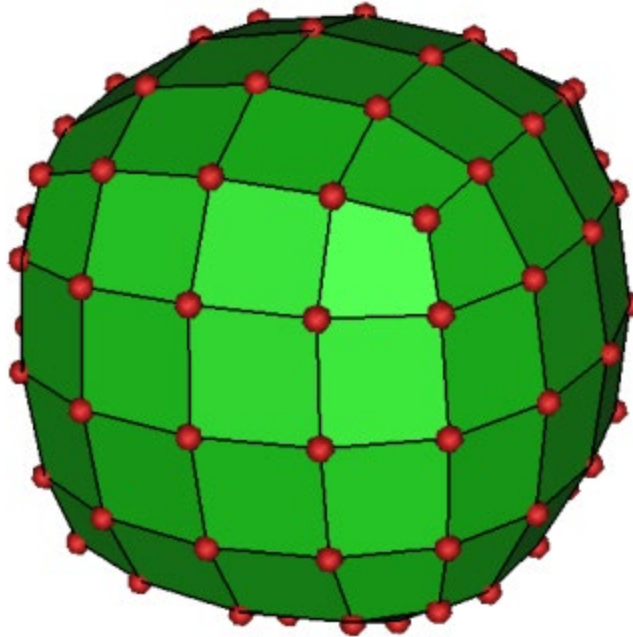
Topology refinement

Scott Schaefer

Linear Subdivision



- Example



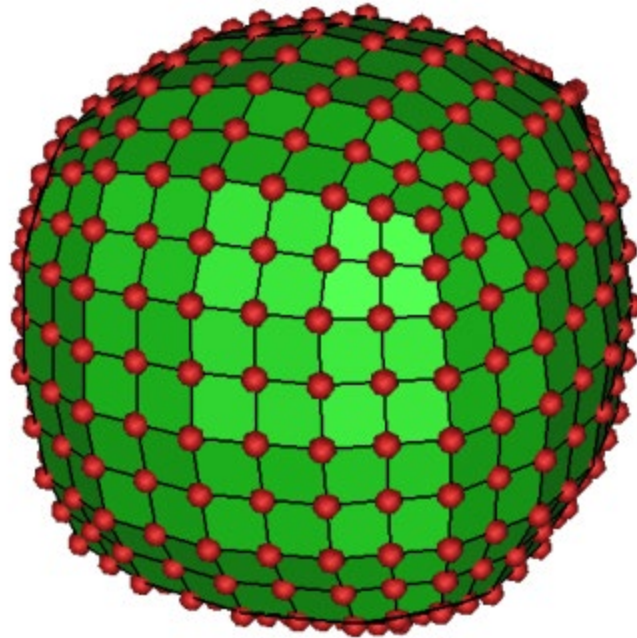
Geometry refinement

Scott Schaefer

Linear Subdivision



- Example



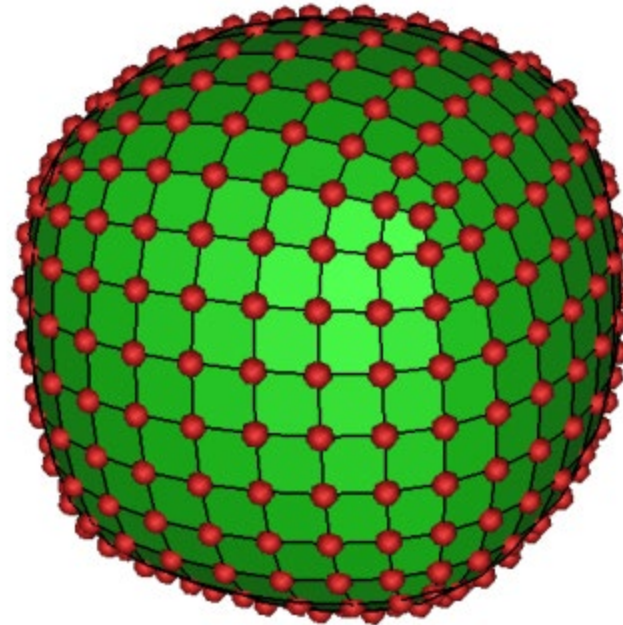
Topology refinement

Scott Schaefer

Linear Subdivision



- Example



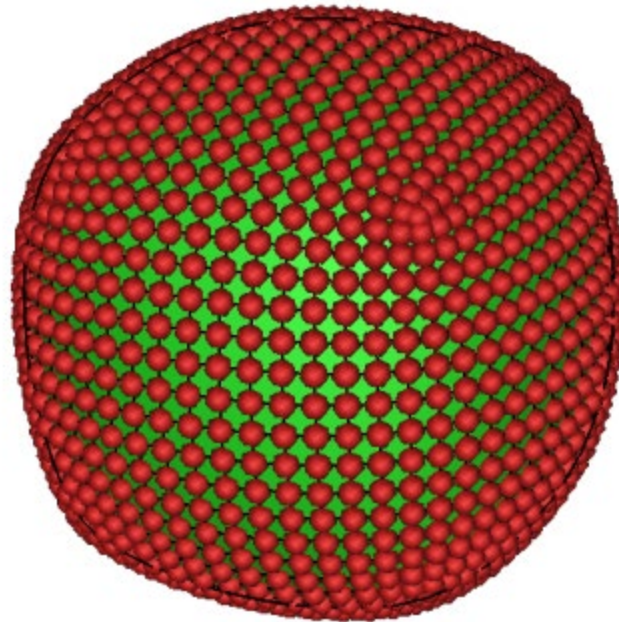
Geometry refinement

Scott Schaefer

Linear Subdivision



- Example



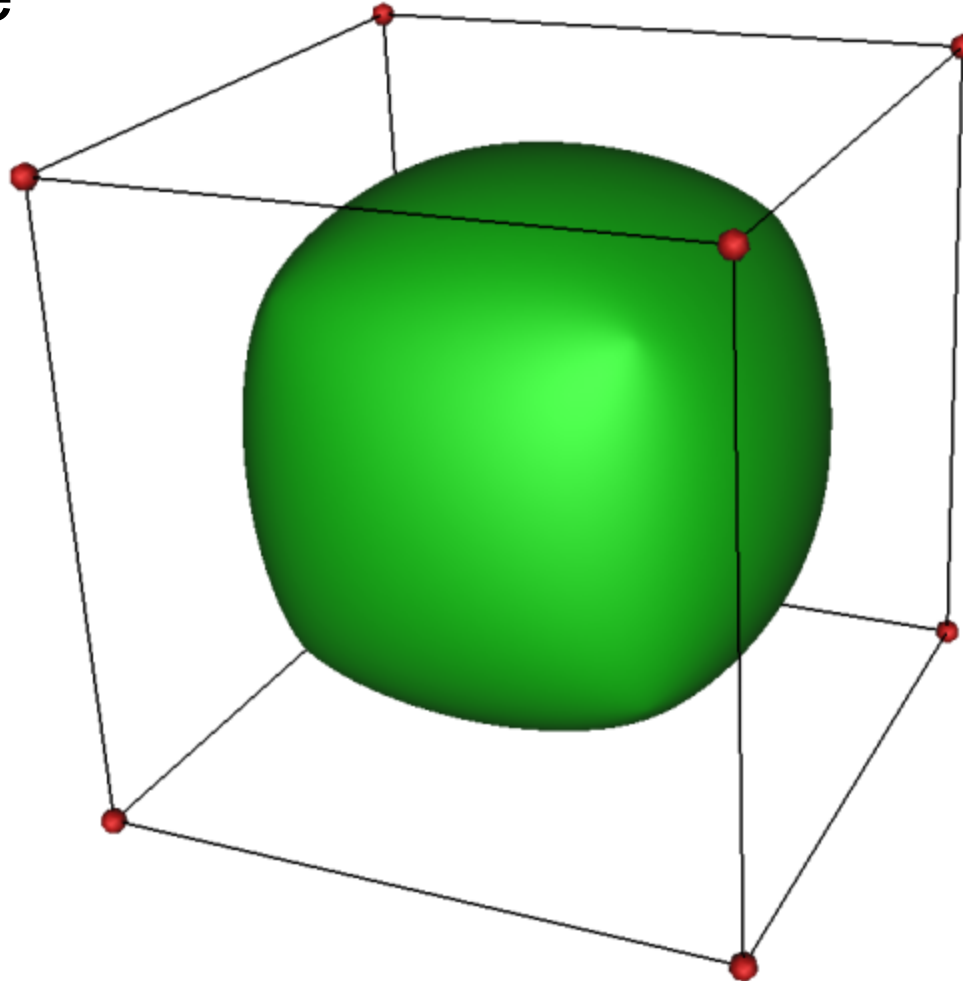
Topology refinement

Scott Schaefer

Linear Subdivision



- Example

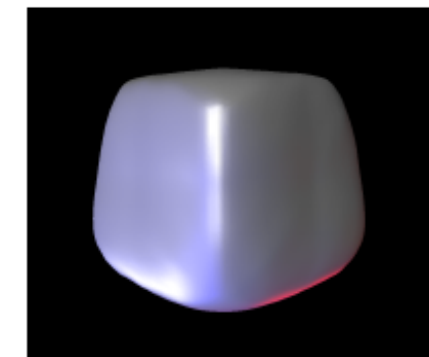
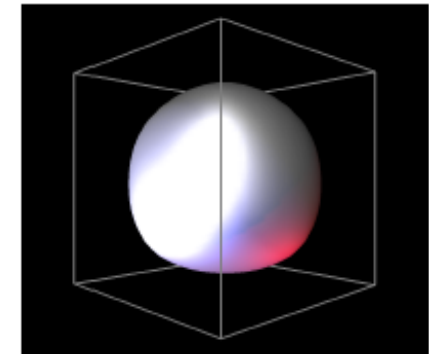
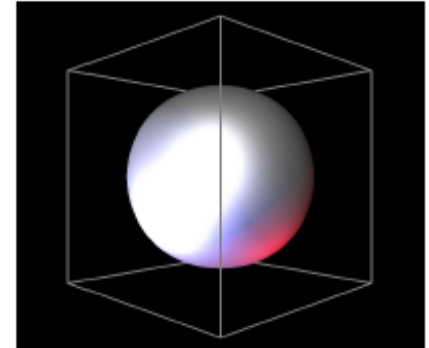


