Computer Graphics I

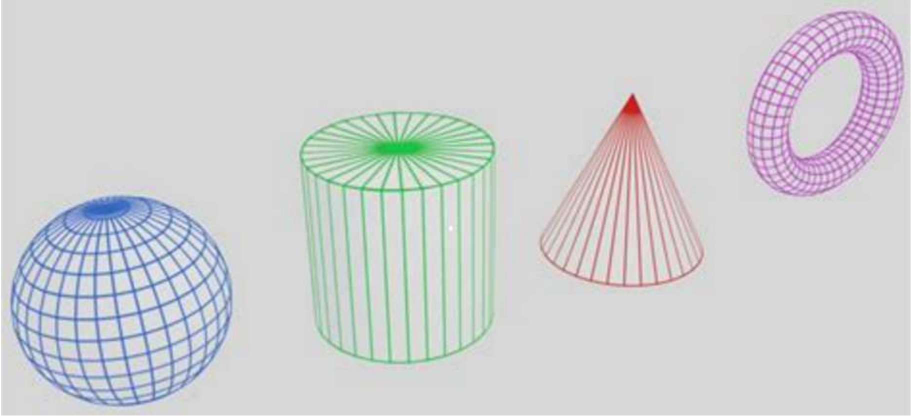
Lecture 6:
Geometric modeling 2

Xiaopei LIU

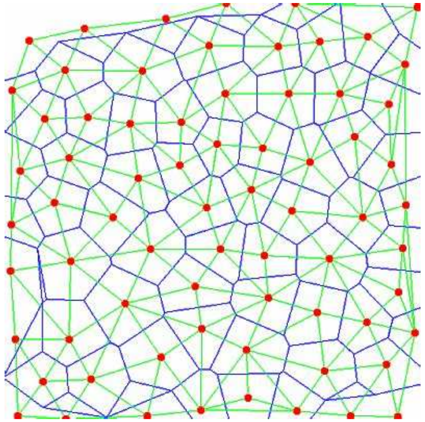
School of Information Science and Technology
ShanghaiTech University

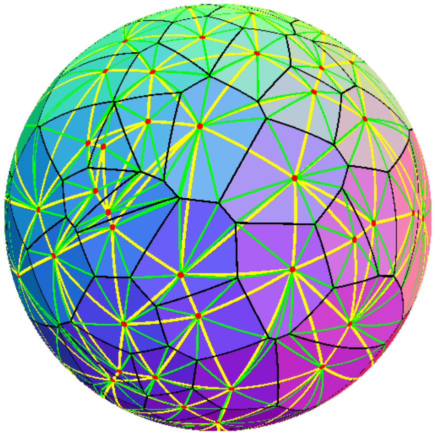
What can we do now?

- **Draw simple geometries**



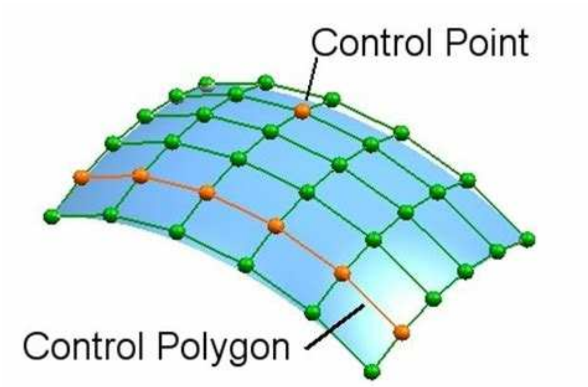
- **Triangulate scattered points to form triangle meshes**

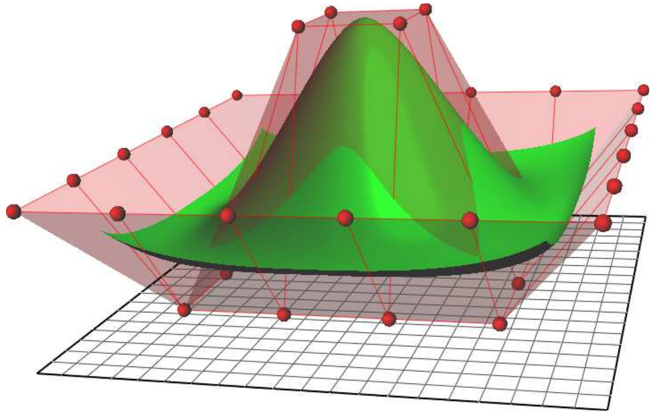




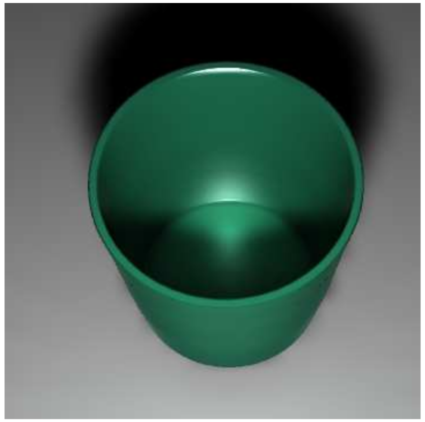
What can we do now?

- **Construct free surface meshes and draw them**





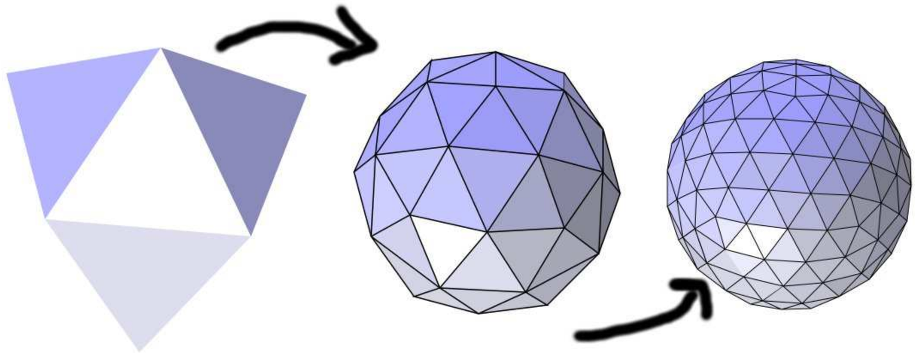
- **Project & rasterize geometries and render them**



Additional modeling techniques?

- **Subdivision and simplification**

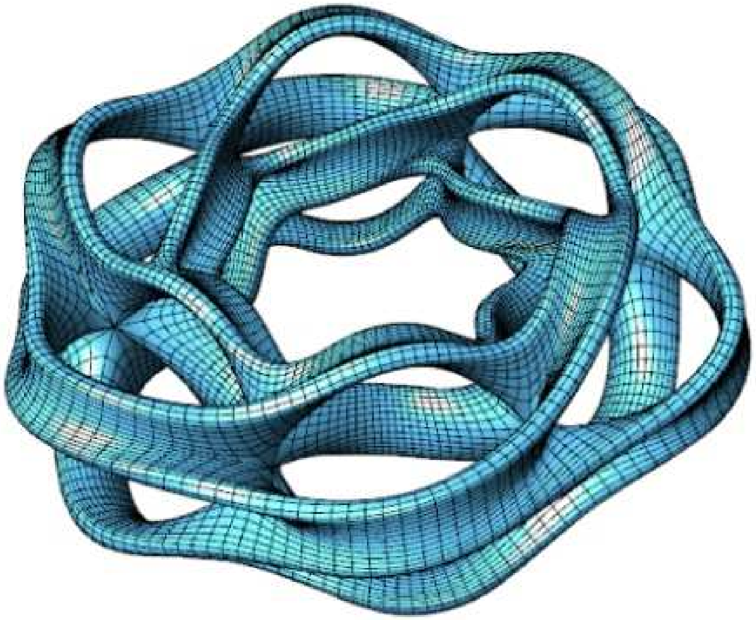
Subdivision

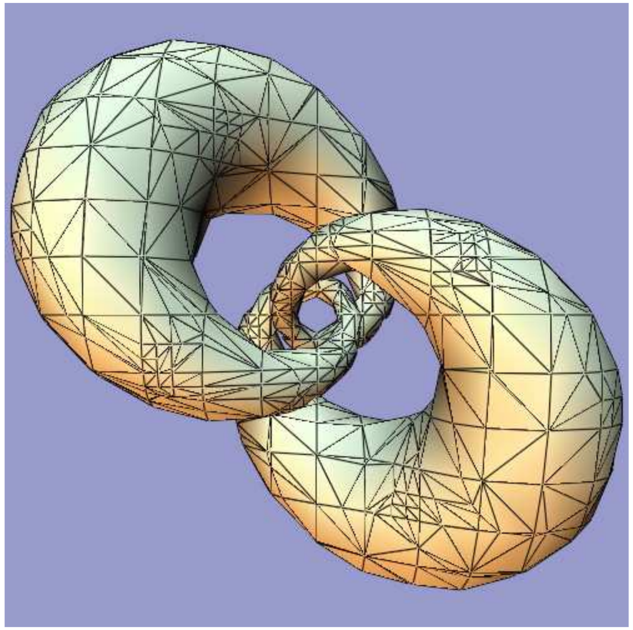


Simplification

Additional modeling techniques?

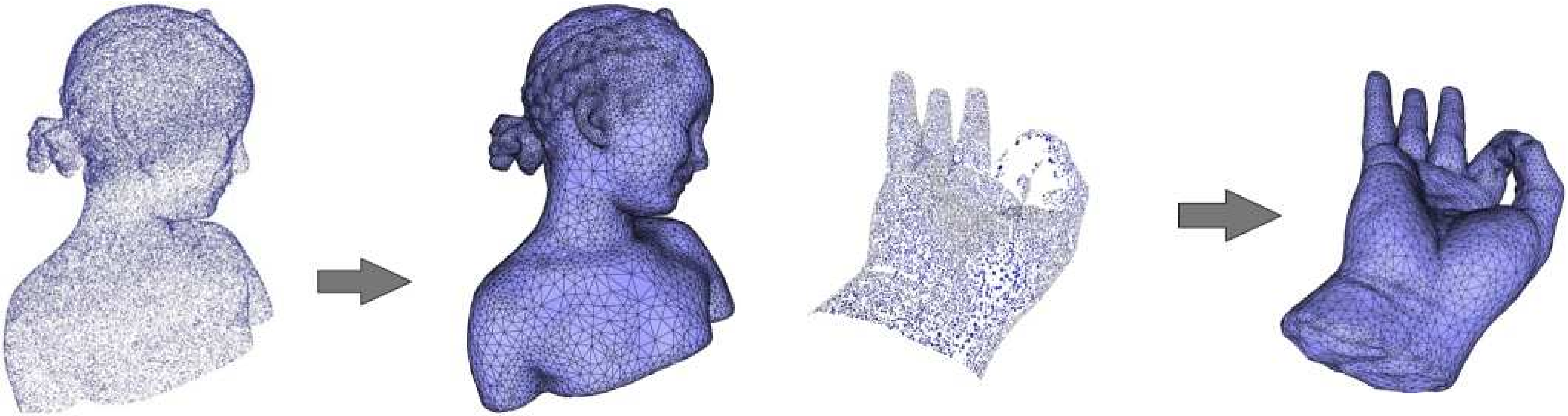
- **Implicit approach**
  - Surface meshes from implicit functions (sampled data)
  - Isosurface from level set





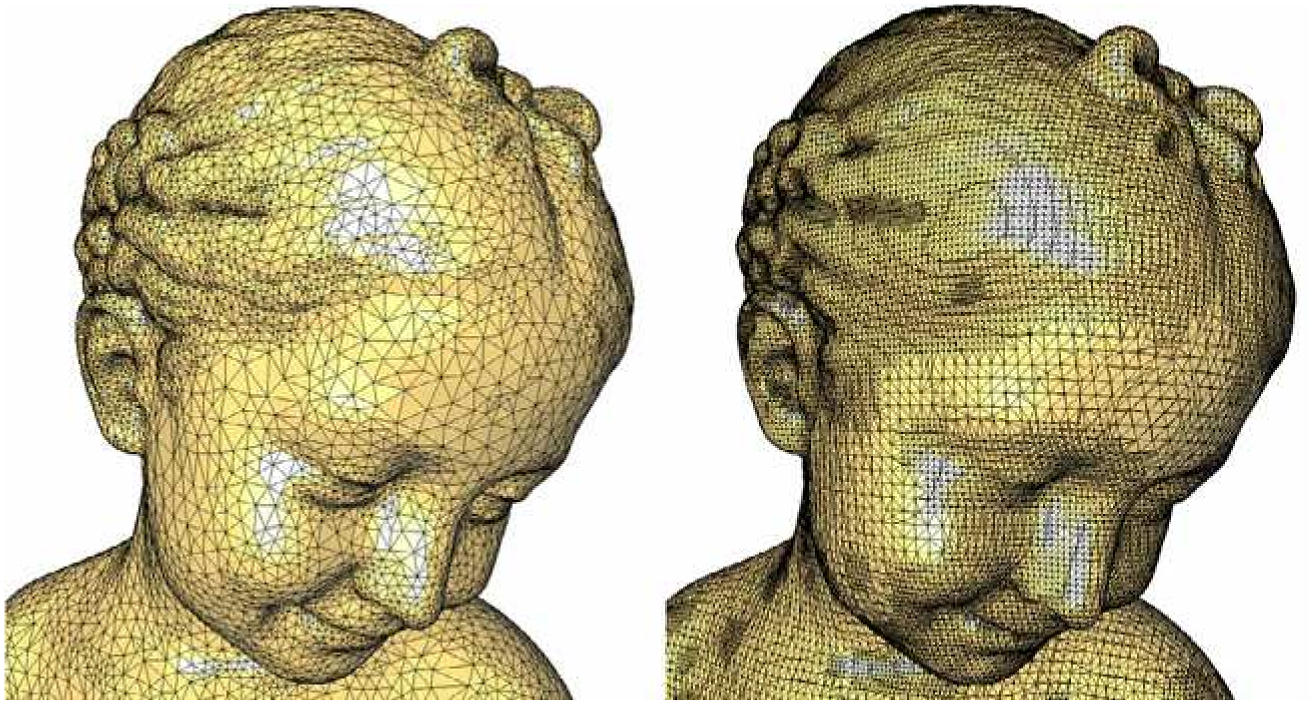
Additional modeling techniques?

- **Implicit approach**
  – Surface mesh reconstruction from point cloud



Additional modeling techniques?

