Lecture 17:

Subdivision Surfaces

Contents

How do we ...

- Represent 3D objects in a computer?
- Construct such representations quickly and / or automatically with a computer?
- Manipulate 3D objects with a computer?

3D Representations provide the foundations for

- Computer Graphics
- Computer-Aided Geometric Design
- Visualization
- Robotics, …

Different methods for different object representations

Raw data

- Range image
- Point cloud
- Polygon soup

Surfaces

- Mesh
- Subdivision
- Parametric
- Implicit

Solids

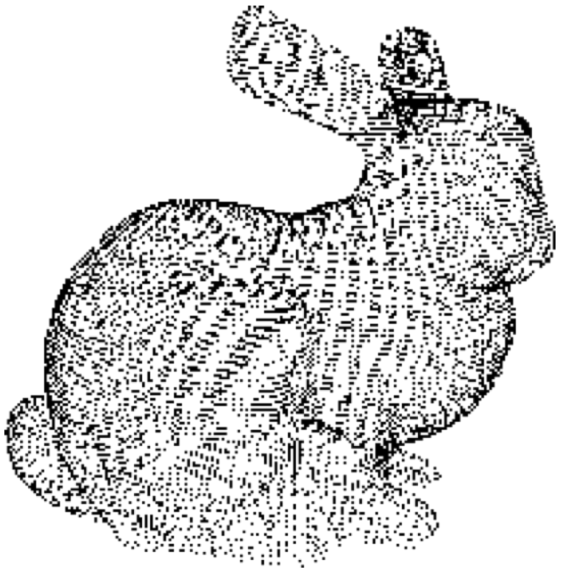
- Voxels
- BSP tree
- Constructive solid geometry

Range image

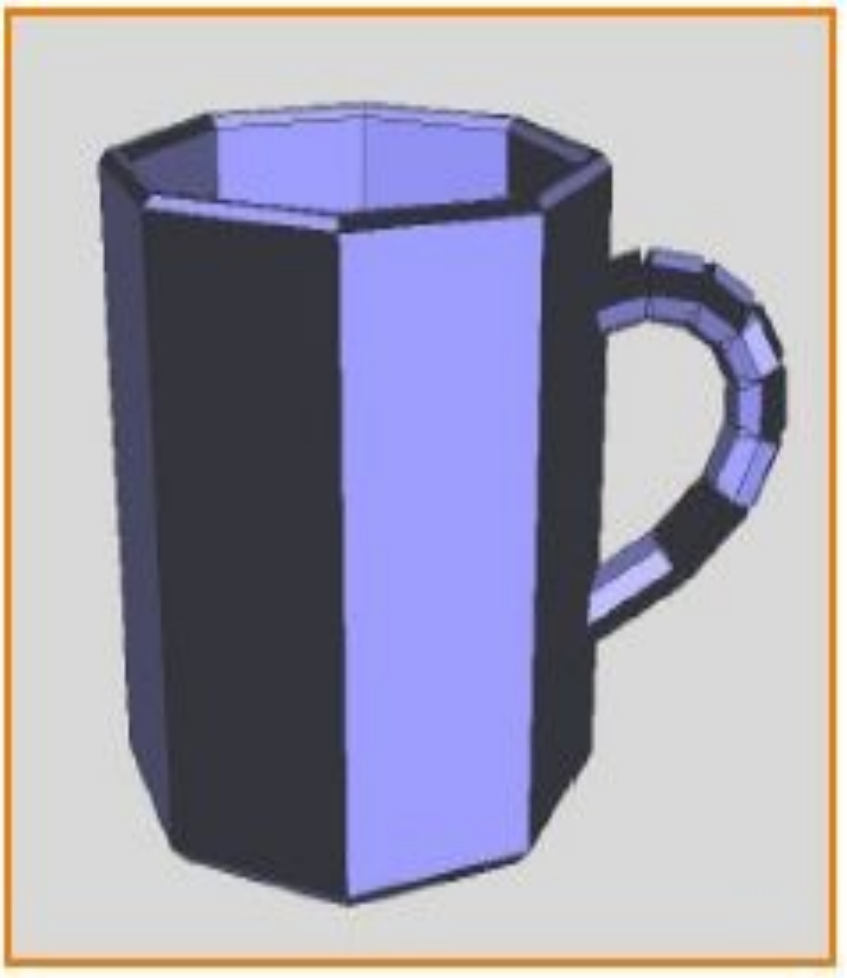
- Acquired from range scanner
  - *E.g.* laser range scanner, structured light, phase shift approach
- Structured point cloud
  - Grid of depth values with calibrated camera
  - 2,5D: 2D plus depth

Unstructured set of 3D point samples

- Often constructed from many range images



Unstructured set of polygons



Raw data

- Range image
- Point cloud
- Polygon soup

Surfaces

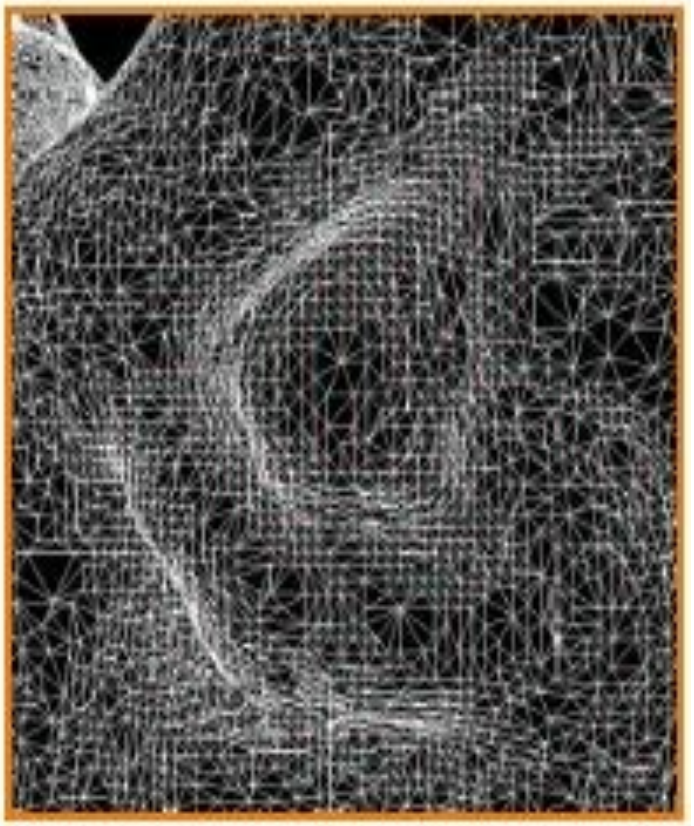
- Mesh
- Subdivision
- Parametric
- Implicit

Solids

- Voxels
- BSP tree
- Constructive solid geometry

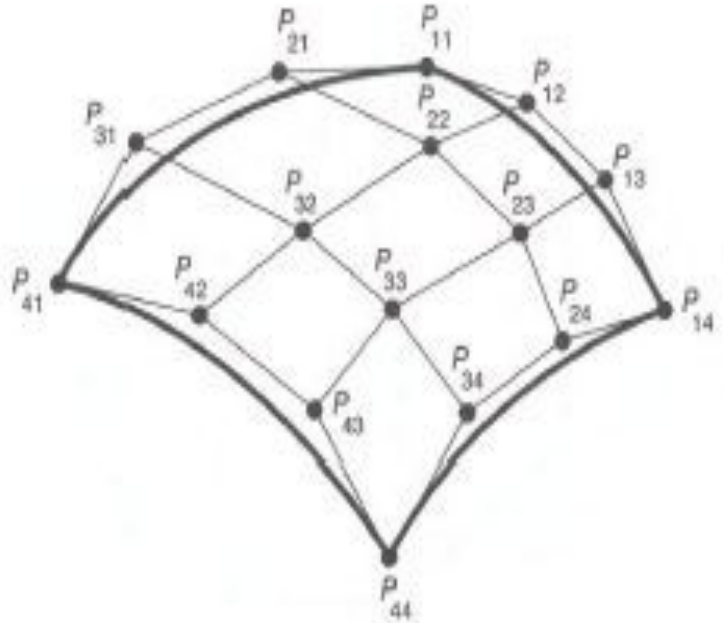
Connected set of polygons

- (usually triangles)



Tensor product spline patches

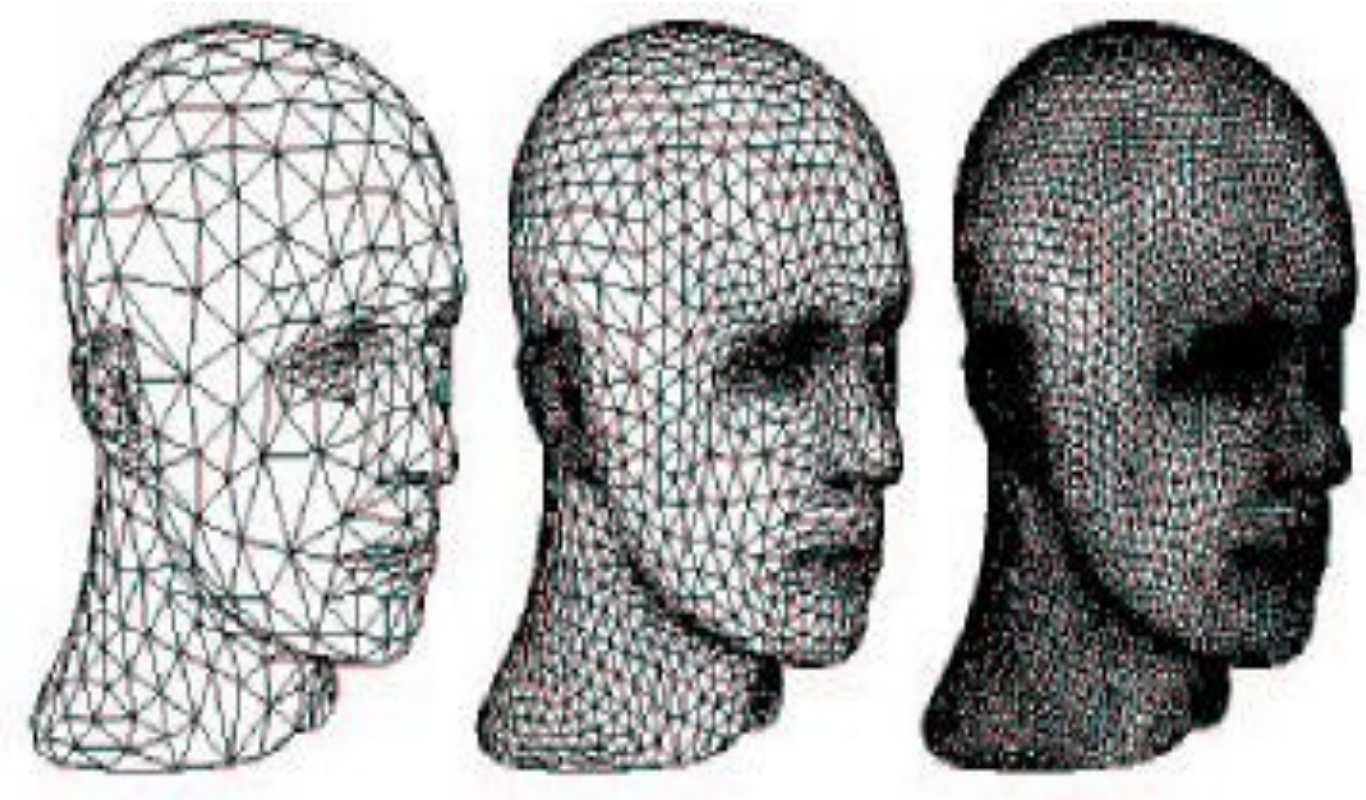
- Careful constraints to maintain continuity



FvDFH Figure 11.44

Coarse mesh & subdivision rule

- Define smooth surface as limit of sequence of refinements



Points satisfying: $F(x, y, z) = 0$

Polygonal Model

Implicit Model

Raw data

- Range image
- Point cloud
- Polygon soup

Surfaces

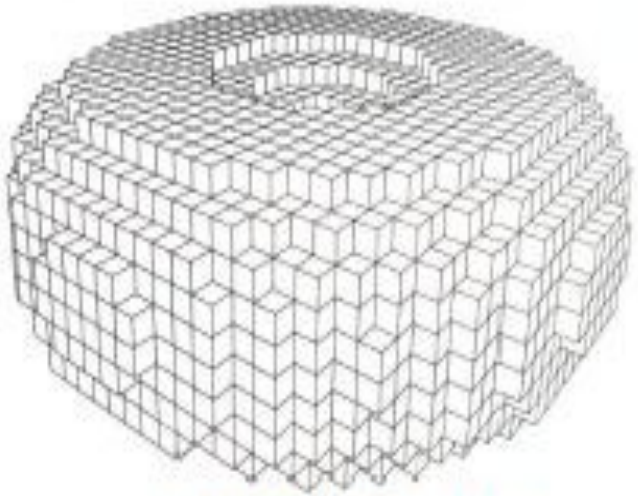
- Mesh
- Subdivision
- Parametric
- Implicit

Solids

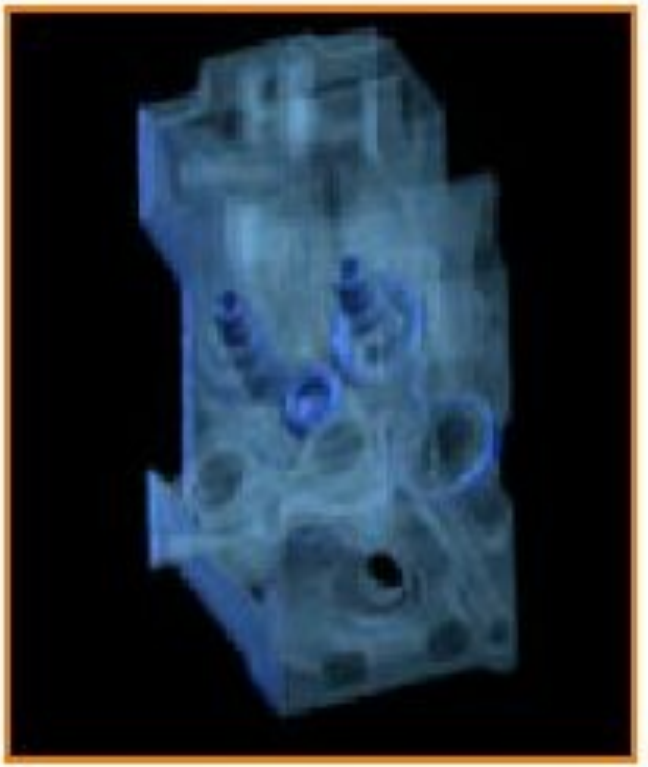
- Voxels
- BSP tree
- Constructive solid geometry

Uniform grid of volumetric samples

- Acquired from CAT, MRI, *etc.*



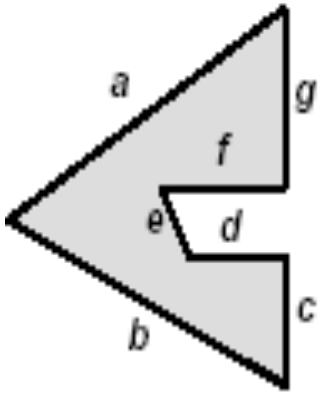
FvDFH Figure 12.20



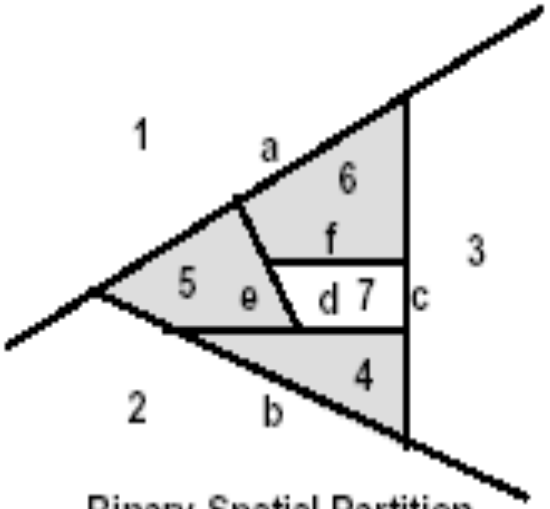
Stanford Graphics Laboratory

Binary space partition with solid cells labeled

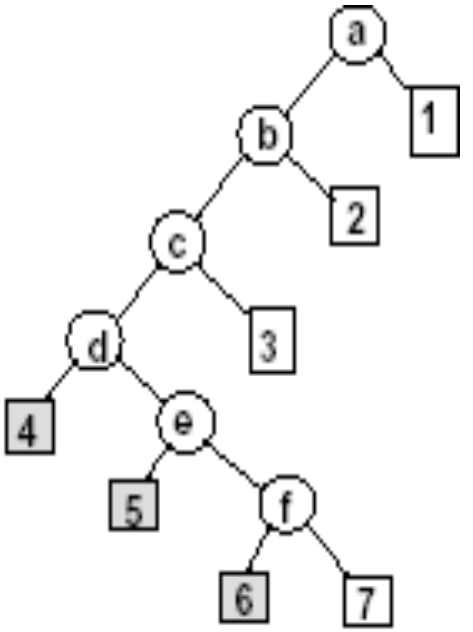
- Constructed from polygonal representations



Object



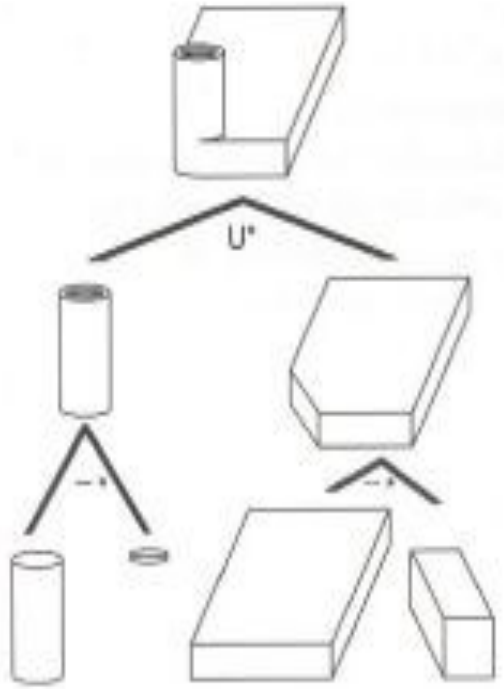
Binary Spatial Partition



Binary Tree

Hierarchy of boolean set operations

- (Union, difference, intersect) applied to simple shapes



FvDFH Figure 12.27

H&B Figure 9.9

Splines

- Traditionally spline patches (NURBS)
  have been used in production for character animation

Difficult to stitch together

- Maintaining continuity is hard

Difficult to model objects with complex topology

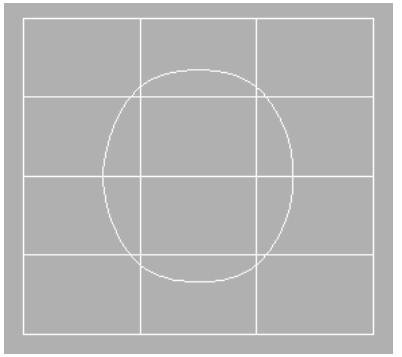
(Geri's Game, Pixar 1998)

