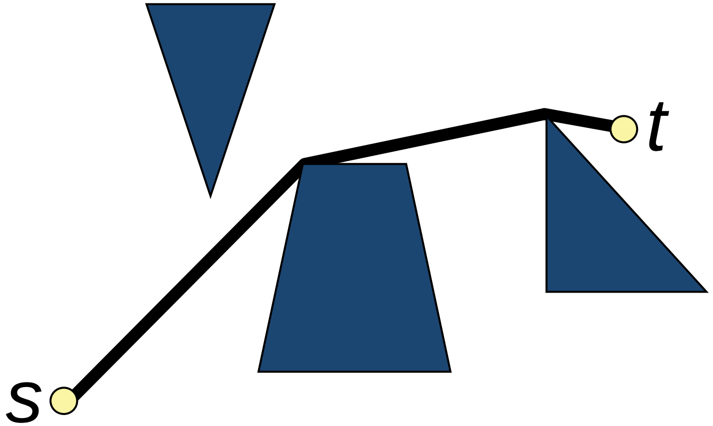
Part A
Basics of Computational Geometry

Computational geometry: develop efficient algorithms and data structures for solving problems stated in terms of basic geometrical objects, eg points, line segments, polygons or polyhedra

Problem examples

Shortest Path



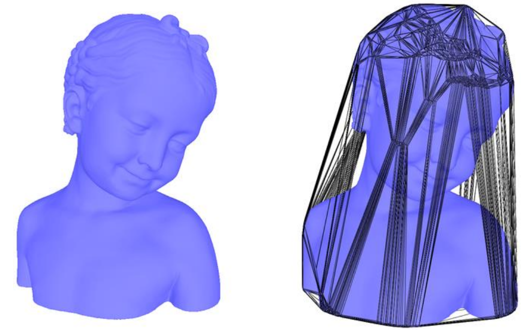
- Problem definition:
  - Input: Obstacles locations and *query* endpoints *s* and *t*.

Output: shortest path between *s* and *t* that avoids all obstacles.

Rules of the game: One obstacle set, multiple queries.

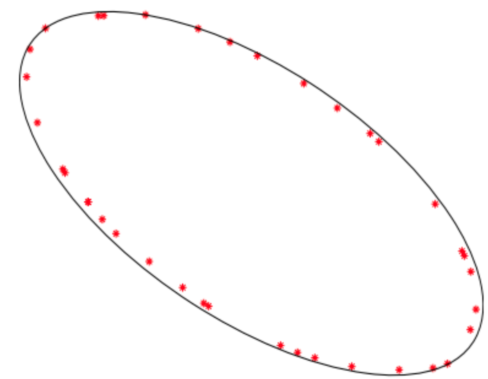
Problem examples

Bounding Volumes

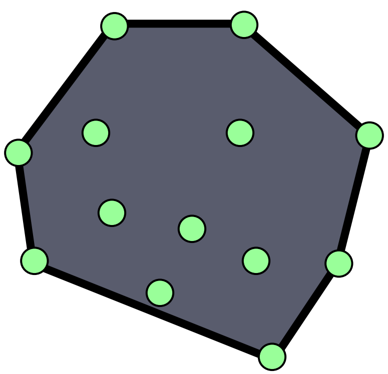


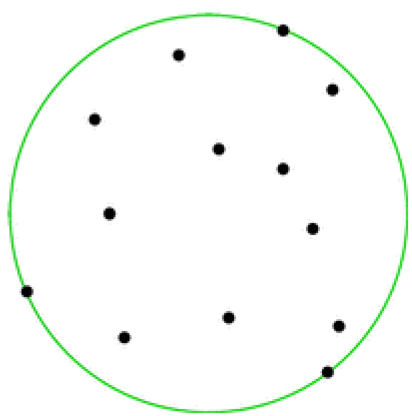
Problem definition:

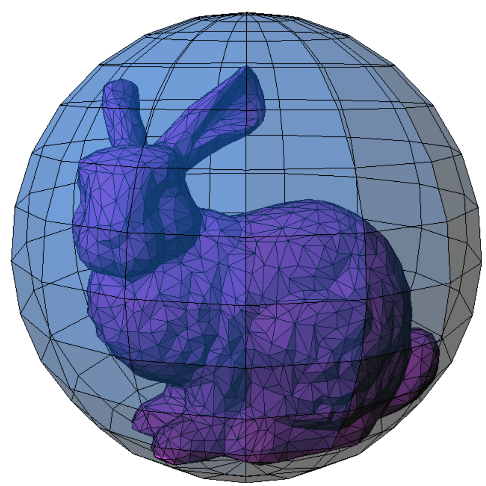
Input: Set of points *P* in the plane



Output: (report) Smallest enclosing polygon, disk, ellipse, annulus, rectangles, parallelograms, k>=2 axis-aligned rectangles



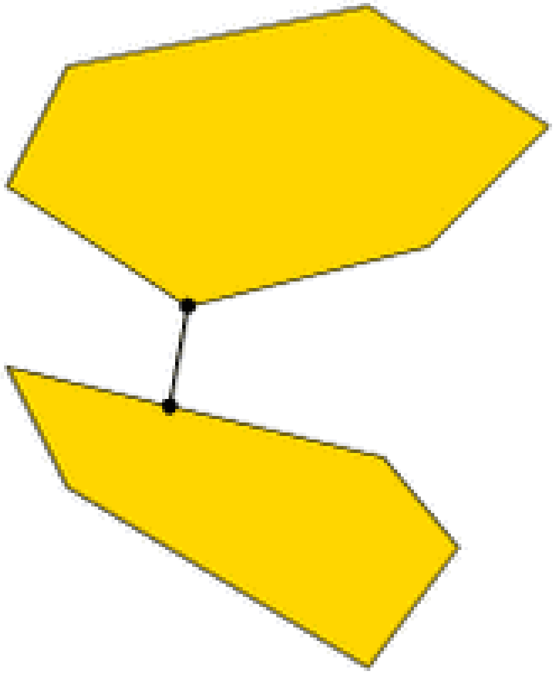




Problem examples

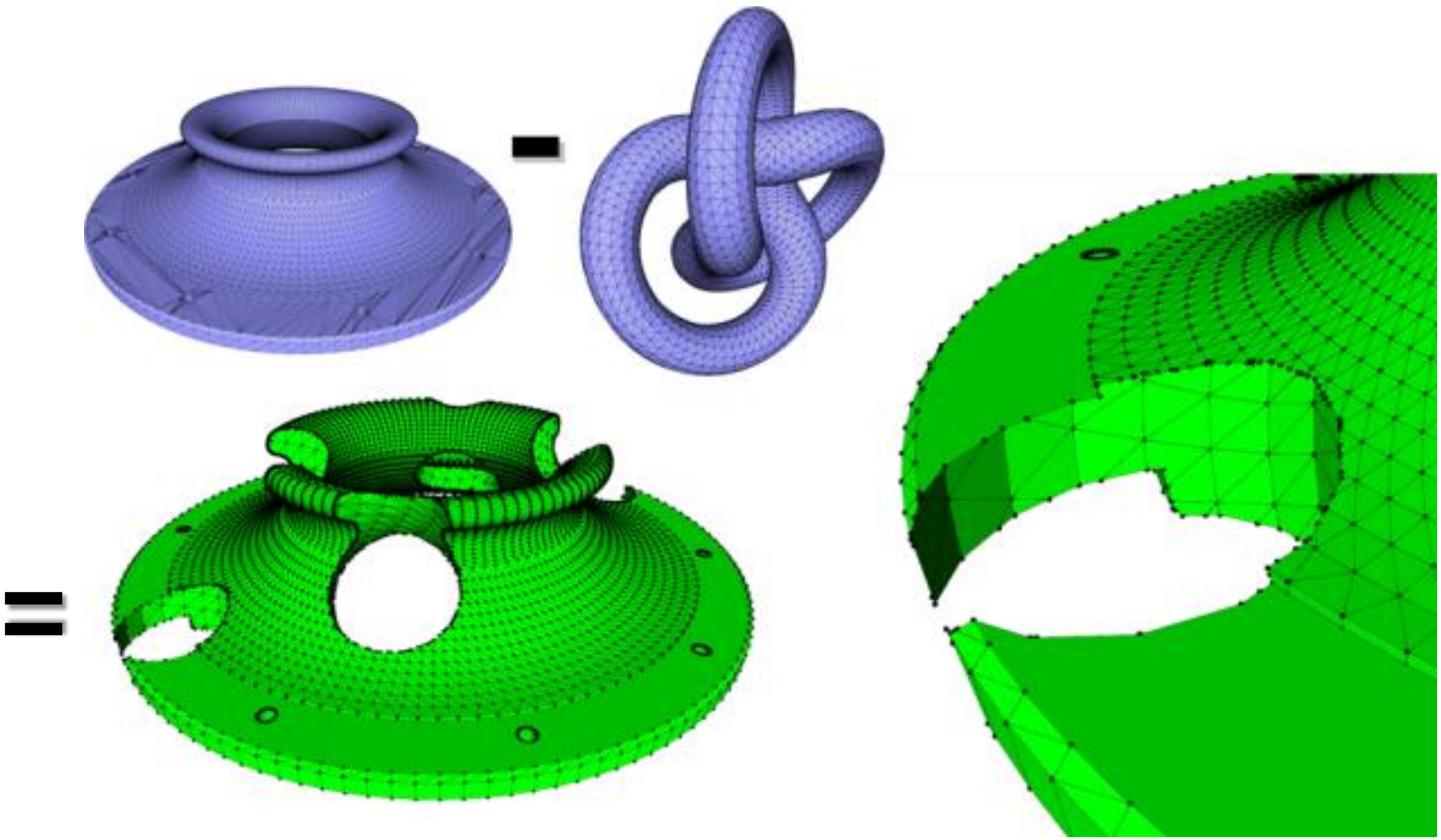
Optimal Distances

Distance between convex hulls of two point sets in Euclidean space (in dD!)



Problem examples

Boolean Operations



Outlines

Some basic geometric objects and data-structures
- Voronoi diagram and Delaunay triangulation
- polygonal mesh

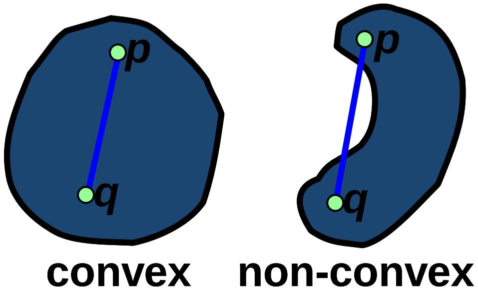
Shape reconstruction
- simple
- smooth
- primitive-based

1

Some basic geometric objects and data-structures

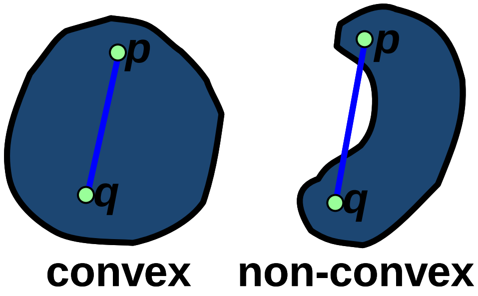
Convex hulls

A set *S* is *convex* if for any two points *a*,*b*∈*S*, the line segment between *a* and *b* is also in *S*.

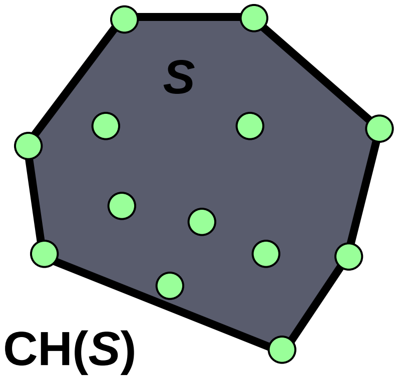


Convex hulls

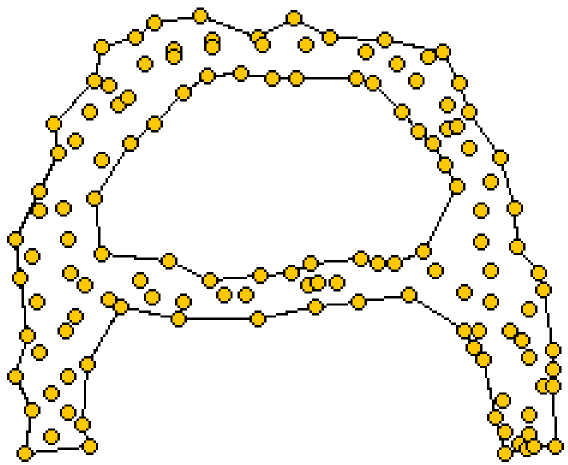
A set *S* is *convex* if for any two points *a*,*b*∈*S*, the line segment between *a* and *b* is also in *S*.

