From Point Clouds to tessellated surfaces *explicit methods*

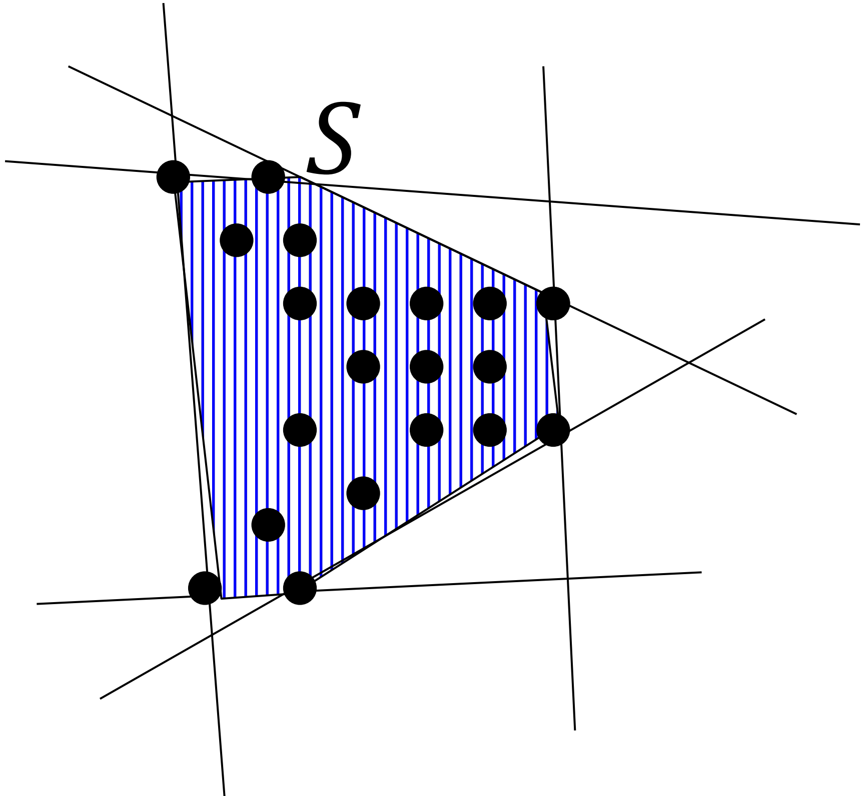
Paolo Cignoni,
Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche

Alpha Shapes [Edelsbrunner83]

Convex Hull

$$CH(S) = \mathbb{R}^d \setminus \bigcup EH(S)$$

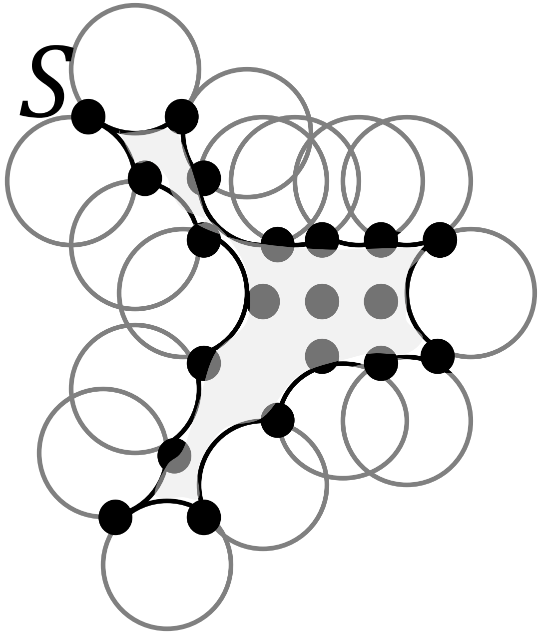
$EH(S)$: halfspace not containing any point in S

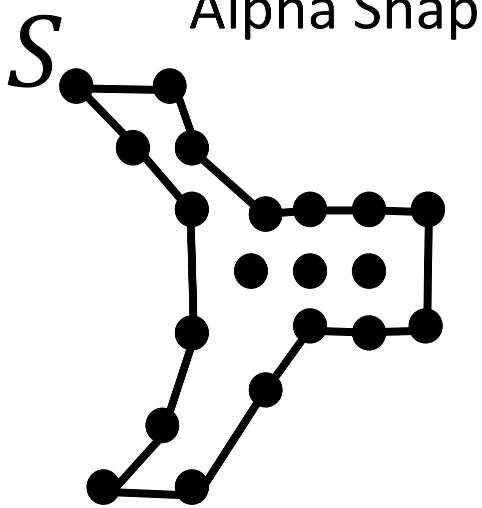


Alpha Hull

$$\alpha H(S) = \mathbb{R}^d \setminus \bigcup EB_\alpha(S)$$

EB_α(S): ball with radius α not containing any point in S





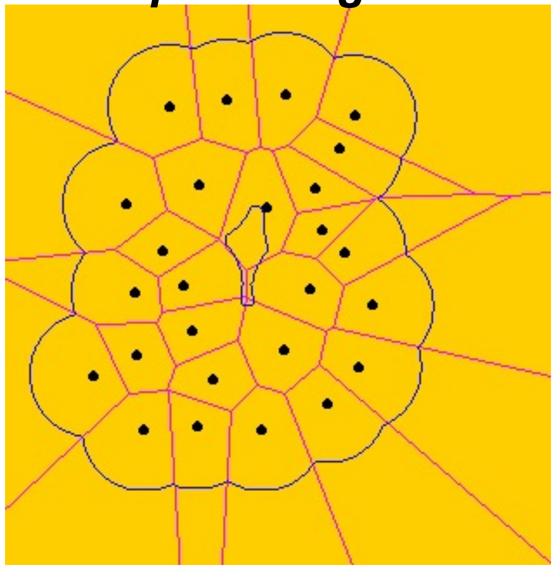
Computing Alpha Shapes

- Alpha Diagram: Voronoi Diagram restricted to space closest than α to one point in S
- Alpha Complex: Subset of Delaunay Triangulation computed as the dual of the alpha diagram

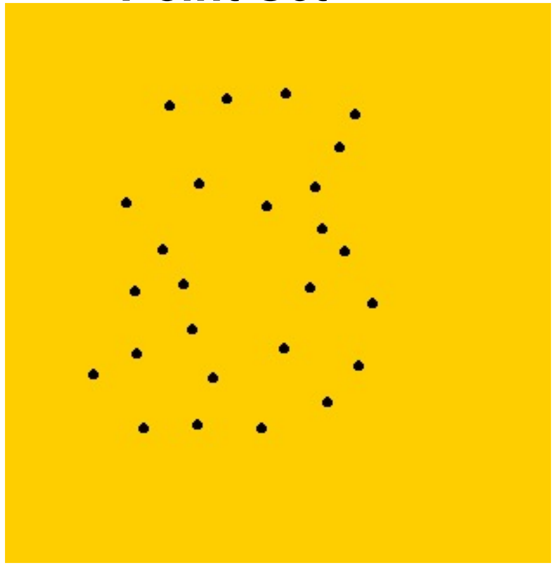
Computing Alpha Shapes

- Alpha Diagram: Voronoi Diagram restricted to space closest than α to one point in S
- Alpha Complex: Subset of Delaunay Triangulation computed as the dual of the alpha diagram