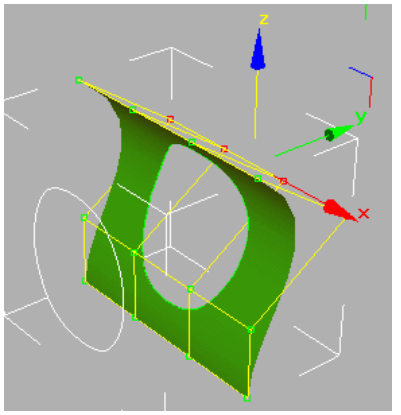
Subdivision in Character Animation

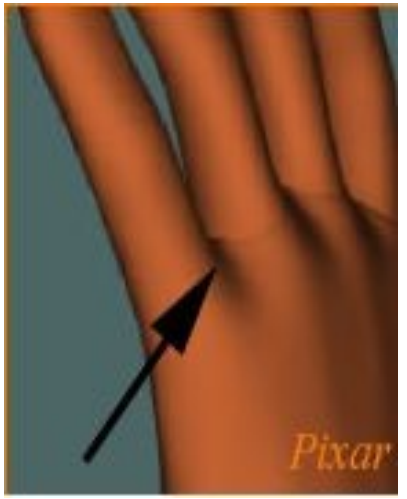
Tony Derose, Michael Kass, Tien Troung (SIGGRAPH '98)

Splines (Bézier, NURBS, …)

- Easy and commonly used in CAD systems
- Most surfaces are not made of quadrilateral patches
  - Need to trim surface: Cut of parts
- Trimming NURBS is expensive and often has numerical errors
- Very difficult to stich together separate surfaces
- Very hard to hide seams

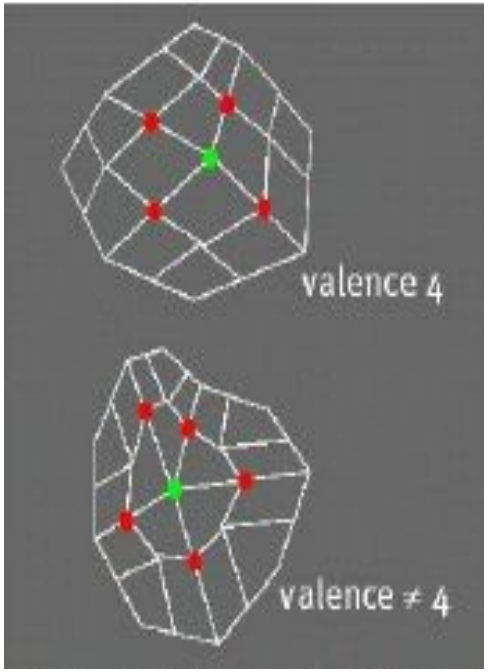


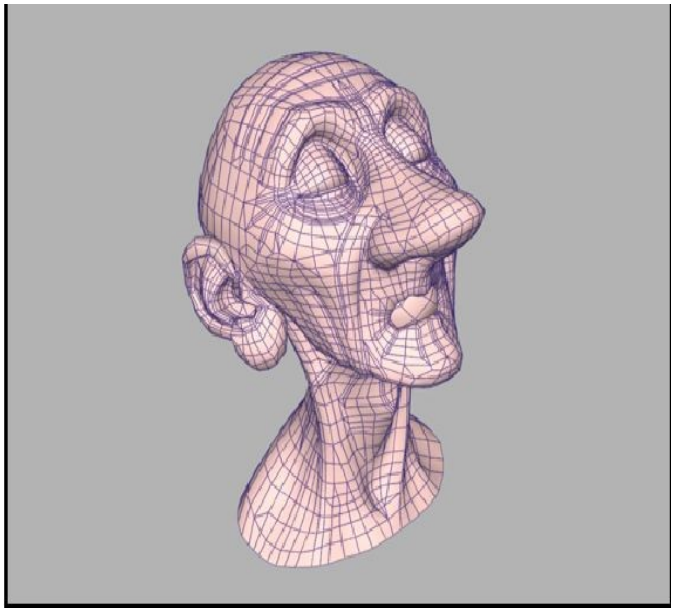




Subdivision methods have a series of interesting properties:

- Applicable to meshes of arbitrary topology (non-manifold meshes)
- No trimming needed
- Scalability, level-of-detail
- Numerical stability
- Simple implementation
- Compact support
- Affine invariance
- Continuity
- Still less tools in CAD systems (but improving quickly)





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

Subdivision surfaces are used for:

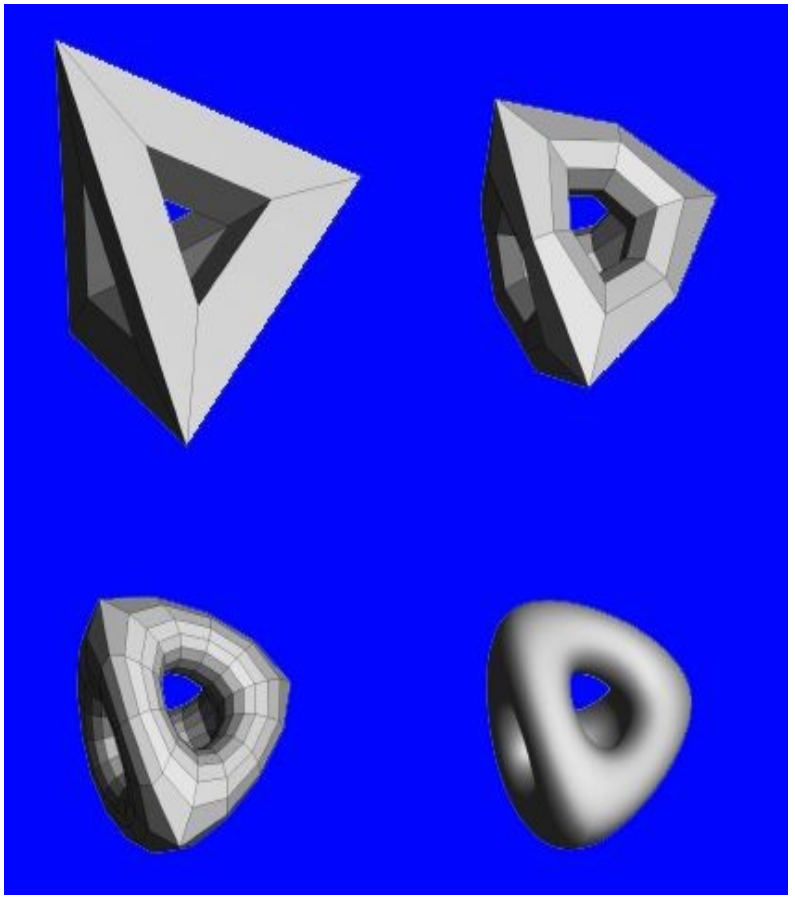
- Geri's hands and head
- Clothes: Jacket, Pants, Shirt
- Tie and Shoes

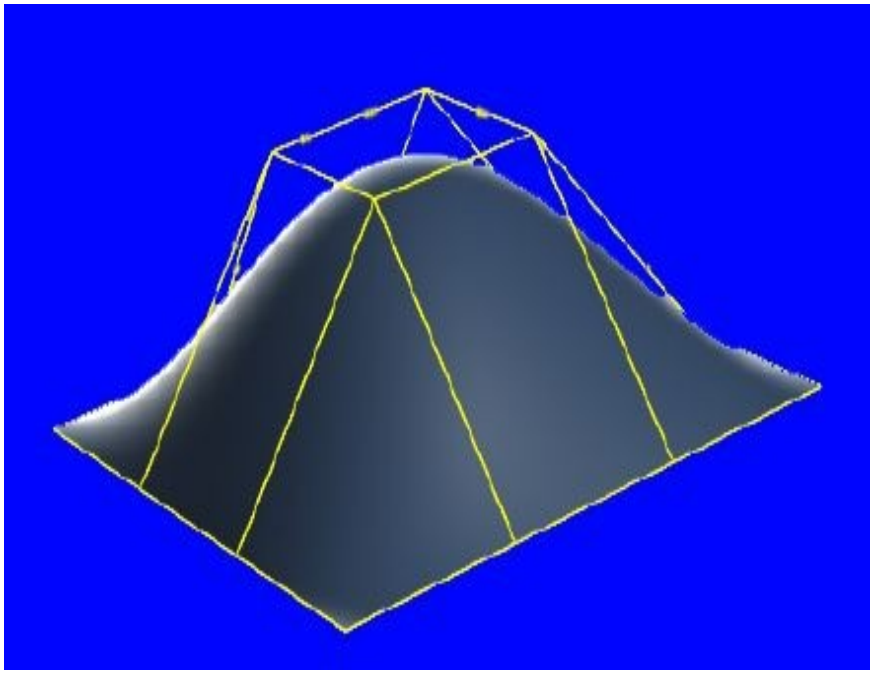
(Geri's Game, Pixar 1998)

Construct a surface from an arbitrary polyhedron

- Subdivide each face of the polyhedron

The limit will be a smooth surface



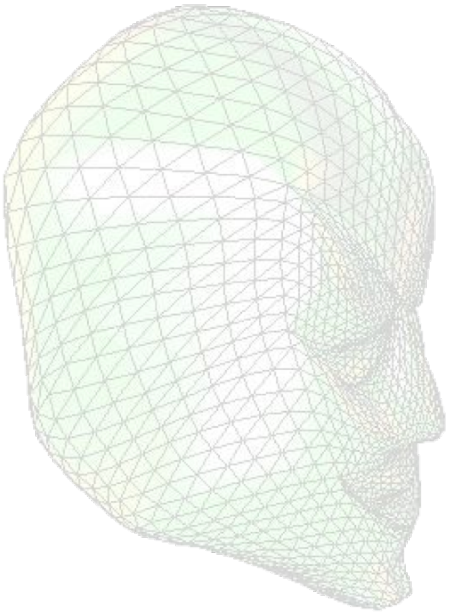


Subdivision curves

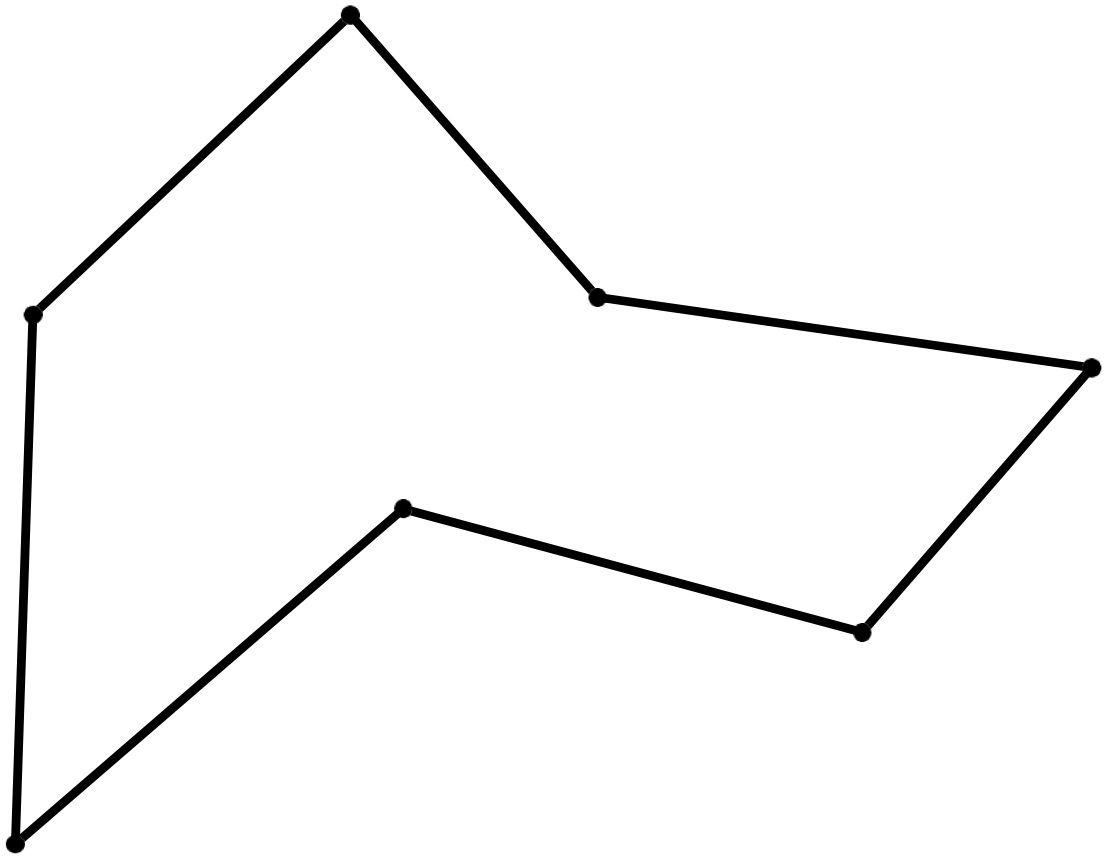
- The basic concepts of subdivision

Subdivision surfaces

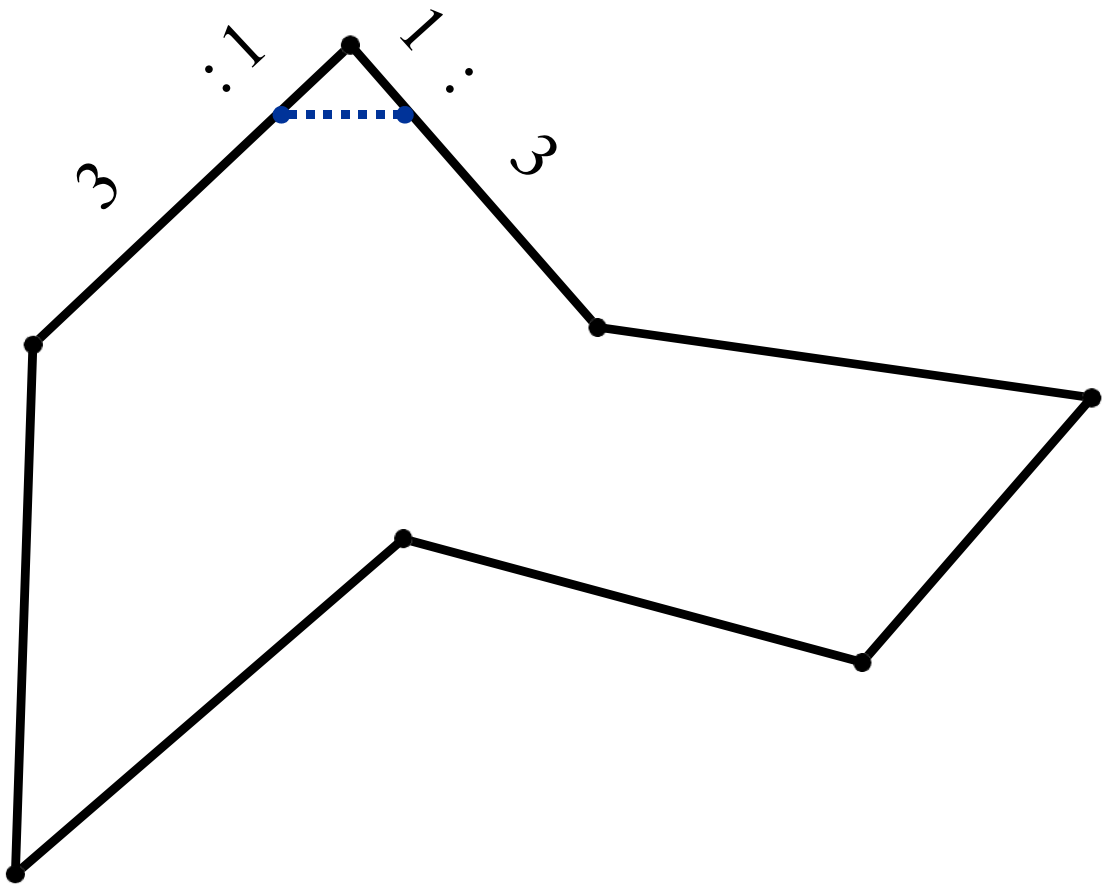
- Important known methods
- Discussion: subdivision vs. parametric surfaces