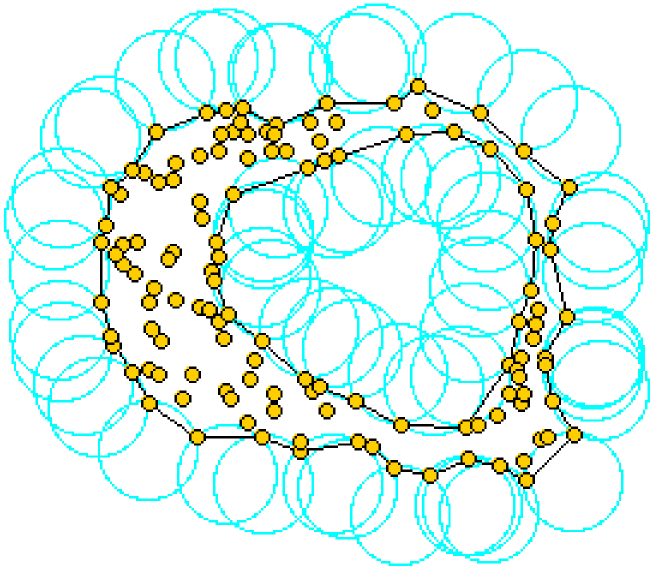


The *convex hull* of a set of points is the smallest convex set containing *S*.



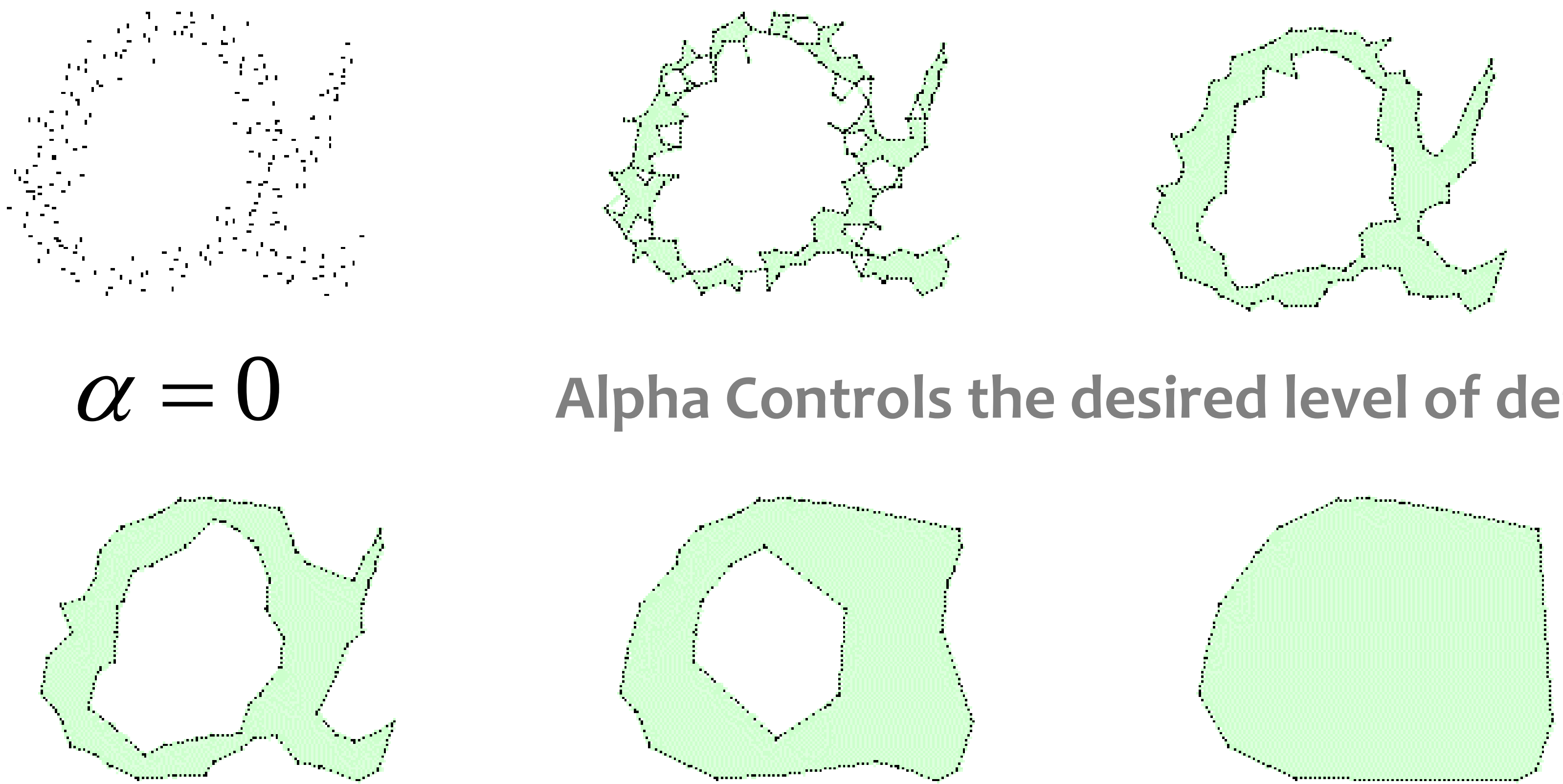
Alpha-shapes





The space generated by point pairs that can be touched by an empty disc of radius alpha.

Alpha-shapes



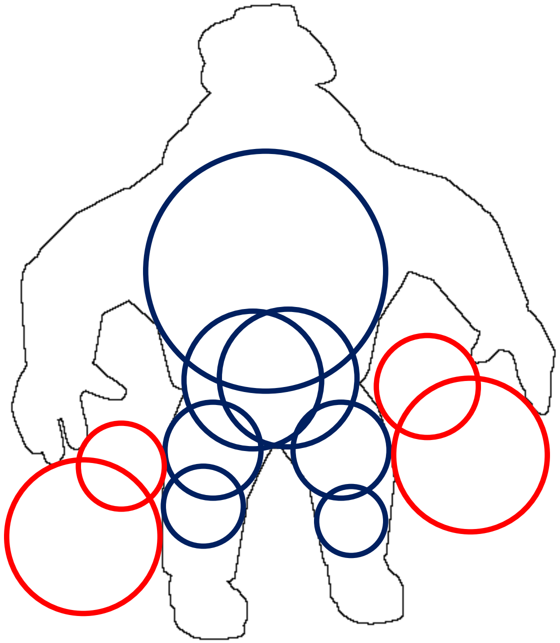
$\alpha = 0$

Alpha Controls the desired level of detail

$\alpha = \infty$

Medial axis

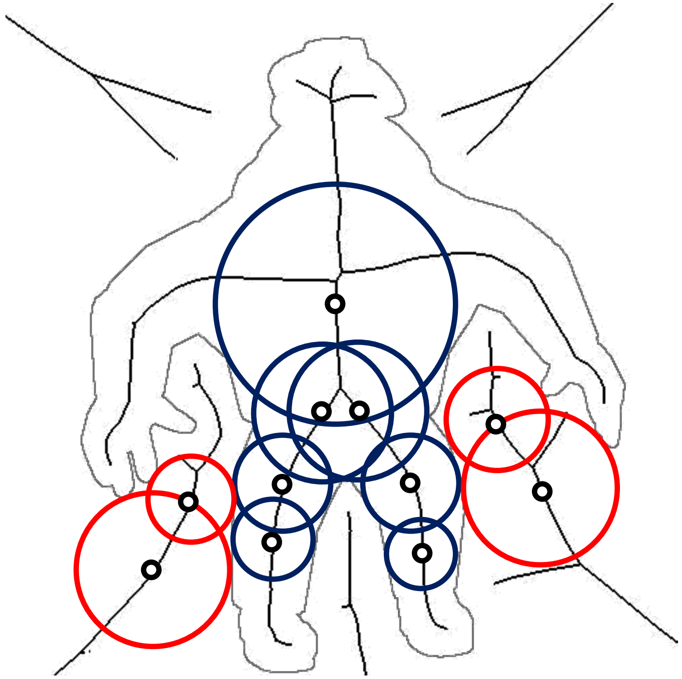
For a shape (curve/surface) a *Medial Ball* is a circle/sphere that only meets the shape tangentially, in at least two points.



Medial axis

For a shape (curve/surface) a *Medial Ball* is a circle/sphere that only meets the shape tangentially, in at least two points.

The centers of all such balls make up the *medial axis/skeleton*.

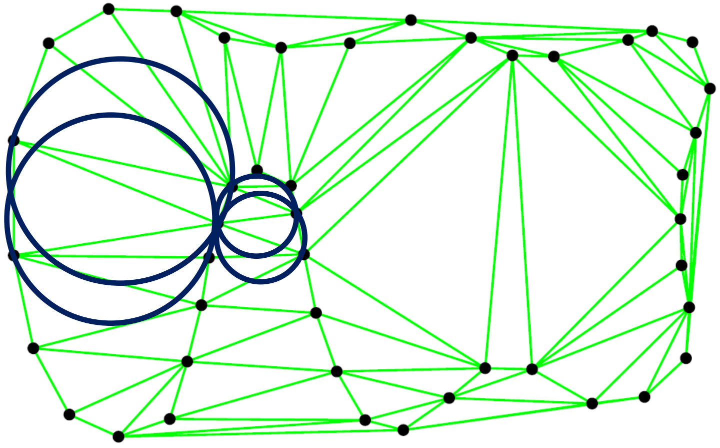


Delaunay triangulation

A *Delaunay Triangulation* of *S* is the set of all triangles with vertices in *S* whose circumscribing circle contains no other points in *S**.

Compactness Property:

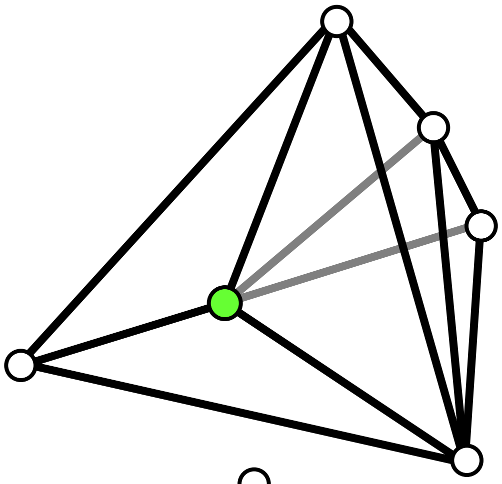
This is a triangulation that maximizes the min angle.

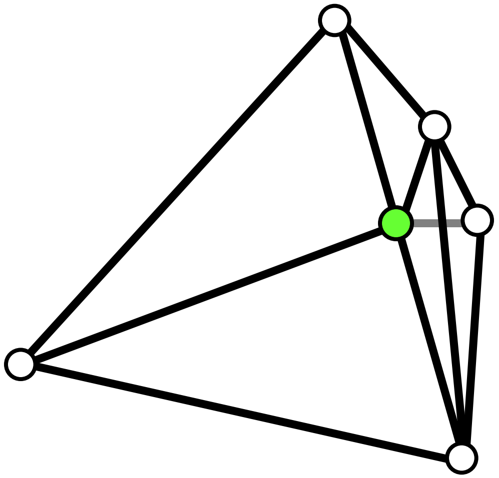


Construction of a Delaunay triangulation

O(n log n) Incremental algorithm:

- Form bounding triangle which encloses all the sites.
- Add the sites one after another in random order and update triangulation.
- **If the site is inside an existing triangle:**
  - Connect site to triangle vertices.
  - Check if a 'flip' can be performed on one of the triangle edges. If so – check recursively the neighboring edges.
- **If the site is on an existing edge:**
  - Replace edge with four new edges.
  - Check if a 'flip' can be performed on one of the opposite edges. If so – check recursively the neighboring edges.





Construction of a Delaunay triangulation

Algorithm complexity

- **Point location for every point:** O(log n) time.
- **Flips:** Θ(n) expected time in total (for all steps).

Total expected time: O(n log n).

Space: Θ(n).

Voronoi diagrams

The *Voronoi Diagram* of *S* is a partition of space into regions *V*(*p*) (*p*∈*S*) such that all points in *V*(*p*) are closer to *p* than any other point in *S*.

