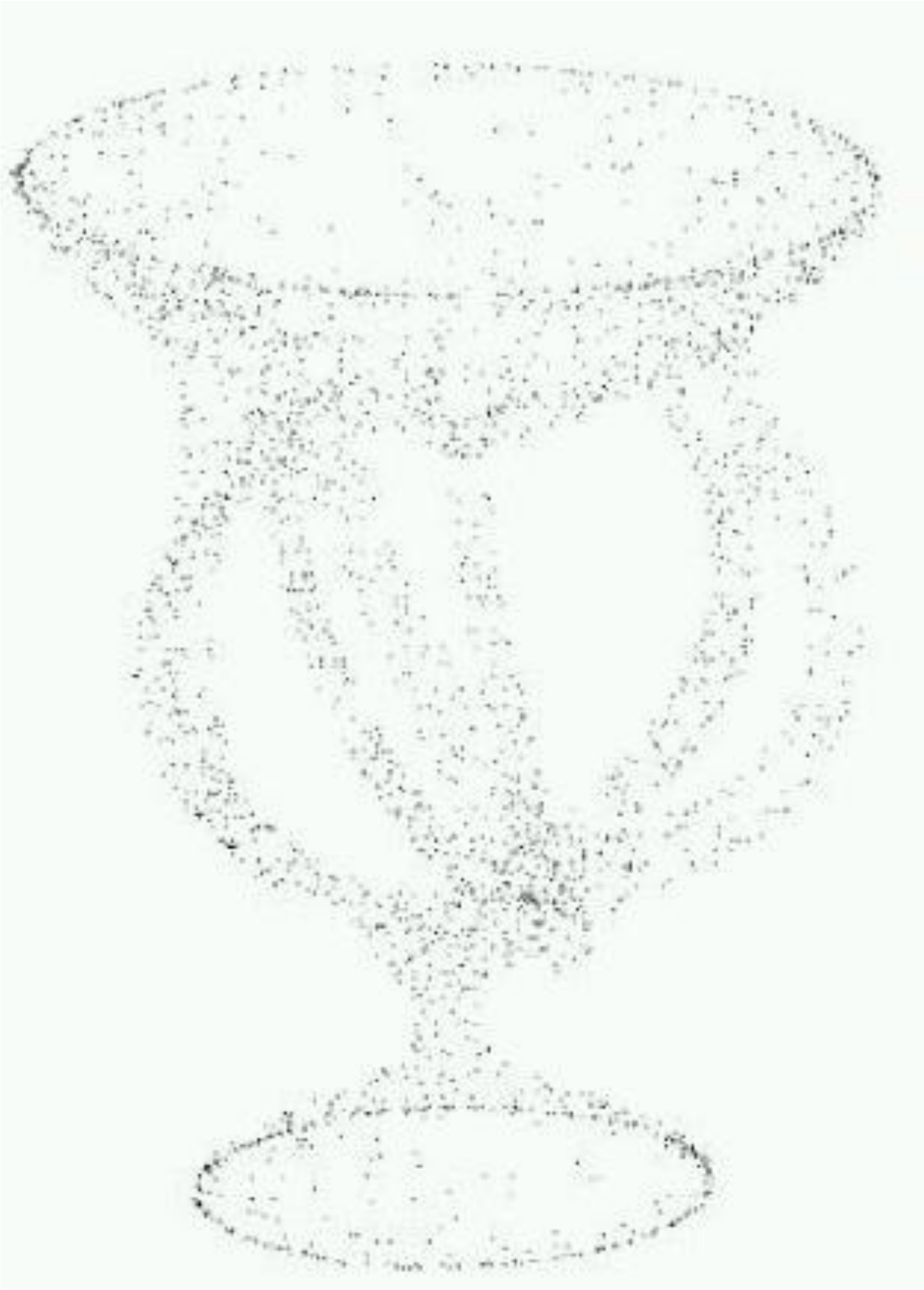
Delaunay Triangulation: Applications

Reconstruction

Meshing

Reconstruction

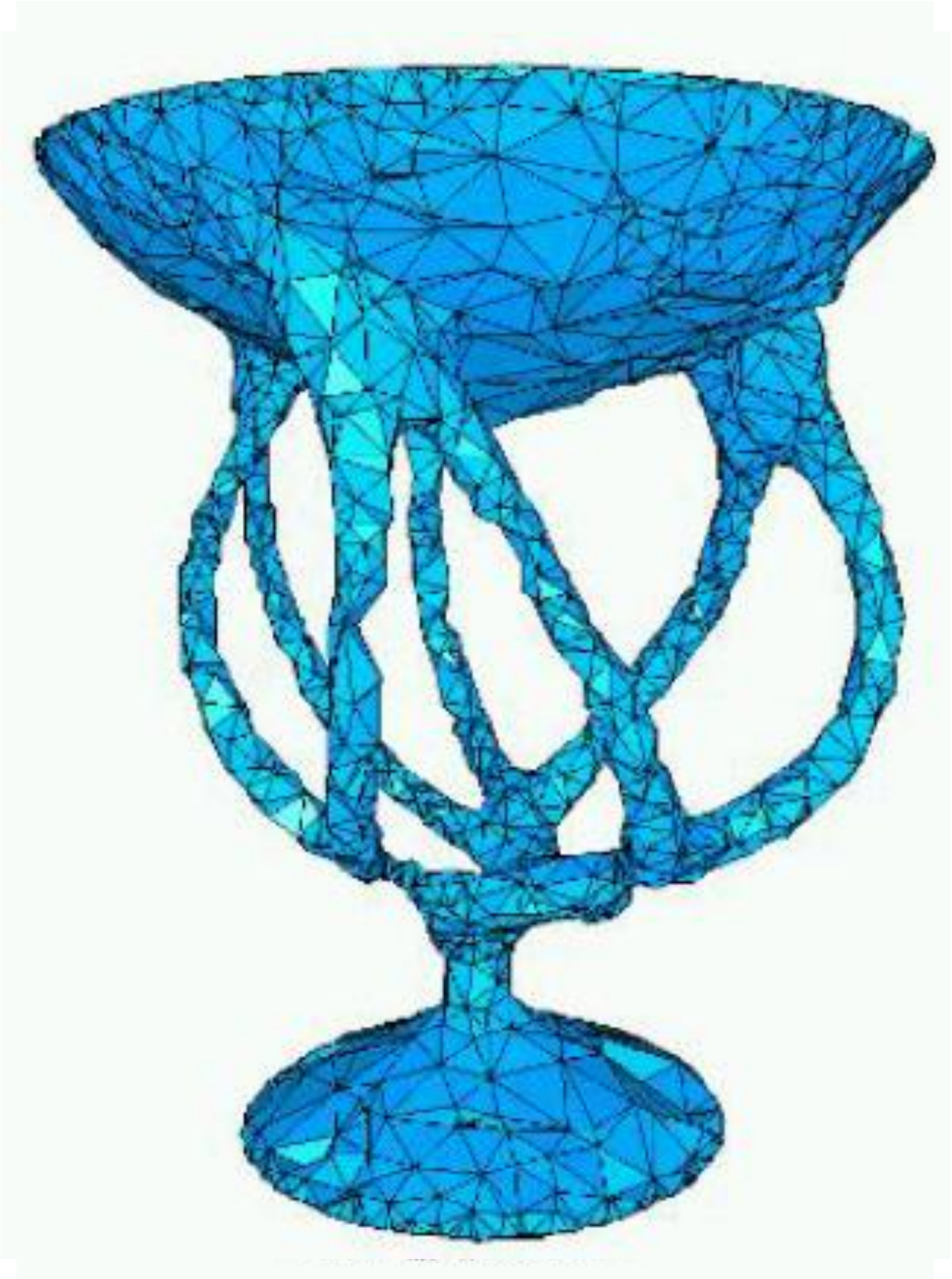
From points



Reconstruction

From points

to shape



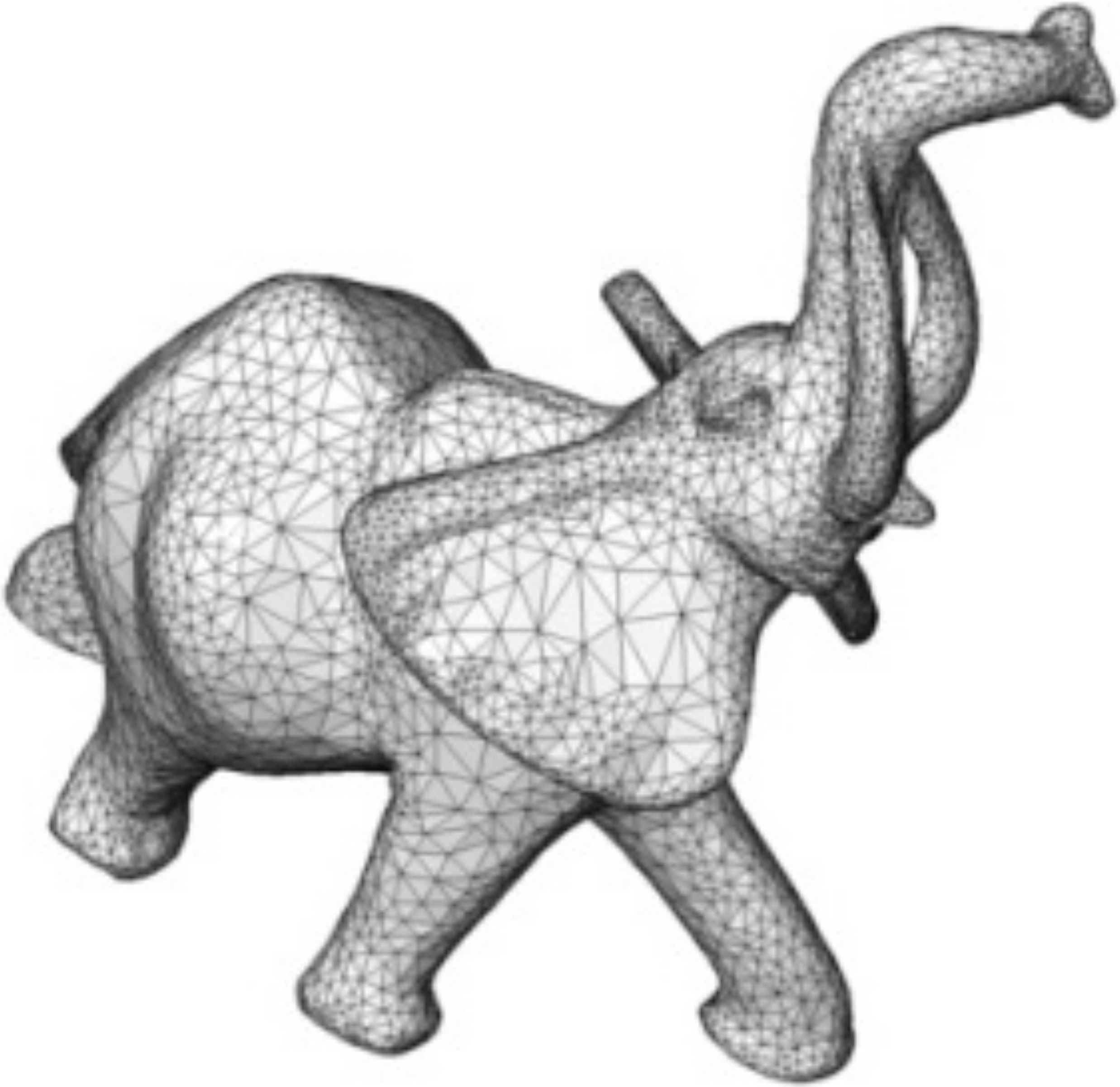
Reconstruction

From points

Reconstruction

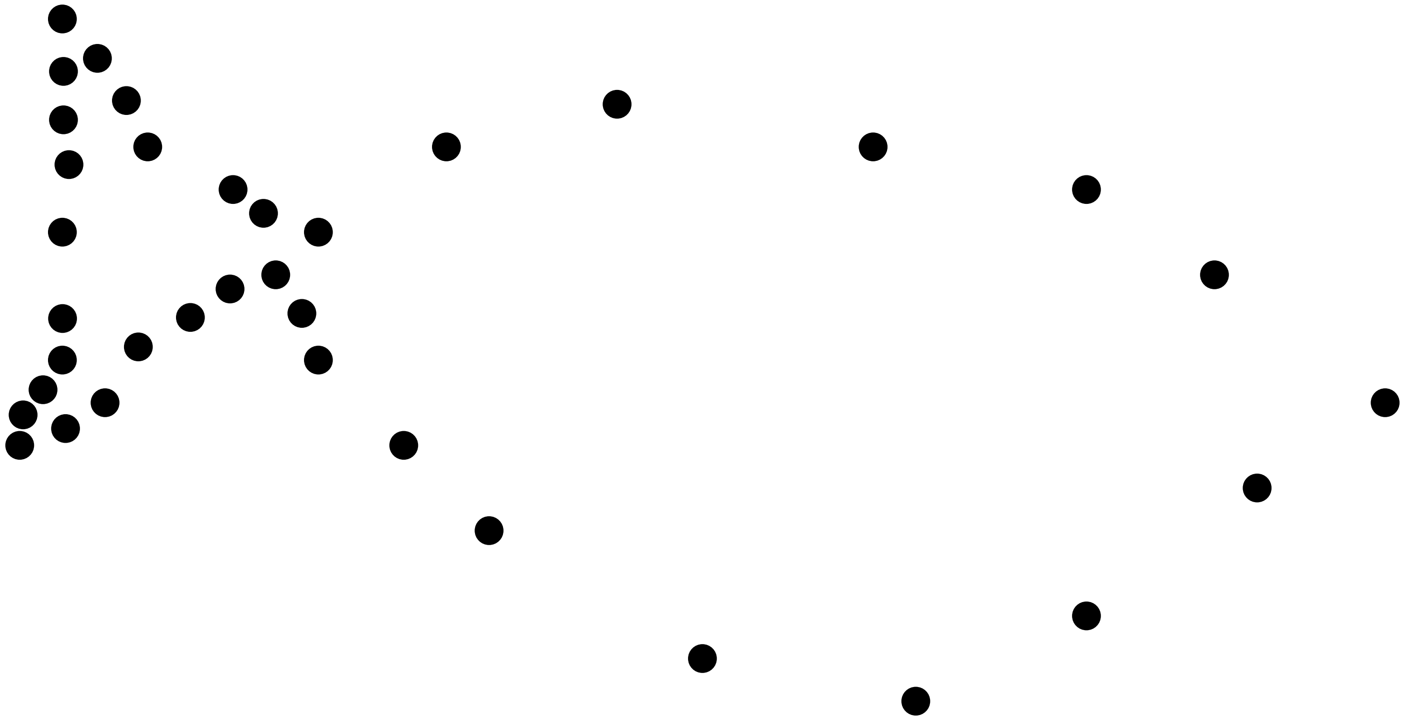
From points

to shape



Reconstruction

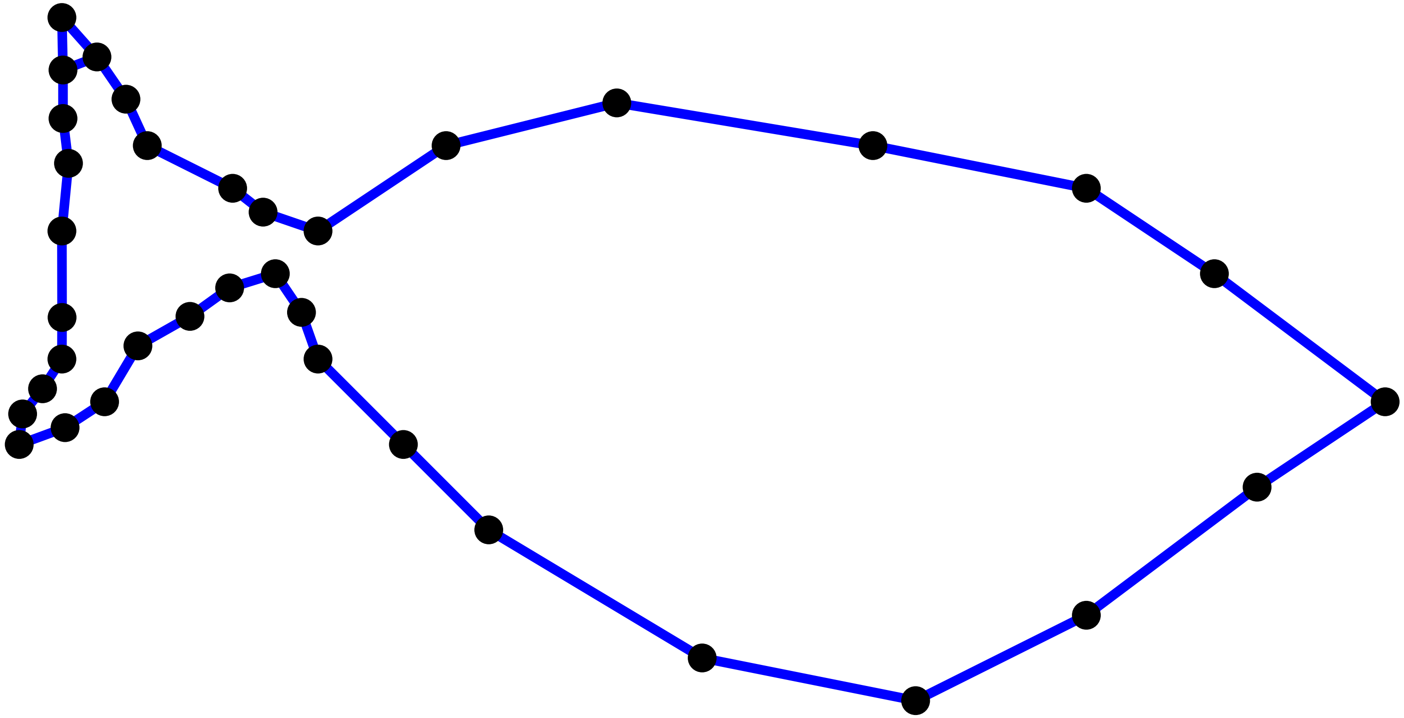
From points



Reconstruction

From points

to shape



Reconstruction

Context

Delaunay is a good start (wanted result $\subset$ Delaunay)

Crust 2D Algorithm

0.4 sample $\Rightarrow$ wanted result $\subset$ crust

0.25 sample $\Rightarrow$ crust $\subset$ wanted result

3D

Reconstruction

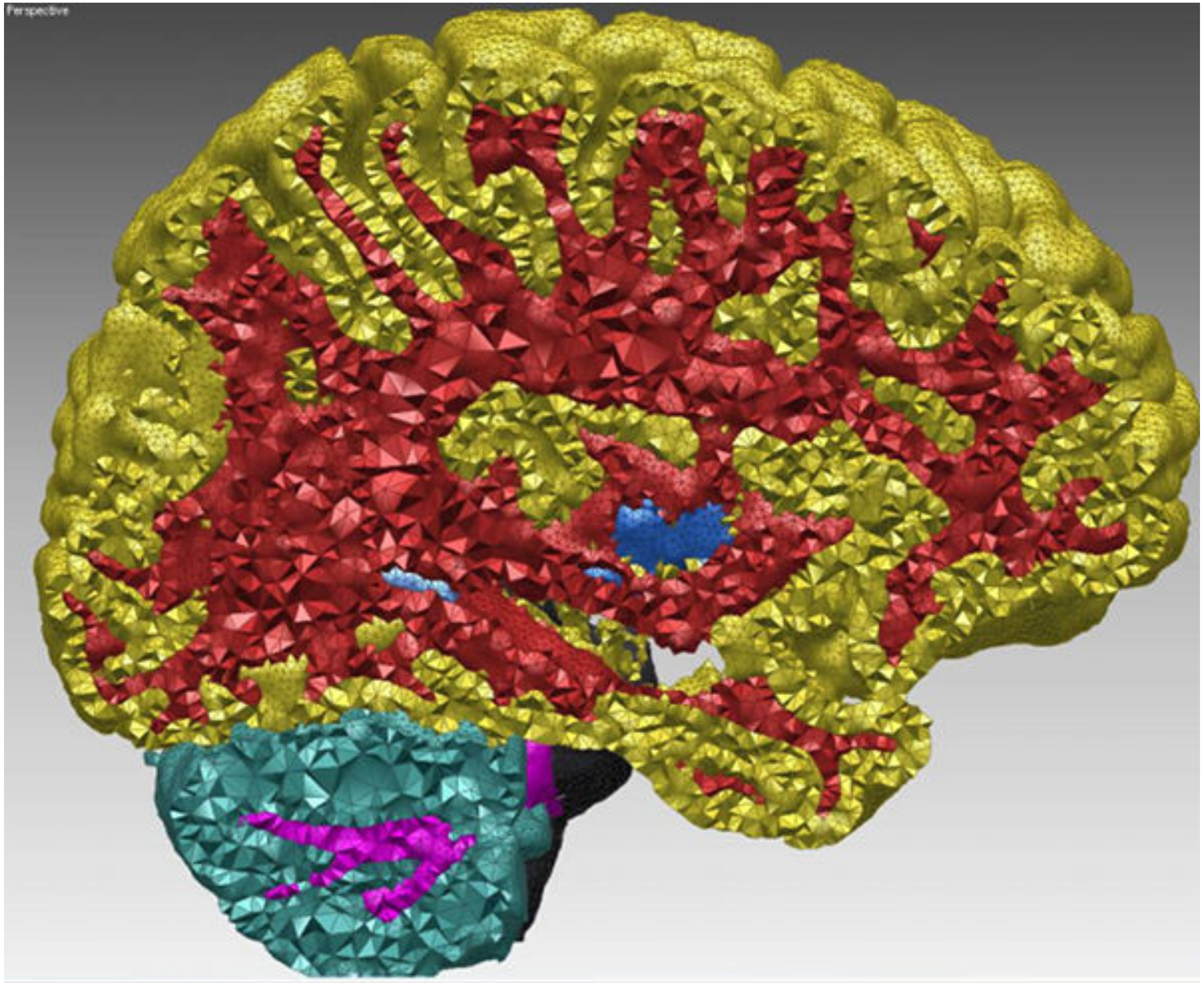
Context

Sensor ⟶ Point set (no structure or unknown)

Reconstruction

Context

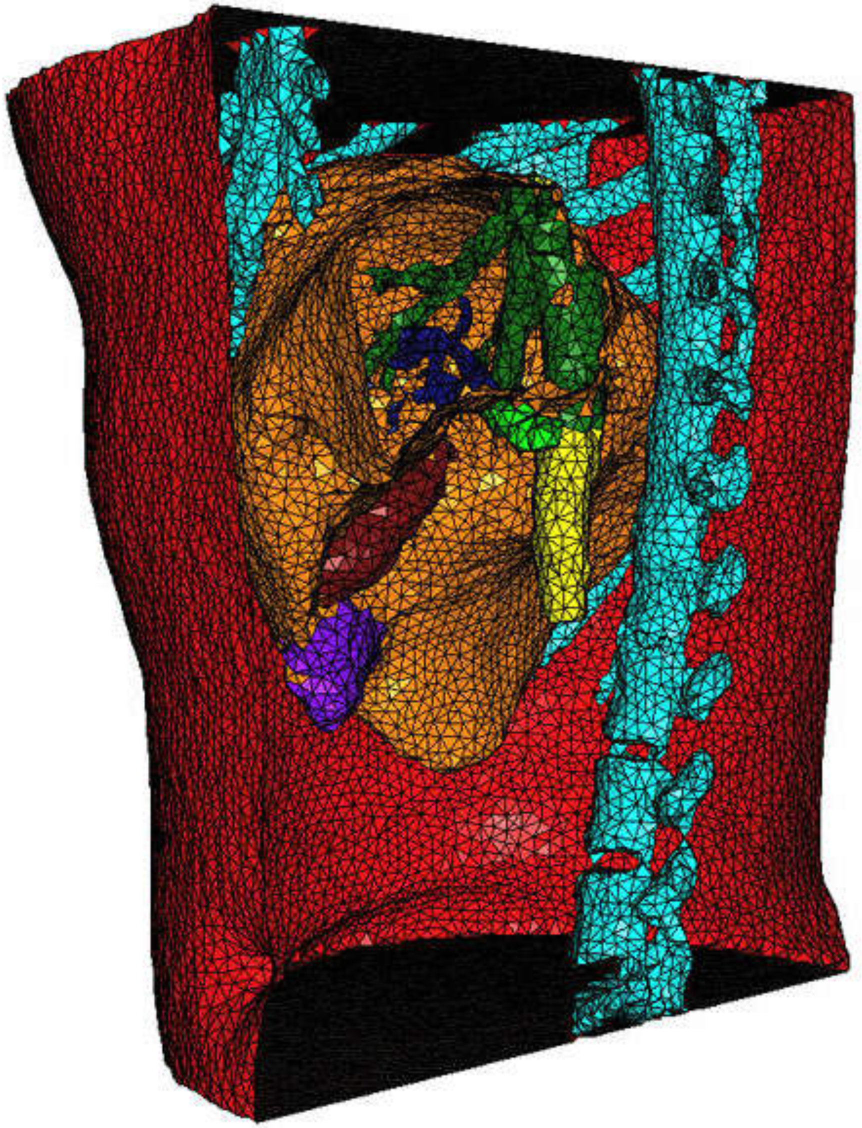
Medical Images



Reconstruction

Context

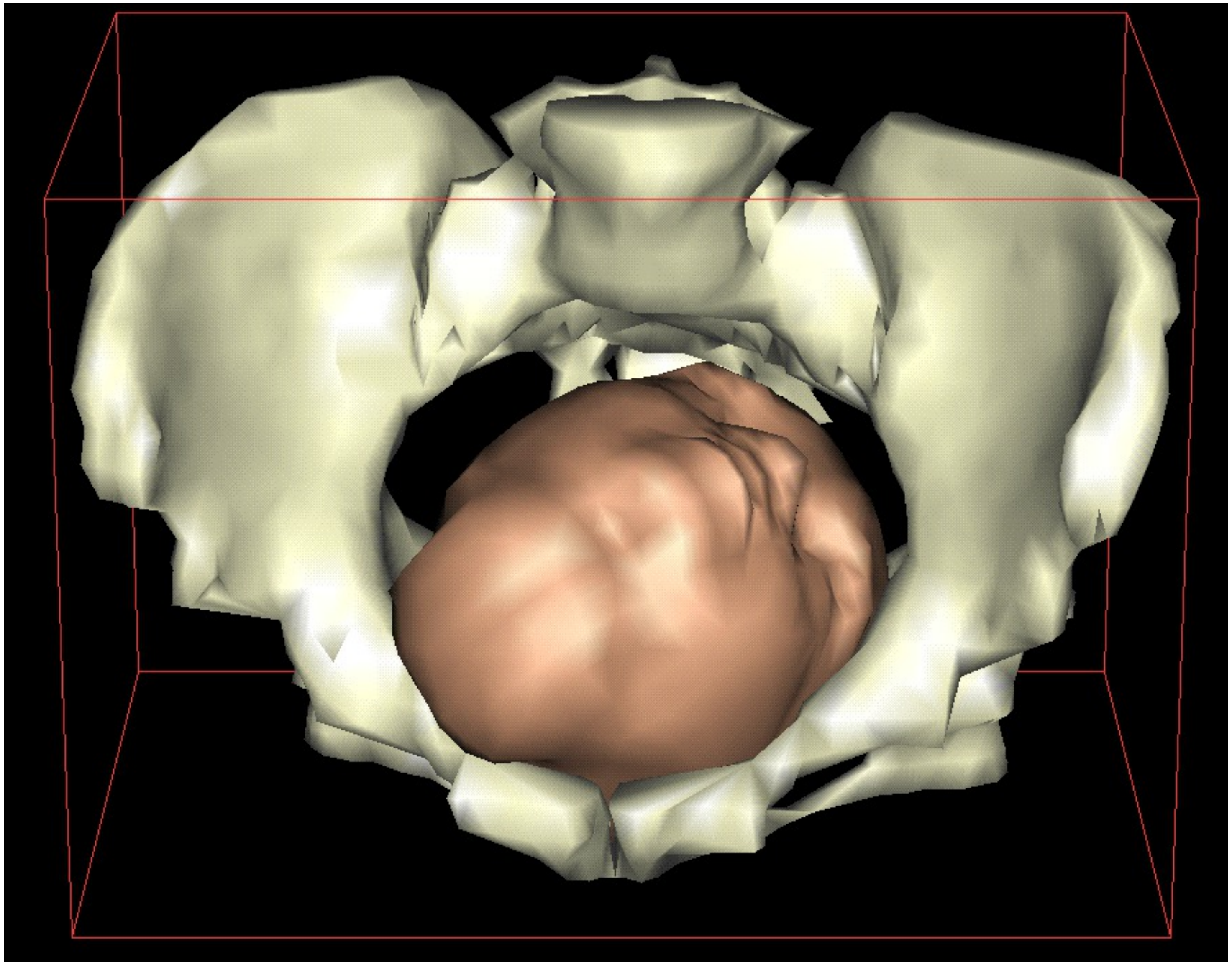
Medical Images



Reconstruction

Context

Childbirth simulation



Reconstruction

Context

Childbirth simulation

Surgery planning

Radiotherapy planing

Endoscopy simulation

• • •

Reconstruction

Context

Sensor → Point set (no structure or unknown)