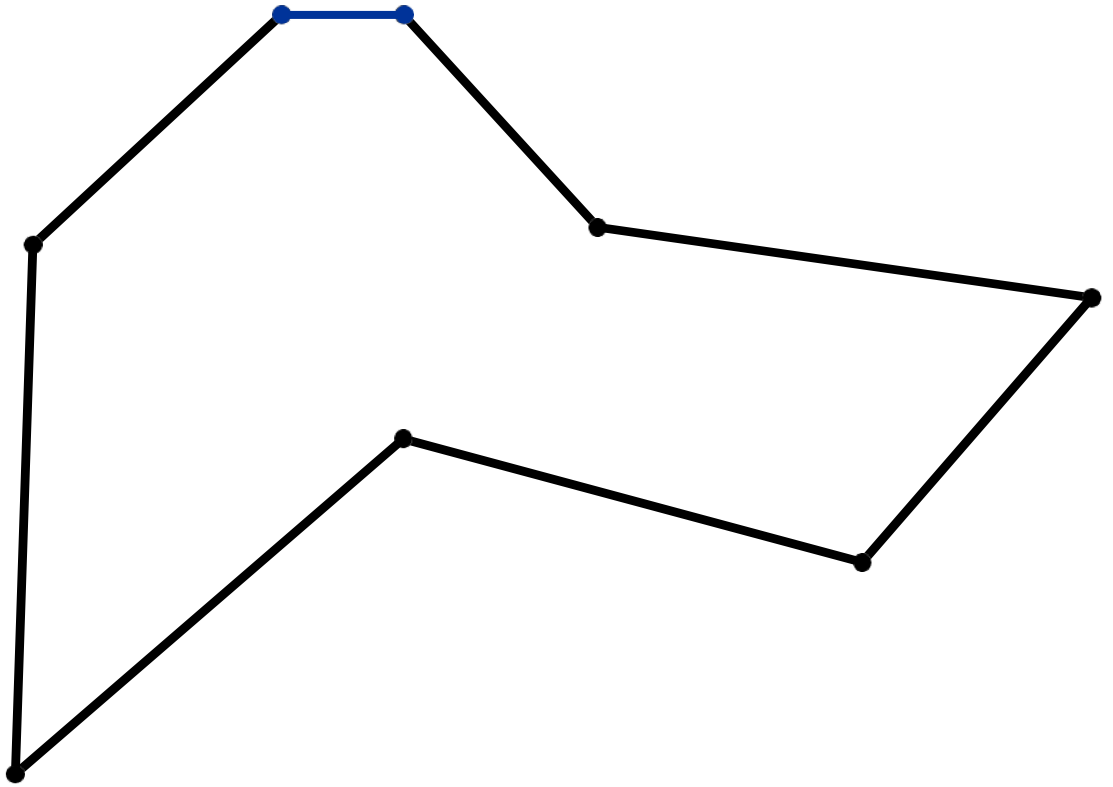


Curves: Corner Cutting

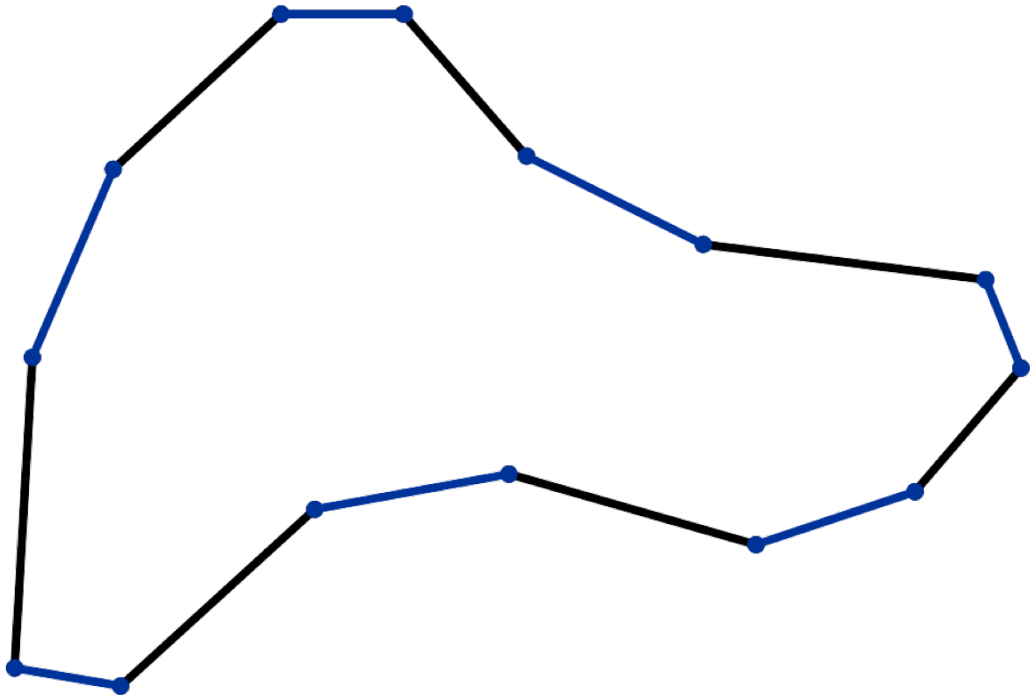


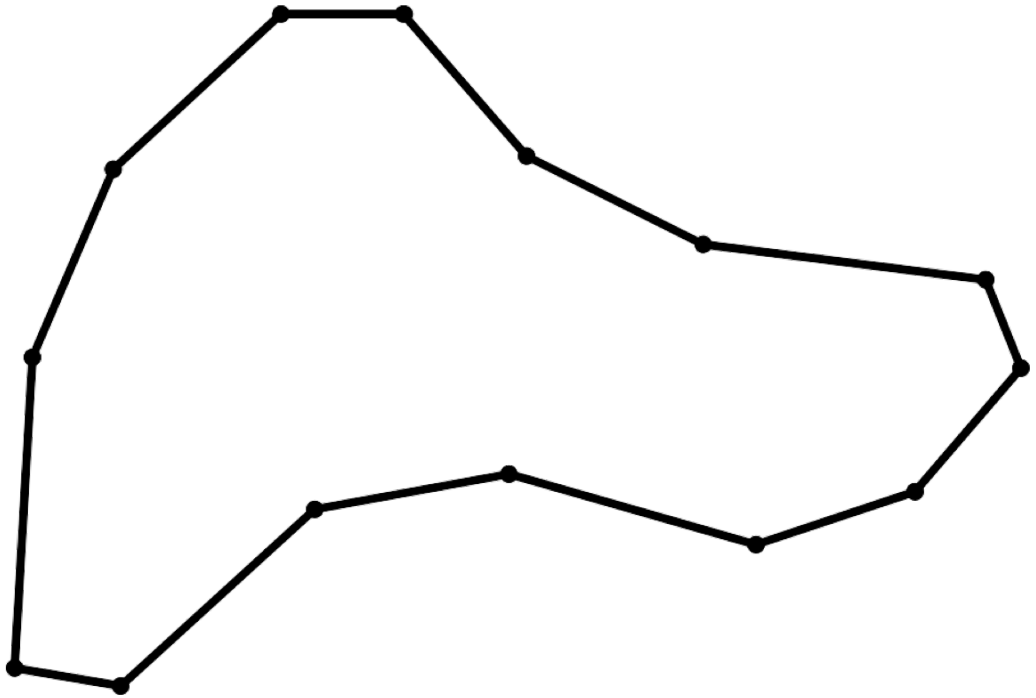
Corner Cutting



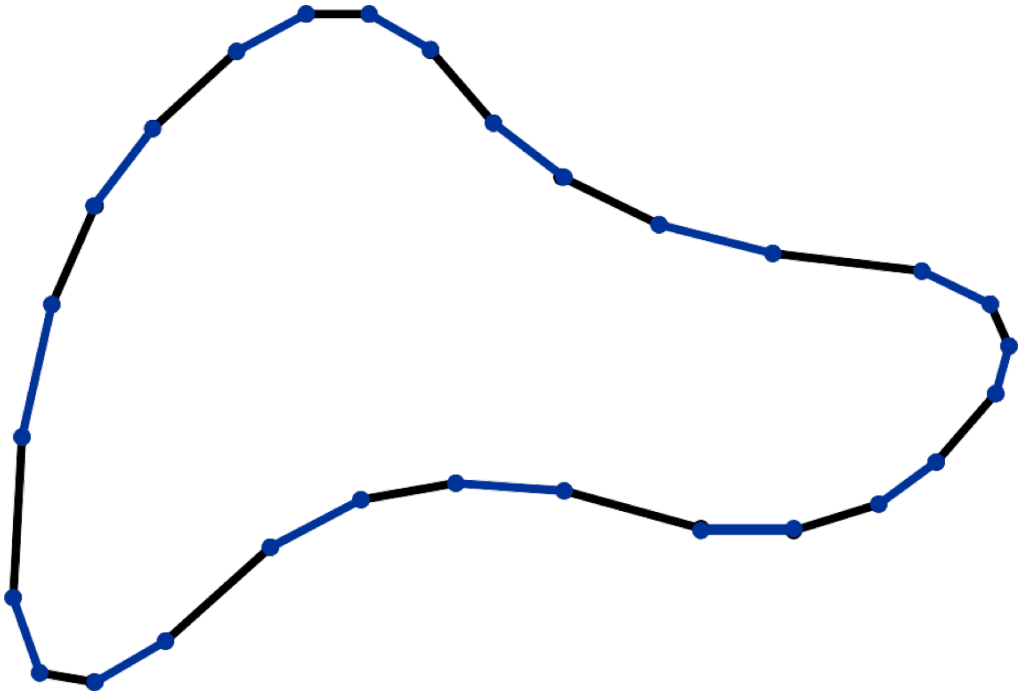


Corner Cutting

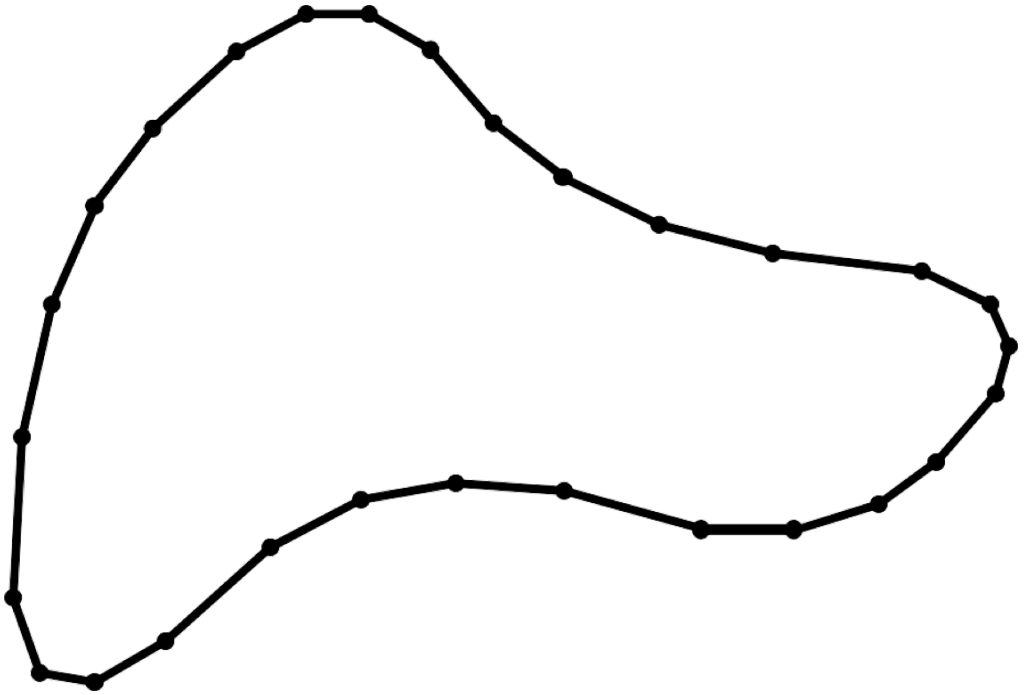


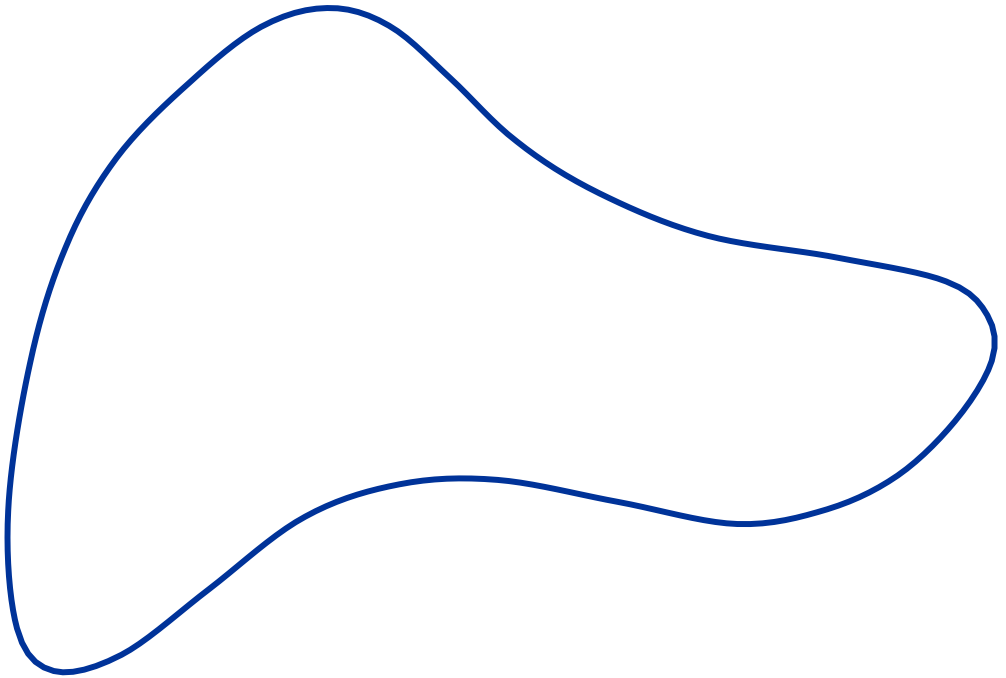


Corner Cutting

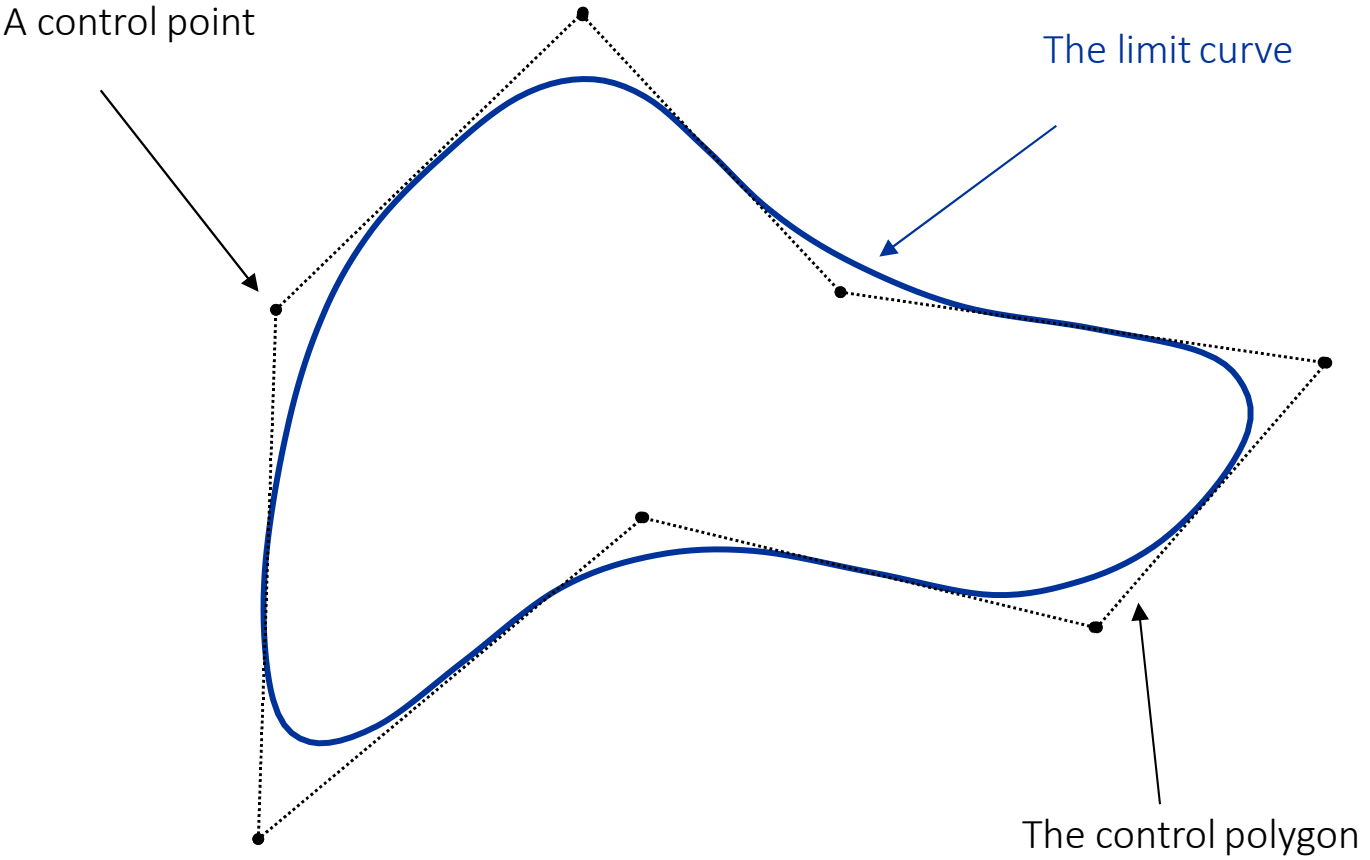


Corner Cutting

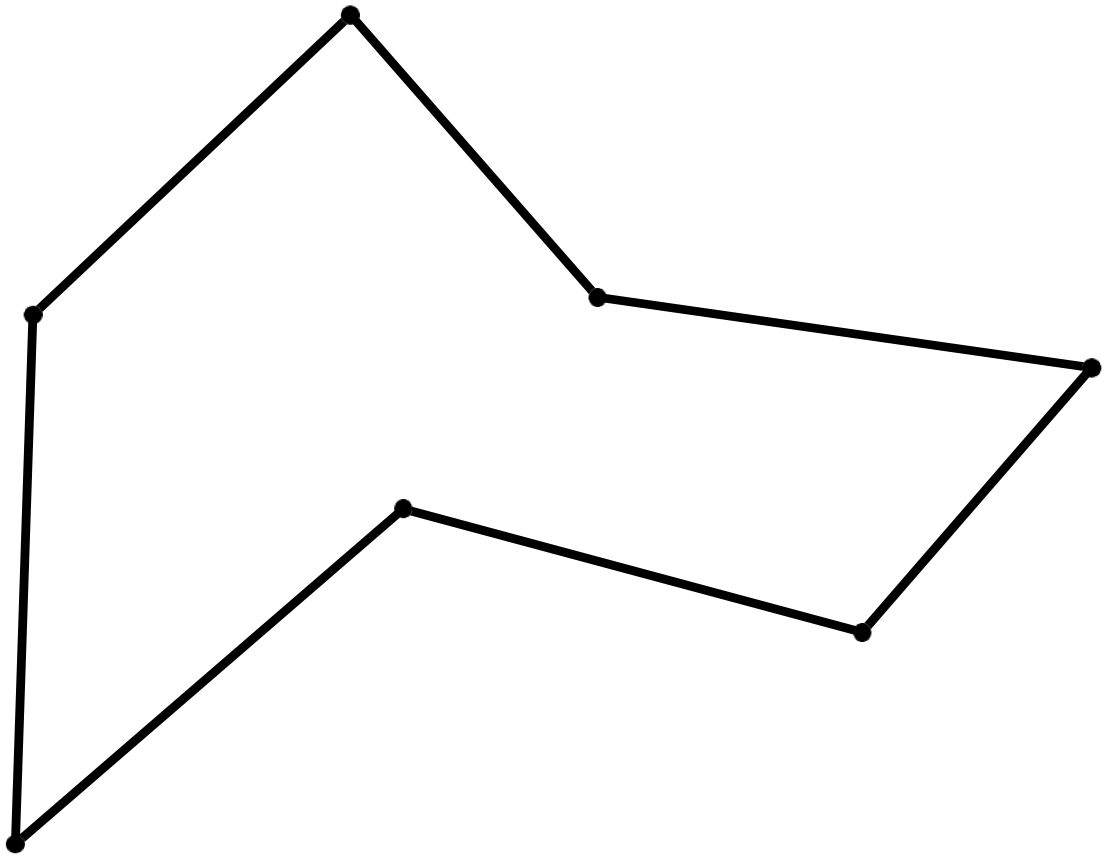




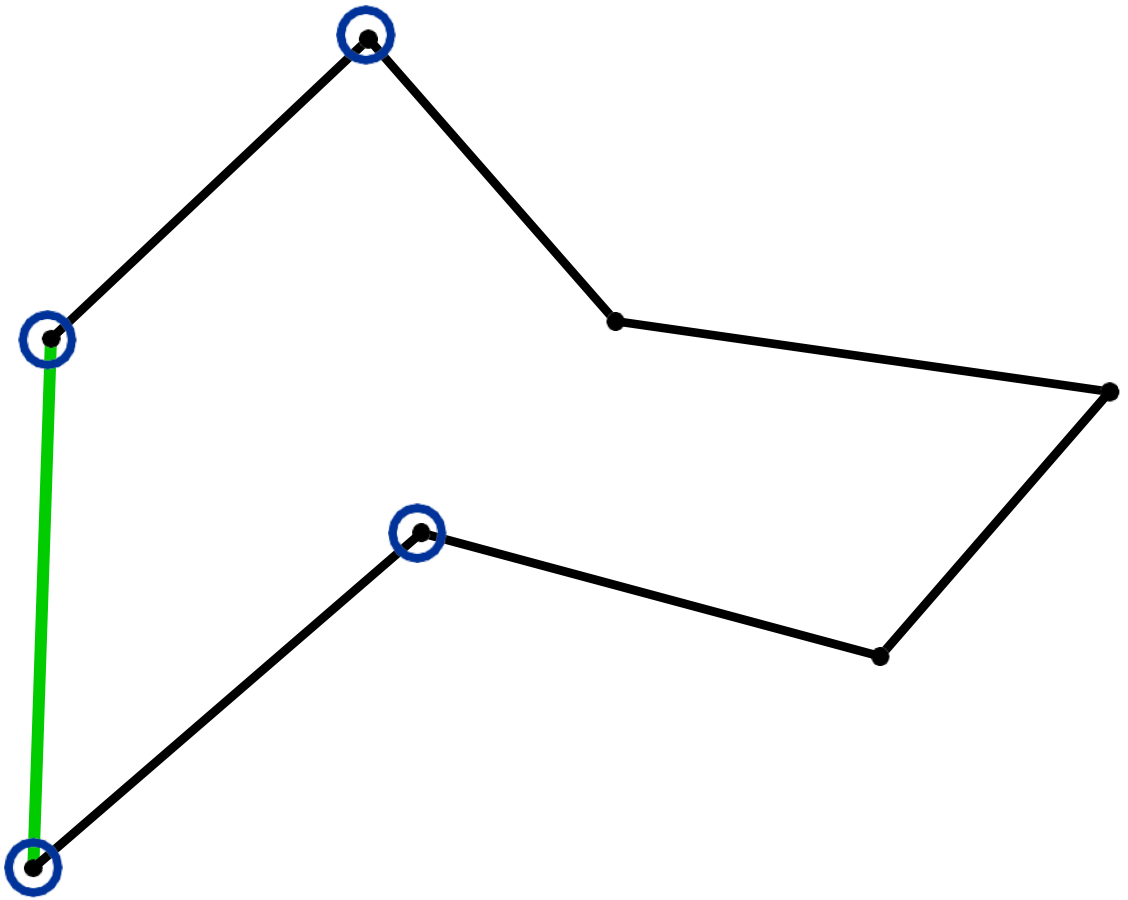
Corner Cutting



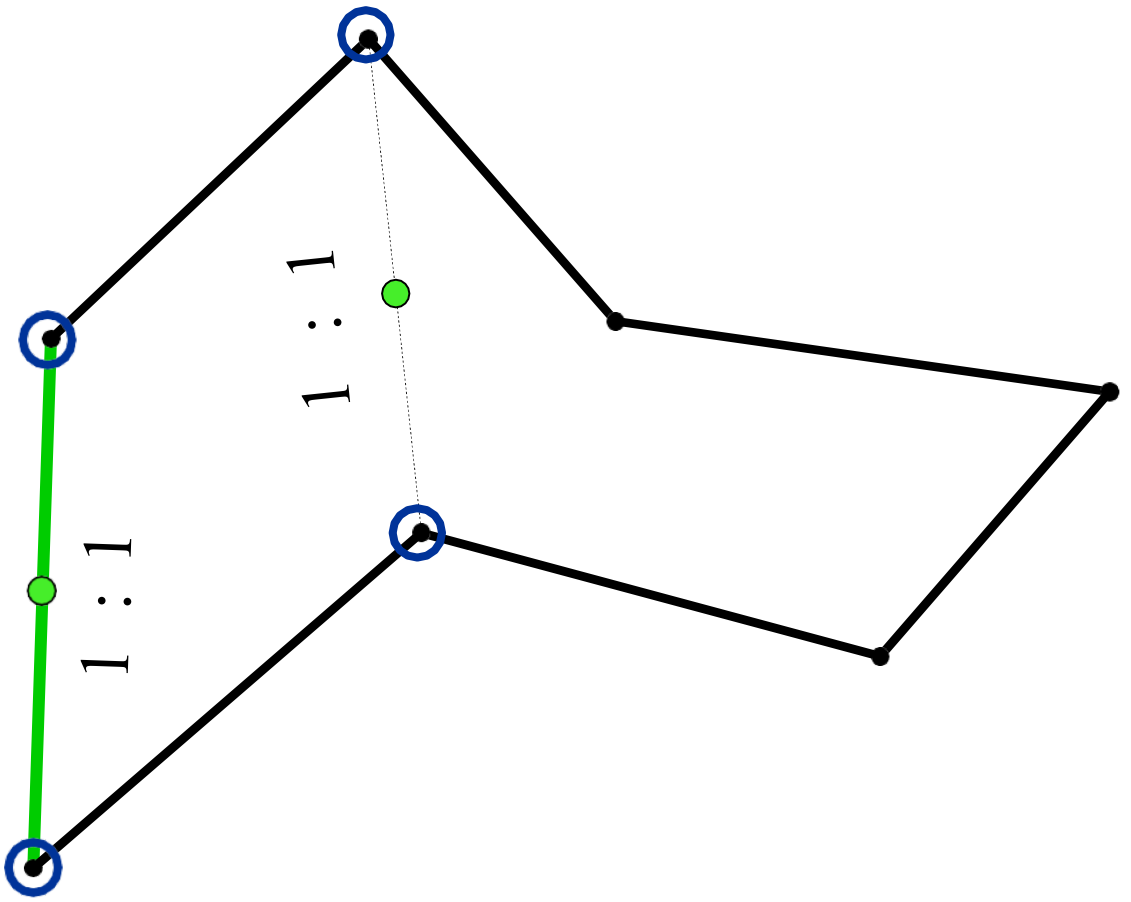
The 4-Point Scheme



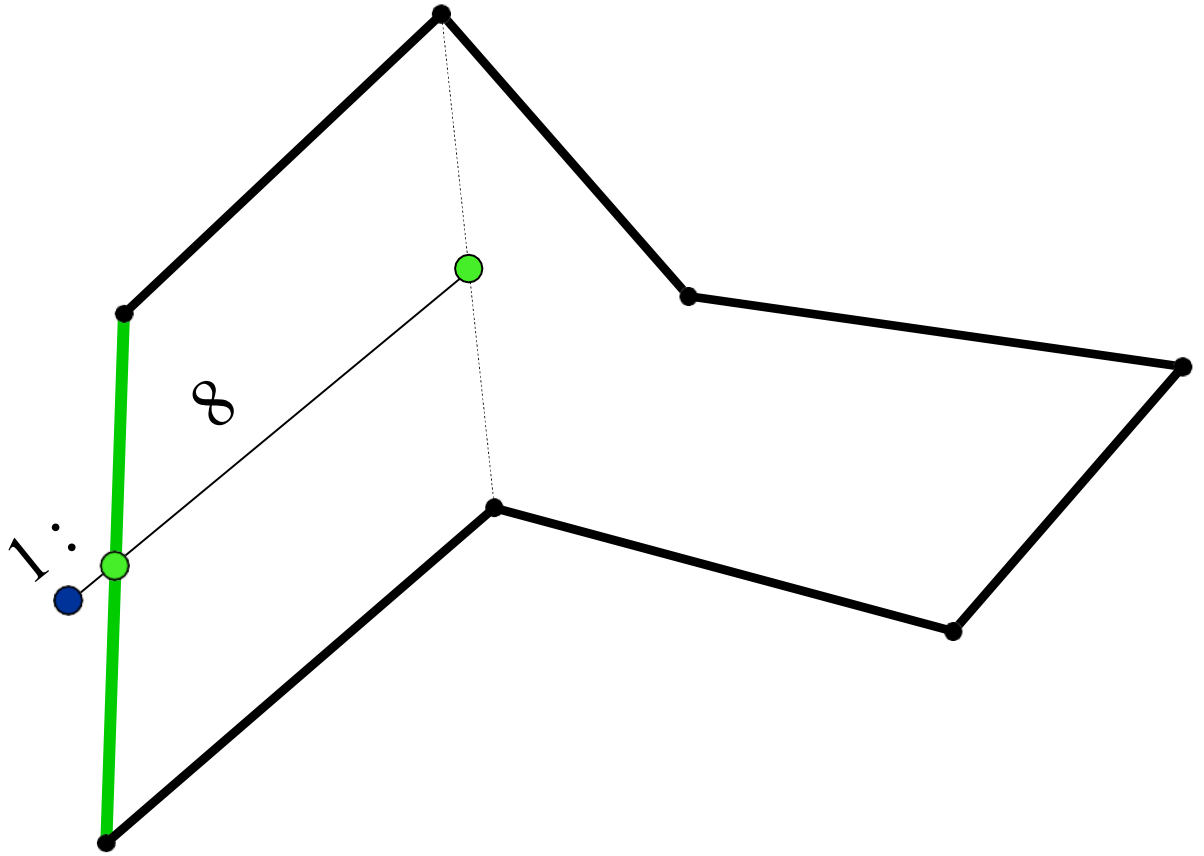
The 4-Point Scheme



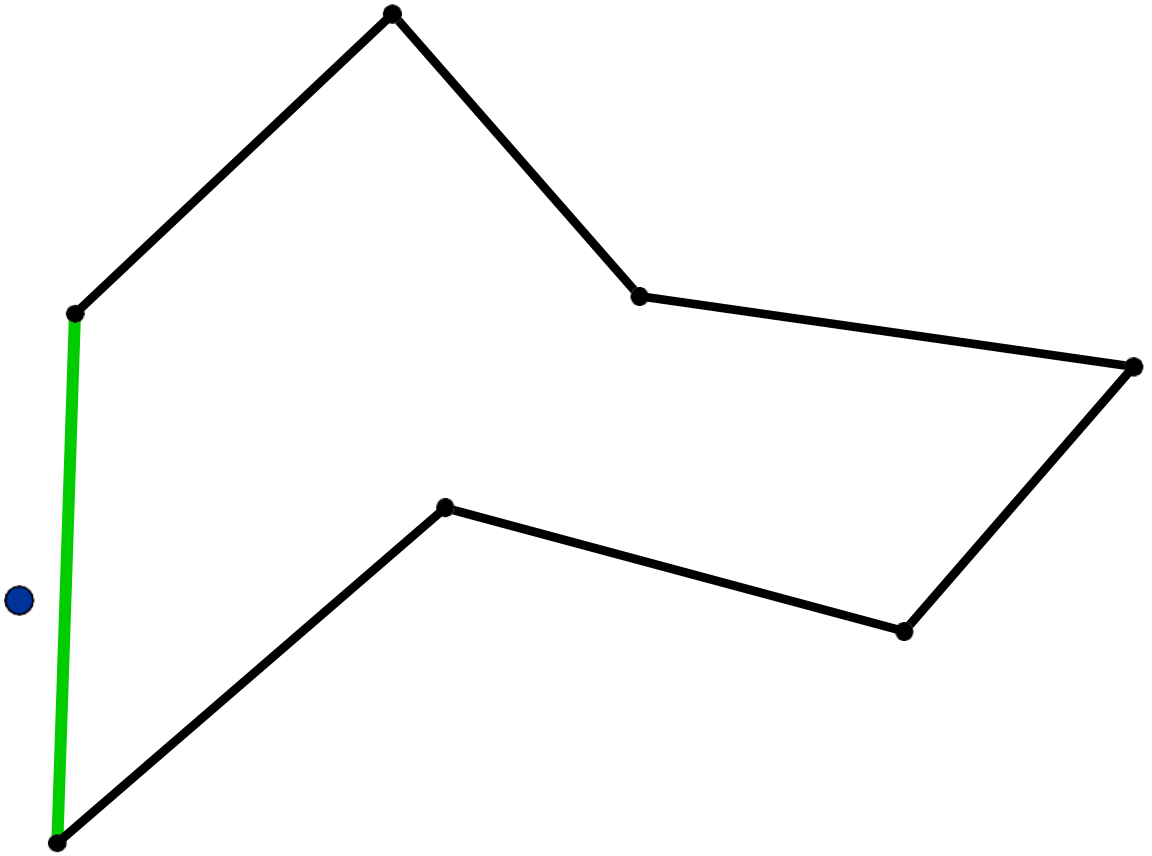
The 4-Point Scheme



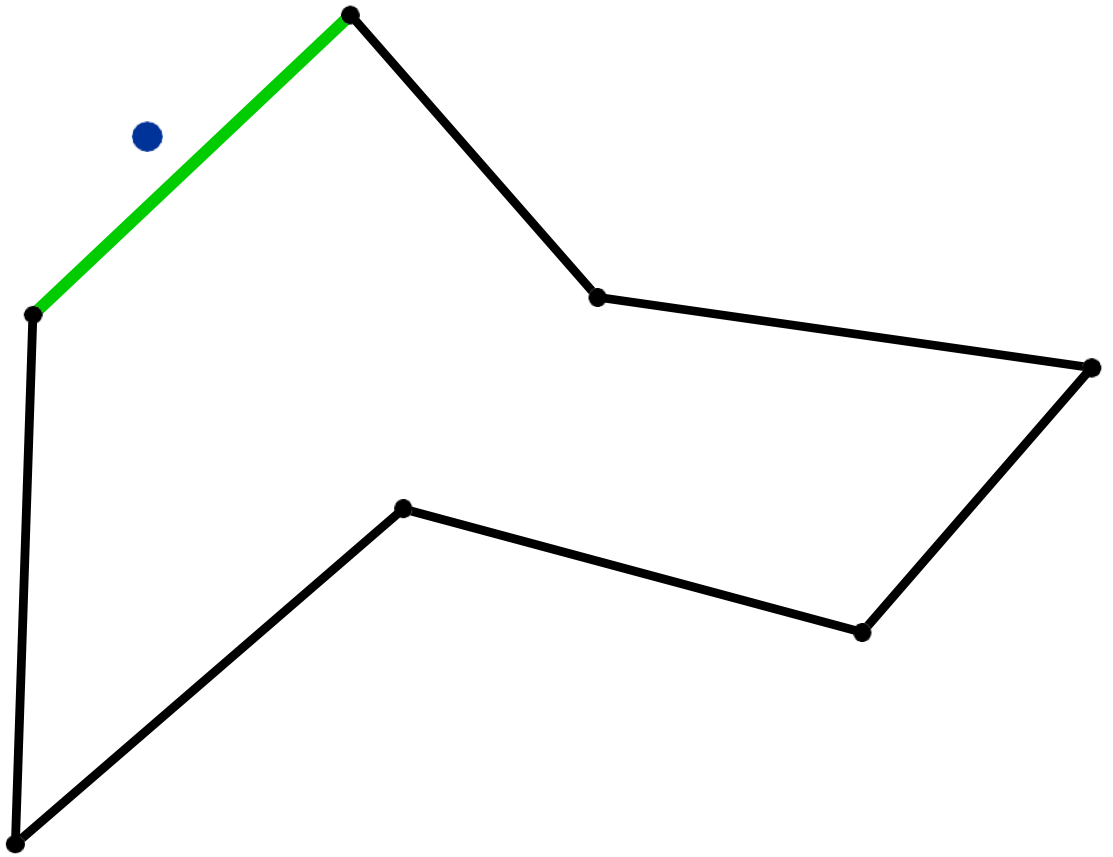
The 4-Point Scheme



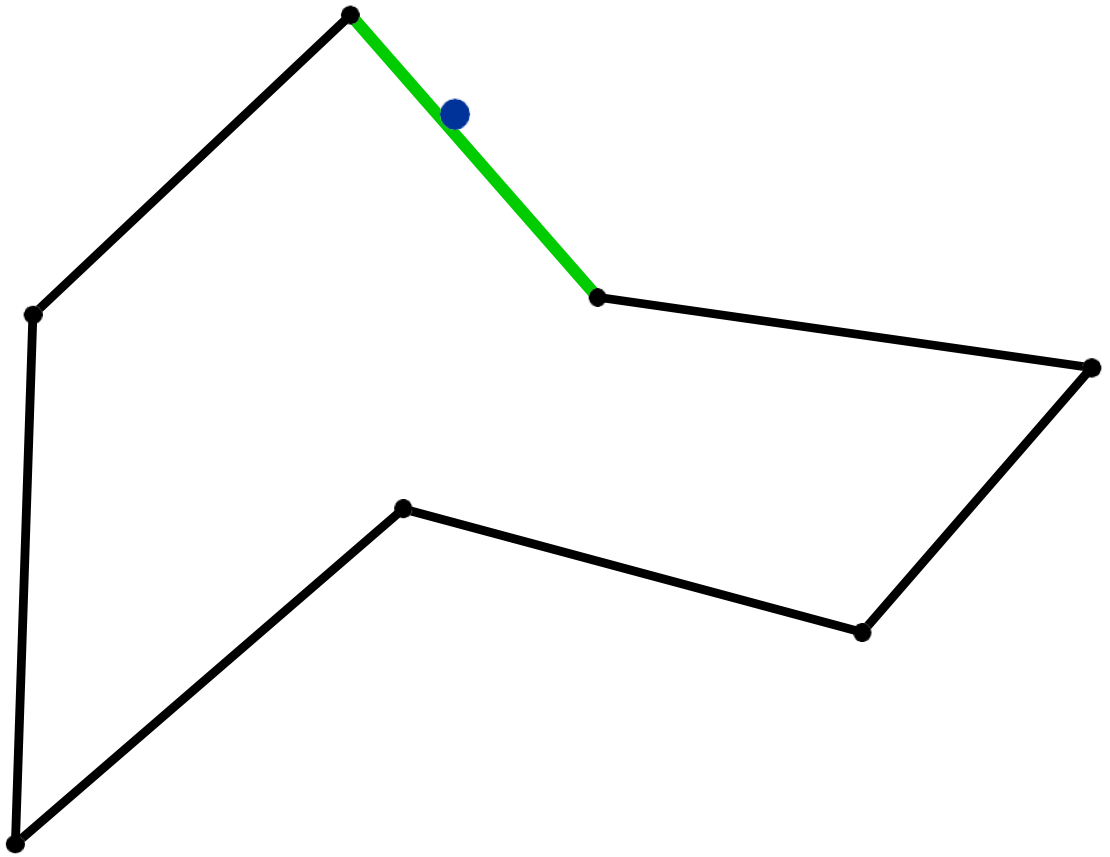
The 4-Point Scheme



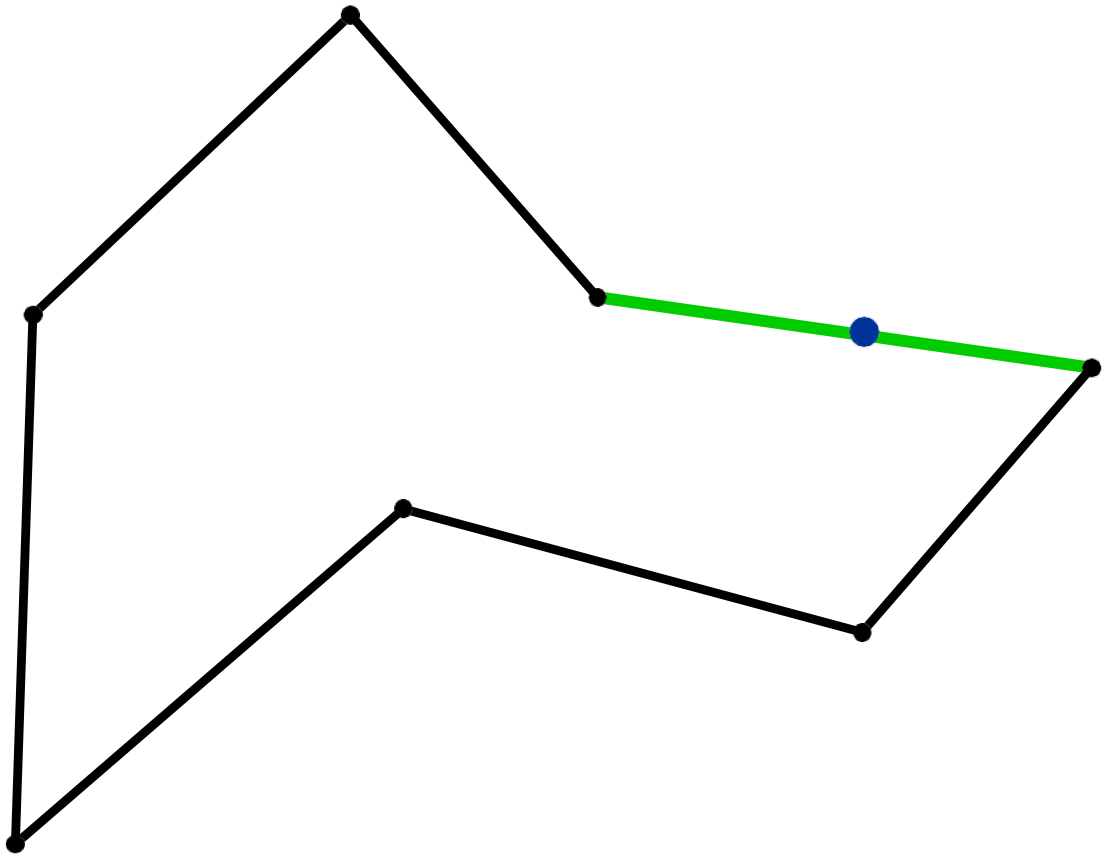
The 4-Point Scheme



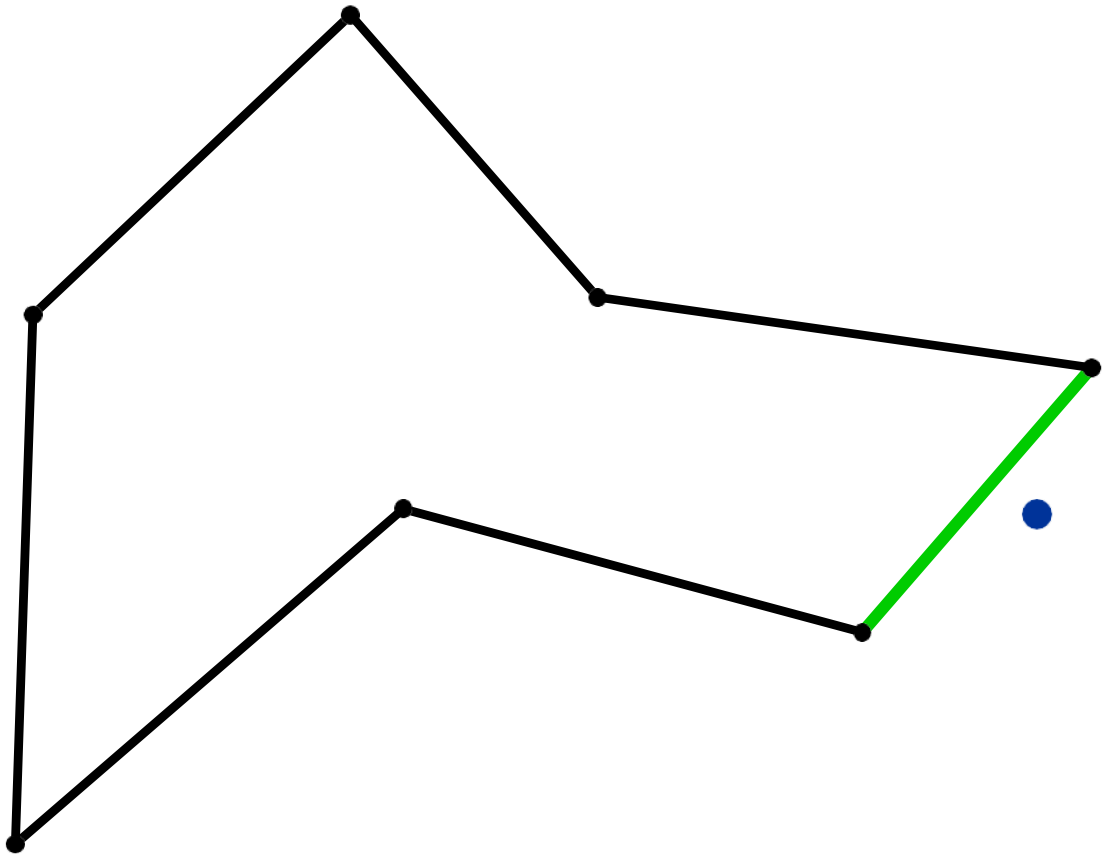
The 4-Point Scheme



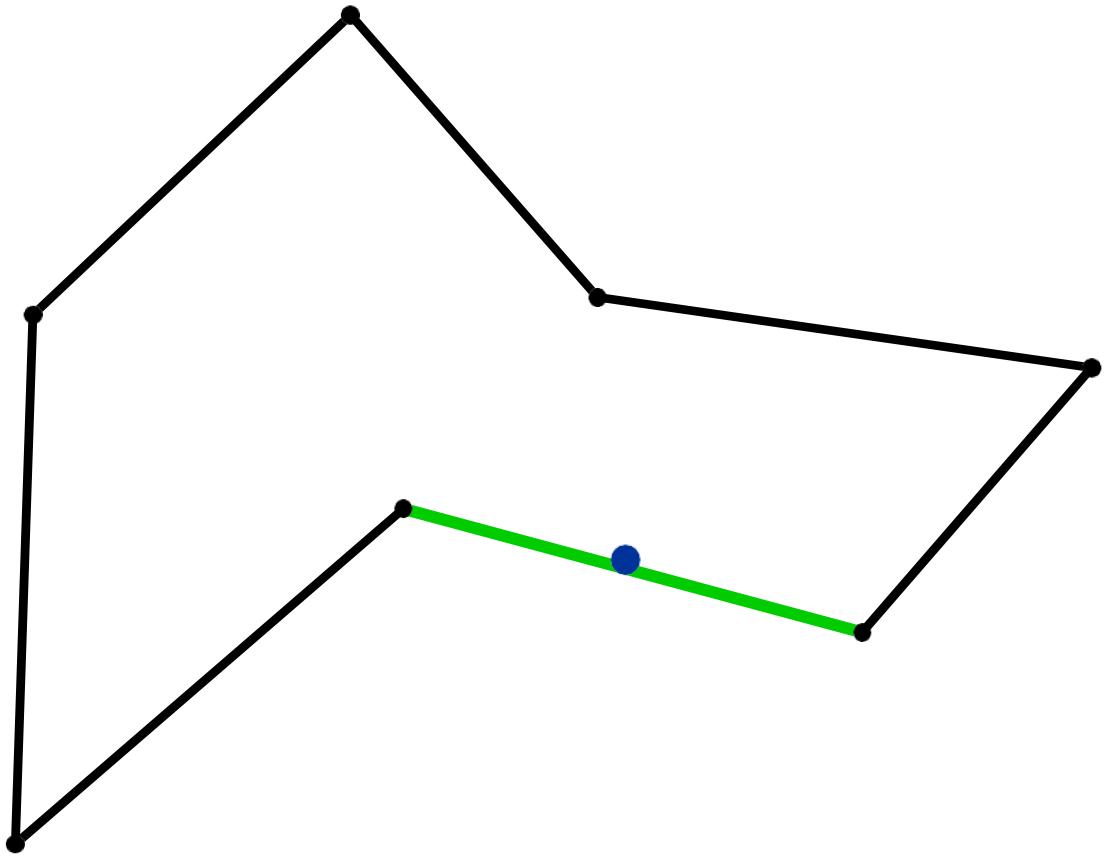