Final result

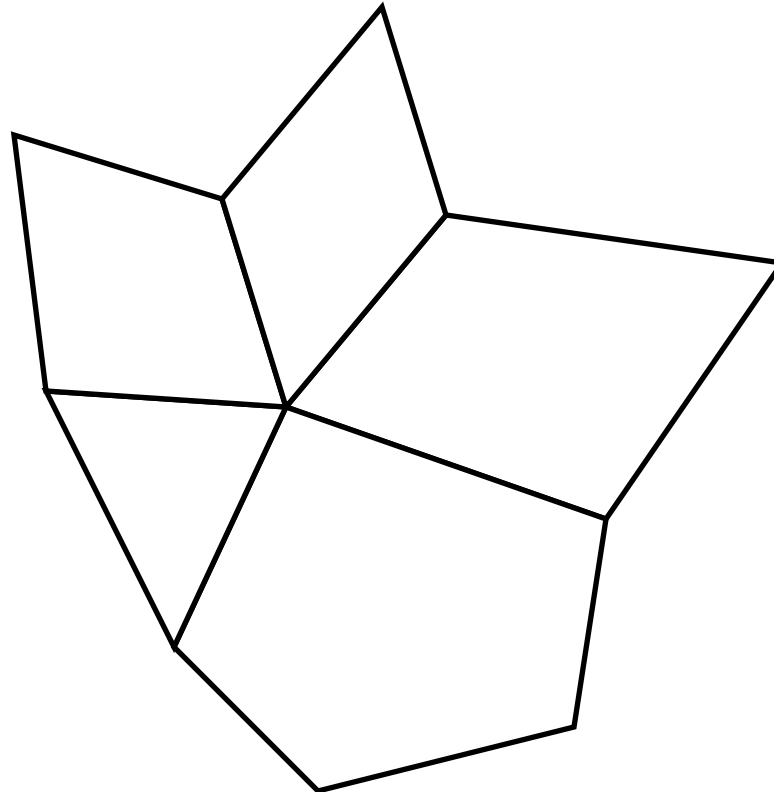
Scott Schaefer

Subdivision Schemes

- Common subdivision schemes
 - Catmull-Clark
 - Loop
 - Many others
 - Differ in ...
 - Input topology
 - How refine topology
 - How refine geometry
- ... which makes differences in ...
- Provable properties