Scanner

Reconstruction

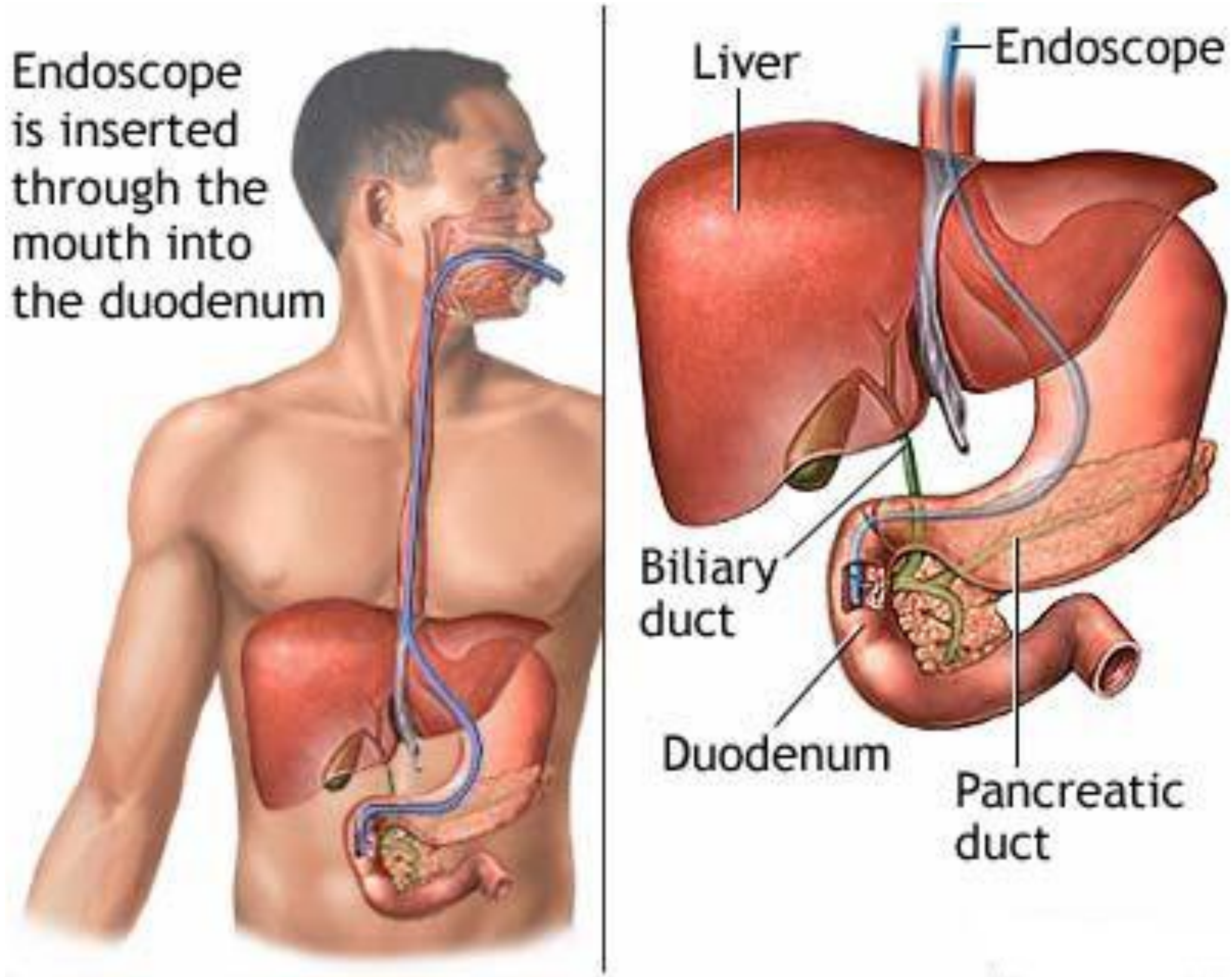
Context

Sensor → Point set (no structure or unknown)

Scanner

Endoscope

• • •

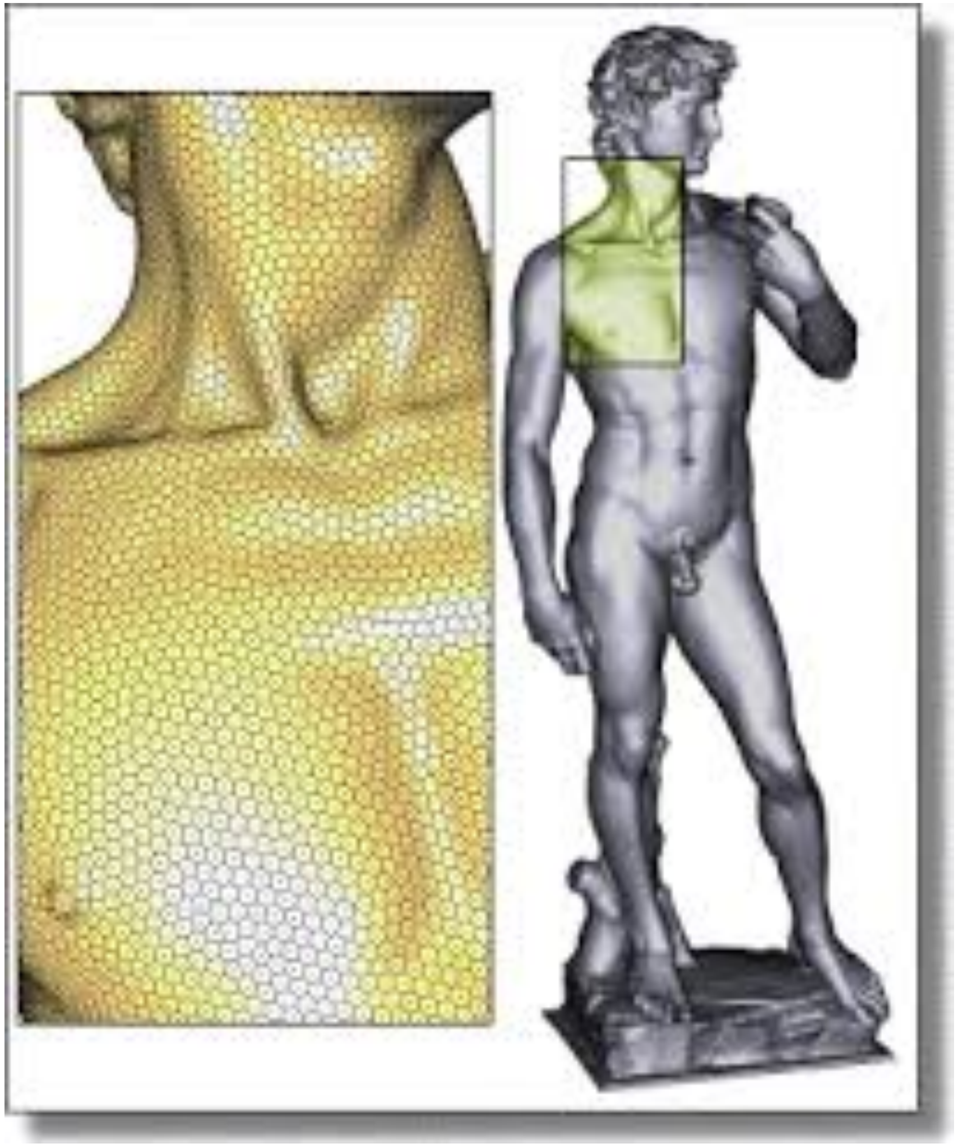


Reconstruction

Cultural heritage

Reconstruction

Cultural heritage

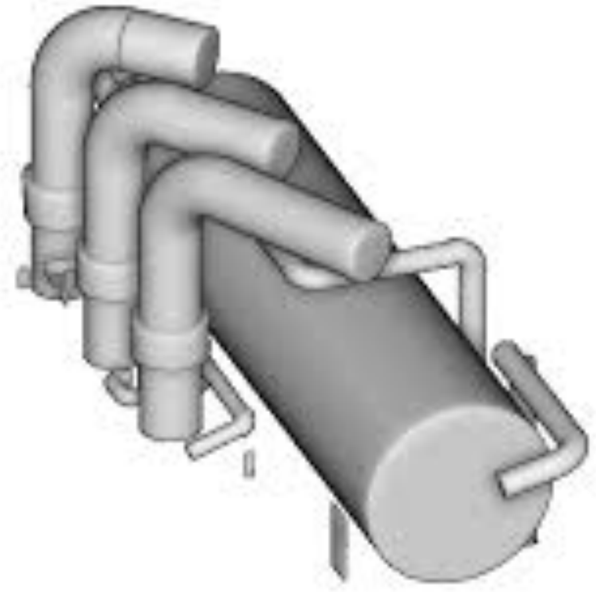


Reconstruction Context

Reconstruction

Context

Reverse engineering



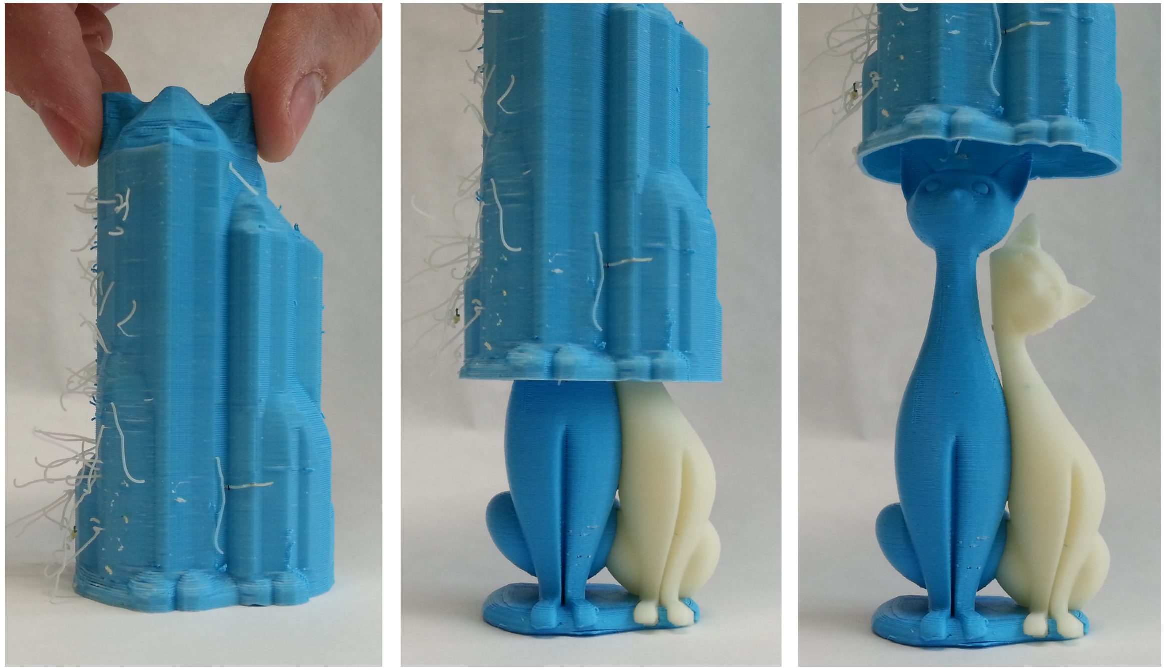
Reconstruction

Context

Reverse engineering

Prototyping (3D print)

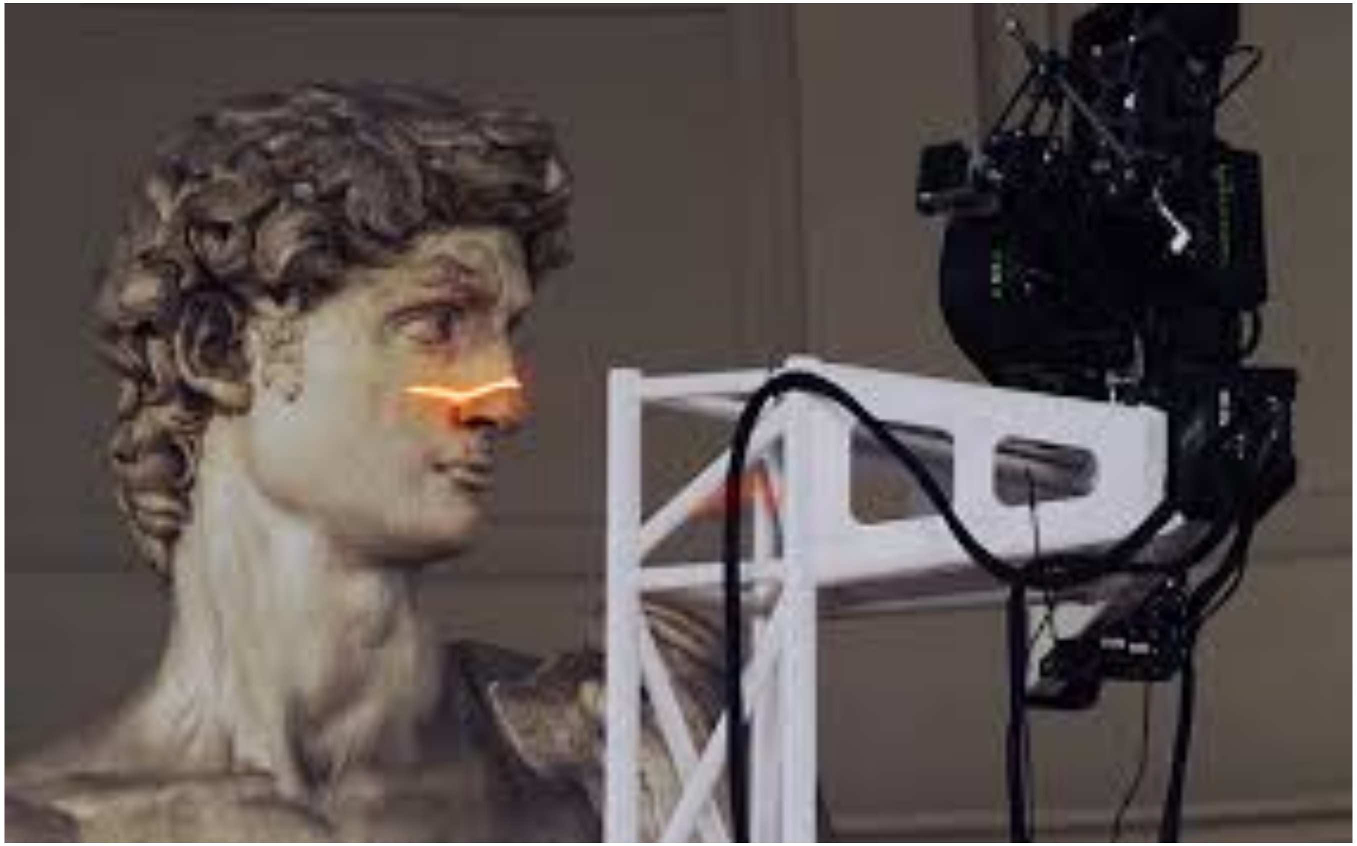
Quality control



Reconstruction

Context

Sensor → Point set (no structure or unknown)

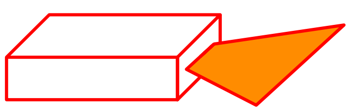


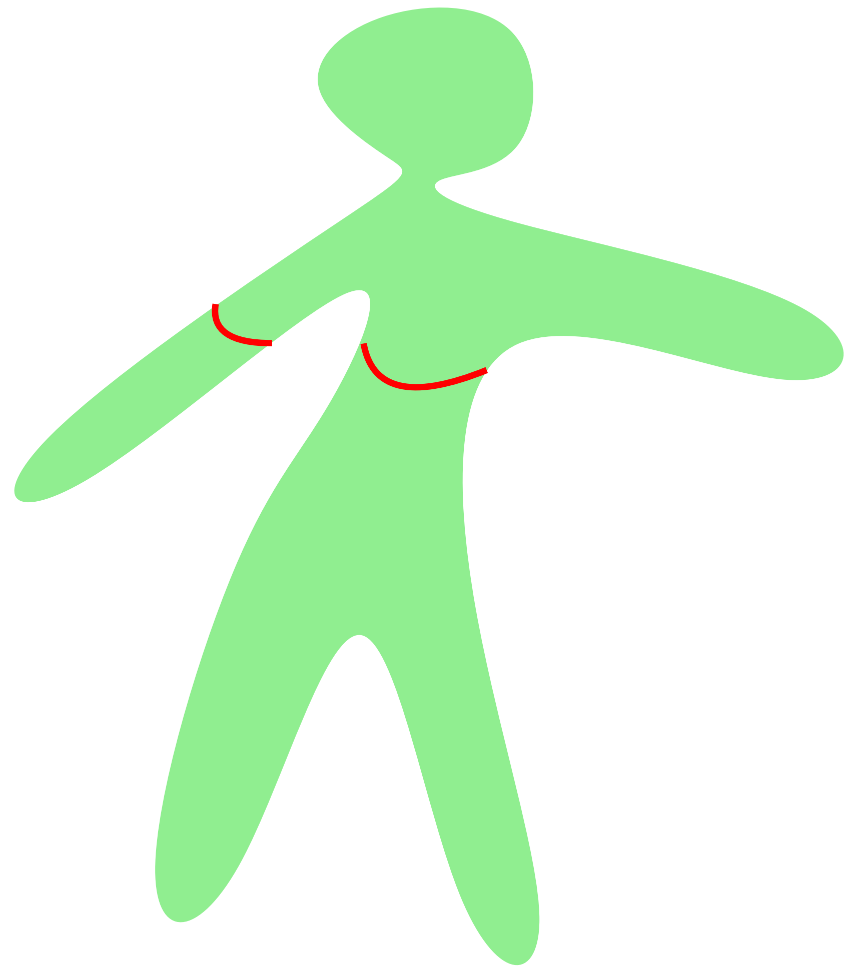
Reconstruction

Context

Sensor → Point set (no structure or unknown)

Laser illuminate in a plane





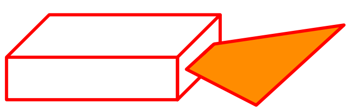
Reconstruction

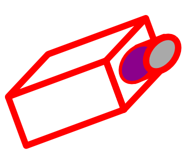
Context

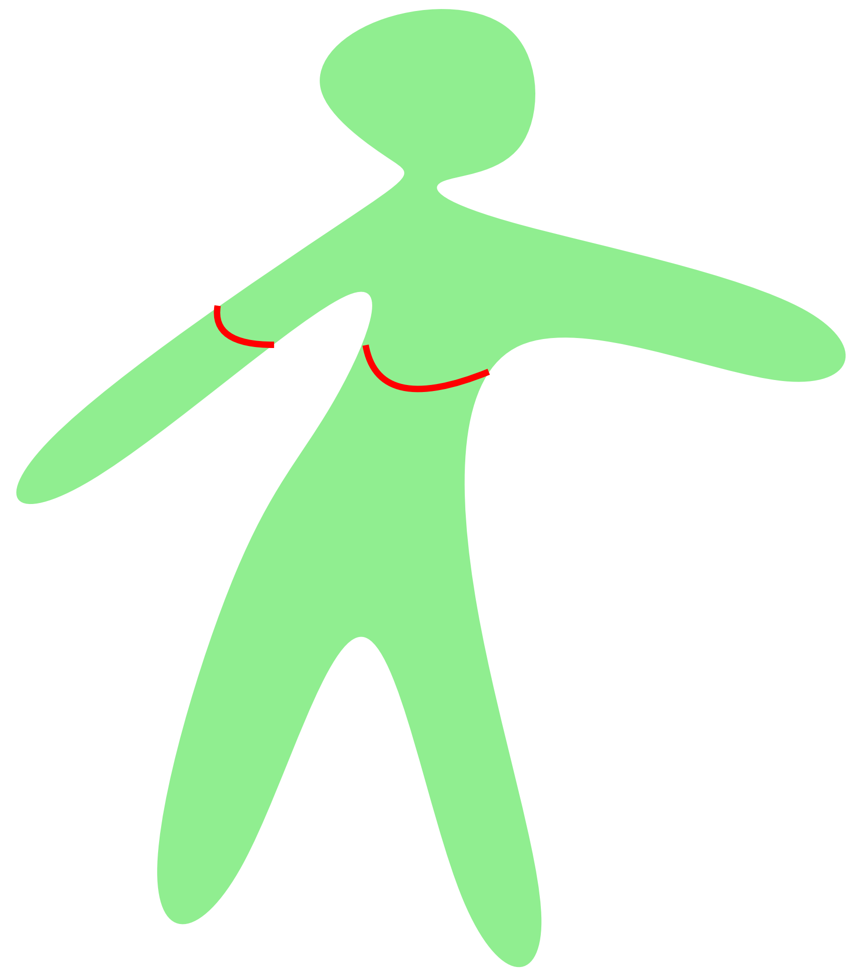
Sensor → Point set (no structure or unknown)

Laser illuminate in a plane

Camera







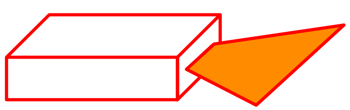
Reconstruction

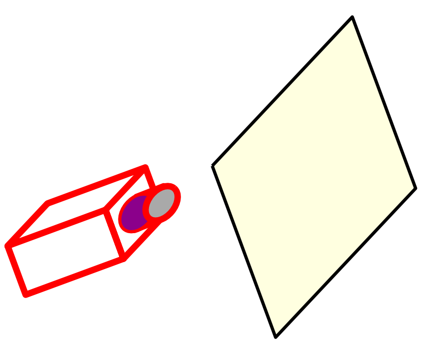
Context

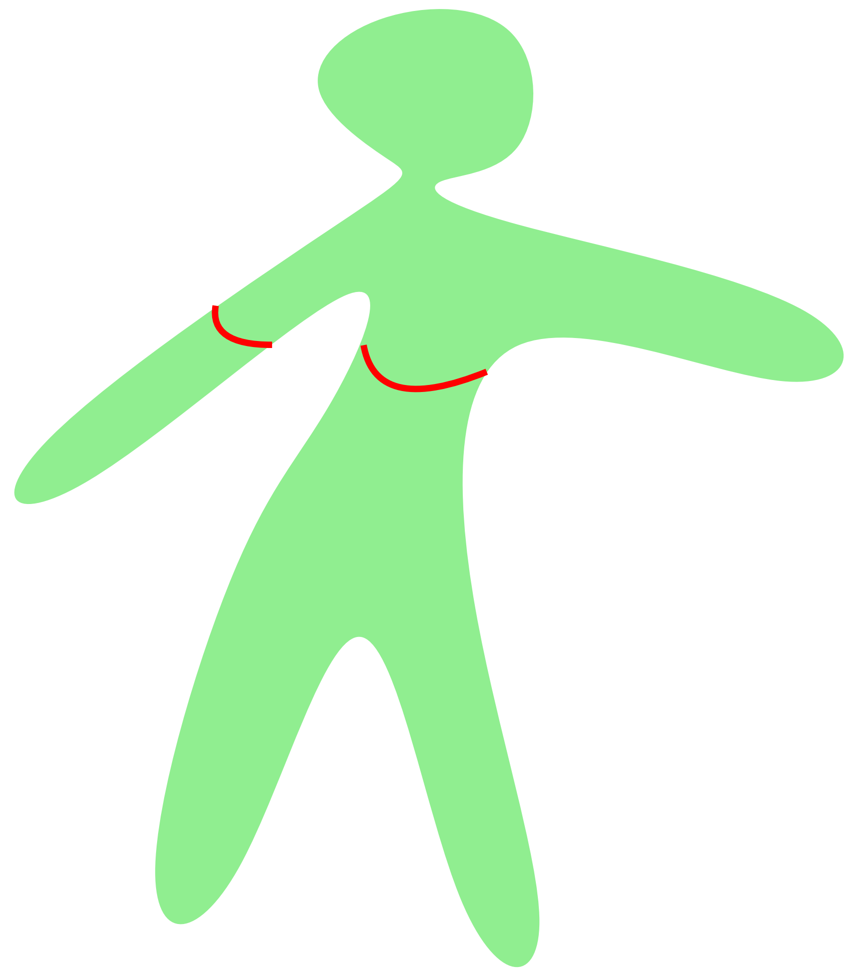
Sensor → Point set (no structure or unknown)

Laser illuminate in a plane

Camera







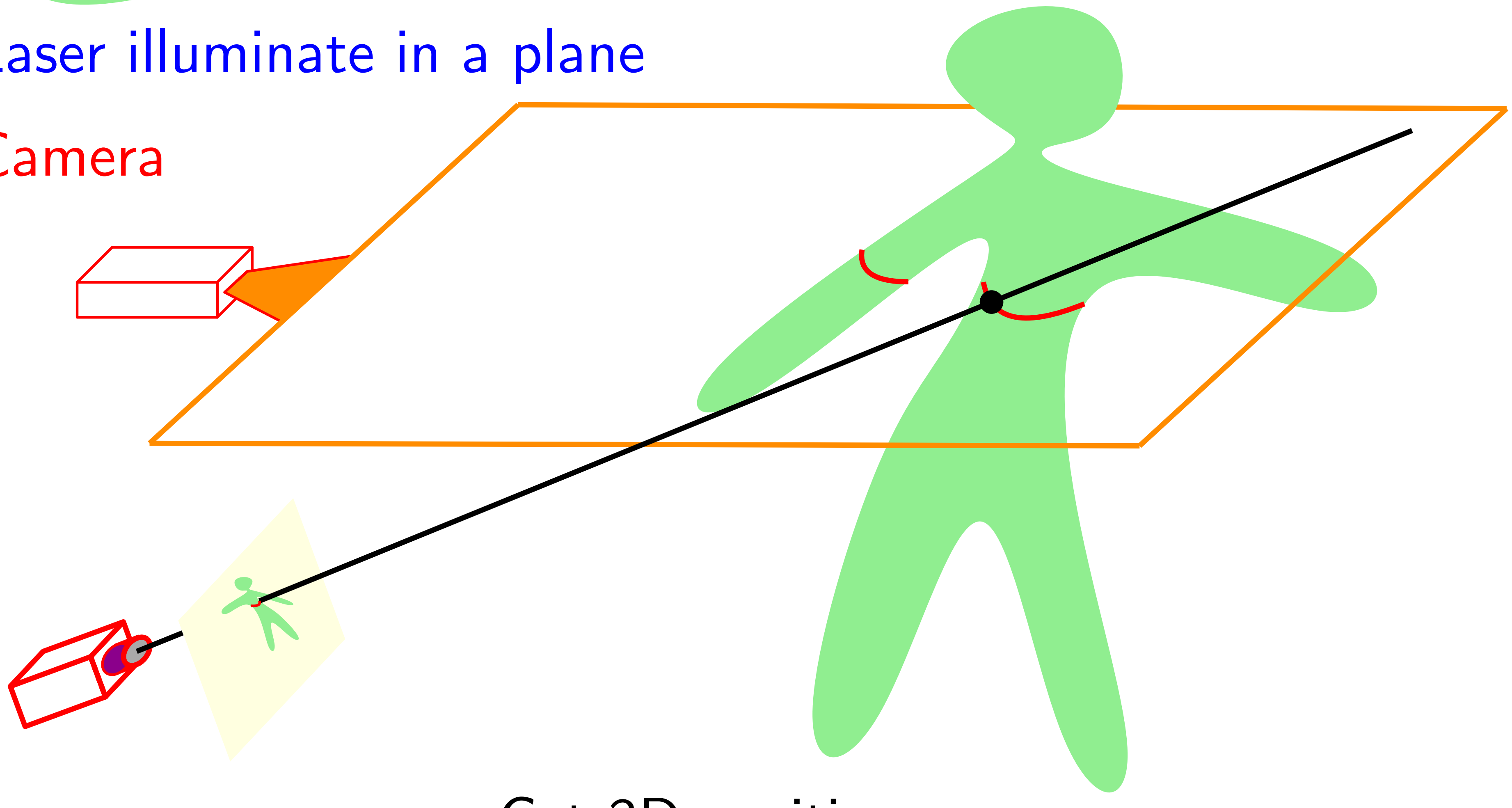
Reconstruction

Context

Sensor → Point set (no structure or unknown)