The 4-Point Scheme



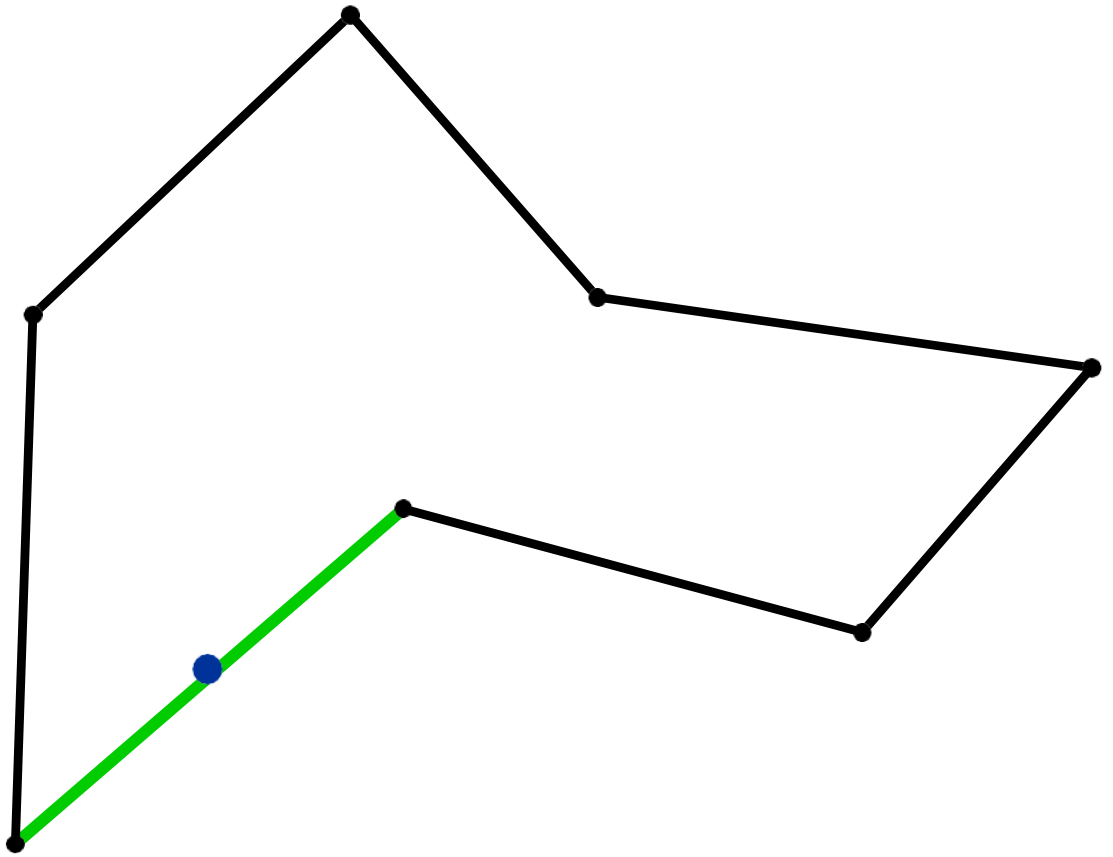
The 4-Point Scheme



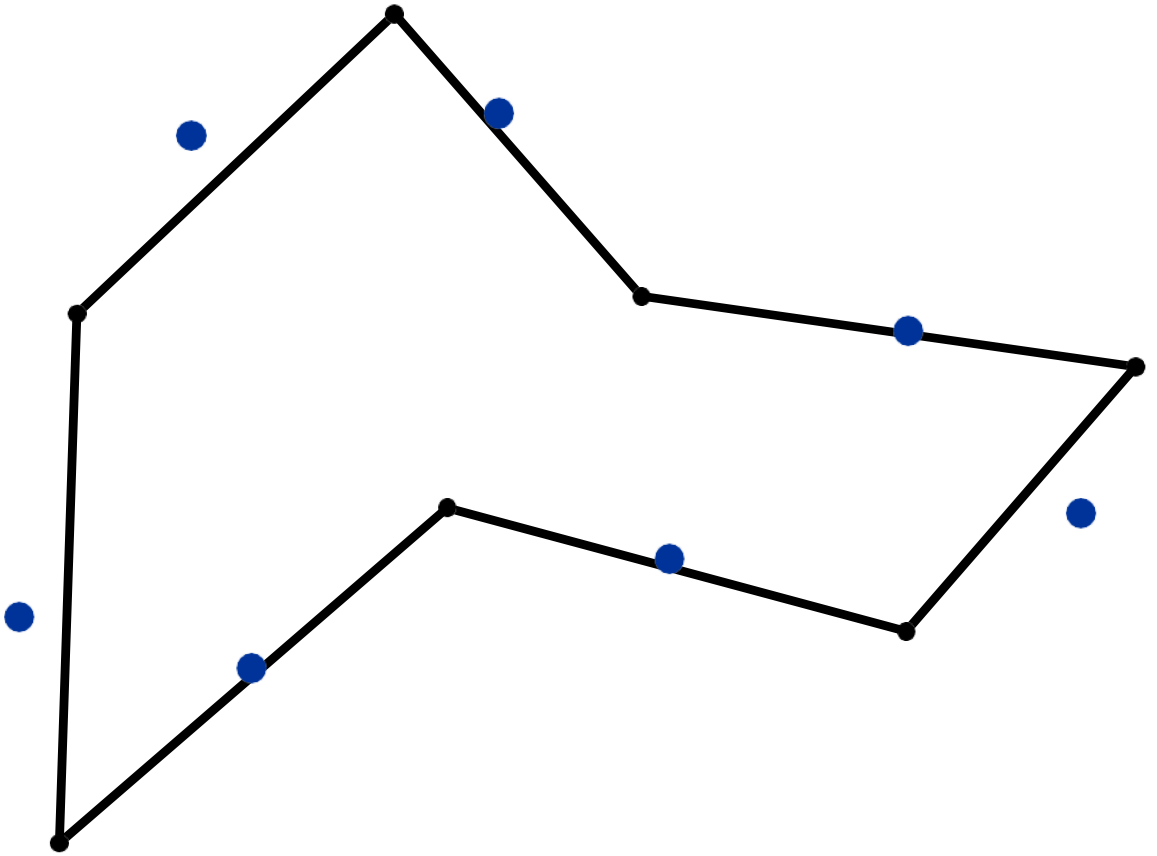
The 4-Point Scheme



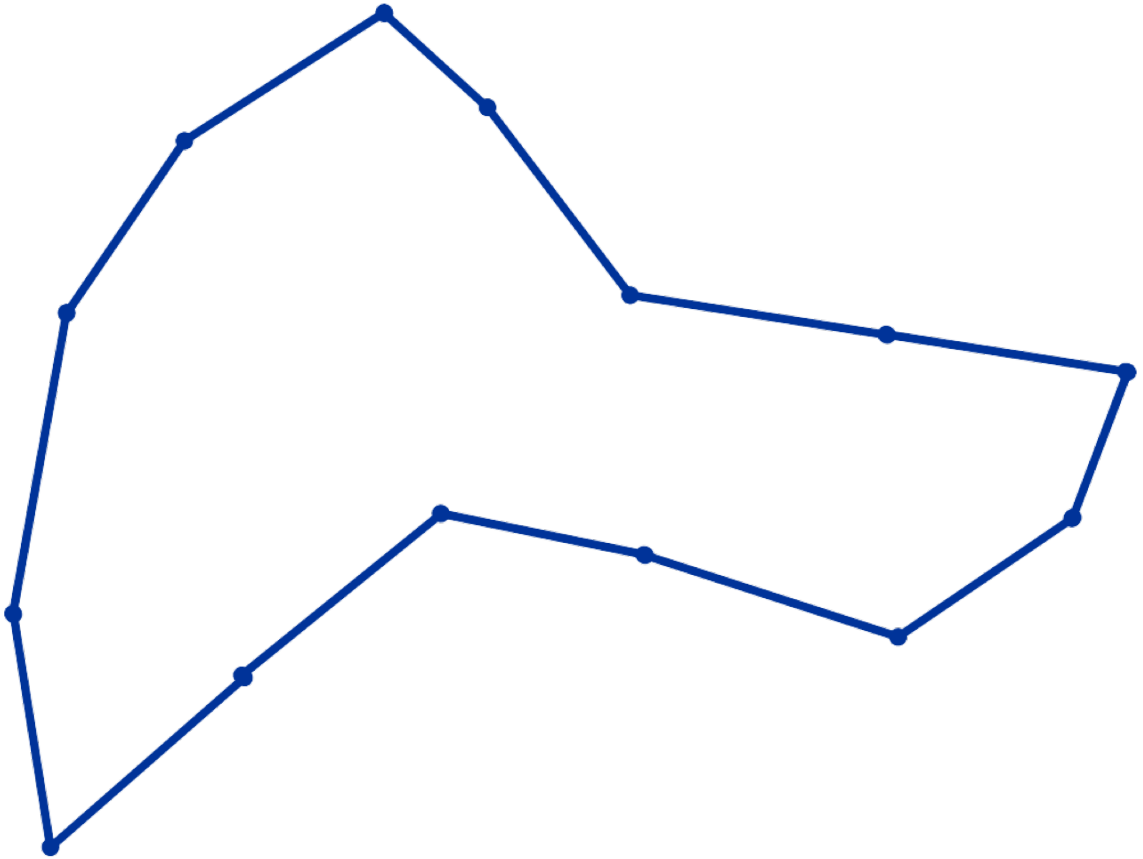
The 4-Point Scheme



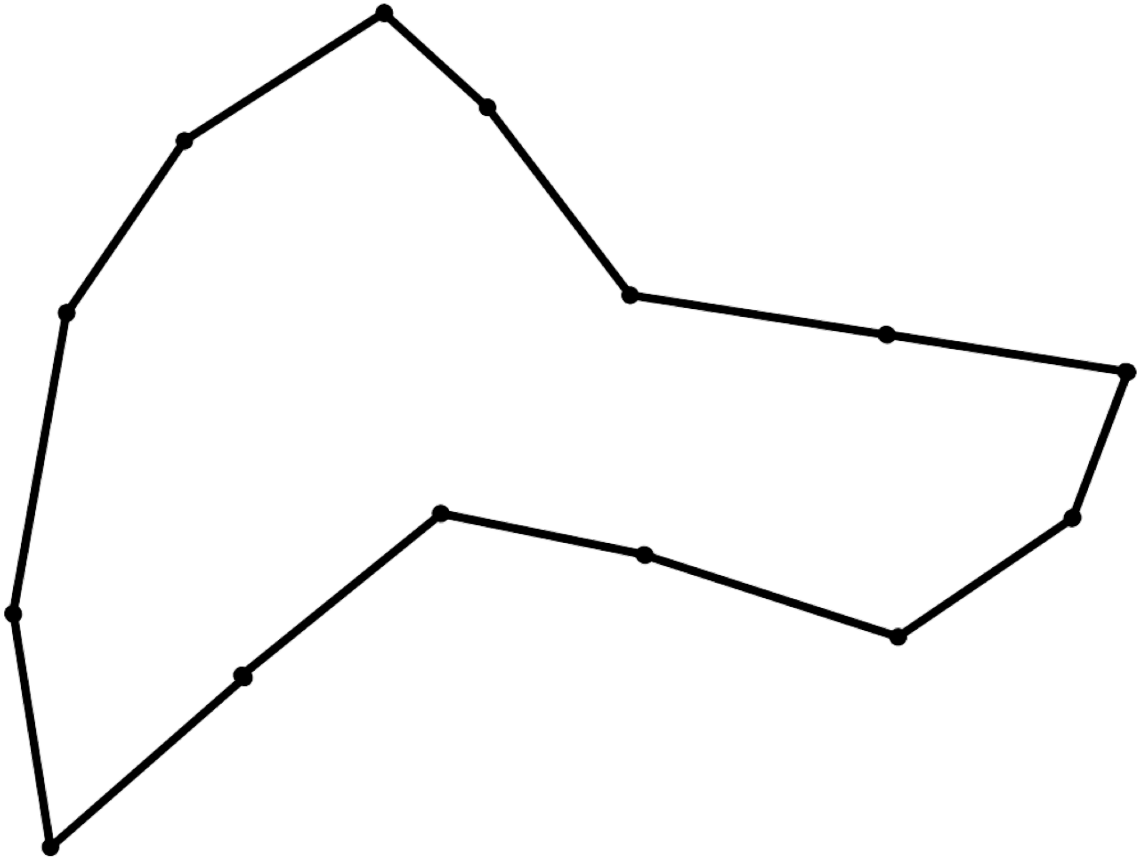
The 4-Point Scheme

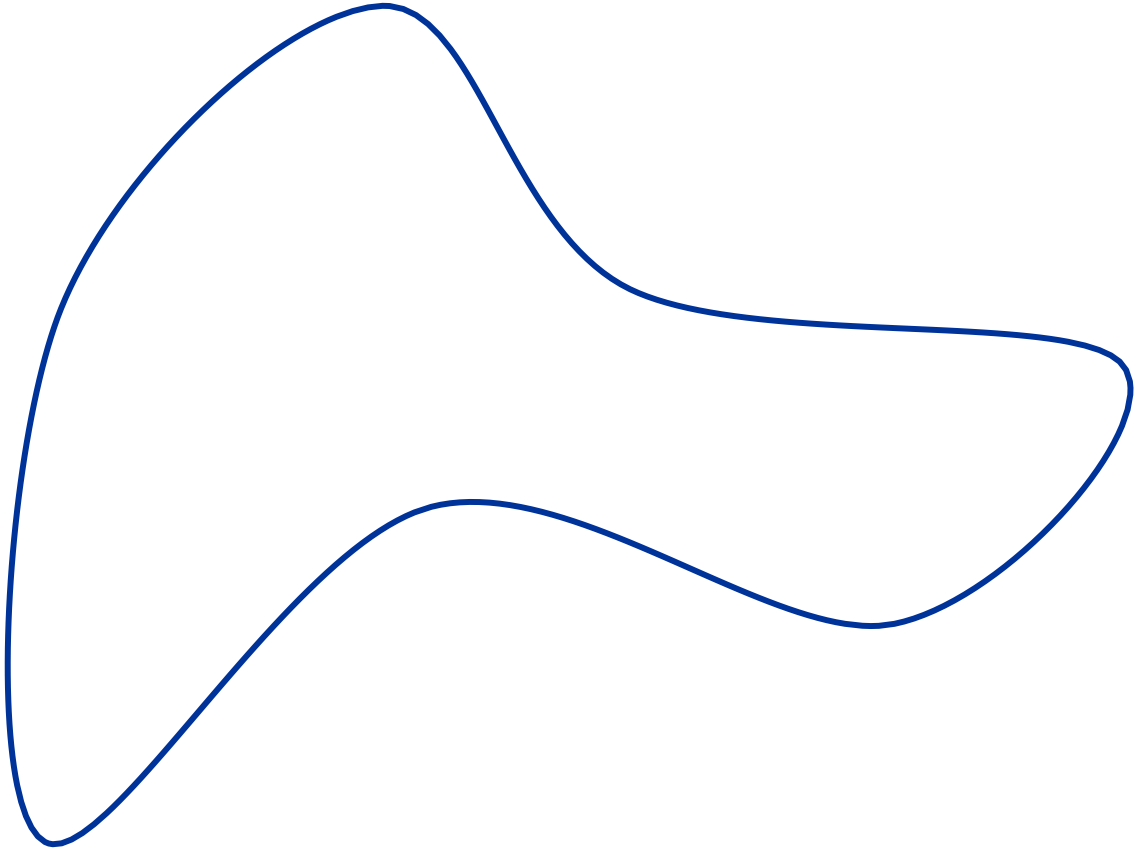


The 4-Point Scheme

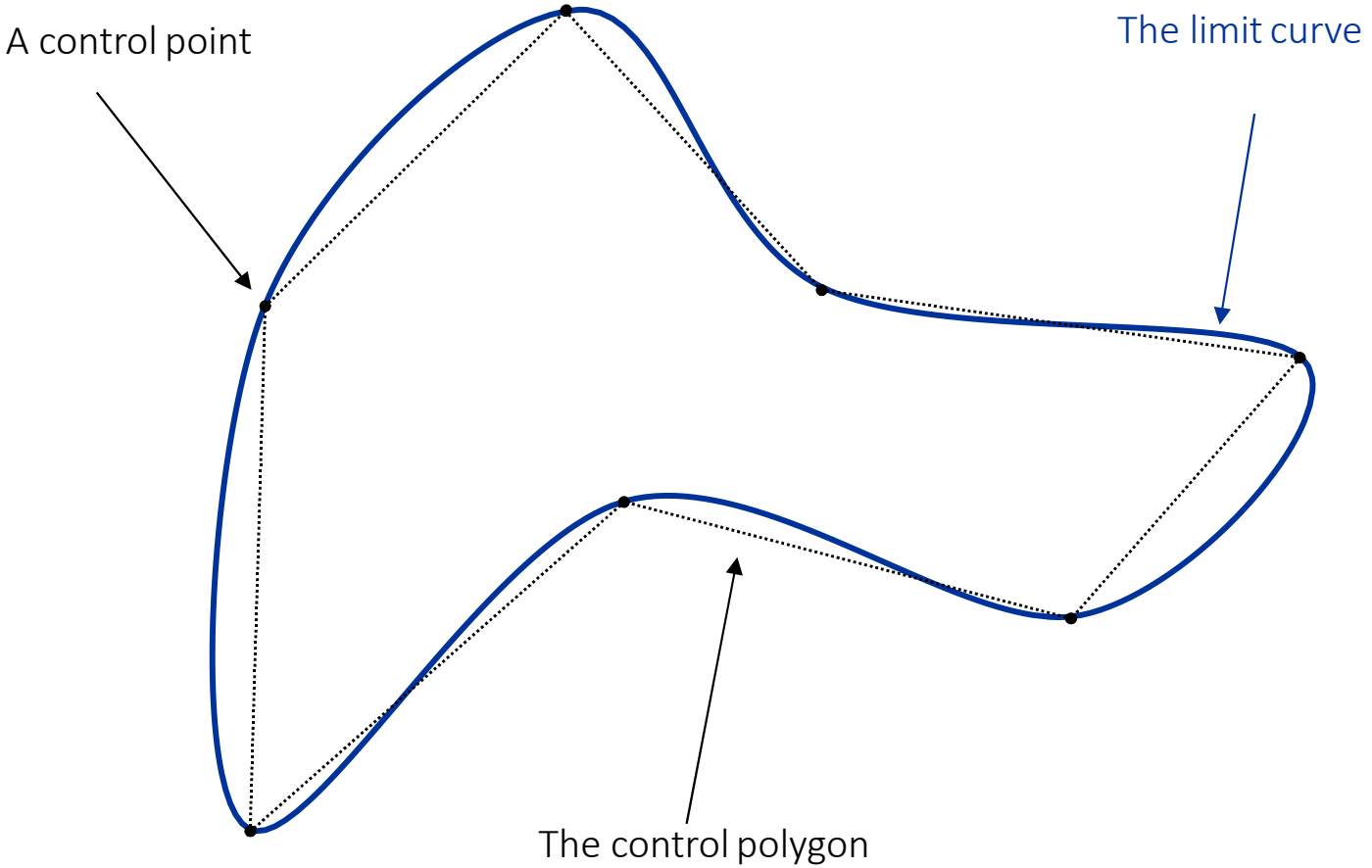


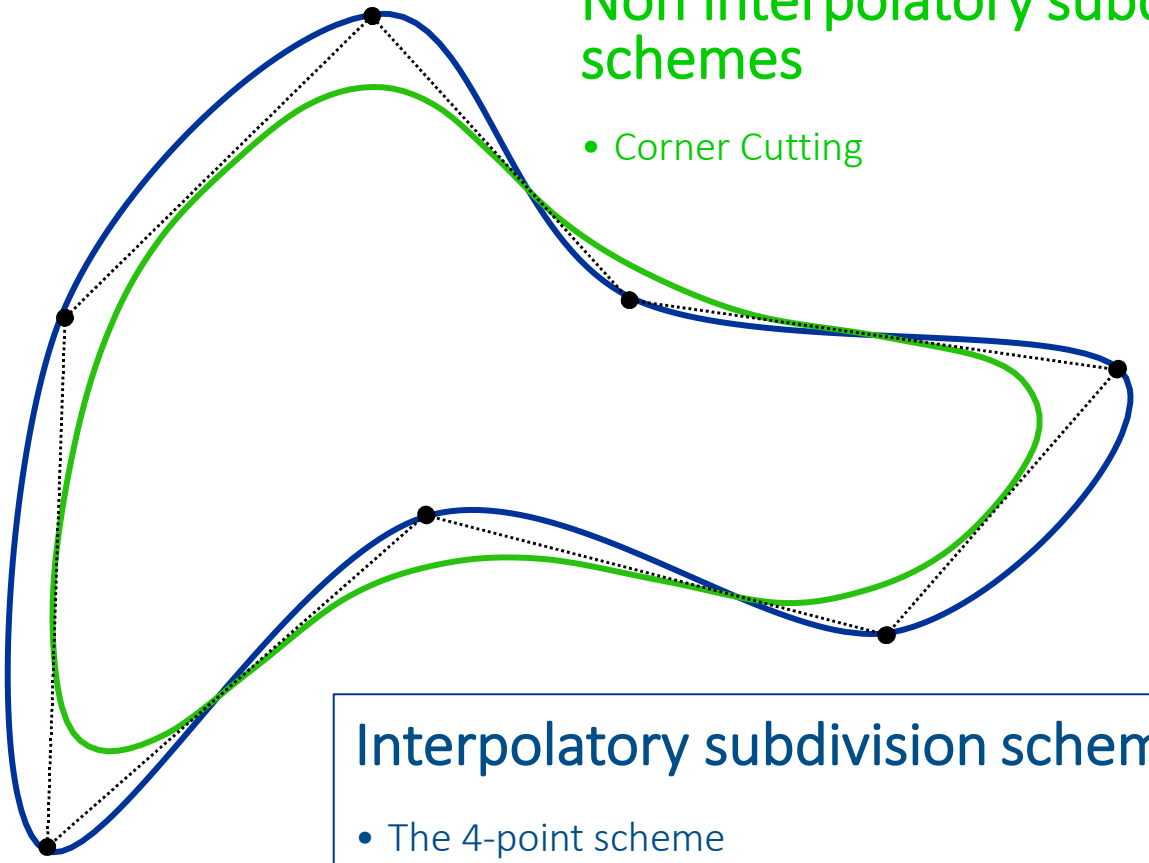
The 4-Point Scheme





The 4-Point Scheme





Non interpolatory subdivision schemes

- Corner Cutting

Interpolatory subdivision schemes

- The 4-point scheme

Definition

- A subdivision curve is generated by repeatedly applying a subdivision operator to a given polygon (called the control polygon)

The central theoretical questions:

- **Convergence:** Given a subdivision operator and a control polygon, does the subdivision process converge?
- **Smoothness:** Does the subdivision process converge to a smooth curve?