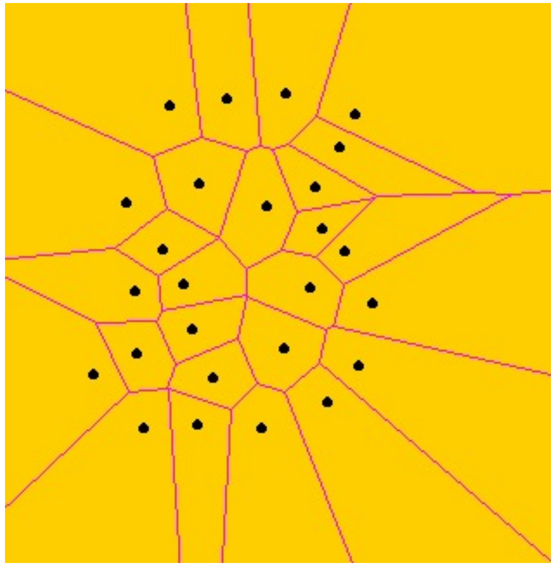
Alpha Diagram



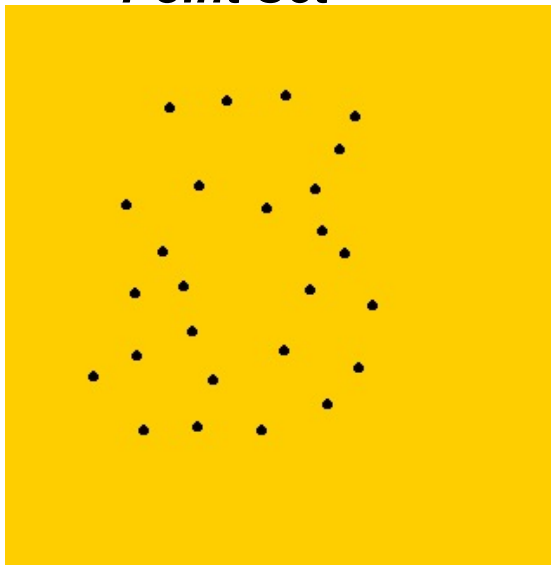
Point Set



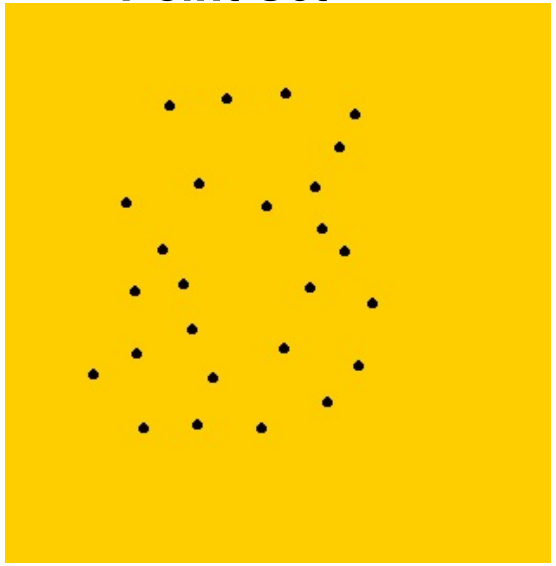
Voronoi Diagram



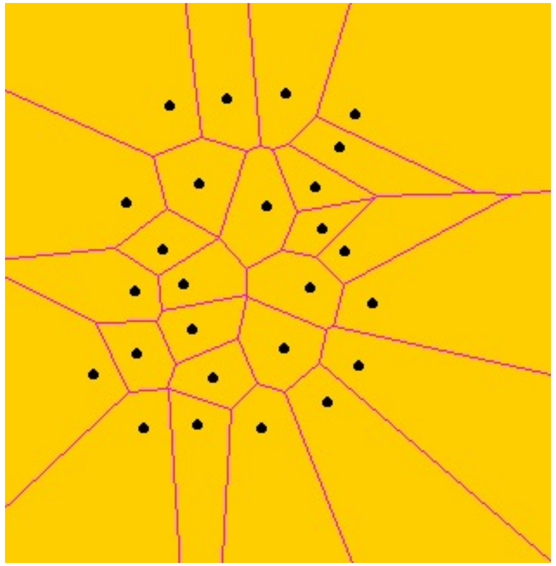
Point Set



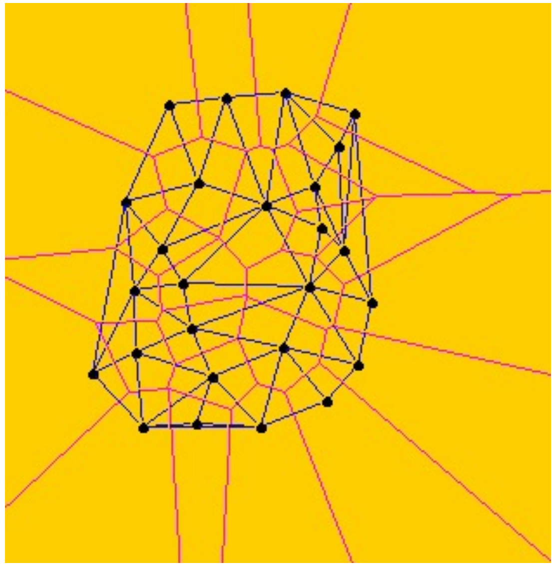
Point Set



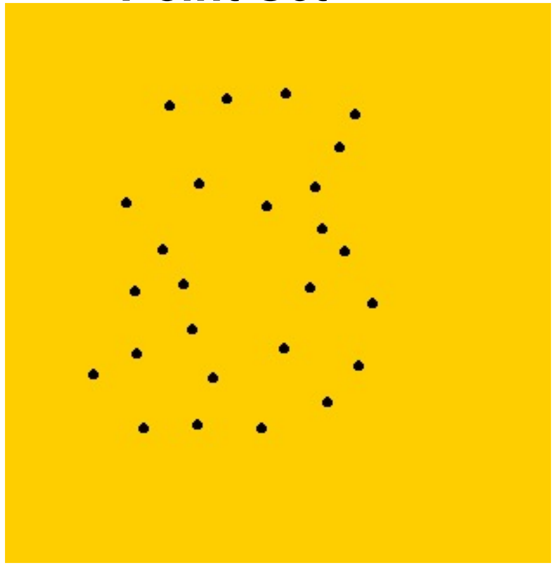
Voronoi Diagram



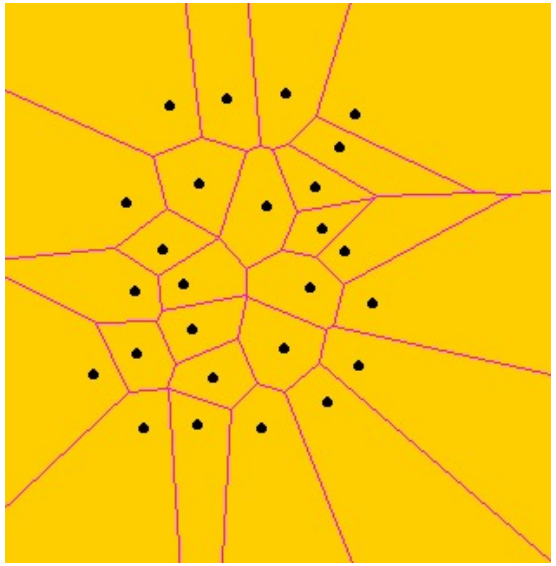
Delaunay Triangulation



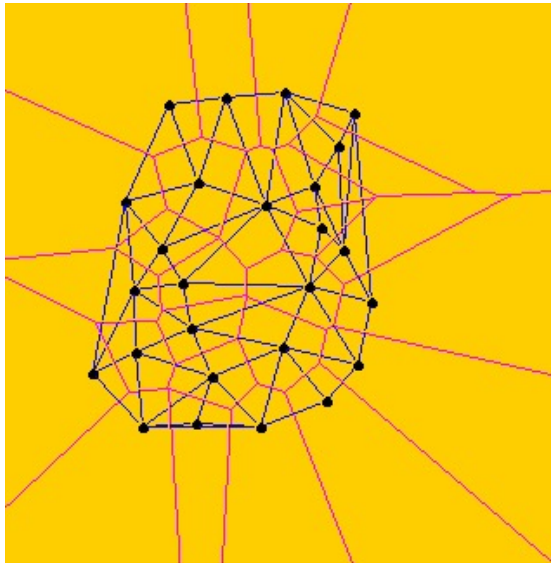
Point Set



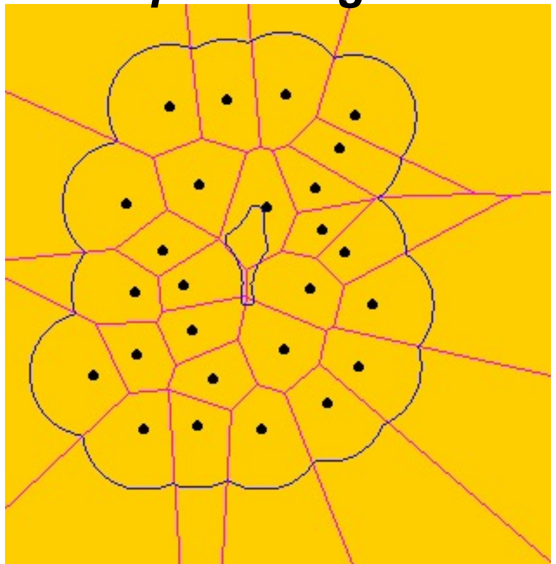
Voronoi Diagram



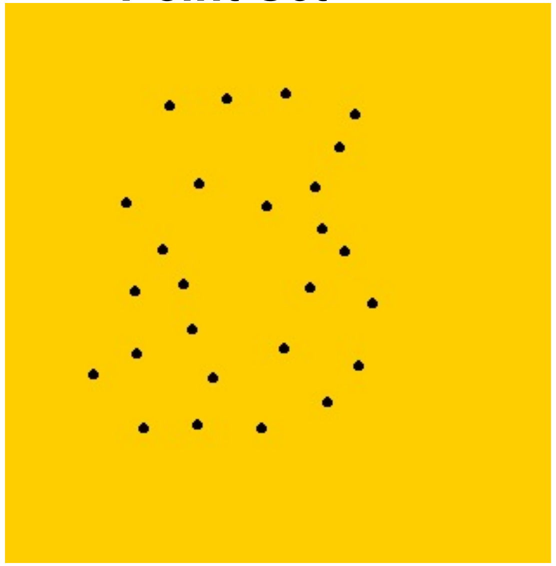
Delaunay Triangulation



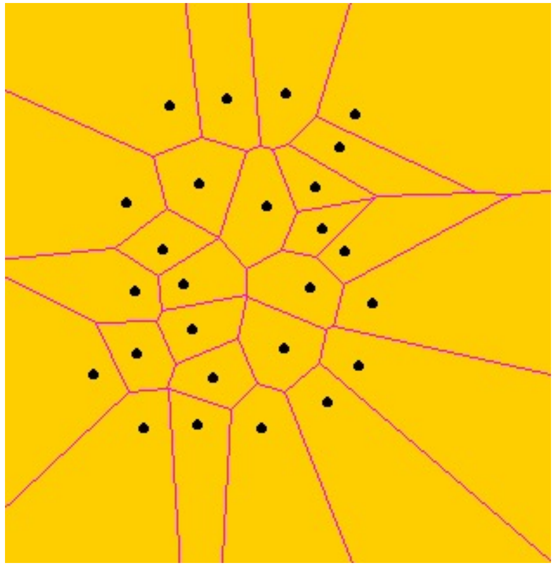
Alpha Diagram



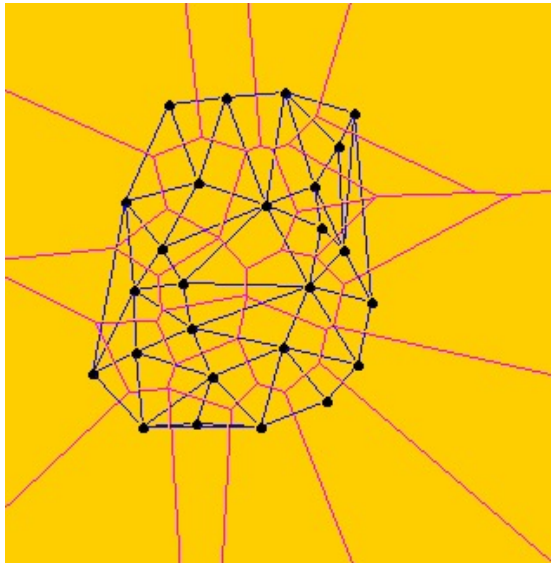
Point Set



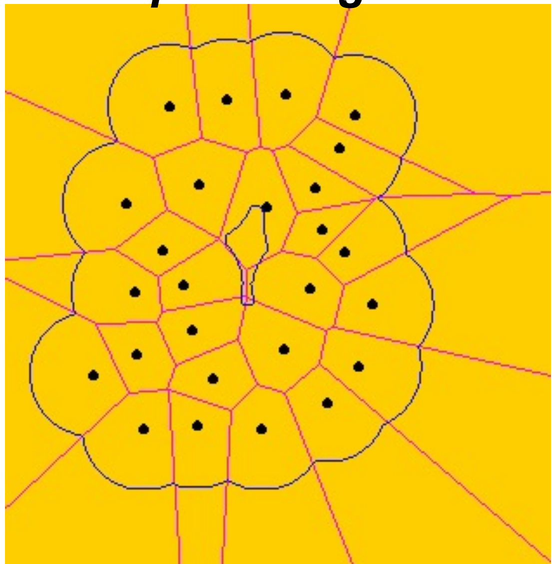
Voronoi Diagram



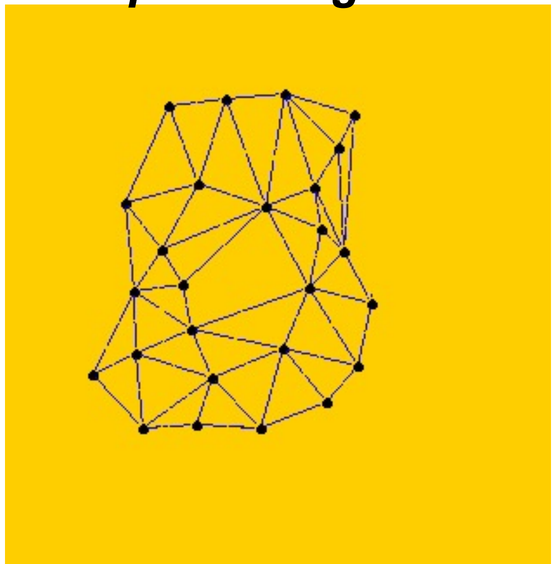
Delaunay Triangulation



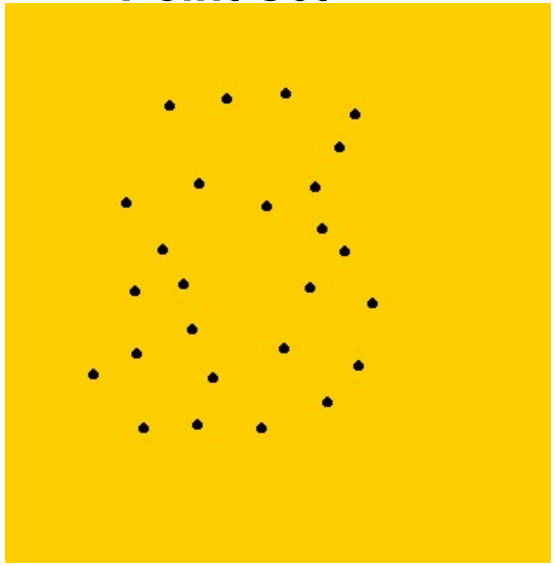