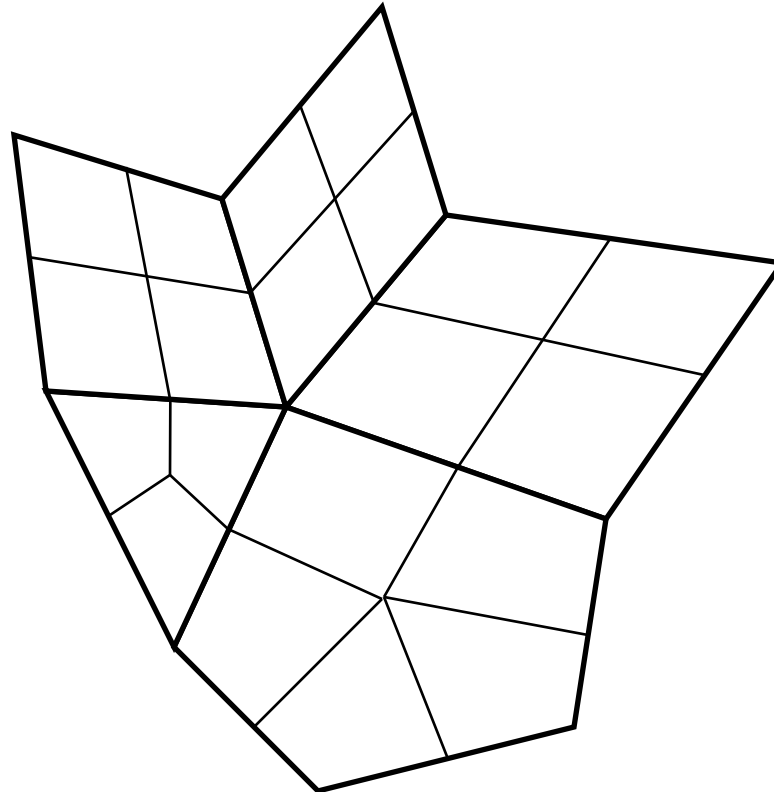


Catmull-Clark Subdivision



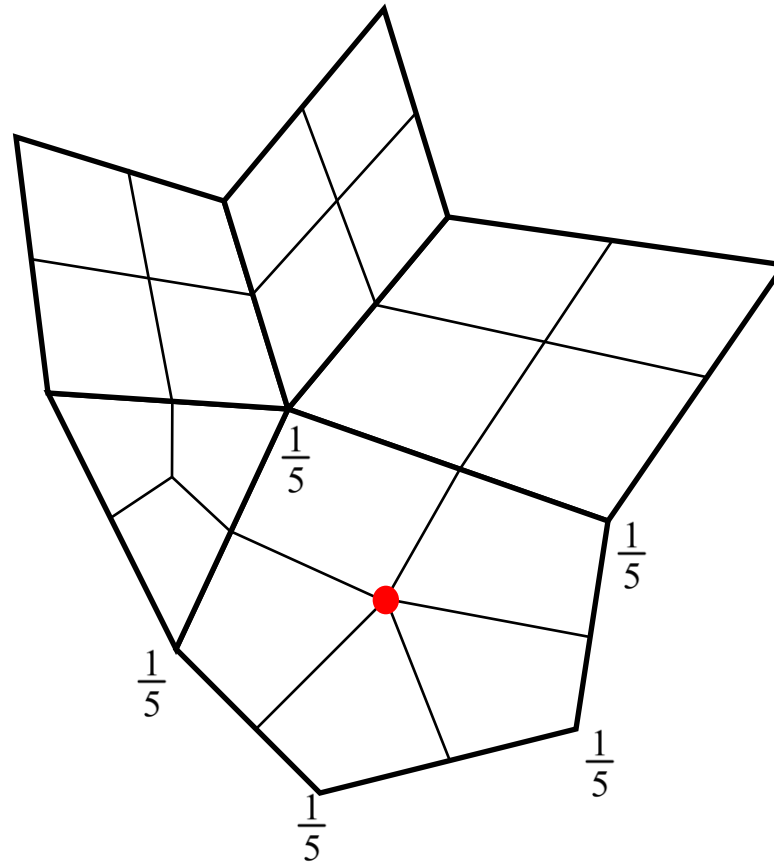
Scott Schaefer

Catmull-Clark Subdivision

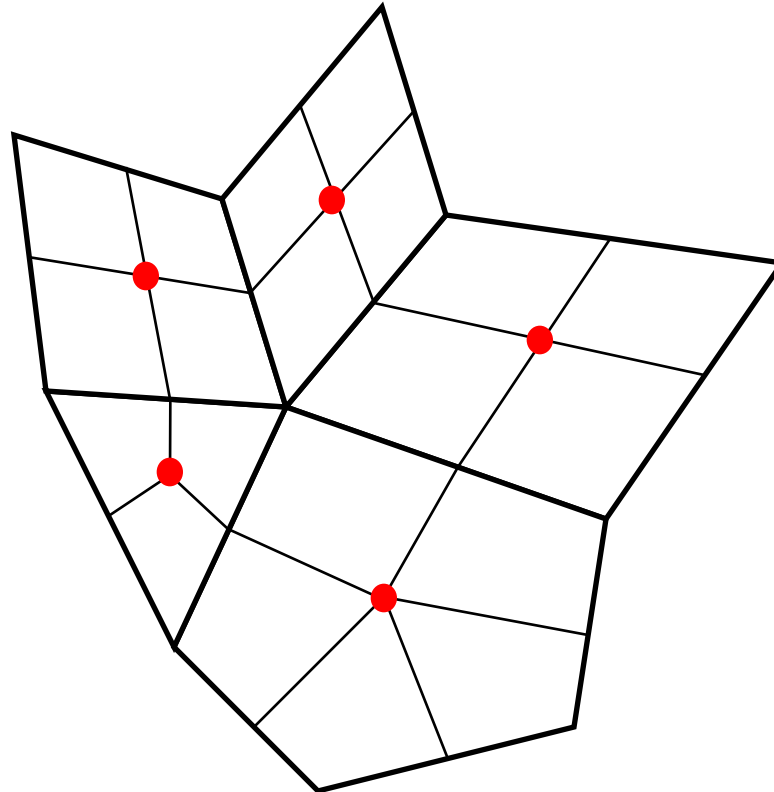


Scott Schaefer

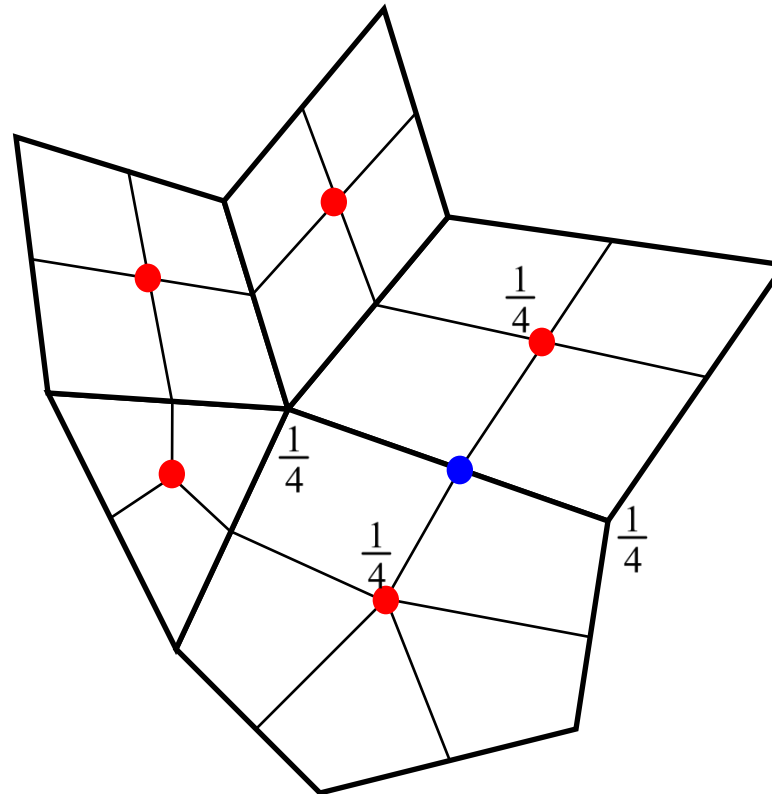
Catmull-Clark Subdivision



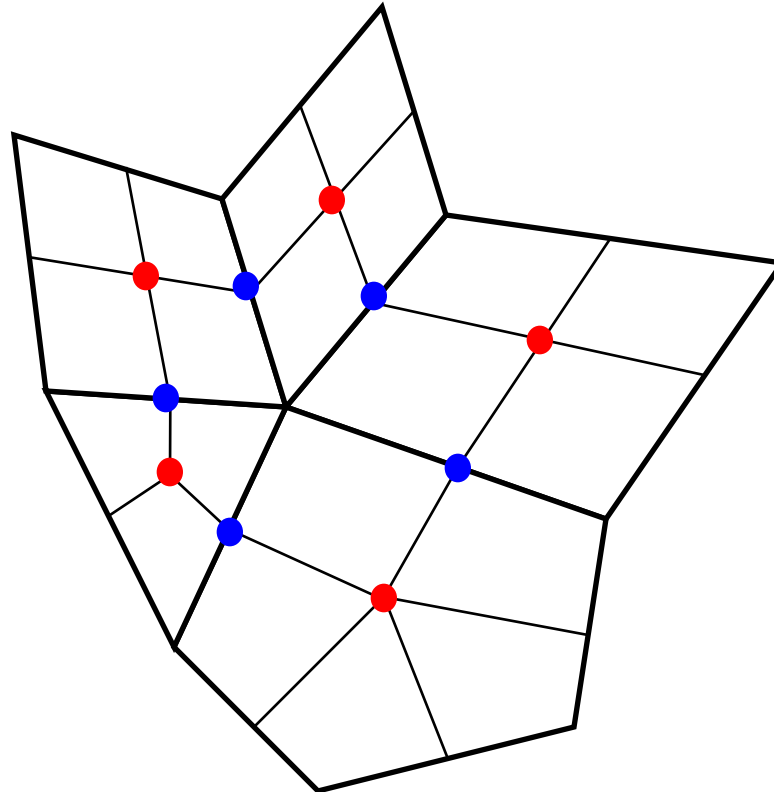
Catmull-Clark Subdivision



Catmull-Clark Subdivision



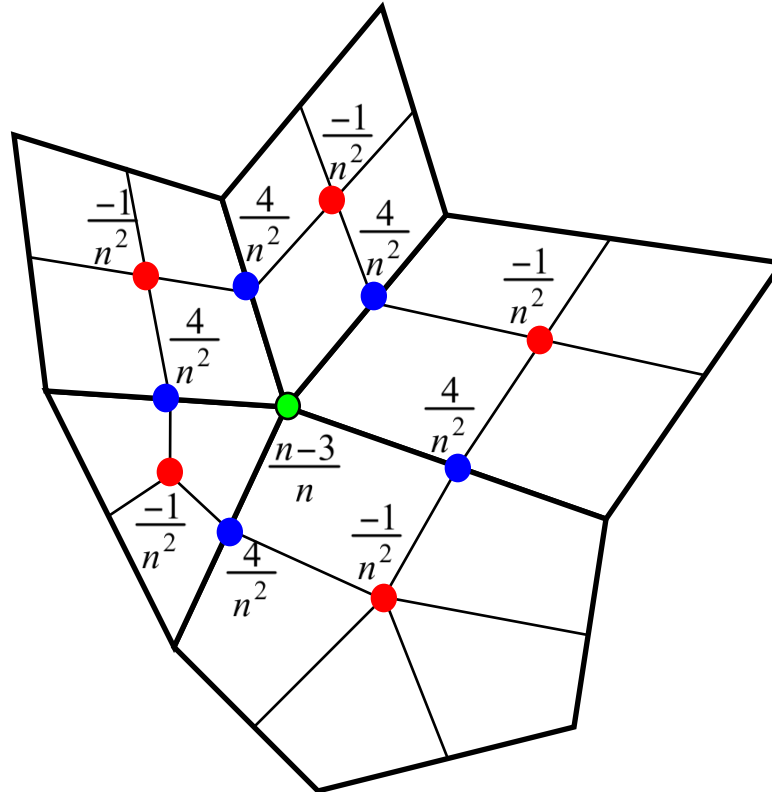
Catmull-Clark Subdivision



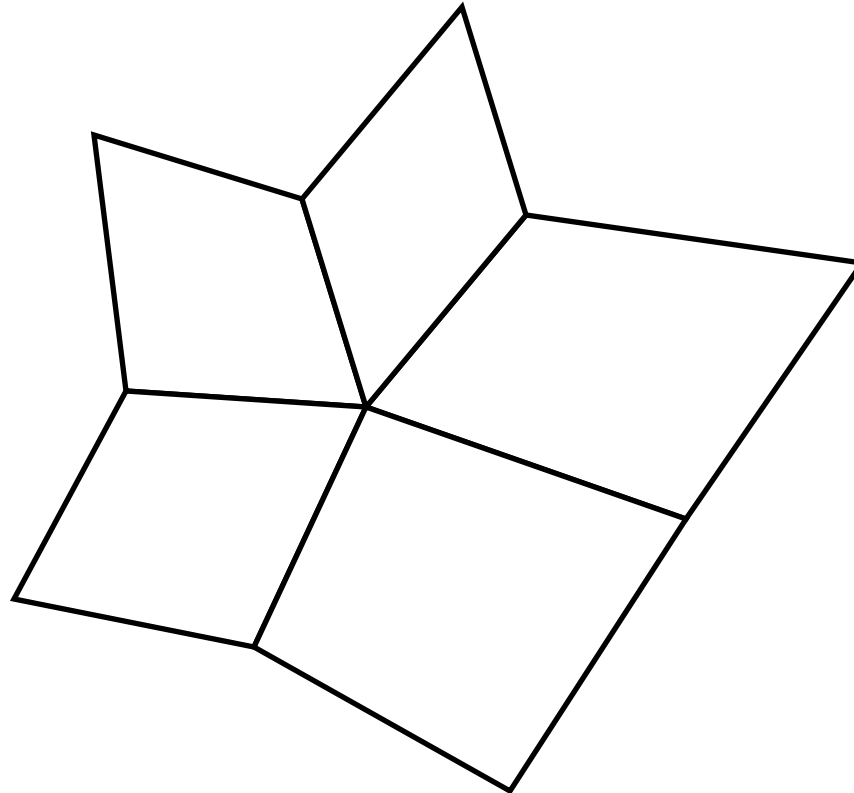


Catmull-Clark Subdivision

$$\text{New } \bullet = \left(4 * \text{avg of } \bullet - 1 * \text{avg of } \bullet + (n-3) * \bullet \right) / n$$