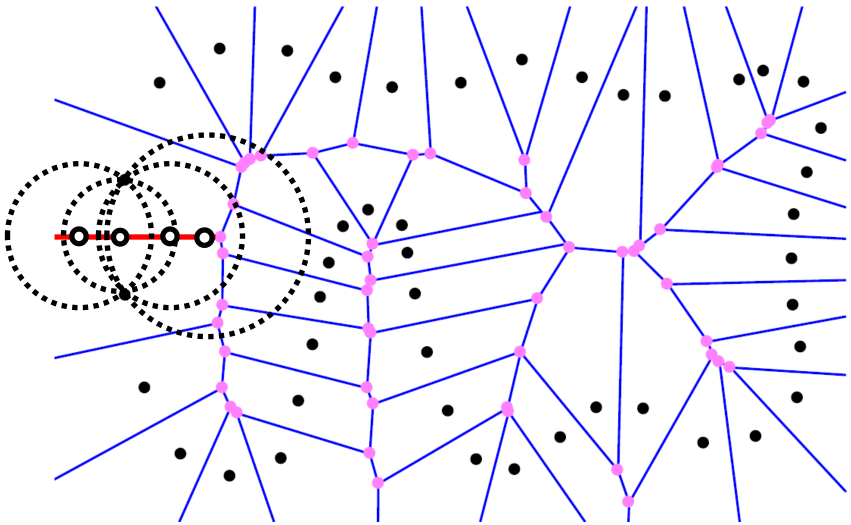
Let $\mathcal{E} = \{\mathbf{p_1}, \ldots, \mathbf{p_n}\}$ be a set of points (so-called sites) in $\mathbb{R}^d$. We associate to each site $\mathbf{p_i}$ its Voronoi region $V(\mathbf{p_i})$ such that:

$$V(\mathbf{p_i}) = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x} - \mathbf{p_i}\| \le \|\mathbf{x} - \mathbf{p_j}\|, \forall j \le n\}.$$

Voronoi diagrams

The *Voronoi Diagram* of *S* is a partition of space into regions *V*(*p*) (*p*∈*S*) such that all points in *V*(*p*) are closer to *p* than any other point in *S*.

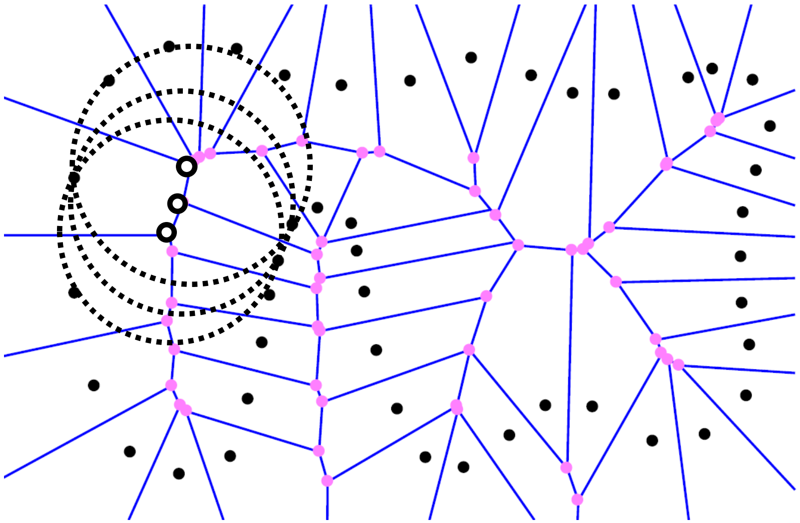
For a point on an edge, we can draw an empty circle that only touches the points in *S* separated by the edge.



Voronoi diagrams

The *Voronoi Diagram* of *S* is a partition of space into regions *V*(*p*) (*p*∈*S*) such that all points in *V*(*p*) are closer to *p* than any other point in *S*.

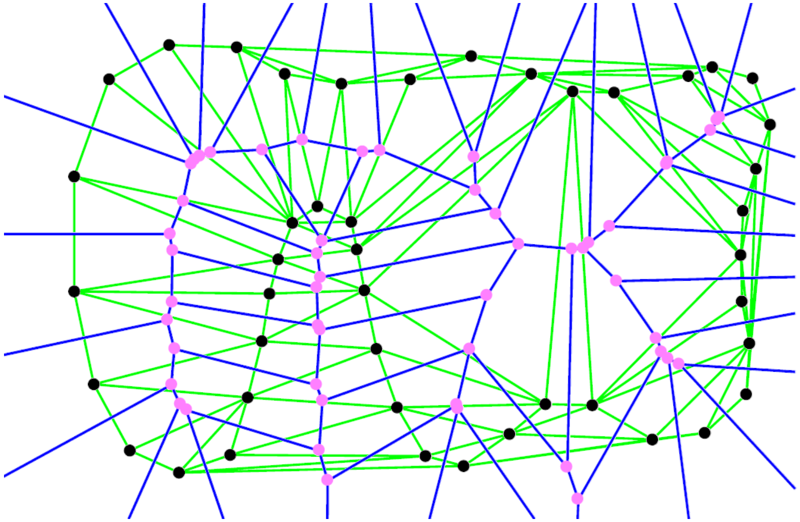
For a vertex, we can draw an empty circle that just touches the three points in *S* around the vertex.



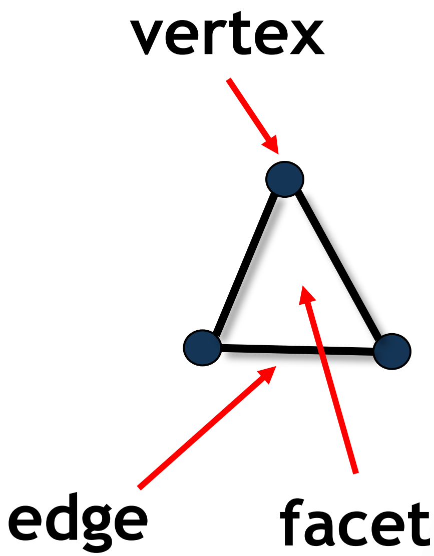
Voronoi diagrams

The *Voronoi Diagram* of *S* is a partition of space into regions *V*(*p*) (*p*∈*S*) such that all points in *V*(*p*) are closer to *p* than any other point in *S*.

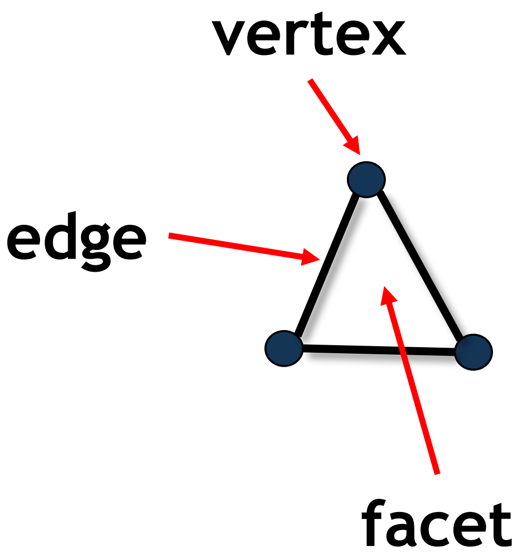
Duality:
Each Voronoi vertex is in one-to-one correspondence with a Delaunay triangle.



Polygonal meshes



Polygonal meshes

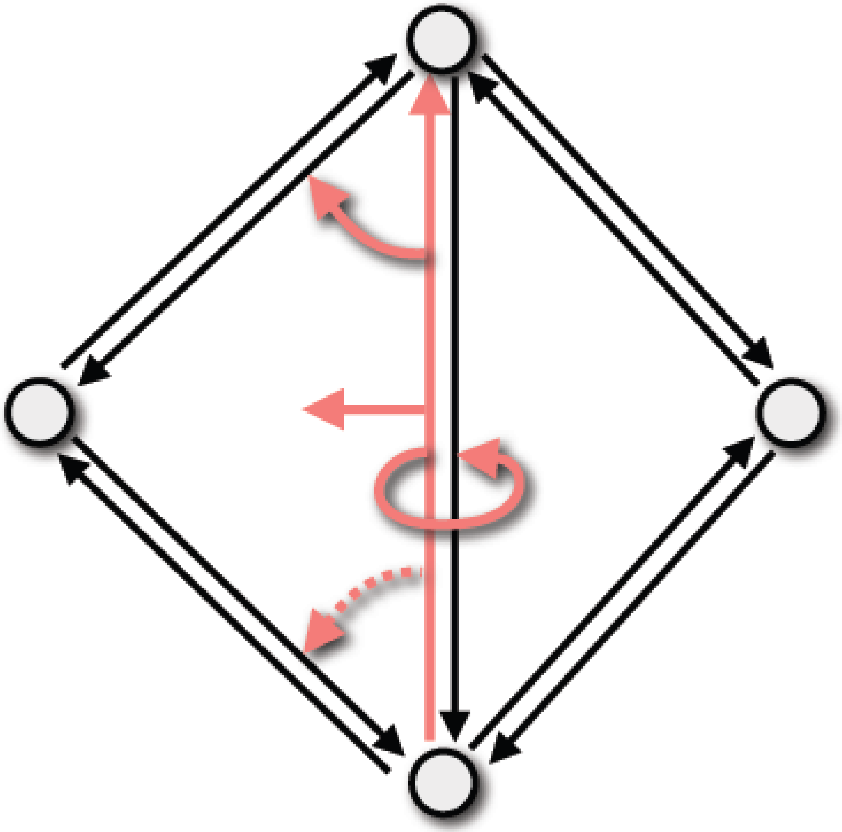


How good is a mesh data-structure?

- Time to construct (preprocessing)
- Time to answer a query
- Time to perform an operation
- Space complexity
- Redundancy