Alpha Diagram



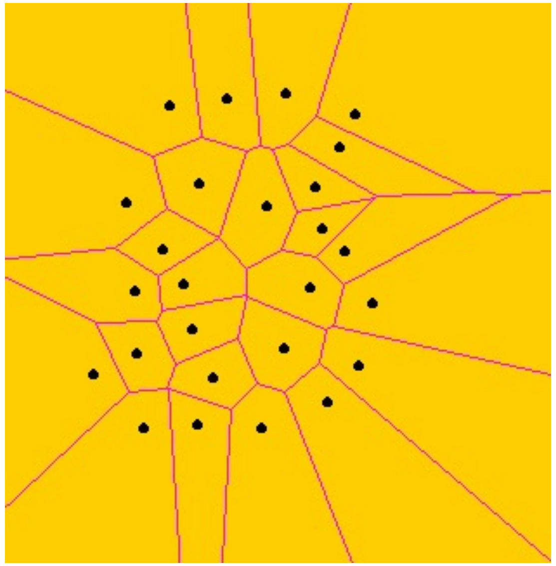
Alpha triangulation



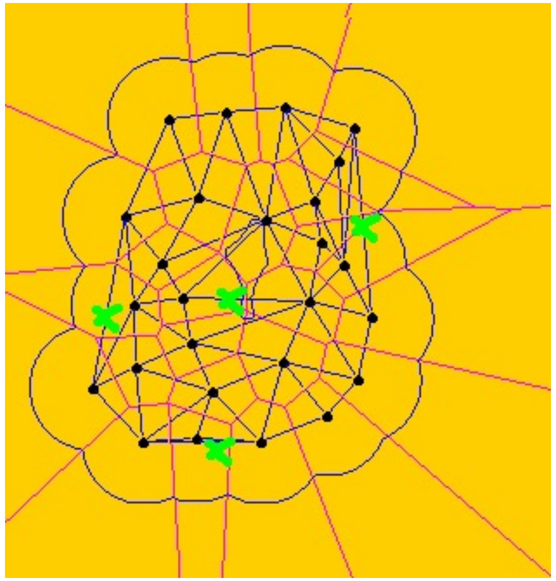
Point Set



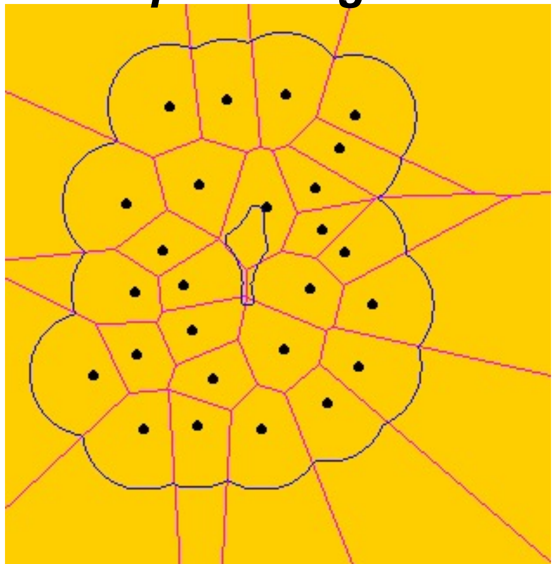
Voronoi Diagram



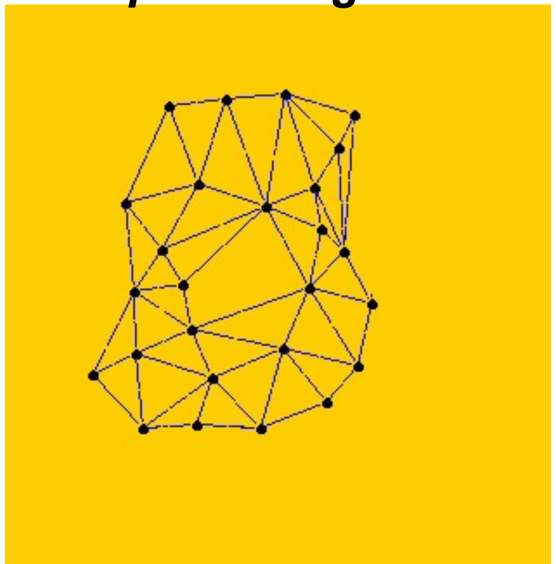
Delaunay Triangulation



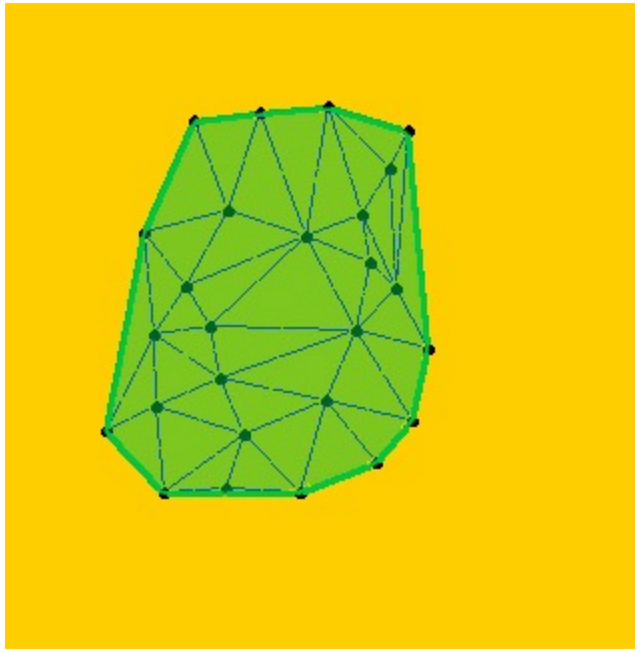
Alpha Diagram



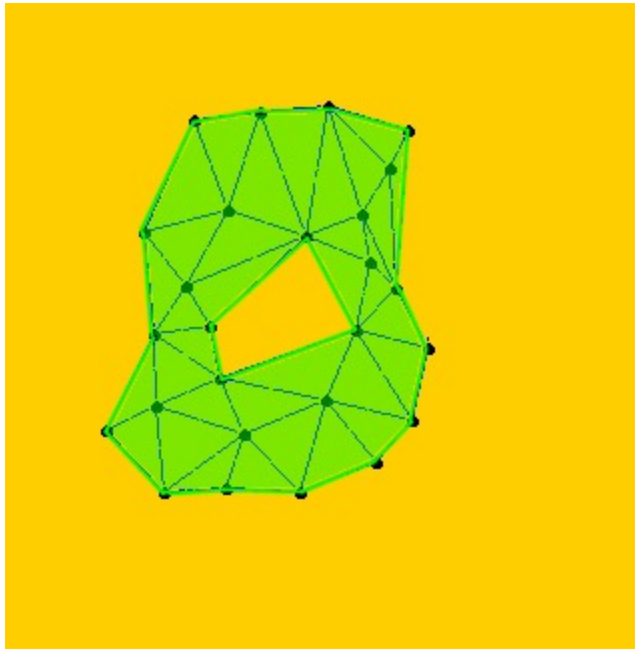
Alpha triangulation



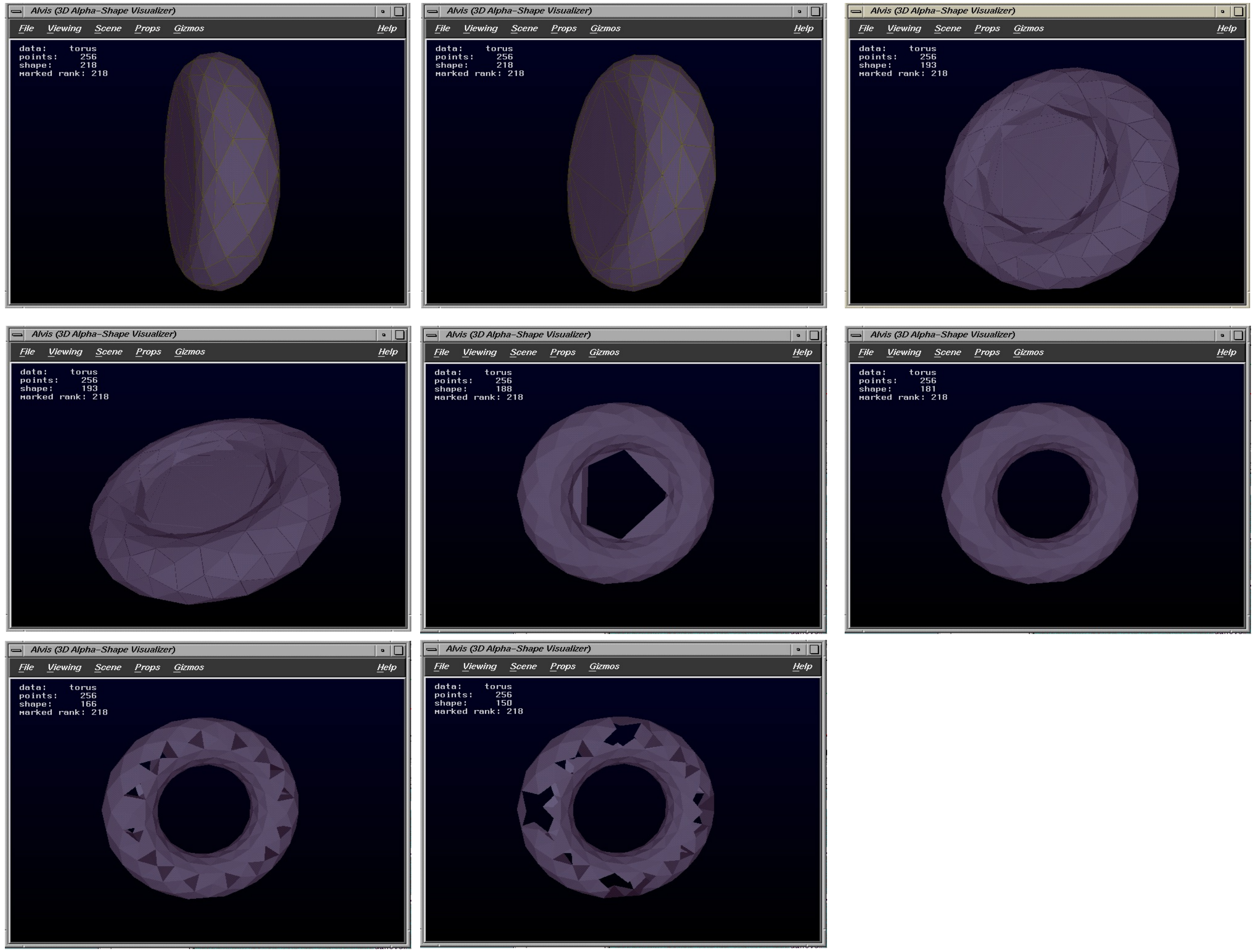
Delaunay triangulation



Alpha Complex



- $\alpha = 0$ then α -shape is the point set
- $\alpha \to \infty$ α -shape tends to the convex hull
- A finite number of thresholds $\alpha_0 < \alpha_1 < \ldots < \alpha_n$ defines all possible shapes (at most $2n^2 - 5n$)


Alvis (3D Alpha-Shape Visualizer)
File Viewing Scene Props Gizmos Help
data: torus
points: 256
shape: 218
marked rank: 218
Alvis (3D Alpha-Shape Visualizer)
File Viewing Scene Props Gizmos Help
data: torus
points: 256
shape: 218
marked rank: 218
Alvis (3D Alpha-Shape Visualizer)
File Viewing Scene Props Gizmos Help
data: torus
points: 256
shape: 193
marked rank: 218
Alvis (3D Alpha-Shape Visualizer)
File Viewing Scene Props Gizmos Help
data: torus
points: 256
shape: 193
marked rank: 218
Alvis (3D Alpha-Shape Visualizer)
File Viewing Scene Props Gizmos Help
data: torus
points: 256
shape: 188
marked rank: 218
Alvis (3D Alpha-Shape Visualizer)
File Viewing Scene Props Gizmos Help
data: torus
points: 256
shape: 181
marked rank: 218
Alvis (3D Alpha-Shape Visualizer)
File Viewing Scene Props Gizmos Help
data: torus
points: 256
shape: 166
marked rank: 218
Alvis (3D Alpha-Shape Visualizer)
File Viewing Scene Props Gizmos Help
data: torus
points: 256
shape: 150
marked rank: 218