Laser illuminate in a plane

Camera

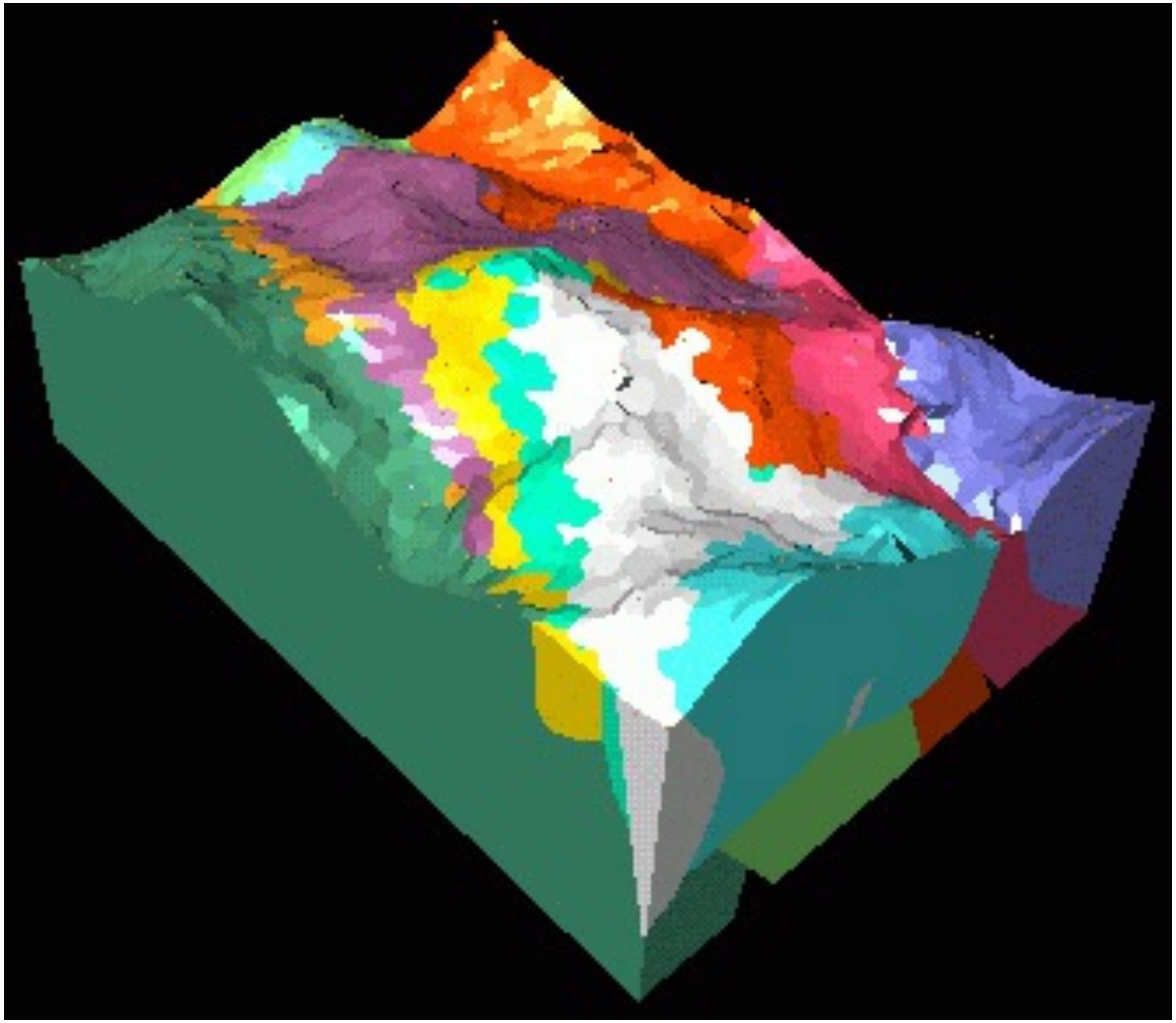


Get 3D position

Reconstruction

Context

Geology

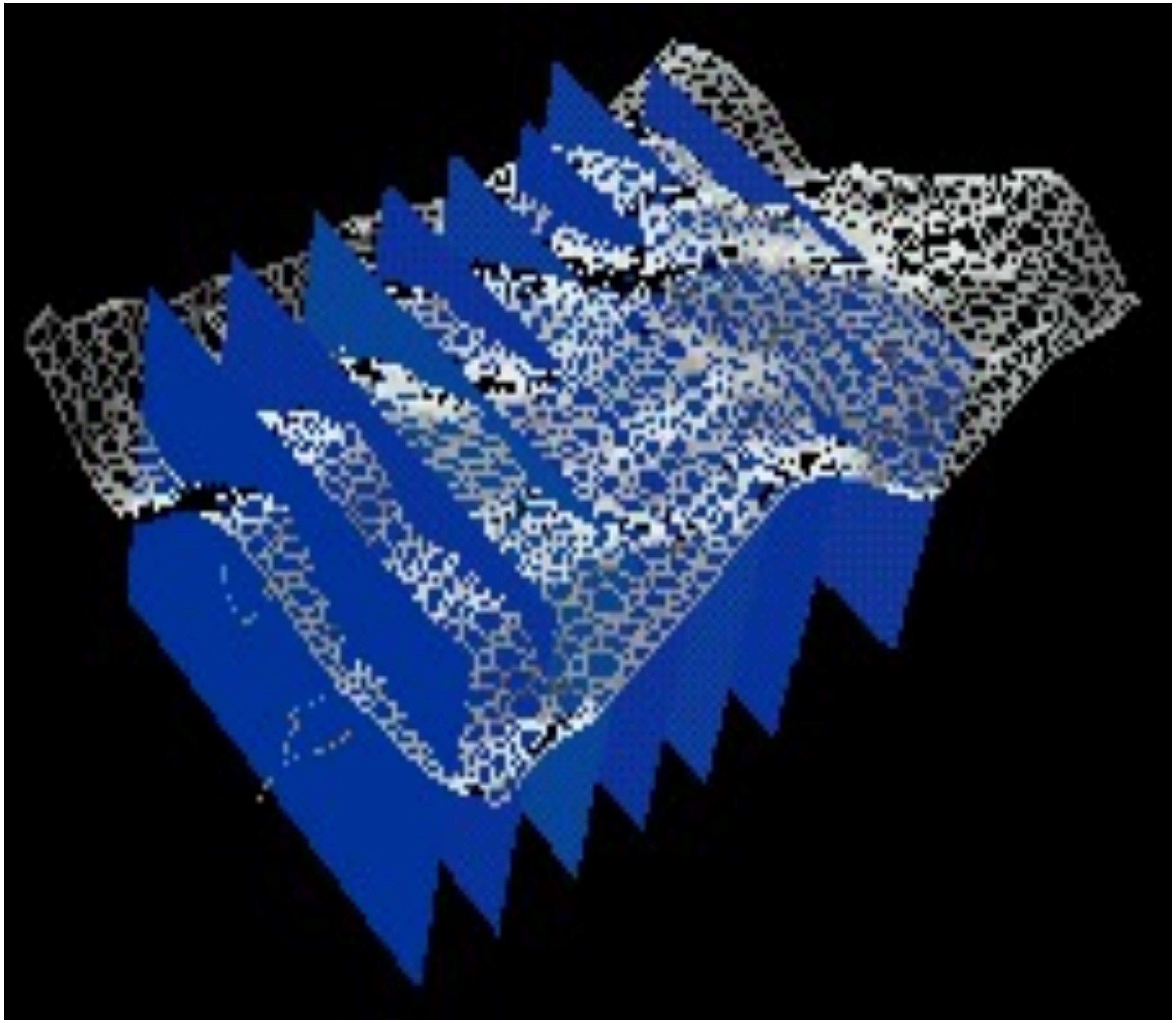


Reconstruction

Context

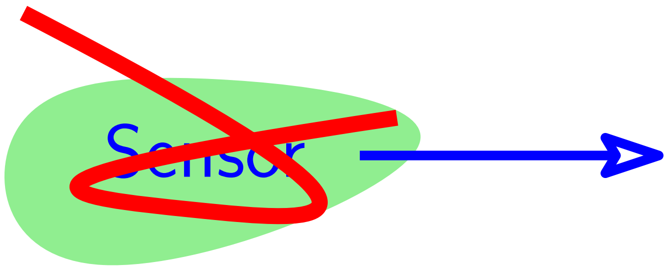
Sensor → Point set (no structure or unknown)

Geology



Reconstruction

Context

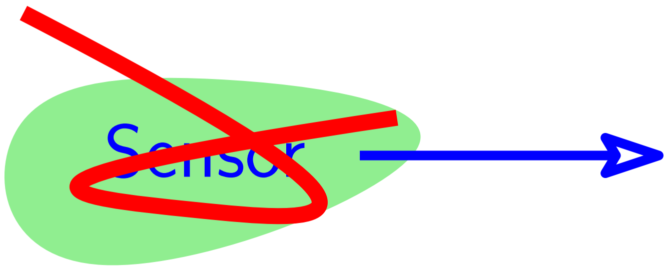


Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

Reconstruction

Context

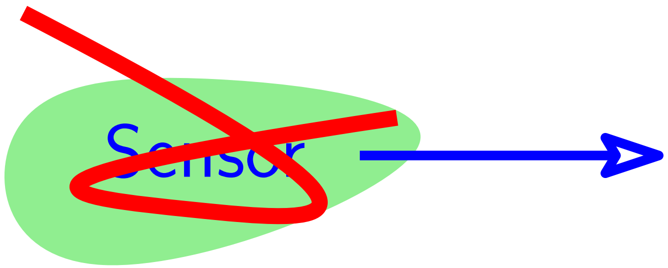


Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

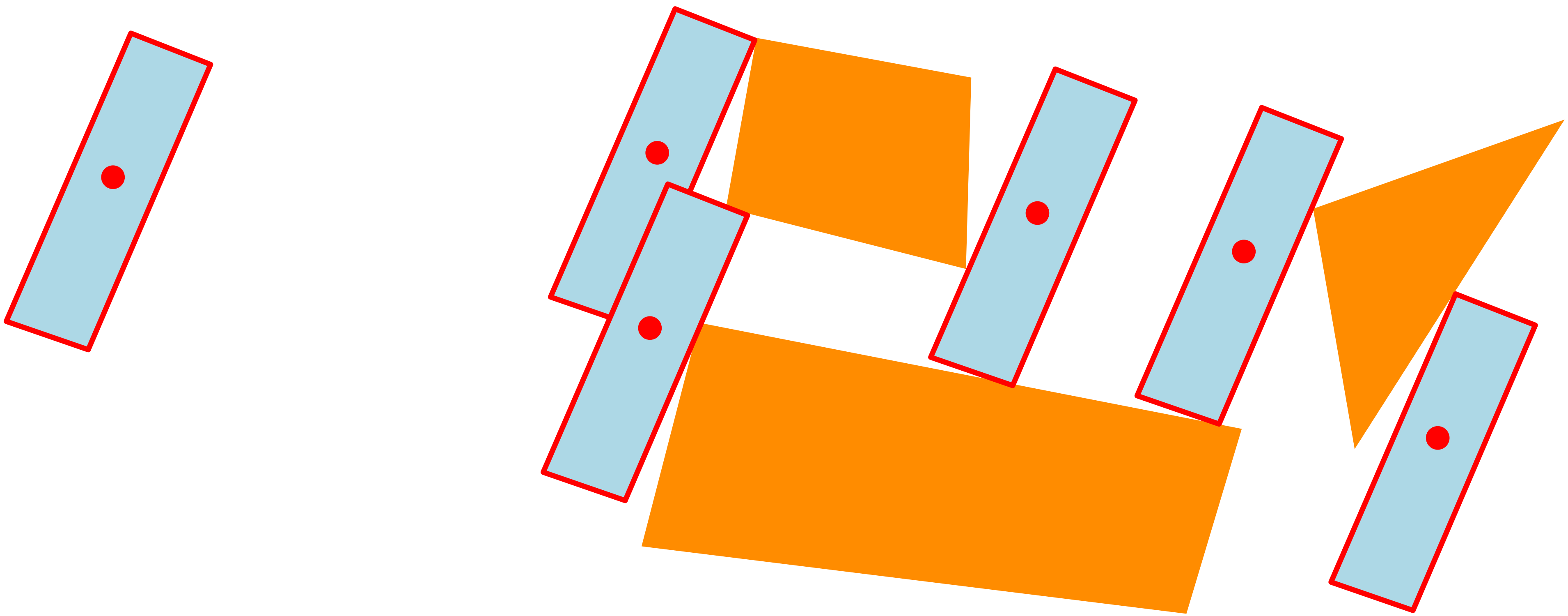
Reconstruction

Context



Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

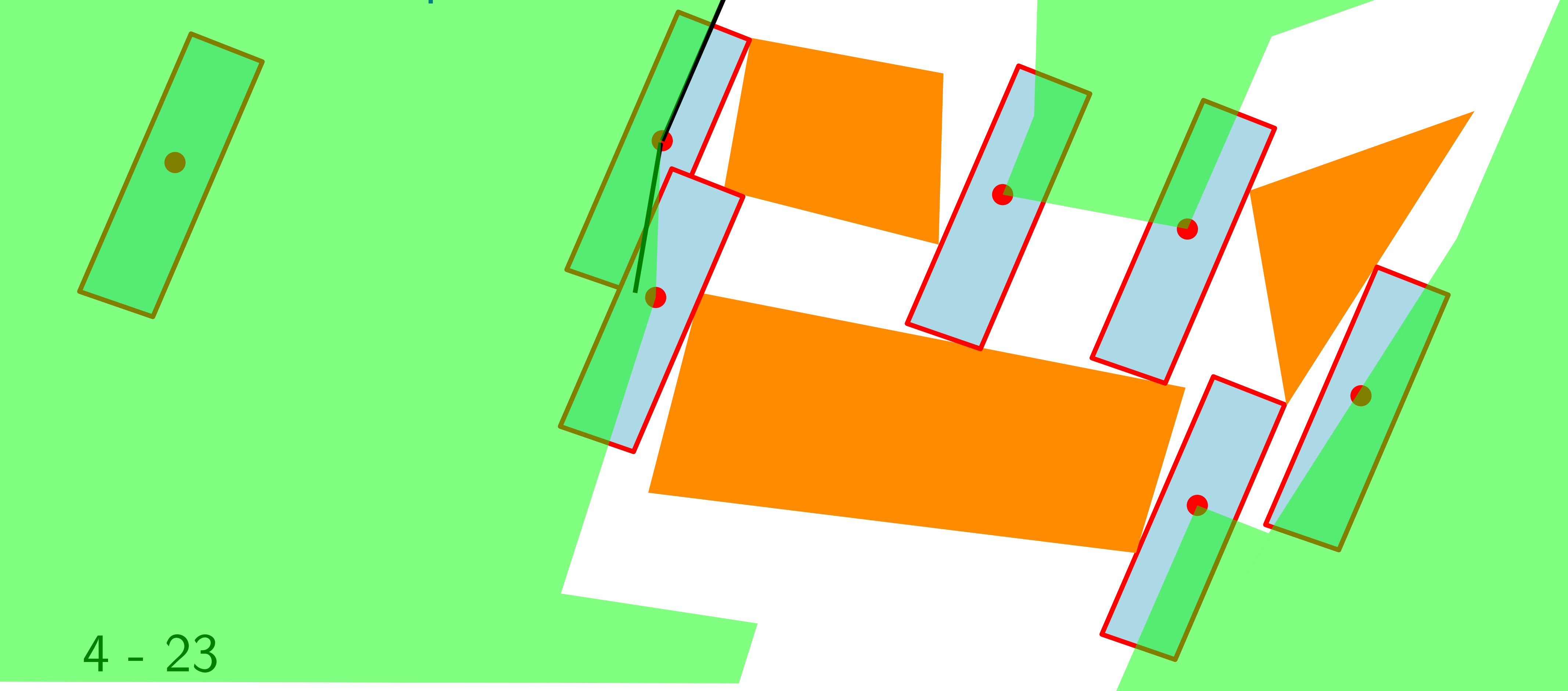


Reconstruction

Context

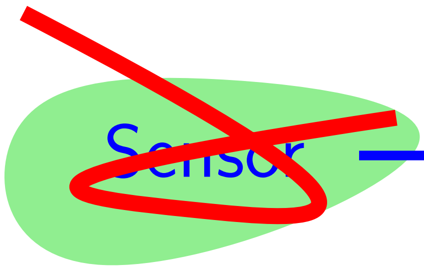
Sensor → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section



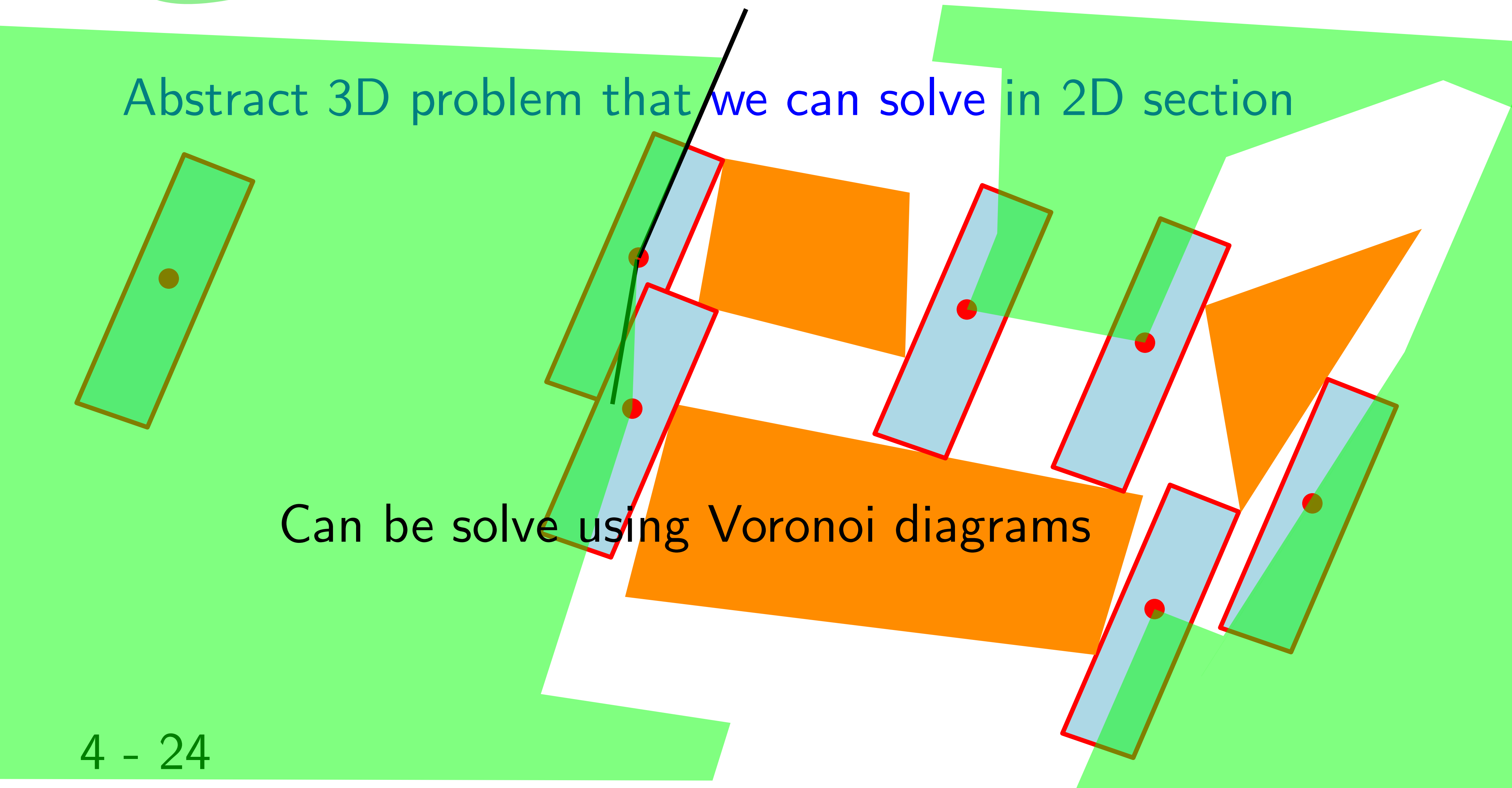
Reconstruction

Context



→ Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section



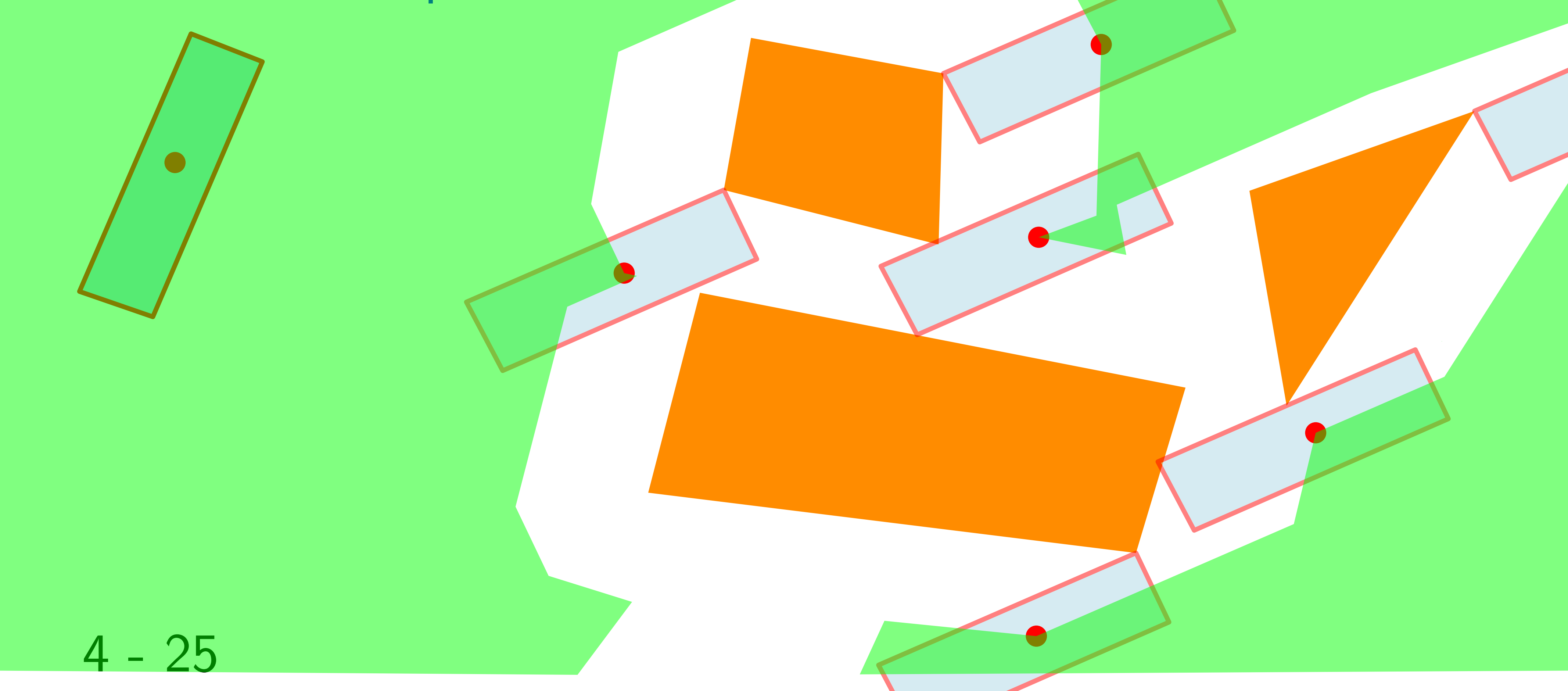
Can be solve using Voronoi diagrams

Reconstruction

Context

~~Sensor~~ → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

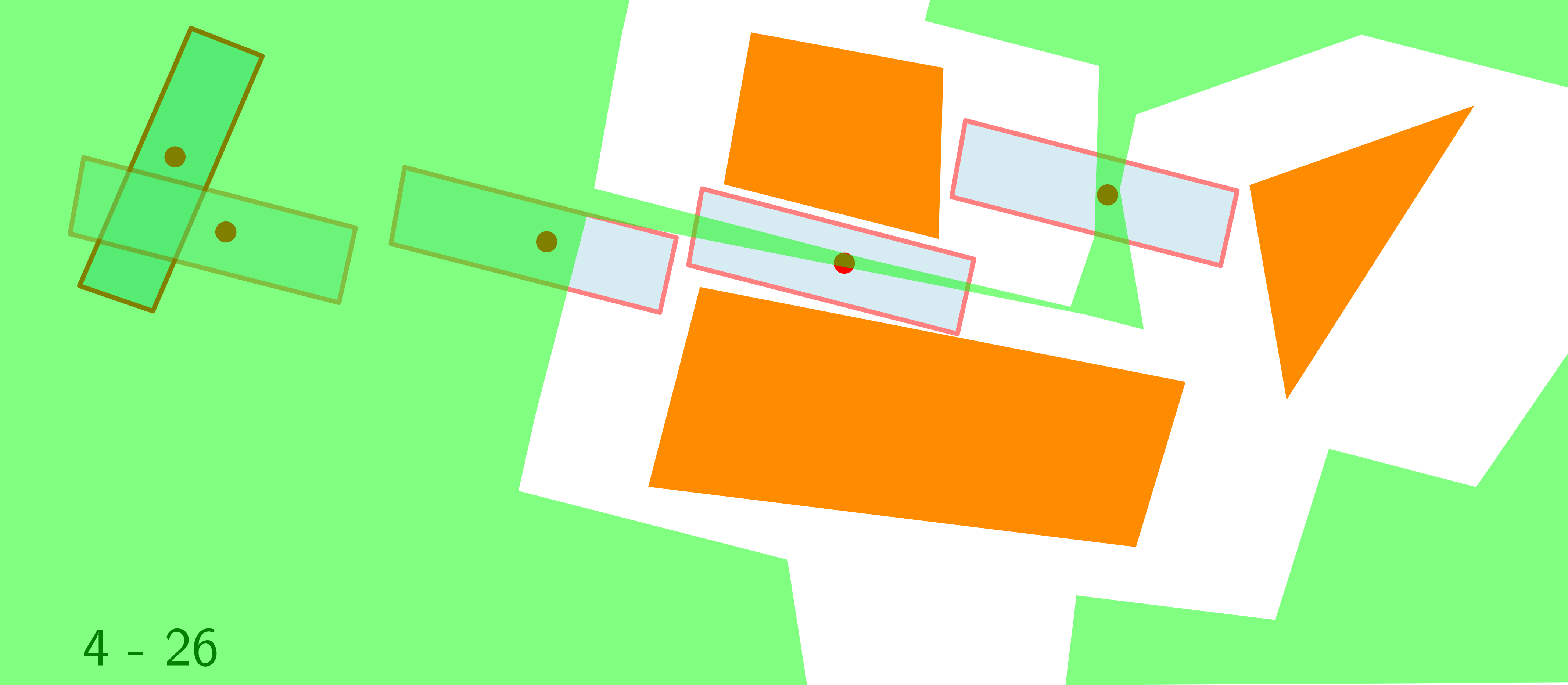


Reconstruction

Context

~~Sensor~~ → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

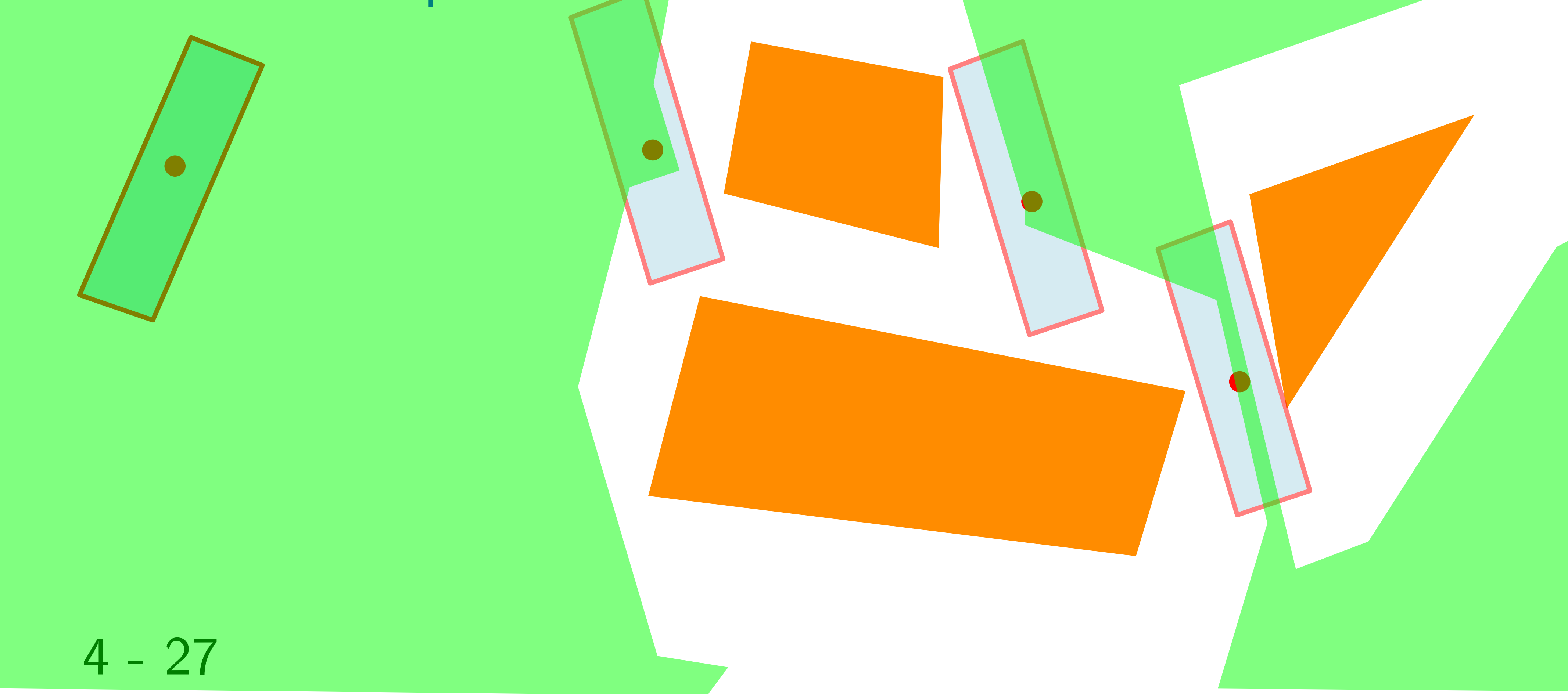


Reconstruction

Context

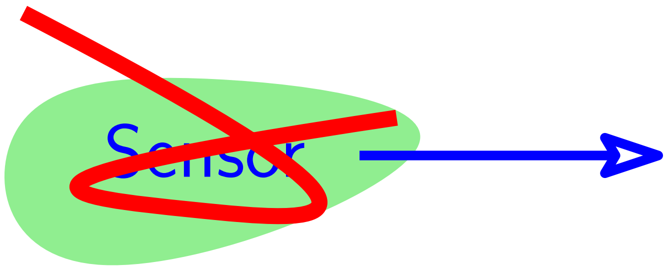
~~Sensor~~ → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section