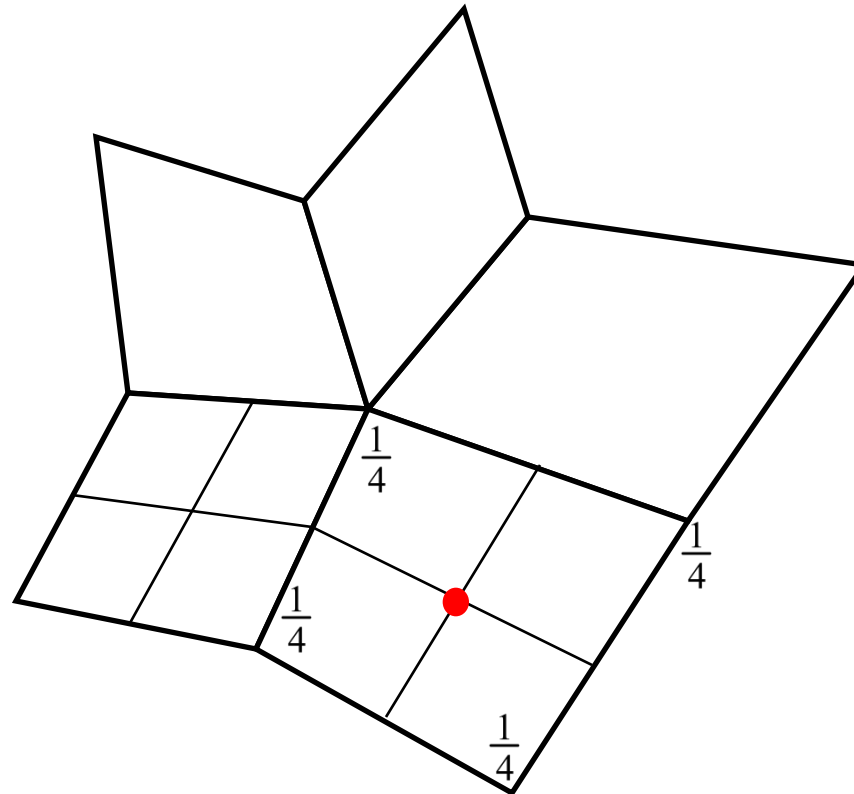


Catmull-Clark Subdivision

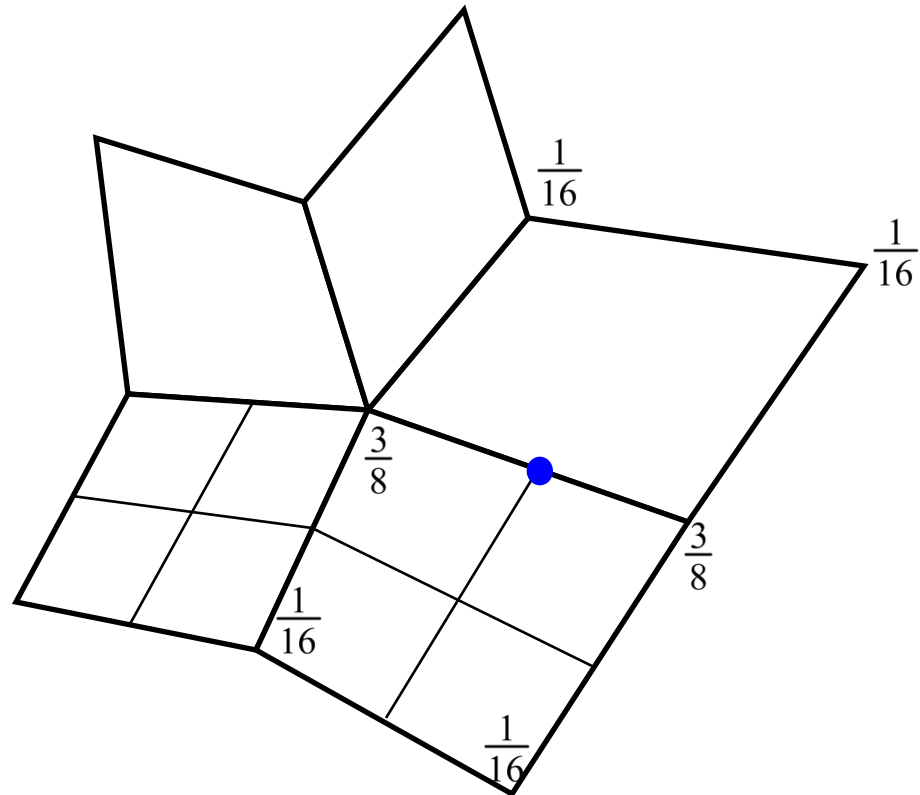


Scott Schaefer

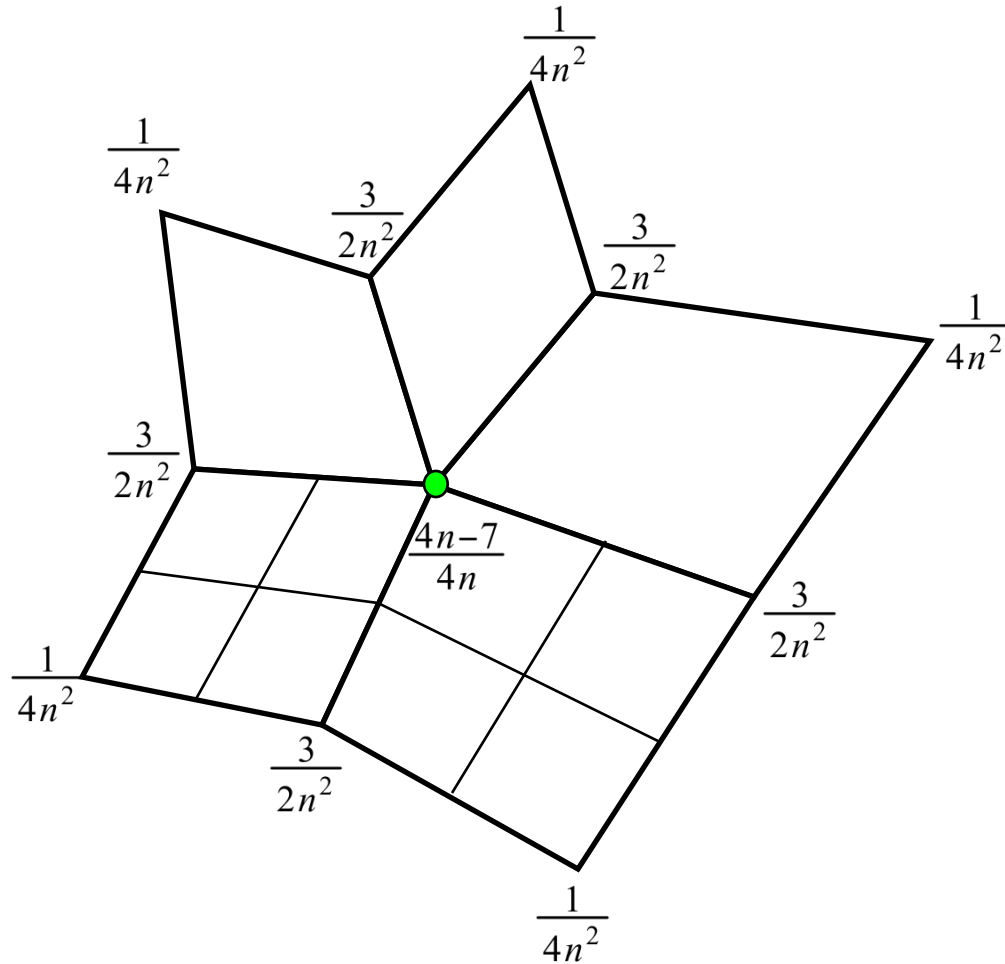
Catmull-Clark Subdivision



Catmull-Clark Subdivision



Catmull-Clark Subdivision



Catmull-Clark Subdivision

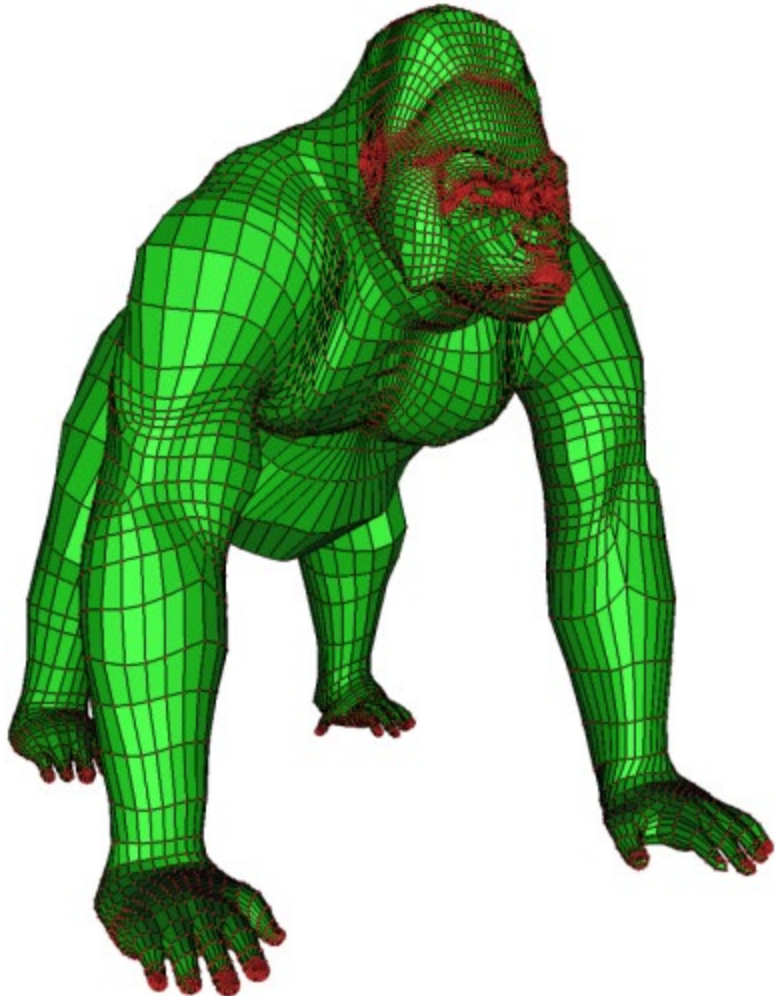


Repeated
Averaging



Catmull-Clark
Subdivision

Catmull-Clark Subdivision



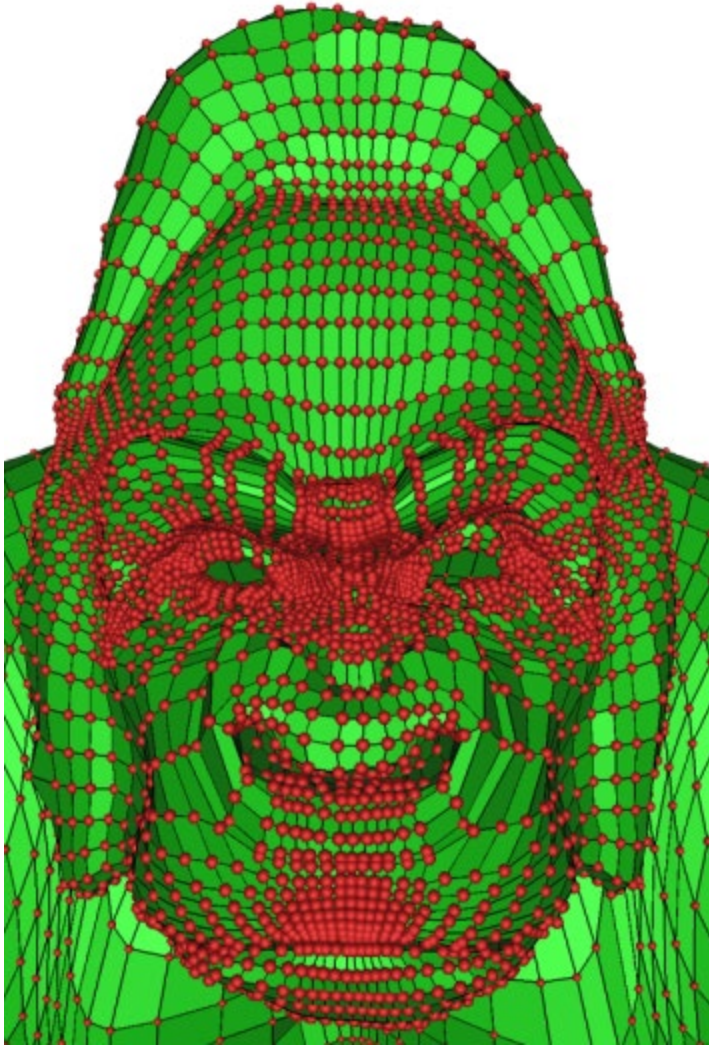
Scott Schaefer

Catmull-Clark Subdivision



Scott Schaefer

Catmull-Clark Subdivision



Scott Schaefer

Catmull-Clark Subdivision



Scott Schaefer

Catmull-Clark Subdivision

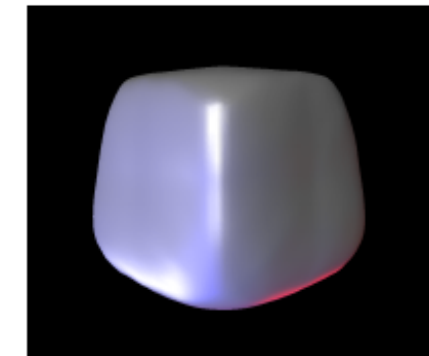
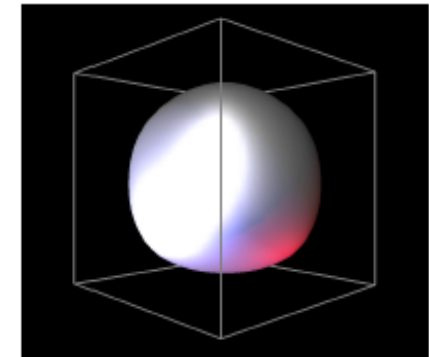
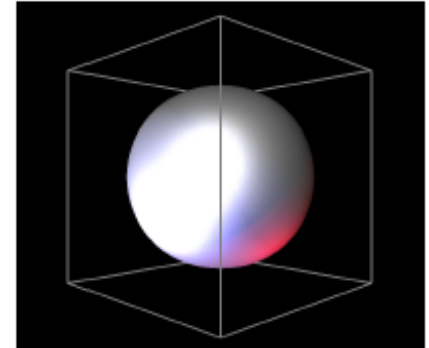


- One round of subdivision produces all quads
- Smoothness of limit surface
 - C^2 almost everywhere
 - C^1 at vertices with valence $\neq 4$
- Relationship to control mesh
 - Does not interpolate input vertices
 - Within convex hull
- Most commonly used subdivision scheme in the movies...



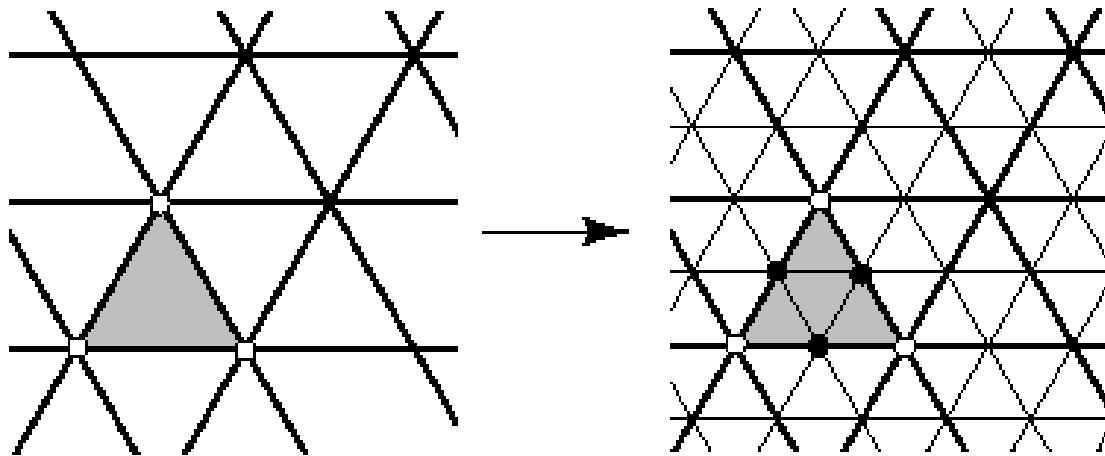
Subdivision Schemes

- Common subdivision schemes
 - Catmull-Clark
 - Loop
 - Many others
 - Differ in ...
 - Input topology
 - How refine topology
 - How refine geometry
- ... which makes differences in ...
- Provable properties