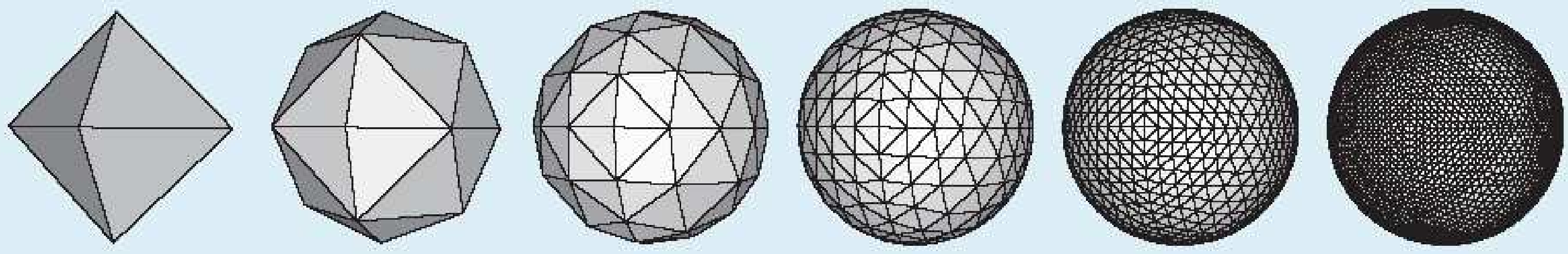
- **Remeshing**
  - Reorganize mesh elements with better quality



1. Mesh subdivision/refinement

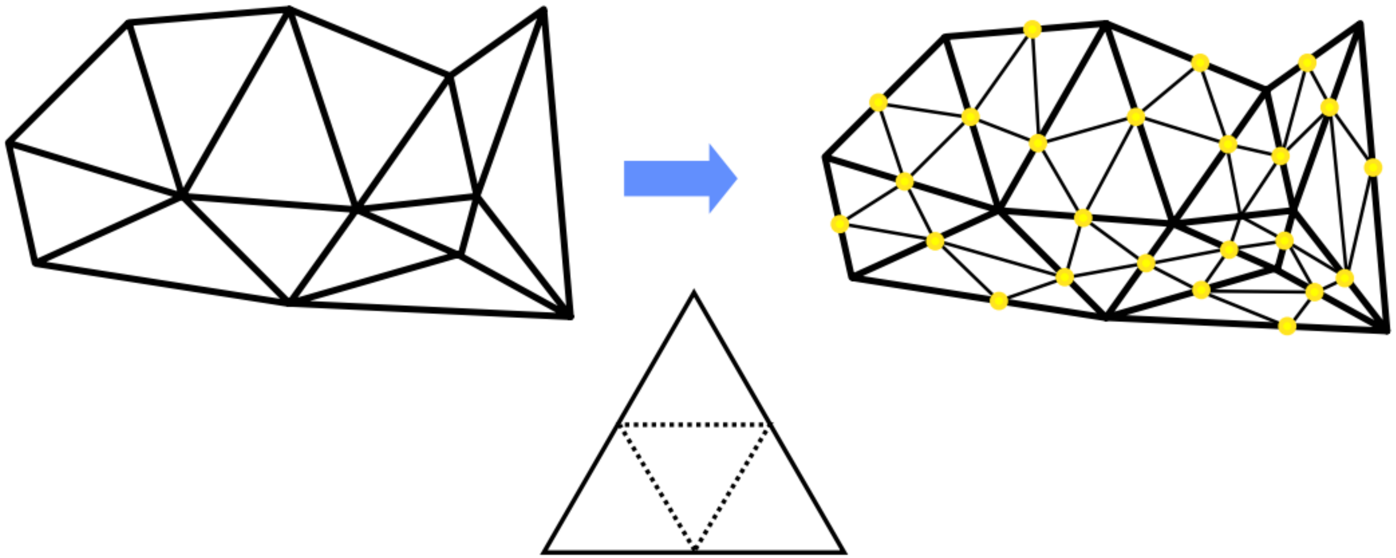
Mesh subdivision

- **Subdivision surface**
  - A method of representing a smooth surface via the specification of a coarser piecewise linear polygon mesh
  - The underlying concepts are derived from spline refinement algorithms



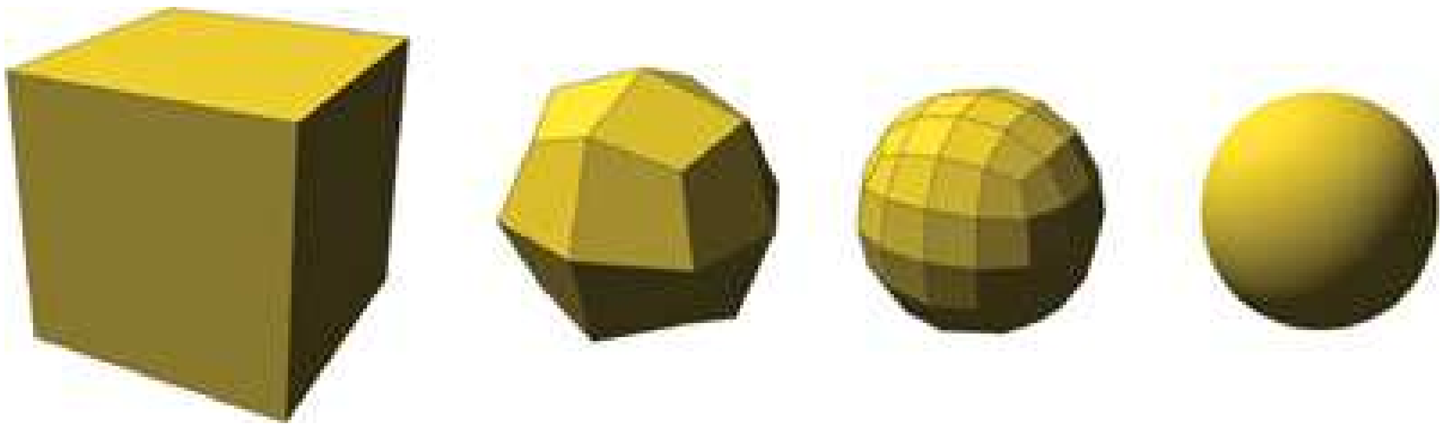
Mesh subdivision

- **Overview**
  - Subdivision surfaces are defined ***recursively***
  - Starting with a given polygonal mesh, a (convergent) *subdivision scheme* is applied to this mesh



Mesh subdivision

- **Catmull–Clark subdivision scheme**
  – Devised by Edwin Catmull and Jim Clark in 1978
  – A generalization of bi-cubic uniform B-spline surfaces to arbitrary topology



Mesh subdivision

- **Catmull–Clark subdivision scheme**
  - Add a new face point

$$v_F = \Sigma_{i=1}^{n} \frac{1}{n} v_i$$

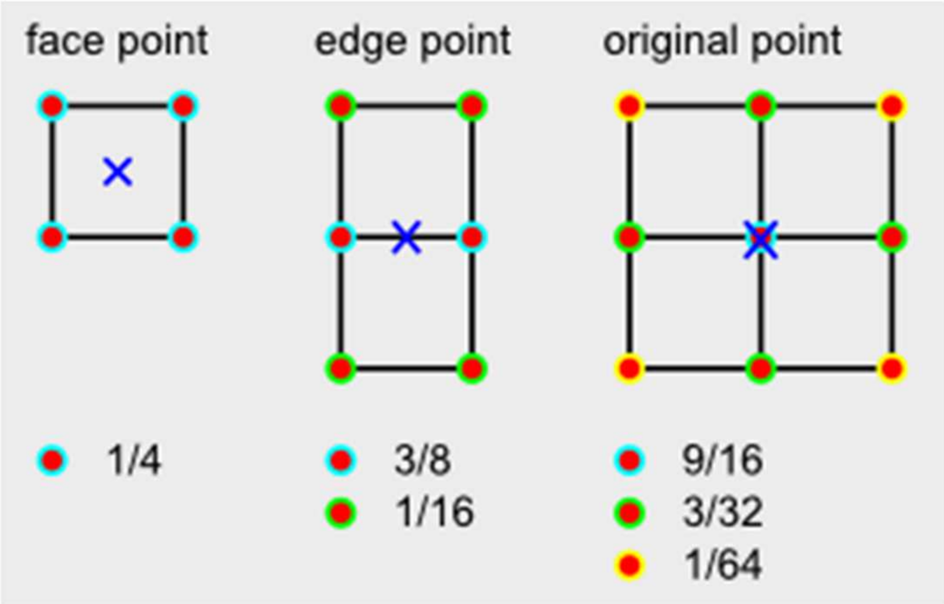
  - Add a new edge point
    - End points v and w and adjacent faces F1 and F2

$$v_E = \frac{v + w + v_{F_1} + v_{F_2}}{4}$$

  - Update the original vertex point $v' = \frac{1}{n}Q + \frac{2}{n}R + \frac{n-3}{n}v$
    - v: the original vertex point
    - Q: average of the new face points for all faces adjacent to v
    - R: average of the midpoints of the n edges connected to v

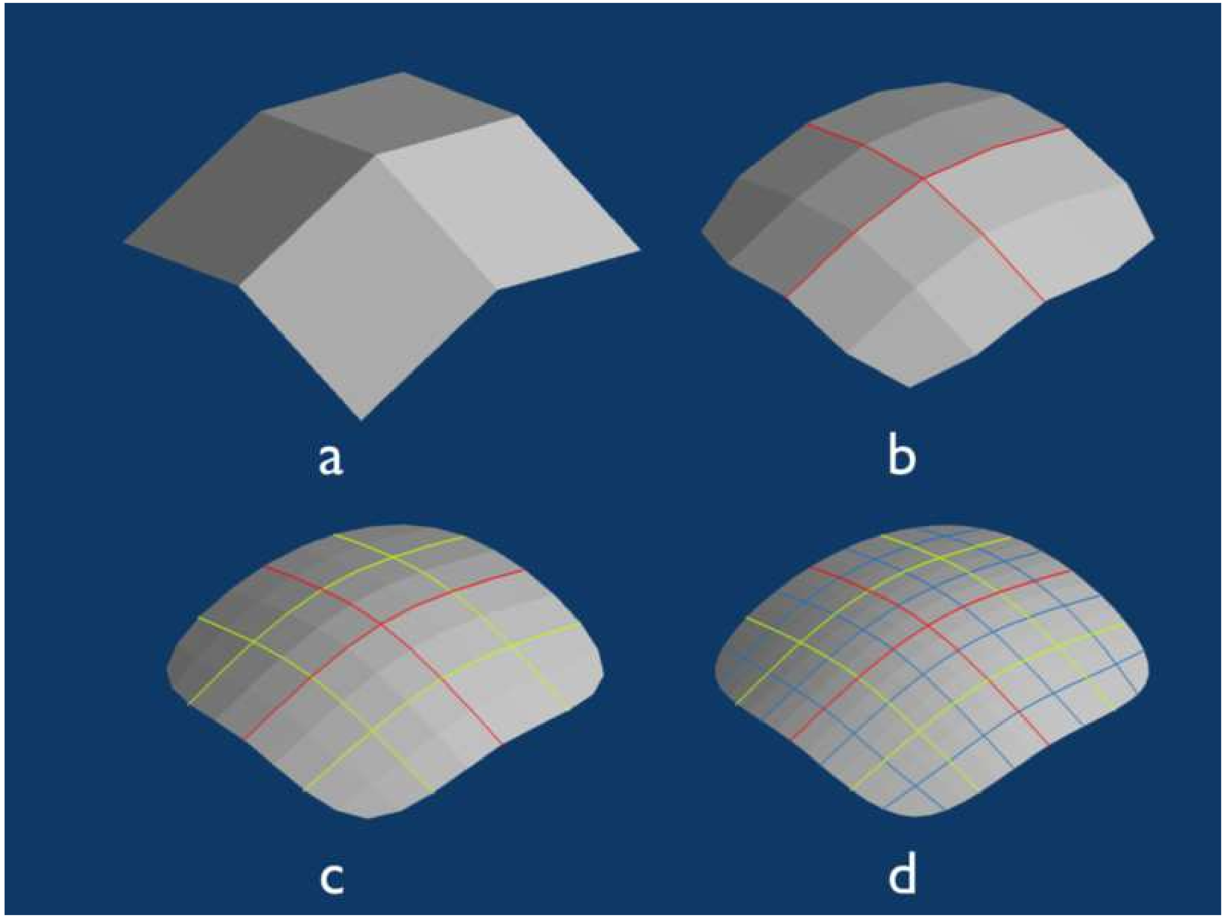
Mesh subdivision

- **Catmull–Clark subdivision scheme**
  – Subdivision scheme Illustration



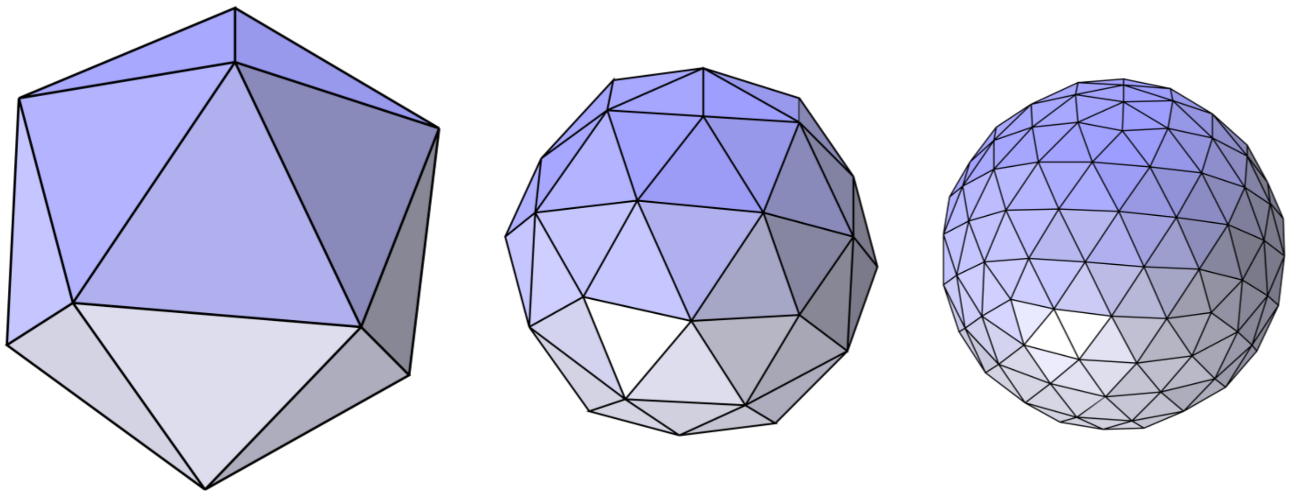
Mesh subdivision

- **Catmull–Clark subdivision scheme**
  - Mesh structure



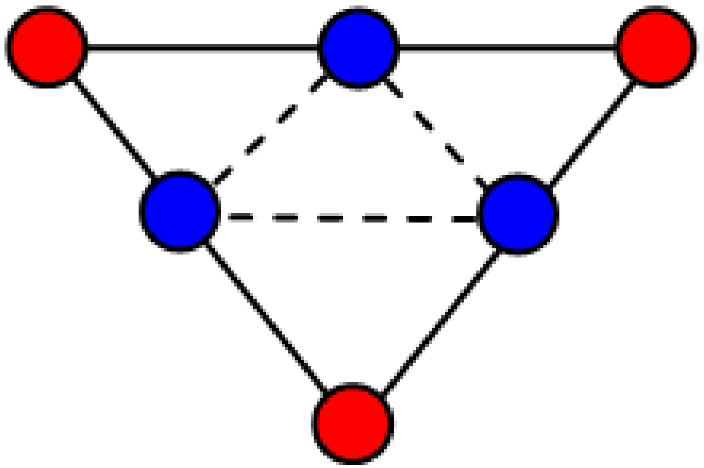
Mesh subdivision

- **Loop subdivision surfaces**
  - Quadrilateral based meshes generally use Catmull-Clark subdivision scheme
  - Triangle based meshes generally use loop subdivision



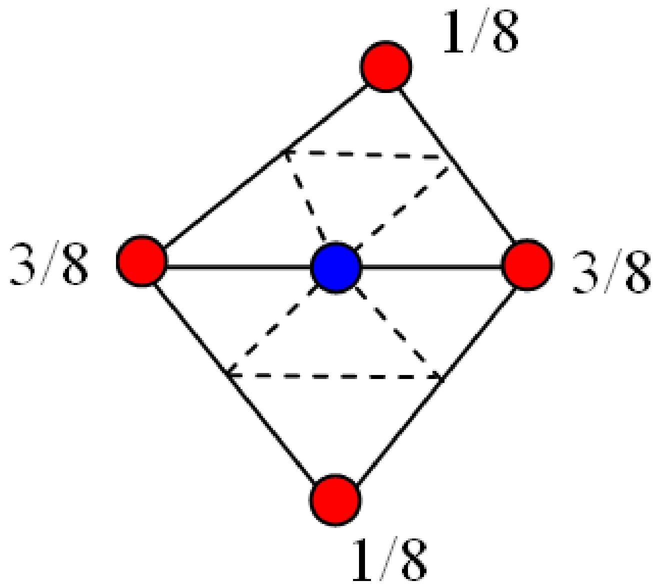
Mesh subdivision

- **Loop subdivision surfaces**
  - For every edge in the source mesh, add a vertex (shown in blue) and for every triangle on the mesh, create the four triangles



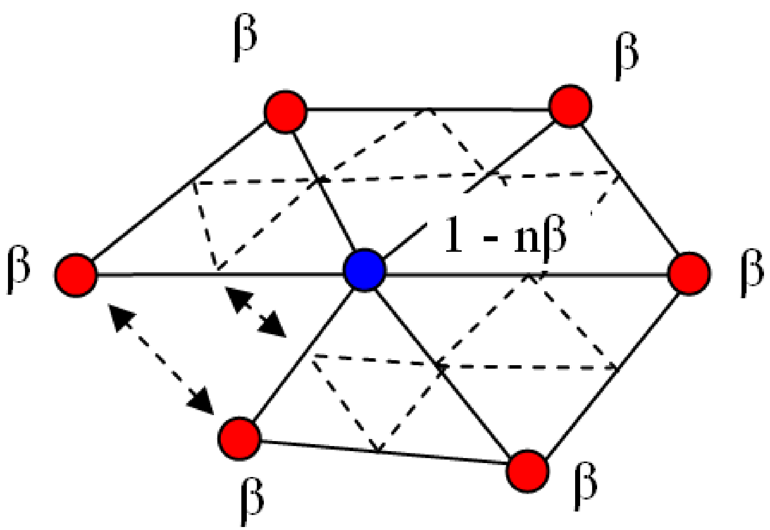
Mesh subdivision

- **Loop subdivision surfaces**
  - Every edge in the source mesh has two adjacent faces
  - We just take the a linear combination of the source vertices to have the location of the vertex associated with this edge.



