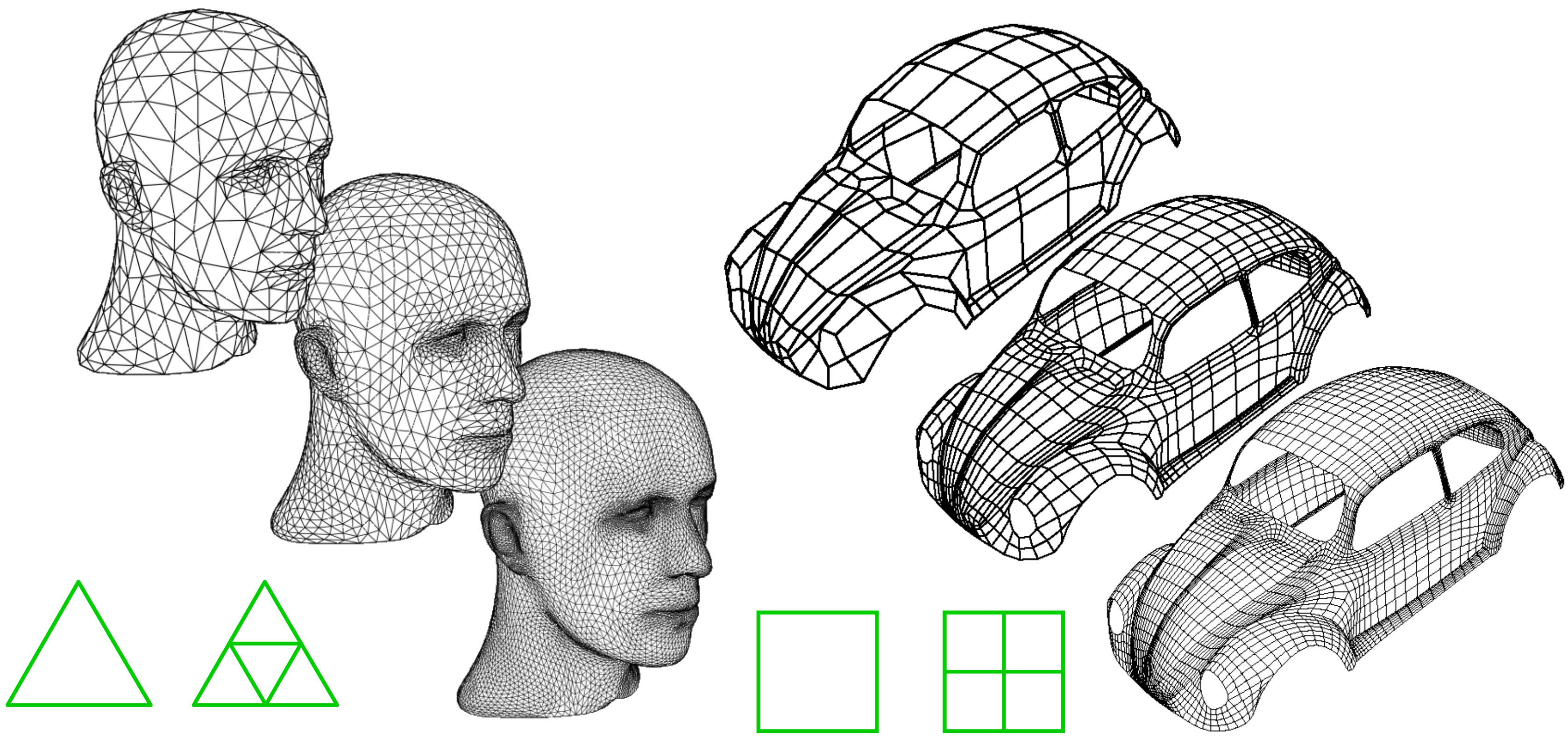
Subdivision Surfaces
(from discrete to continuous)

Pierre Alliez
Inria

Subdivision Surfaces?

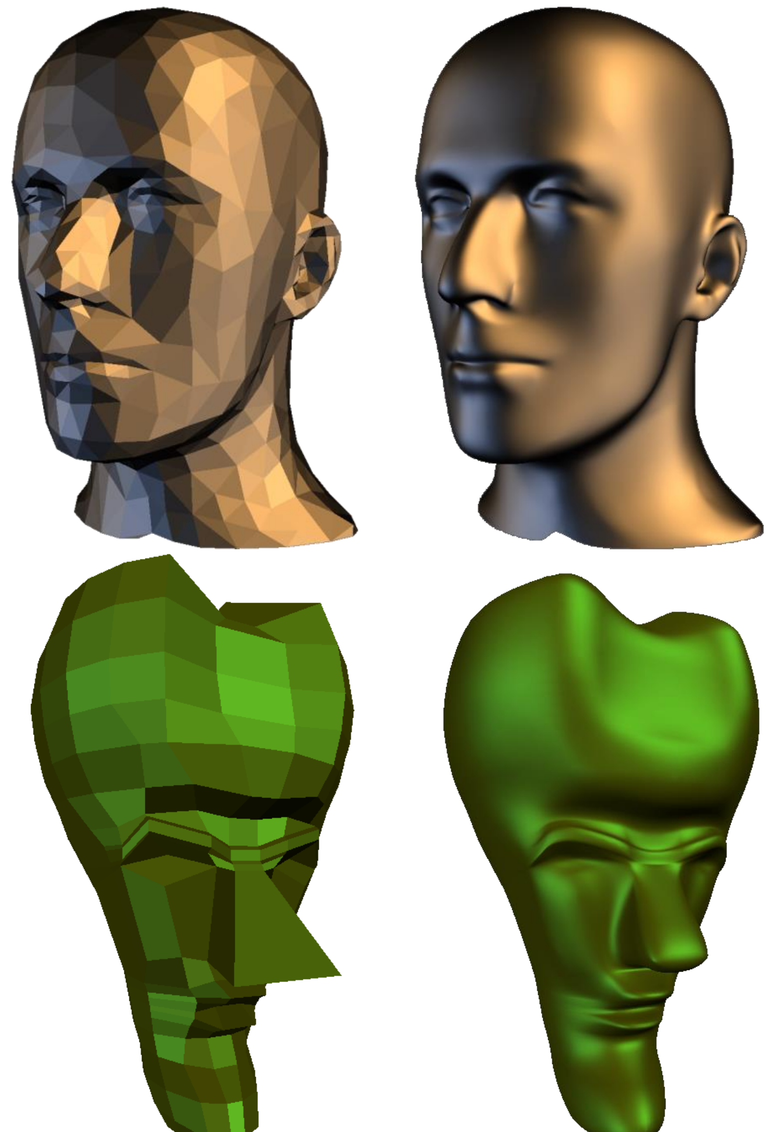
Smooth surfaces as the limit of a sequence of refinements



Why Subdivision

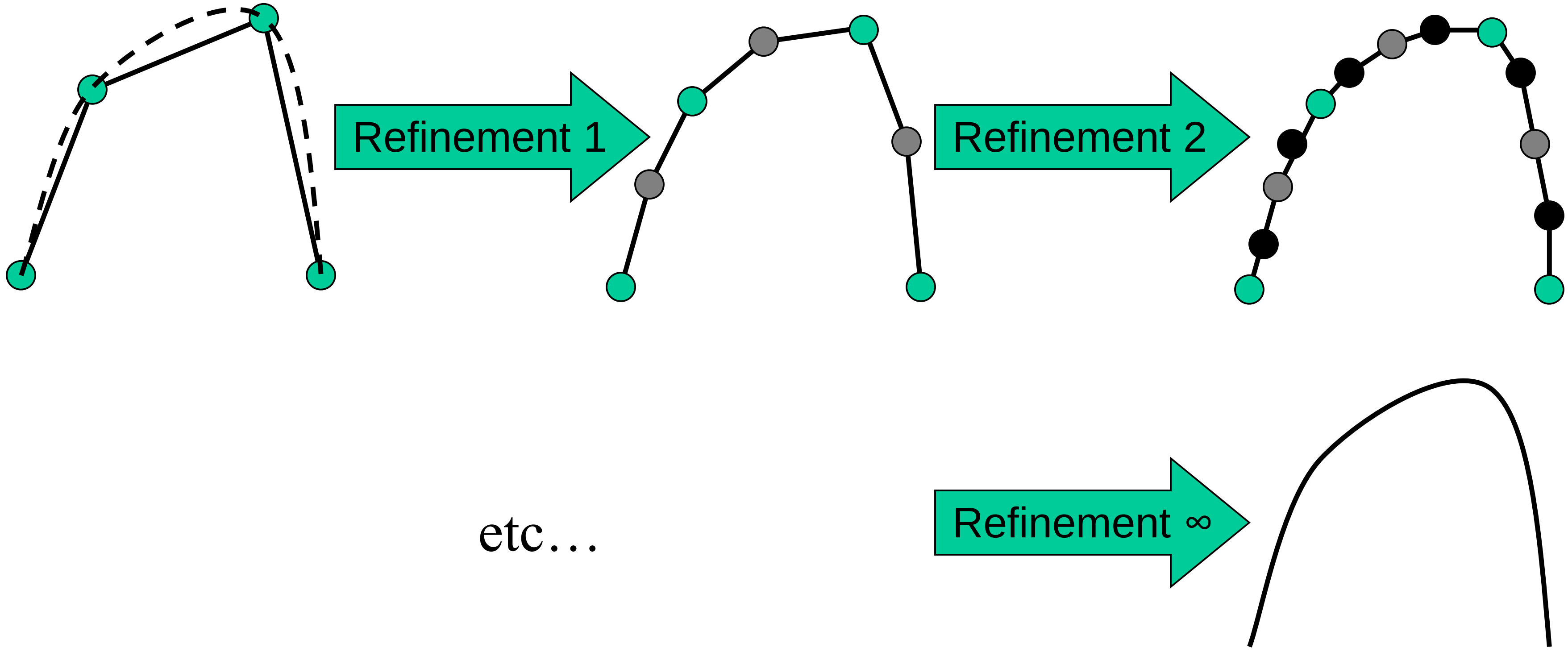
Many advantages

- arbitrary topology
- scalable
- wavelet connection
- easy to implement
- efficient



Let’s Start with Subdivision Curves

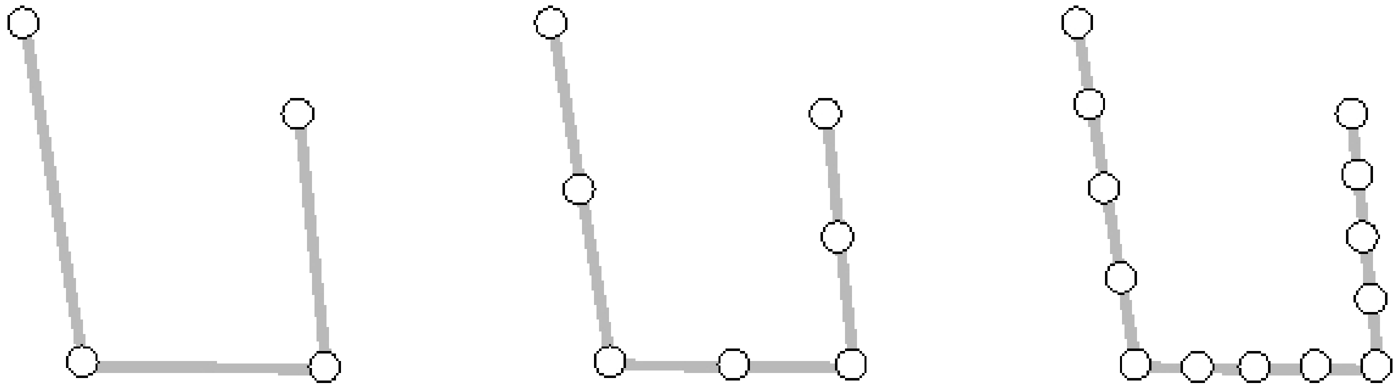
Approach a limit curve through an iterative refinement process.