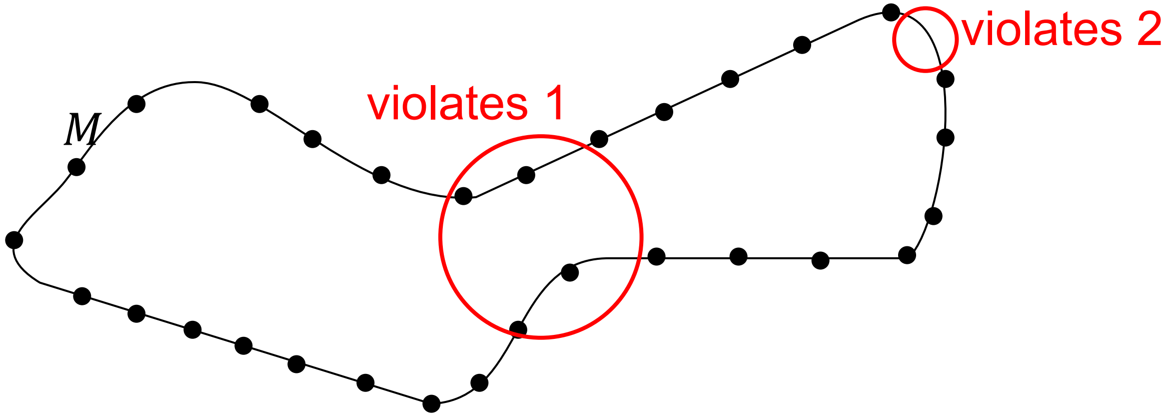
Sampling Conditions for Alpha Shapes

Proposition

Given a smooth manifold M and a sampling S, if it holds that

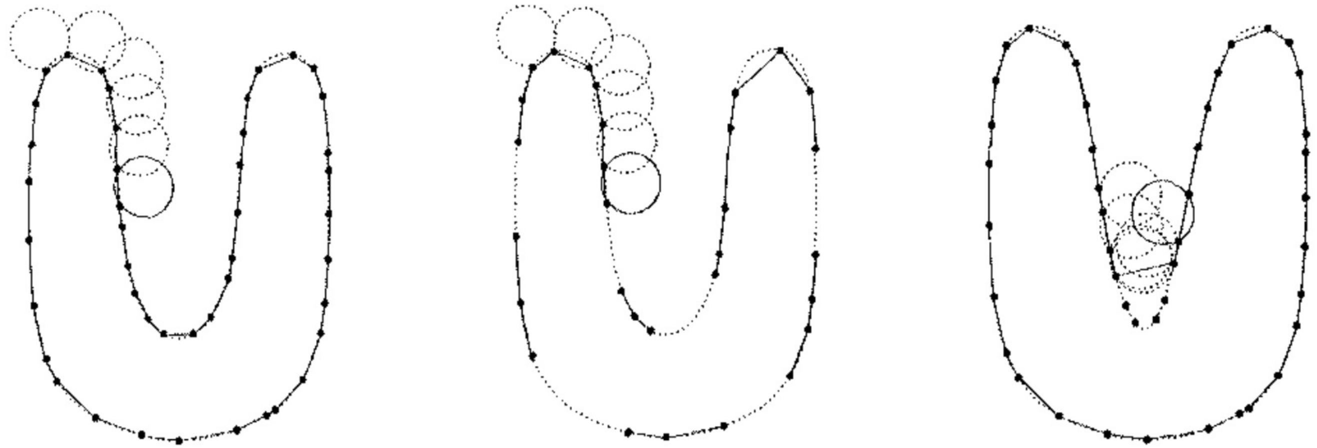
1. the intersection of any ball of radius α with M is homeomorphic to a disk
2. Any ball of radius α centered in the manifold contains at least one point of S

Then the α-shape of S is homeomorphic to M



Ball Pivoting [bernardini99]

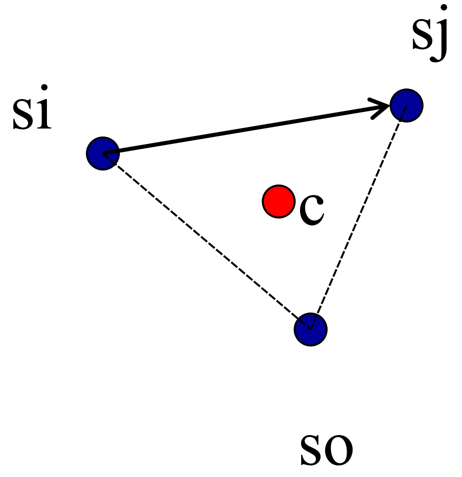
- Motivations
  - Alpha shapes computation is fairly cumbersome
  - May produce non manifold surfaces
- Core idea: approximate the alpha shapes just «rolling» a ball of radius α on the sampling S
- Same sampling conditions as α –shape holds



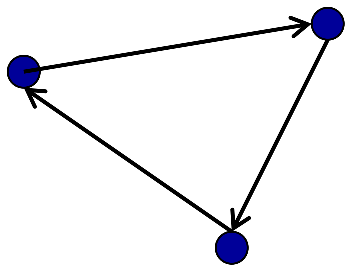
OK | Low sampling density | Curvature grater than $\frac{1}{\alpha}$

The algorithm

- Edge (si, sj)
  - –Opposite point so, center of empty ball c
  - –Edge: “Active”, “Boundary”