Mesh subdivision

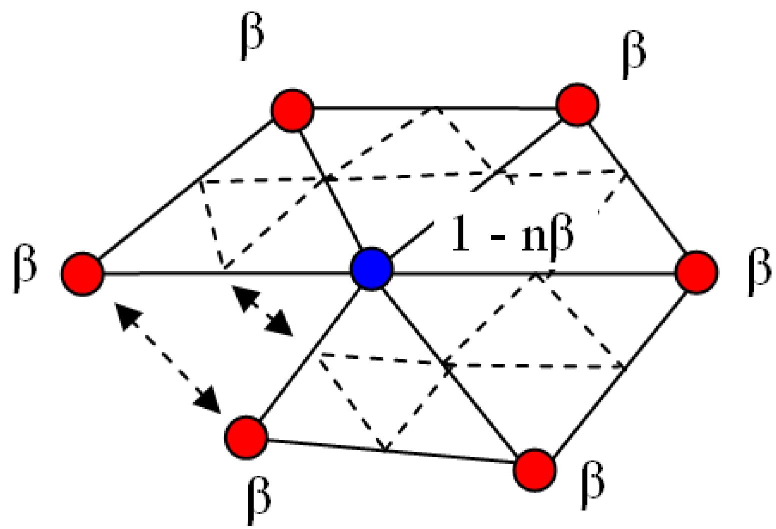
- **Loop subdivision surfaces**
  - Every vertex in the source mesh is also in the subdivided mesh
  - Its new position is computed depending on all the vertices connected to the vertex by an edge



Mesh subdivision

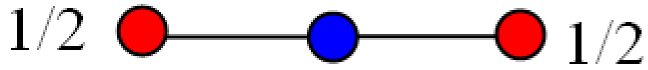
- **Loop subdivision surfaces**
  - The number of such vertices, n, determines the constant beta
  - There are many options available, but the simplest choice is

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

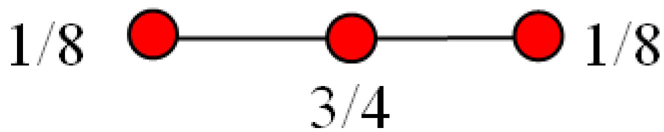


Mesh subdivision

- **Loop subdivision surfaces**
  - The boundary cases are based on basic spline refinement schemes and are equally simple. For a new edge vertex:



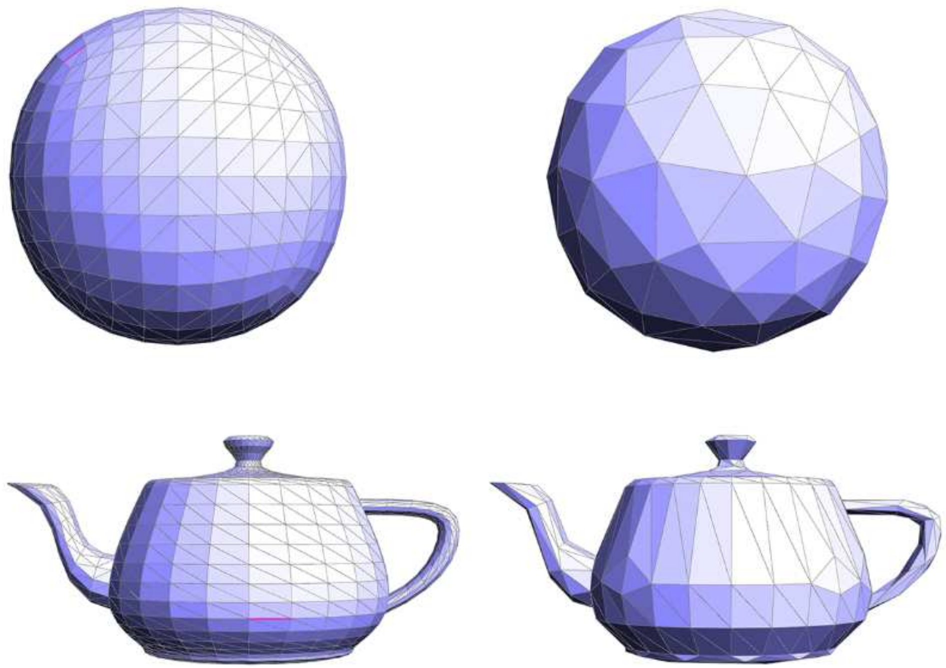
  - And for a boundary vertex:



2. Mesh simplification/coarsening

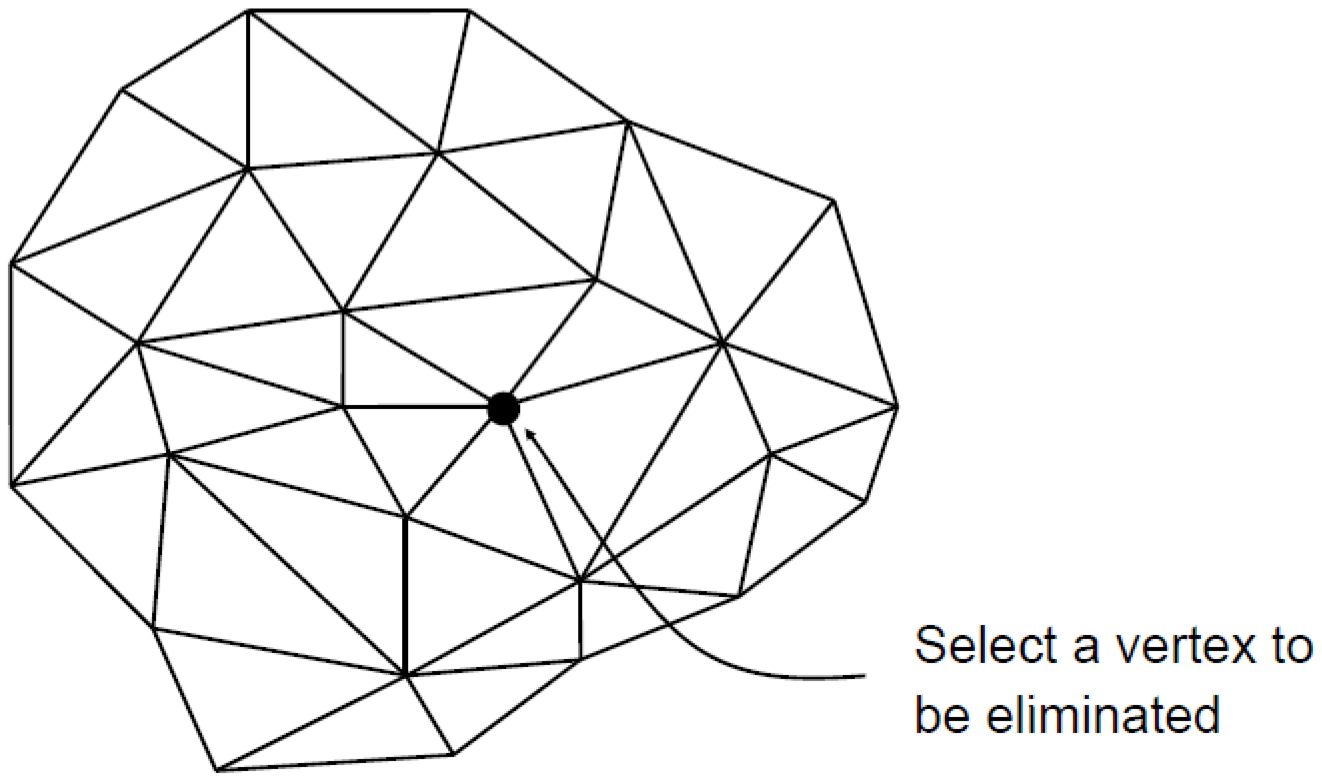
What is mesh simplification/coarsening?

- **The process to reduce the number of vertex/face of a polygonal mesh**
  - Approximate the same shape with fewer primitives
  - Inverse process of mesh subdivision/refinement