Polygonal meshes

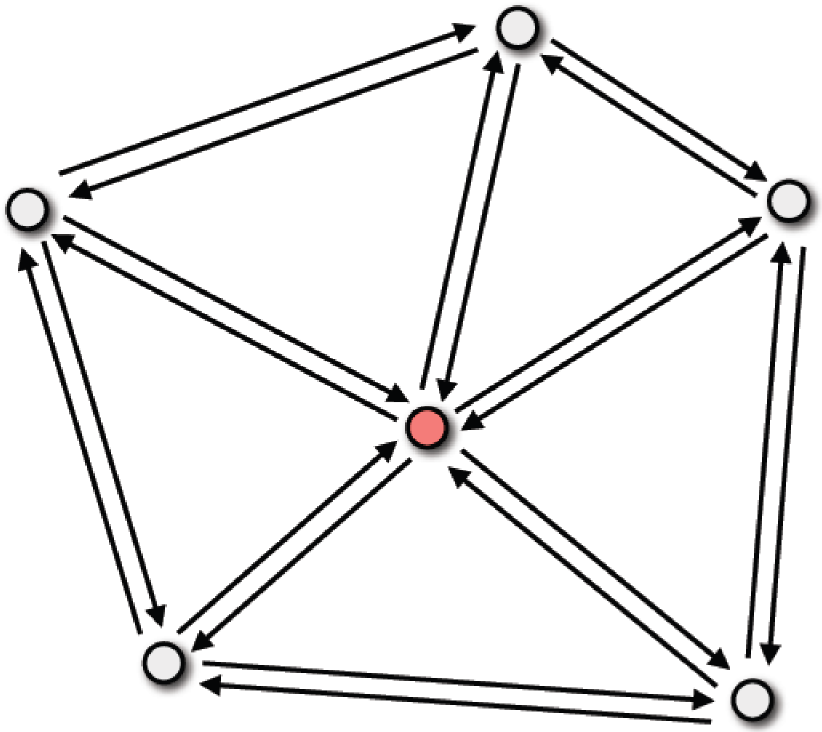
Halfedge-based connectivity



Polygonal meshes

One-ring traversal

1. Start at vertex

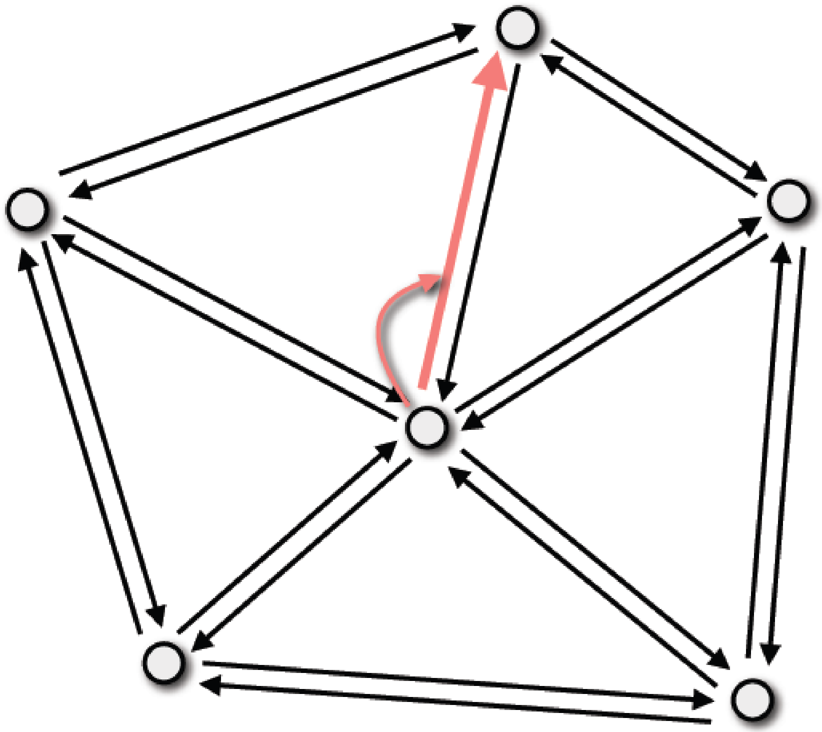


Polygonal meshes

One-ring traversal

1. Start at vertex

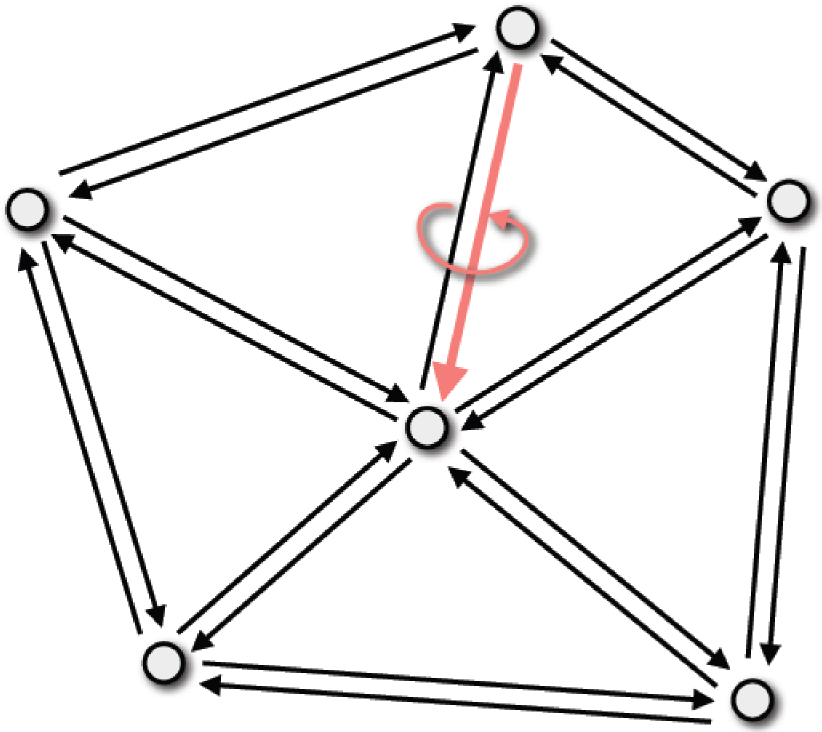
2. Outgoing halfedge



Polygonal meshes

One-ring traversal

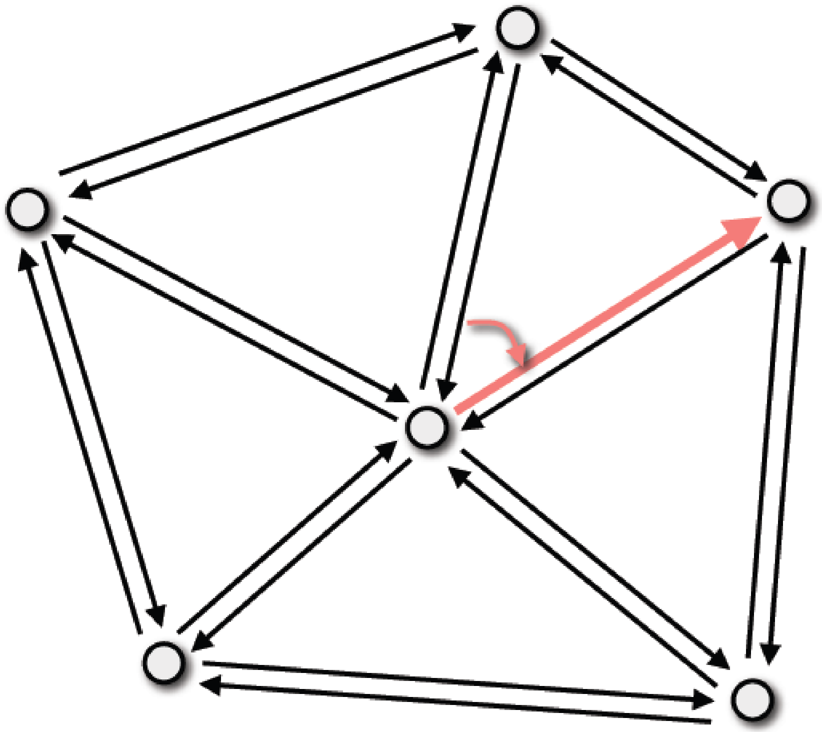
1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge



Polygonal meshes

One-ring traversal

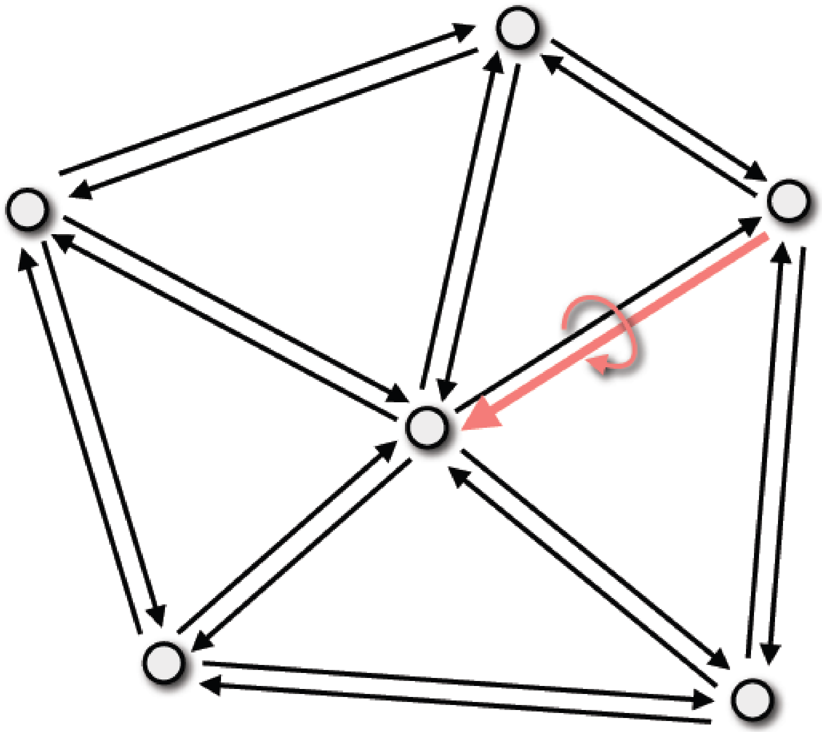
1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge
4. Next halfedge



Polygonal meshes

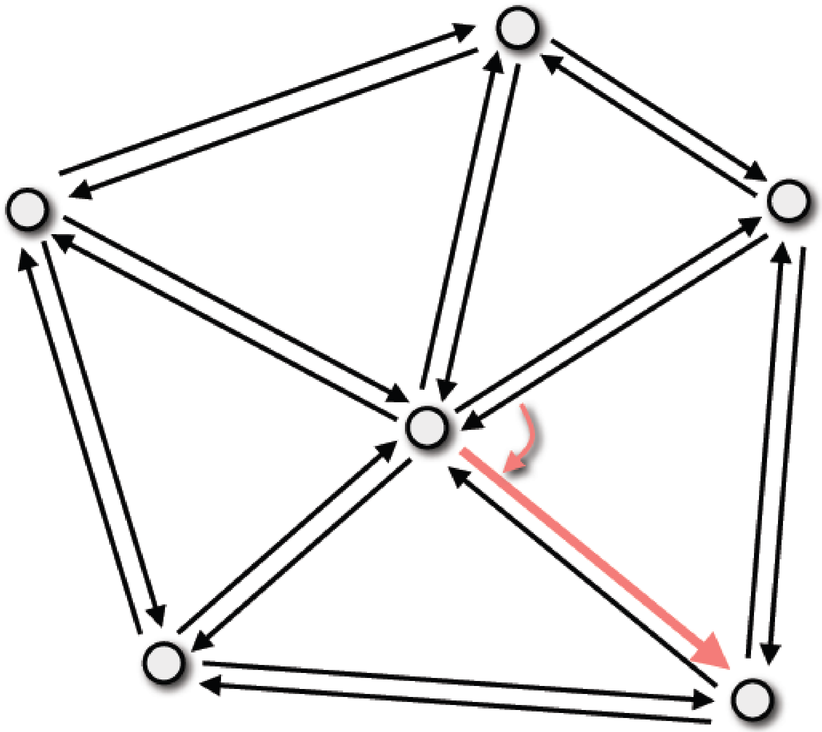
One-ring traversal

1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge
4. Next halfedge
5. Opposite



Polygonal meshes

One-ring traversal



1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge
4. Next halfedge
5. Opposite
6. Next
7. ...

2

Shape reconstruction

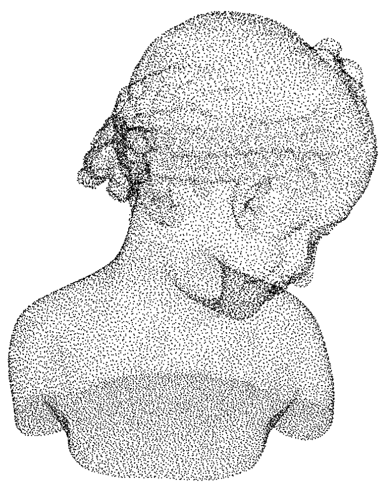
Reconstruction problem

Input: point set *P* sampled over a surface *S*:

- Non-uniform sampling
- With holes
- With uncertainty (noise)

Output: surface

- Approximation of *S* in terms of topology and geometry
- Desired:
  - Watertight
  - Intersection free



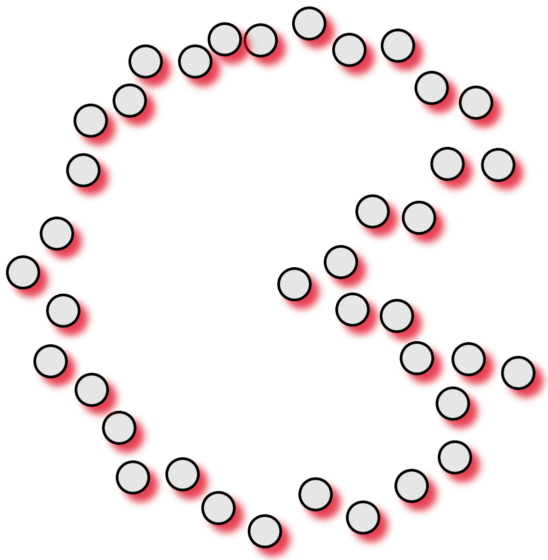
point set

reconstruction

surface

Reconstruction problem

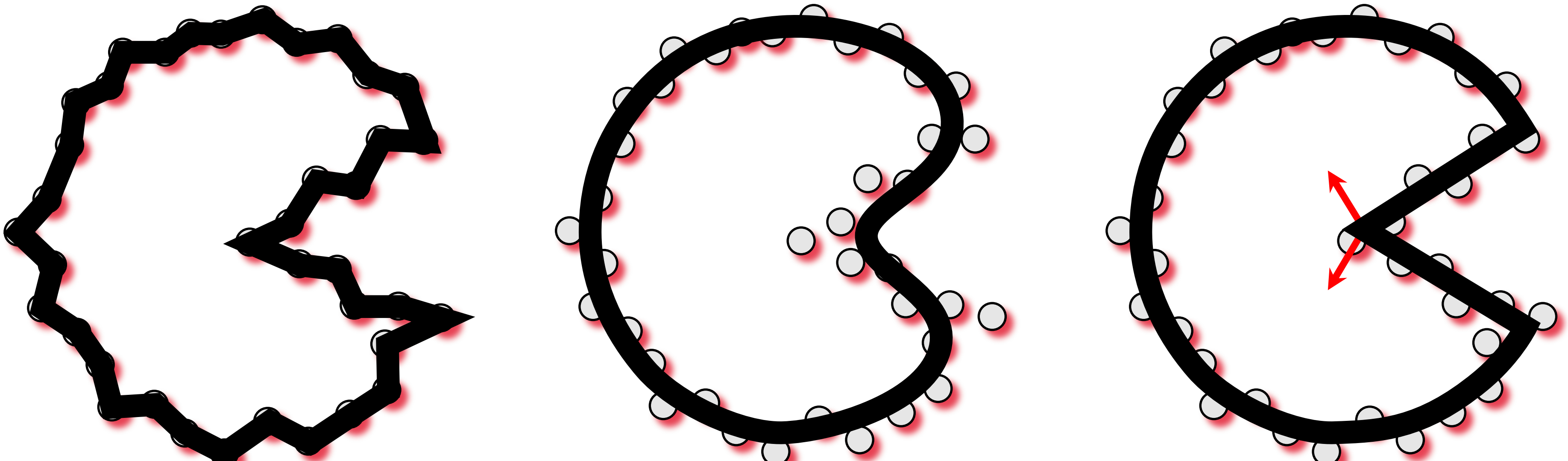
Ill-posed Problem



Many candidate shapes for the reconstruction problem!