Initial Idea

- linear

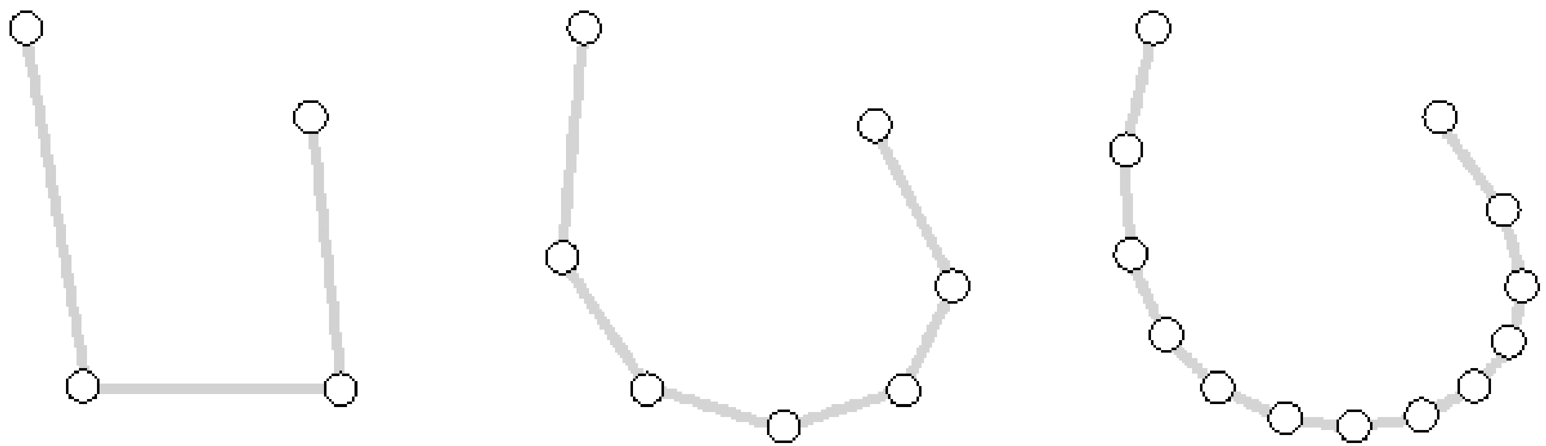
$$x_n = 1/2(x_l + x_r) \qquad y_n = 1/2(y_l + y_r)$$



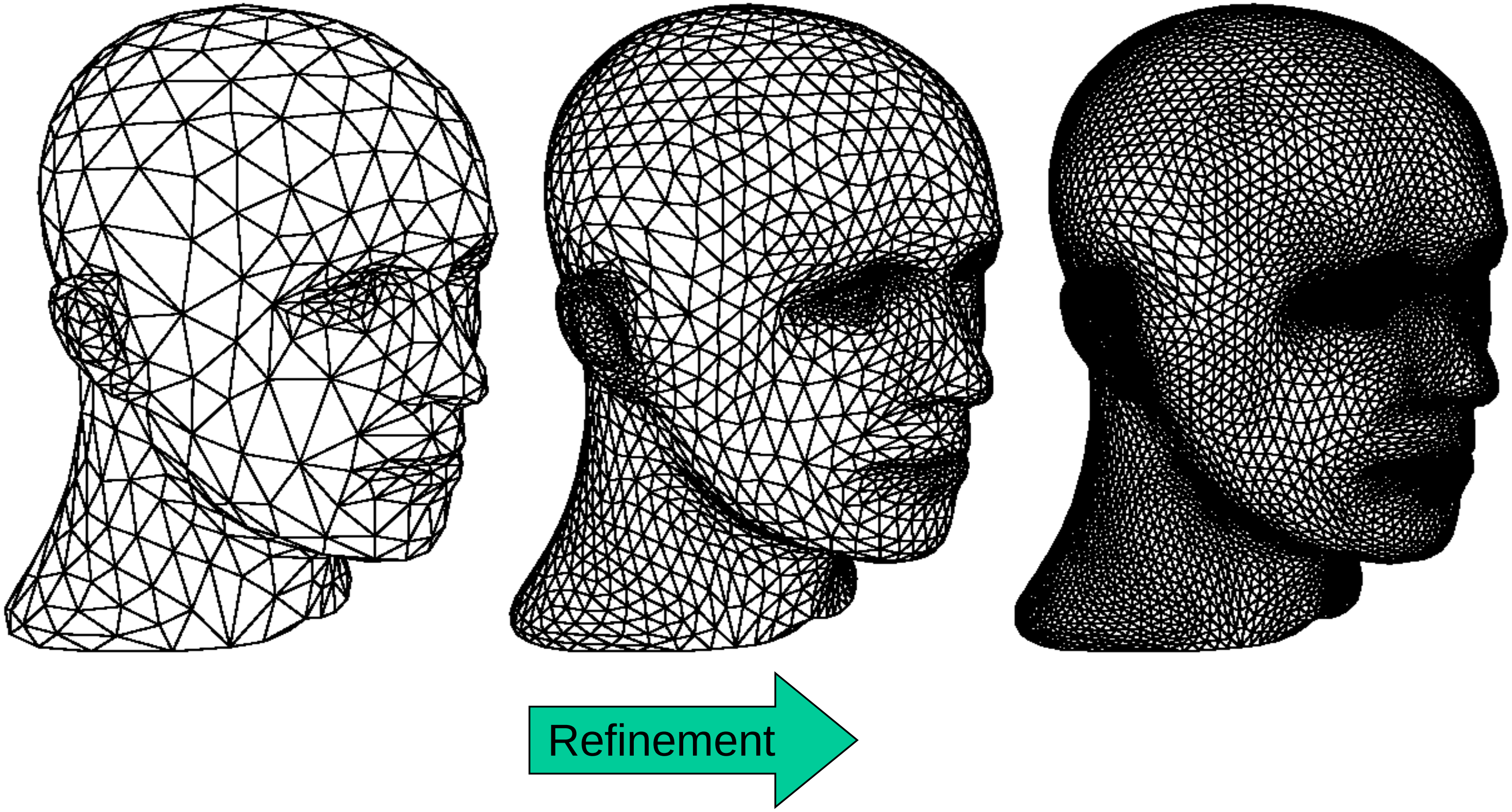
Subdivision

4 points scheme (interpolant):

$$p_{2i+i}^{j+1} = 1/16\left(-p_{i-1}^{j} + 9p_{i}^{j} + 9p_{i+1}^{j} - p_{i+2}^{j}\right)$$



Subdivision in 3D



Goals of Subdivision Surfaces

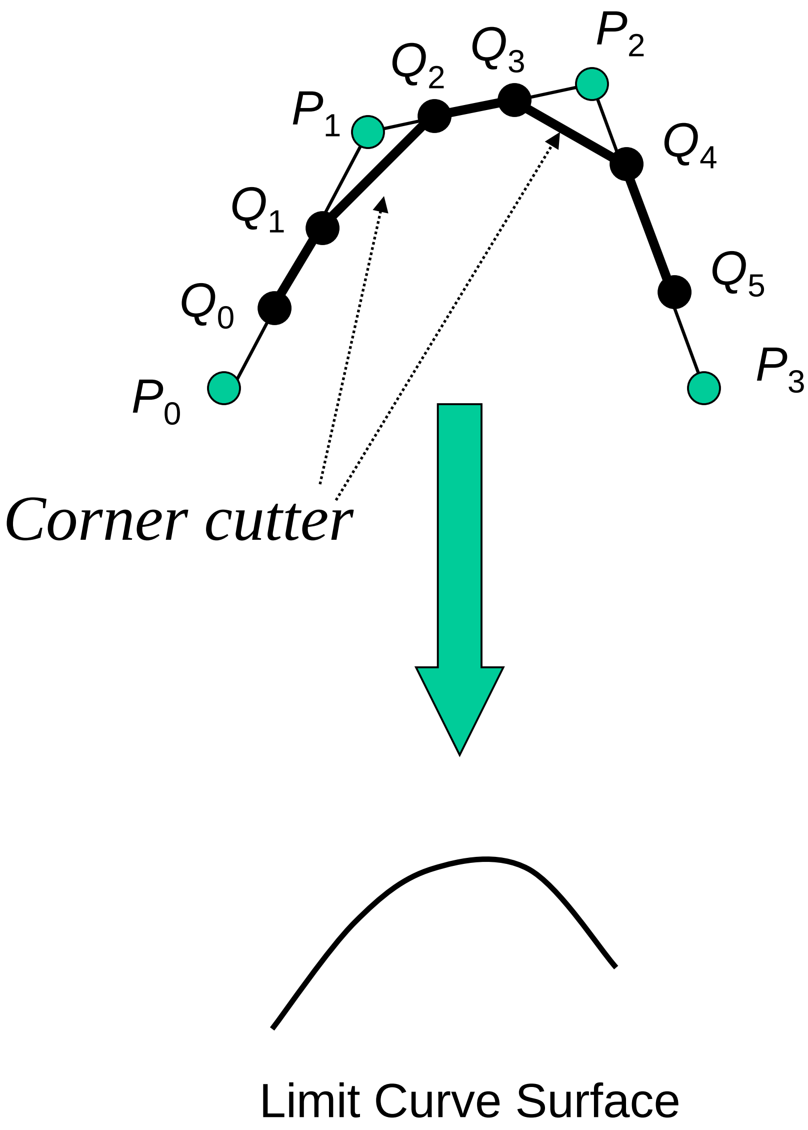
- How do we represent curved surfaces in the computer?
  - Efficiency of representation
  - Continuity
  - Affine Invariance
  - Efficiency of rendering
- How do they relate to spline patches?
- Why use subdivision rather than patches?

Types of Subdivision

- Interpolating Schemes
  - Limit Surfaces/Curve will pass through original set of data points.