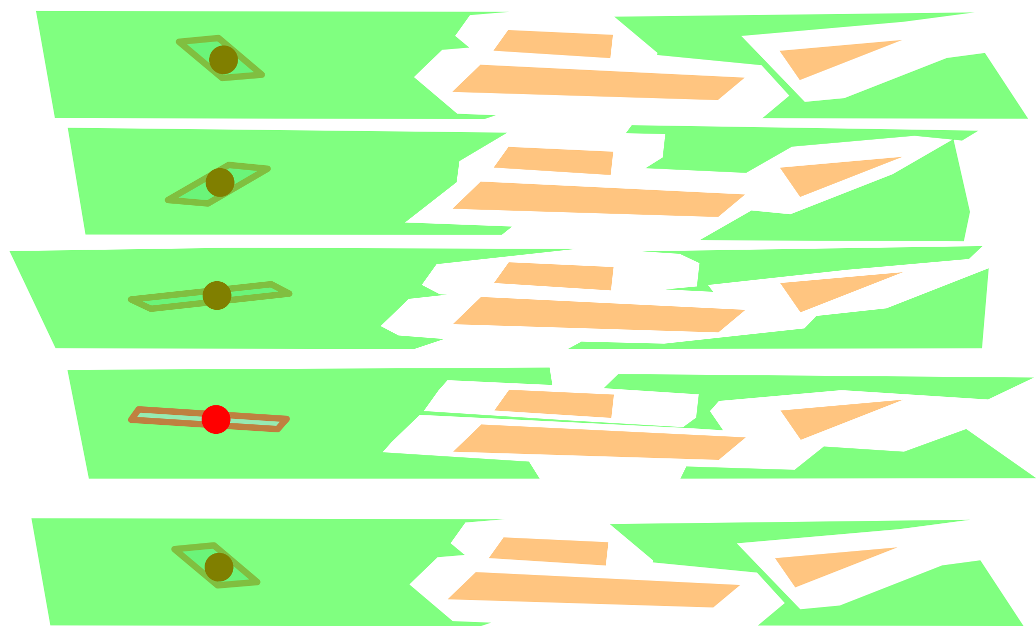
Reconstruction

Context



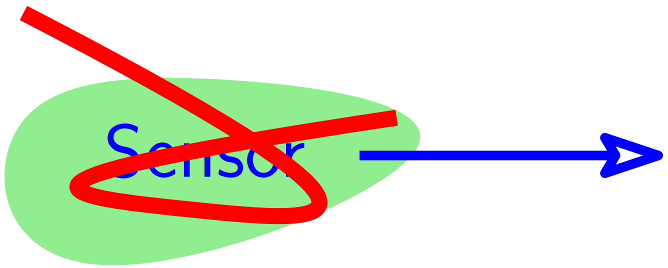
Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section



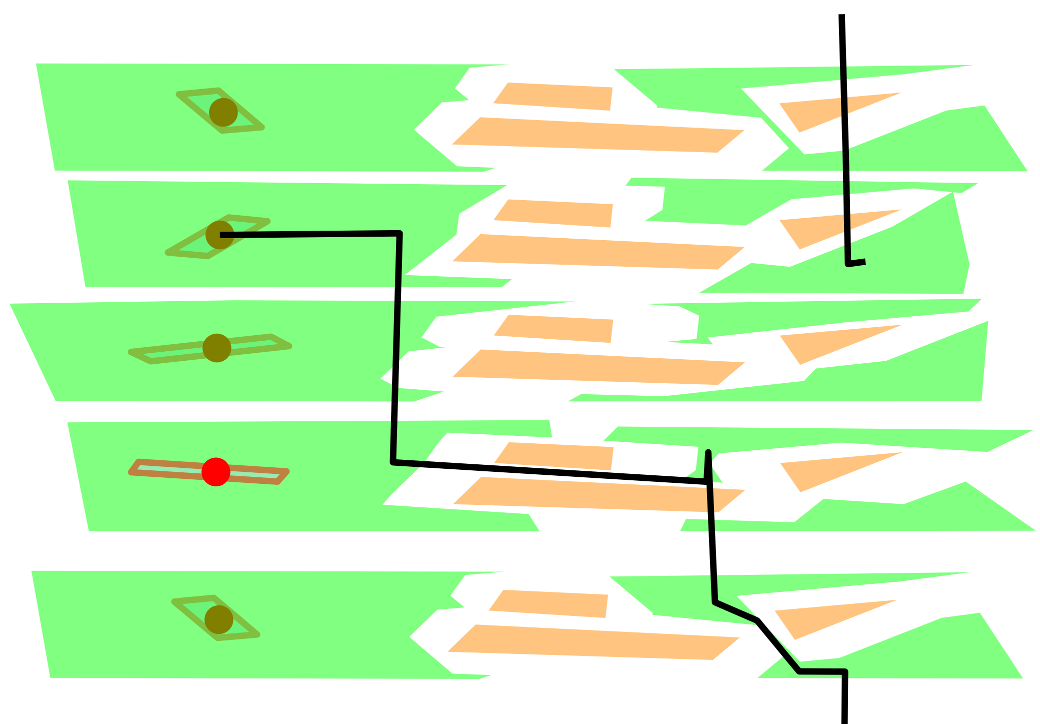
Reconstruction

Context



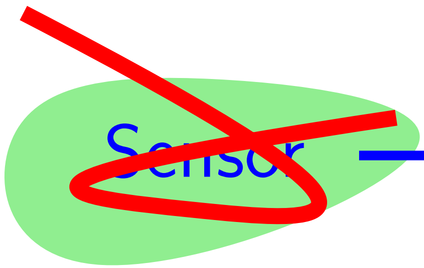
Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section