Reconstruction problem

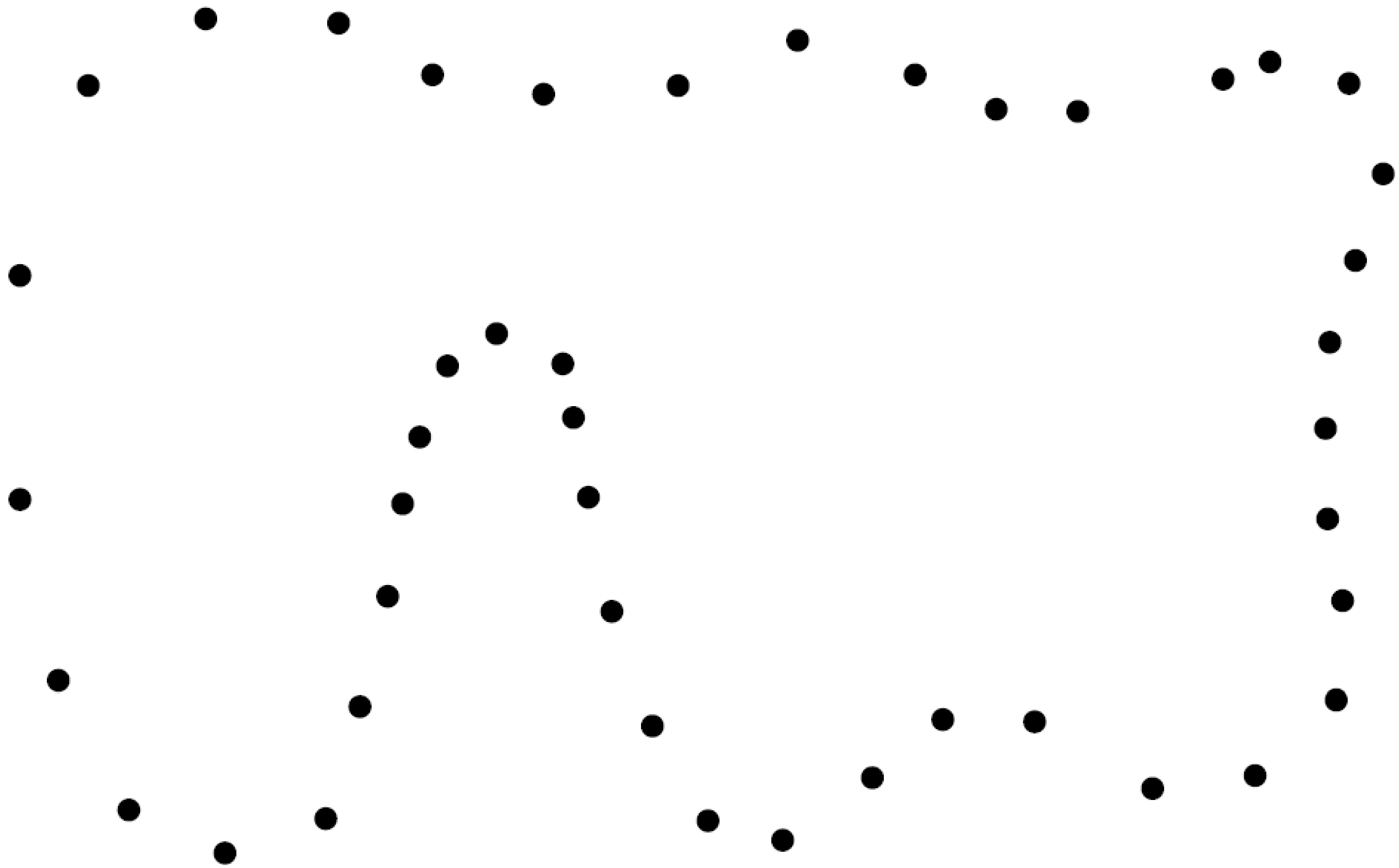
Ill-posed Problem



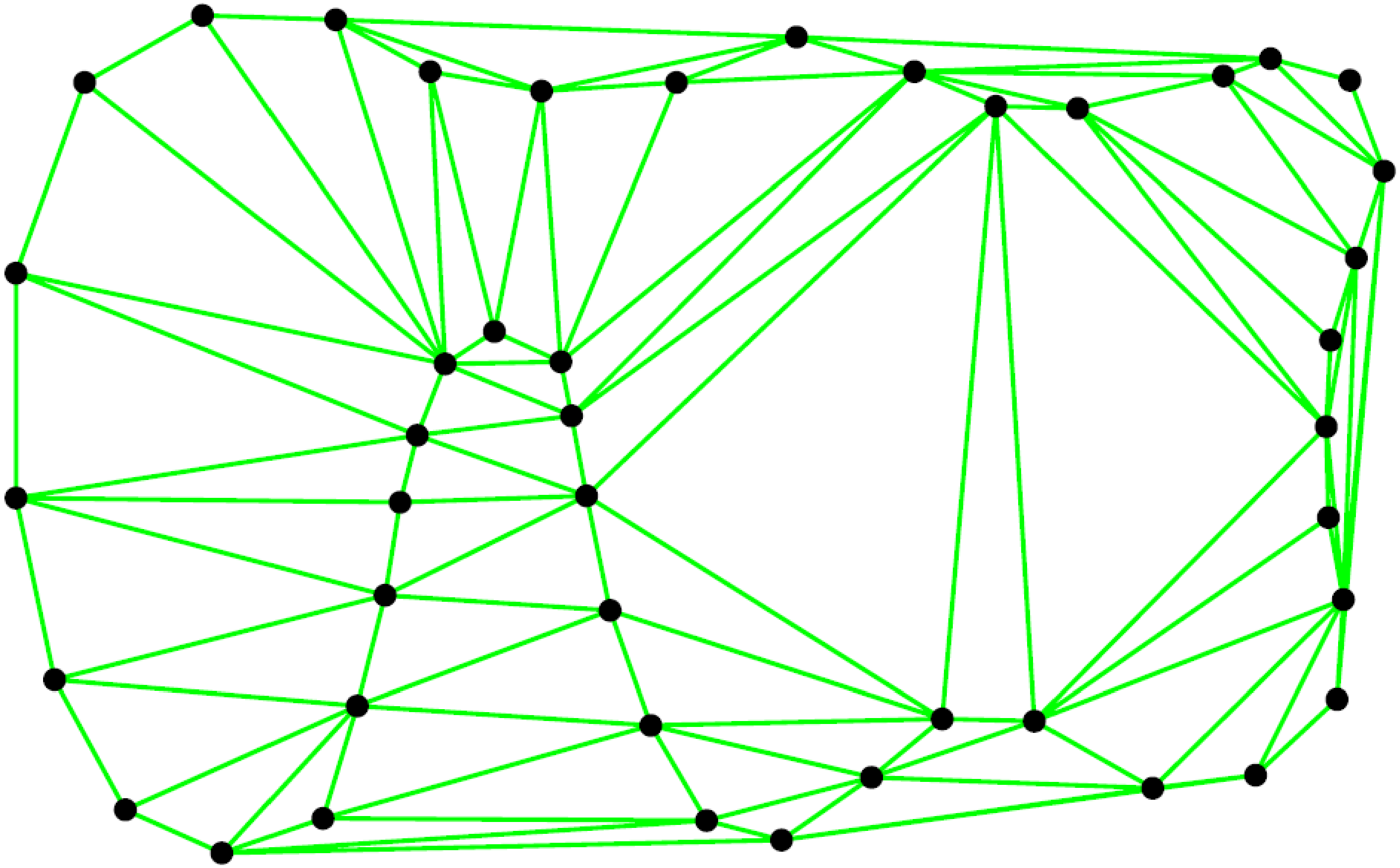
simple smooth primitive-based

Many candidate shapes for the reconstruction problem!