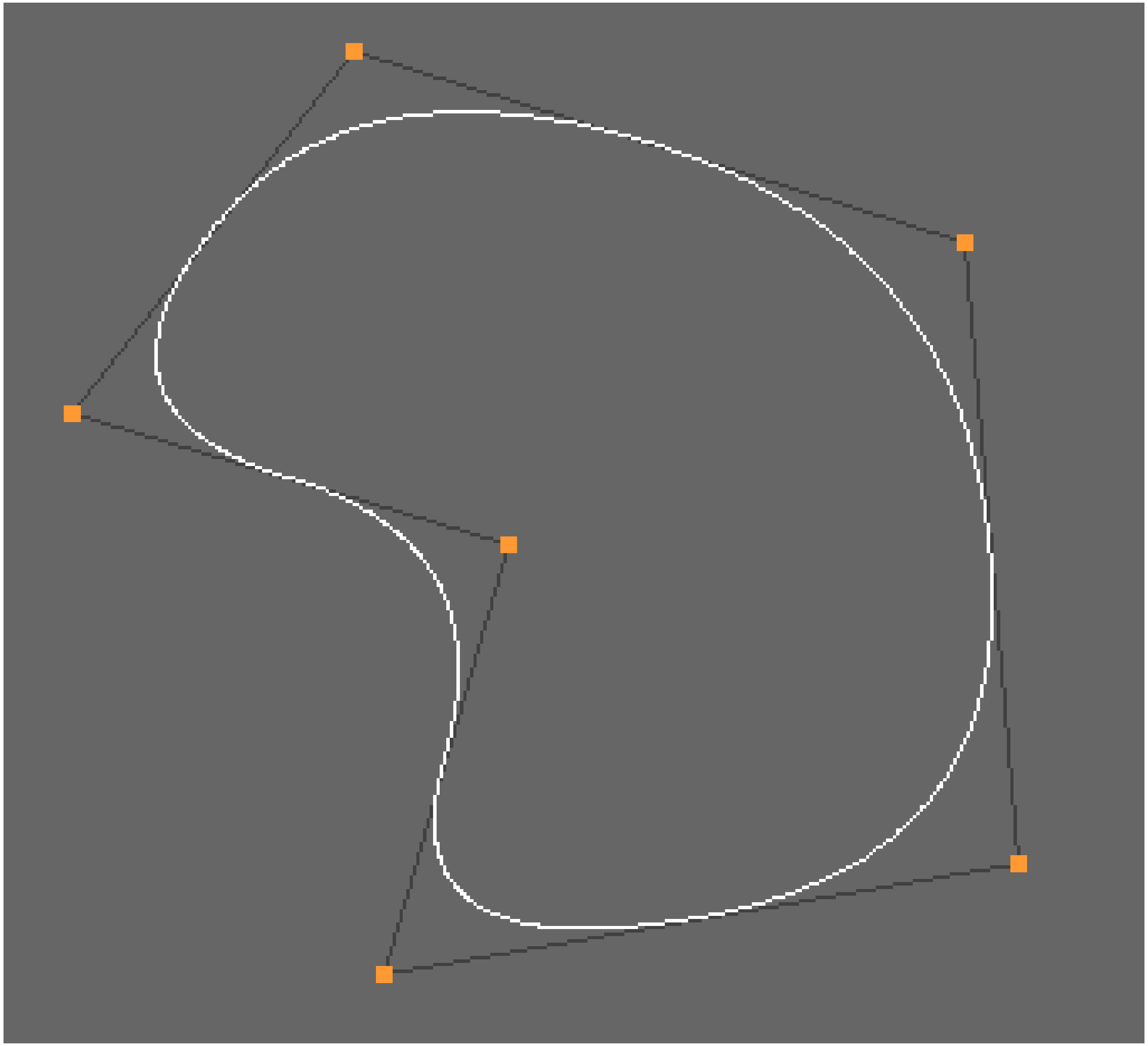
- Approximating Schemes
  - Limit Surface will not necessarily pass through the original set of data points.

A Primer: Chaiken's Algorithm

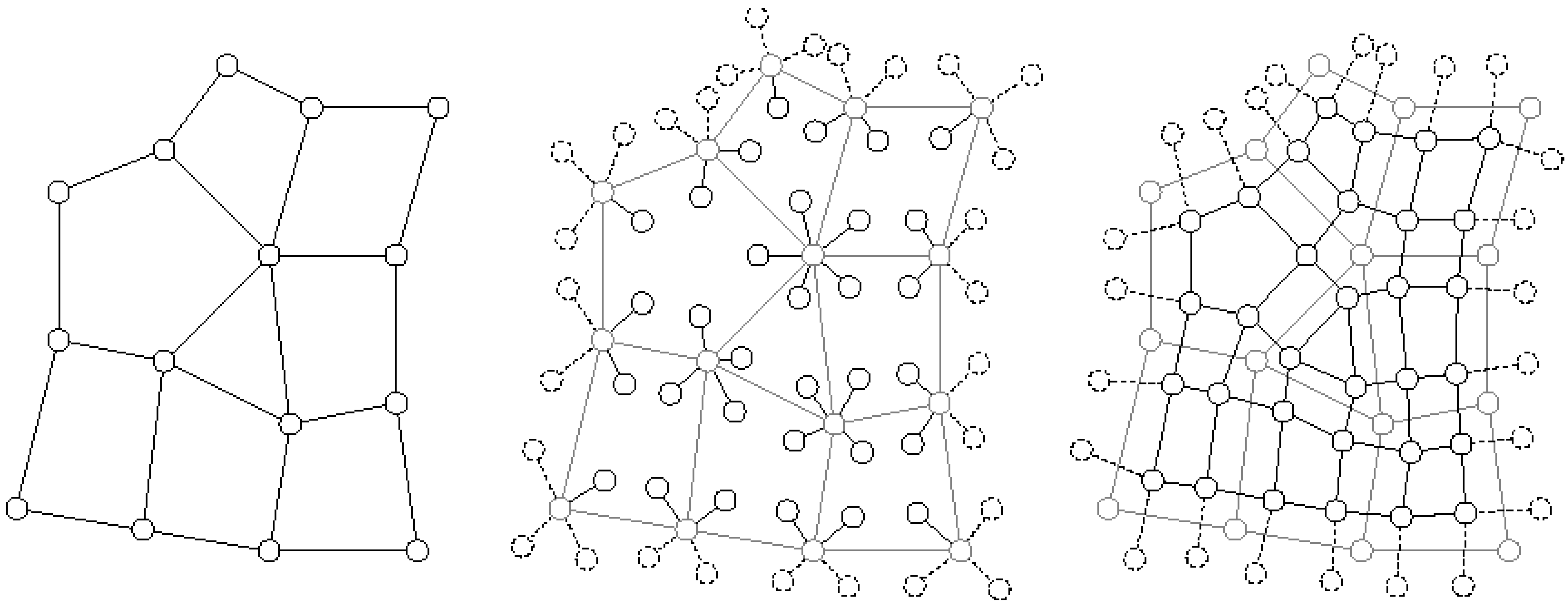


$$Q_{2i} = \frac{3}{4} P_i + \frac{1}{4} P_{i+1}$$

$$Q_{2i+1} = \frac{1}{4} P_i + \frac{3}{4} P_{i+1}$$

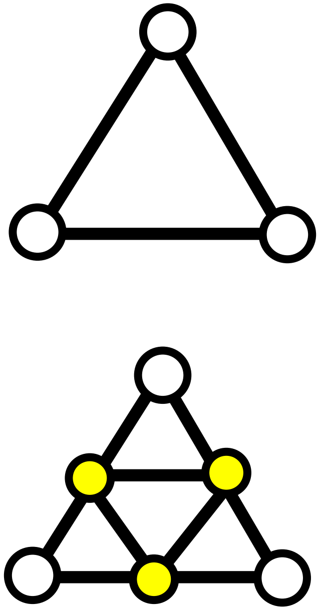


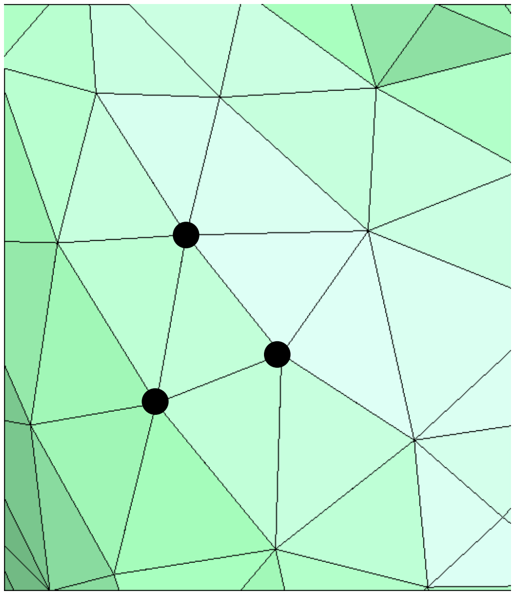
"Cut-corner"

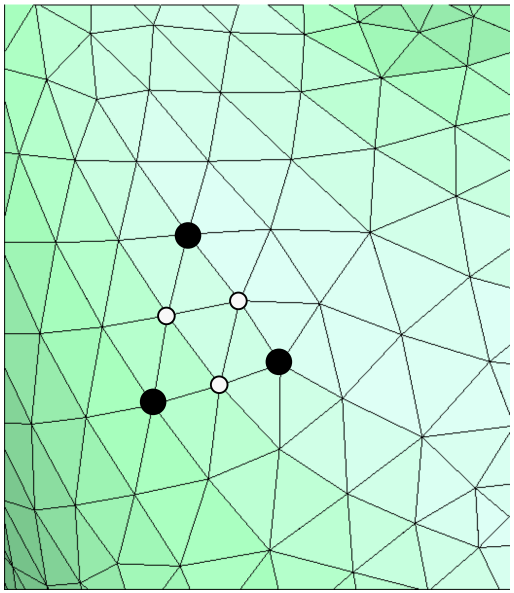


3D Surfaces: Loop Scheme

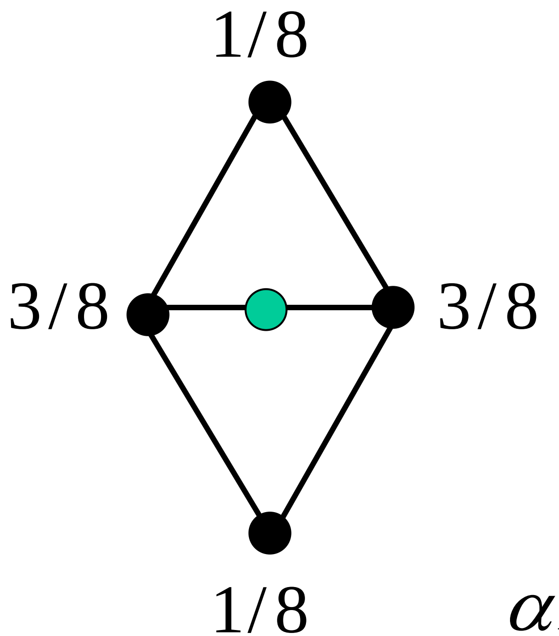
- Works on triangular meshes
- Is an approximating scheme
- Guaranteed to be smooth (C^2) everywhere except at extraordinary vertices.