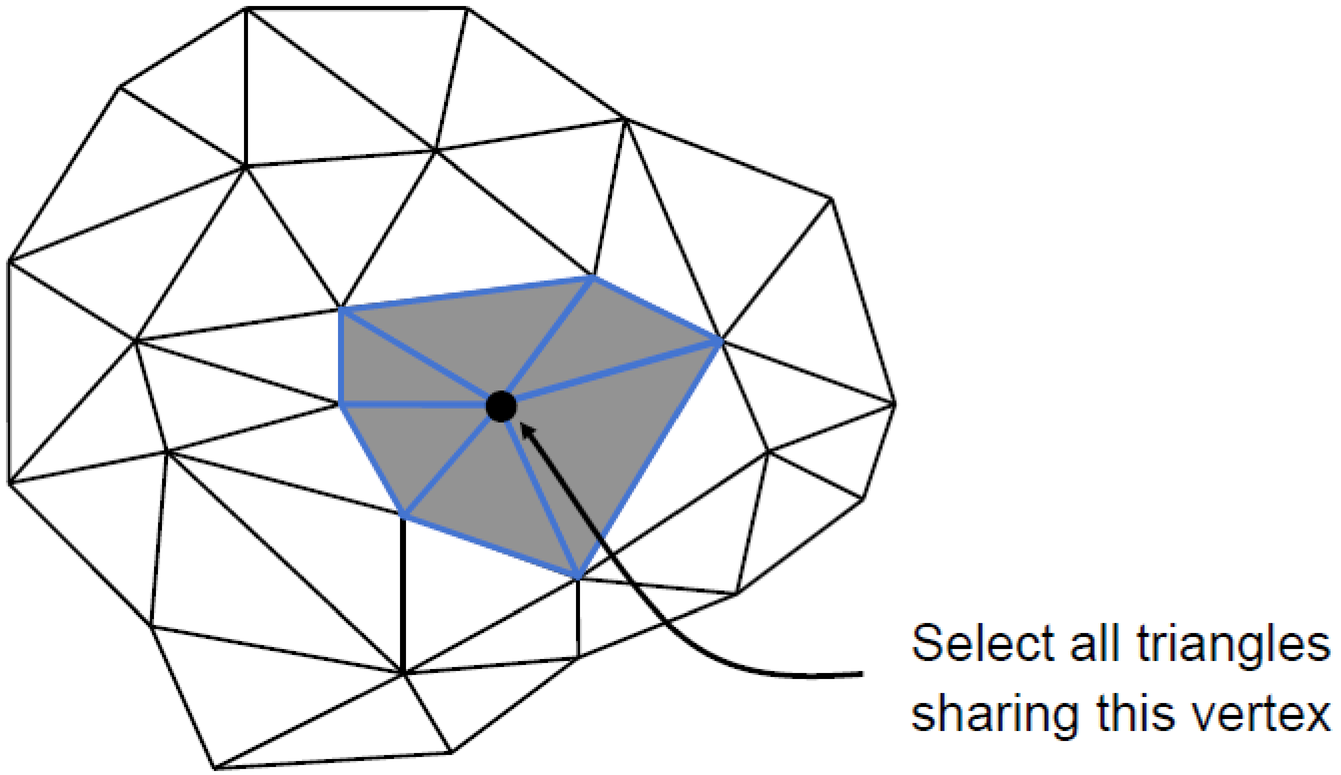
Decimation operator

- **Vertex removal**



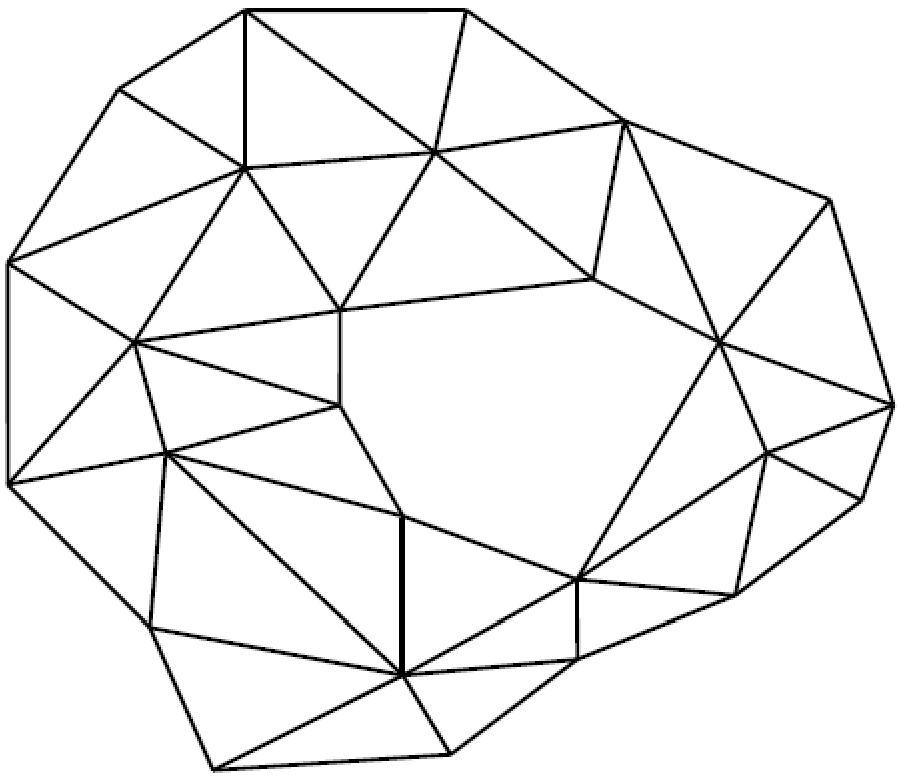
Decimation operator

- **Vertex removal**



Decimation operator

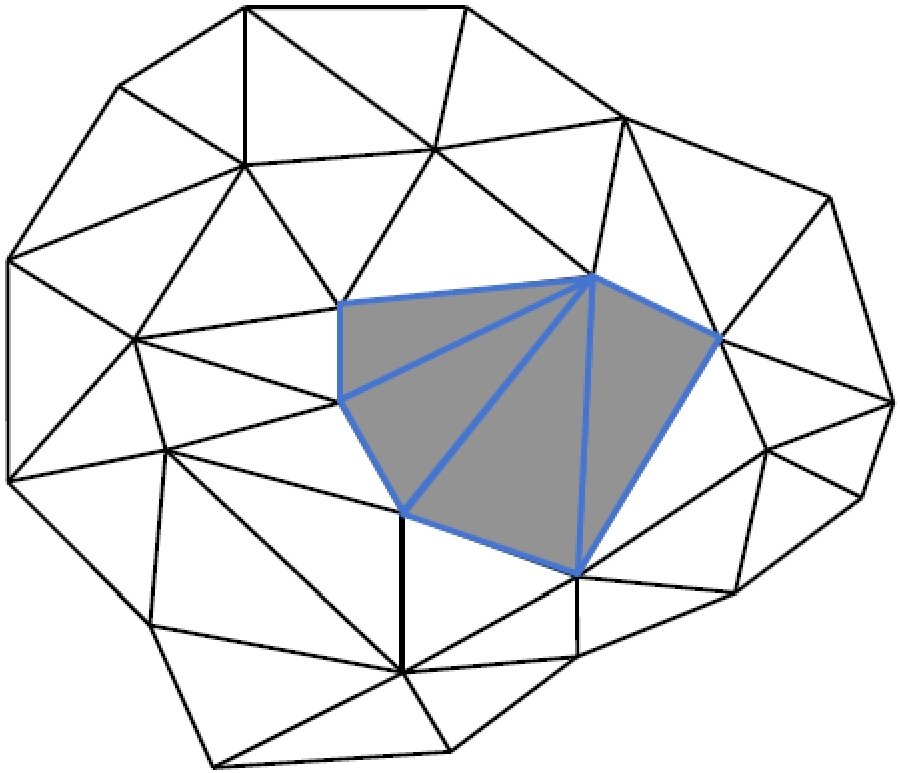
- **Vertex removal**



Remove the
selected triangles,
creating the hole

Decimation operators

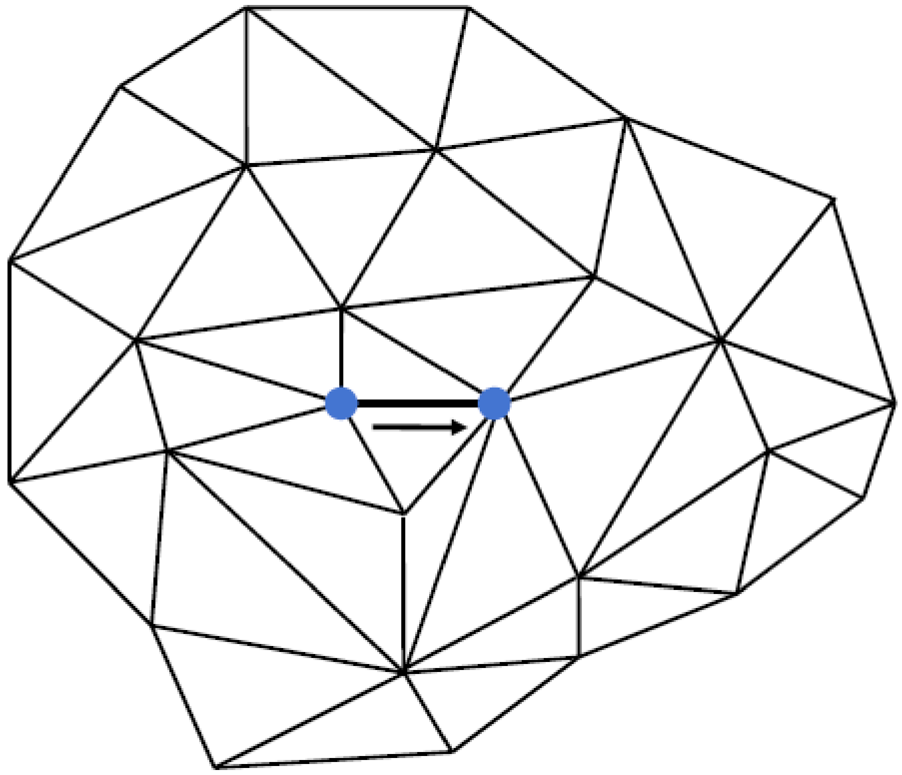
- **Vertex removal**



Fill the hole with new triangles

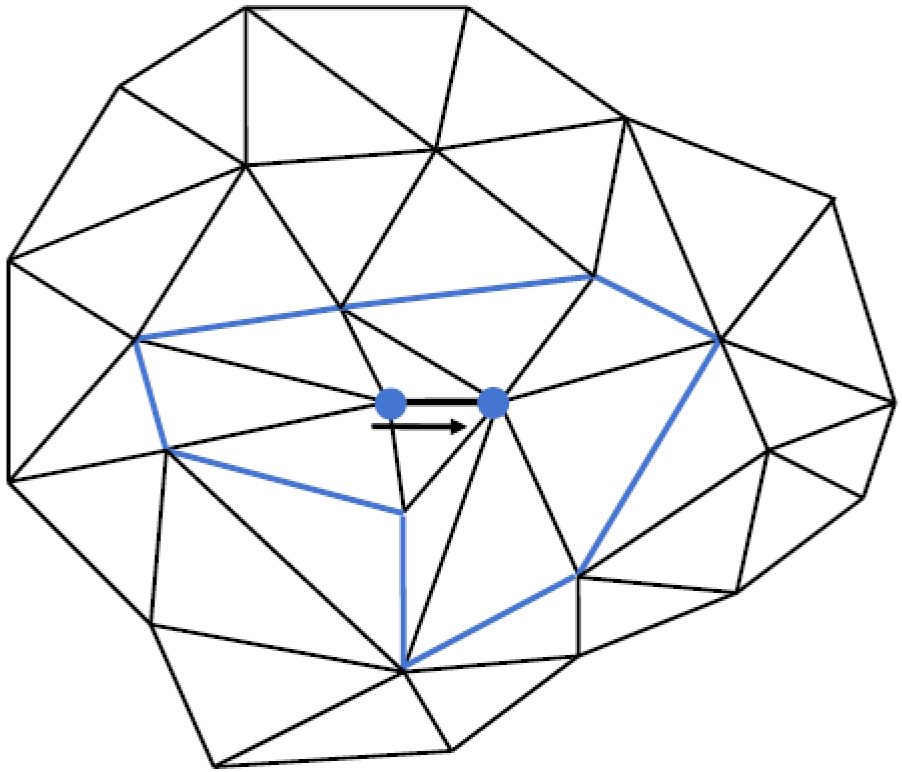
Collapsing operator

- **Half-edge collapse**



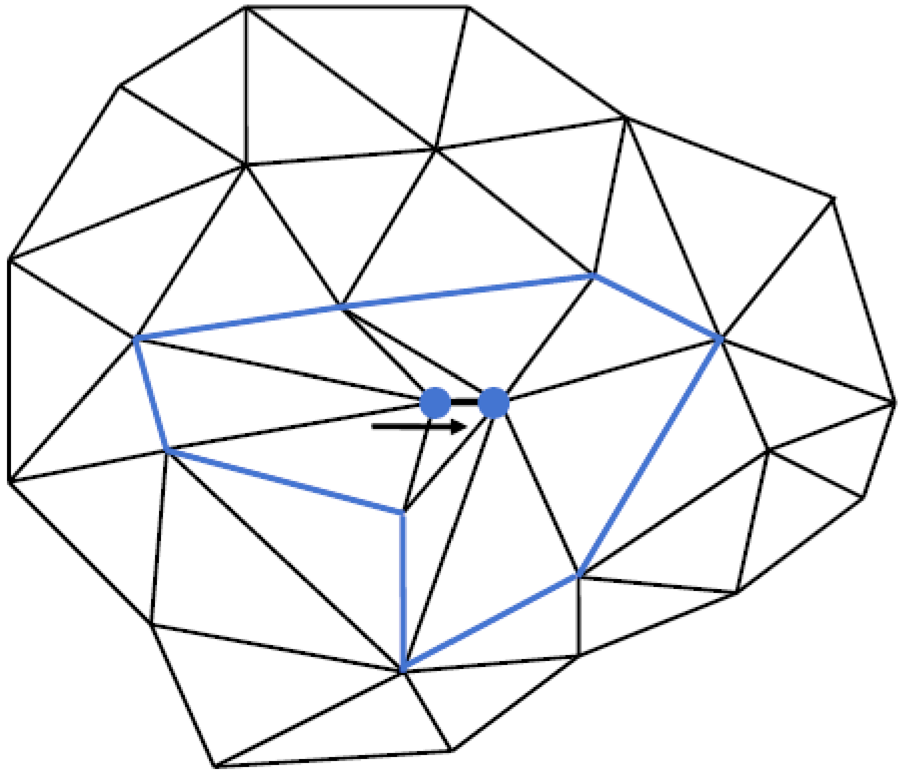
Collapsing operator

- **Half-edge collapse**



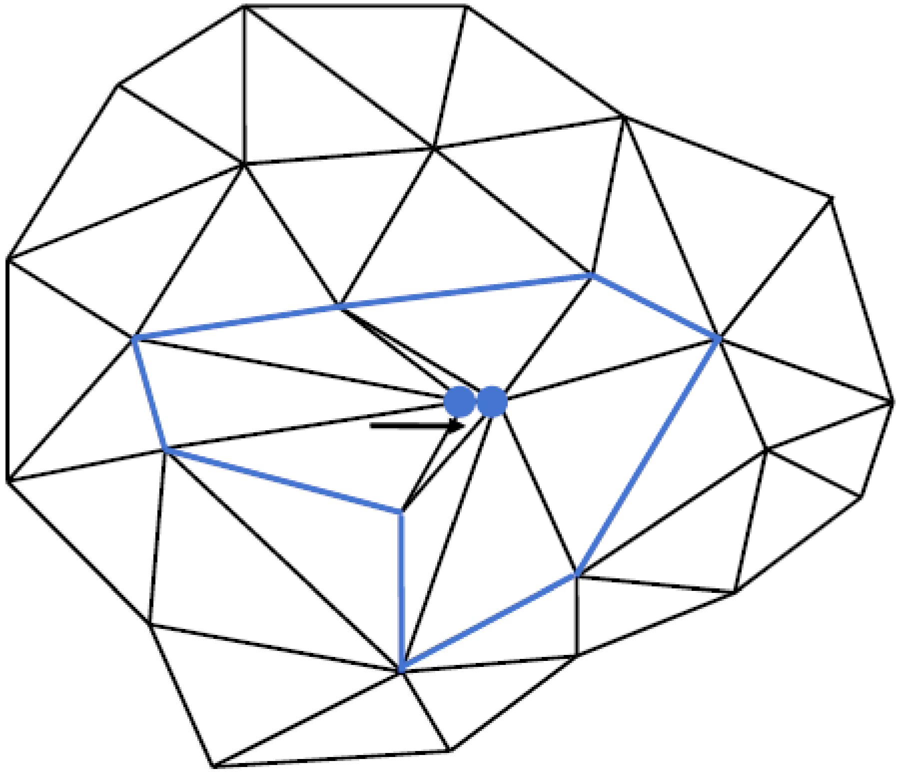
Collapsing operator

- **Half-edge collapse**



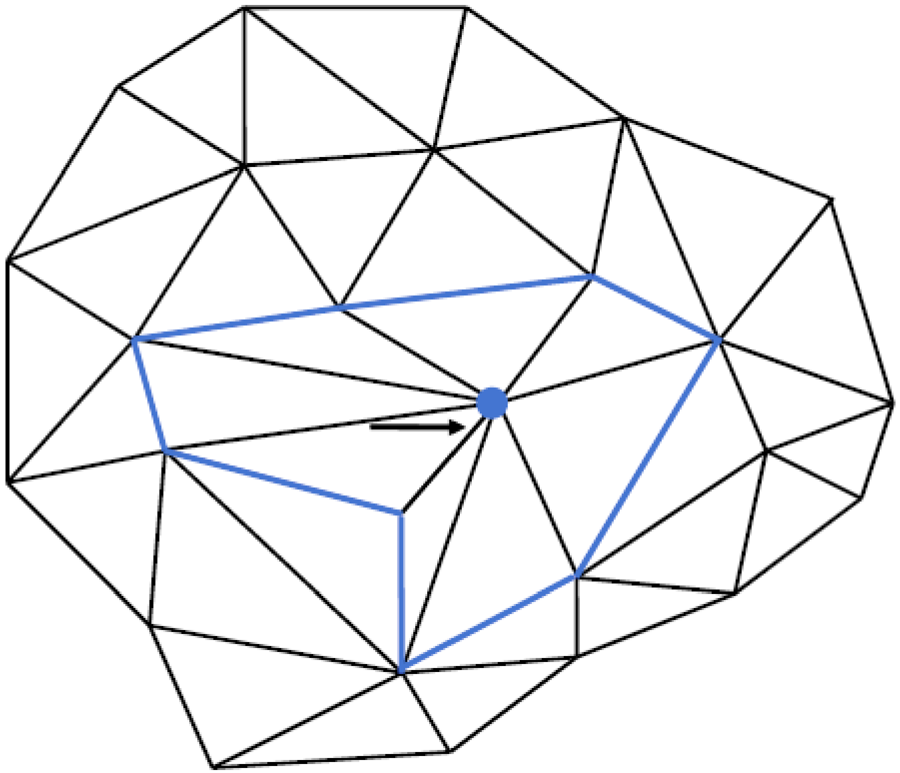
Collapsing operator

- **Half-edge collapse**



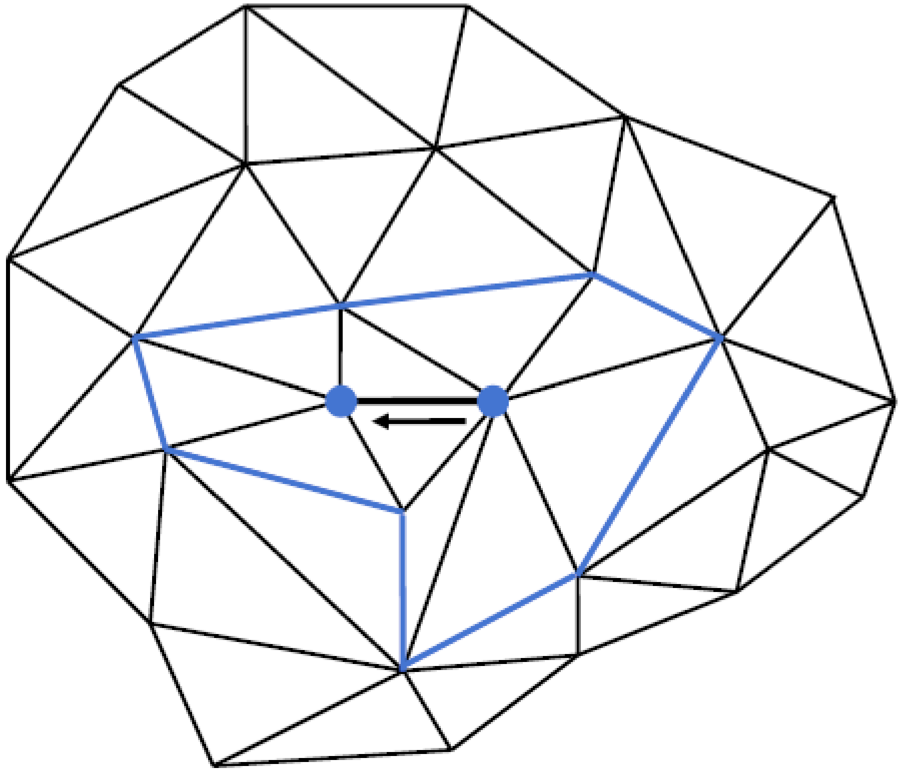
Collapsing operator

- **Half-edge collapse**



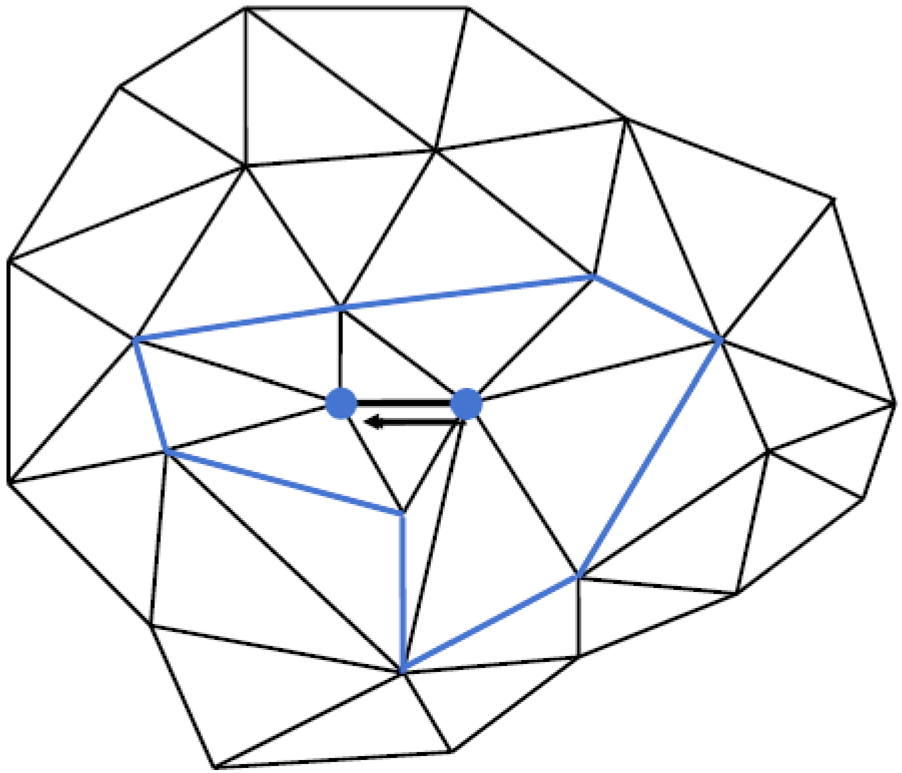
Collapsing operator

- **Half-edge collapse**



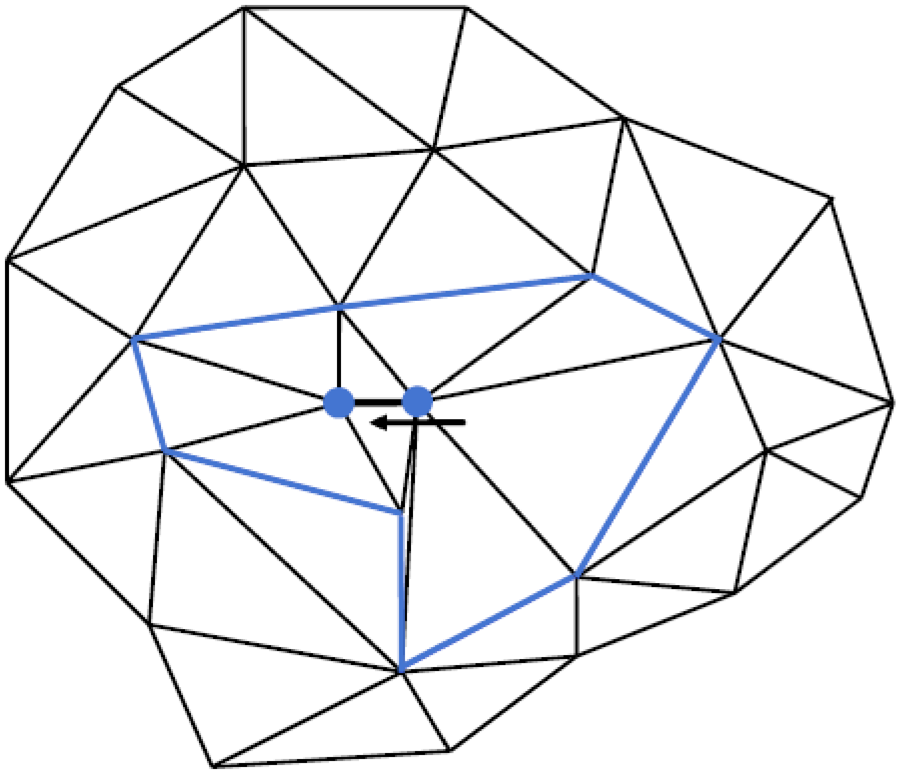
Collapsing operator

- **Half-edge collapse**



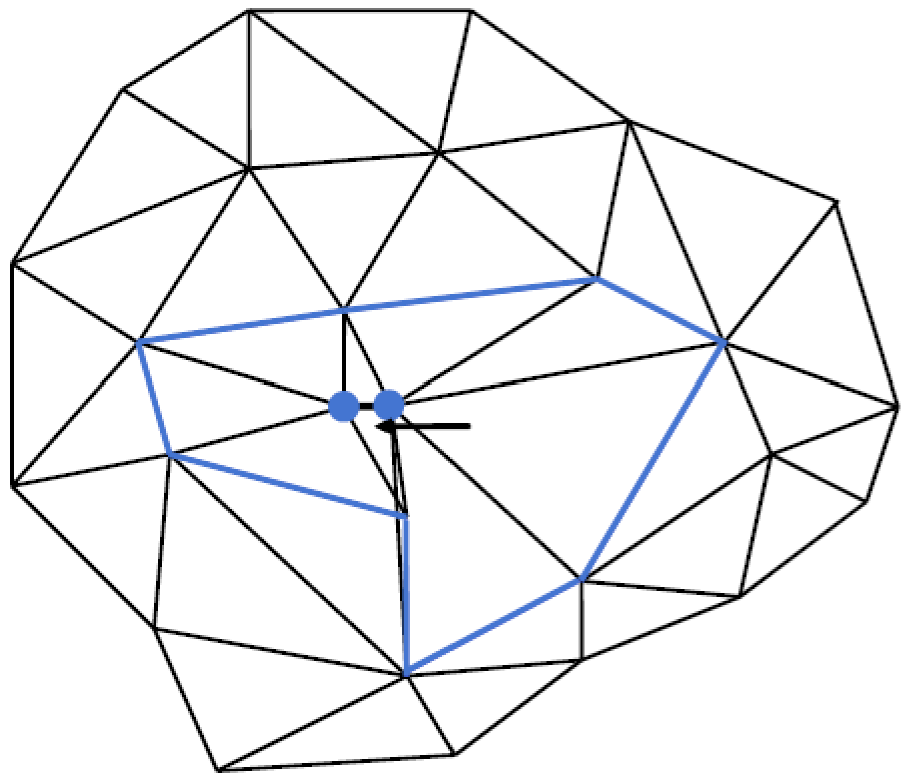
Collapsing operator