Simple : Crust

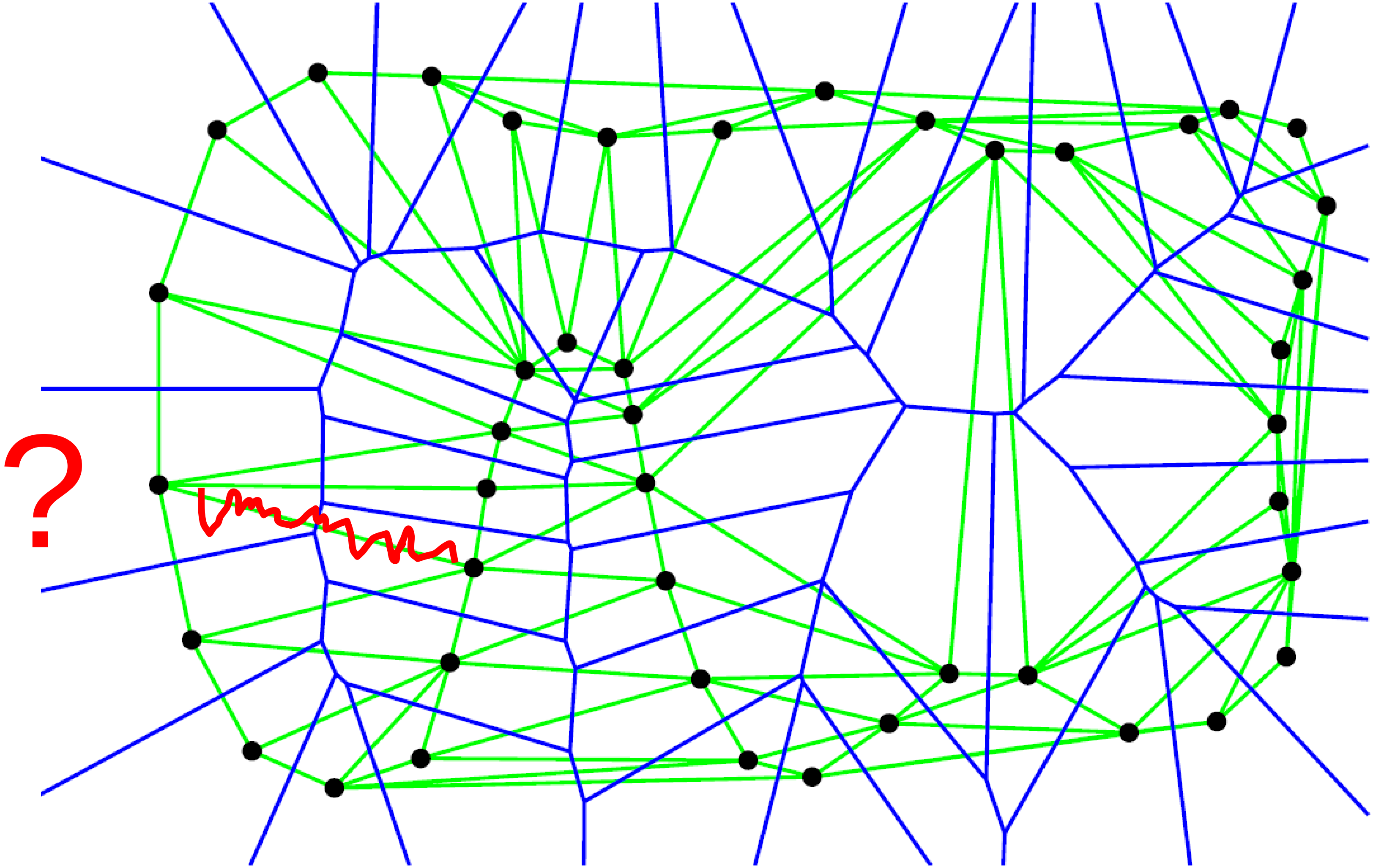


Delaunay Triangulation

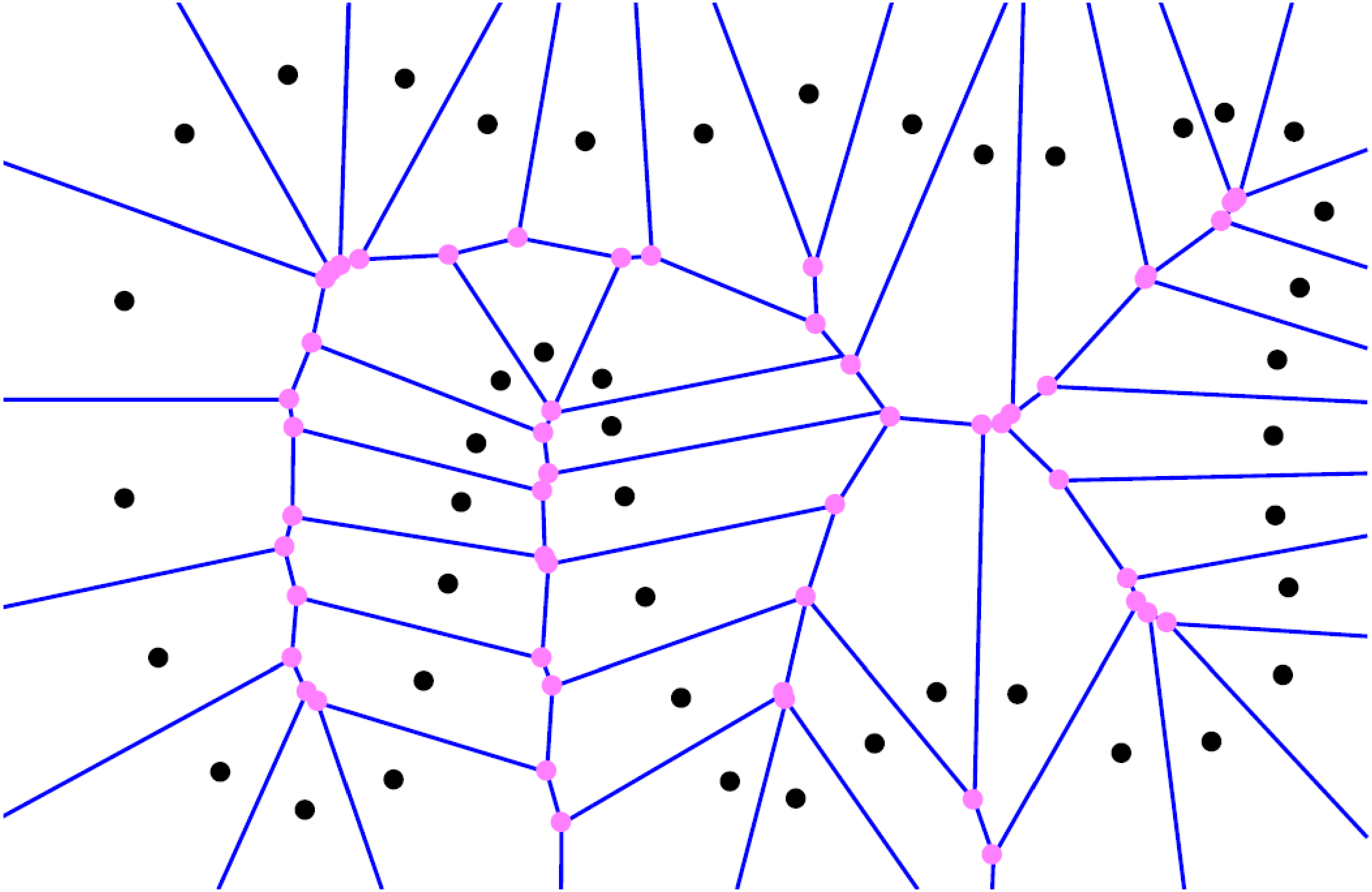


Output surface = a subset of Delaunay edges

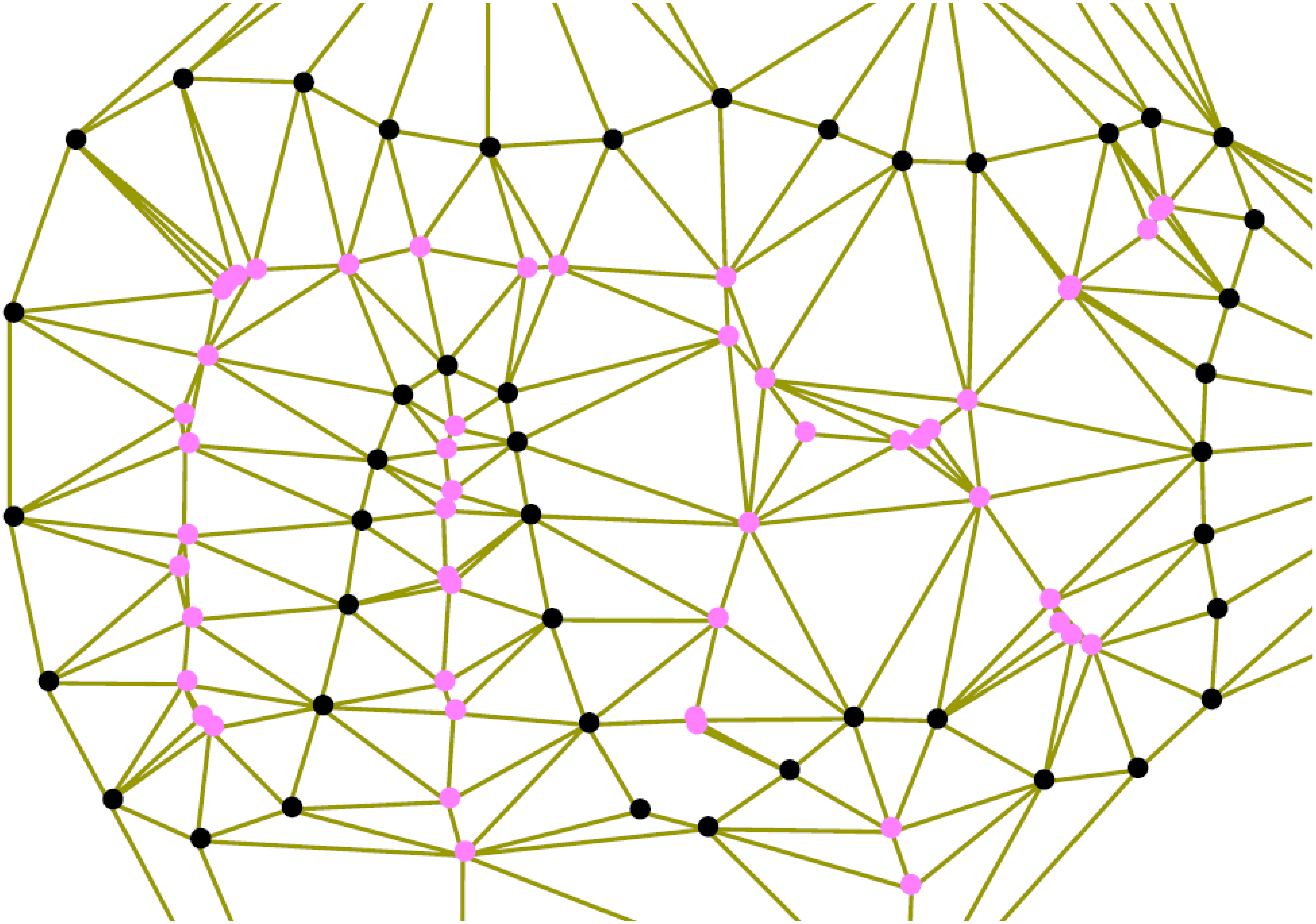
Delaunay Triangulation & Voronoi Diagram