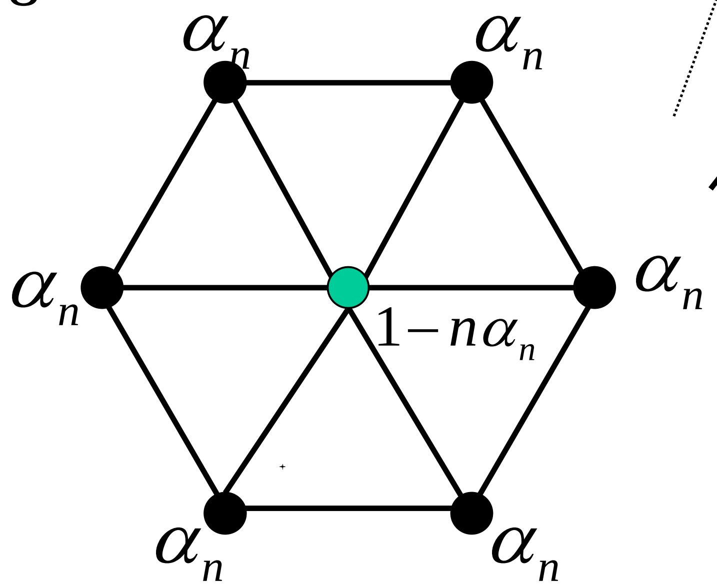


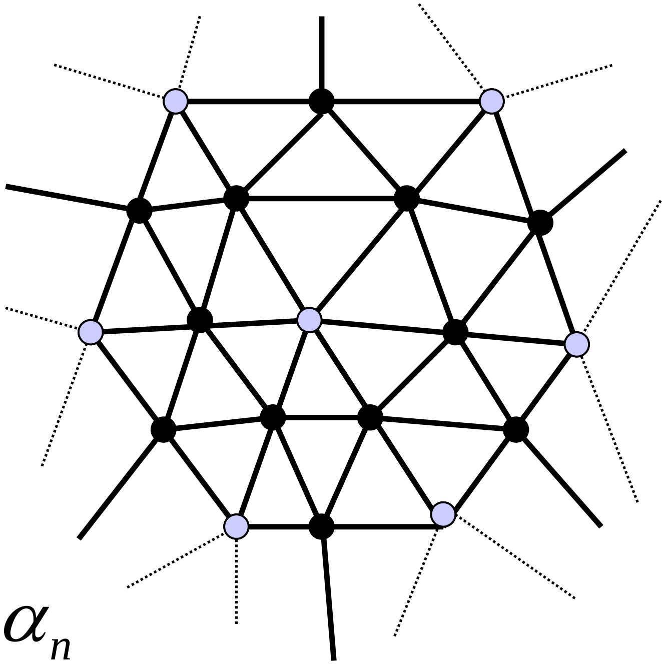




Loop Subdivision Masks





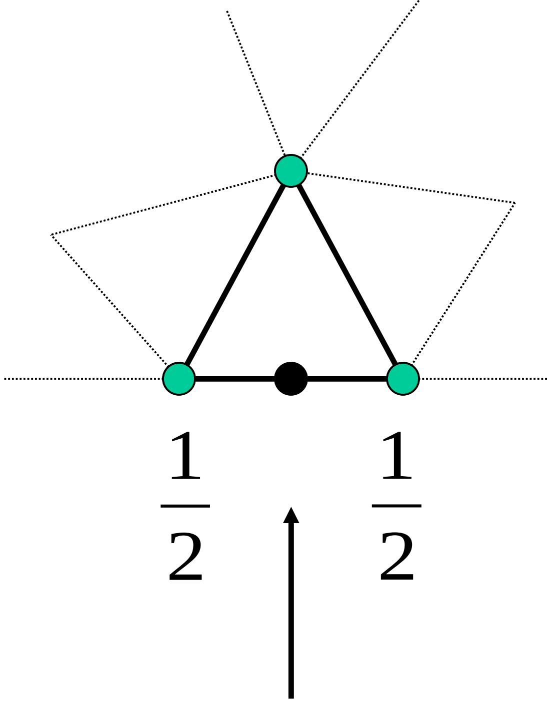


$$\alpha_n = \frac{3}{8n}, \alpha_3 = \frac{3}{16}$$

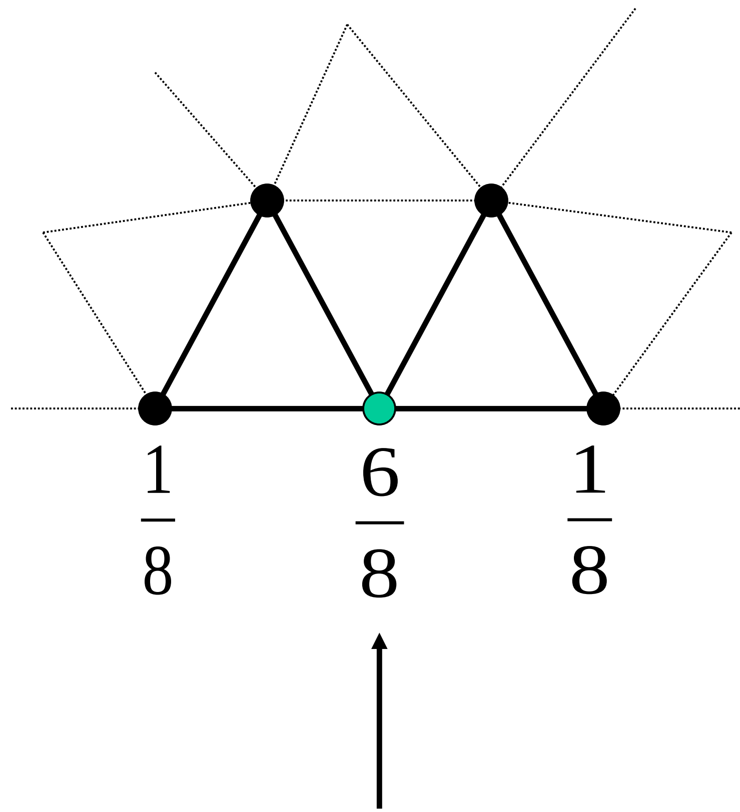
Notice sum of coeffs…

Loop Subdivision Boundaries

Subdivision Mask for boundaries:

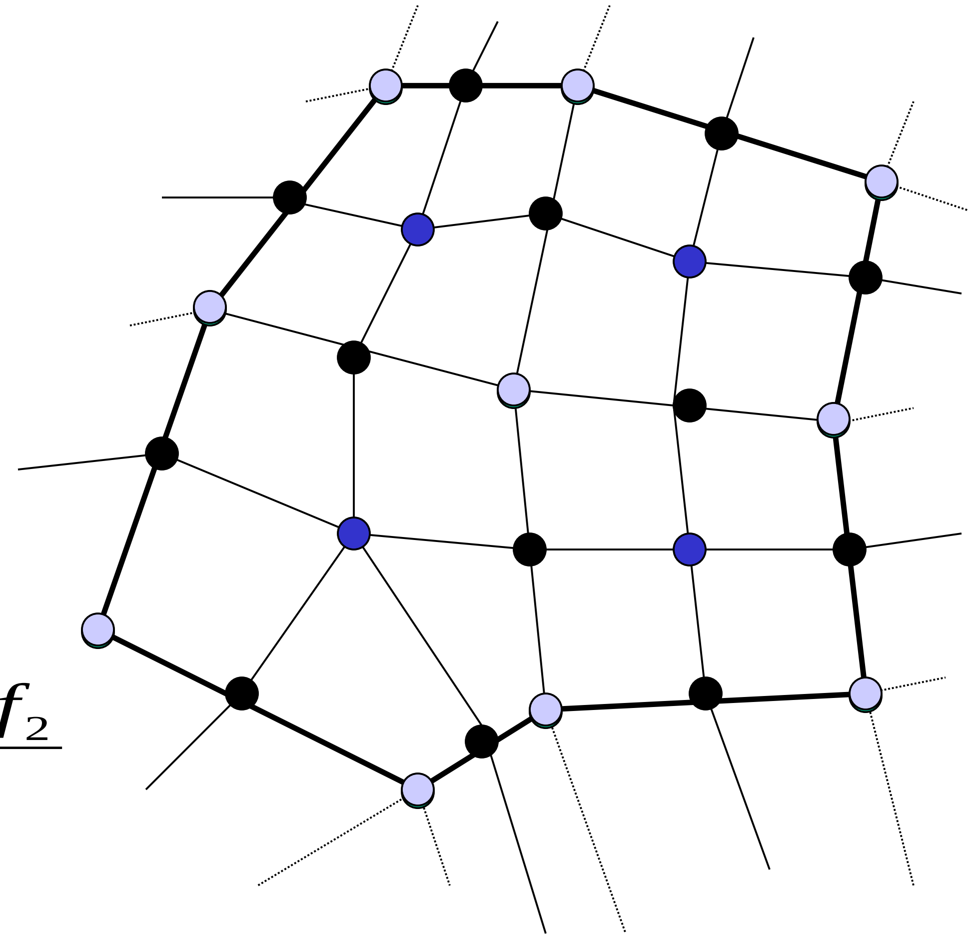


Edge Rule



Vertex Rule

Catmull-Clark Subdivision ('78)