- **Half-edge collapse**



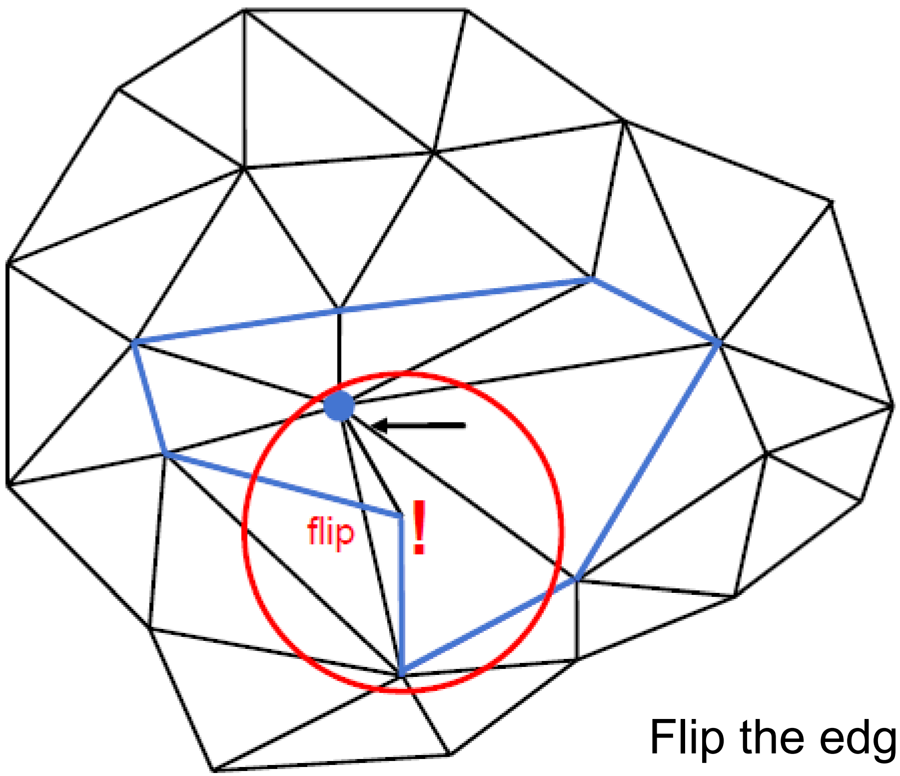
Collapsing operator

- **Half-edge collapse**



Collapsing operator

- **Half-edge collapse**

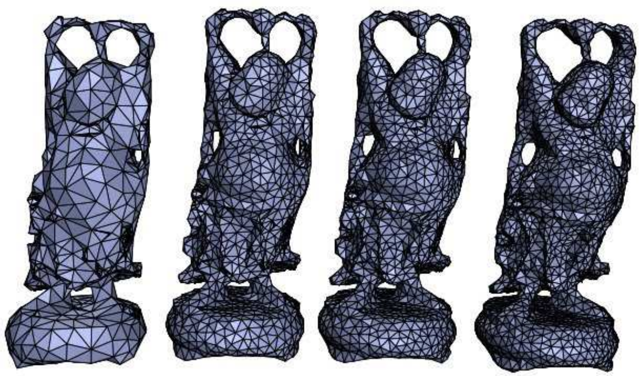


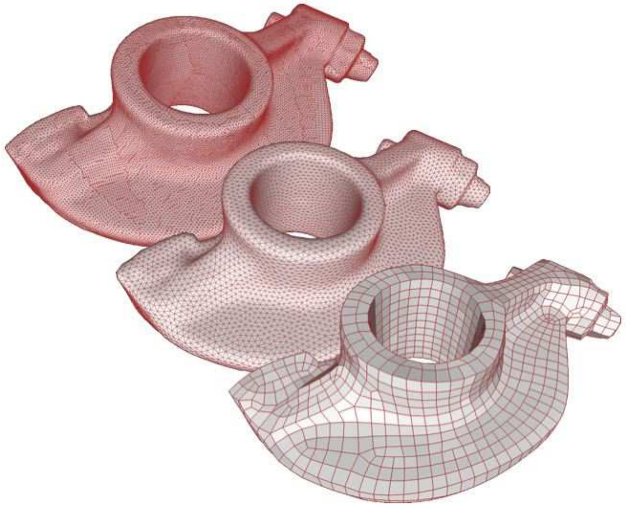
Flip the edge for re-triangulation

3. Level-of-detail and progressive meshes

Level of detail (LoD)

- **Level of detail involves**
  - Decreasing the complexity of a 3D model representation as it moves away from the viewer
  - Level-of-detail techniques increase the efficiency of rendering





Progressive meshes

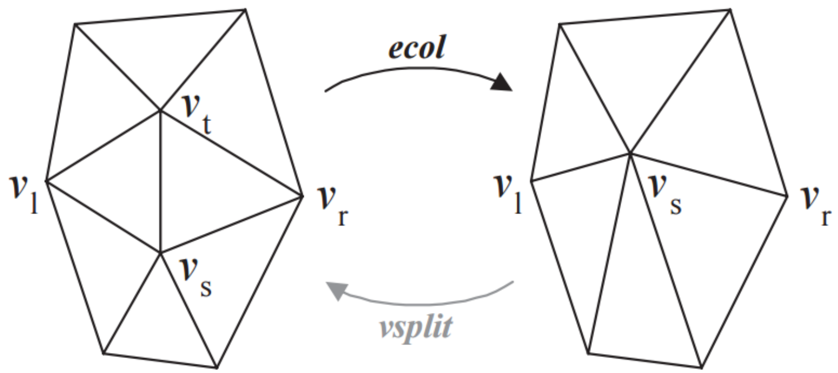
- **Hugues Hoppe**
  - SIGGRAPH 1996
  - Integrated into Direct3D

- **Incorporate geomorph**
  - Allow a smooth choice of detail levels
  - Depending on the smooth view changes

– **Real-time performance**
  - Considerable memory consumption

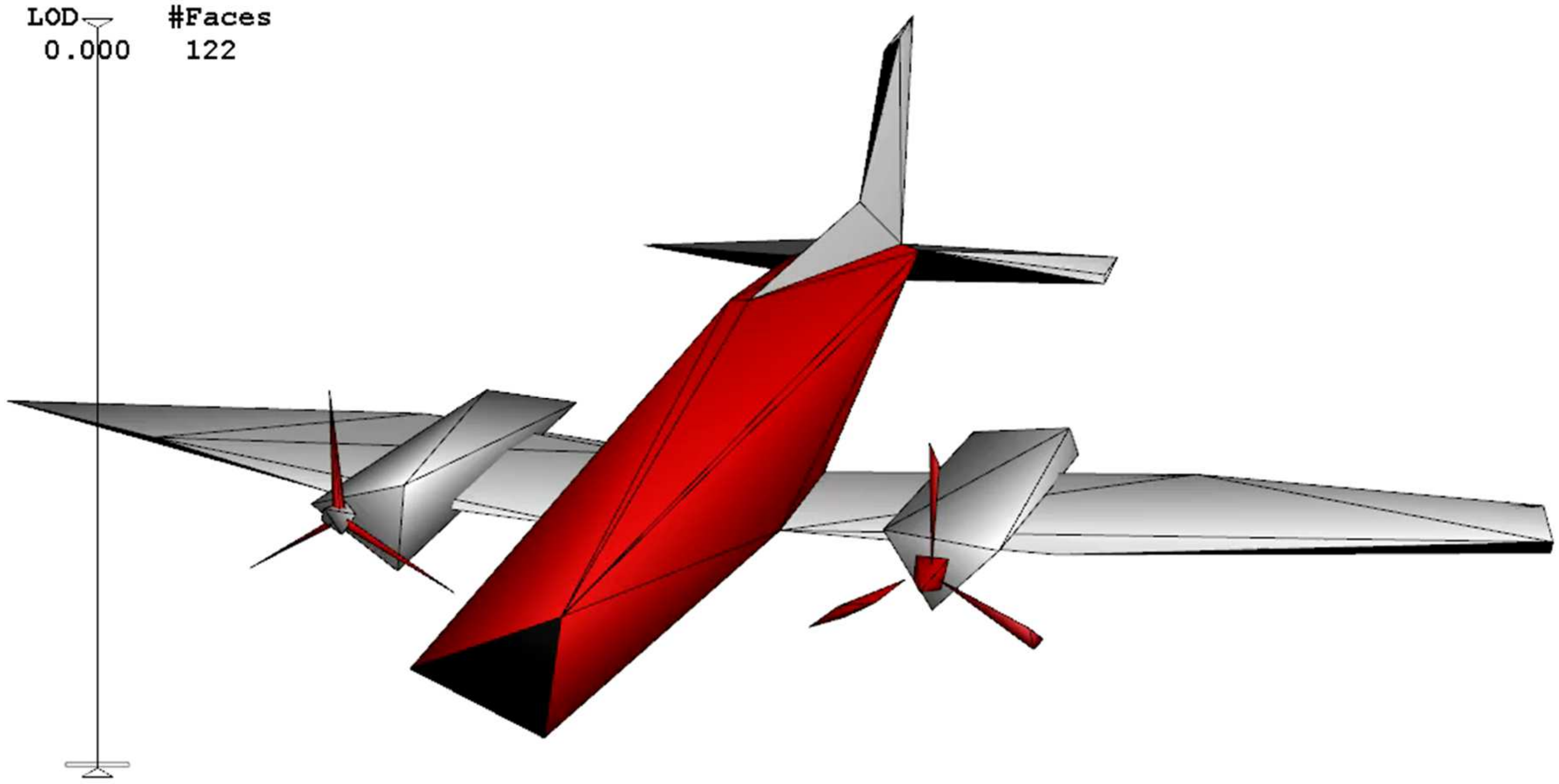
Progressive meshes

- **Edge collapse *ecol***
  - **ecol** takes two connected vertices and replaces them with a single vertex
- **Vertex Split *vsplit***
  - The inverse operation to the edge collapse that divides the vertex into two new vertexes