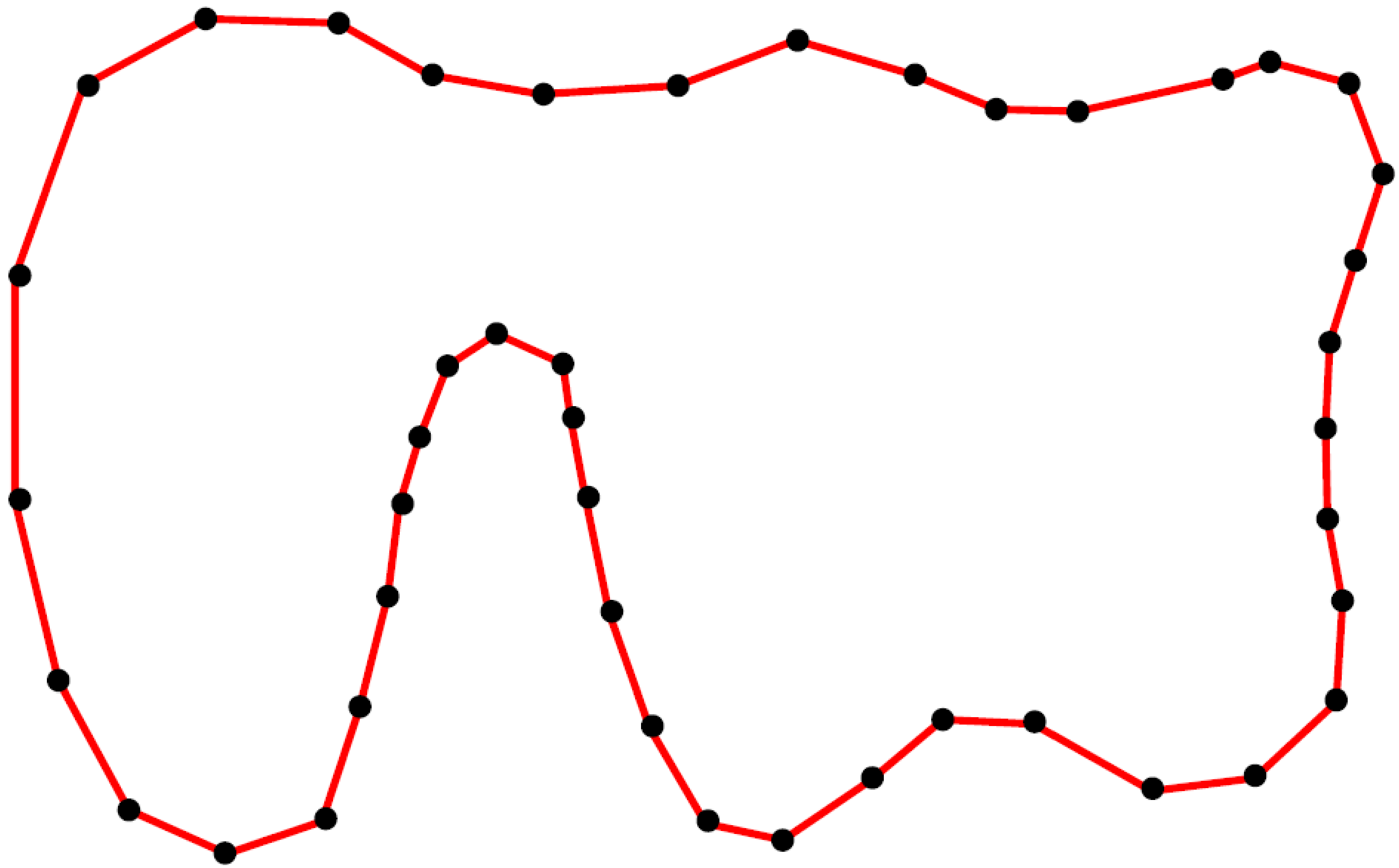
Voronoi Vertices



Refined Delaunay Triangulation

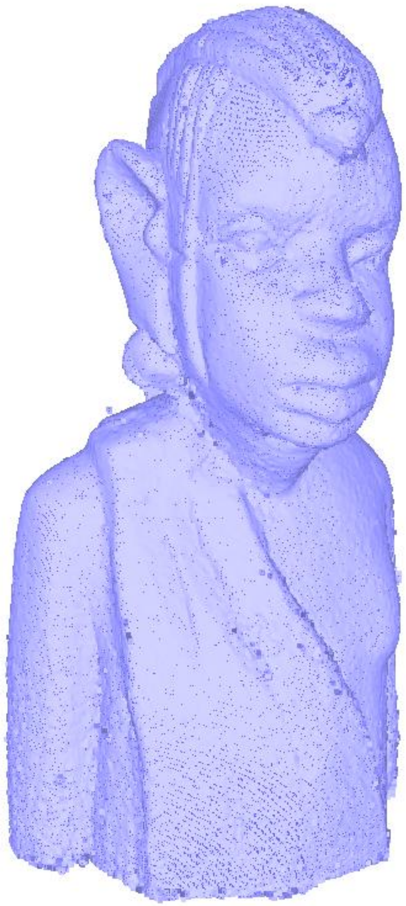


Selected edges

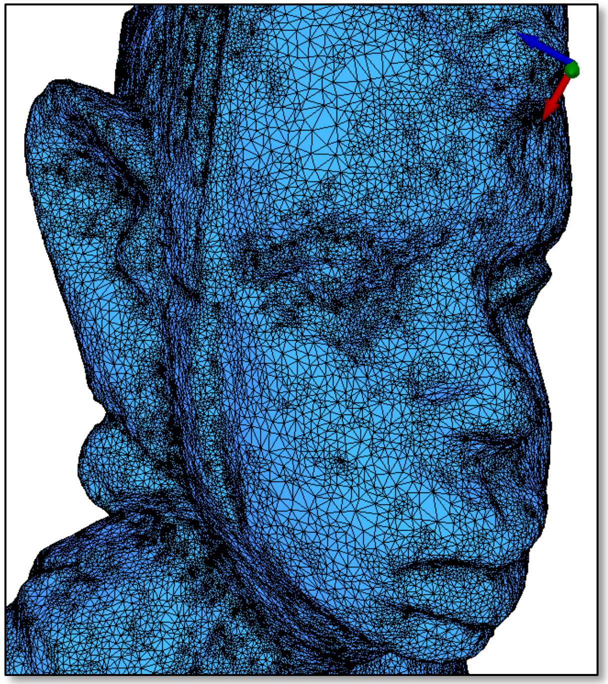


Smooth: Poisson

Poisson Surface Reconstruction

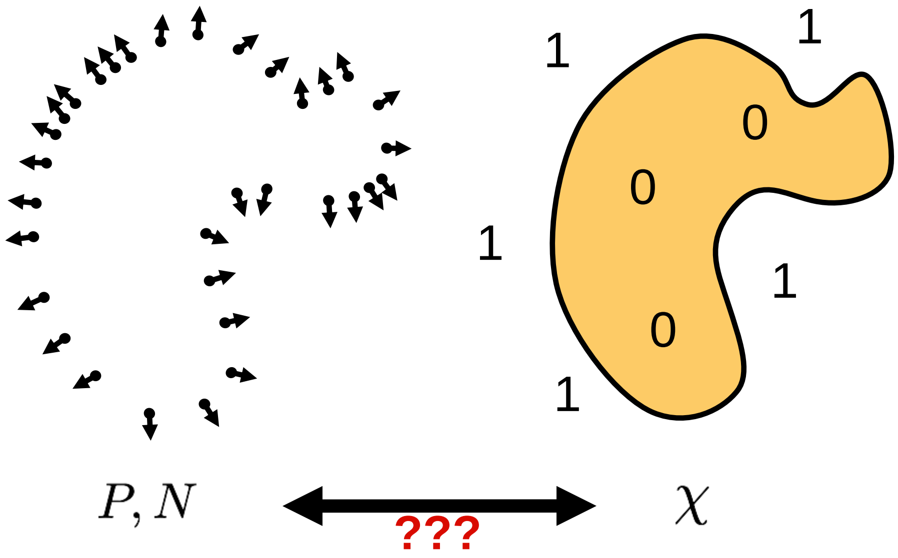


Oriented point set



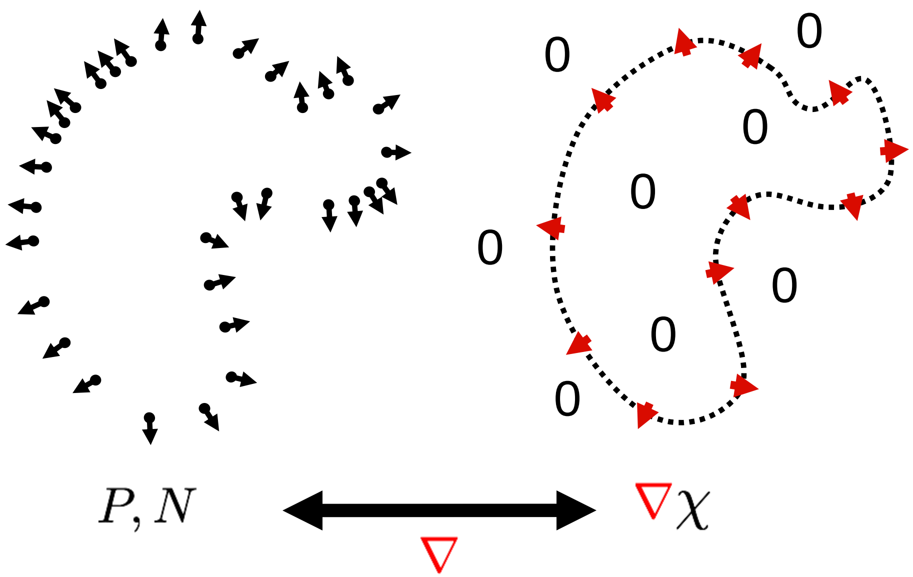
Indicator Function

Construct indicator function from point samples



Indicator Function

Construct indicator function from point samples



splatted normals

$$\min_{\chi} \int \|\nabla\chi(\mathbf{x}) - \boxed{\mathcal{N}(\mathbf{x})}\|_2^2 \, dx$$

variational calculus

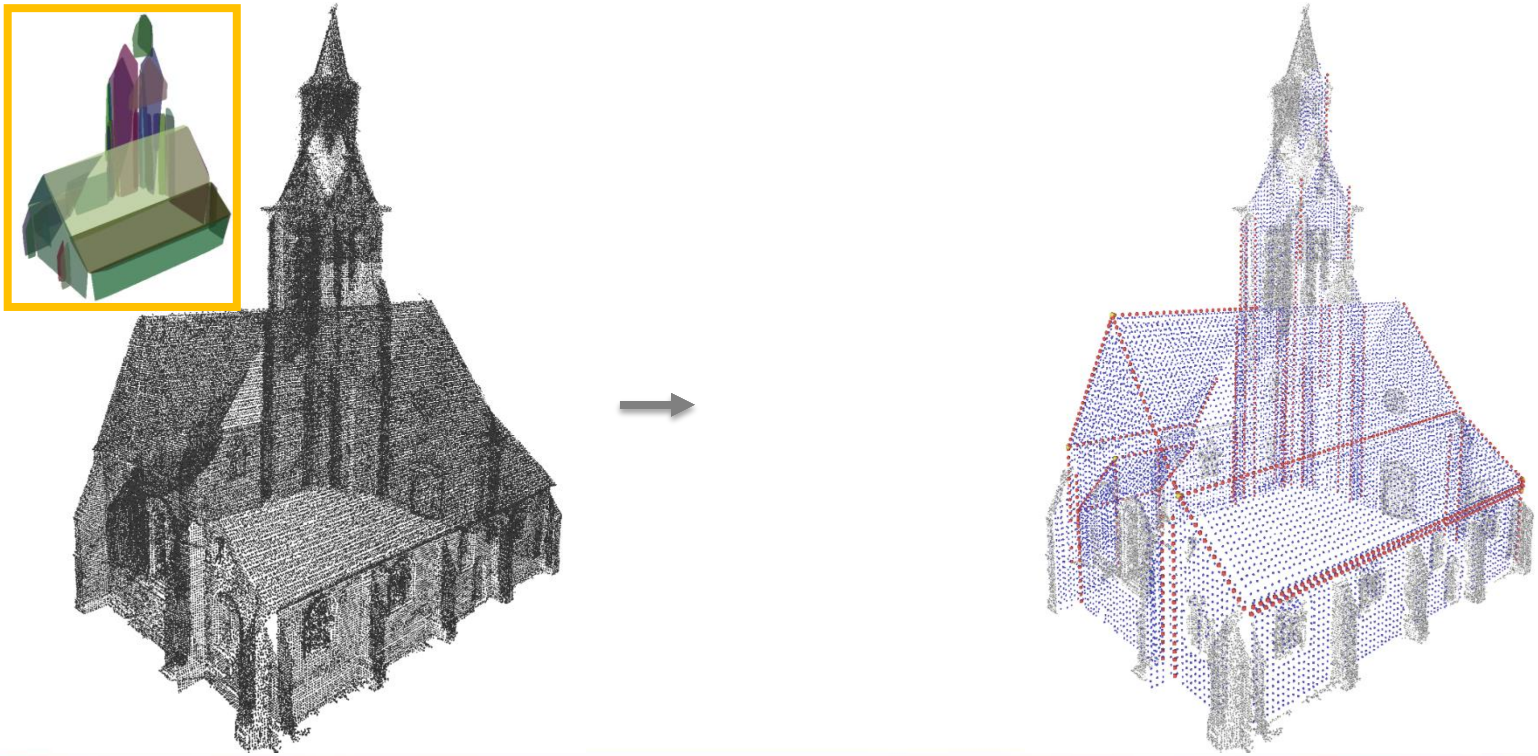
$$\boxed{\Delta\chi} = \nabla \cdot \mathcal{N}$$

sparse linear system

Primitive-based: Structuring

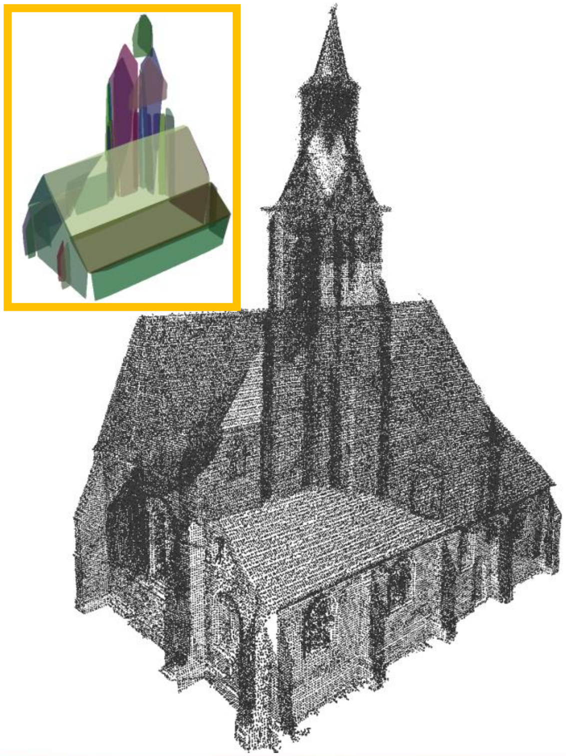
structuring

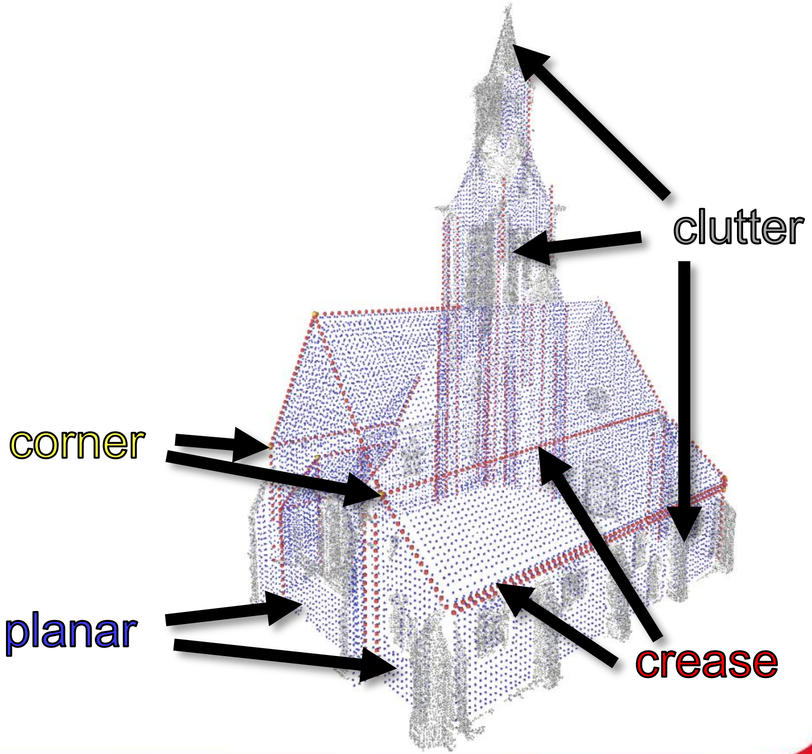
3 ideas



structuring

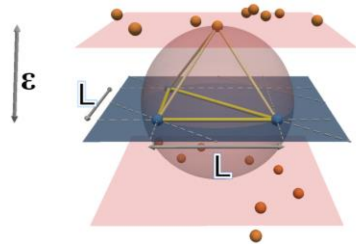
- 3 ideas
  - Meaning insertion



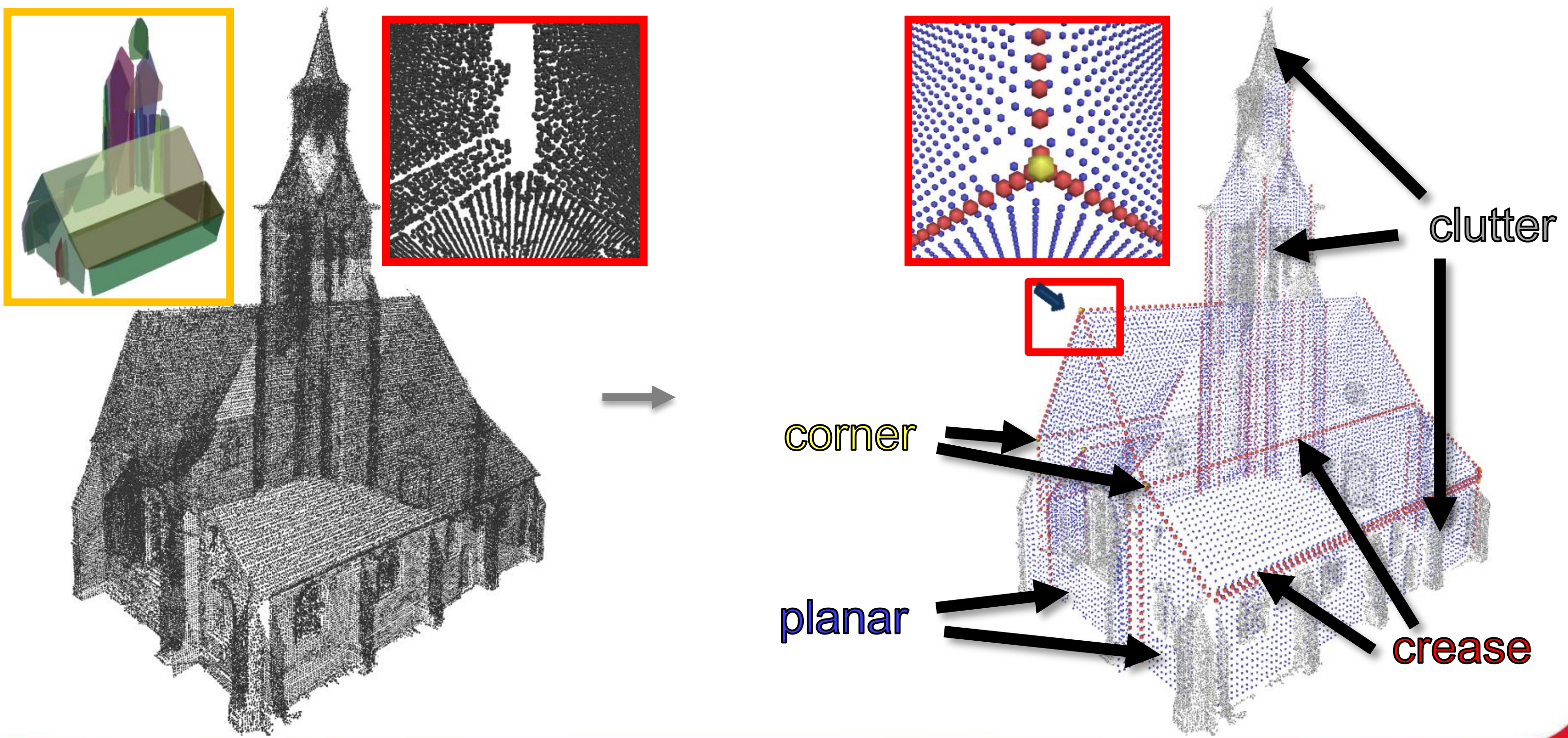


structuring

- 3 ideas
  - Meaning insertion
  - Structure idealization under Delaunay triangulation