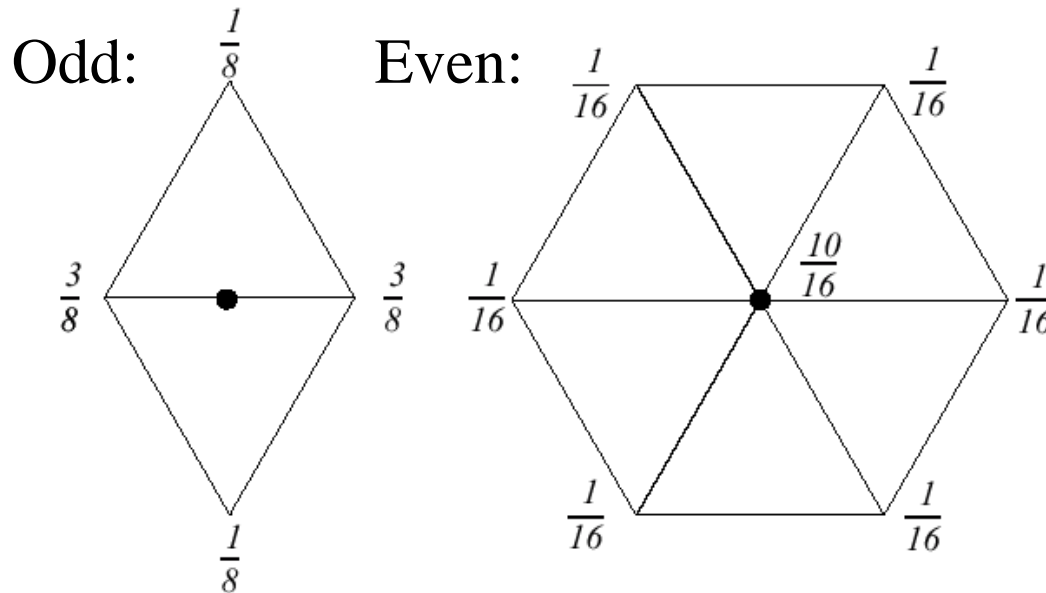
Loop Subdivision

- Operates on pure triangle meshes
- Subdivision rules
 - Linear subdivision
 - Averaging rules for “even / odd” (white / black) vertices



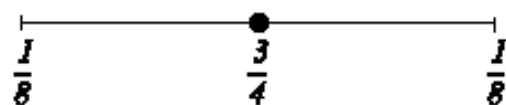
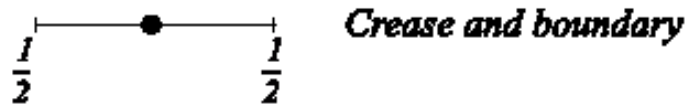
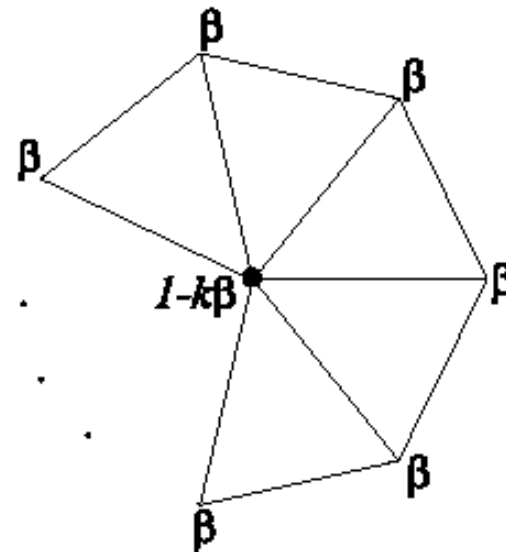
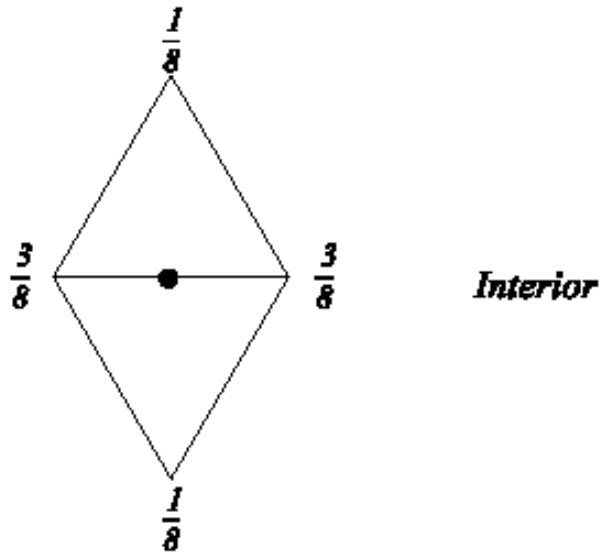
Loop Subdivision

- Averaging rules
 - Weights for “odd” and “even” vertices



Loop Subdivision

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



a. Masks for odd vertices

b. Masks for even vertices



Loop Subdivision

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

» Original Loop

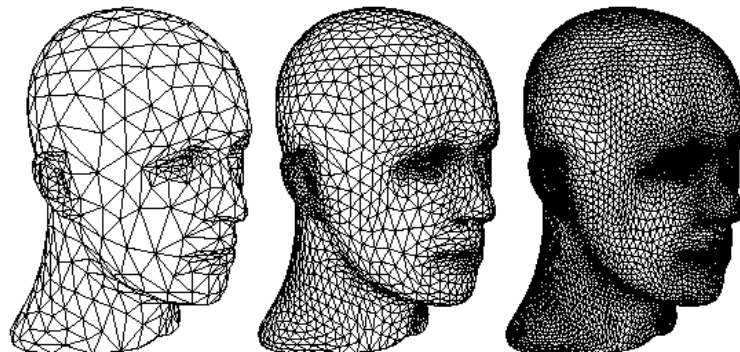
$$\beta = \frac{1}{n} \left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{n} \right)^2 \right)$$

» Warren

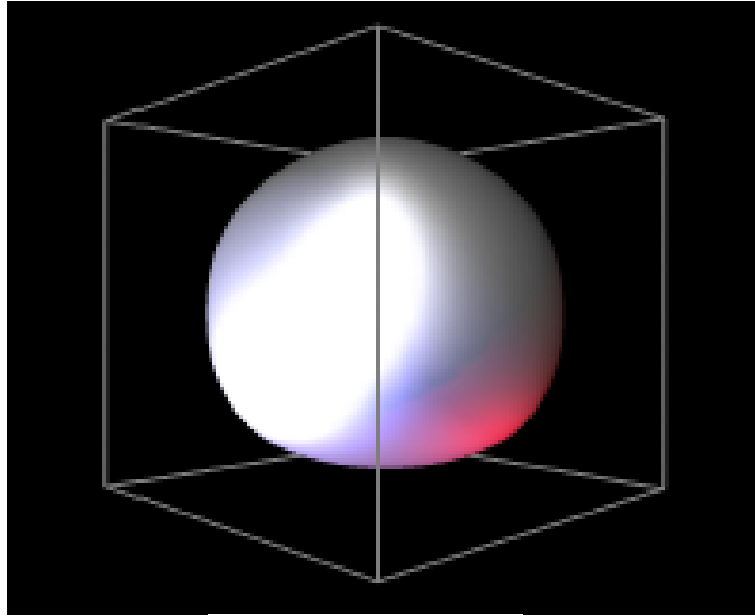
$$\beta = \begin{cases} \frac{3}{8n} & n > 3 \\ \frac{3}{16} & n = 3 \end{cases}$$

Loop Subdivision

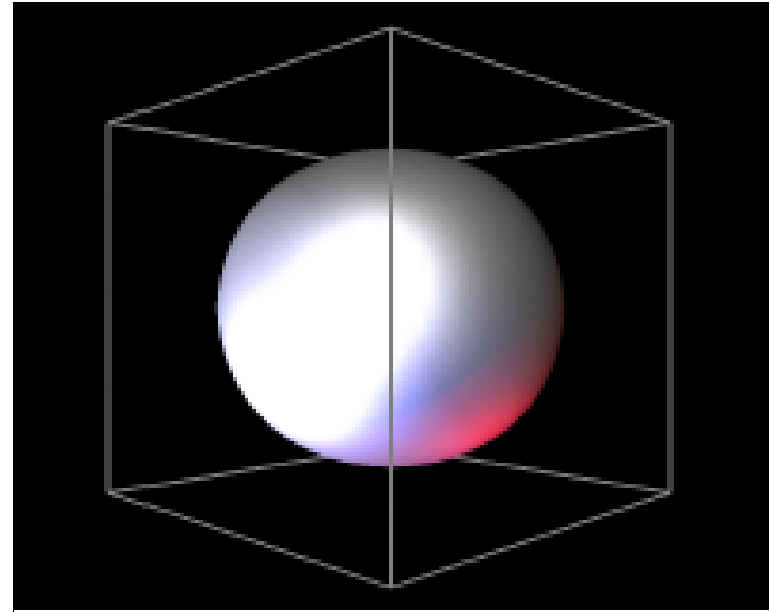
- Operates only on triangle meshes
- Smoothness of limit surface
 - C^2 almost everywhere
 - C^1 at vertices with valence $\neq 6$
- Relationship to control mesh
 - Does not interpolate input vertices
 - Within convex hull