Progressive meshes

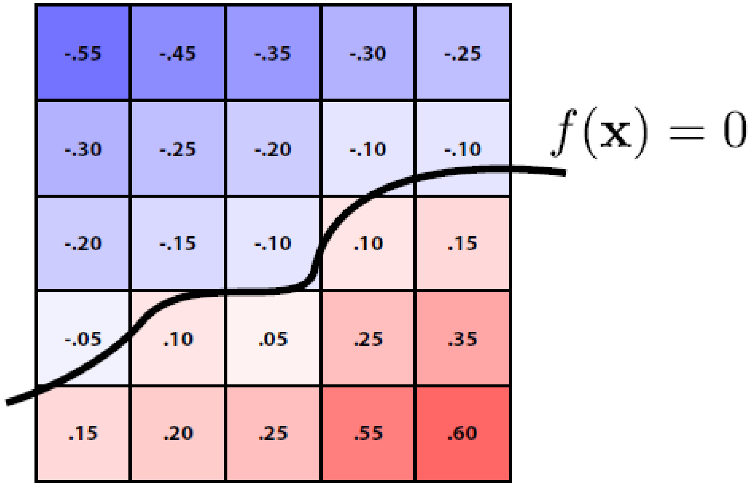
- **Animated Progressive meshes**



3. Isosurface and marching cube algorithm

Implicit representation of a surface

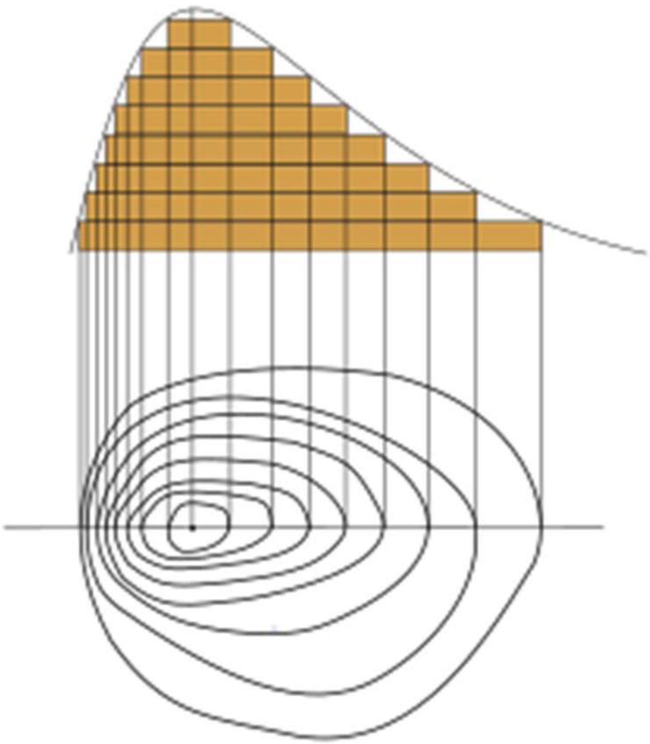
- **Implicit surface representation**
  - Implicit surfaces have some nice features (e.g., merging/splitting)
  - But, hard to describe complex shapes in closed form
  - Alternative: store a grid of values approximating function

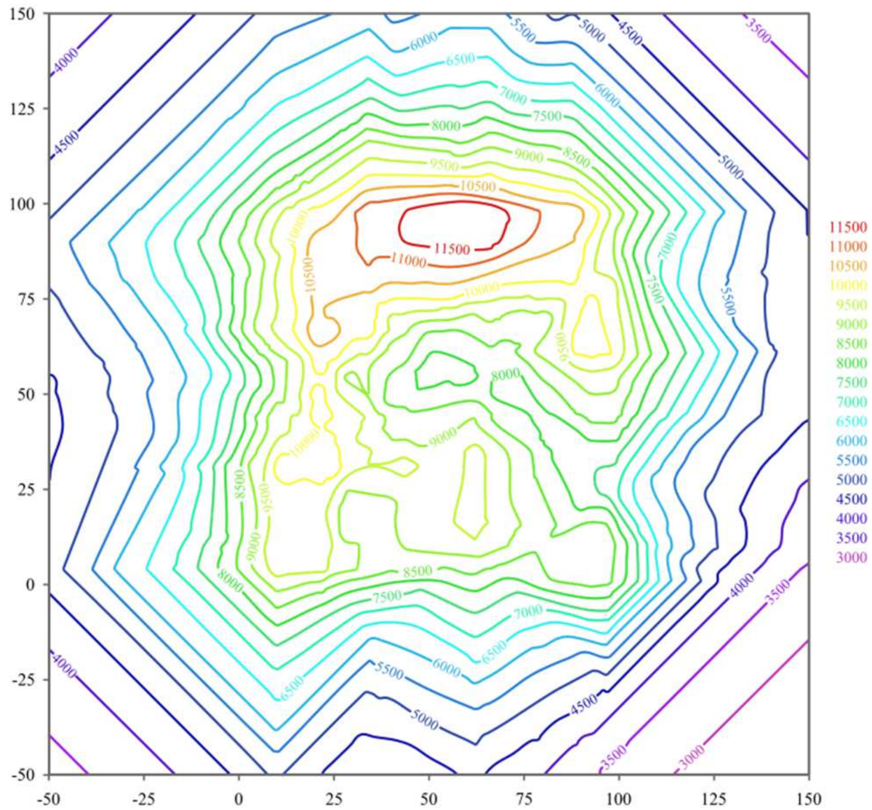


  - Surface is found where *interpolated* values equal zero
  - Provides much more explicit control over shape (like a texture)

Contour line

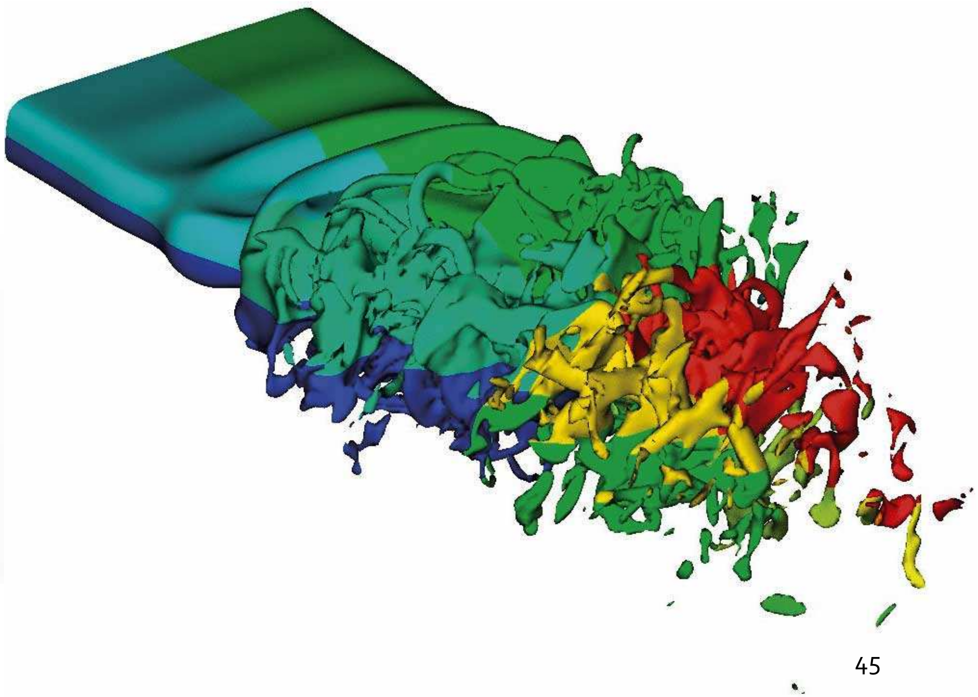
- **A contour line of a function of two variables**
  - A curve along which the function has a constant value





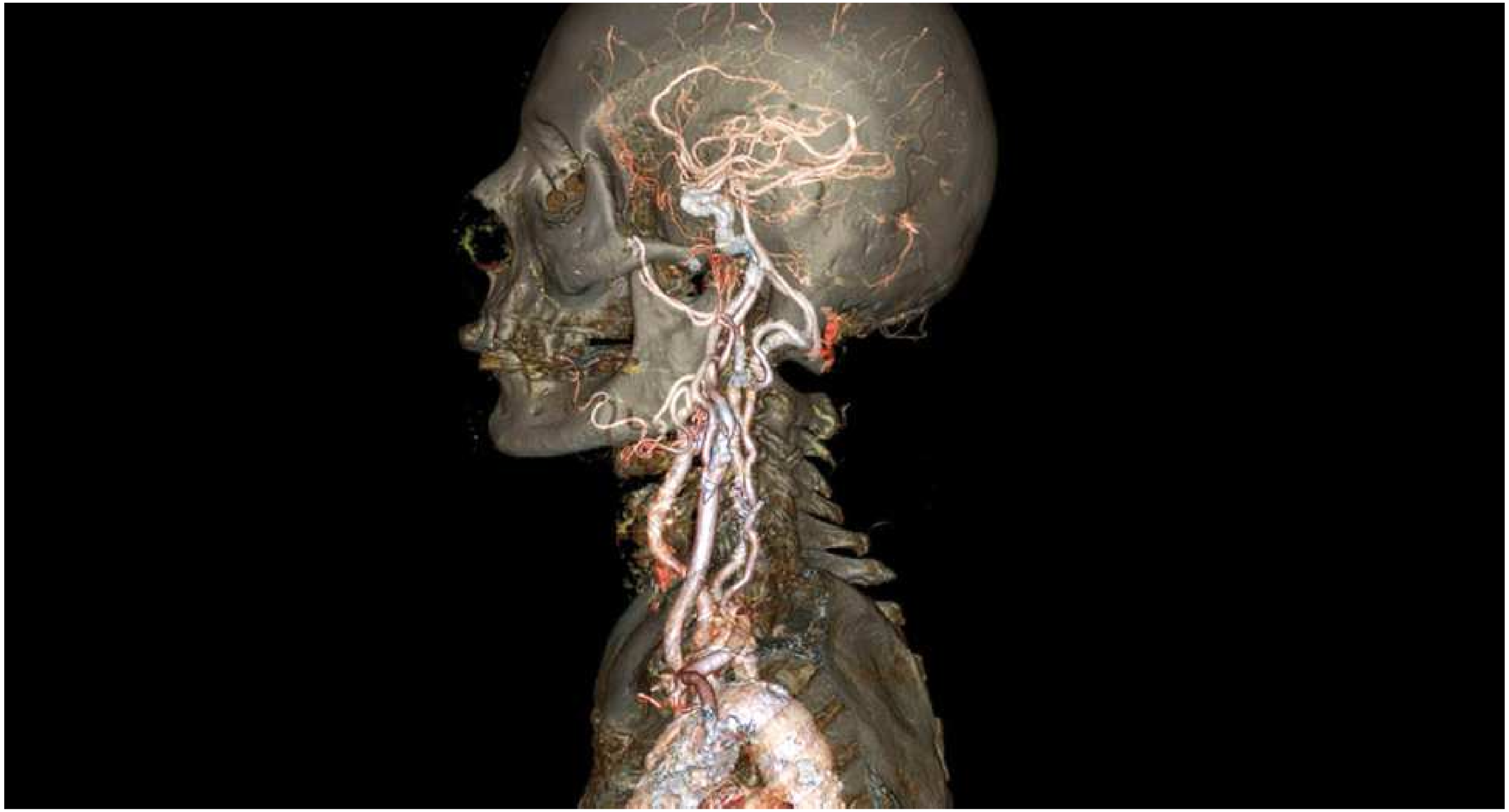
Isosurface

- **A three-dimensional analog of an iso-contour**
  - A surface that represents points of a constant value (iso-value)



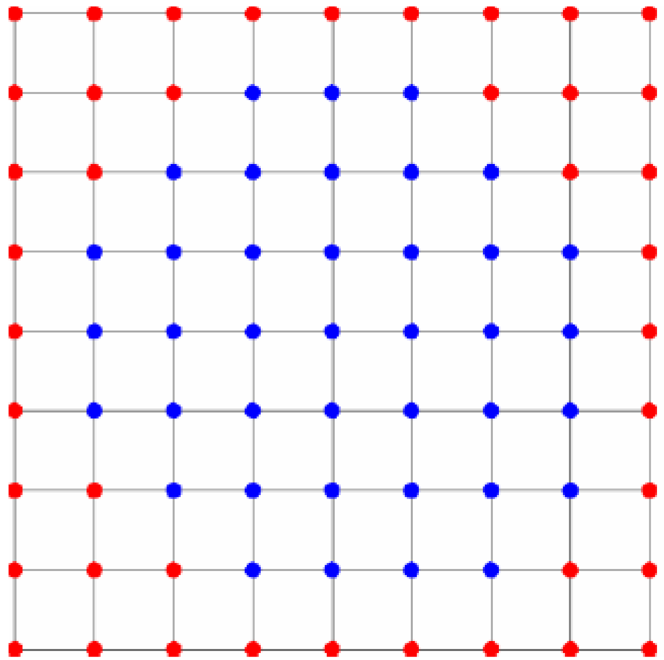
Applications of isosurface

- **Scientific visualization**
  - **For example, medical data visualization**



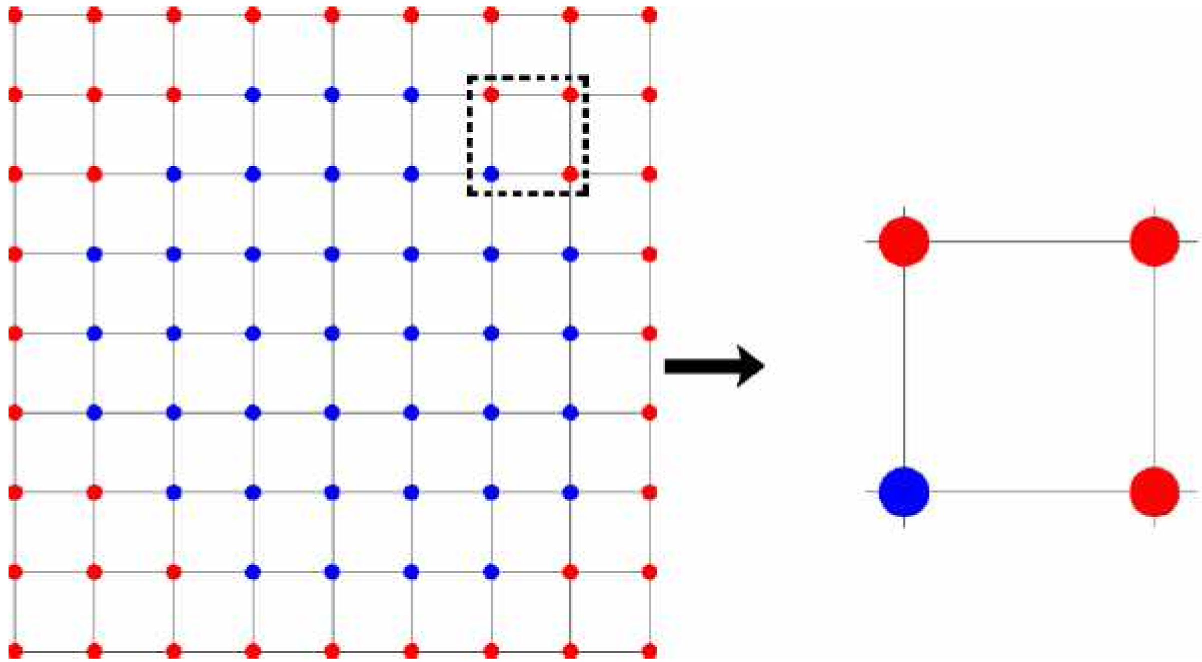
Construction of isosurface

- **Marching cubes**
  - 2D surface reconstruction from samples of a function
  - Sample the function with a uniform grid