FACE

$$f = \frac{1}{n}\sum_{1}^{n} v_i$$

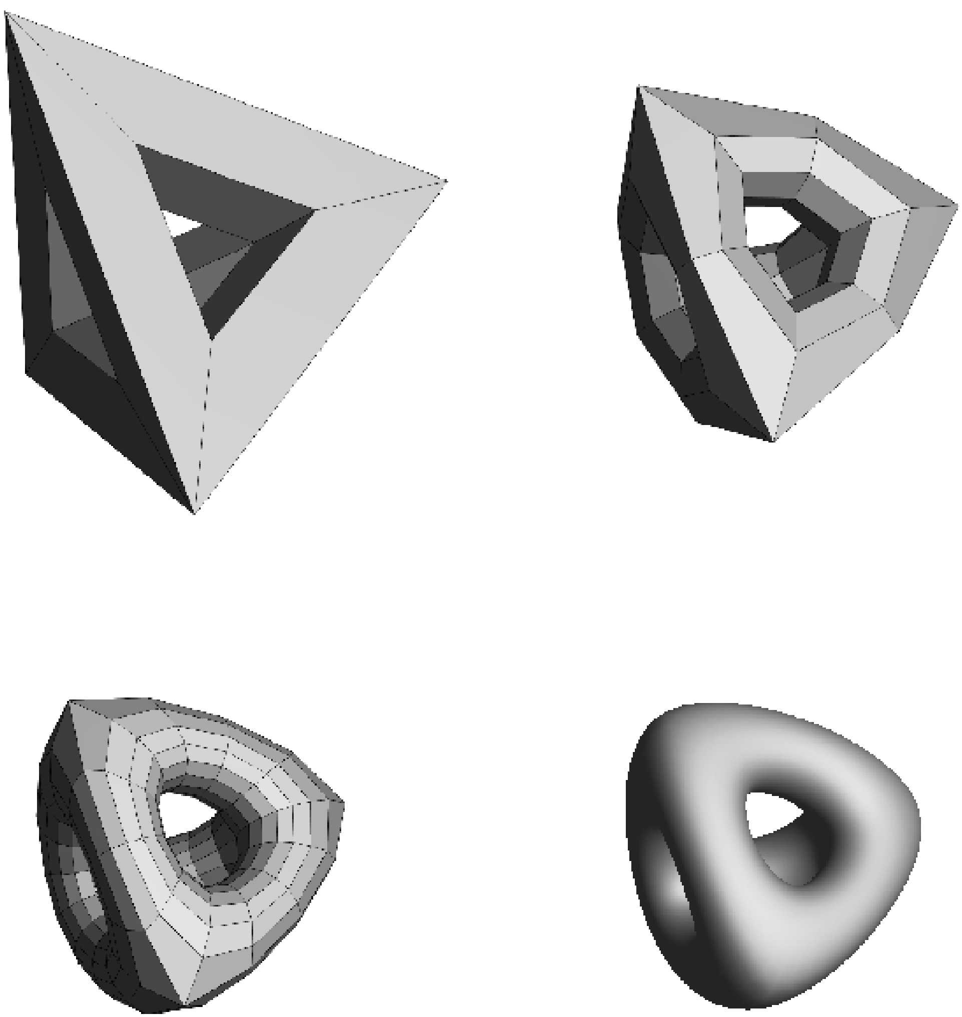
EDGE

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

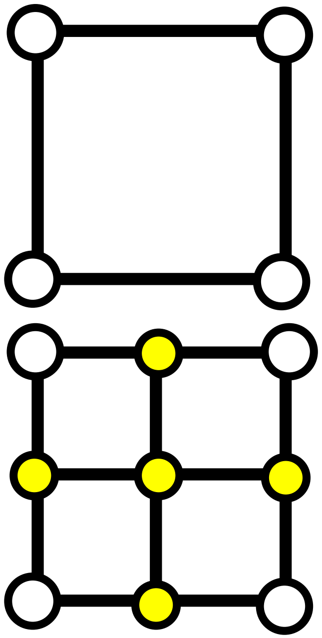
VERTEX

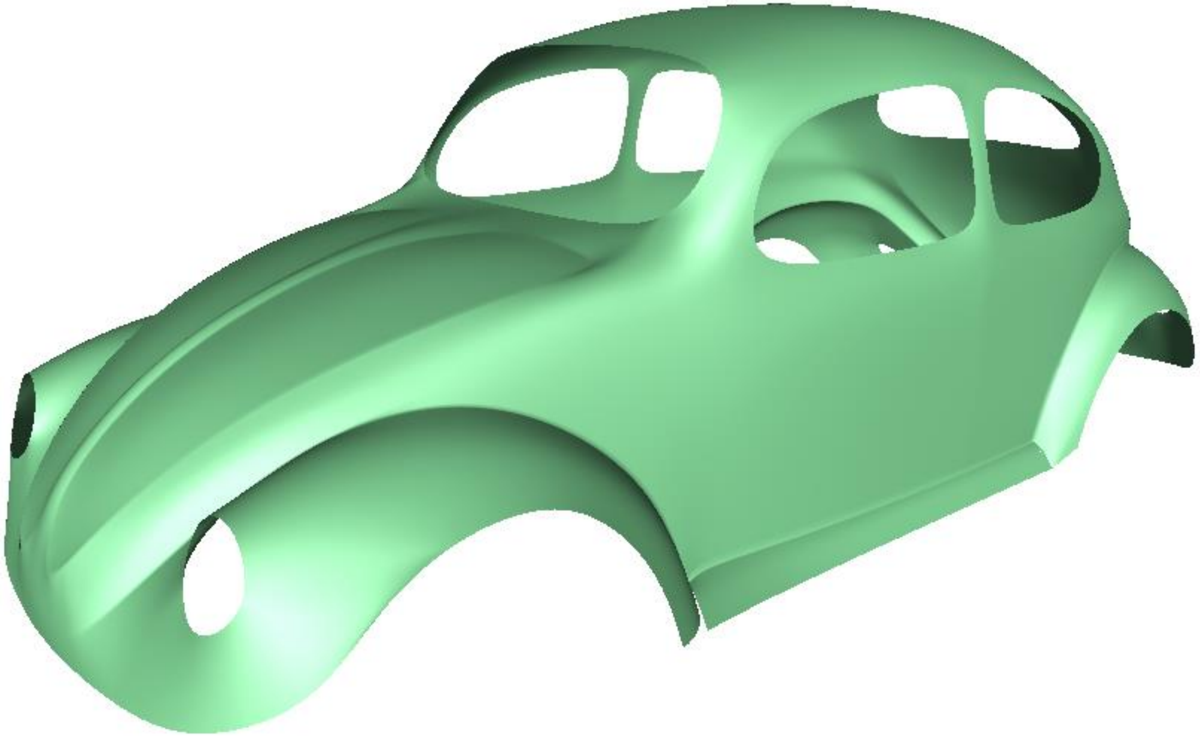
$$v_{i+1} = \frac{n-2}{n} v_i + \frac{1}{n^2}\sum_j e_j + \frac{1}{n^2}\sum_j f_j$$