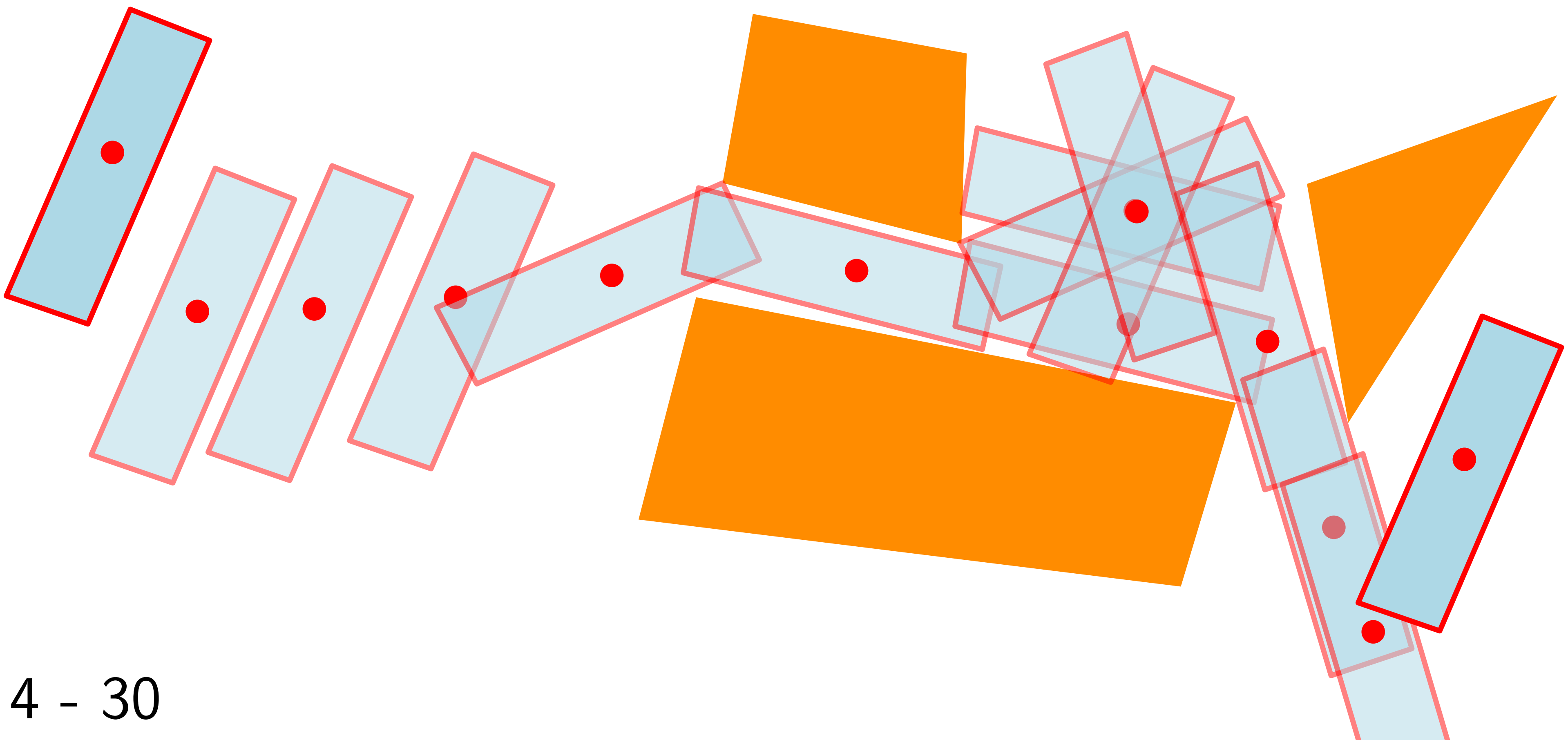
Reconstruction

Context



→ Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section



Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

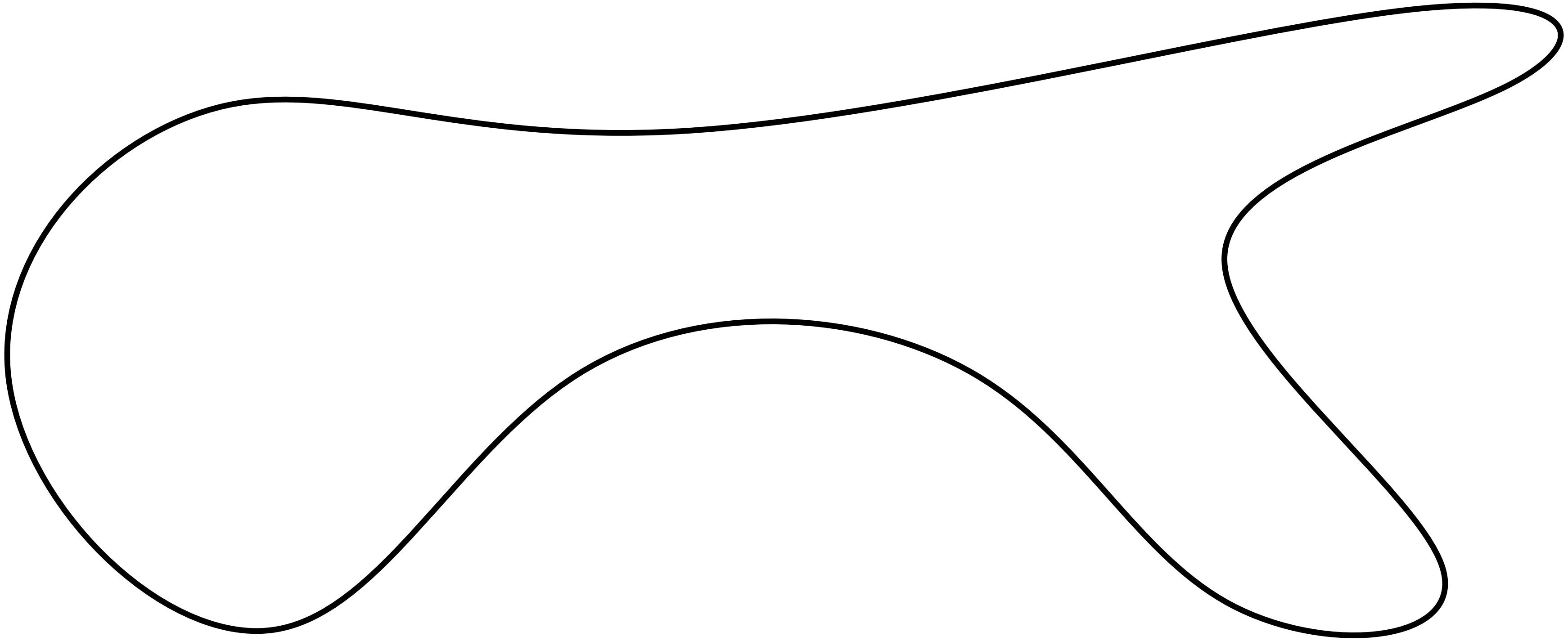
Locus of center of bitangent spheres

Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

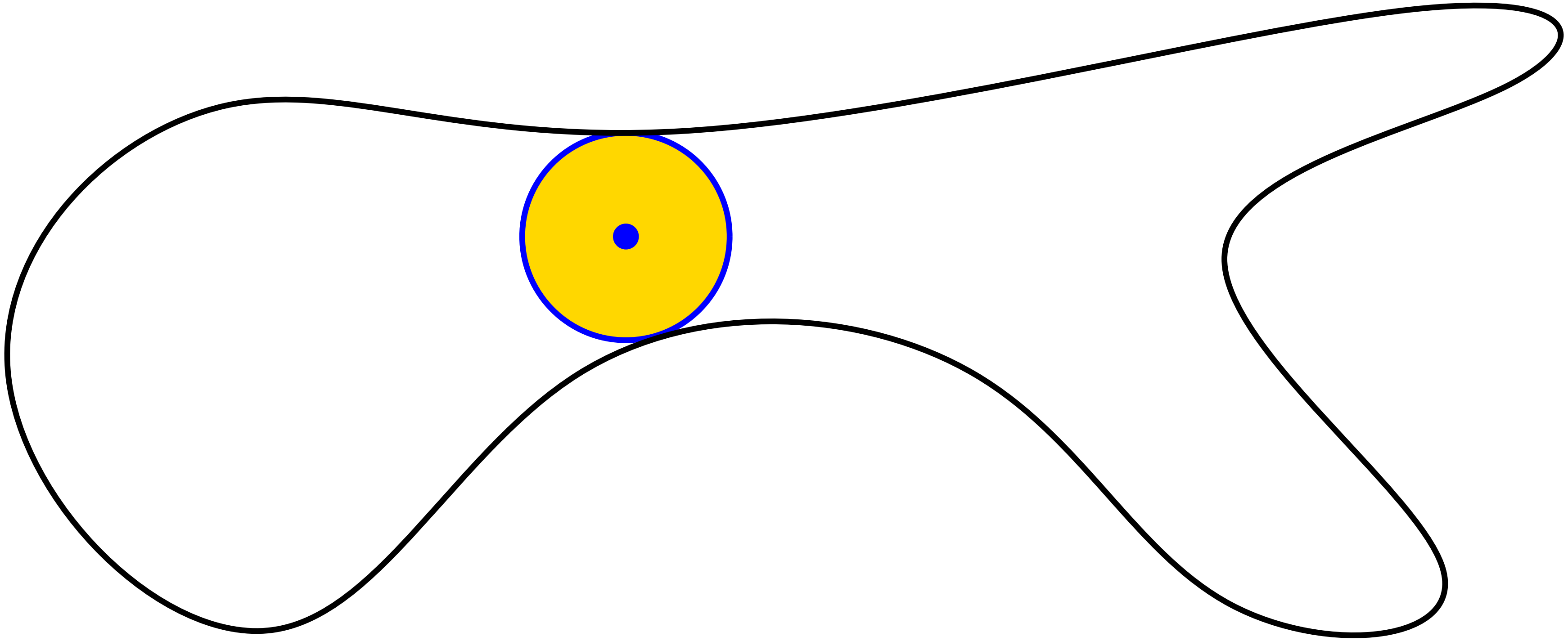


Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

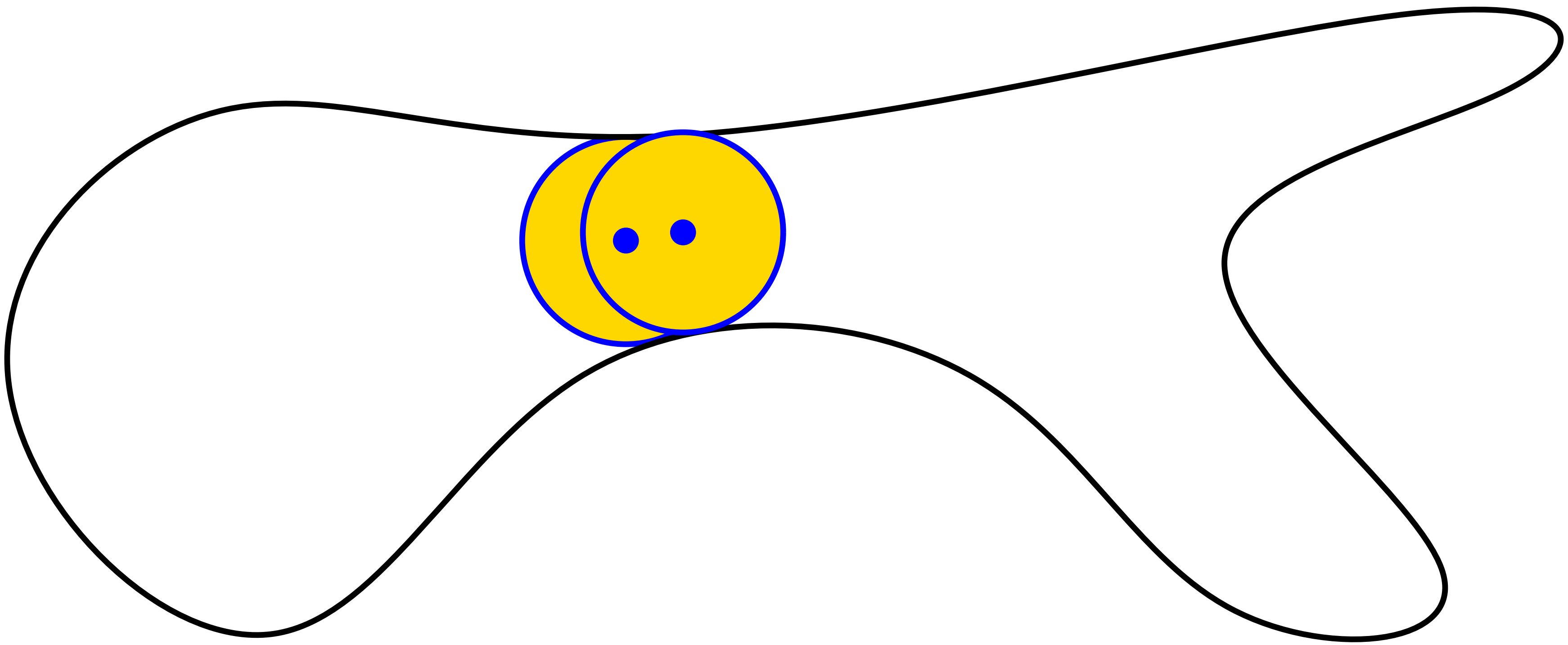


Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

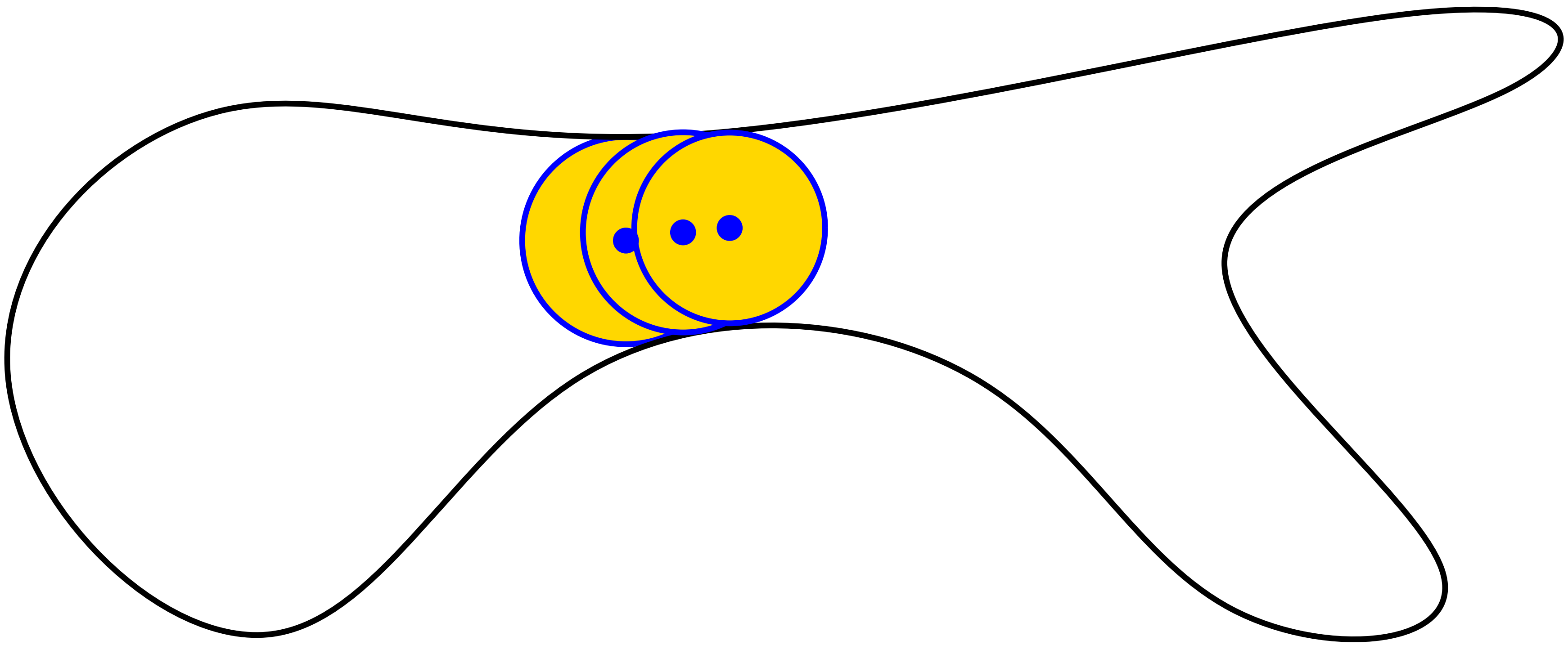


Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

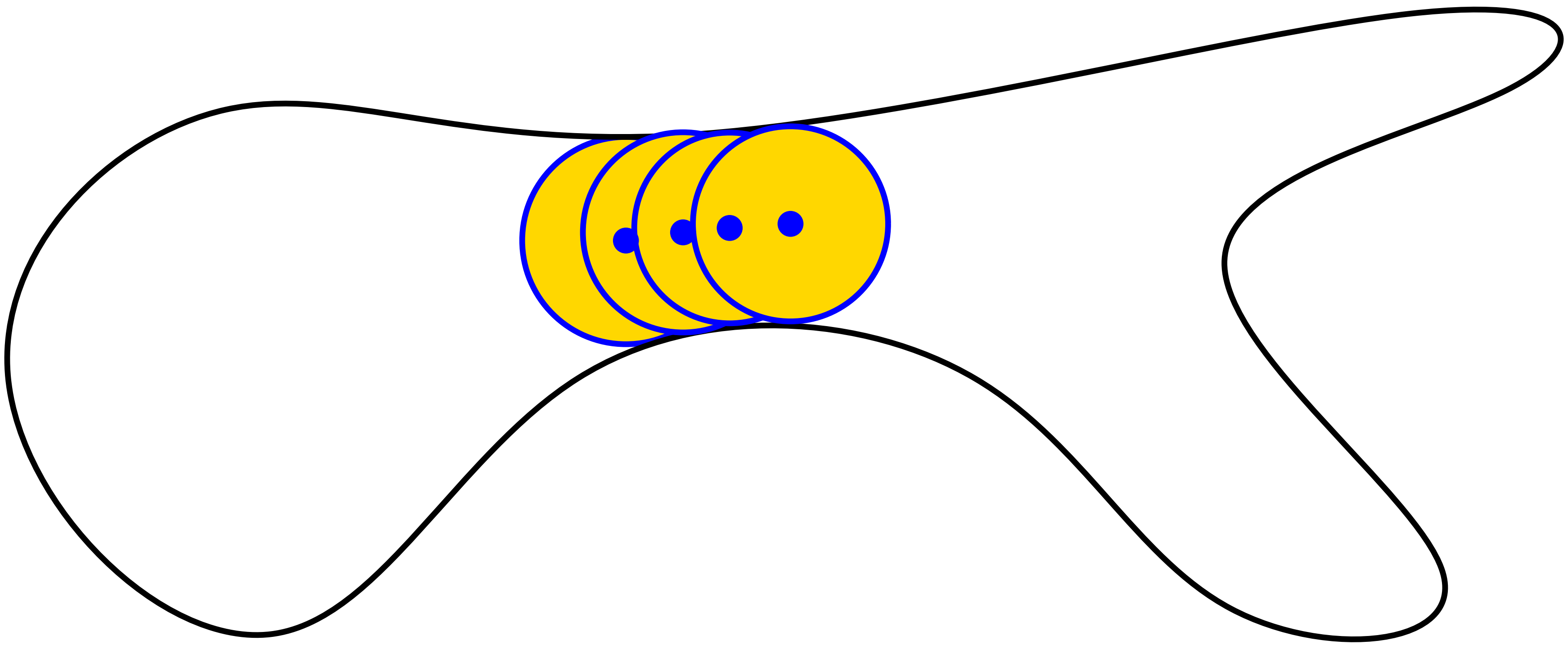


Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

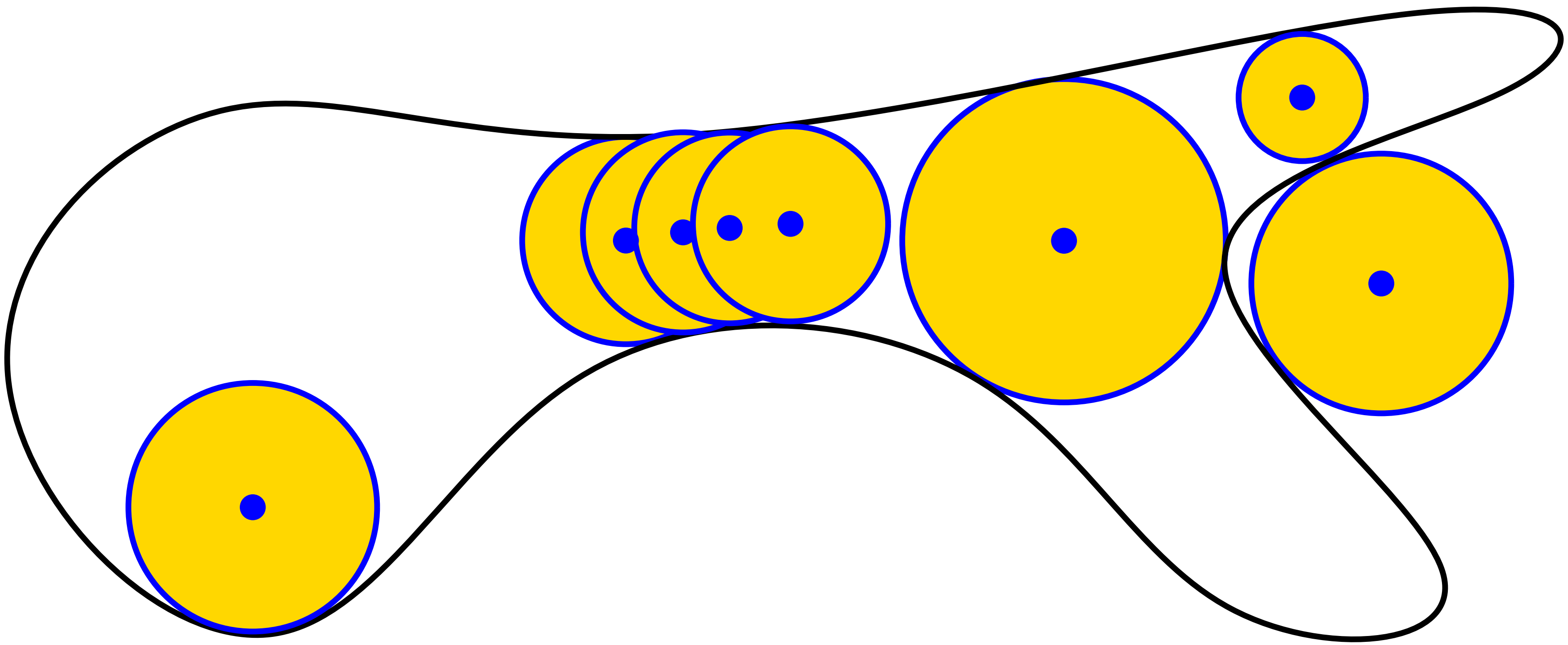


Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitantgent spheres

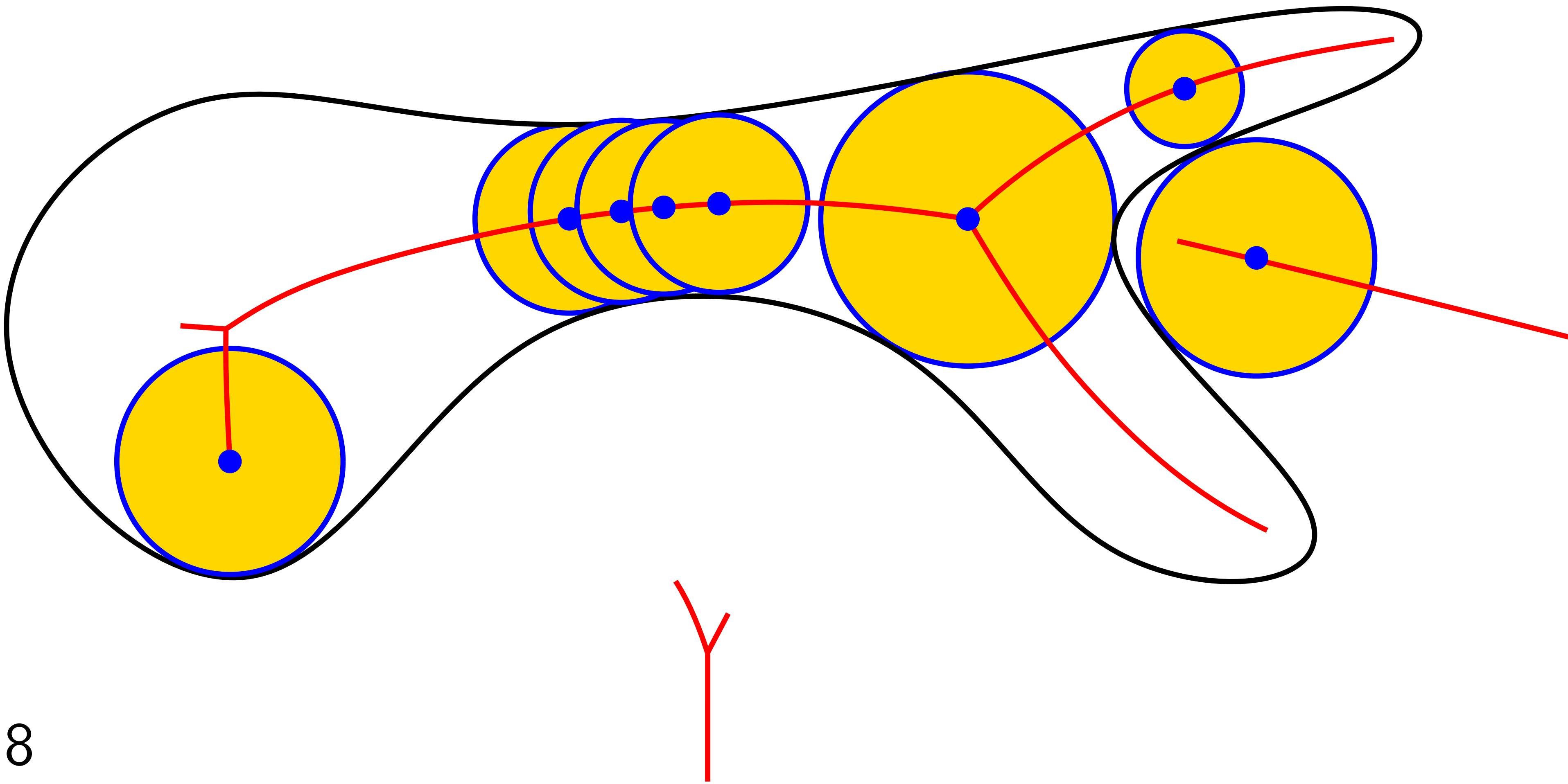


Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

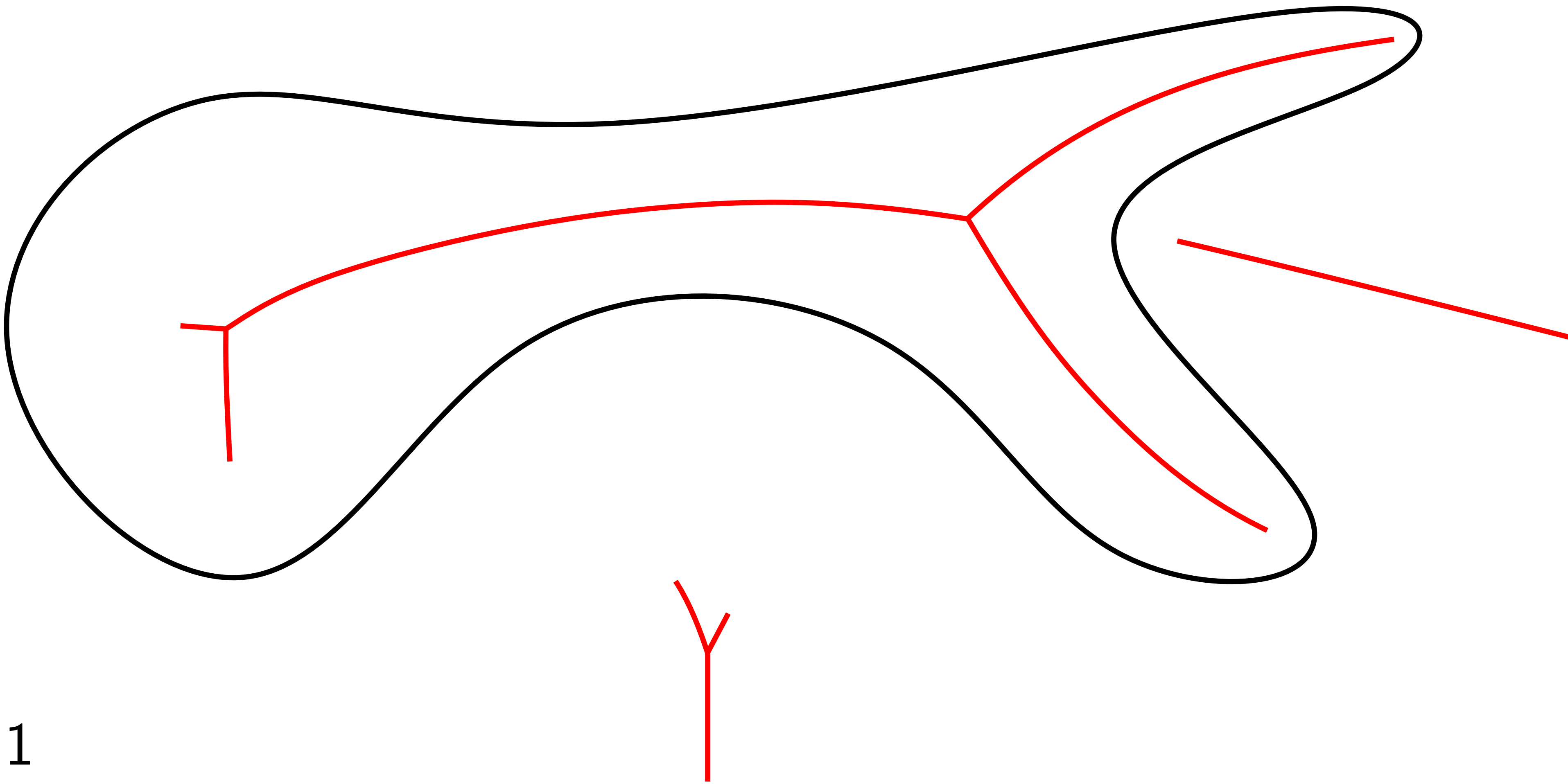


Reconstruction

Delaunay is a good start

ϵ-sample of a curve

Local feature size:

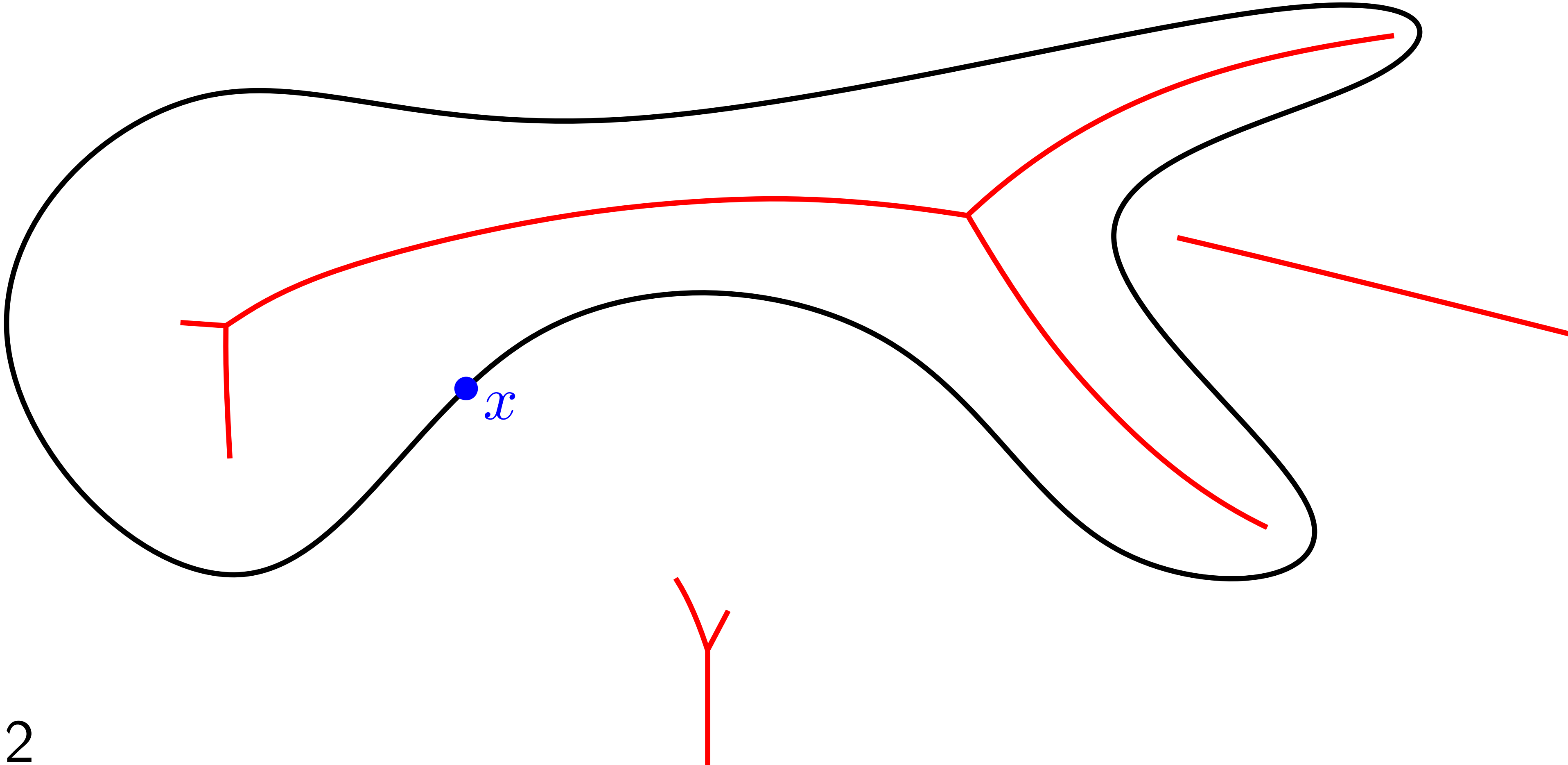


Reconstruction

Delaunay is a good start

ϵ-sample of a curve

Local feature size: $\text{lfs}(x) =$

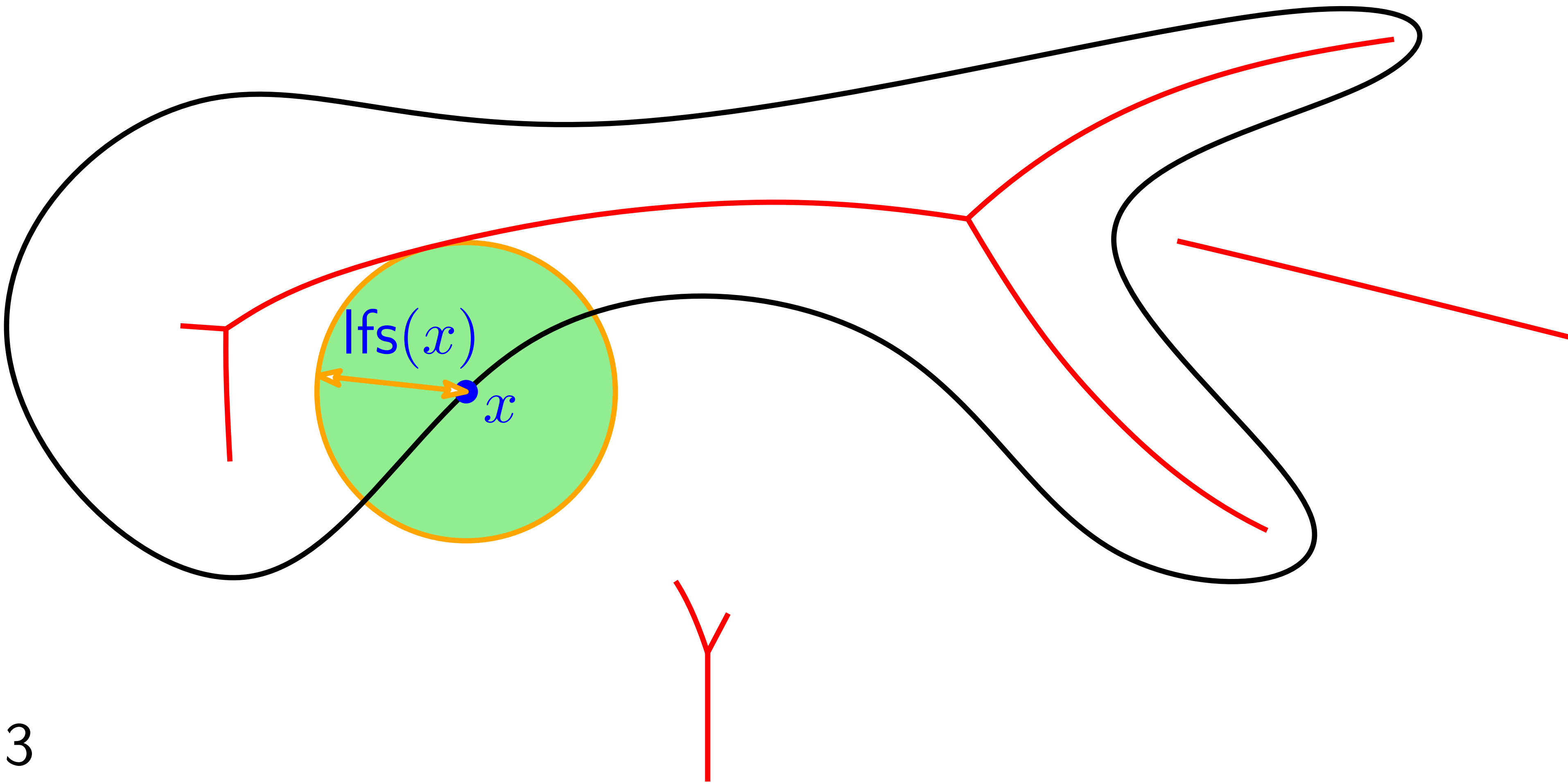


Reconstruction

Delaunay is a good start

ϵ-sample of a curve

Local feature size: lfs(x) = distance$(x$, medial axis$)$



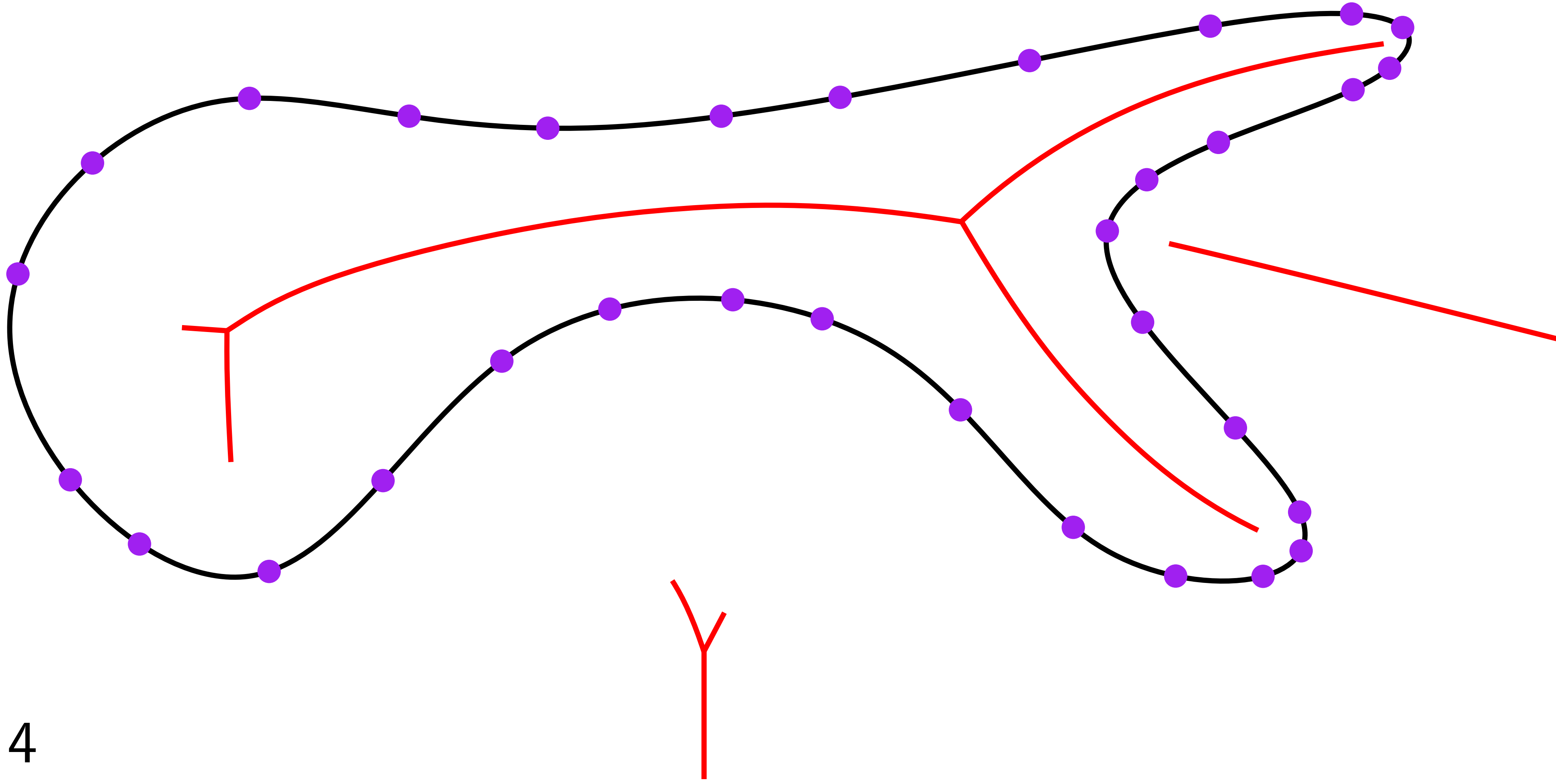
Reconstruction

Delaunay is a good start

Sample is an

ϵ-sample of a curve

Local feature size:



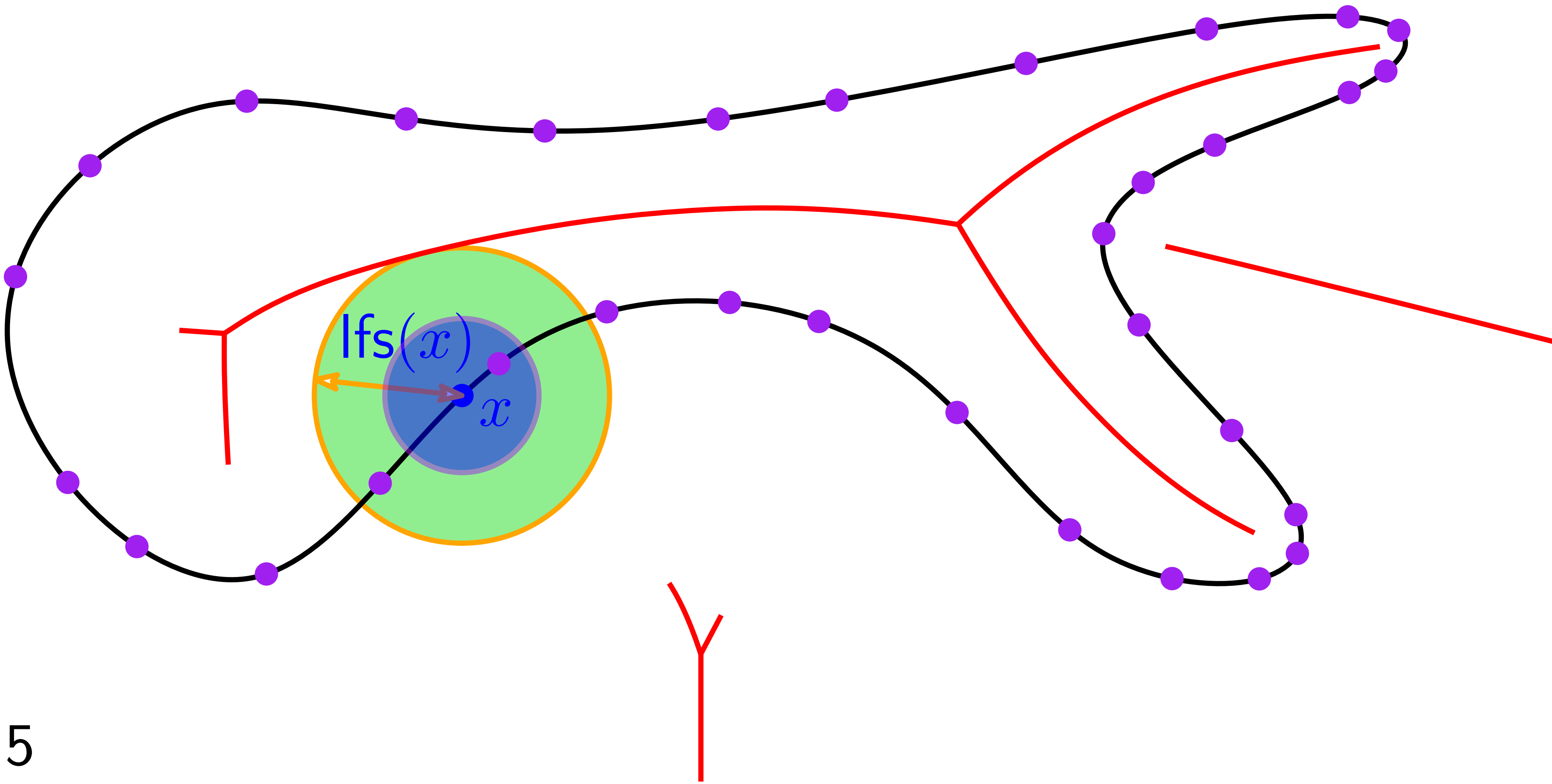
Reconstruction

Delaunay is a good start

Sample is an

ϵ-sample of a curve if $\forall x$, Disk$(x, \epsilon \cdot \text{lfs}(x)) \cap$ Sample $\neq \emptyset$

Local feature size: $\text{lfs}(x) =$ distance$(x,$ medial axis$)$



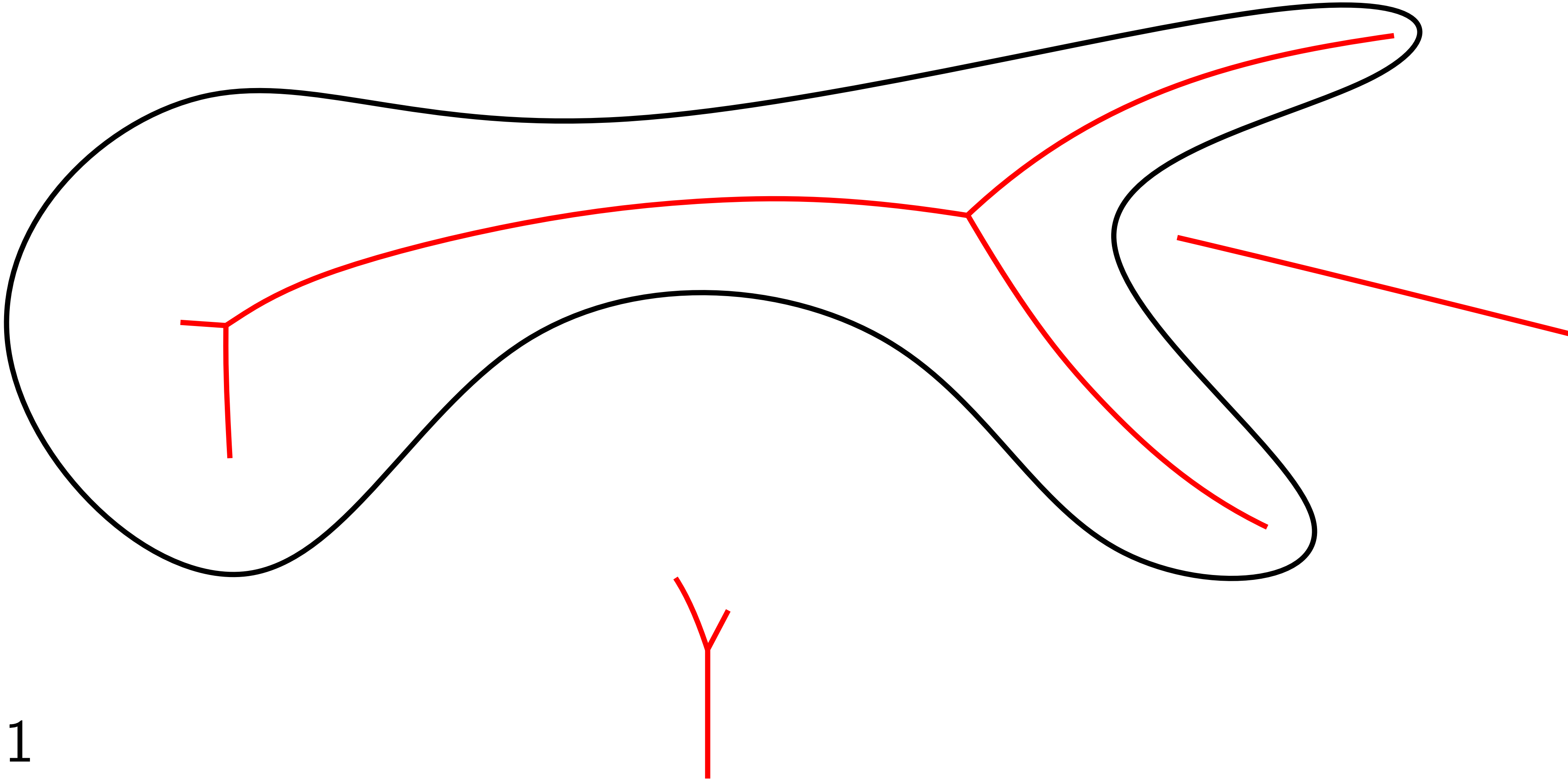
Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$



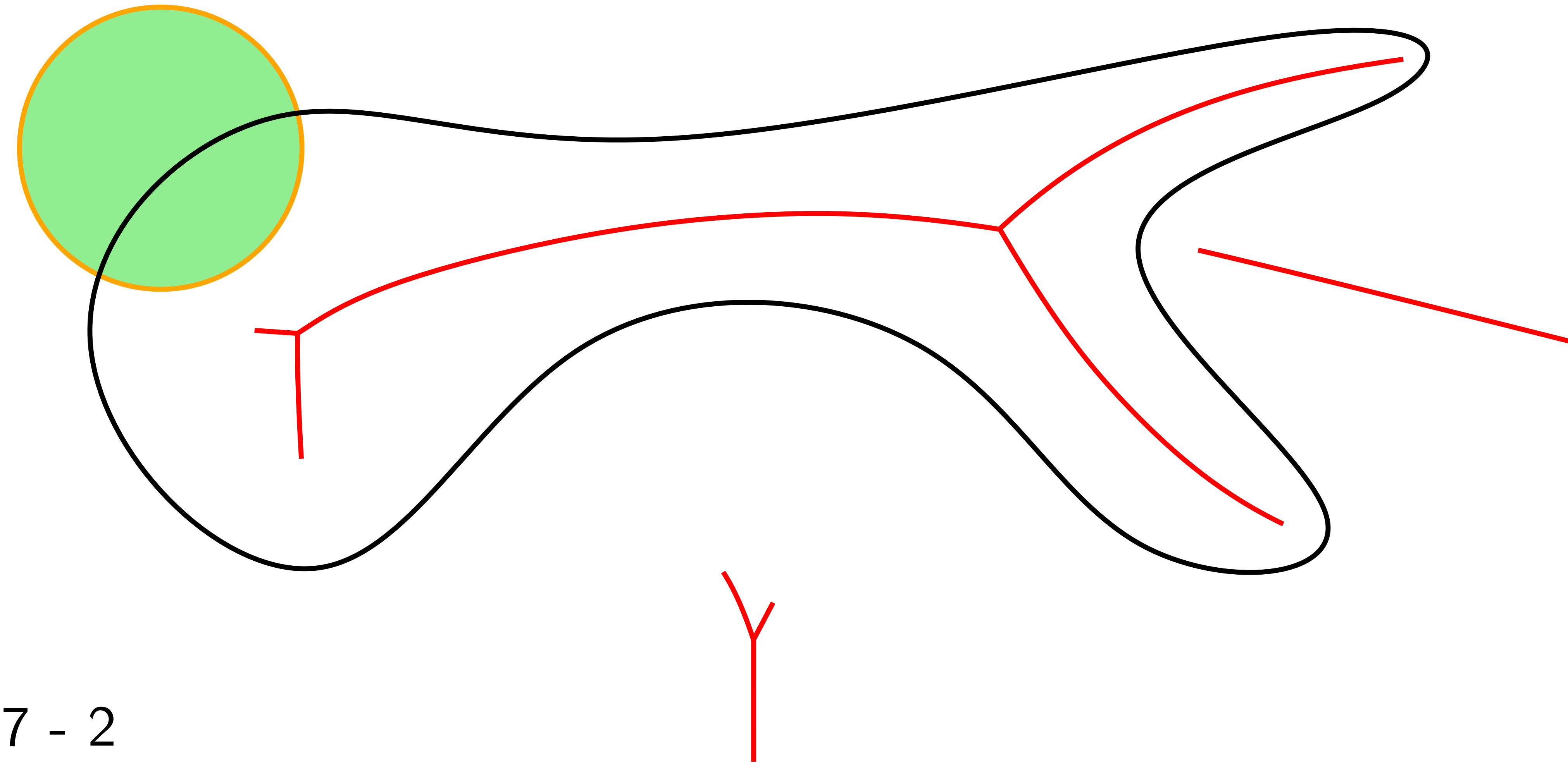
Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$