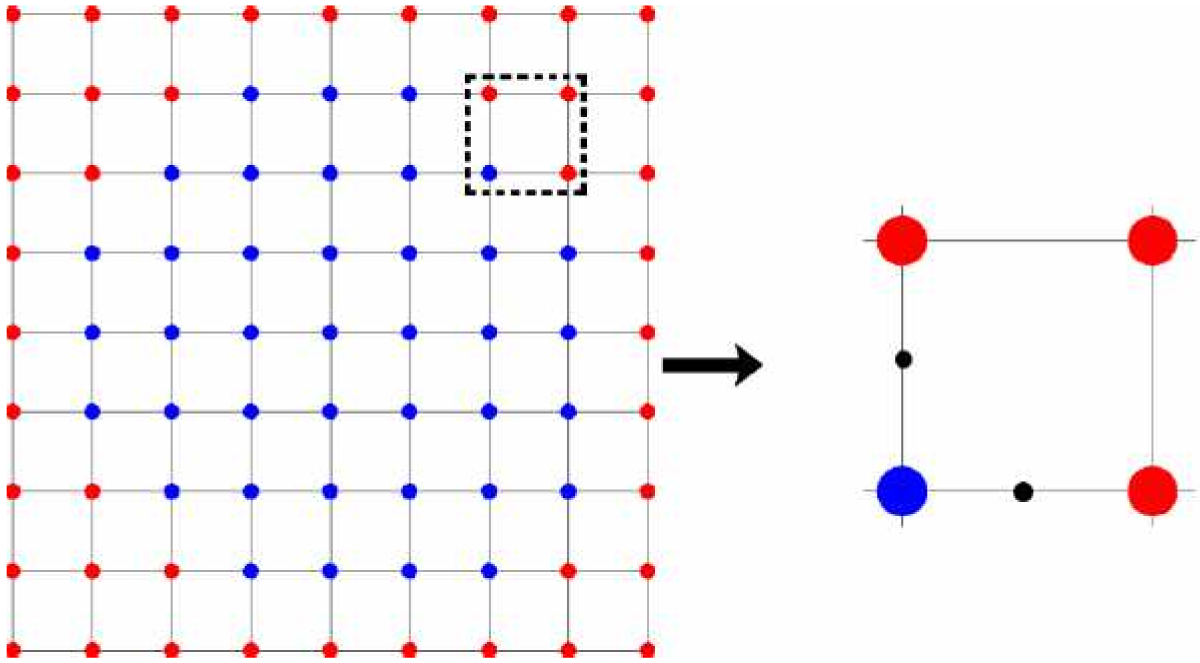
Construction of isosurface

- **Marching cubes**
  - 2D surface reconstruction from samples of a function
  - Sample the function with a uniform grid



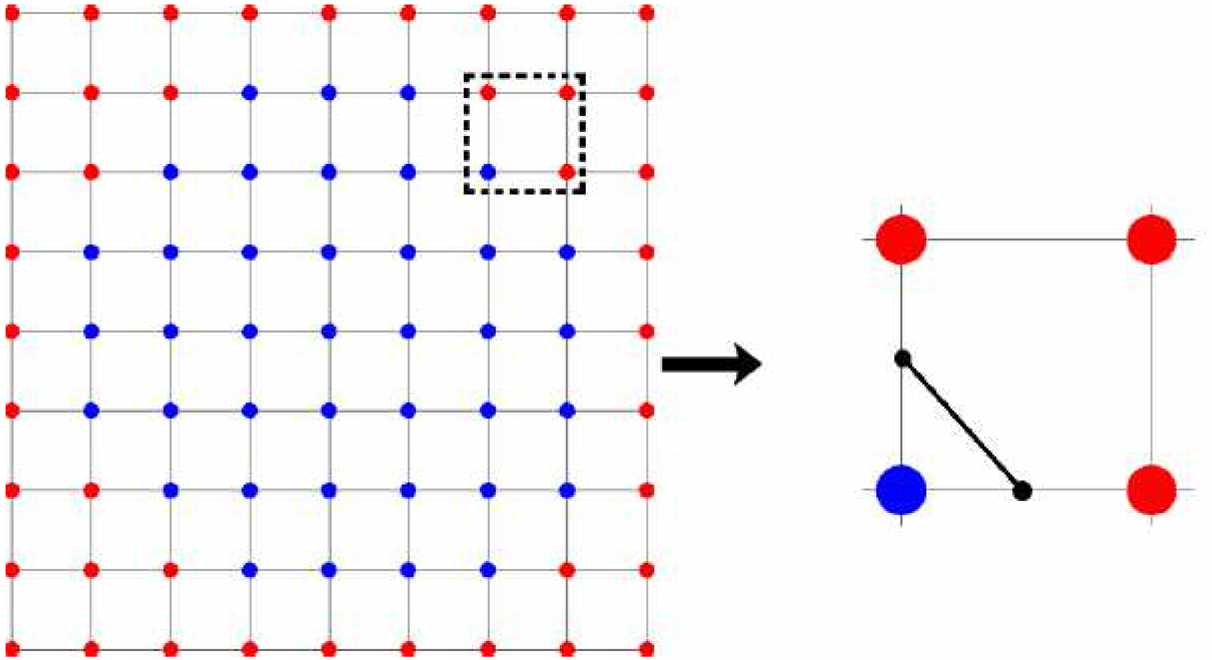
Construction of isosurface

- **Marching cubes**
  - 2D surface reconstruction from samples of a function
  - For each cell, interpolate the values at select the iso-value points



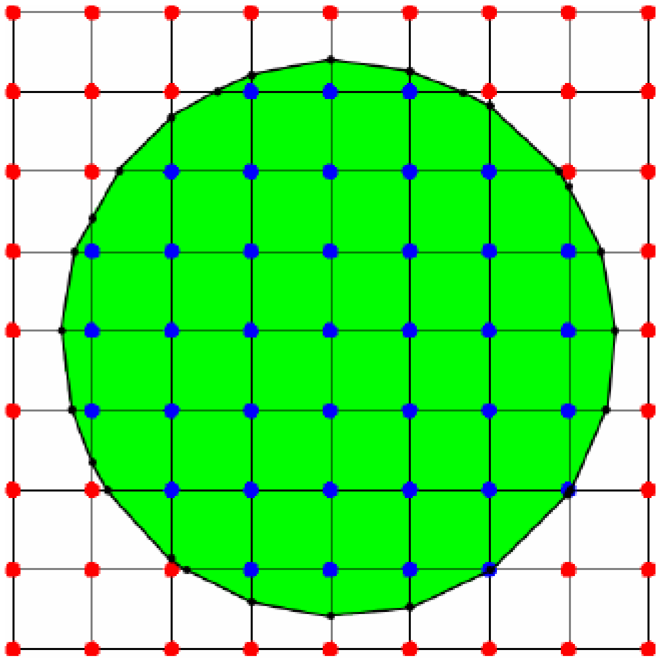
Construction of isosurface

- **Marching cubes**
  - 2D surface reconstruction from samples of a function
  - Then, we connect the two points to form an edge in isosurface



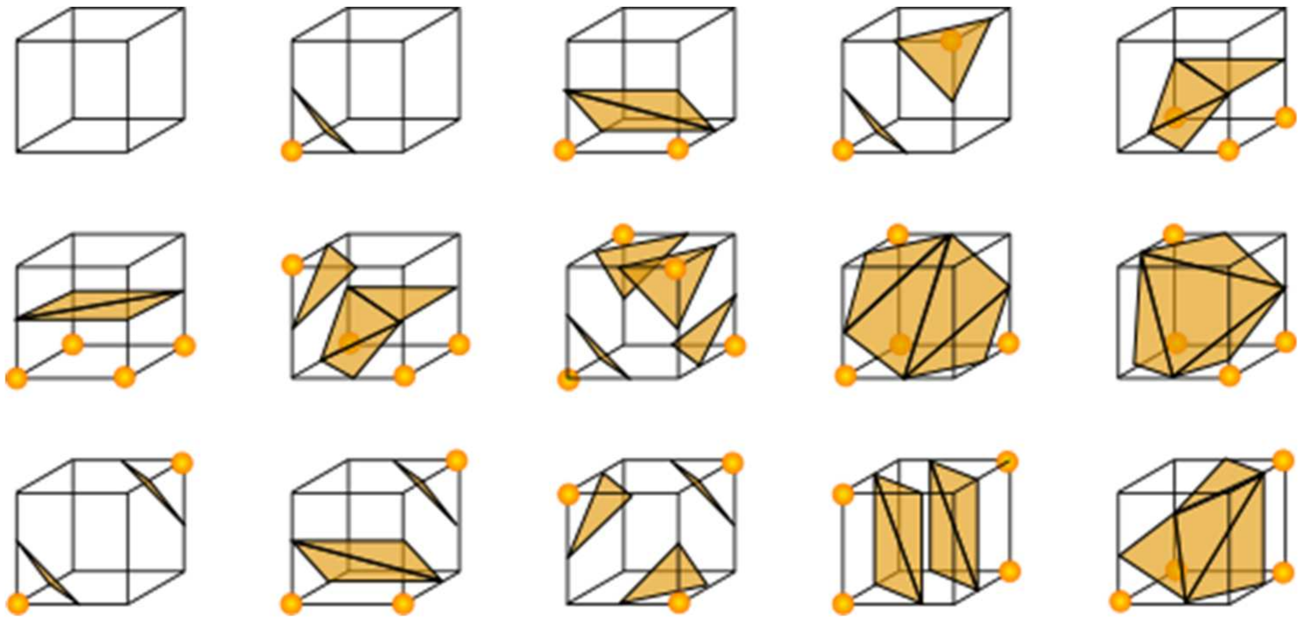
Construction of isosurface

- **Marching cubes**
  - 2D surface reconstruction from samples of a function
  - All the constructed triangle faces in a cell form the whole isosurface



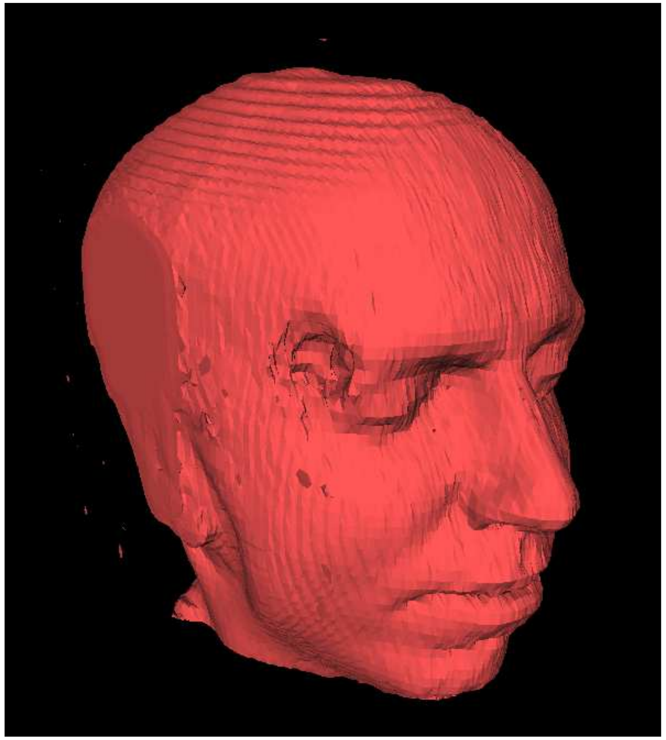
Construction of isosurface

- **Marching cubes**
  - 3D surface reconstruction from samples of a function
  - All the constructed triangle faces in a cell form the whole isosurface



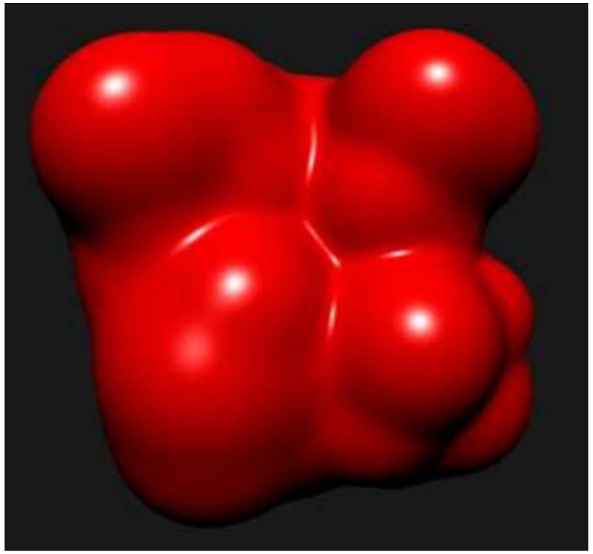
Construction of isosurface

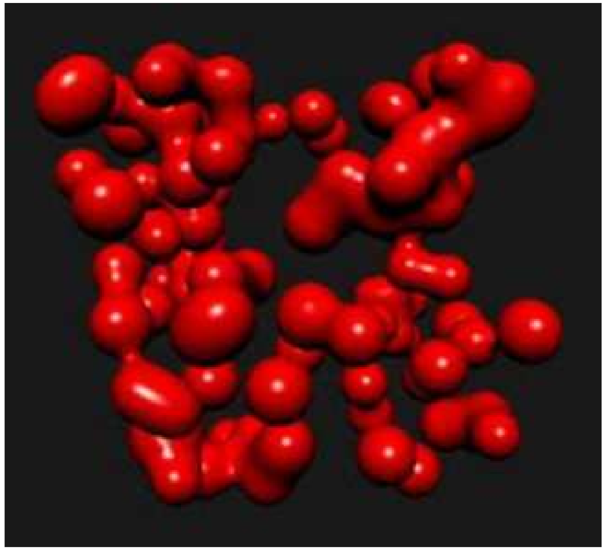
- **Marching cubes**
  - Isosurface construction and rendering with face normal
  - The surface looks non-smooth



Construction of isosurface

- **Marching cubes**
  - How to render smooth surface?
  - Use vertex normal, but how to compute vertex normal?
    - Average from nearby face normals
    - Estimate the gradient of the sampling function

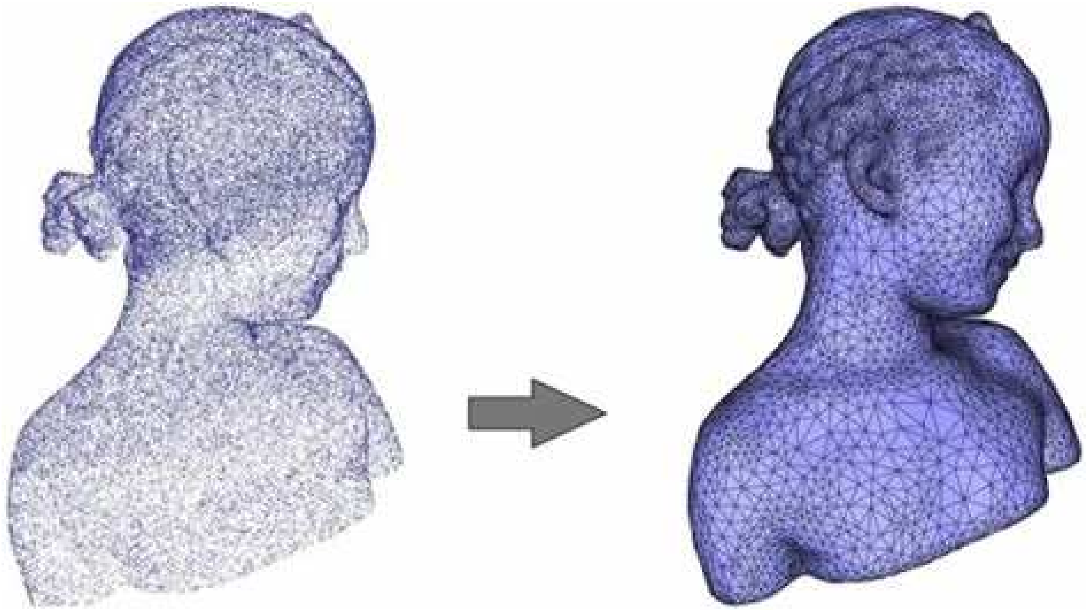




4. Mesh reconstruction from point clouds

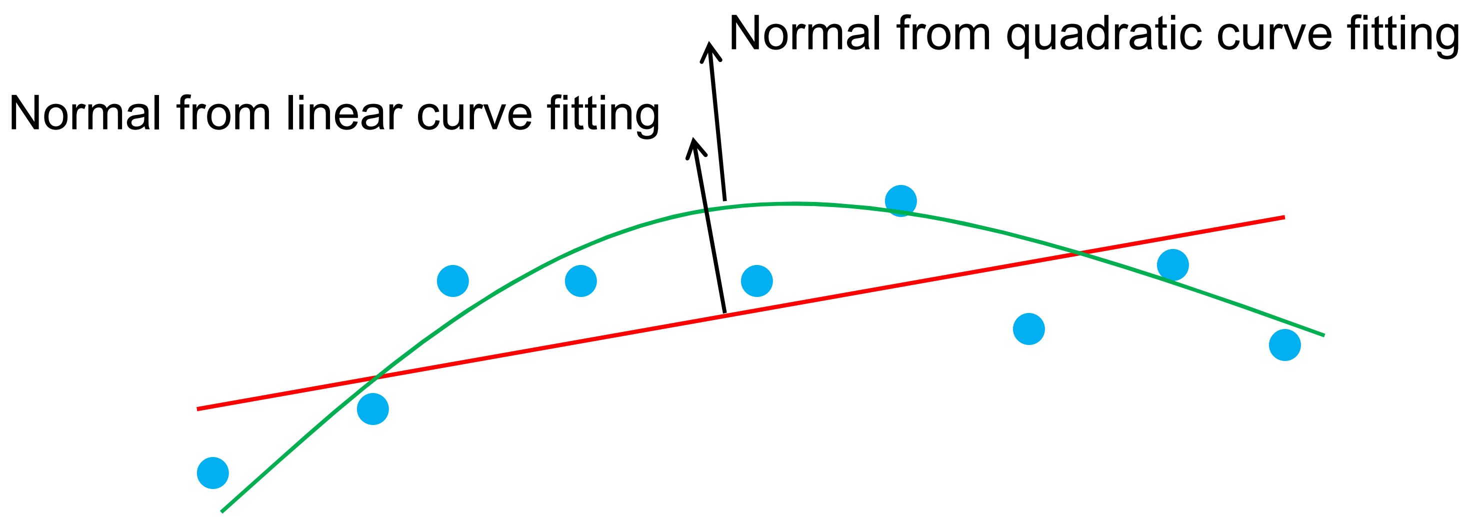
Mesh reconstruction

- **Given a set of points (possibly with normals for each point)**
  - Construct a (triangle) mesh representation that closely fit the points