A *control net* consists of vertices, edges, and faces.

Refinement

- In each iteration, the subdivision operator refines the control net, increasing the number of vertices (approximately) by a factor of 4.

Limit Surface

- In the limit the vertices of the control net converge to a limit surface.

Topology and Geometry

- Every subdivision method has a method to generate the topology of the refined net, and rules to calculate the location of the new vertices.

Works on triangular meshes

Is an Approximating Scheme

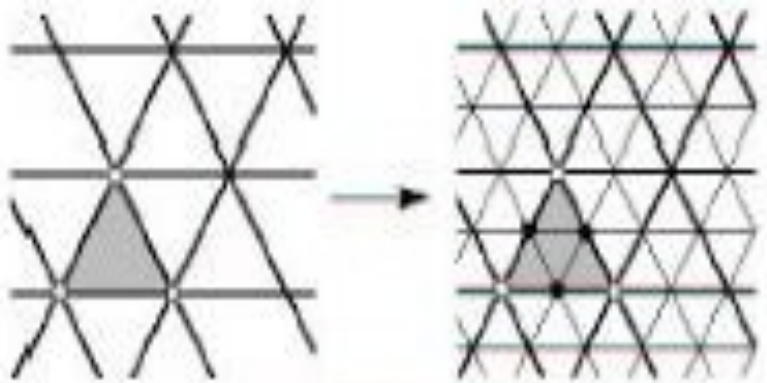
Guaranteed to be smooth everywhere except at *extraordinary* vertices.

There are different subdivision schemes

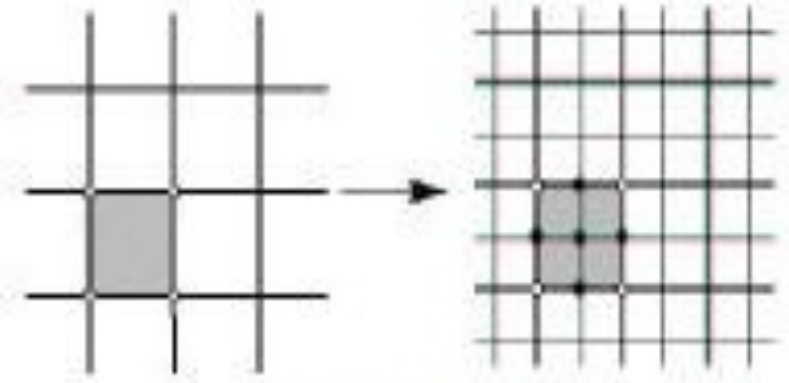
- Different methods for refining topology

Different rules for positioning vertices

- Interpolating versus approximating

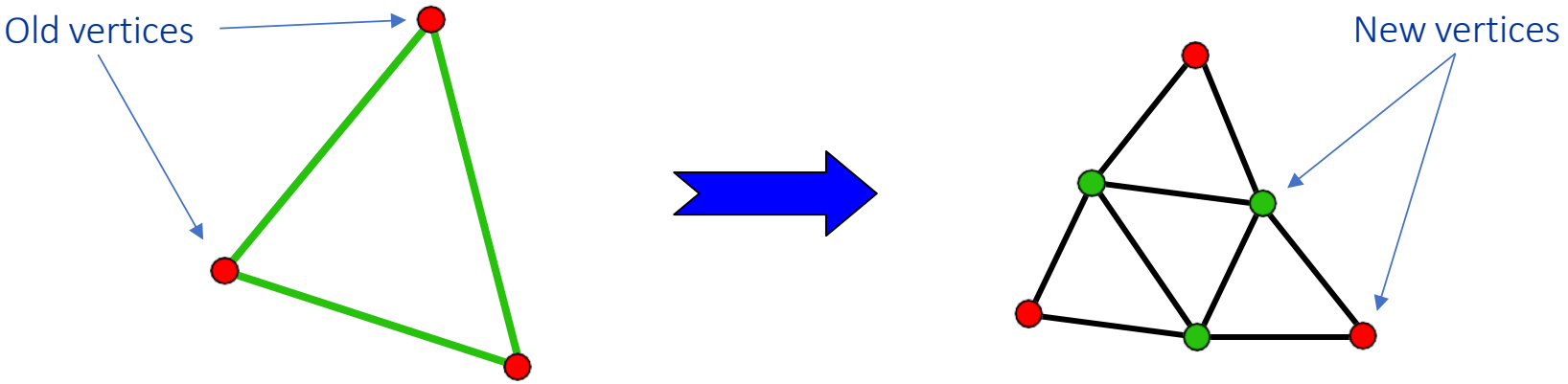


Face split for triangles



Face split for quads

For control nets whose faces are triangular