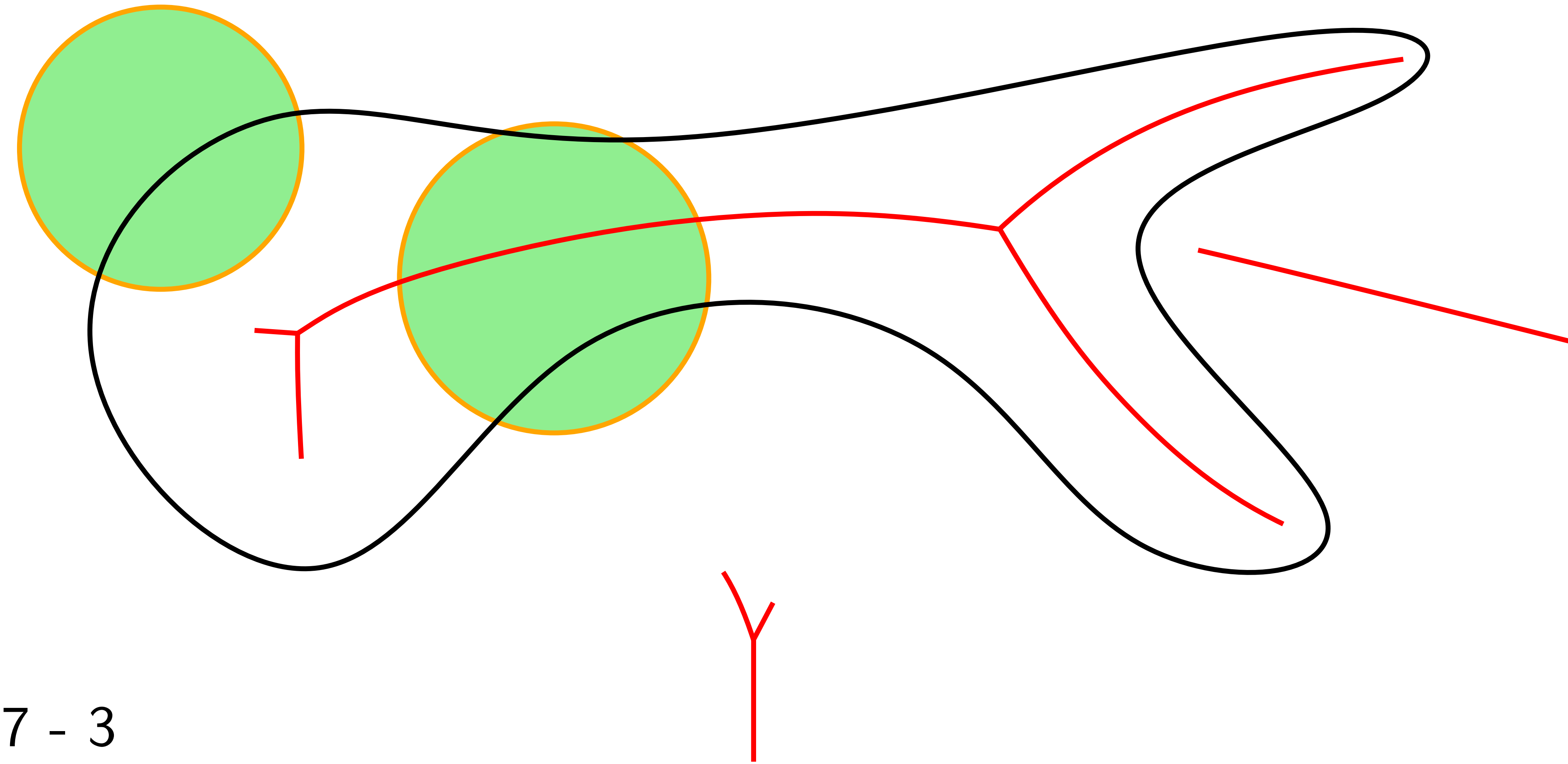
Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$



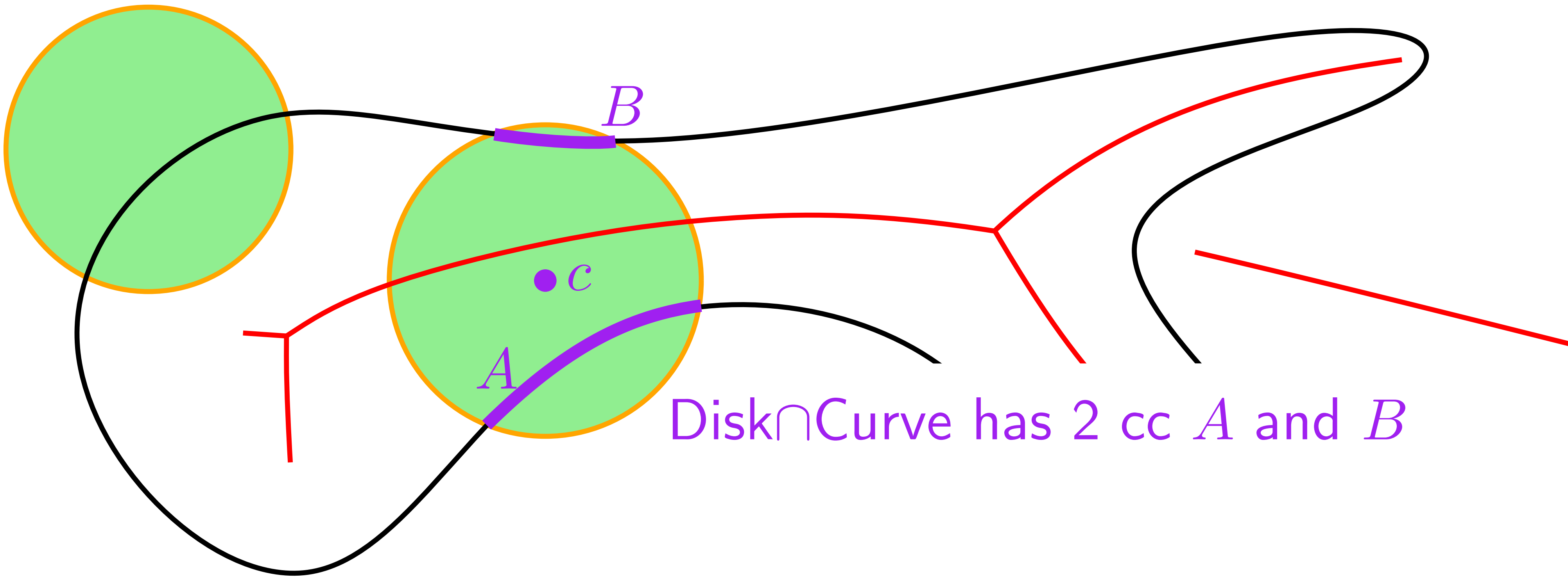
Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$

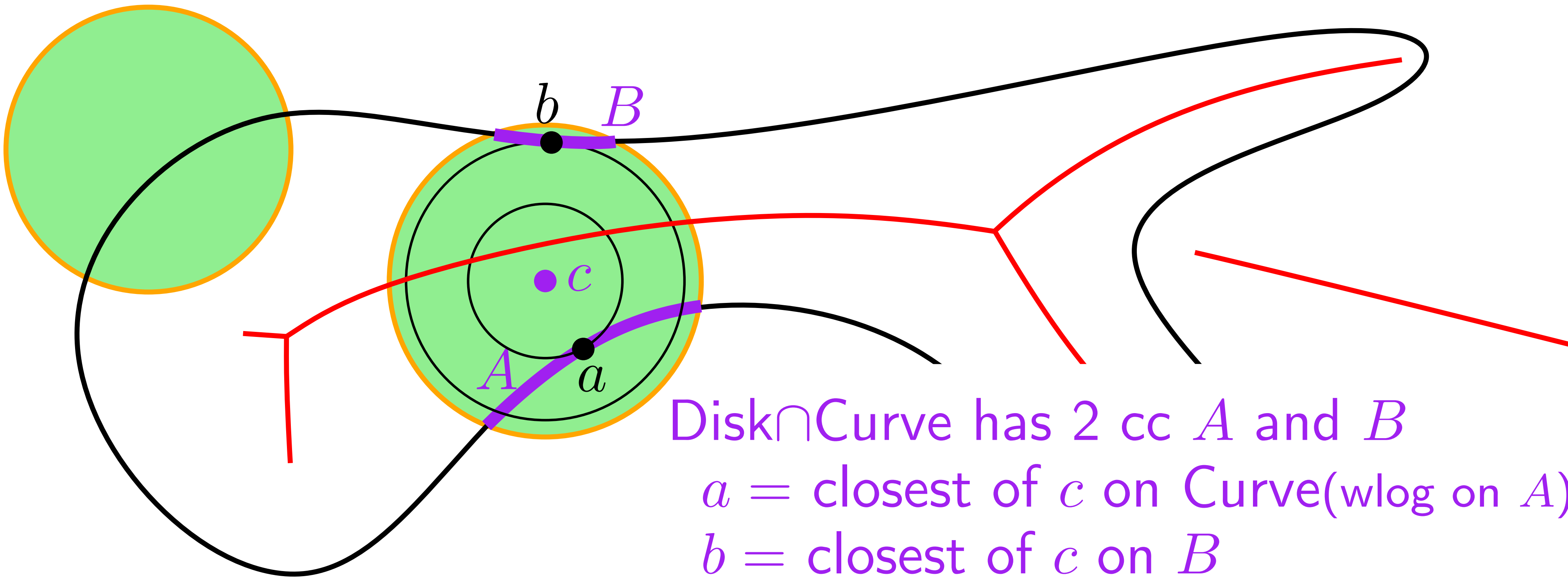


Reconstruction

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$



Disk$\cap$Curve has 2 cc A and B

$a =$ closest of c on Curve(wlog on A)

$b =$ closest of c on B

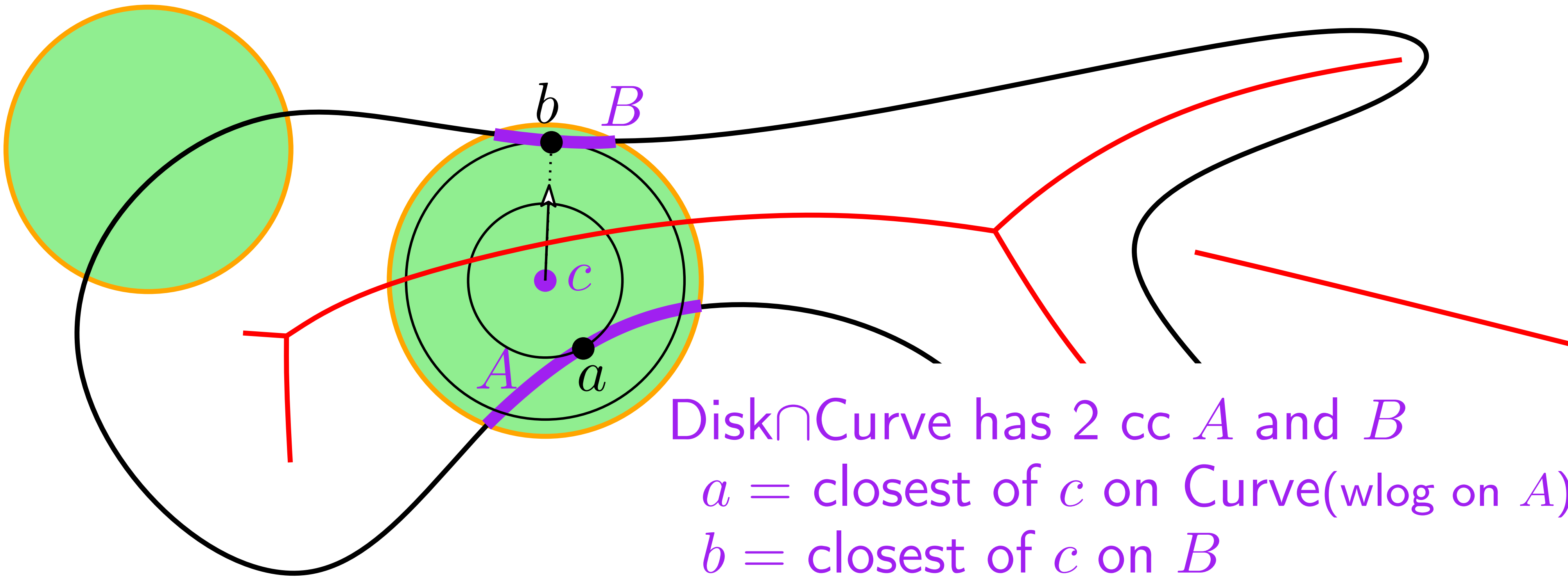
Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$



Disk$\cap$Curve has 2 cc A and B

$a =$ closest of c on Curve(wlog on A)

$b =$ closest of c on B

Moving from c to a dist to B $\nearrow$

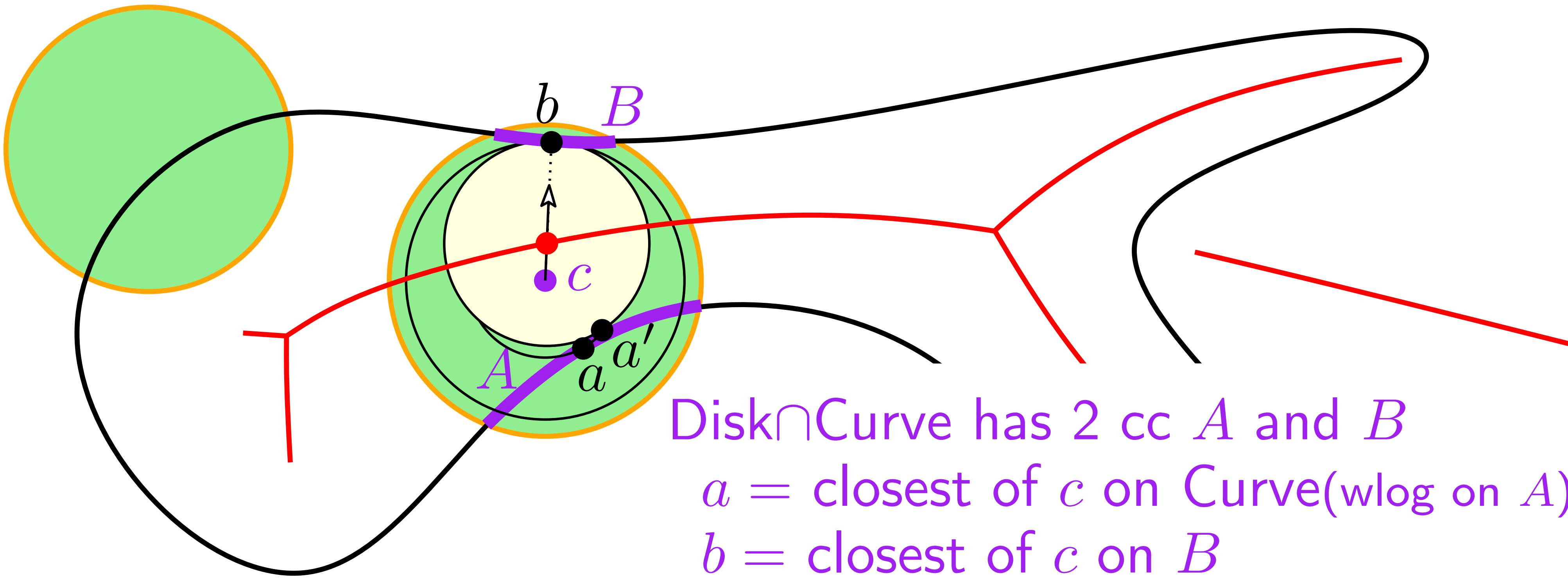
Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$



Disk$\cap$Curve has 2 cc A and B

$a =$ closest of c on Curve(wlog on A)

$b =$ closest of c on B

Moving from c to a dist to $B \nearrow$

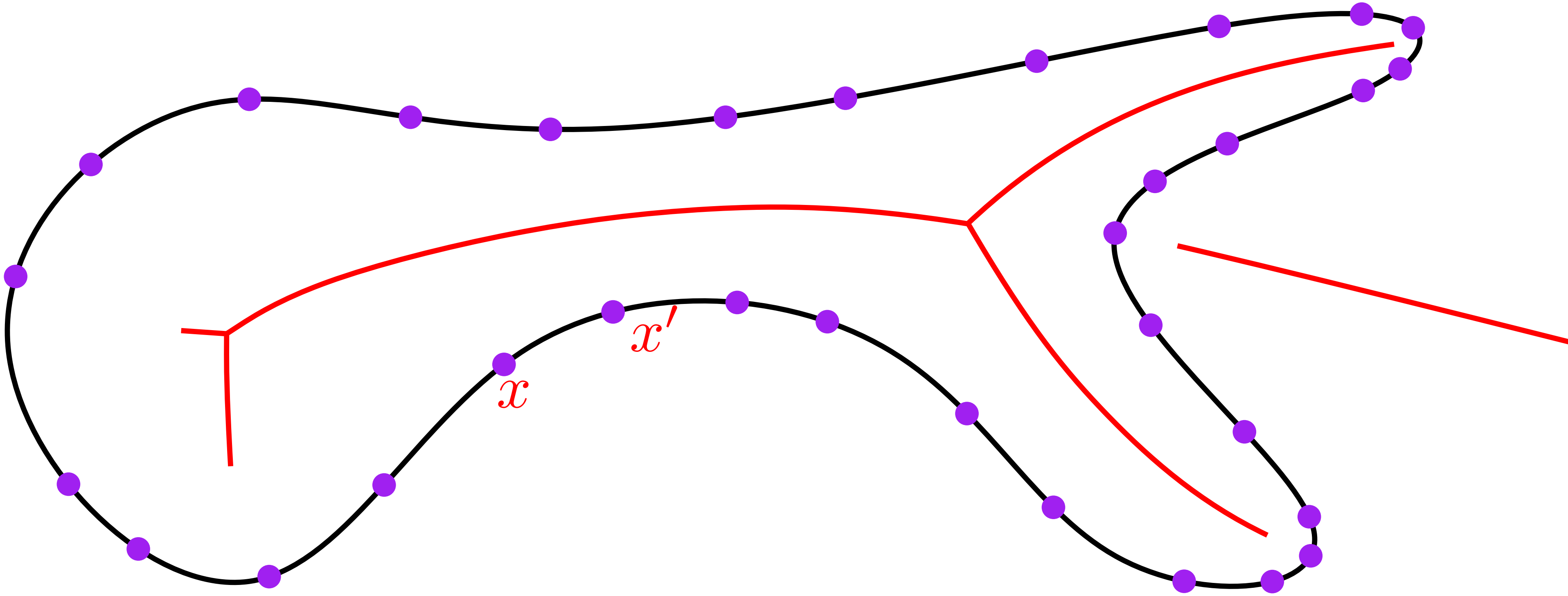
reach center of bitangent disk

Reconstruction

Delaunay is a good start

Theorem
If Sample is a ϵ-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors

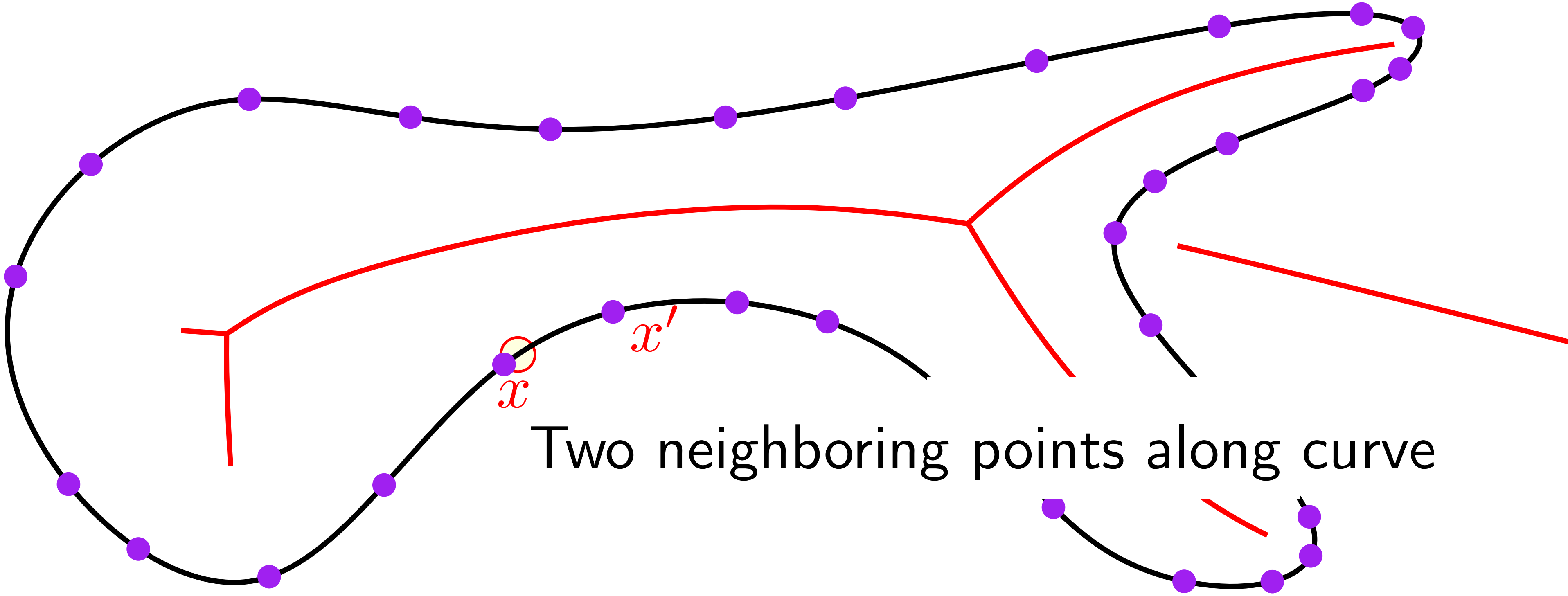


Reconstruction

Delaunay is a good start

Theorem
If Sample is a ϵ-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors



disks centered on Curve, through x

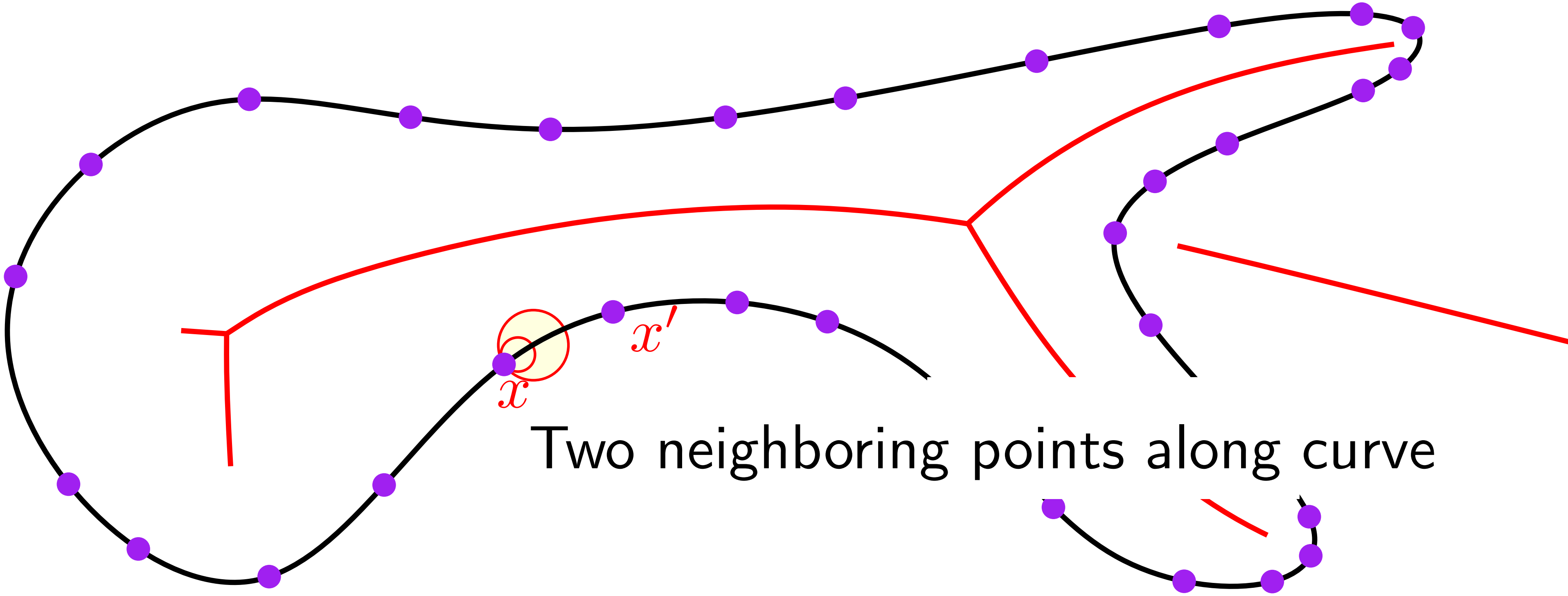
Reconstruction

Delaunay is a good start

Theorem

If Sample is a ϵ-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors



disks centered on Curve, through x

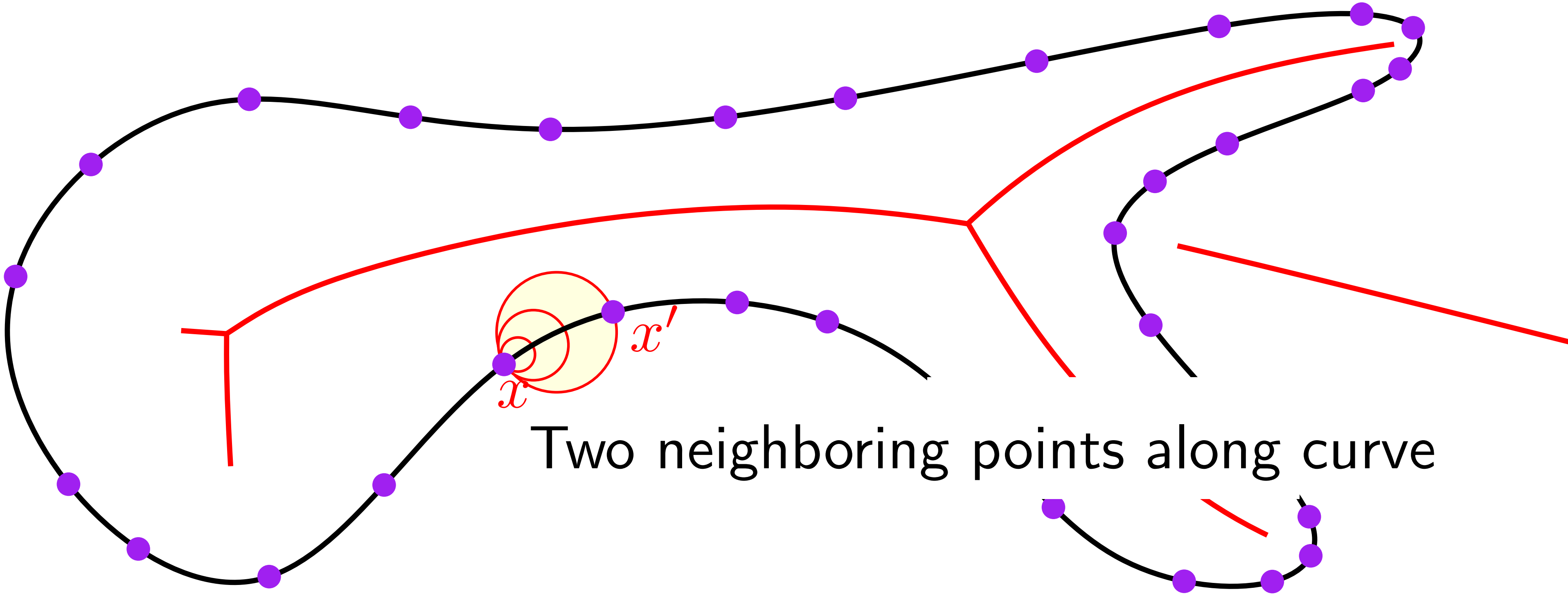
Reconstruction

Delaunay is a good start

Theorem

If Sample is a ϵ-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors



disks centered on Curve, through x

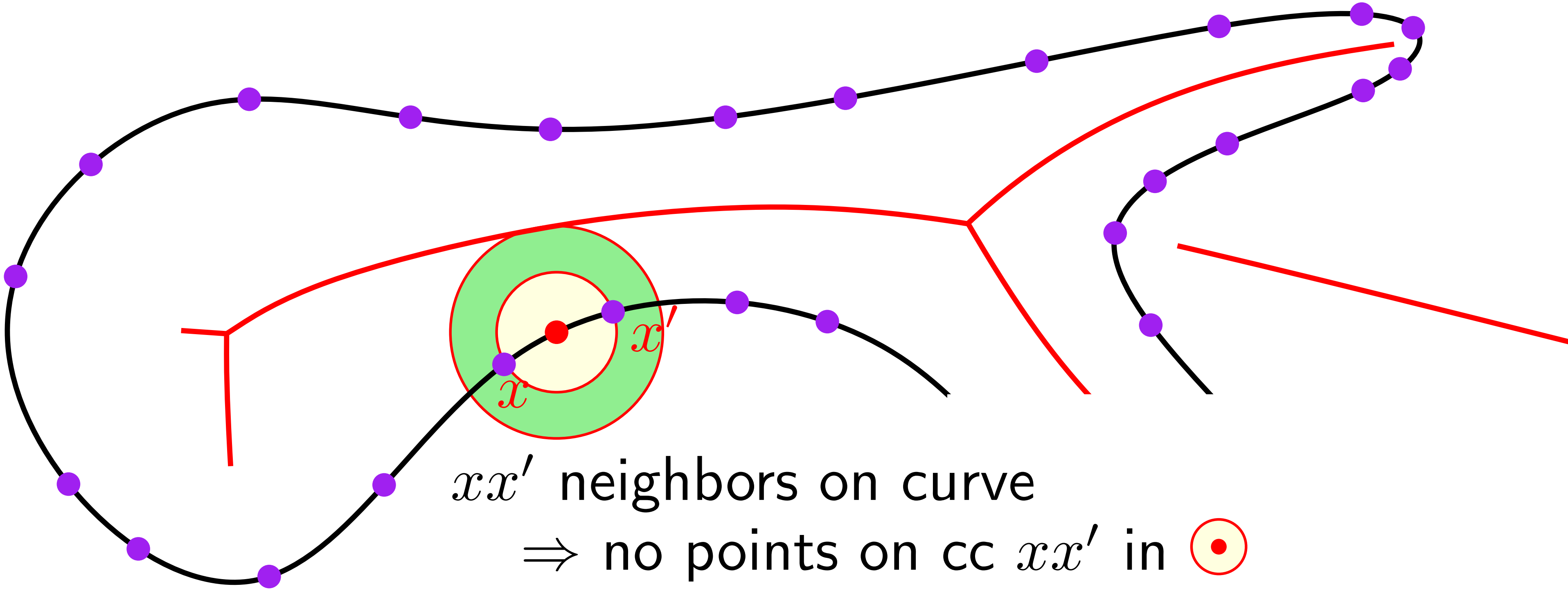
Reconstruction

Delaunay is a good start

Theorem

If Sample is a ϵ-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors



xx' neighbors on curve

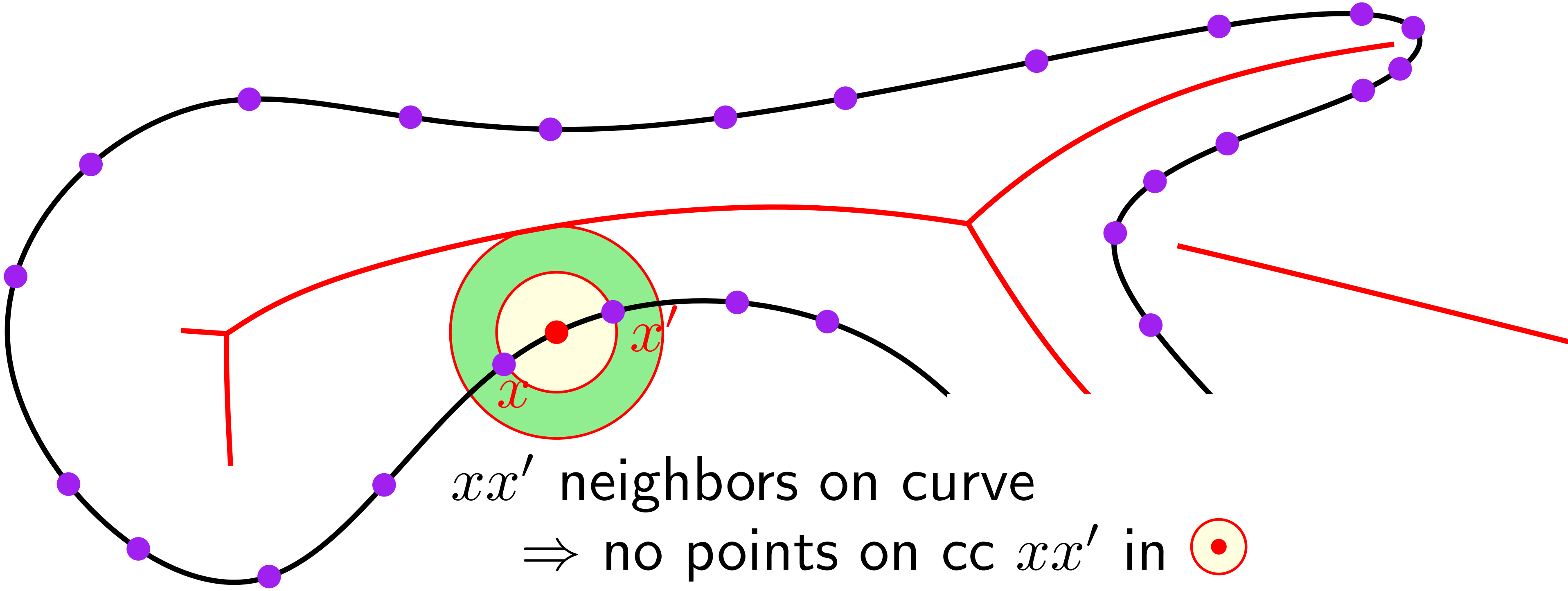
$\Rightarrow$ no points on cc xx' in ⊙

Reconstruction

Delaunay is a good start

Theorem
If Sample is a ϵ-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors