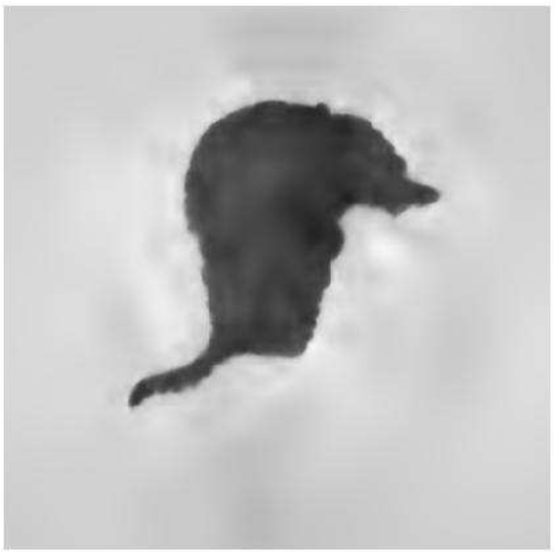
Mesh reconstruction

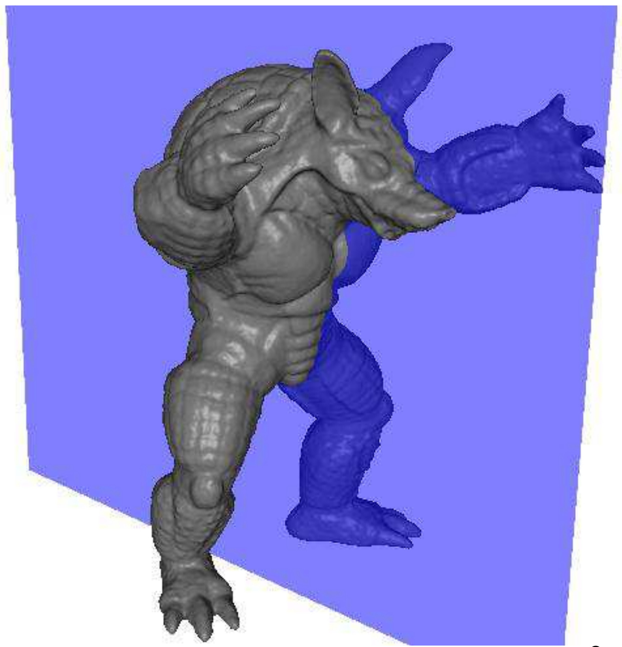
- **What we need?**
  - Point distribution in space
  - Normal for each point
  - If we do not know normal, we can estimate from points



Indicator function

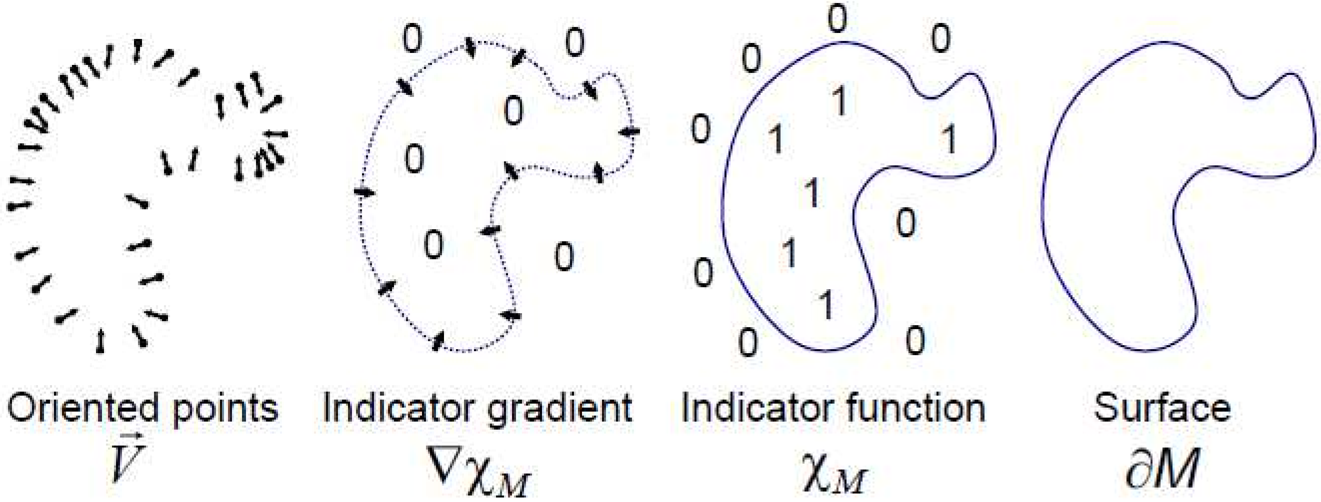
- **A function indicating the inner and outer region of the mesh**
  - The mesh is the isosurface of the indicator function





Mesh reconstruction

- **Poisson mesh reconstruction**
  - We can only specify the normal of the indicator function



  - Problem as a Poisson equation problem

$$\Delta\chi \equiv \nabla \cdot \nabla\chi = \nabla \cdot \vec{V}.$$

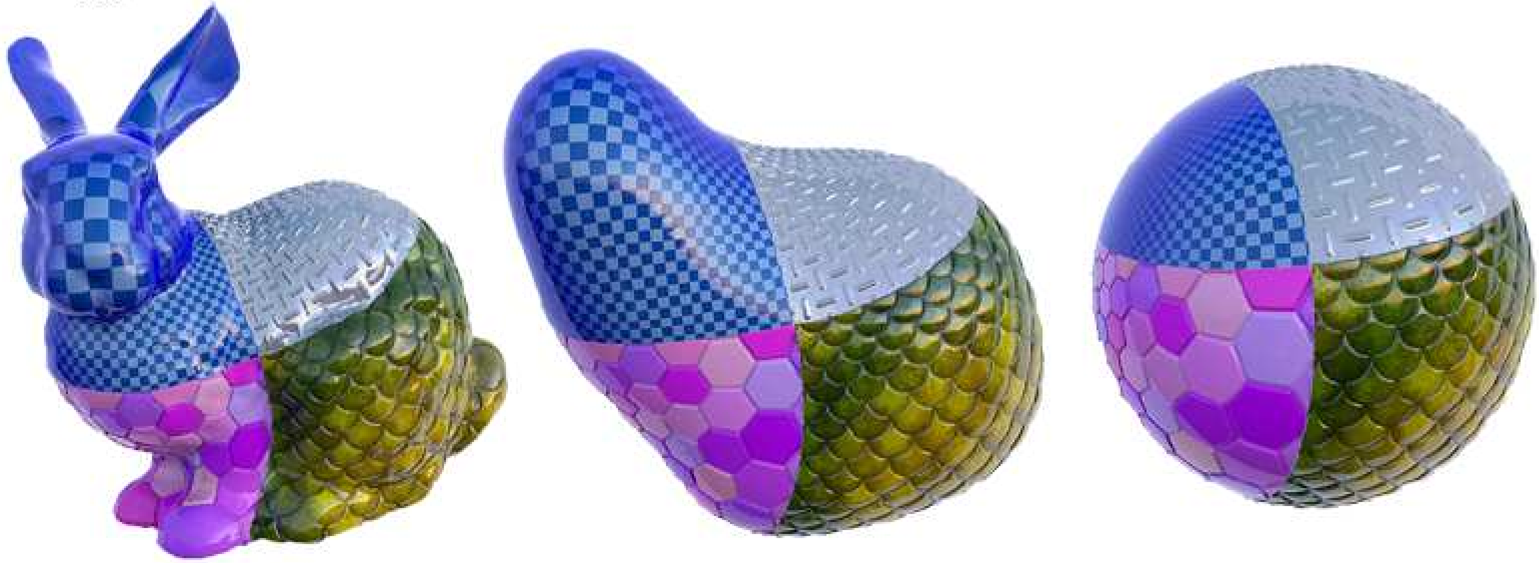
Surface reconstruction as a Poisson problem

- **Poisson mesh reconstruction**
  - After solving the Poisson equation, the surface mesh is the isosurface with an iso-value (>0)
  - Isosurface generation method (marching cubes)

5. Mesh manipulations

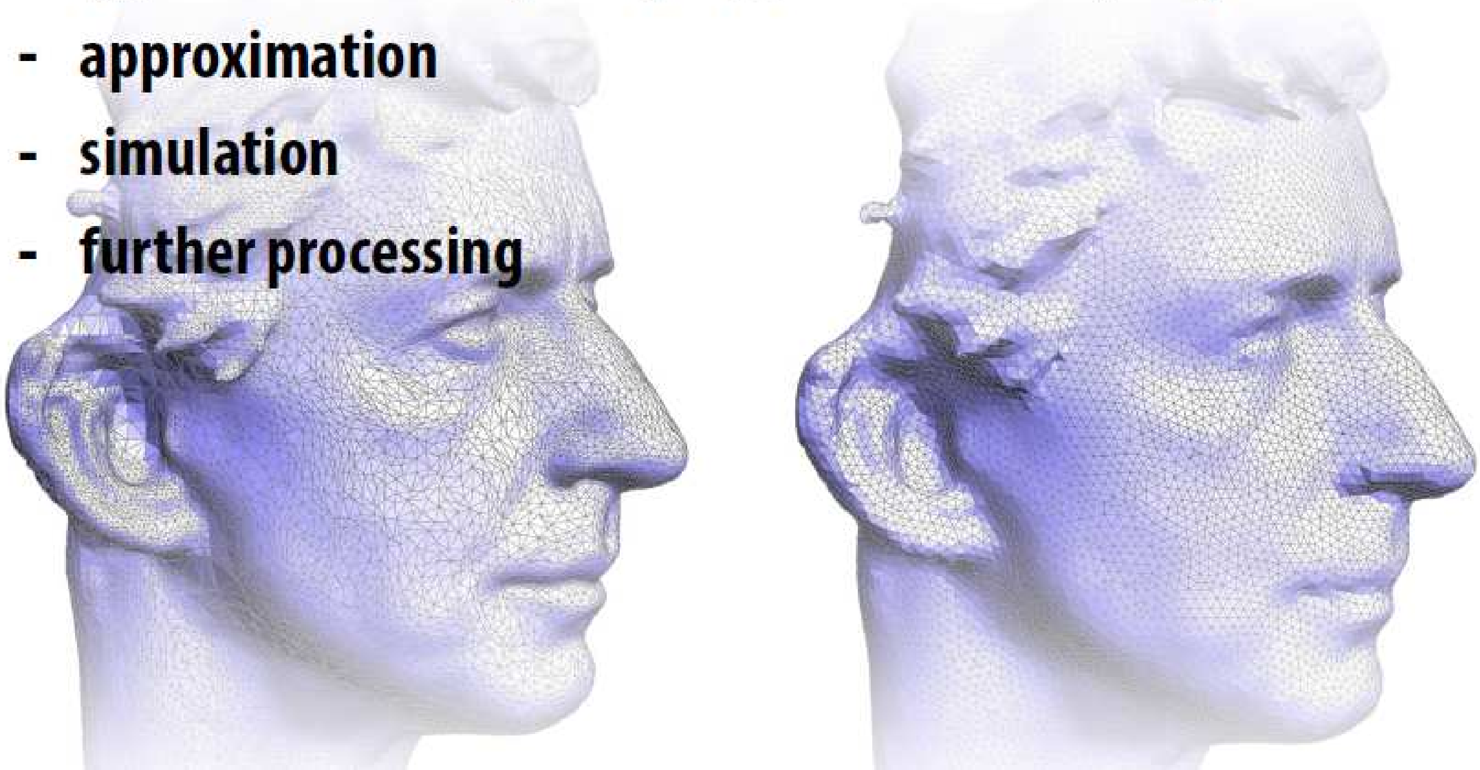
Mesh filtering

- Remove noise, or emphasize important features (e.g., edges)
- Images: blurring, bilateral filter, compressed sensing, ...
- Polygon meshes:
  - curvature flow
  - bilateral filter
  - ...



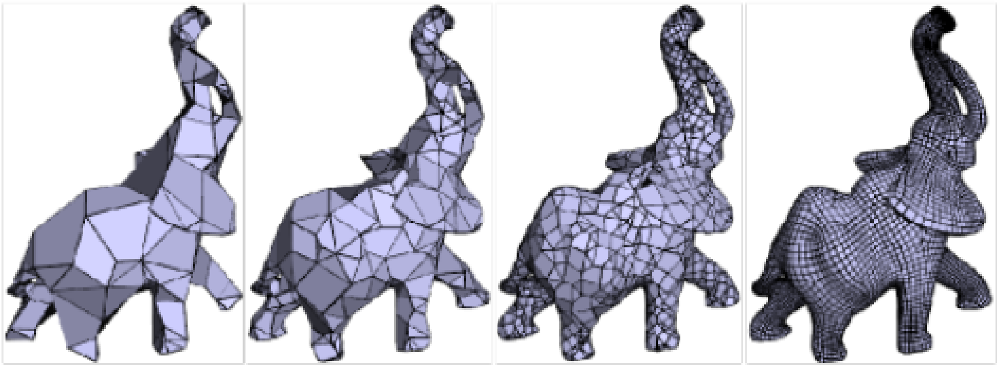
Remeshing

- Modify sample distribution to improve quality
- Images: ...not usually an issue!
  - pixels are always stored on a regular grid
- Polygon meshes: shape of polygons extremely important!
  - approximation
  - simulation
  - further processing



Mesh compression

- Reduce storage size by eliminating redundant data/ approximating unimportant data
- Images:
  - run-length encoding (RLE) - no loss of information
  - spectral/wavelet encoding (e.g., JPEG/MPEG) - lossy
- Polygon meshes:
  - have to compress geometry *and* connectivity
  - many techniques (spectral, diffusion, ...)



Shape analysis

- Identify/understand important semantic features
- Images: computer vision, segmentation, face detection, ...
- Polygon meshes:
  - segmentation
  - correspondence
  - symmetry detection
  - ...

Next lecture: Rendering geometries