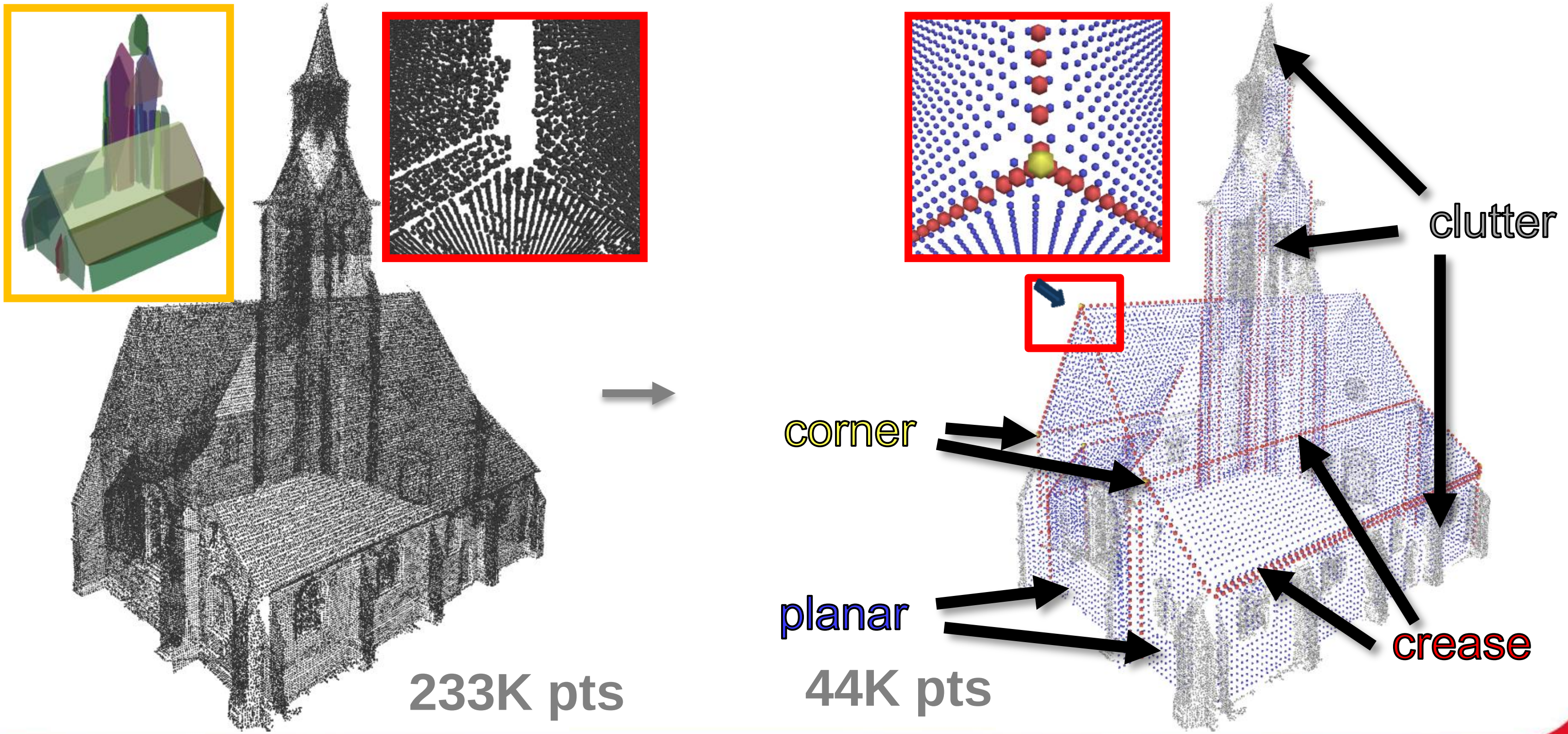


$$L_p < \sqrt{2}\,\varepsilon$$

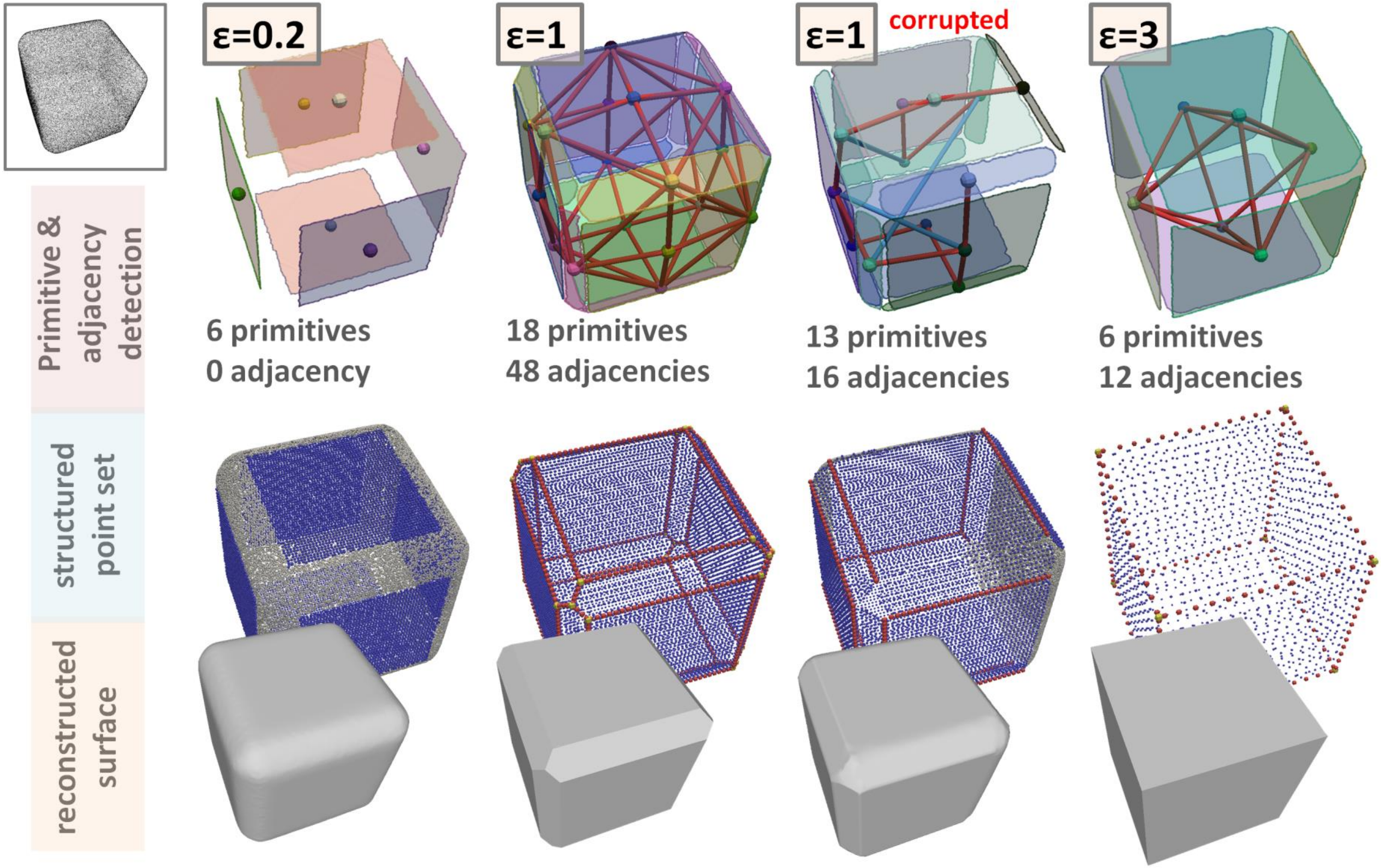


structuring

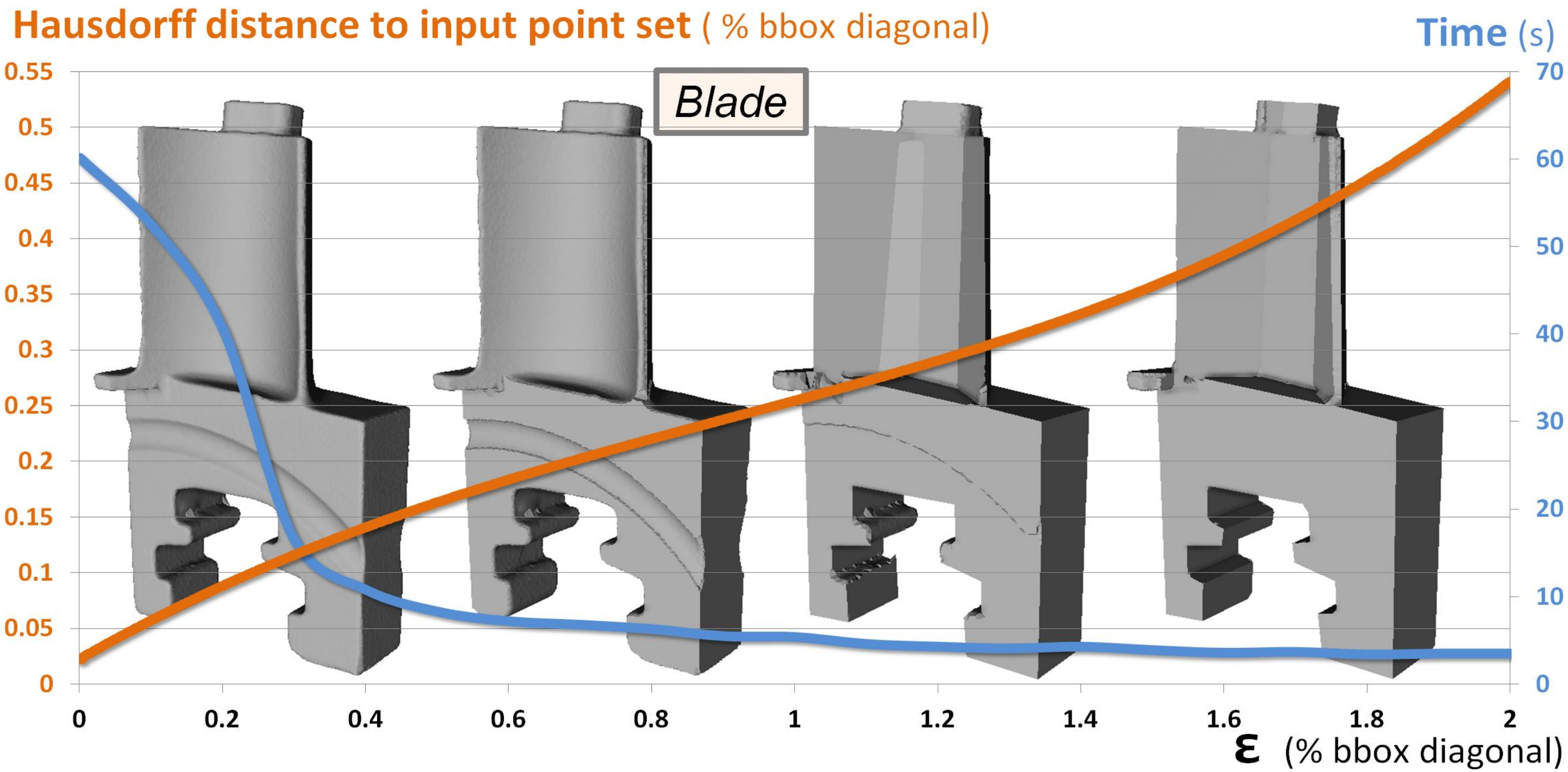
- 3 ideas
  - Meaning insertion
  - Structure idealization under Delaunay triangulation
  - Complexity reduction



structuring



structuring



structuring