Pivoting example

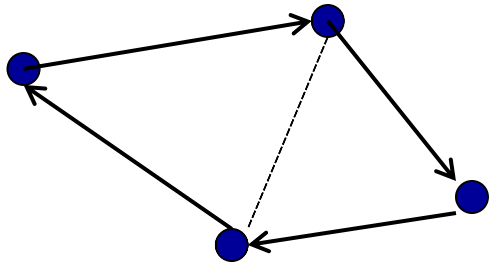


Initial seed triangle:
Empty ball of radius ρ passes through the three points

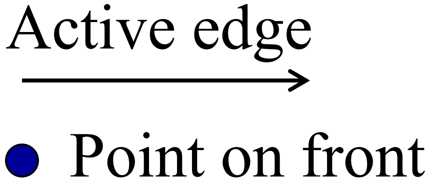
Active edge

Point on front

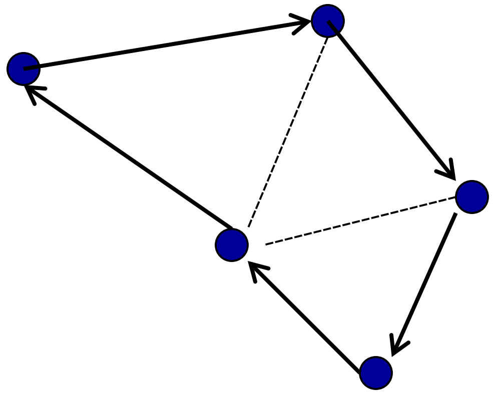
Pivoting example



Ball pivoting around active edge



Pivoting example

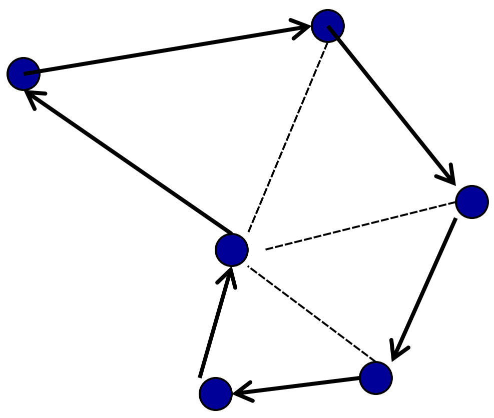


Ball pivoting around active edge

Active edge

Point on front

Pivoting example

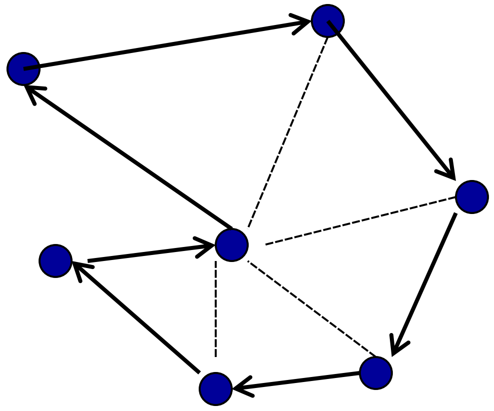


Ball pivoting around active edge

Active edge

Point on front

Pivoting example

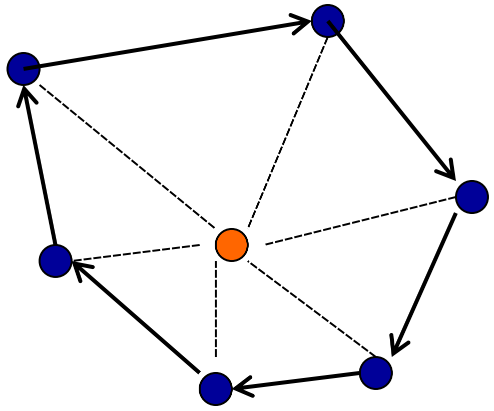


Ball pivoting around active edge

Active edge

Point on front

Pivoting example



Ball pivoting around active edge

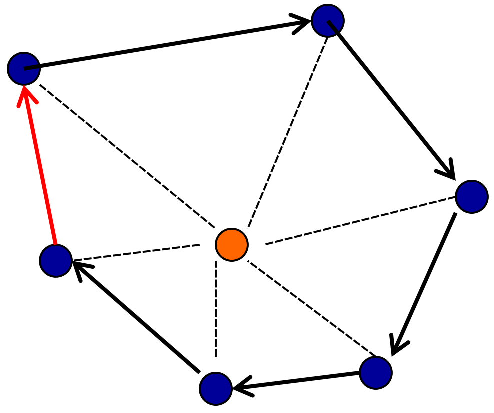
Active edge

Point on front

Internal point

Pivoting example

Boundary edge



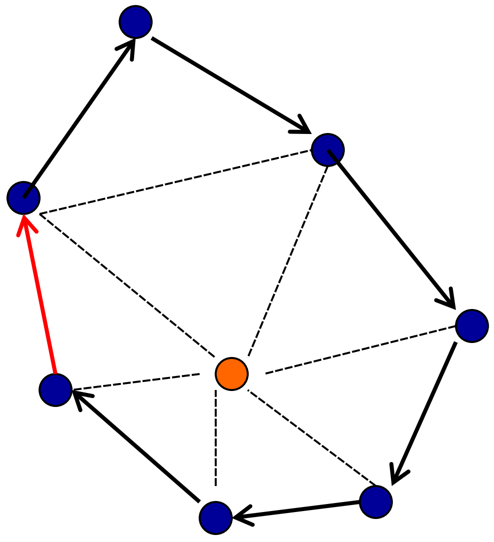
Ball pivoting around active edge
No pivot found

Active edge

Point on front
Internal point

Pivoting example

Boundary edge



Active edge

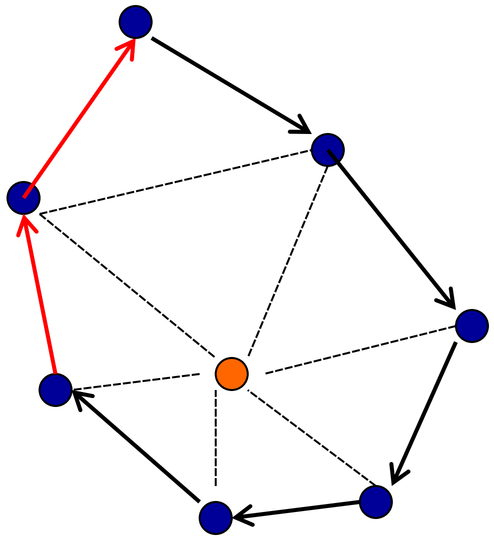
Ball pivoting around active edge

Point on front

Internal point

Pivoting example

Boundary edge



Active edge

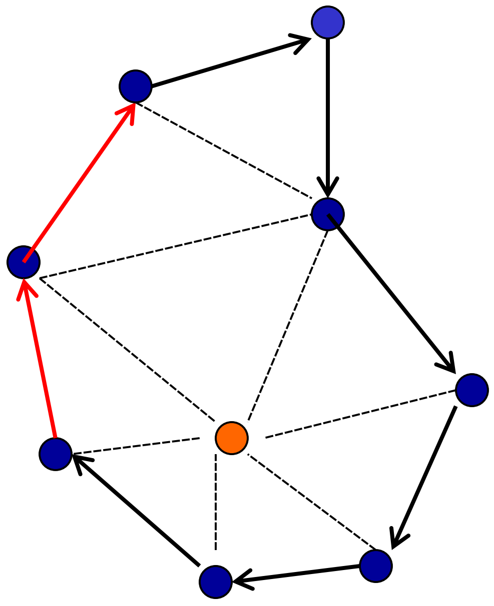
Point on front

Internal point

Ball pivoting around active edge
No pivot found

Pivoting example

Boundary edge



Active edge

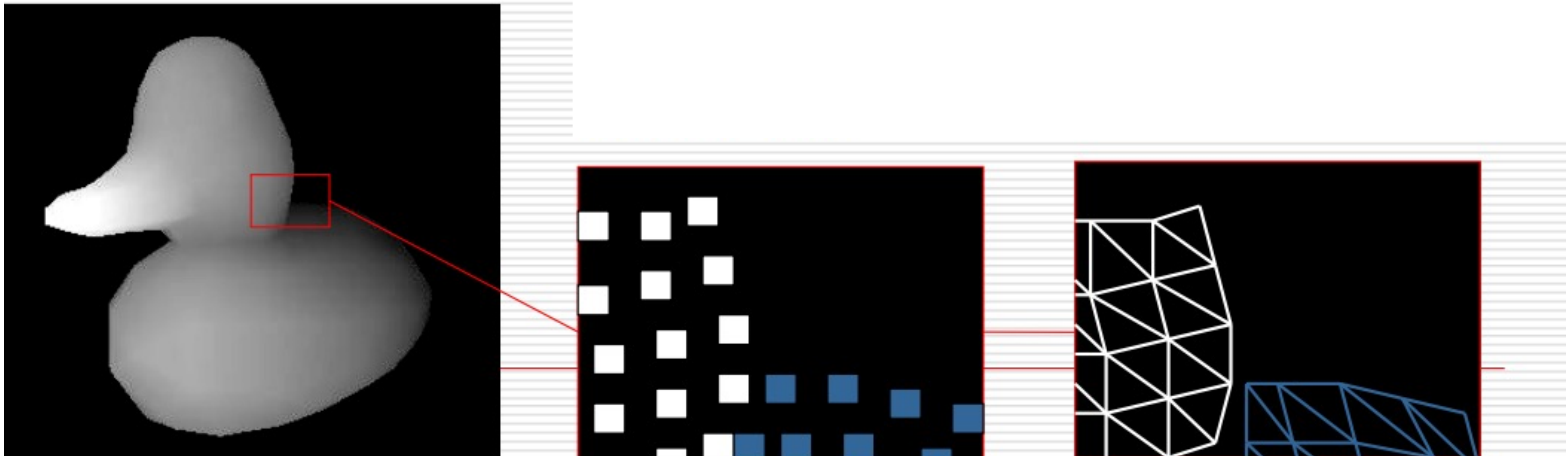
Ball pivoting around active edge

Point on front

Internal point

Not any point clouds: the Range Maps

- 3D scanners produce a numner of dense structured height fields, that is, a regular (X,Y) grid of points with a distance Z value. These are called **range maps**
- Trivial to triangulate but: How to merge different range maps?



Mesh Zippering [Turk94]

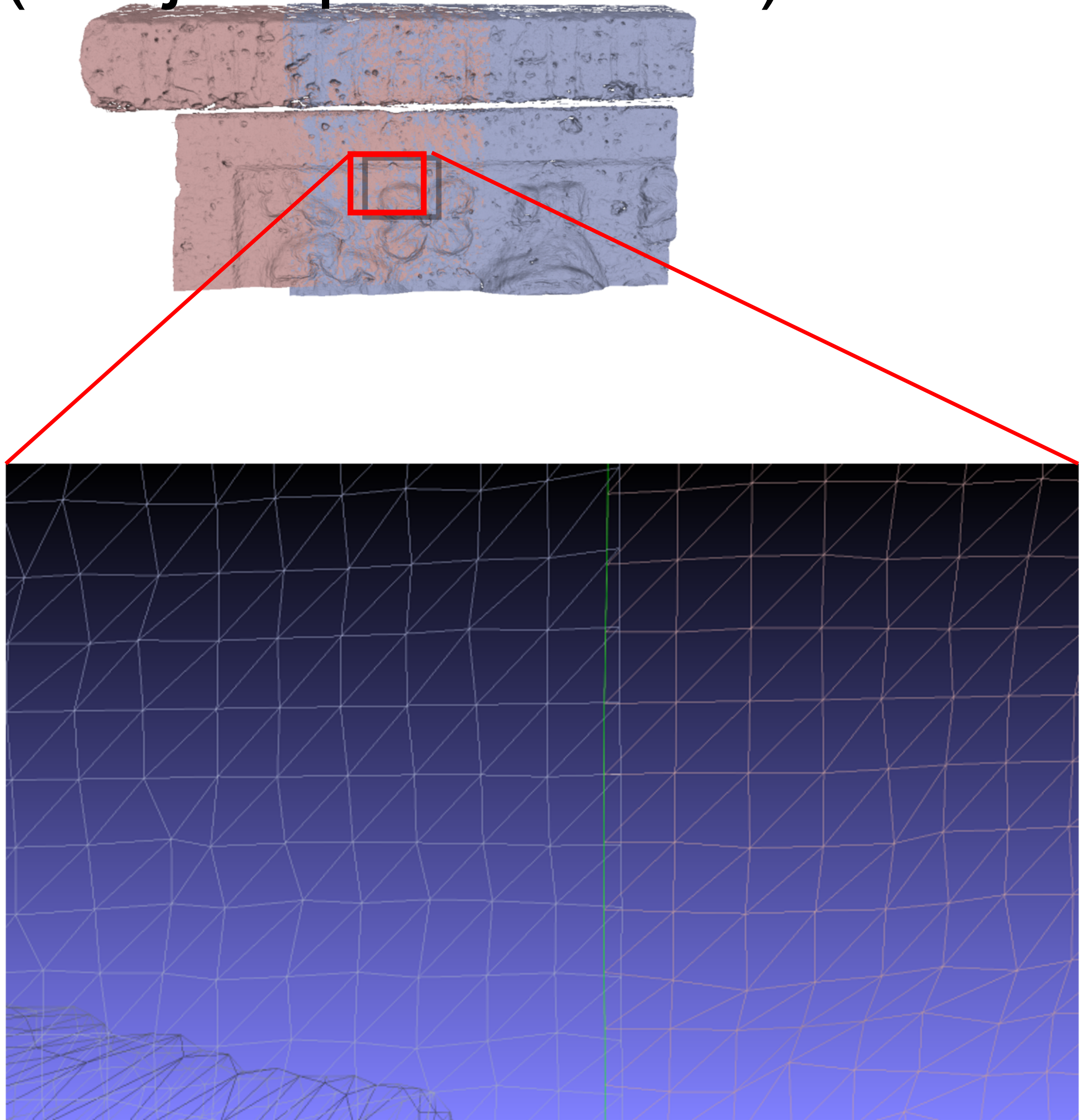
- Input: triangulated ranges maps (not just point clouds)
- Works in pairs:
  - **Remove overlapping portions**
  - Clip one RM against the other
  - Remove small triangles