CC Subdivision - Example

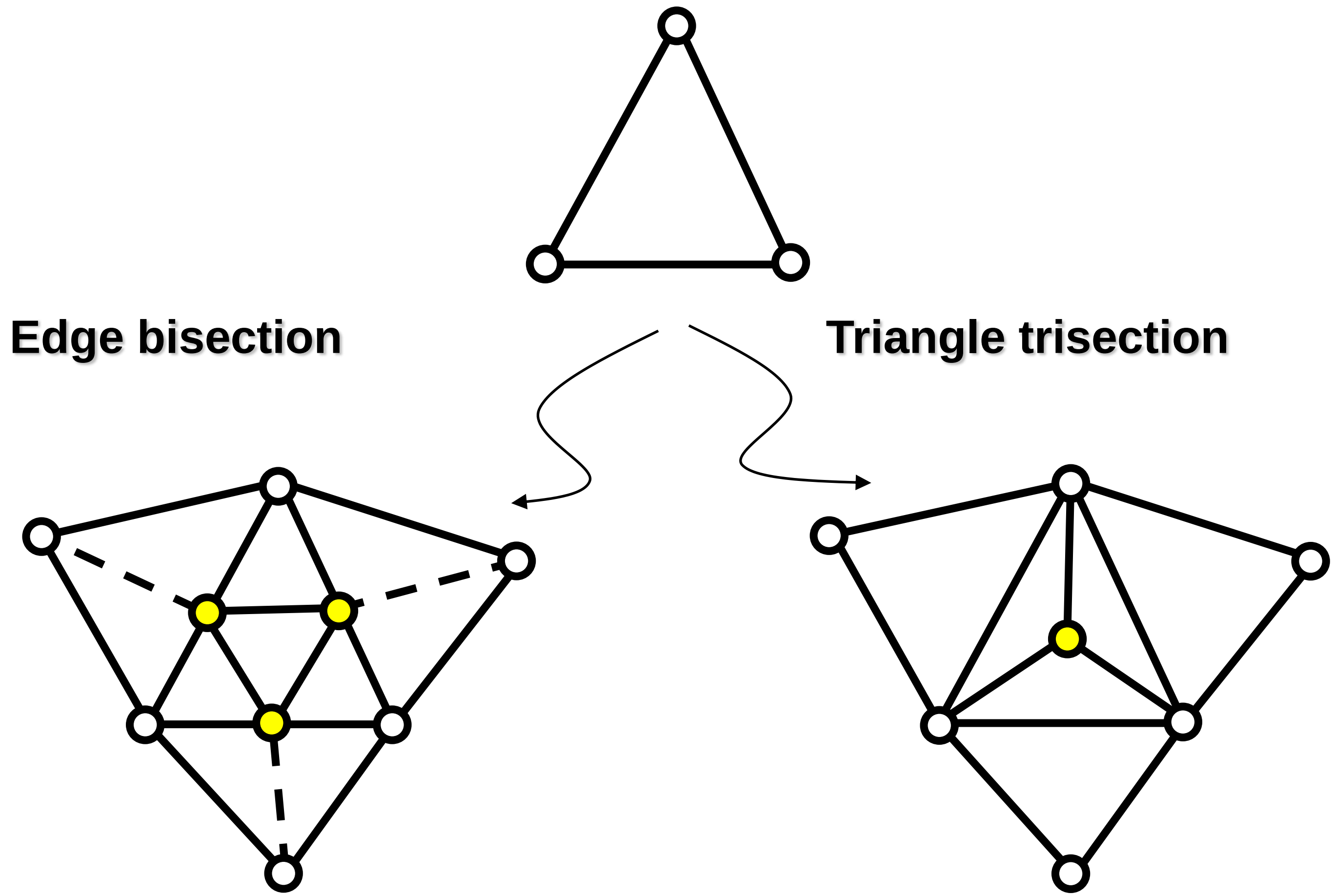


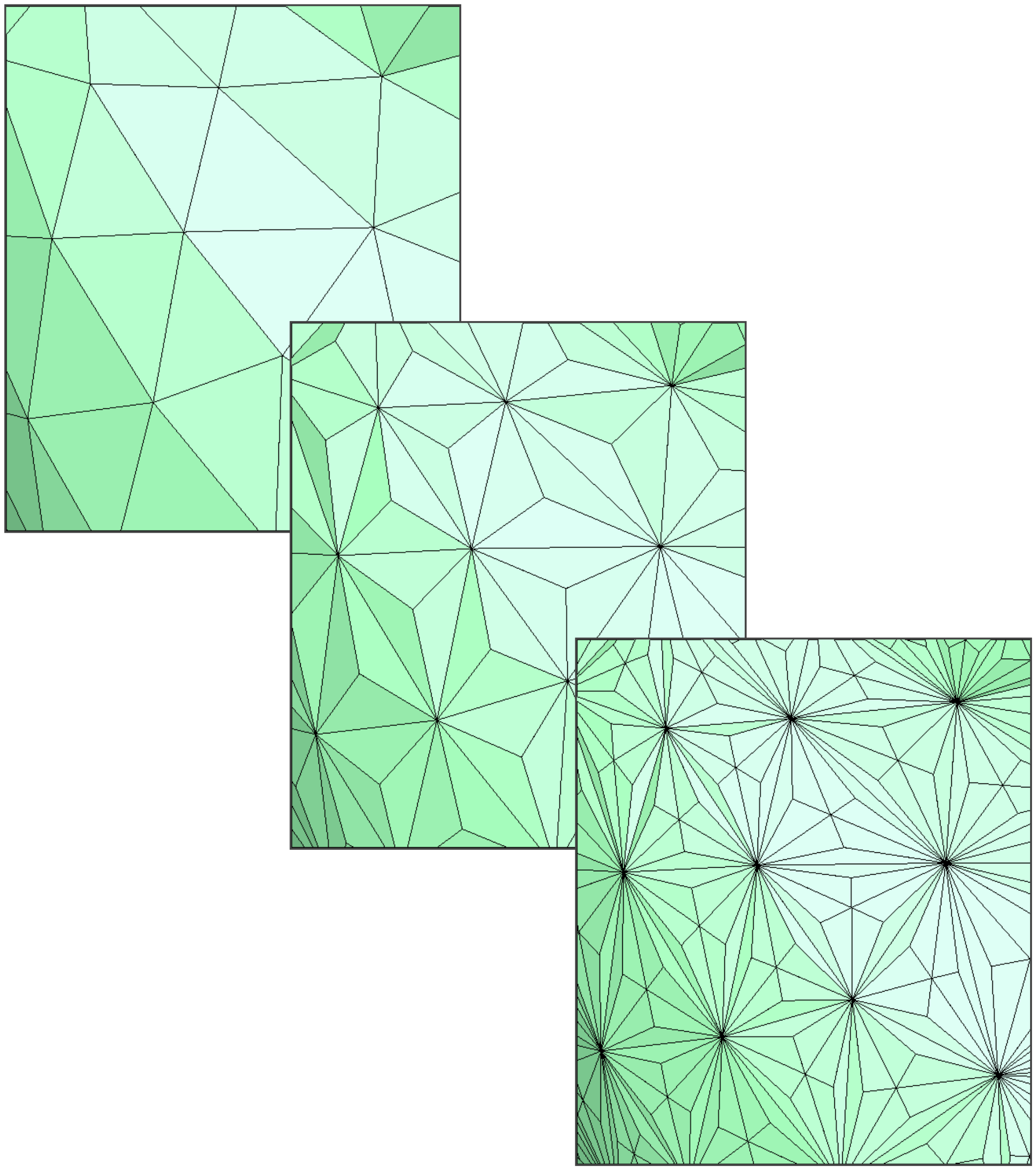
Catmull-Clark

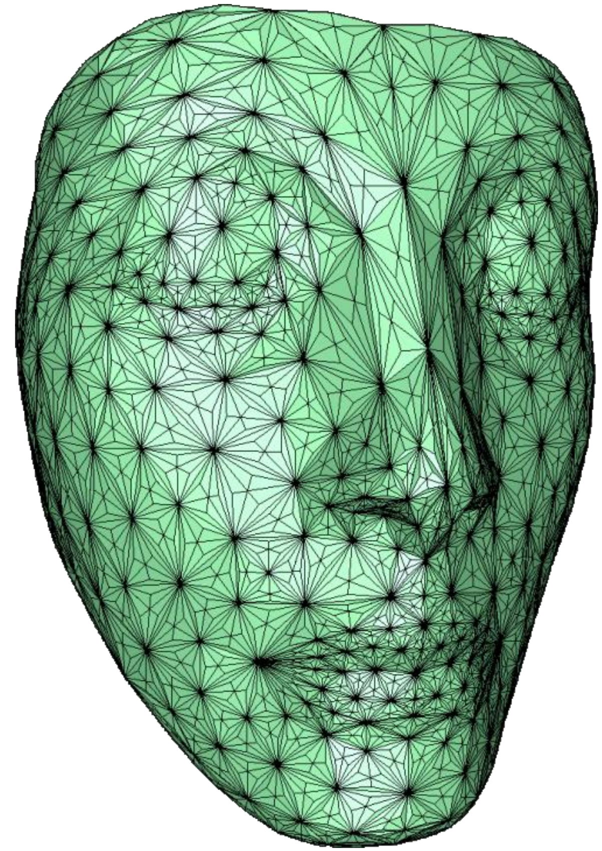




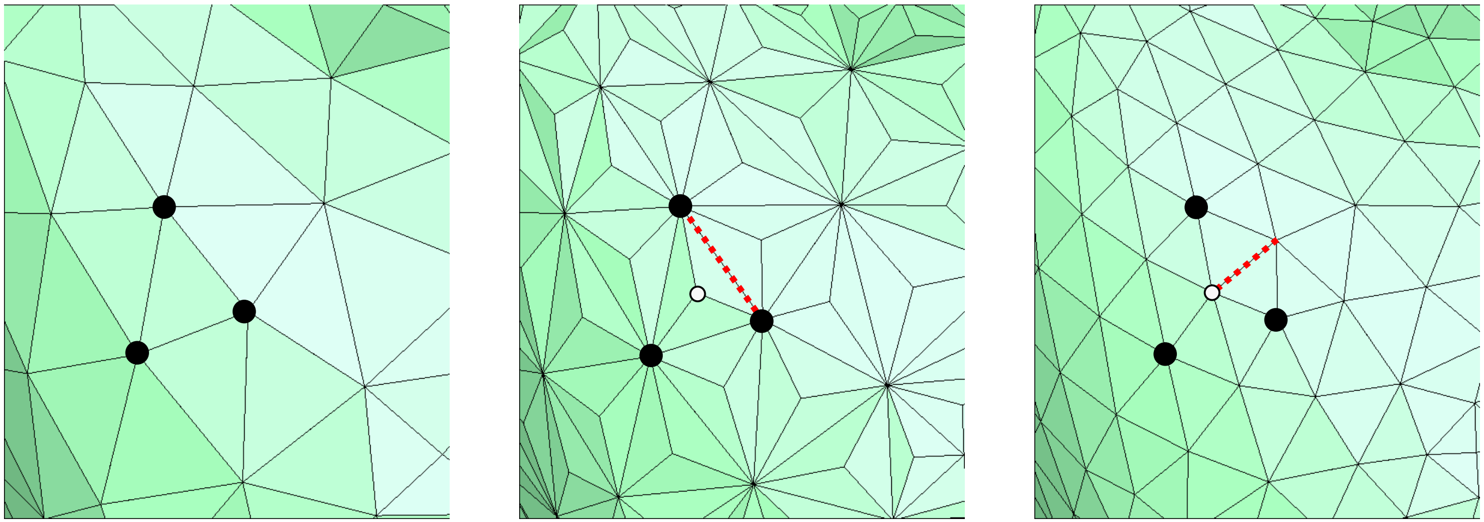
Localised Subdivision?



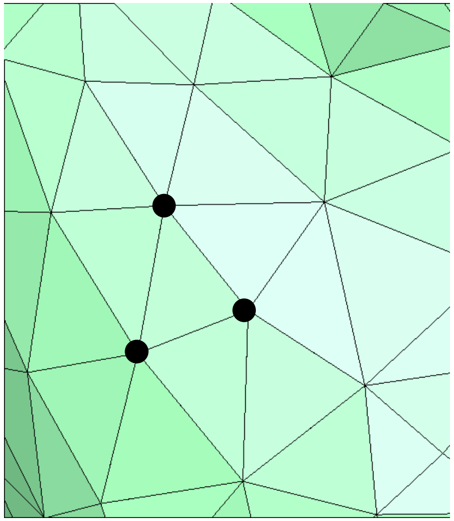




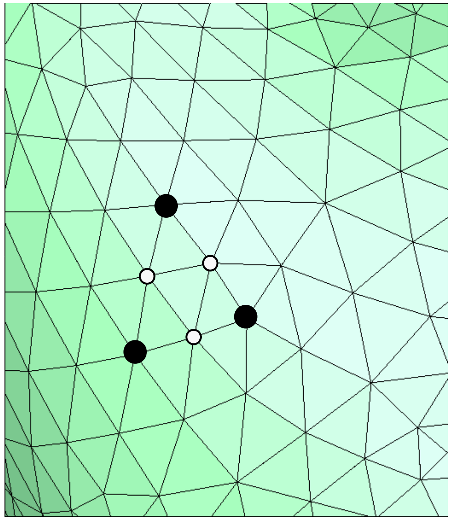
√3-Subdivision [Kobbelt]

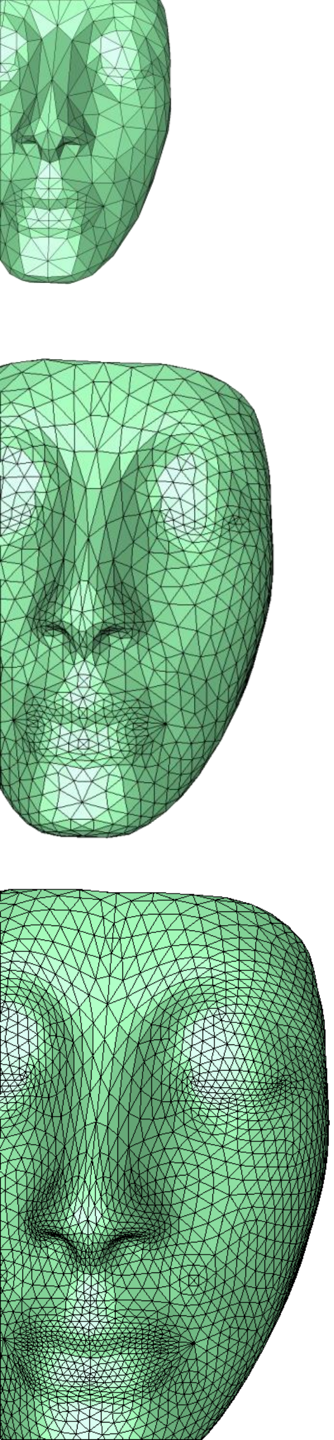


1->3

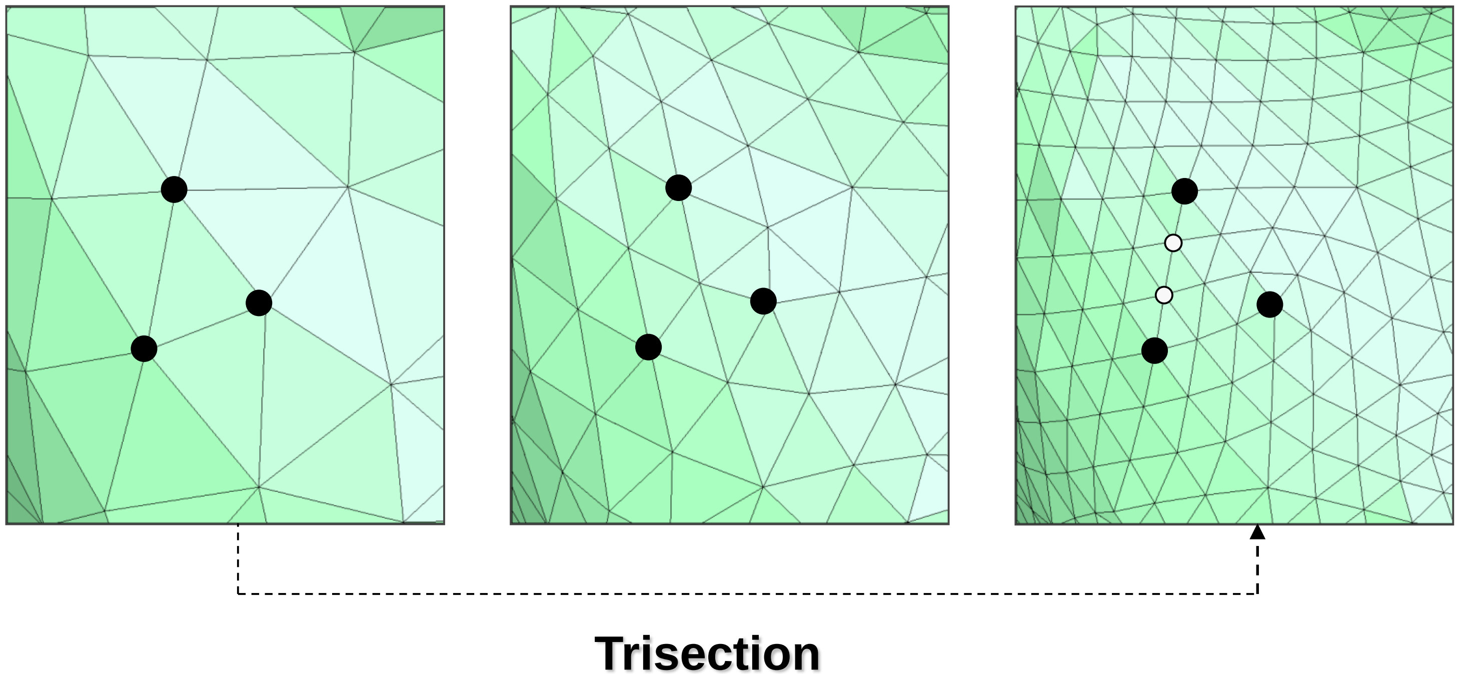


1->4 [Loop]

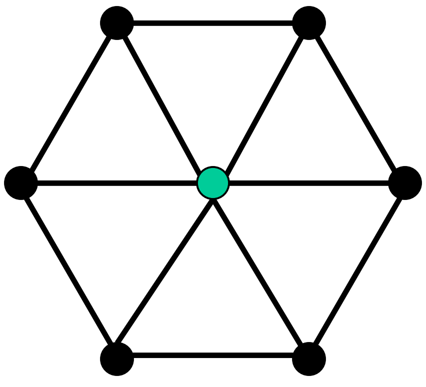




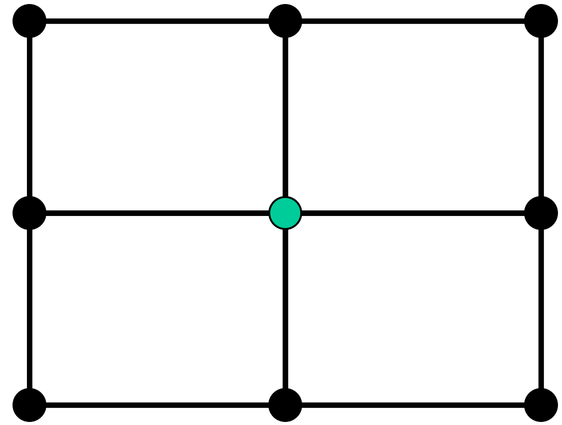
√3-Subdivision [Kobbelt]



Ordinary and *Extraordinary*



Loop Subdivision
Valence 6



Catmull-Clark Subdivision
Valence 4

•Subdividing a mesh does not add ***extraordinary*** vertices.

•Subdividing a mesh does not remove ***extraordinary*** vertices.

How should ***extraordinary*** vertices be handled?

•Make up rules for ***extraordinary*** vertices that keep the surface as “smooth” as possible.

Subdivision as Matrices

Subdivision can be expressed as a matrix S_{mask} of weights w.

- S_{mask} is very sparse
- Never implemented this way
- Allows for analysis of:
  - Curvature
  - Limit Surface

$$S_{mask} P = \hat{P}$$

$$\begin{bmatrix} w_{00} & w_{01} & \cdots & 0 \\ w_{10} & w_{11} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & w_{nj} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} \hat{p}_0 \\ \hat{p}_1 \\ \hat{p}_2 \\ \vdots \\ \hat{p}_0 \end{bmatrix}$$

S_{mask} Weights

Old Control Points

New Points

Subdivision in production environment.

- Traditionally spline patches (NURBS) have been used in production for character animation.
- Difficult to control spline patch density in character modelling.

Subdivision in production environment.