Subdivision Schemes

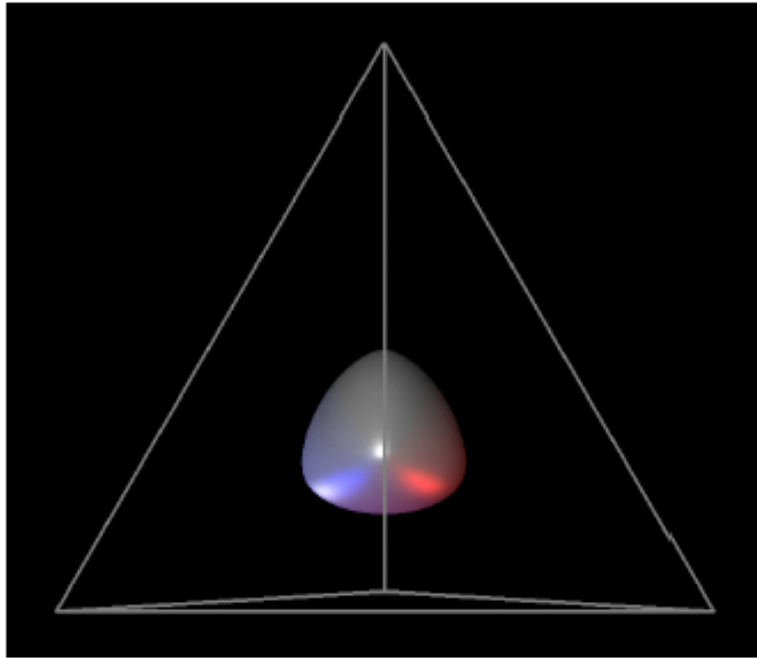


Loop

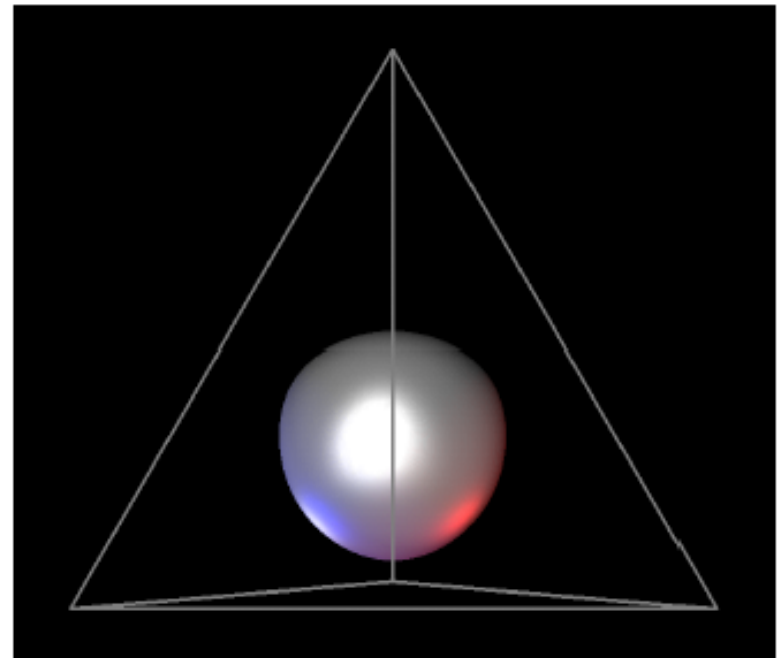


Catmull-Clark

Subdivision Schemes



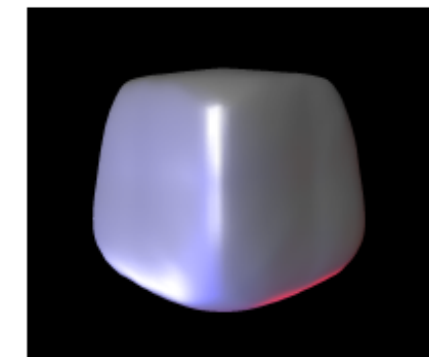
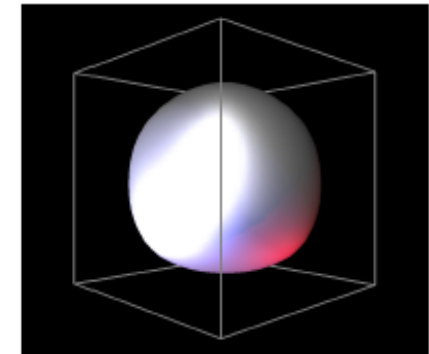
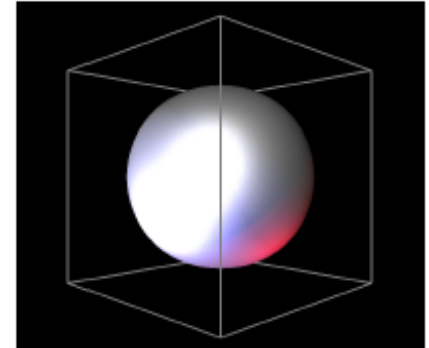
Loop



Catmull-Clark

Subdivision Schemes

- Common subdivision schemes
 - Catmull-Clark
 - Loop
 - Many others
 - Differ in ...
 - Input topology
 - How refine topology
 - How refine geometry
- ... which makes differences in ...
- Provable properties





Subdivision Schemes

- Other subdivision schemes

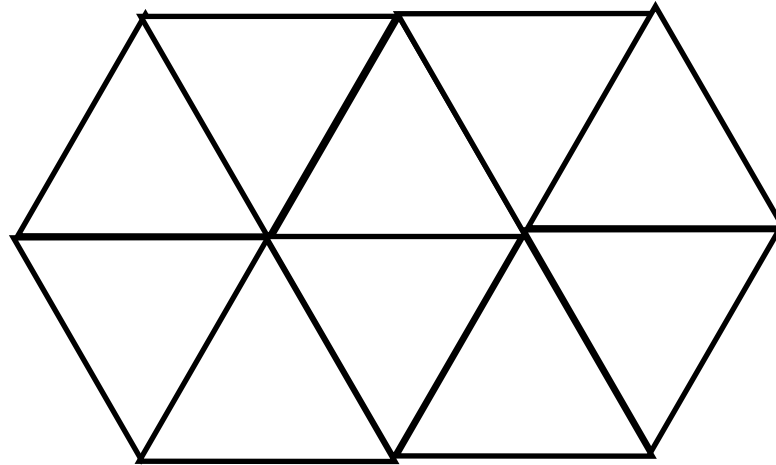
Face split		
	<i>Triangular meshes</i>	<i>Quad. meshes</i>
<i>Approximating</i>	Loop (C^2)	Catmull-Clark (C^2)
<i>Interpolating</i>	Mod. Butterfly (C^1)	Kobbelt (C^1)