Mesh Zippering

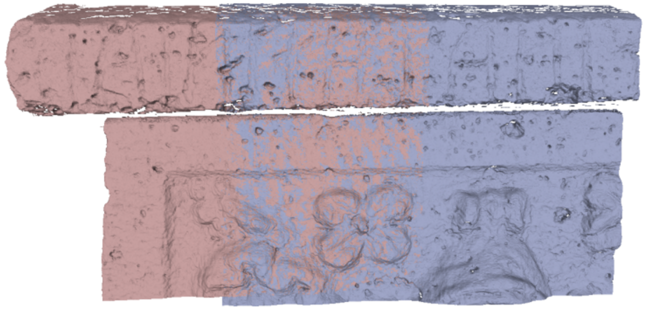
Input: triangulated ranges maps (not just point clouds)

Works in pairs:

- **Remove overlapping portions**
- Clip one RM against the other
- Remove small triangles



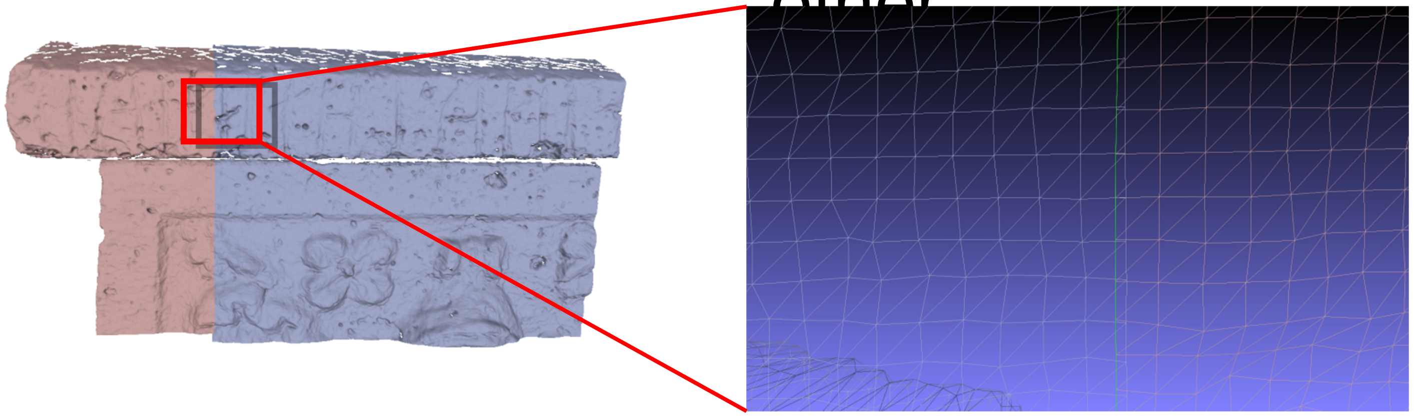
Mesh Zippering



nges maps (not just

- Remove overlapping portions

other

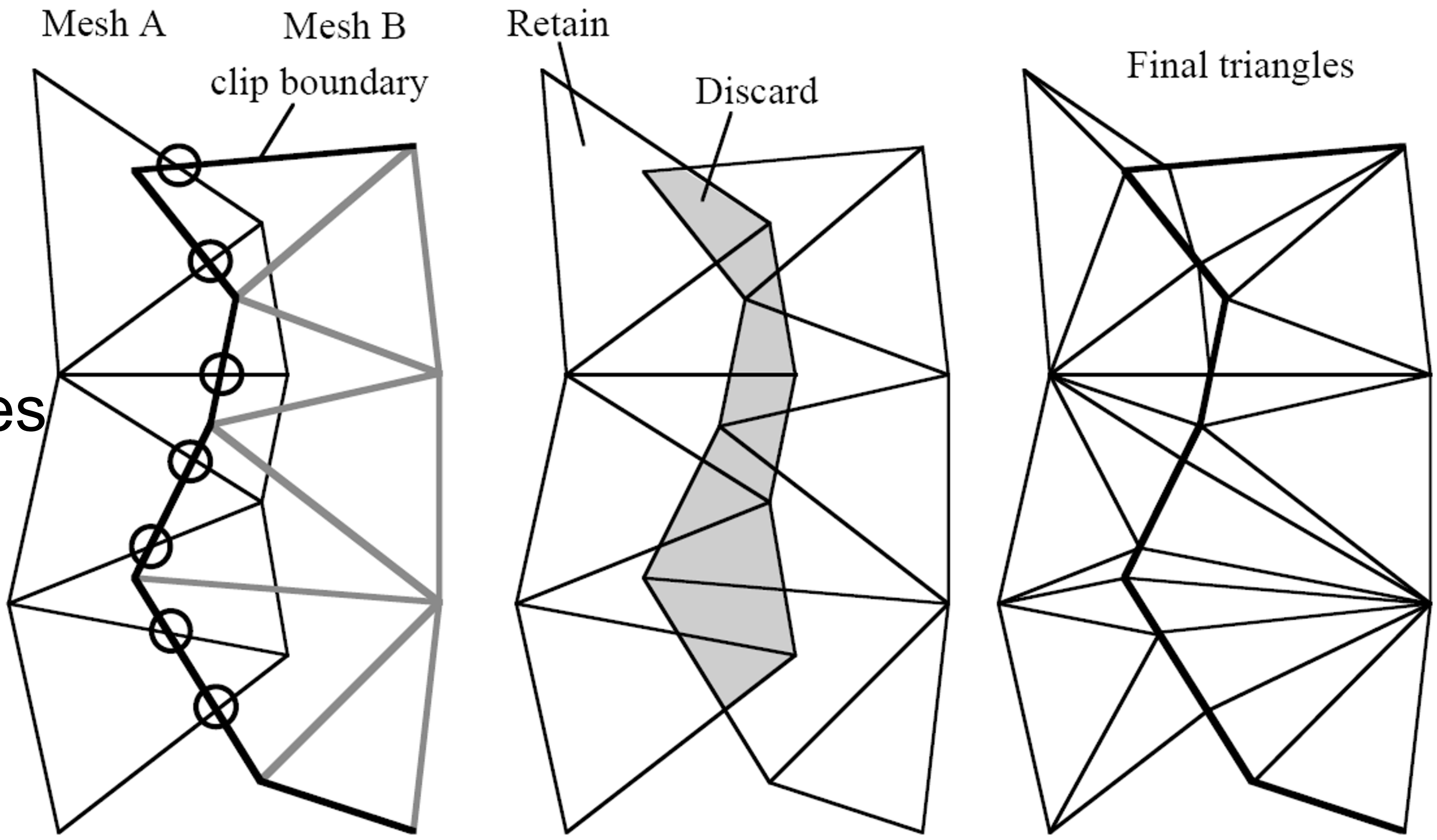


Mesh Zippering

Input: triangulated ranges maps (not just point clouds)