Pixar Animation Studios

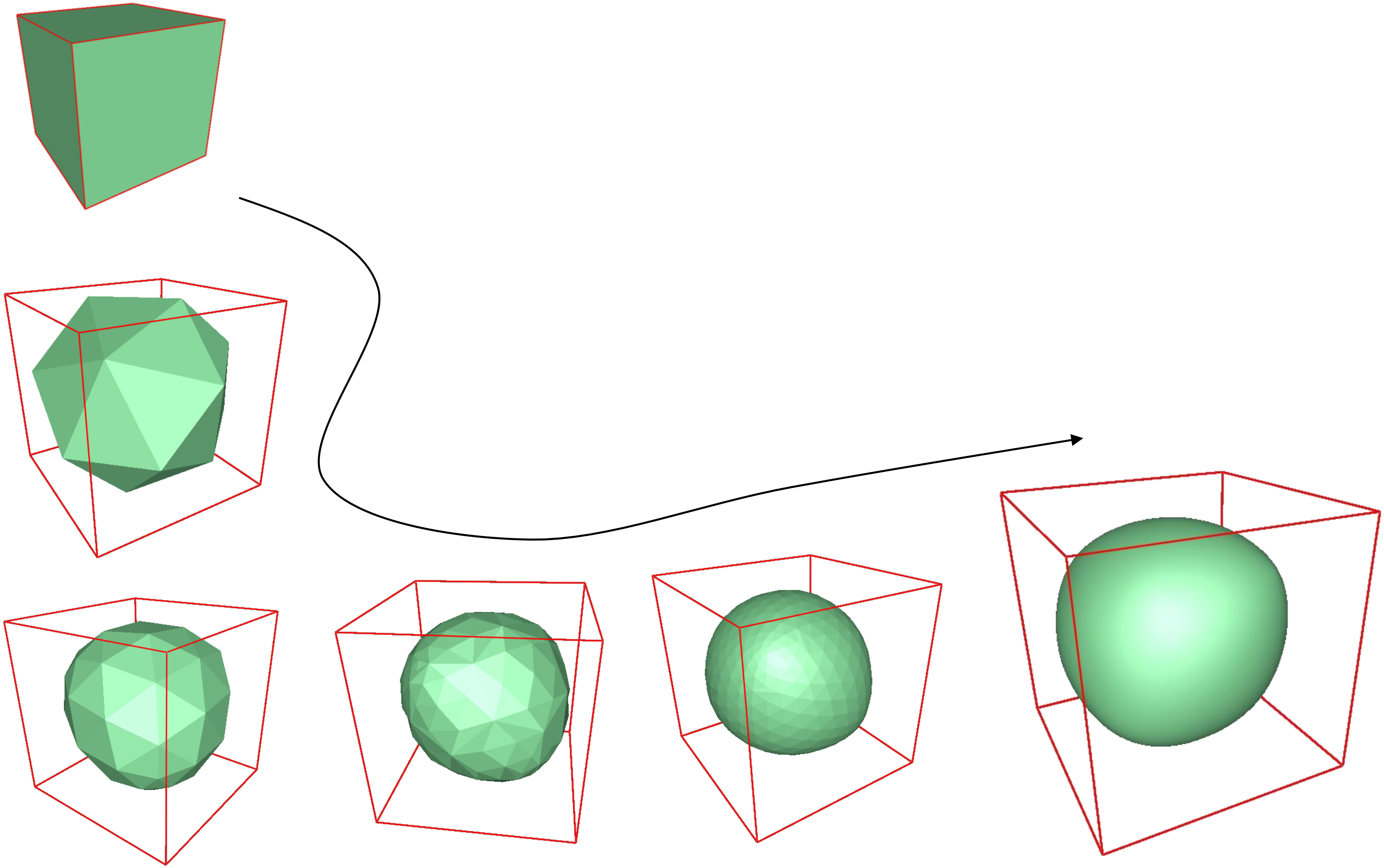
Modeling with Catmull-Clark

- Subdivision produces smooth continuous surfaces.
- How can "sharpness" and creases be controlled in a modeling environment?

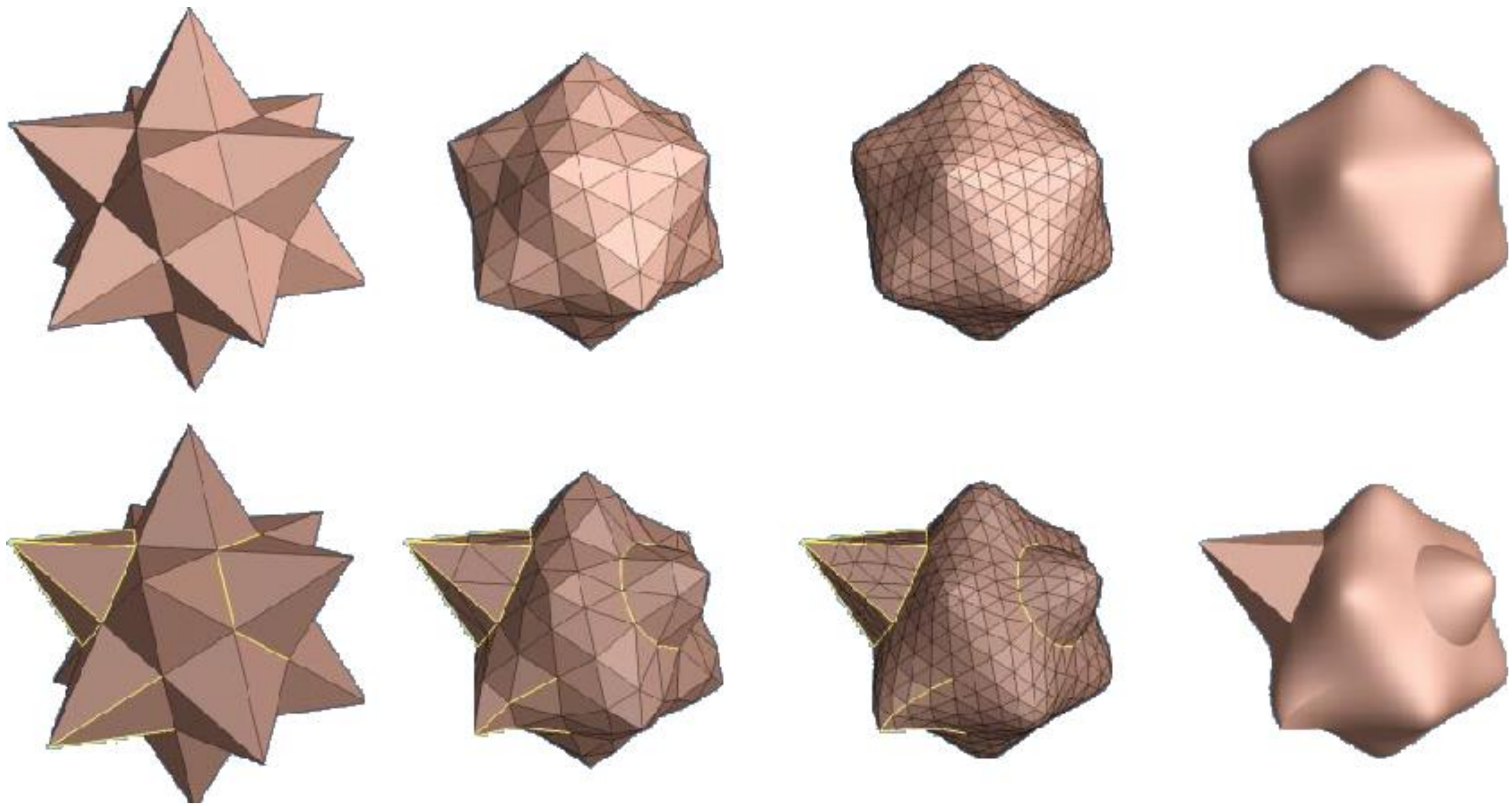
Answer: Define new subdivision rules for "creased" edges and vertices.

1. Tag Edges sharp edges.
2. If an edge is sharp, apply new subdivision rules.
3. Otherwise subdivide with normal rules.

Smoothing (isotropic)