xx' neighbors on curve

$\Rightarrow$ no points on cc xx' in

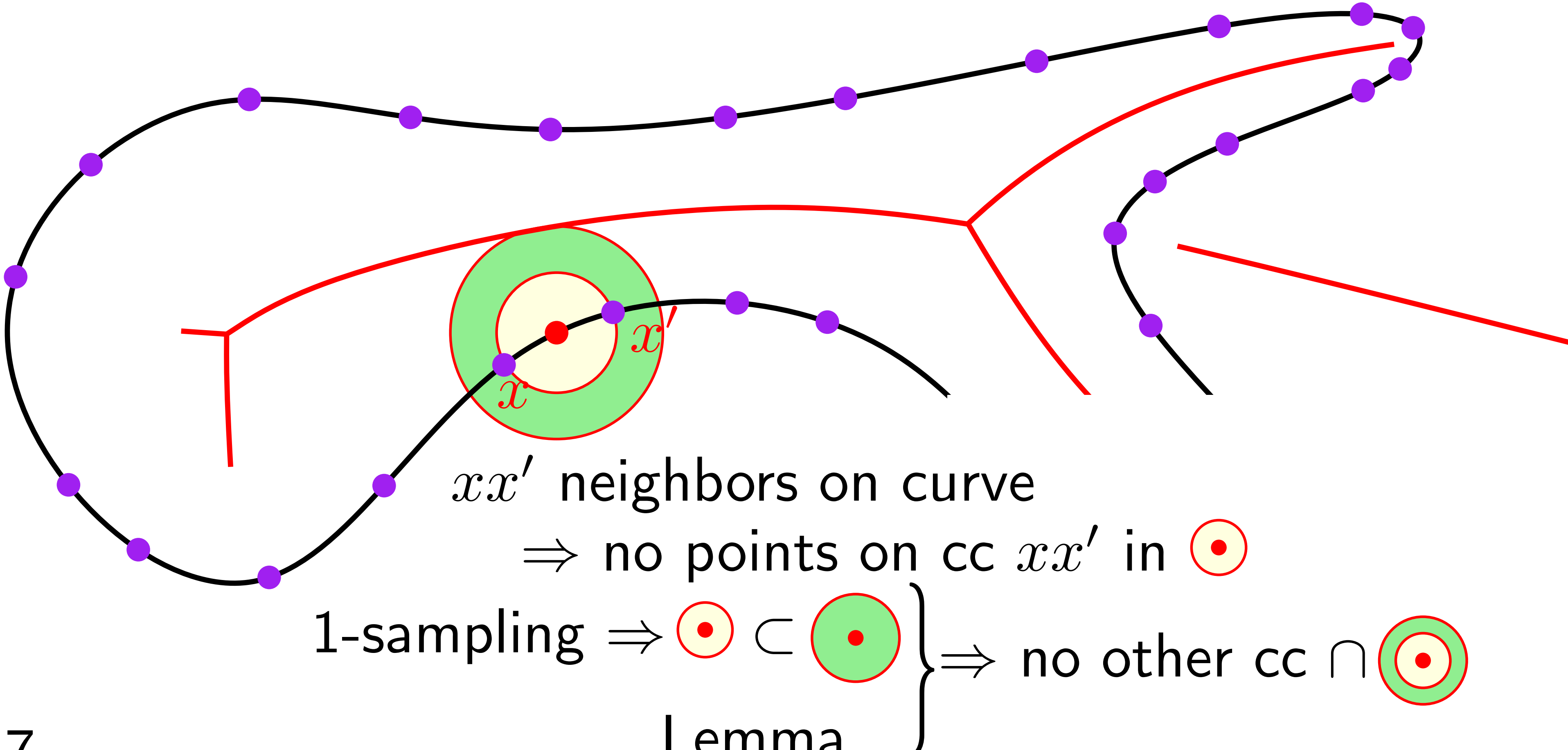
1-sampling $\Rightarrow$ $\subset$

Reconstruction

Delaunay is a good start

Theorem
If Sample is a ϵ-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors



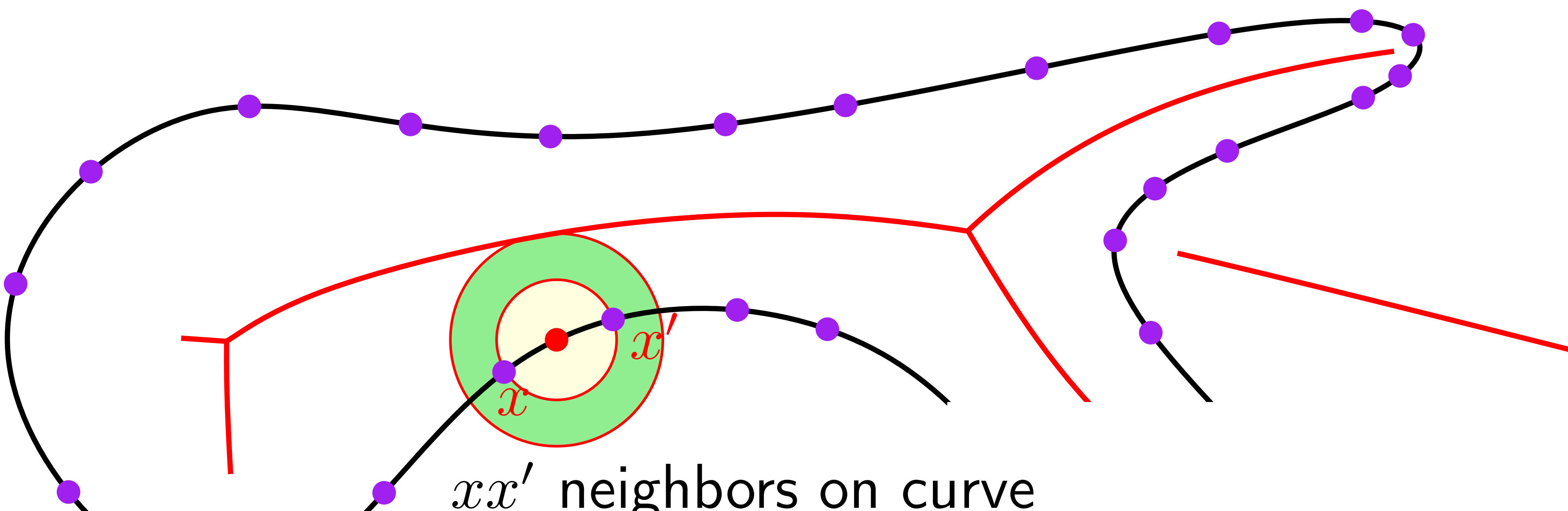
Reconstruction

Delaunay is a good start

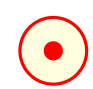
Theorem

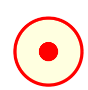
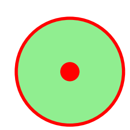
If Sample is a ϵ-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors