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

Other Subdivision Schemes



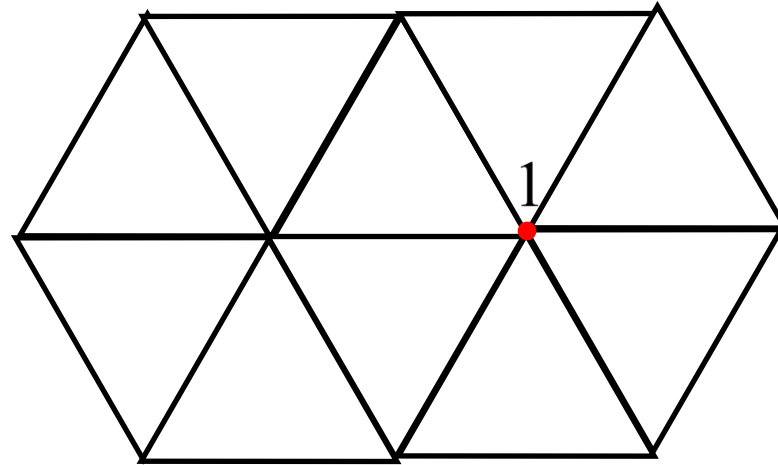
- Butterfly subdivision



Other Subdivision Schemes



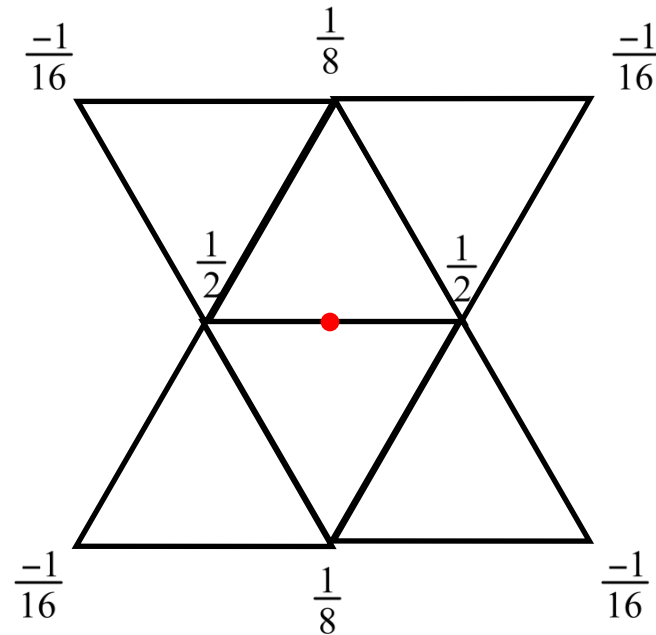
- Butterfly subdivision



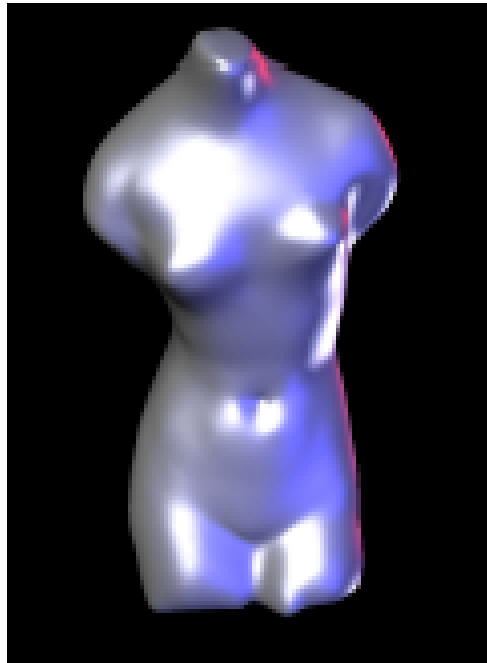
Other Subdivision Schemes



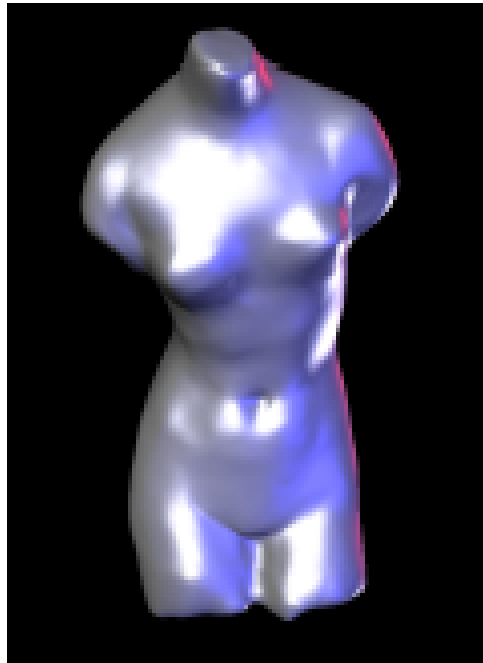
- Butterfly subdivision



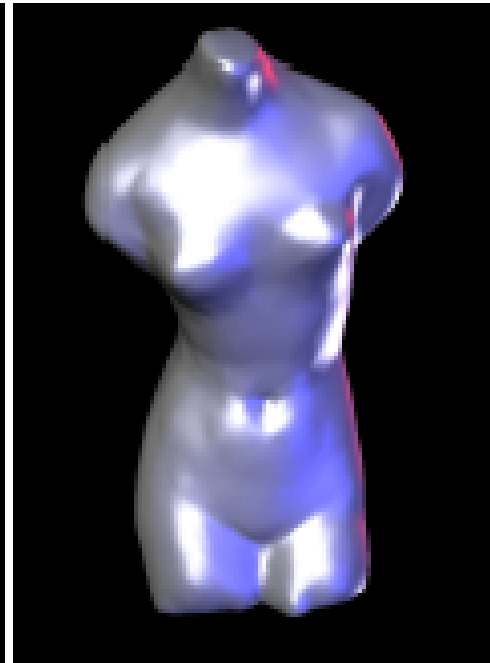
Other Subdivision Schemes



Loop



Butterfly

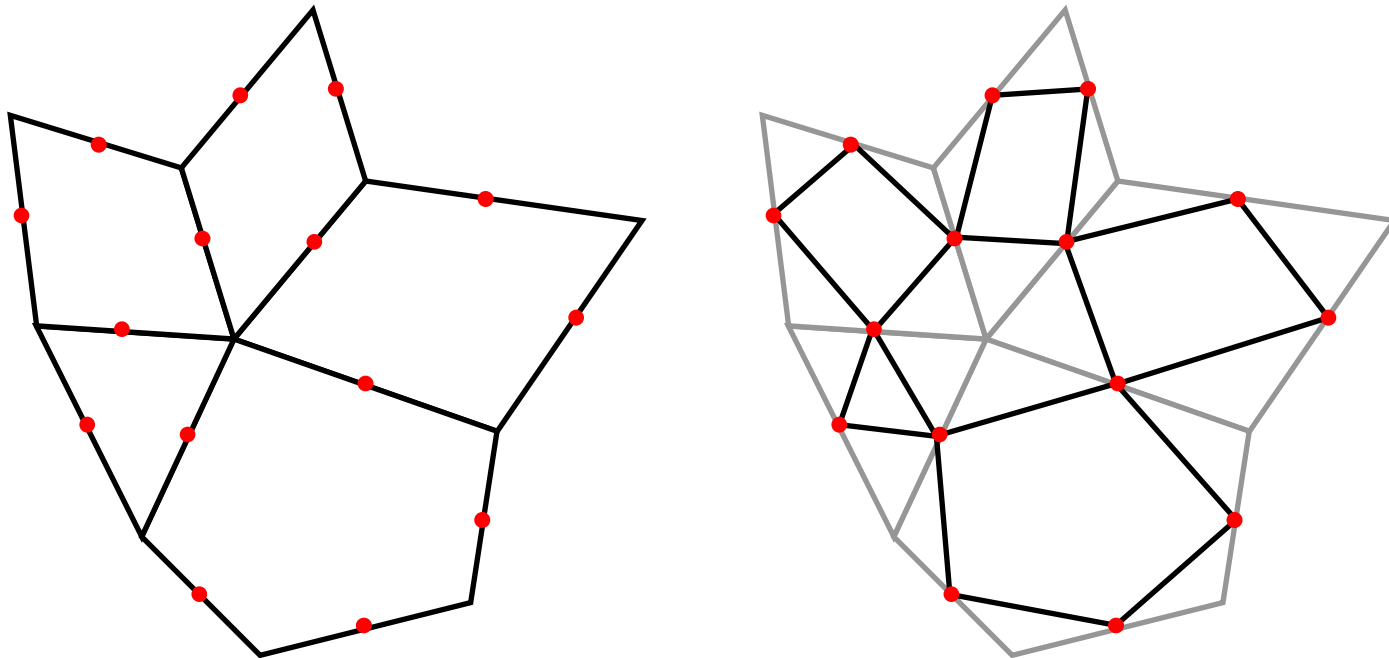


Catmull-Clark

Other Subdivision Schemes



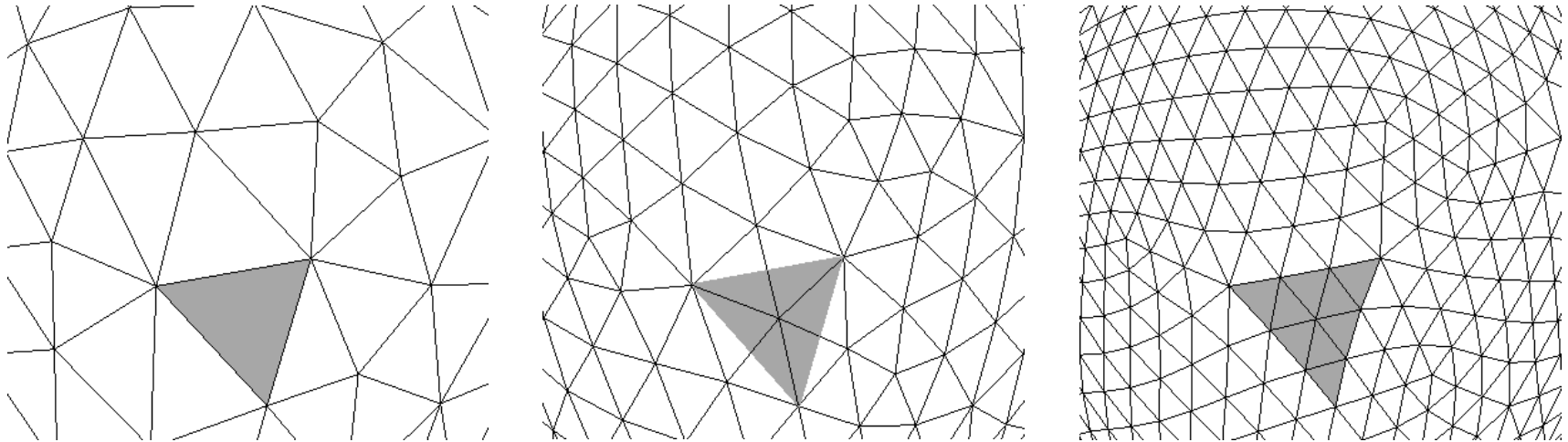
- Vertex-split subdivision
(Doo-Sabin, Midedge, Biquartic)



One step of Midedge subdivision

Other Subdivision Schemes

- Sqrt(3) subdivision



Rotating grid of sqrt(3) subdivision



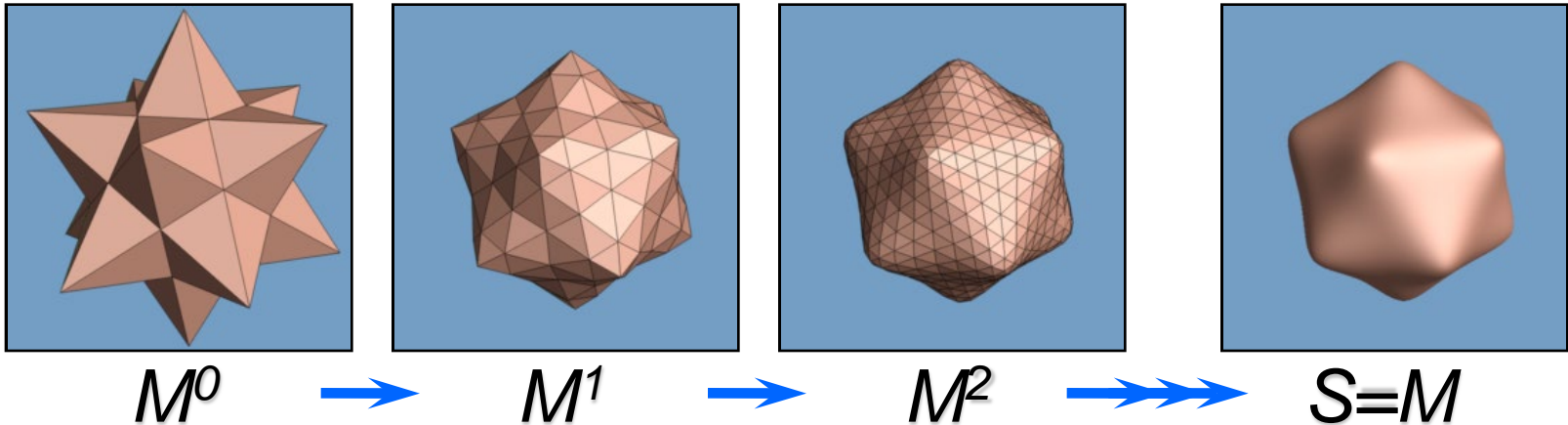
Extensions

- Common schemes assume *stationary* rules
- Locally altering subdivision rules allows for additional effects
- Examples for non-stationary rules:
 - Edge preservation
 - Adaptive subdivision

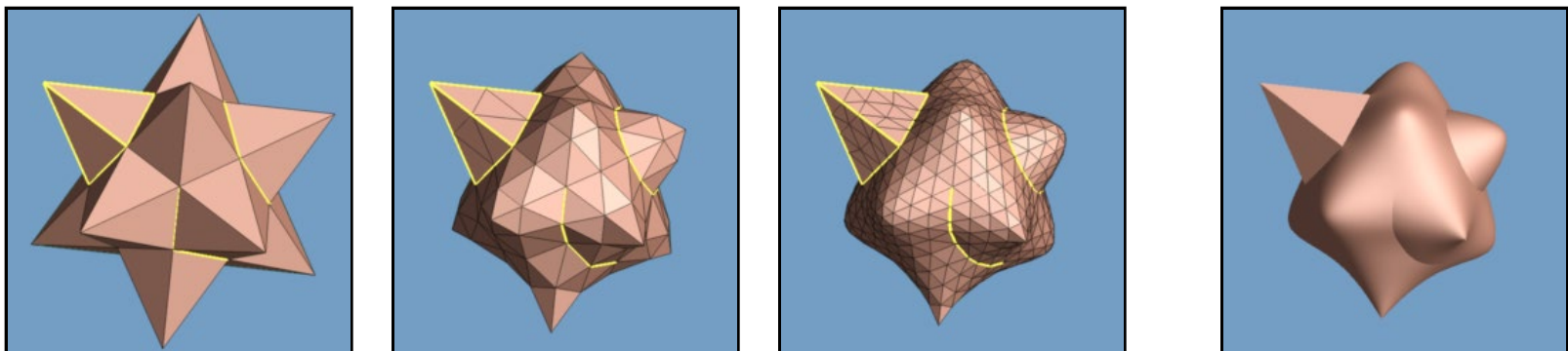
Edge Preservation

- Treat selected control mesh edges as boundary:

[Loop '87]



[Hoppe et al. '94]



tagged mesh

Adaptive Subdivision



- Goal:
 - Best possible approximation of smooth limit surface
 - With limited triangle budget
- Quality of approximation can be defined by
 - Projected (screen) area of final triangles
 - Local surface curvature

Adaptive Subdivision

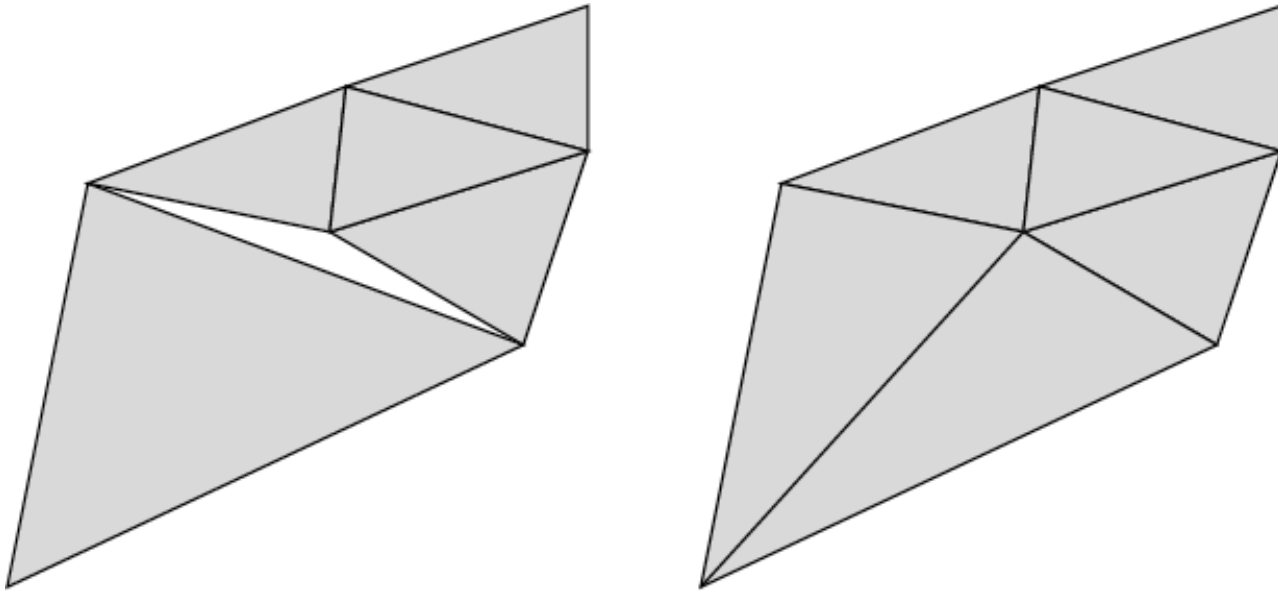


- Goal:
 - Best possible approximation of smooth limit surface
 - With limited triangle budget
- Quality of approximation can be defined by
 - Projected (screen) area of final triangles
 - Local surface curvature
- Solution:
 - Stop subdivision at different levels across the surface
 - Stop-criterion depending on quality measure
 - Project each vertex onto limit surface