Sharp Edges



~anisotropic smoothing

Courtesy H.Hoppe

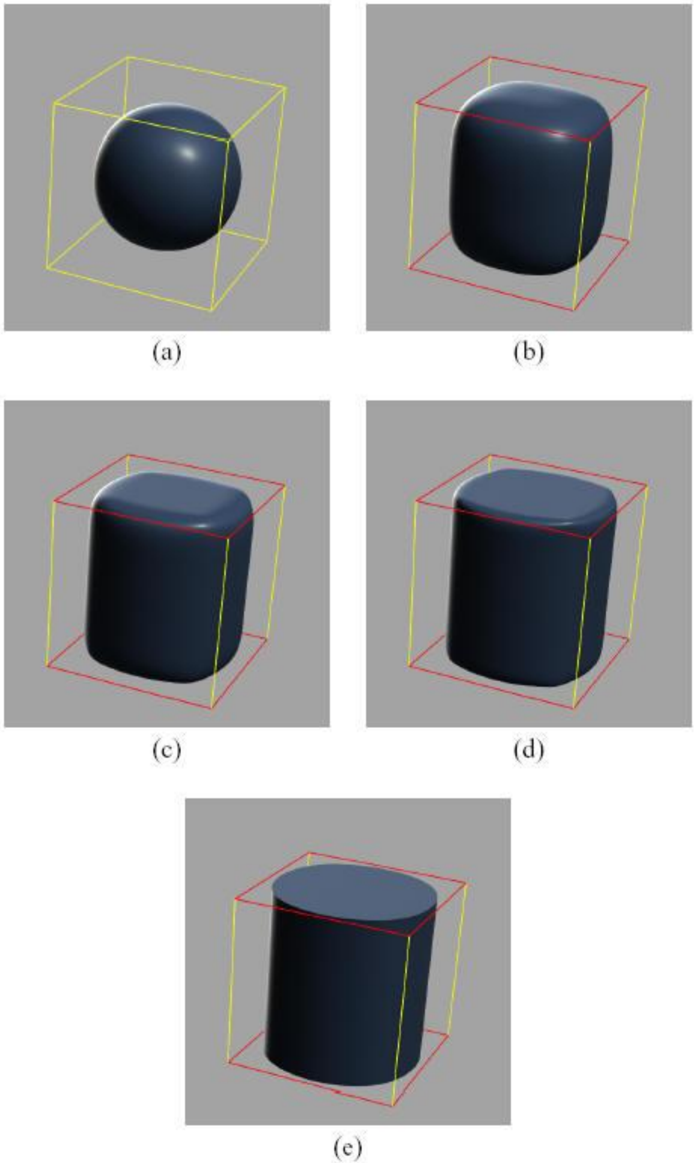
Semi-Sharp Edges…

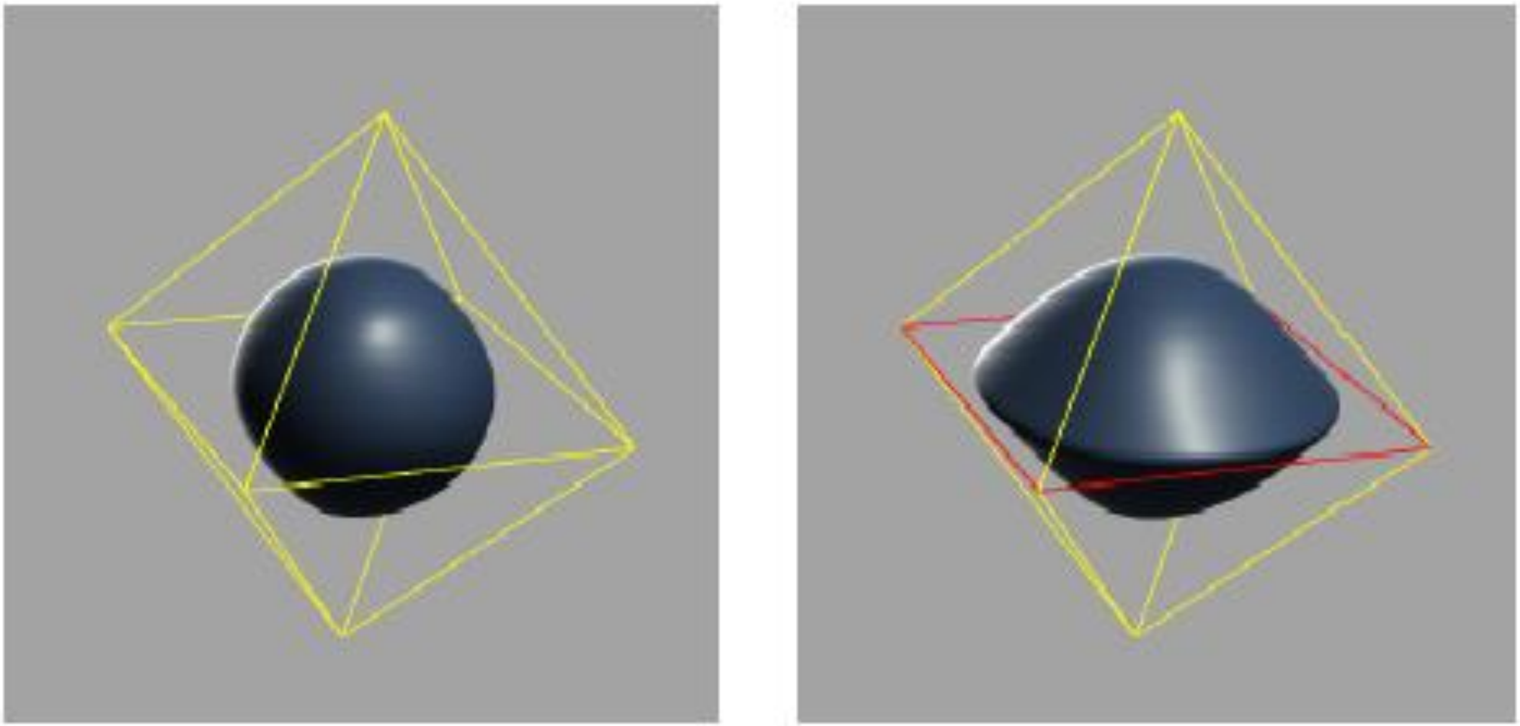
1. Tag Edges as “**sharp**” or “not-sharp”
   - n = 0 – “**not sharp**”
   - n > 0 – sharp

During Subdivision,

2. if an edge is “sharp”, use sharp subdivision rules. Newly created edges, are assigned a sharpness of n-1.
3. If an edge is “not-sharp”, use normal smooth subdivision rules.

IDEA: Edges with a sharpness of “n” do not get subdivided smoothly for “n” iterations of the algorithm.

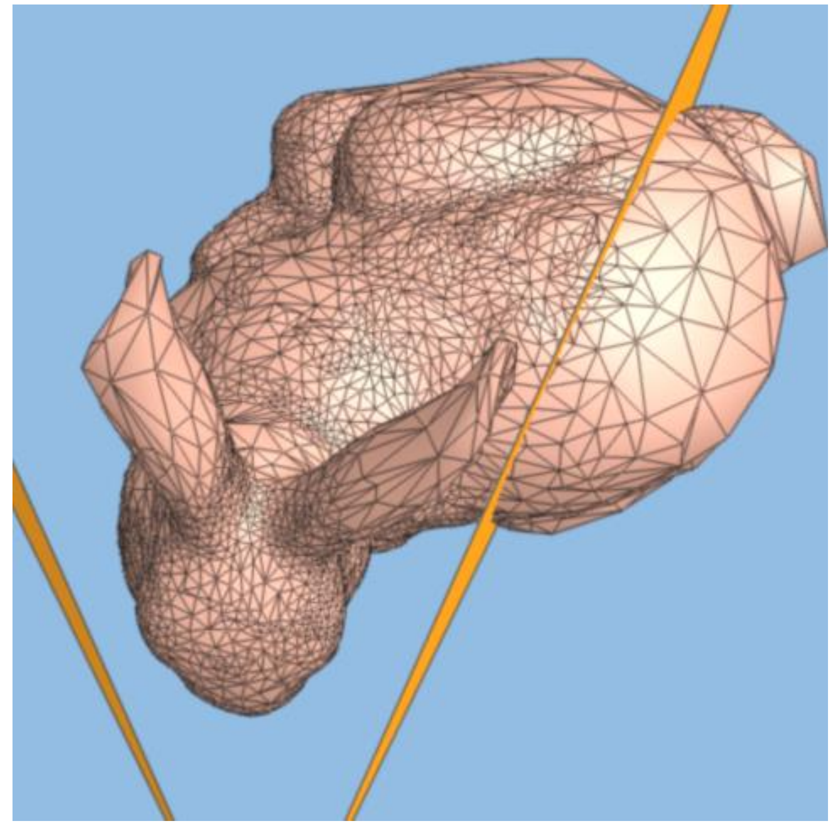


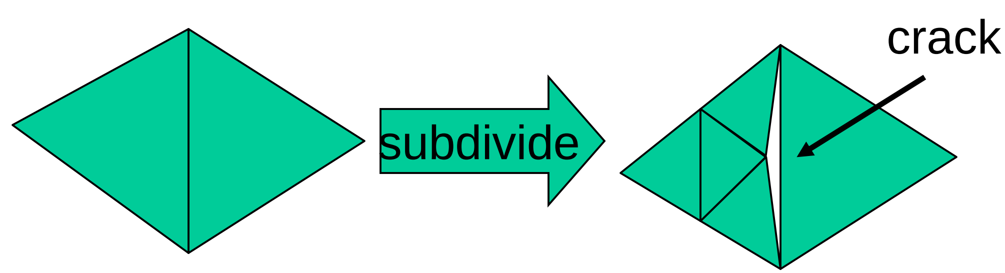


Courtesy T.DeRose

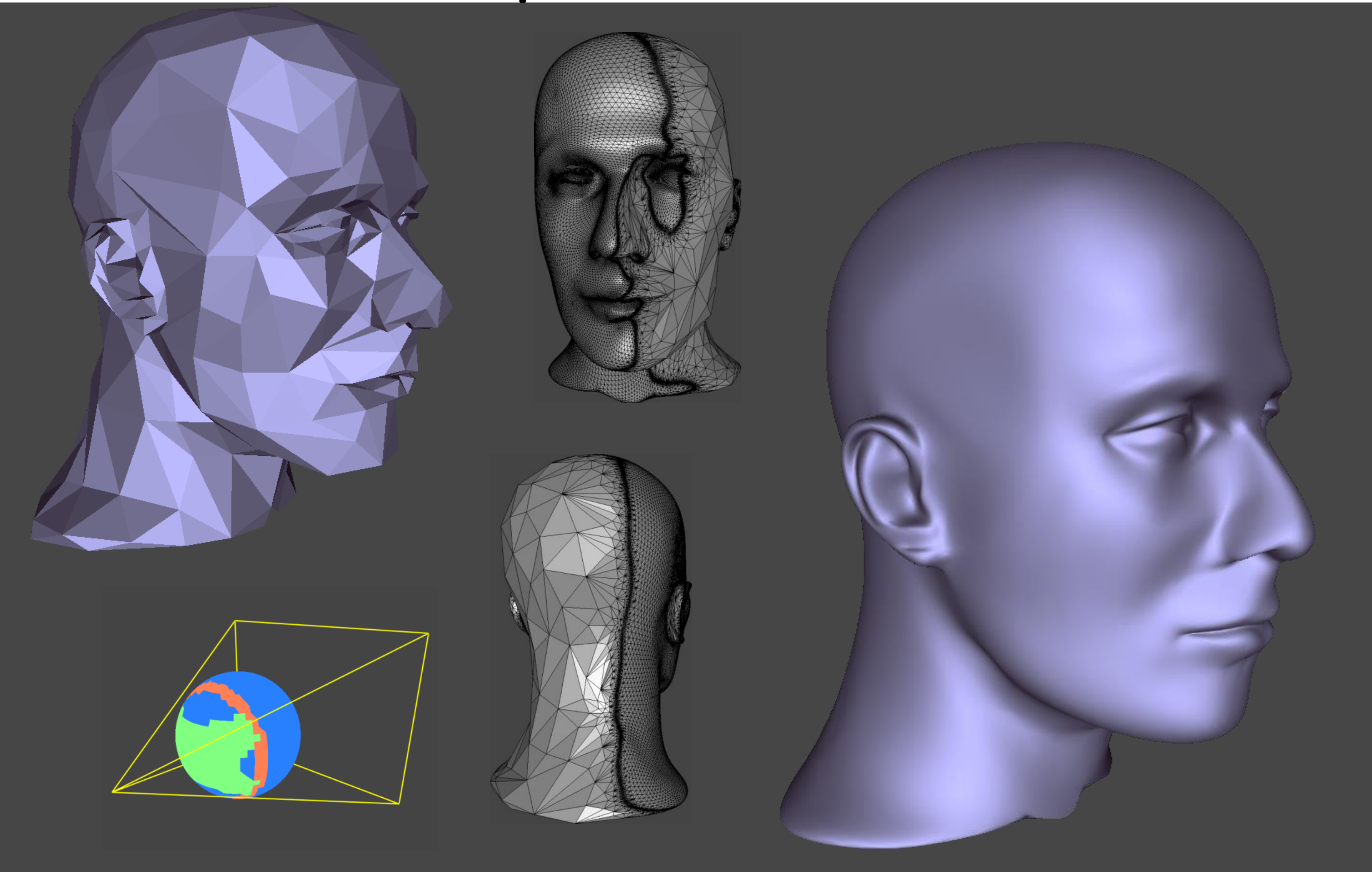
Adaptive Subdivision

- Not all regions of a model need to be subdivided.
- Idea: Use some criteria and adaptively subdivide mesh where needed.
  - Curvature
  - Screen size (make triangles < size of pixel)
  - View dependence
    - Distance from viewer
    - Silhouettes
    - In view frustum
  - Careful! Must ensure that "cracks" aren't made





Adaptive Subdivision

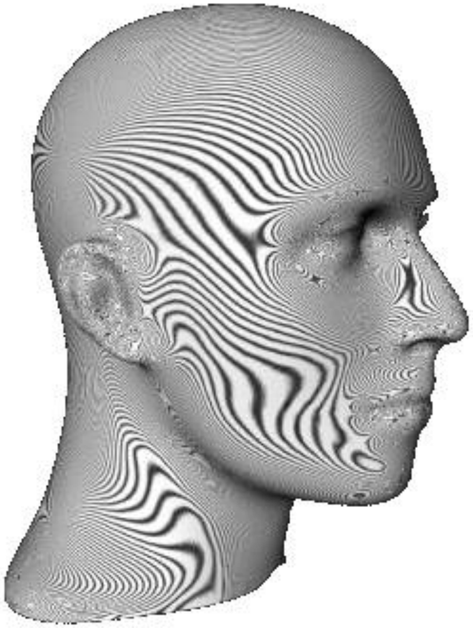


Zoo

- Schemes:
  - interpolants vs approximants
  - dual, primal, √3, quads, triangles, polygons
  - bisection, N-section, alternate dual/primal
- Convergence -> limit surface
- Continuity and differentiability
- Piecewise smooth
- Exact evaluation
- Non stationary schemes
- Distinguish concave / convex

Challenges

- Non-stationnarity
- Variational approach
- Behavior / extraordinary vertices
- G1 differentiability
- etc.



courtesy L.Kobbelt