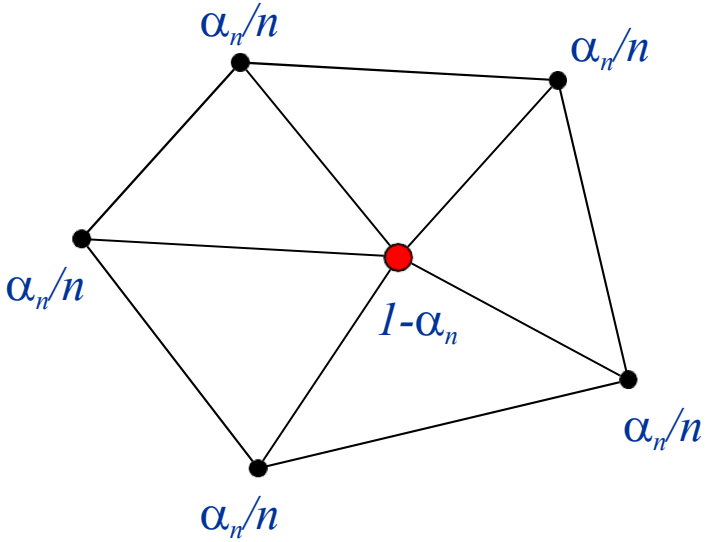
Every face is replaced by 4 new triangular faces. The are two kinds of new vertices:

- Green vertices are associated with old edges
- Red vertices are associated with old vertices.

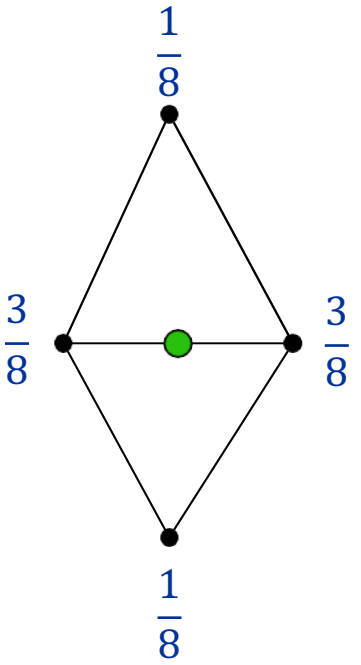
Location of New Vertices

- Every new vertex is a weighted average of the old vertices. The list of weights is called the subdivision mask or the stencil

A rule for new red vertices



A rule for new green vertices



$$\alpha_n = \frac{1}{64}\left(40 - \left(3 + 2\cos\frac{2\pi}{n}\right)^2\right)$$

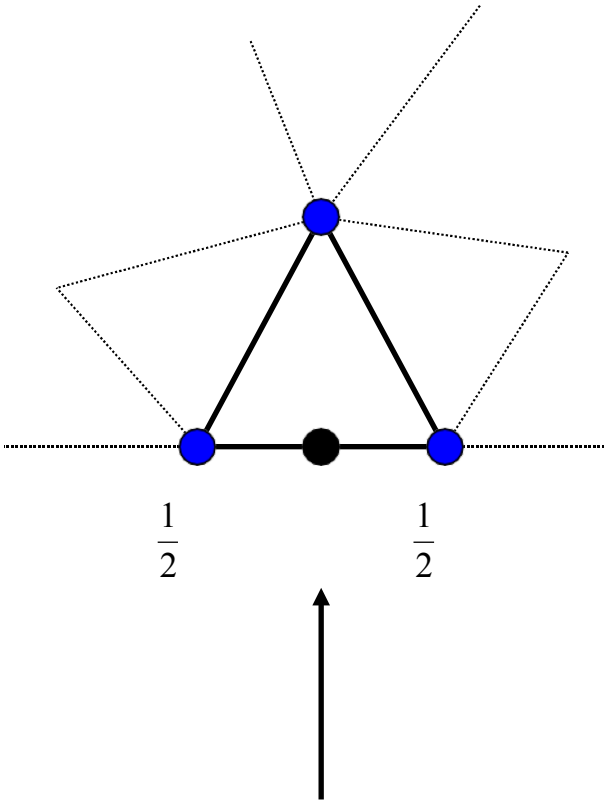
Original

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

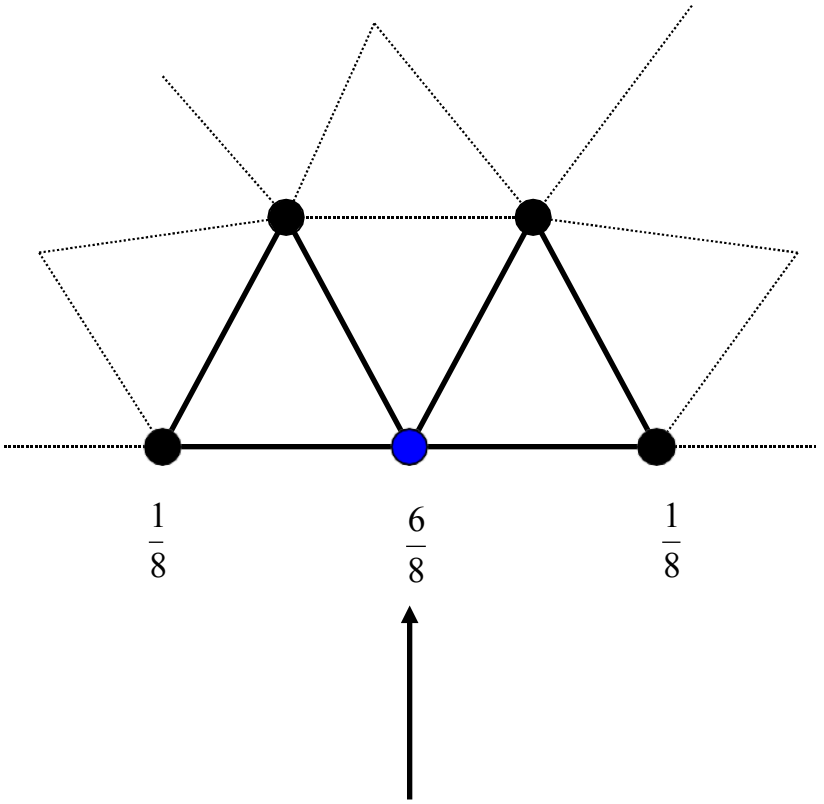
Warren

n - the vertex valence

Subdivision Mask for Boundary Conditions



Edge Rule



Vertex Rule

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

- S_{mask} is very sparse
- Never Implement this way!
- Allows for analysis
  - Curvature
  - Limit Surface