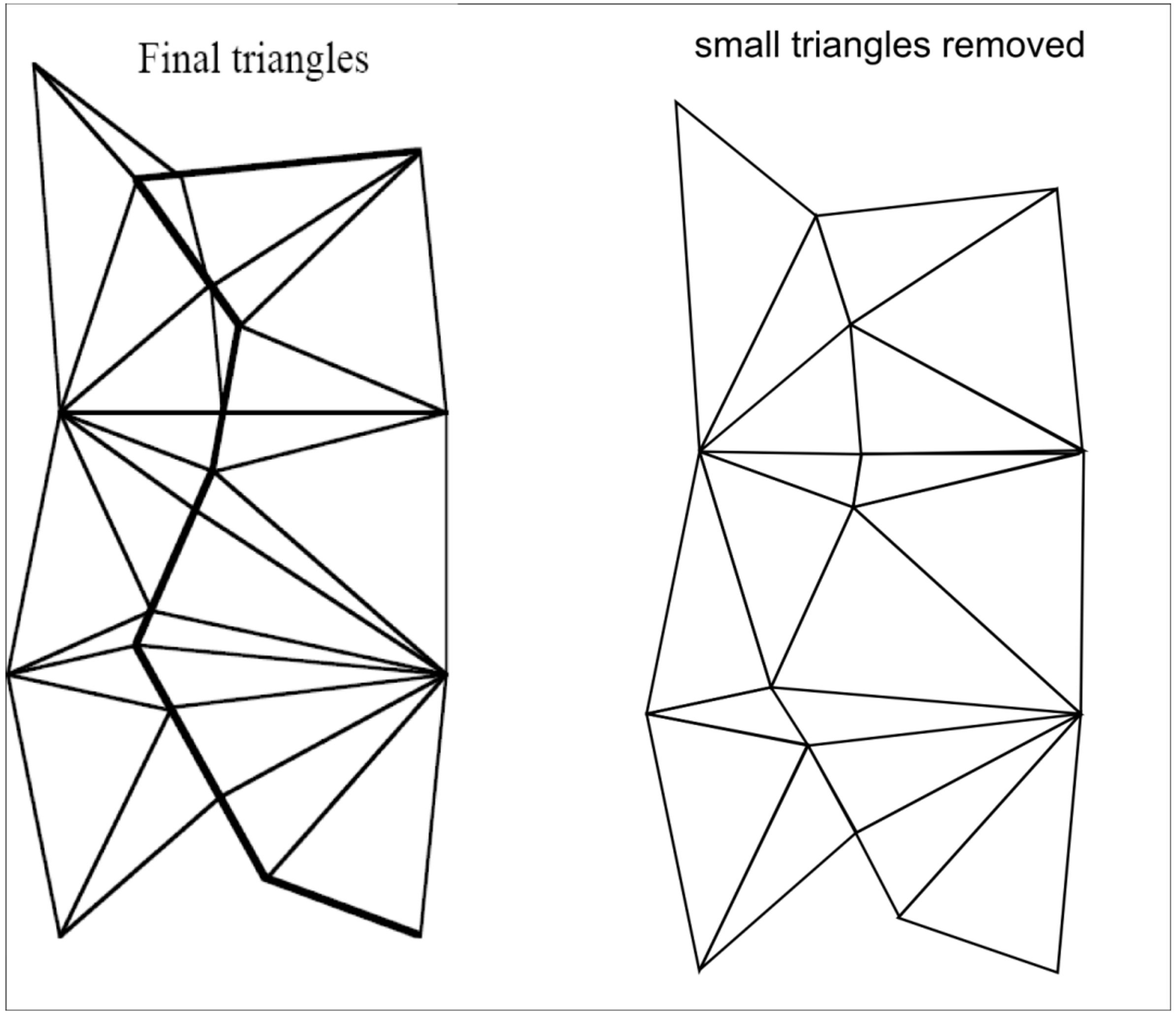
Works in pairs:

- Remove overlapping portions
- **Clip one RM against the other**
- Remove small triangles



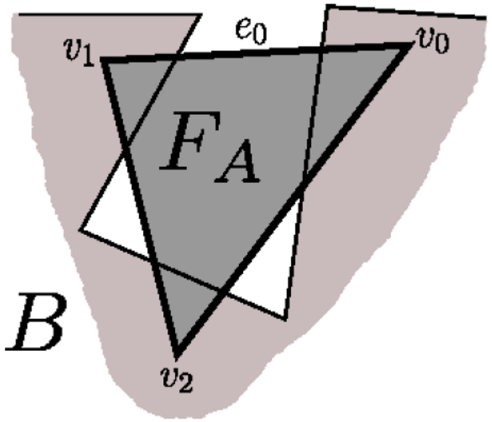
Mesh Zippering

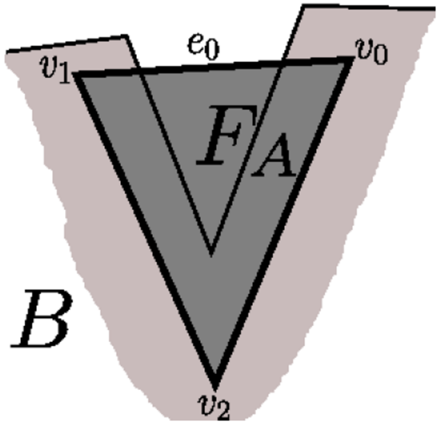
- Input: triangulated ranges maps (not just point clouds)
- Works in pairs:
  - Remove overlapping portions
  - Clip one RM against the other
  - **Remove small triangles**



Mesh Zippering

- Not so trivial to implement…for example..
  - **remove overlapping regions**: «a face of mesh A overlaps if its 3 vertices project on mesh B»
- Hole may appear, to be fixed later…

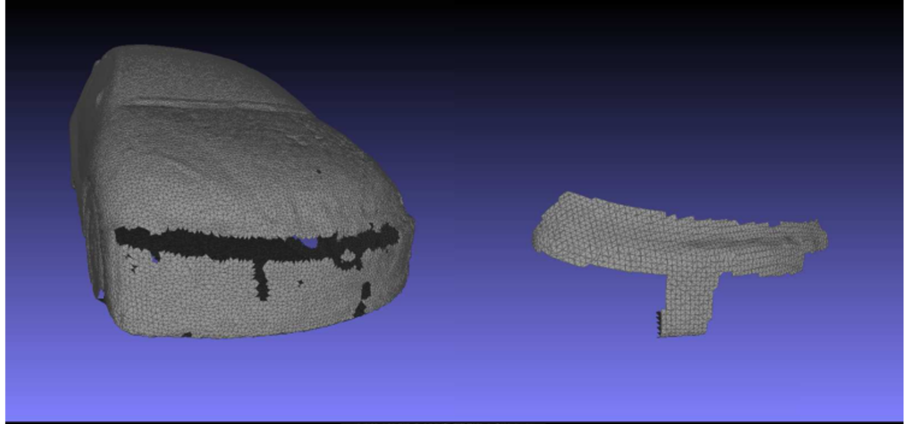




Mesh Zippering

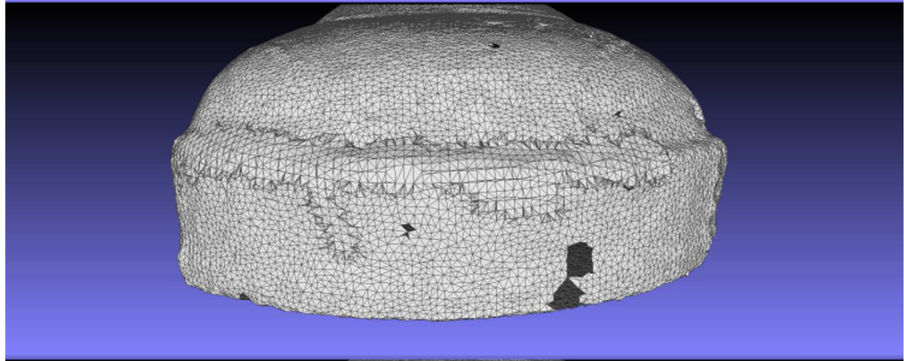
- Not so trivial to implement…for example..
  - **remove overlapping regions**: criterion?

Mesh Zippering

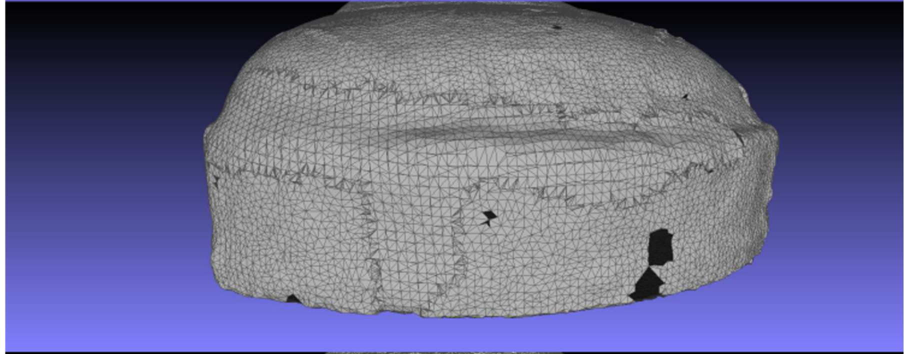


- Not so trivial to implement…for example..
  - **remove overlapping regions**: criterion?

Preserve faces from left



Preserve faces from right



Halfway (distance from the border)