Adaptive Subdivision



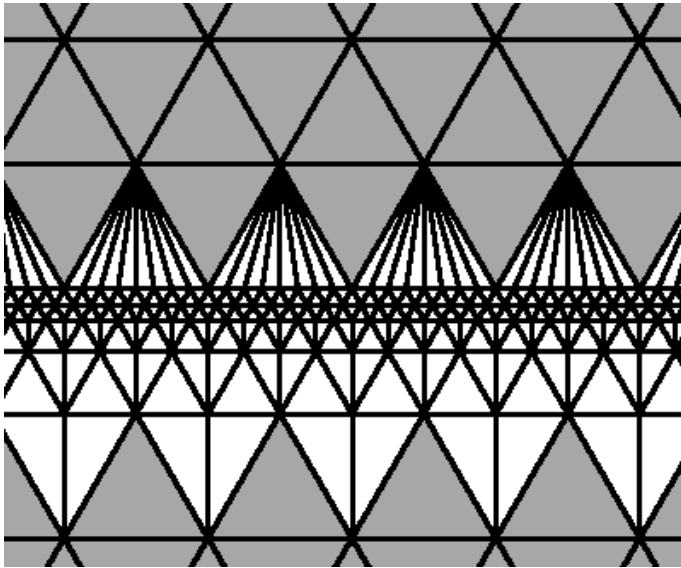
- Problem:
 - Different levels of subdivision may lead to gaps in the surface



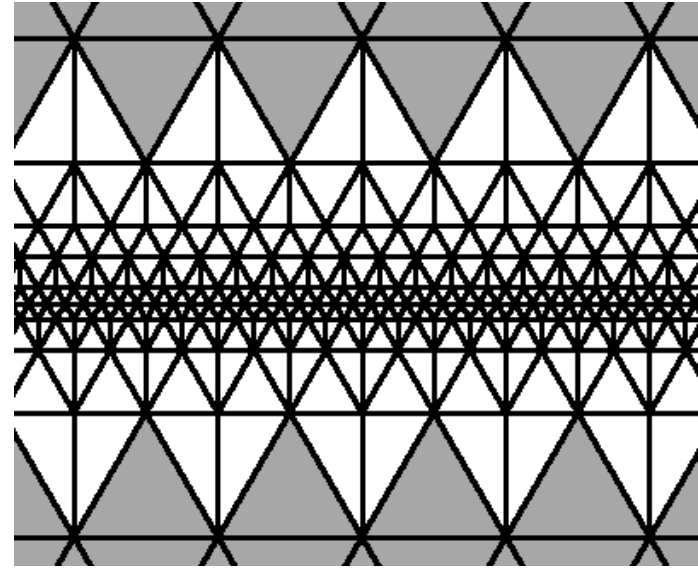
Adaptive Subdivision



- Solution:
 - Replacing incompatible coarse triangles by *triangle fan*
 - Balanced subdivision: neighboring subdivision levels must not differ by more than one



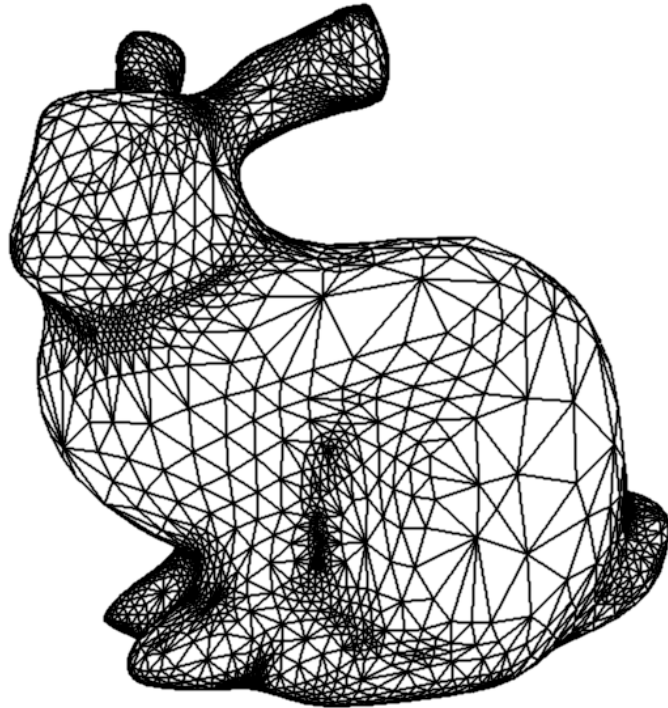
Unbalanced



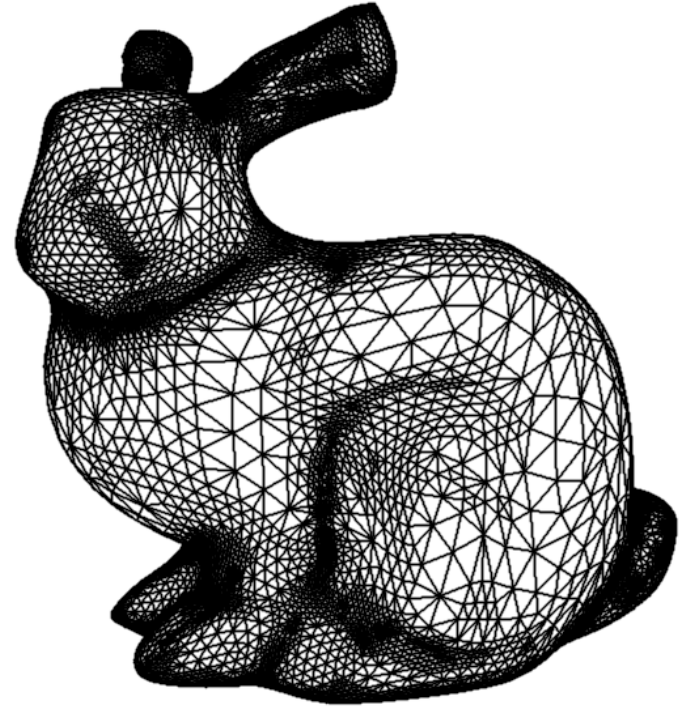
Balanced

[Kobbelt 2000]

Adaptive Subdivision



10072 Triangles



228654 Triangles

[Kobbelt 2000]

Subdivision Surfaces

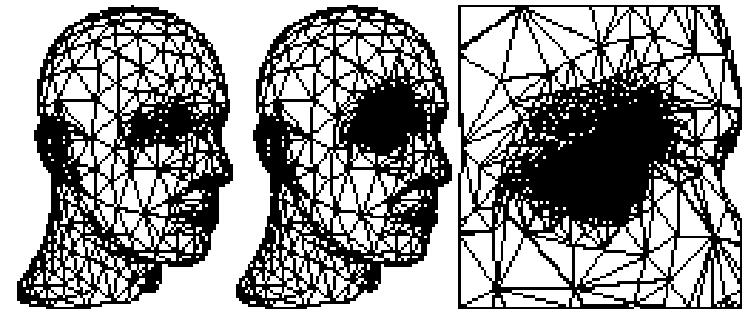


- Properties:
 - o Accurate
 - o Concise
 - o Intuitive specification
 - o Local support
 - o Affine invariant
 - o Arbitrary topology
 - o Guaranteed continuity
 - o Natural parameterization
 - o Efficient display
 - o Efficient intersections



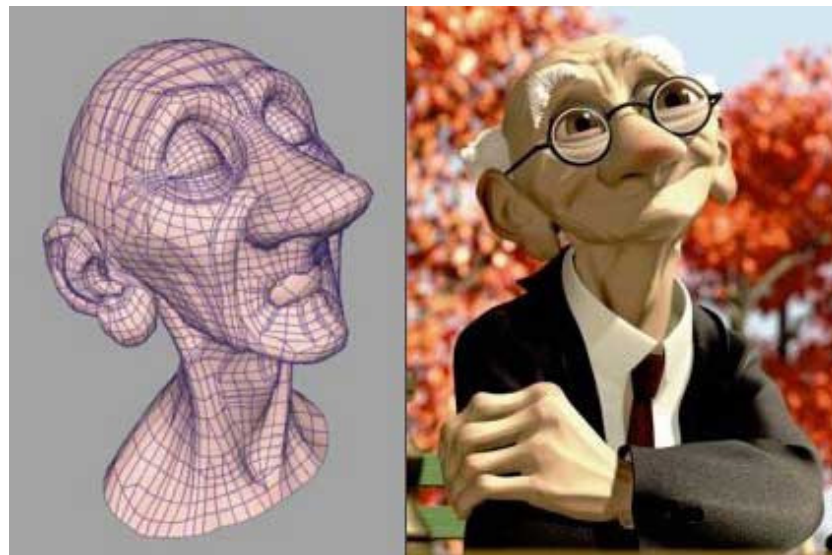
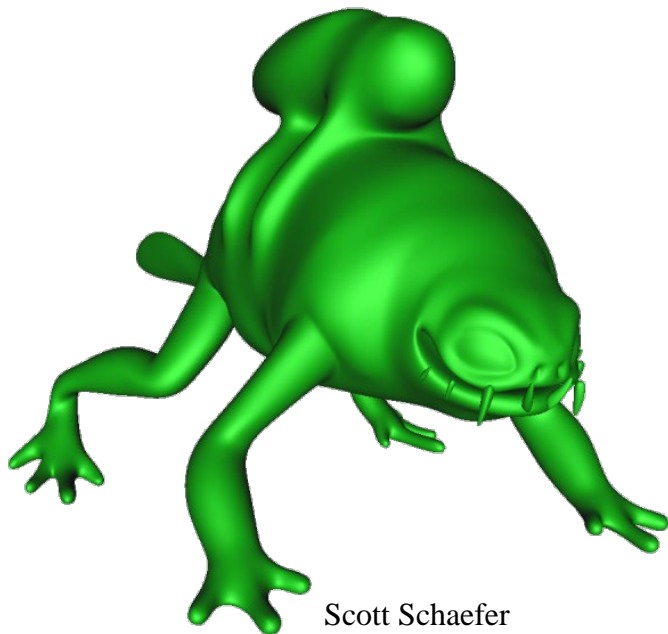
Subdivision Surfaces

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



Subdivision Surfaces

- Used in movie and game industries
- Supported by most 3D modeling software



Geri's Game © Pixar Animation Studios

Summary



Feature	Polygonal Mesh	Subdivision Surface
Accurate	No	Yes
Concise	No	Yes
Intuitive specification	No	No
Local support	Yes	Yes
Affine invariant	Yes	Yes
Arbitrary topology	Yes	Yes
Guaranteed continuity	No	Yes
Natural parameterization	No	No
Efficient display	Yes	Yes
Efficient intersections	No	No