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

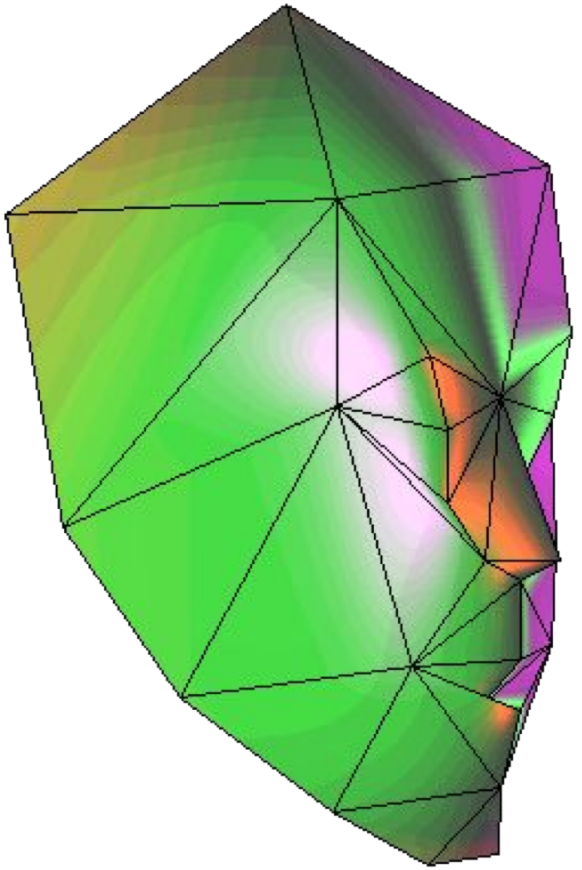
$$\begin{bmatrix} w_{00} & w_{01} & \cdots & 0 \\ w_{10} & w_{11} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & w_{nj} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} \hat{p}_0 \\ \hat{p}_1 \\ \hat{p}_2 \\ \vdots \\ \hat{p}_j \end{bmatrix}$$

S_{mask} Weights

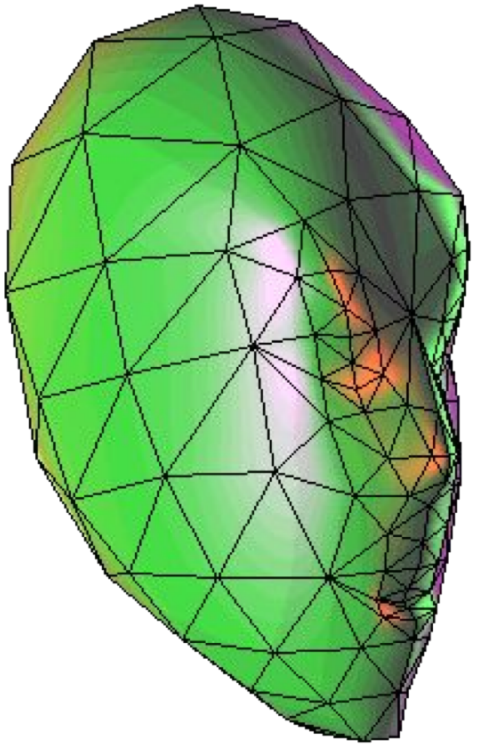
Old Control Points

New Points

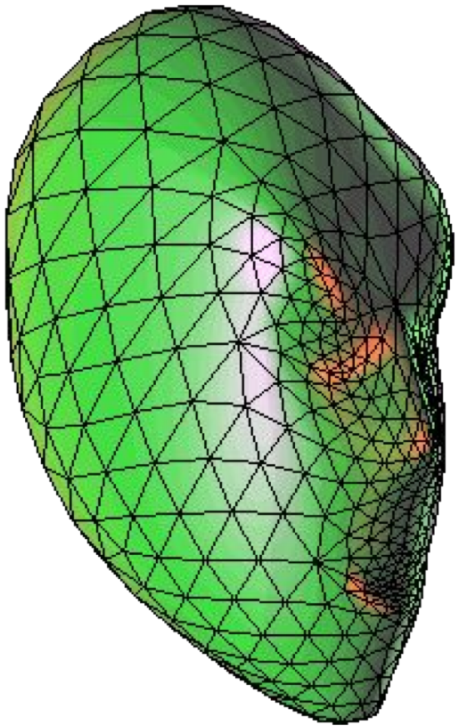
The Original Control Net



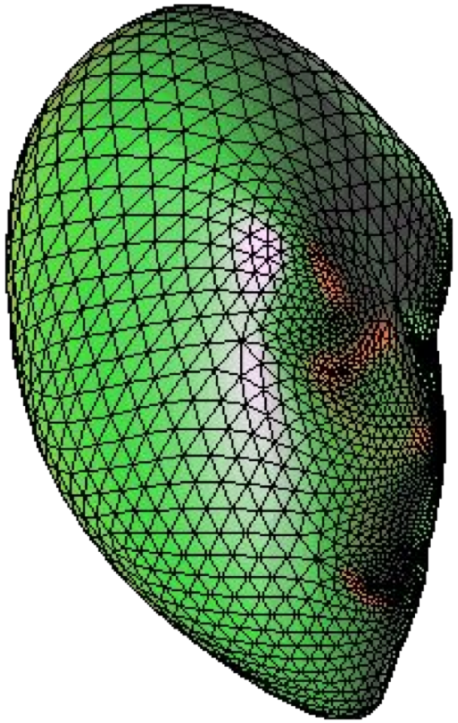
After 1st Iteration

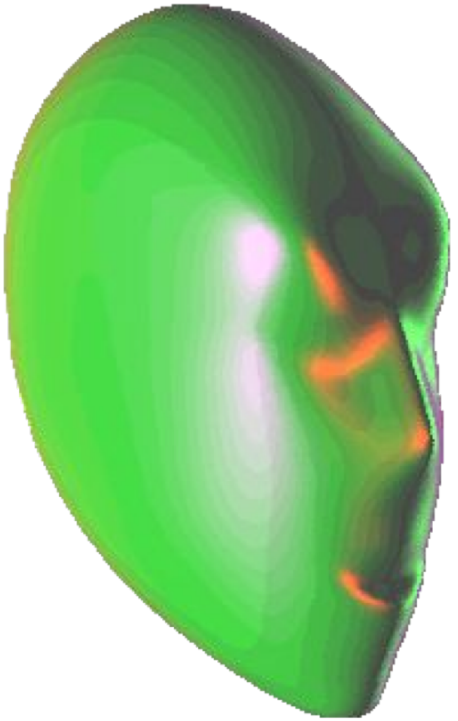


After 2nd Iteration



After 3rd Iteration

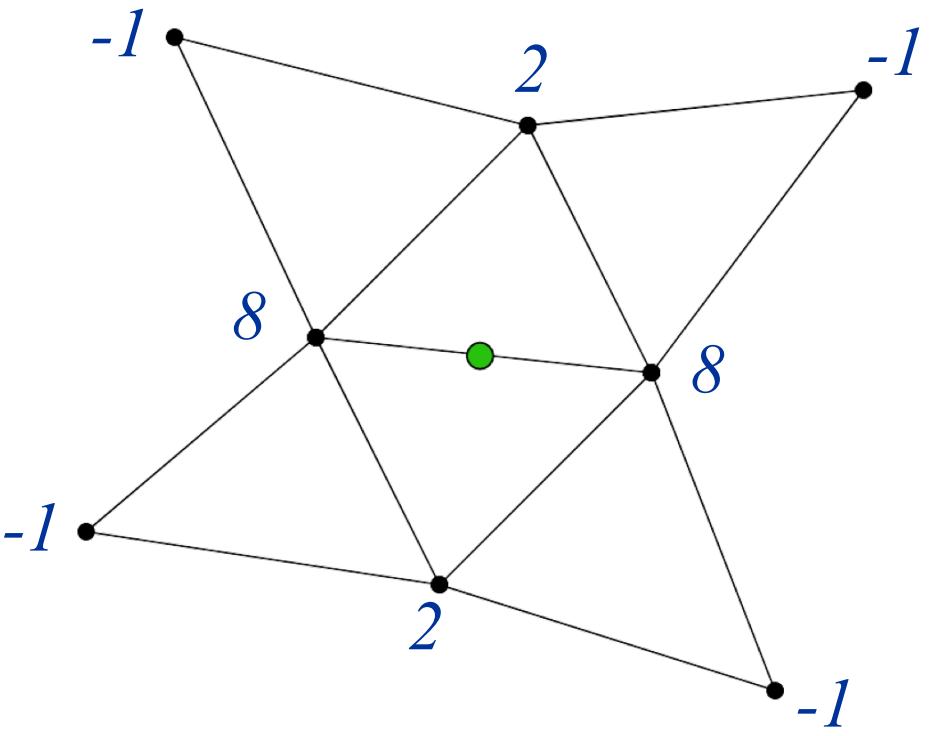




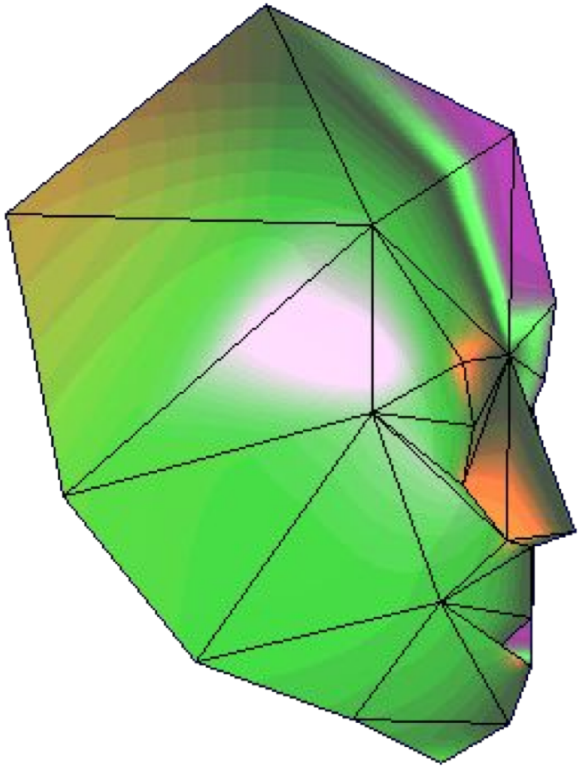
The limit surfaces of Loop’s subdivision have continuous curvature almost everywhere

Butterfly Scheme

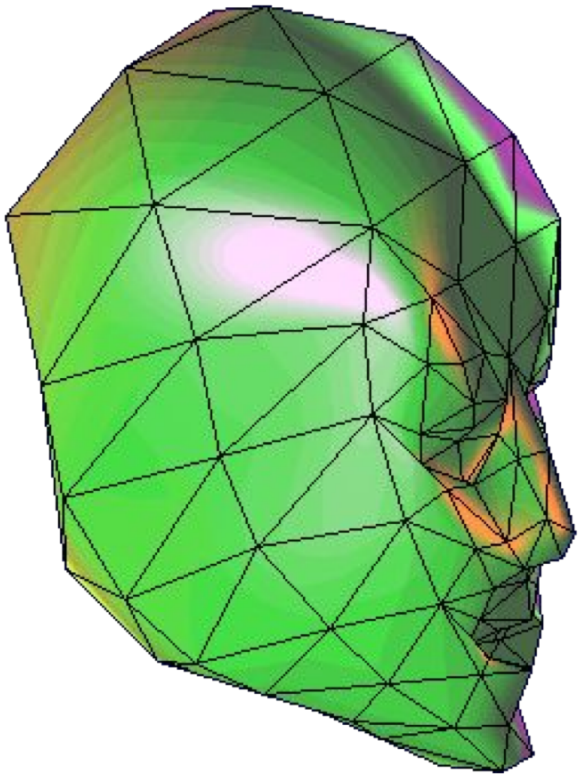
- This is an interpolatory scheme
- The new red vertices inherit the location of the old vertices
- The new green vertices are calculated by the following stencil



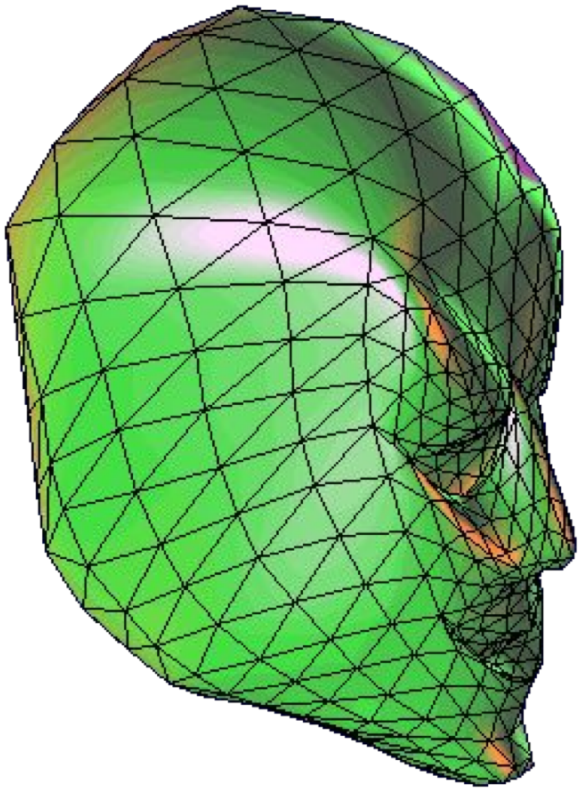
The Original Control Net



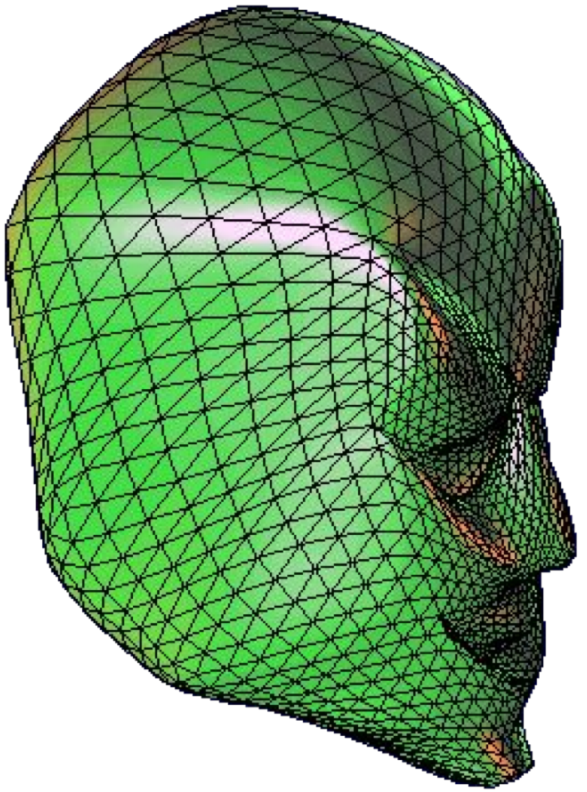
After 1st Iteration

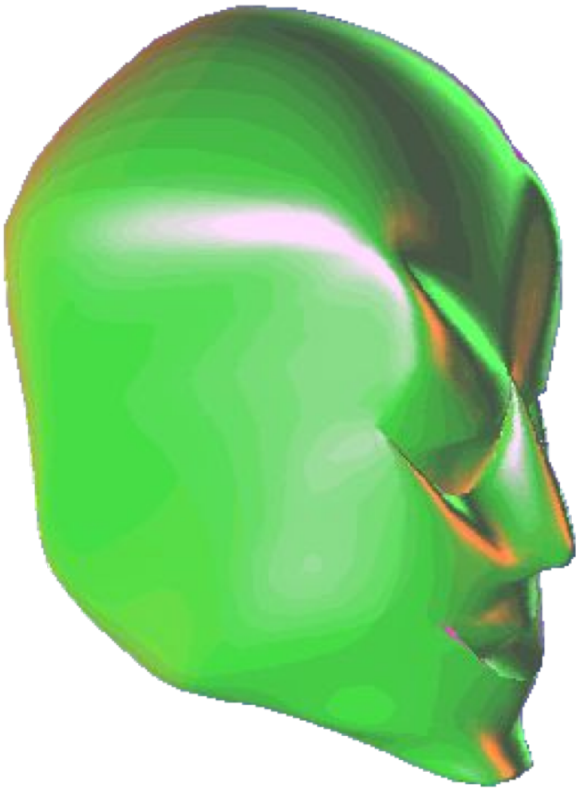


After 2nd Iteration



After 3rd Iteration

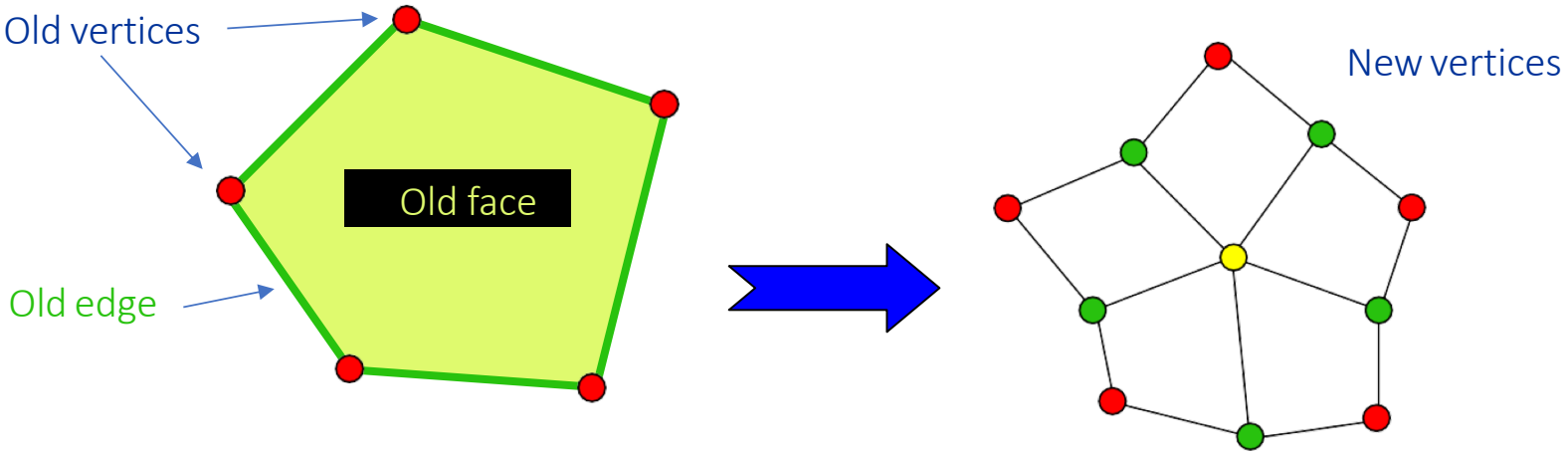




The limit surfaces of the Butterfly subdivision are smooth but are nowhere twice differentiable.

Works for control nets of arbitrary topology

- After one iteration, all the faces are quadrilateral

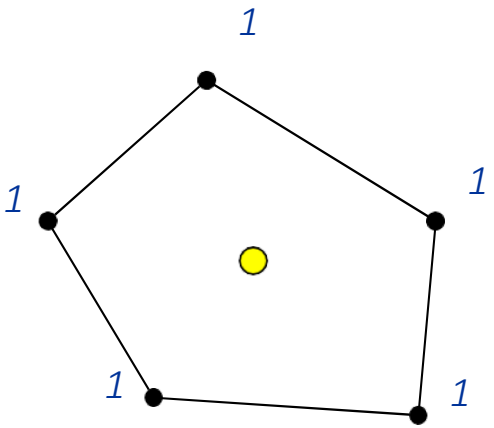


Every face is replaced by quadrilateral faces. The are three kinds of new vertices:

- Yellow vertices are associated with old faces
- Green vertices are associated with old edges
- Red vertices are associated with old vertices.

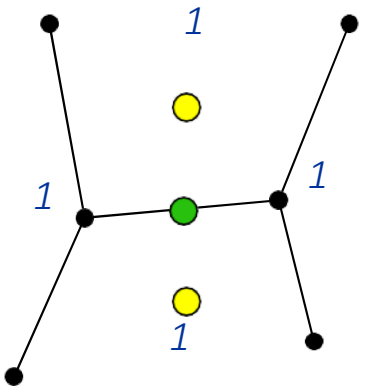
Step 1

First, all the yellow vertices are calculated



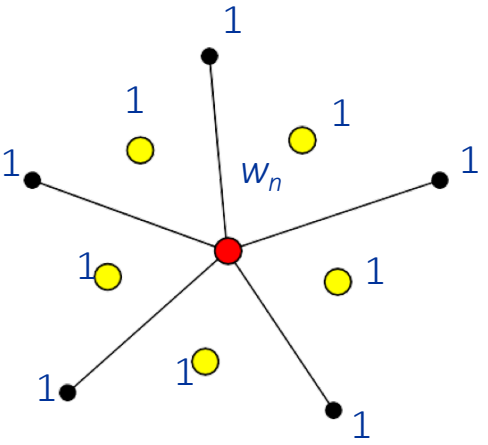
Step 2

Then the green vertices are calculated using the values of the yellow vertices



Step 3

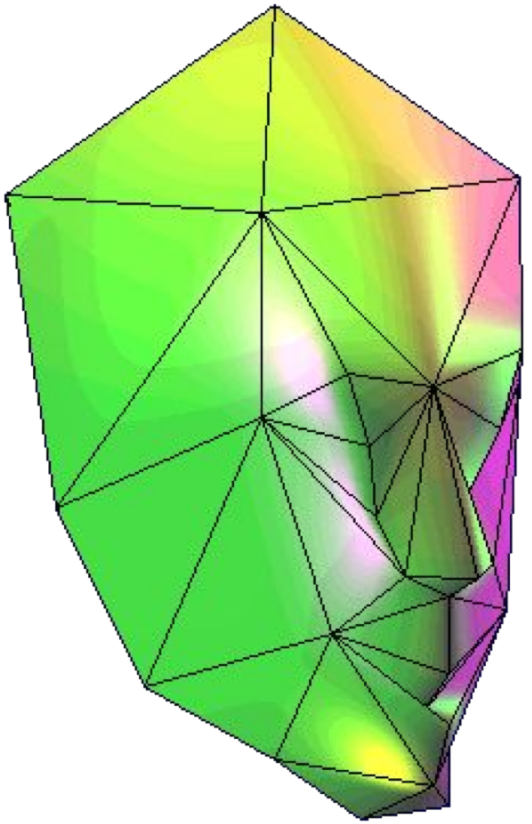
Finally, the red vertices are calculated using the values of the yellow vertices



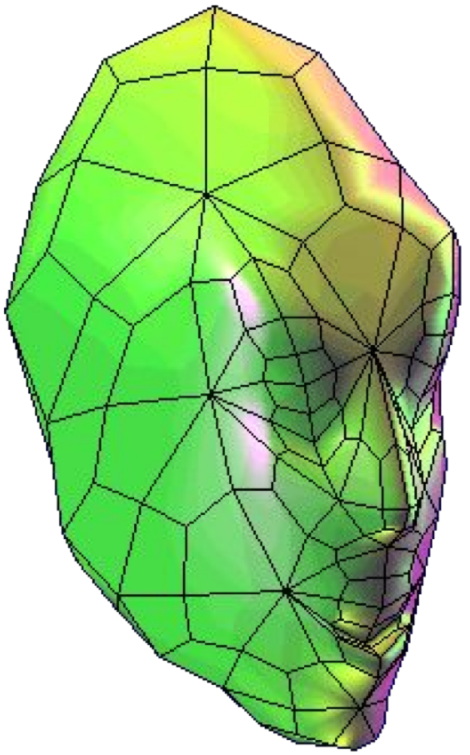
n - the vertex valence

$$w_n = n(n-2)$$

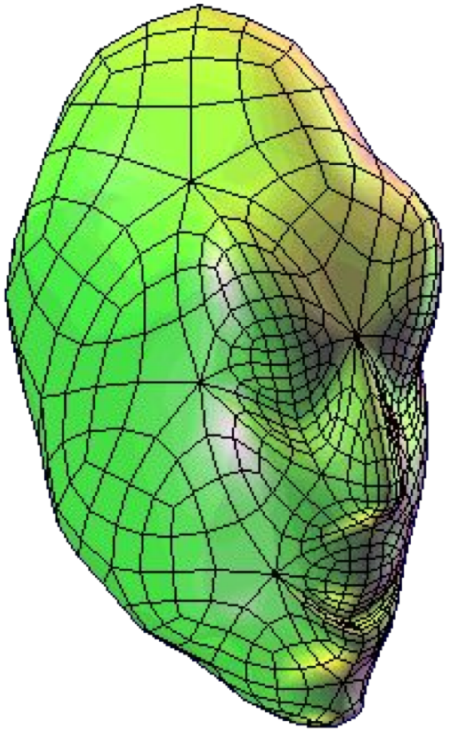
The Original Control Net



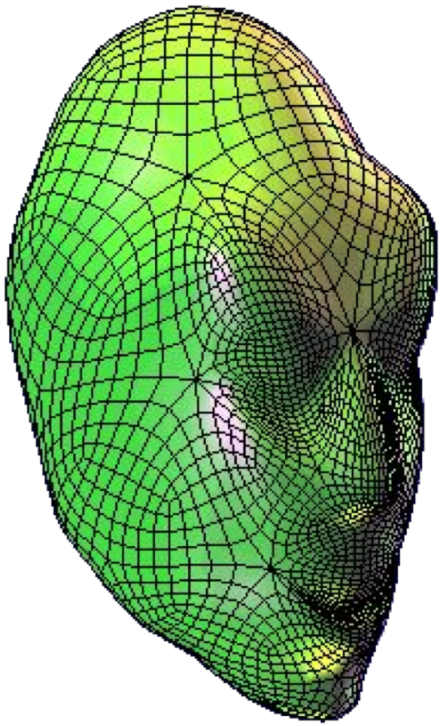
After 1st Iteration



After 2nd Iteration



After 3rd Iteration



The limit surfaces of Catmull-Clarks's subdivision have continuous curvature almost everywhere

Most surface are not smooth everywhere

- Edges & creases
- Can be marked in model
  - Weighting is changed to preserve edge or crease

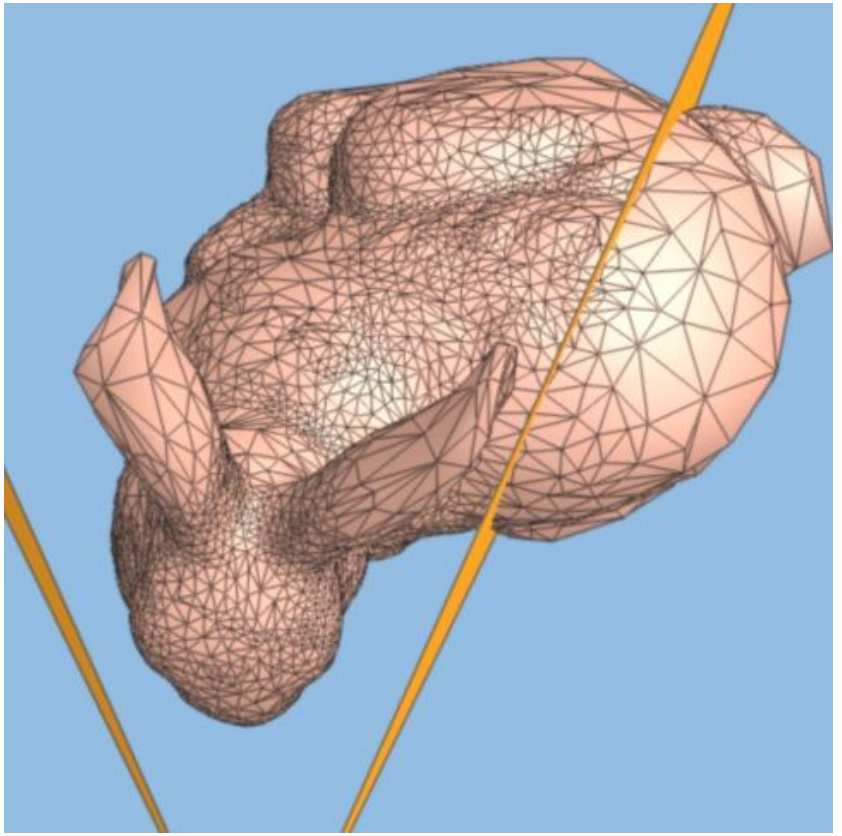
Generalization to semi-sharp creases (Pixar)

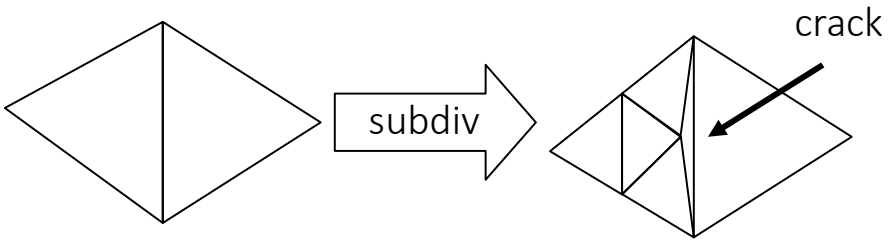
- Controllable sharpness
- Sharpness $(s) = 0$, smooth
- Sharpness $(s) = \infty$, sharp
- Achievable through hybrid subdivision step
  - Subdivision iff $s == 0$
  - Otherwise parameter is decremented

Not all regions of a model need to be subdivided

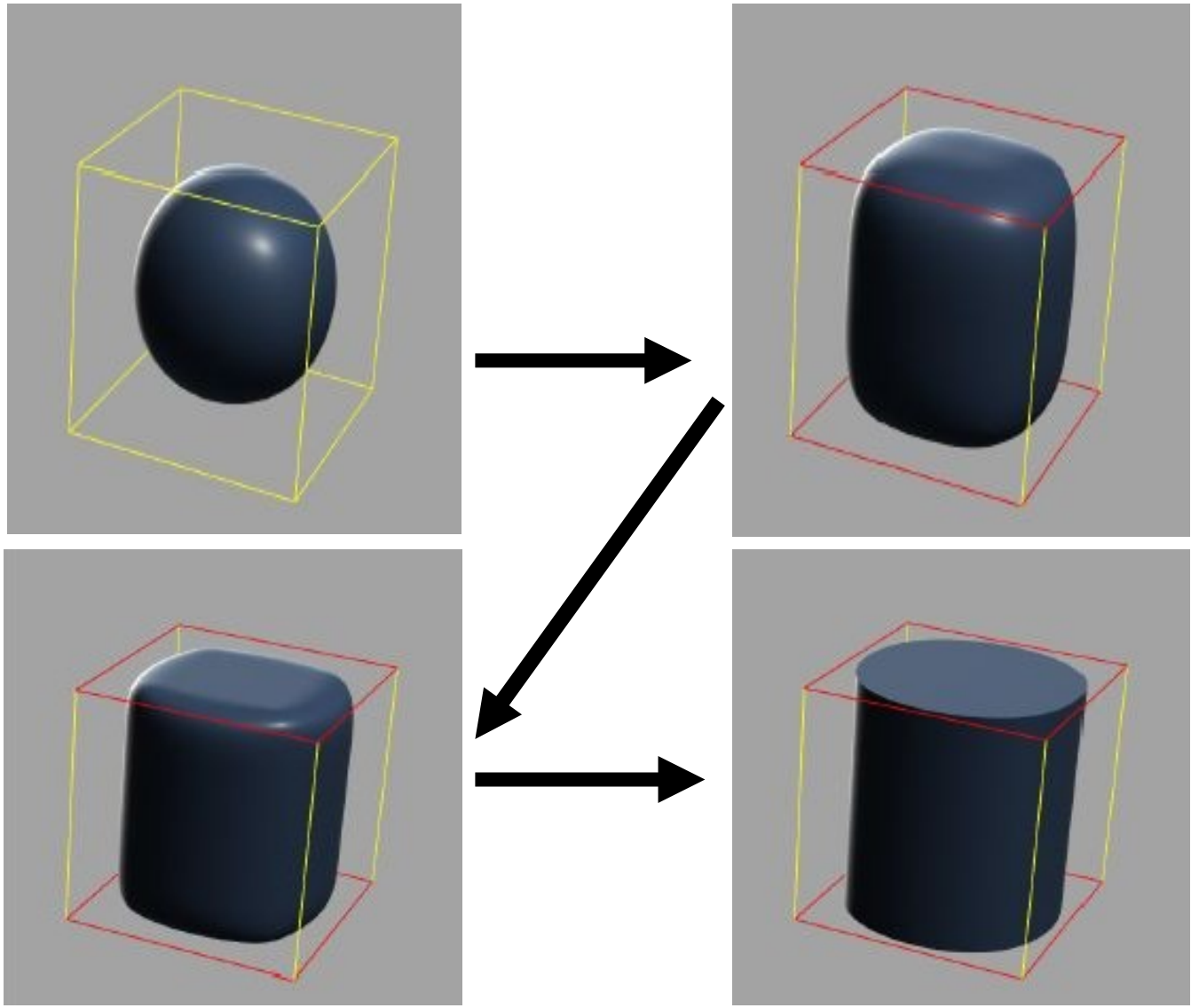
Idea: Use some criteria and adaptively subdivide mesh where needed

- Curvature
- Screen size
  - Make triangles < size of pixel
- View dependence
  - Distance from viewer
  - Silhouettes
  - In view frustum
- Careful!
  - Must avoid “cracks”

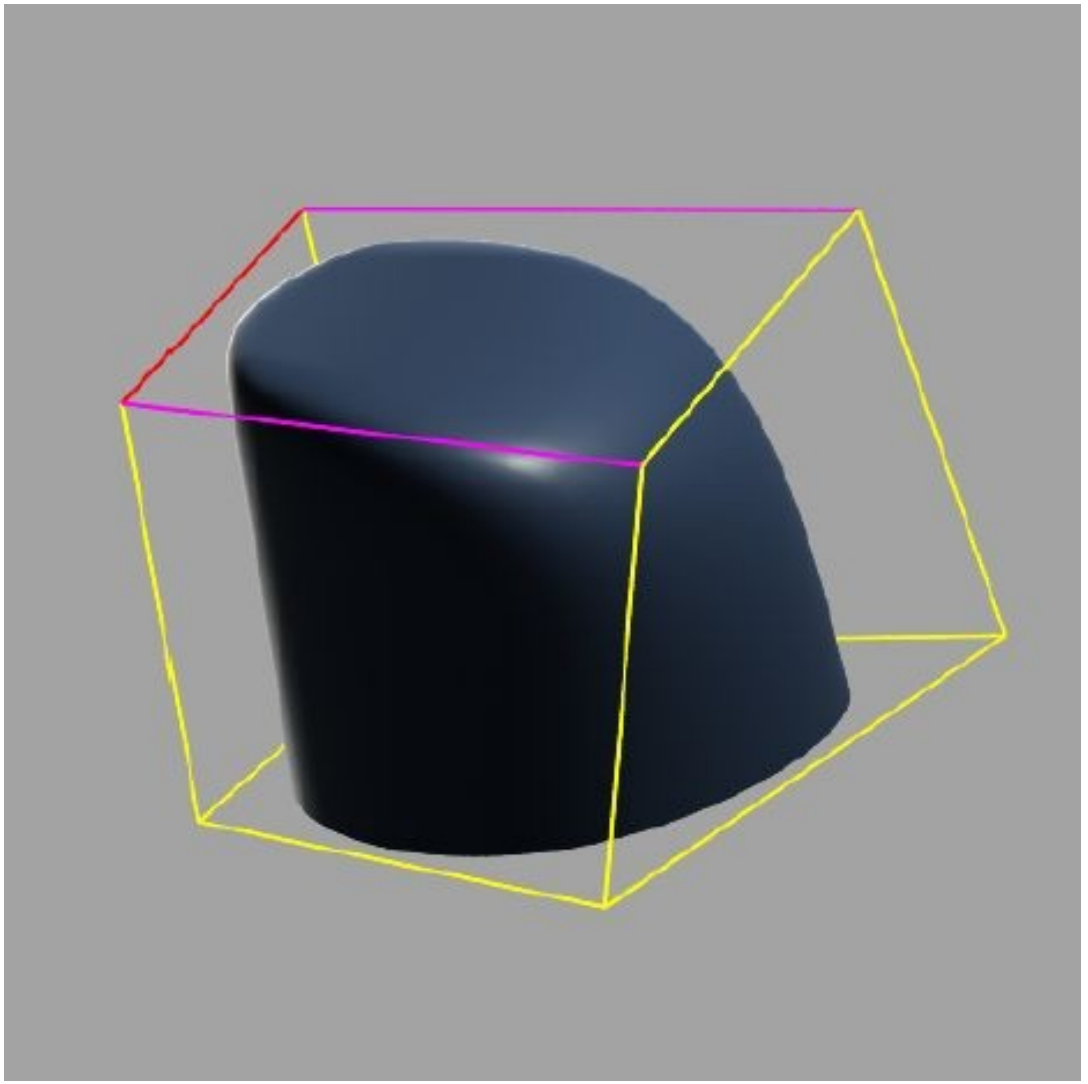




Increasing sharpness of edges



Can be changed on a edge by edge basis



Solid color painting is easy, already defined

Texturing is not so easy

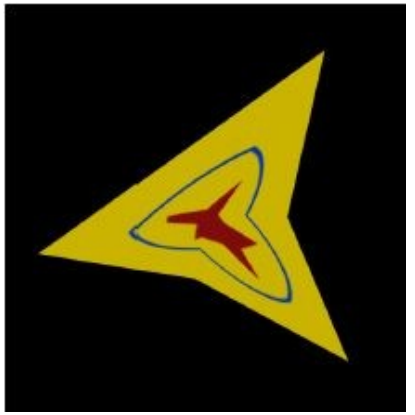
- Using polygonal methods can result in distortion

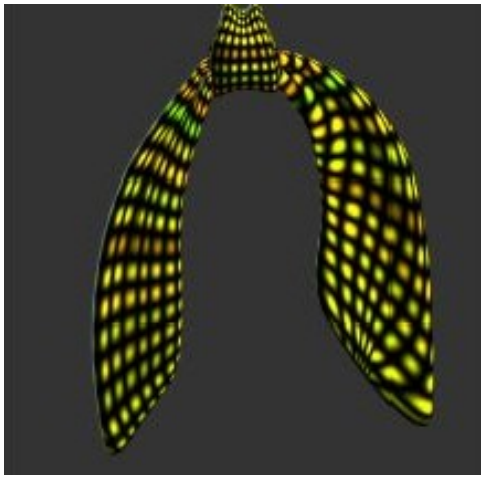
Solution

- Assign texture coordinates to each original vertex
- Subdivide them just like geometric coordinates

Introduces a smooth scalar field

- Used for texturing in Geri's jacket, ears, nostrils





Hierarchical Modeling

- Store offsets to vertices at different levels
- Offsets performed in normal direction
- Can change shape at different resolutions while rest stays the same

Surface Smoothing

- Can perform filtering operations on meshes
  - *E.g.* (Weigthed) averaging of neighbors

Level-of-Detail

- Can easily adjust maximum depth for rendering

Advantages

- Simple method for describing complex surfaces
- Relatively easy to implement
- Arbitrary topology
- Local support
- Guaranteed continuity
- Multi-resolution

Difficulties

- Intuitive specification
- Parameterization
- Intersections