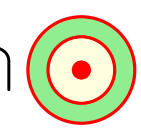


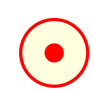
xx' neighbors on curve

$\Rightarrow$ no points on cc xx' in 

1-sampling $\Rightarrow$  $\subset$ 

Lemma

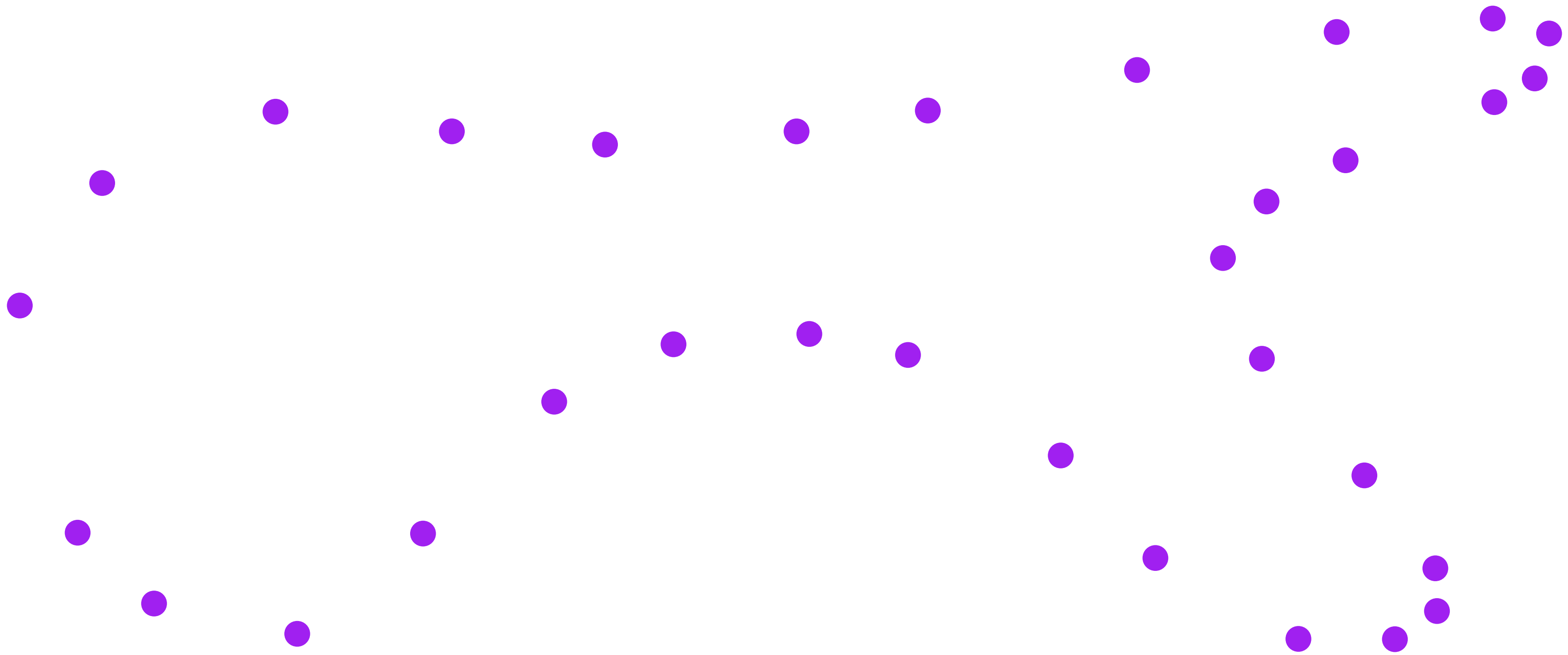
$\Rightarrow$ no other cc $\cap$ 

$\Rightarrow$  empty

Reconstruction

Delaunay is a good start

Given a sampling

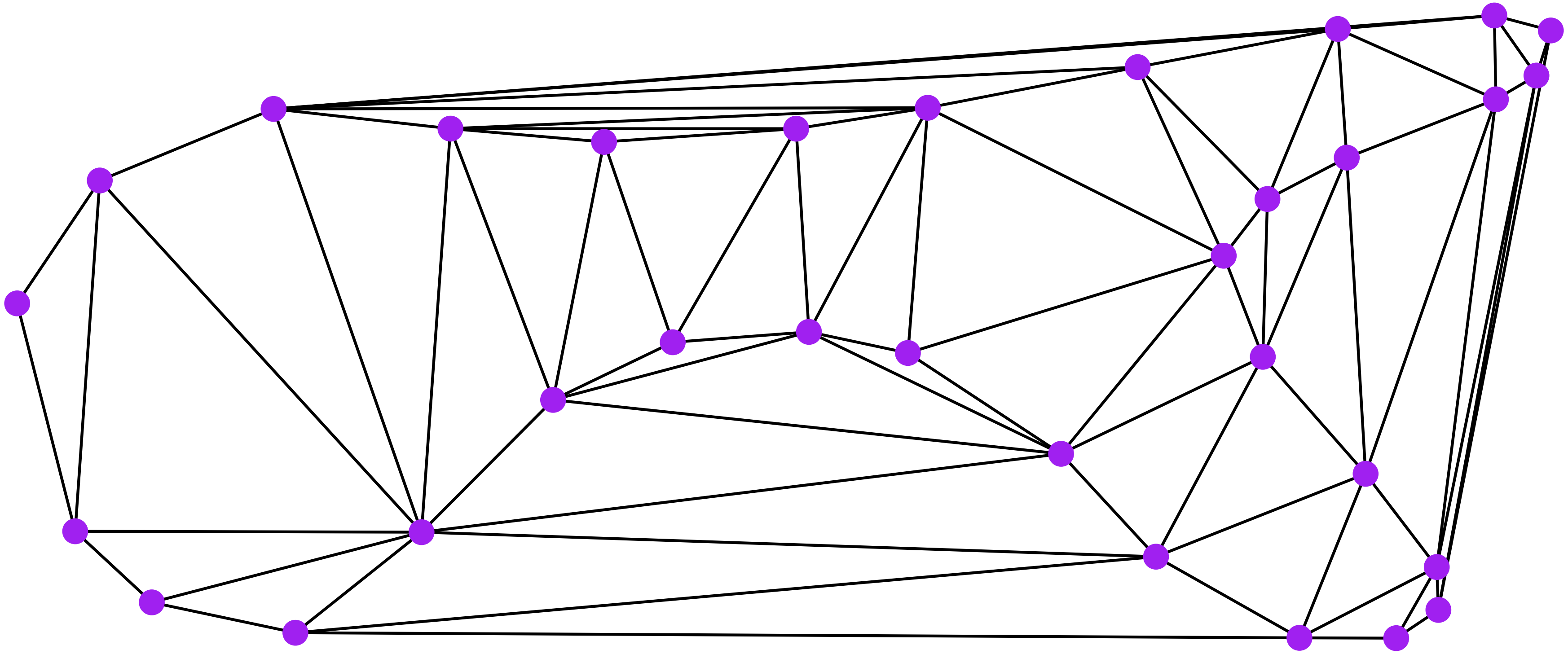


Reconstruction

Delaunay is a good start

Given a sampling

Compute Delaunay



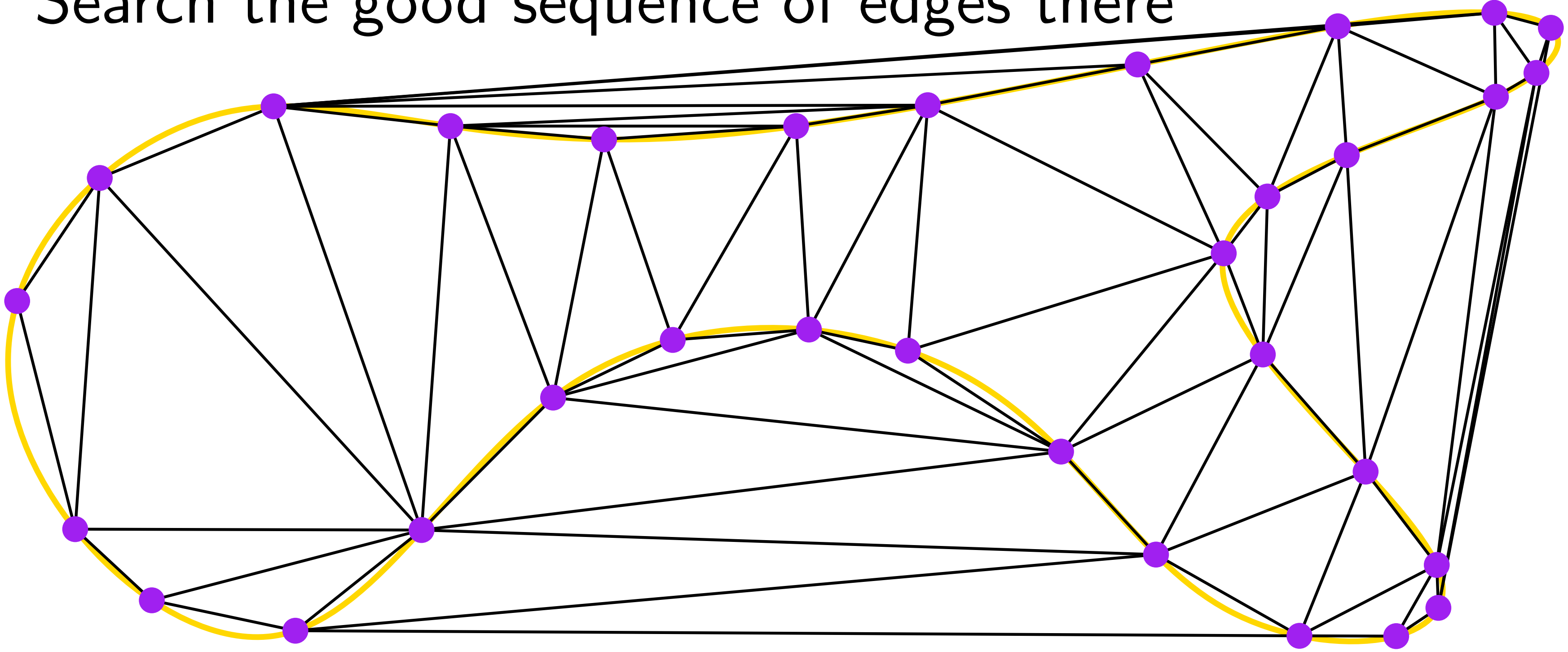
Reconstruction

Delaunay is a good start

Given a sampling

Compute Delaunay

Search the good sequence of edges there



Reconstruction

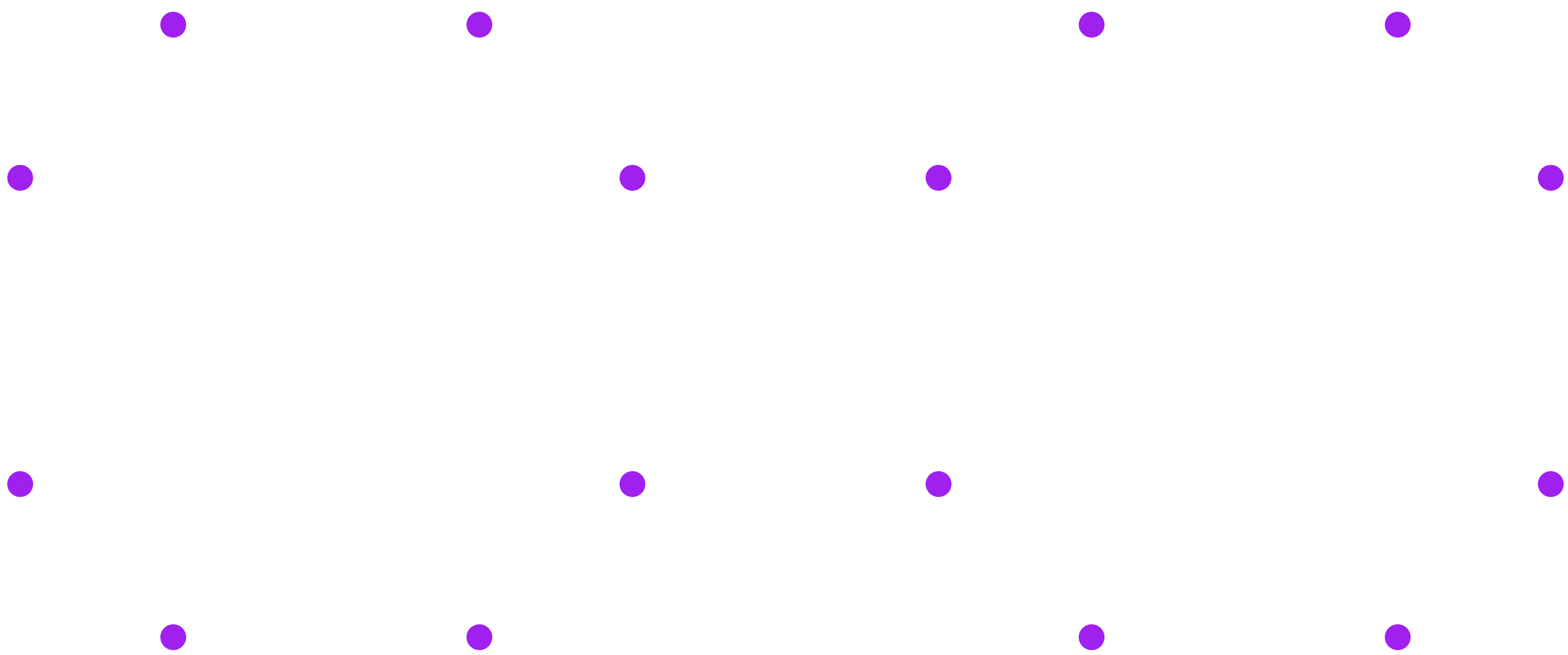
Delaunay is a good start

1-sample is not enough

Reconstruction

Delaunay is a good start

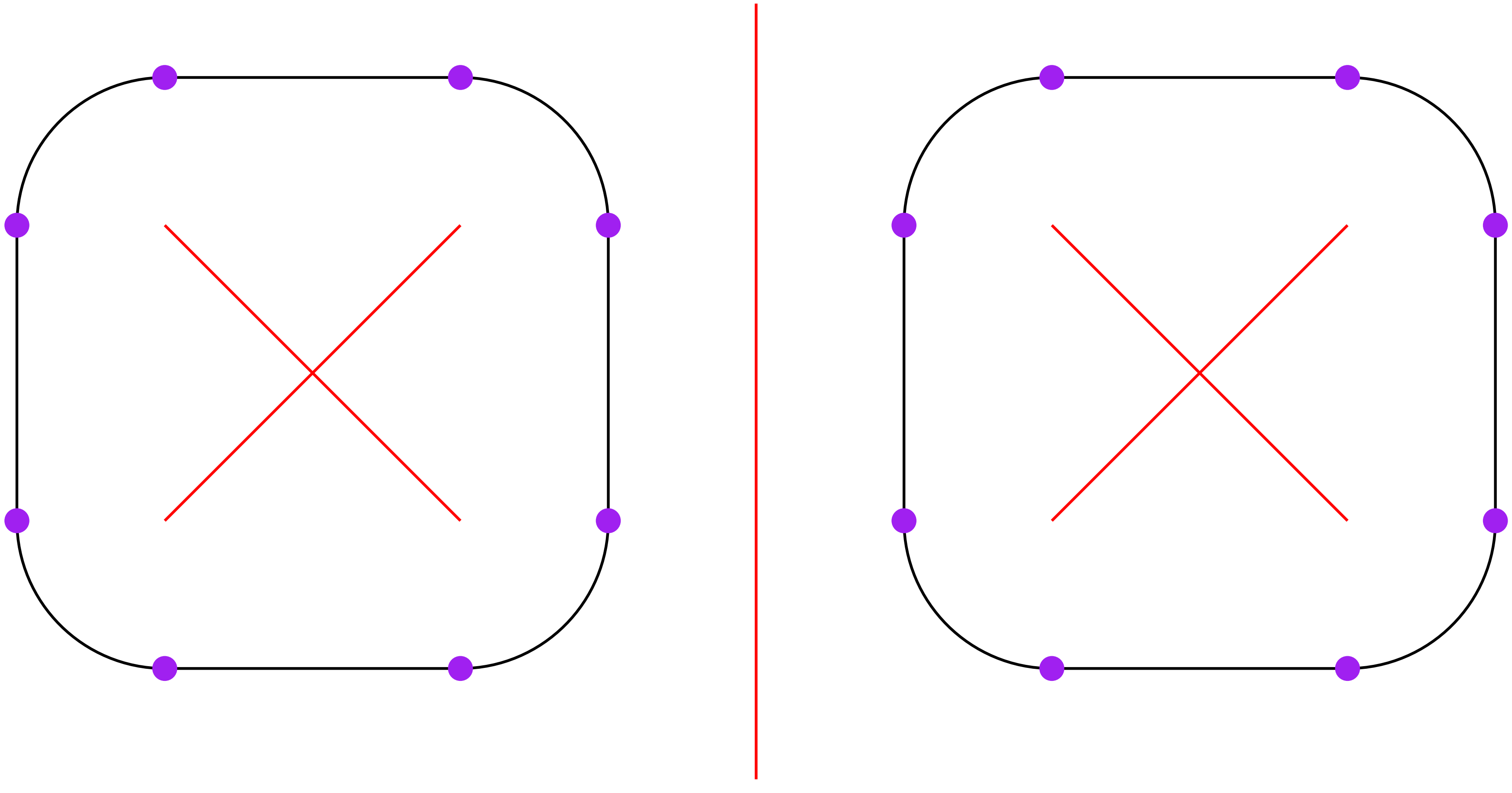
1-sample is not enough



Reconstruction

Delaunay is a good start

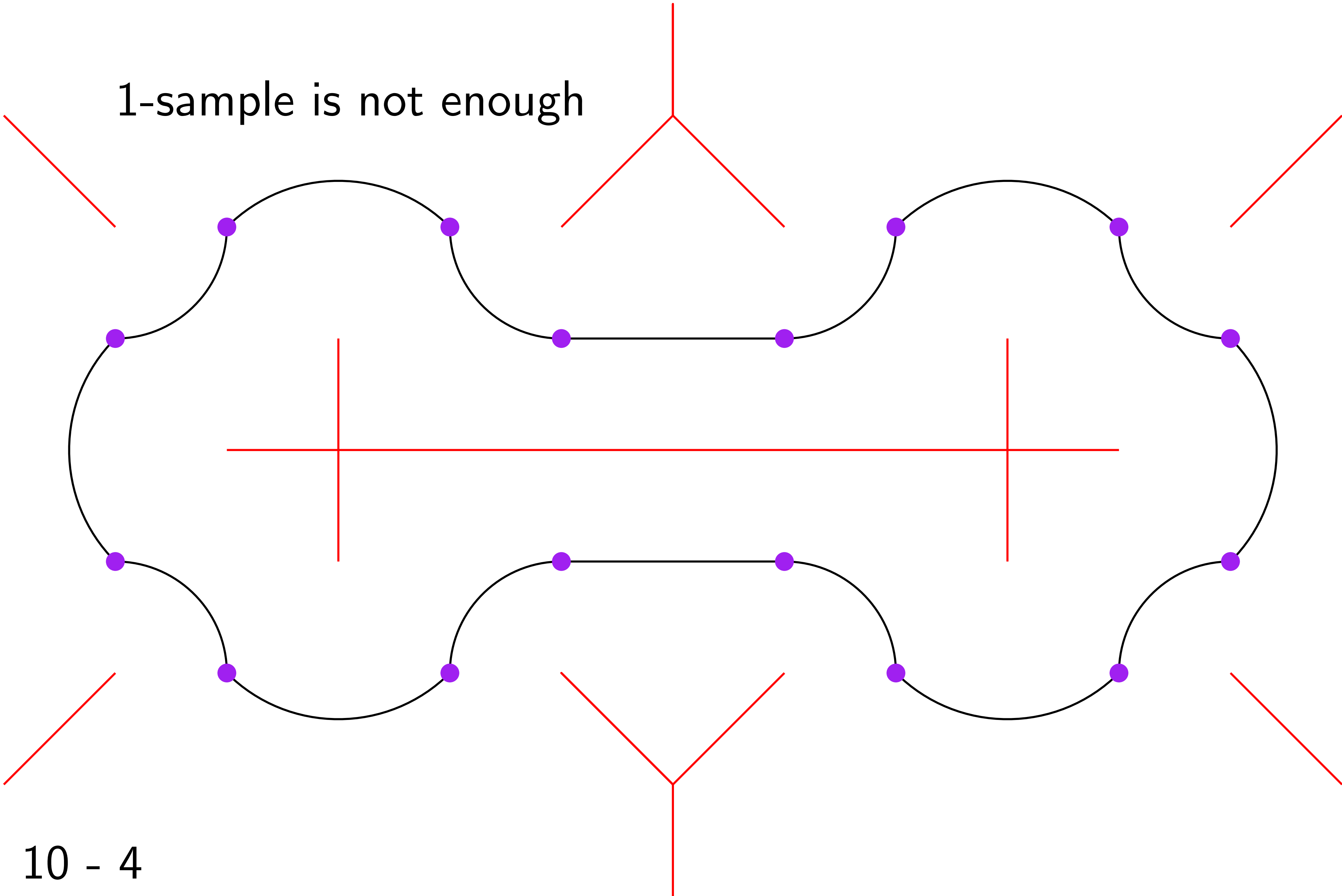
1-sample is not enough

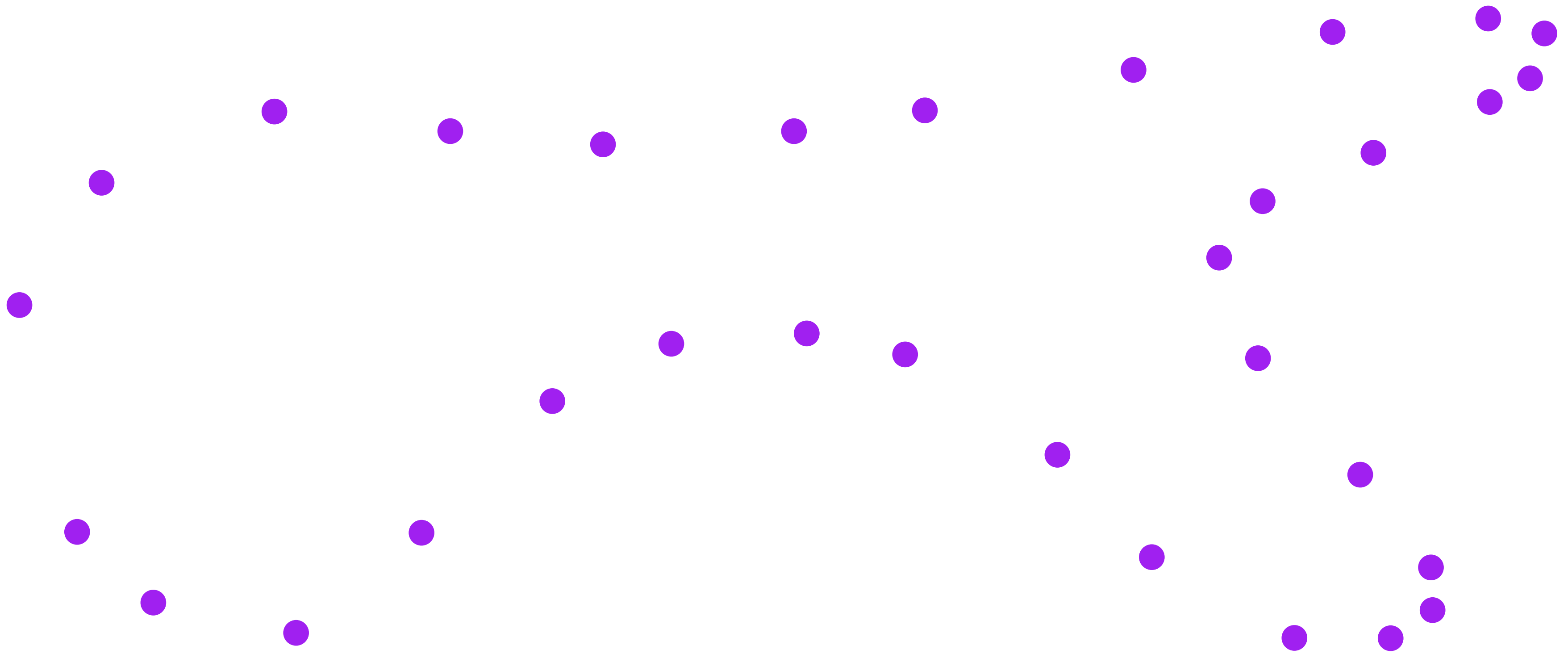


Reconstruction

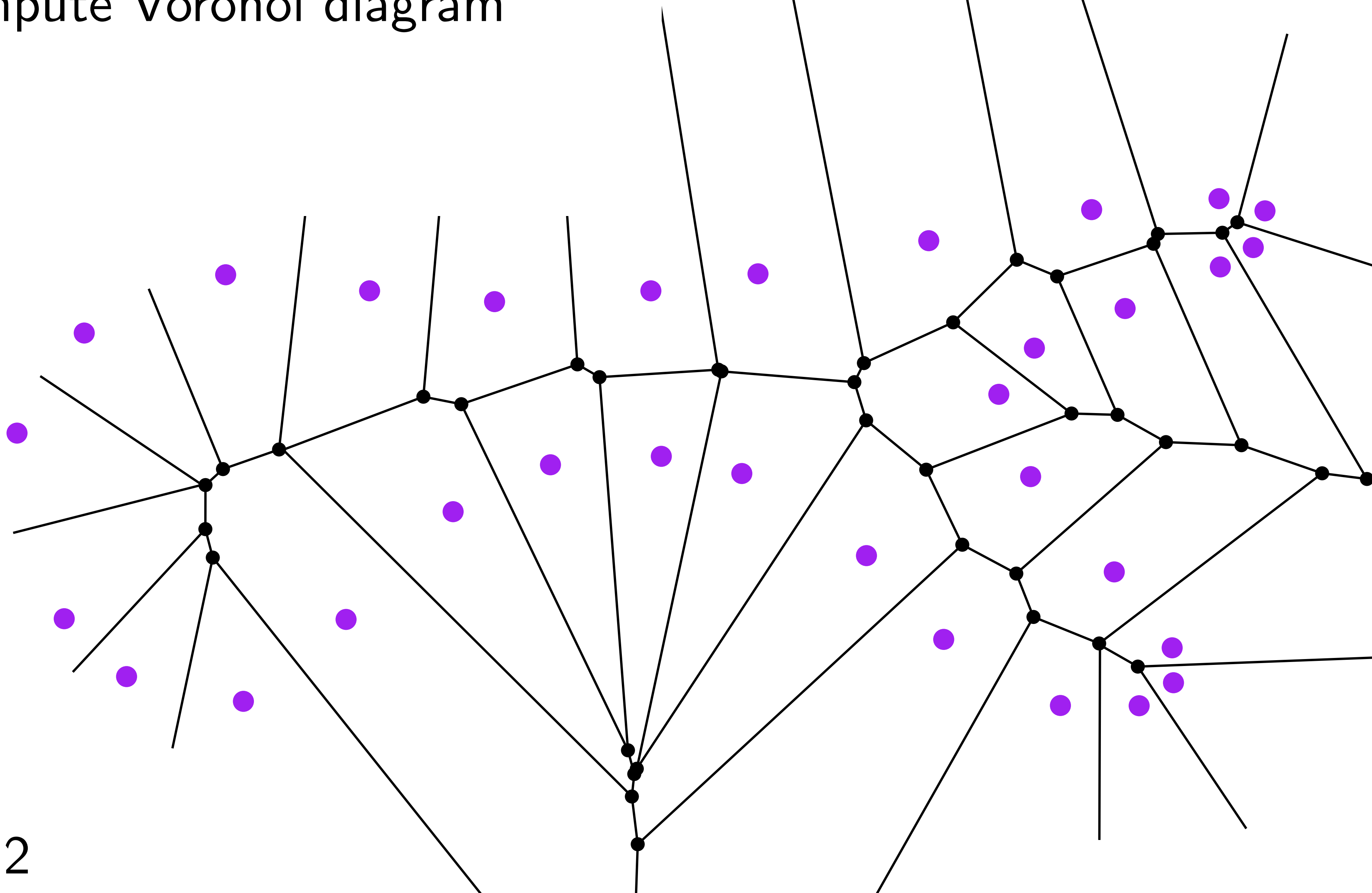
Delaunay is a good start

1-sample is not enough

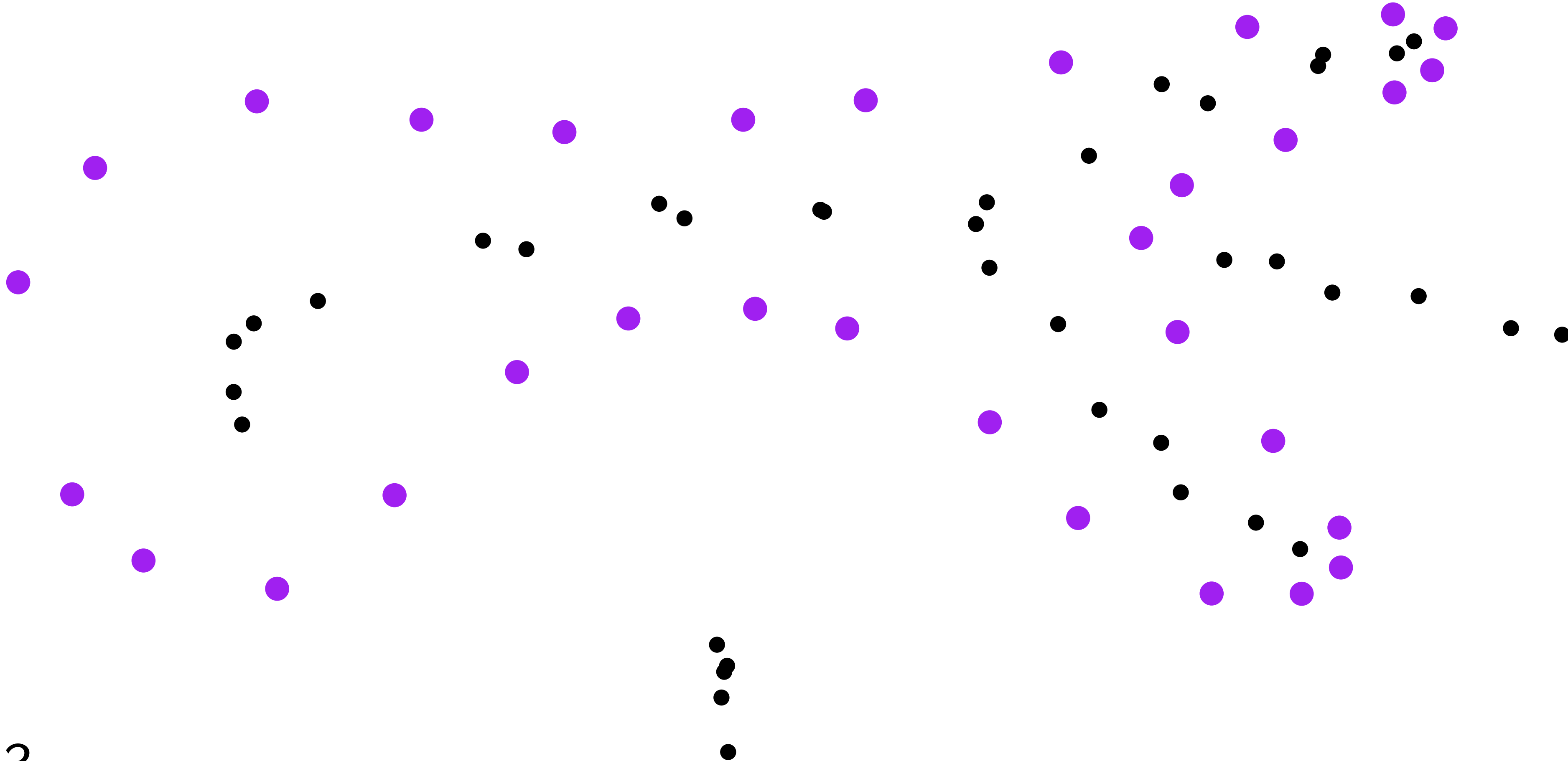




Compute Voronoi diagram



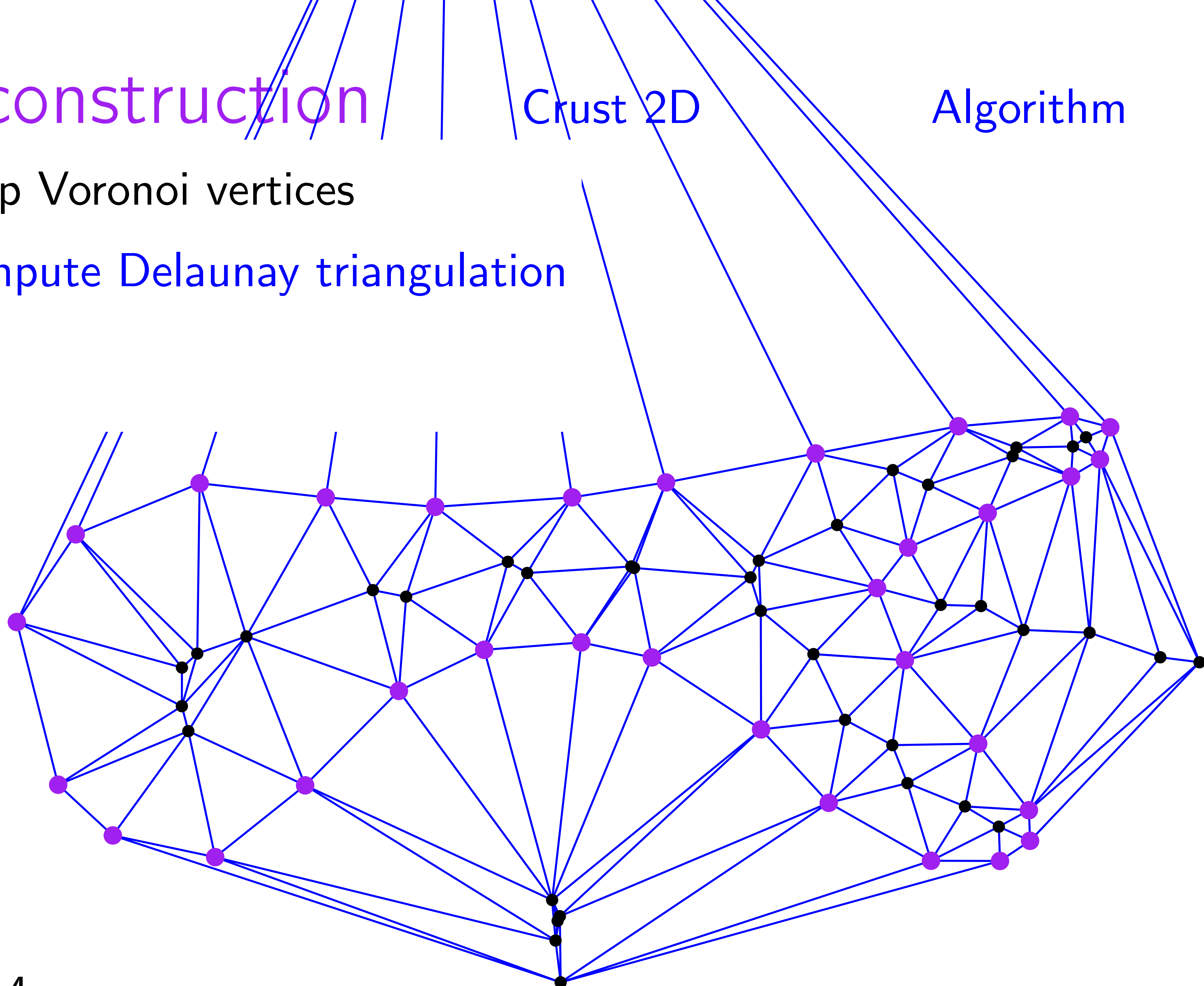
Keep Voronoi vertices



Reconstruction Crust 2D Algorithm

Keep Voronoi vertices

Compute Delaunay triangulation

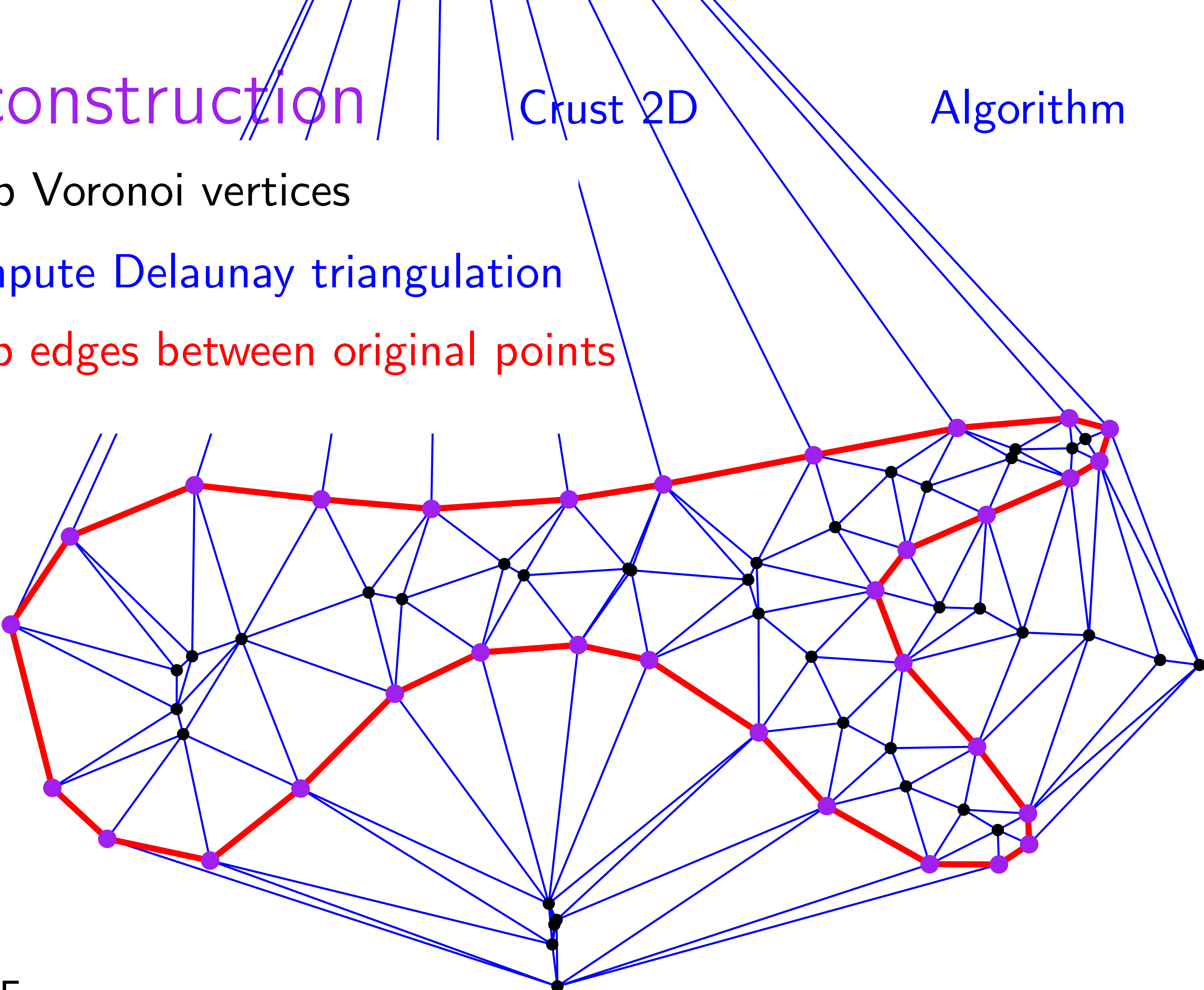


Reconstruction Crust 2D Algorithm

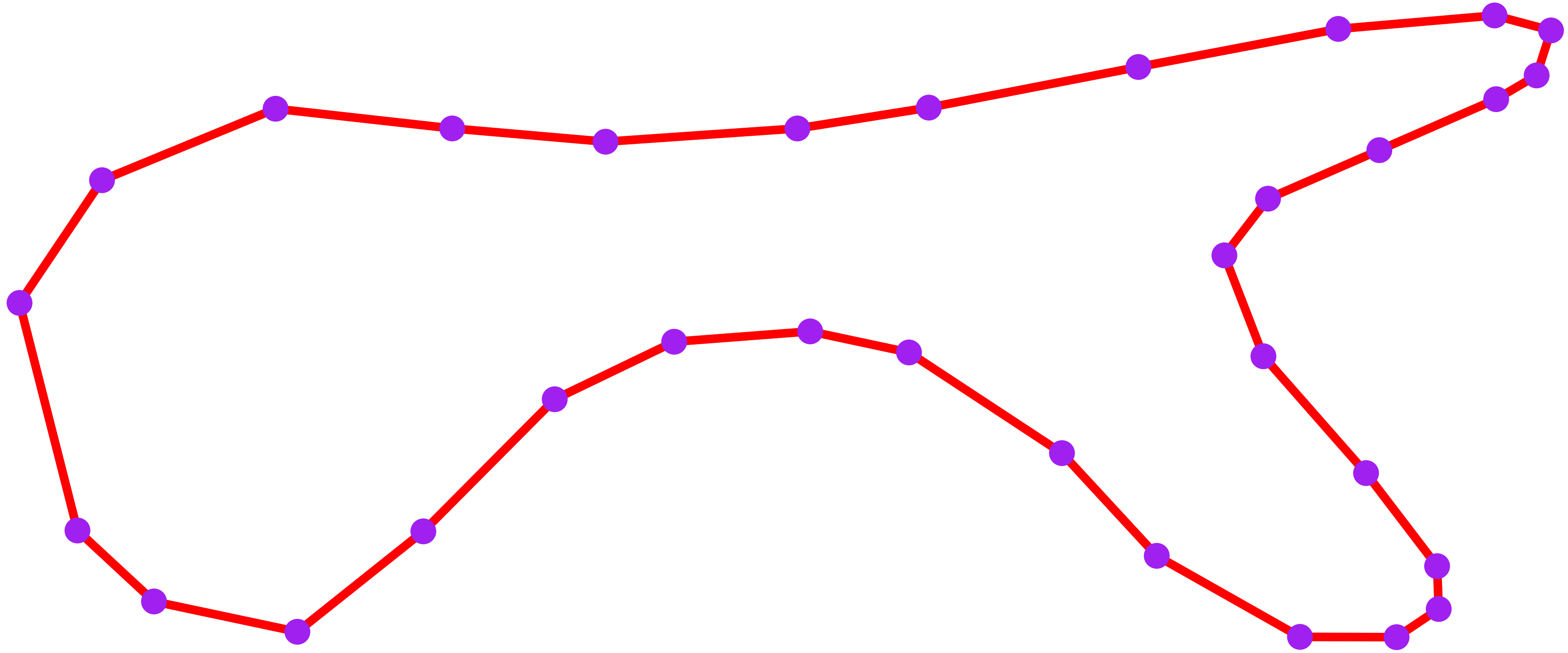
Keep Voronoi vertices

Compute Delaunay triangulation

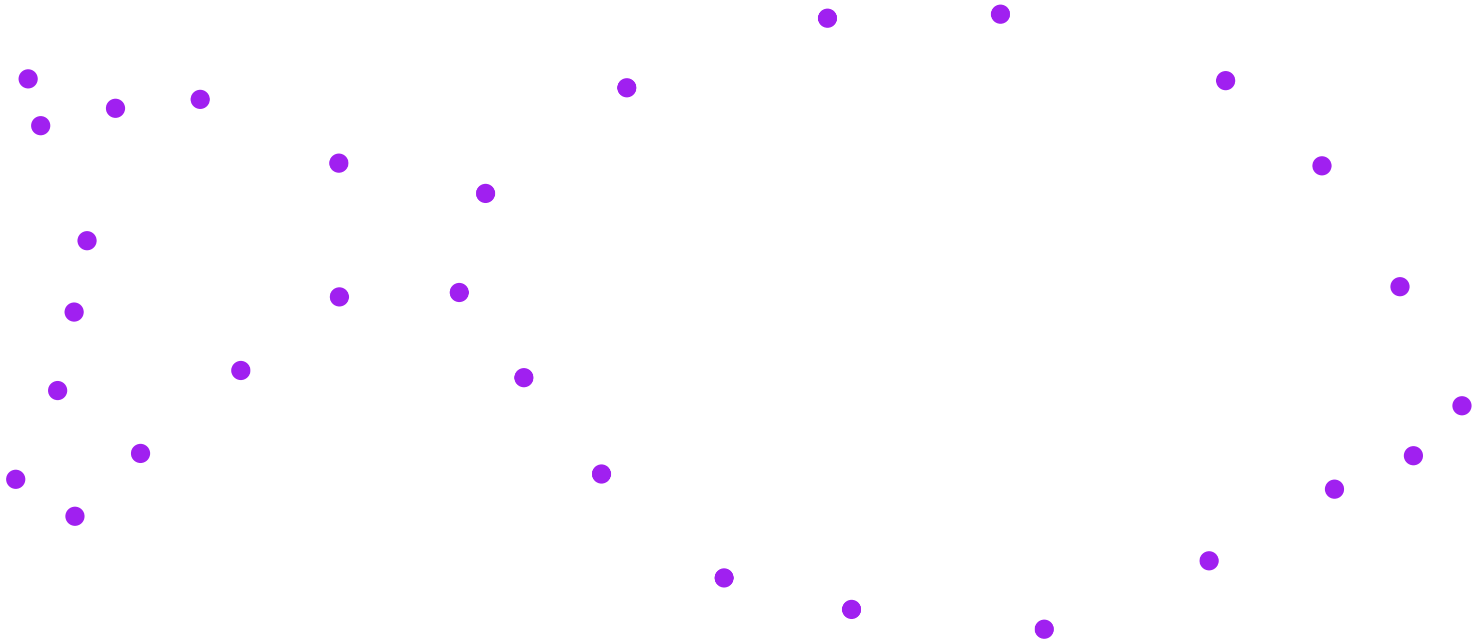
Keep edges between original points

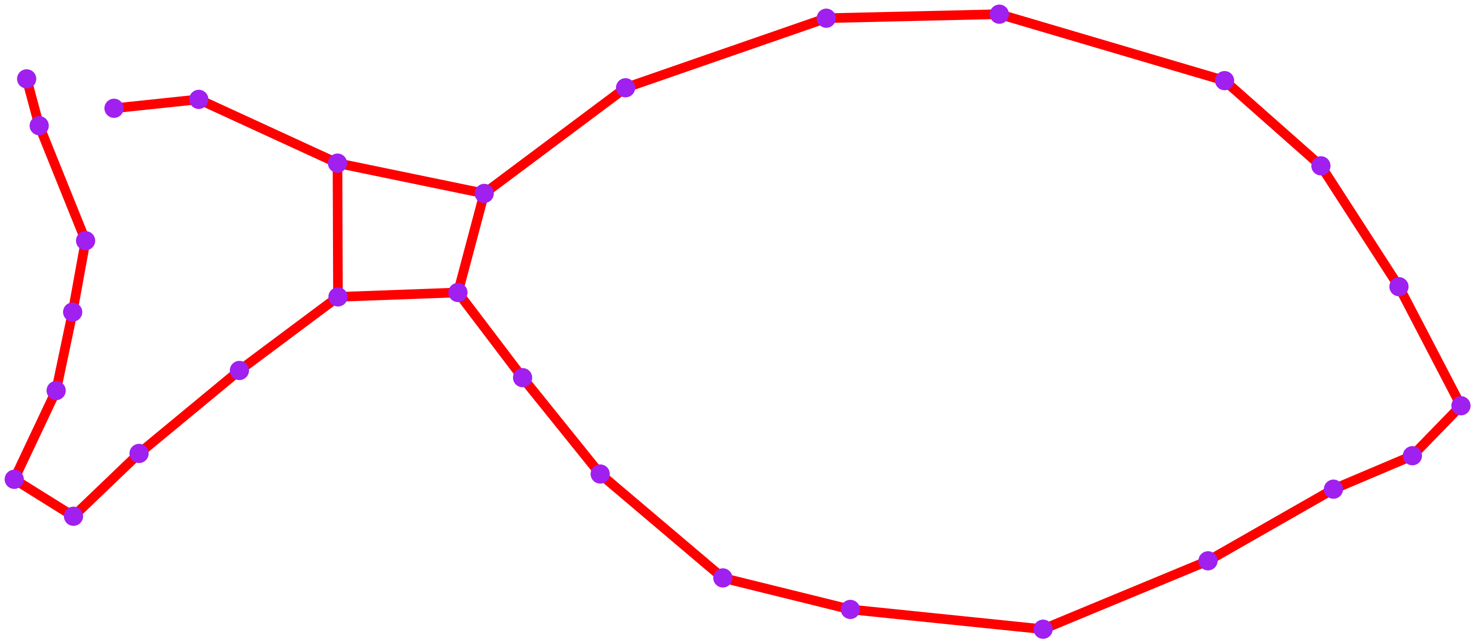


Keep edges between original points

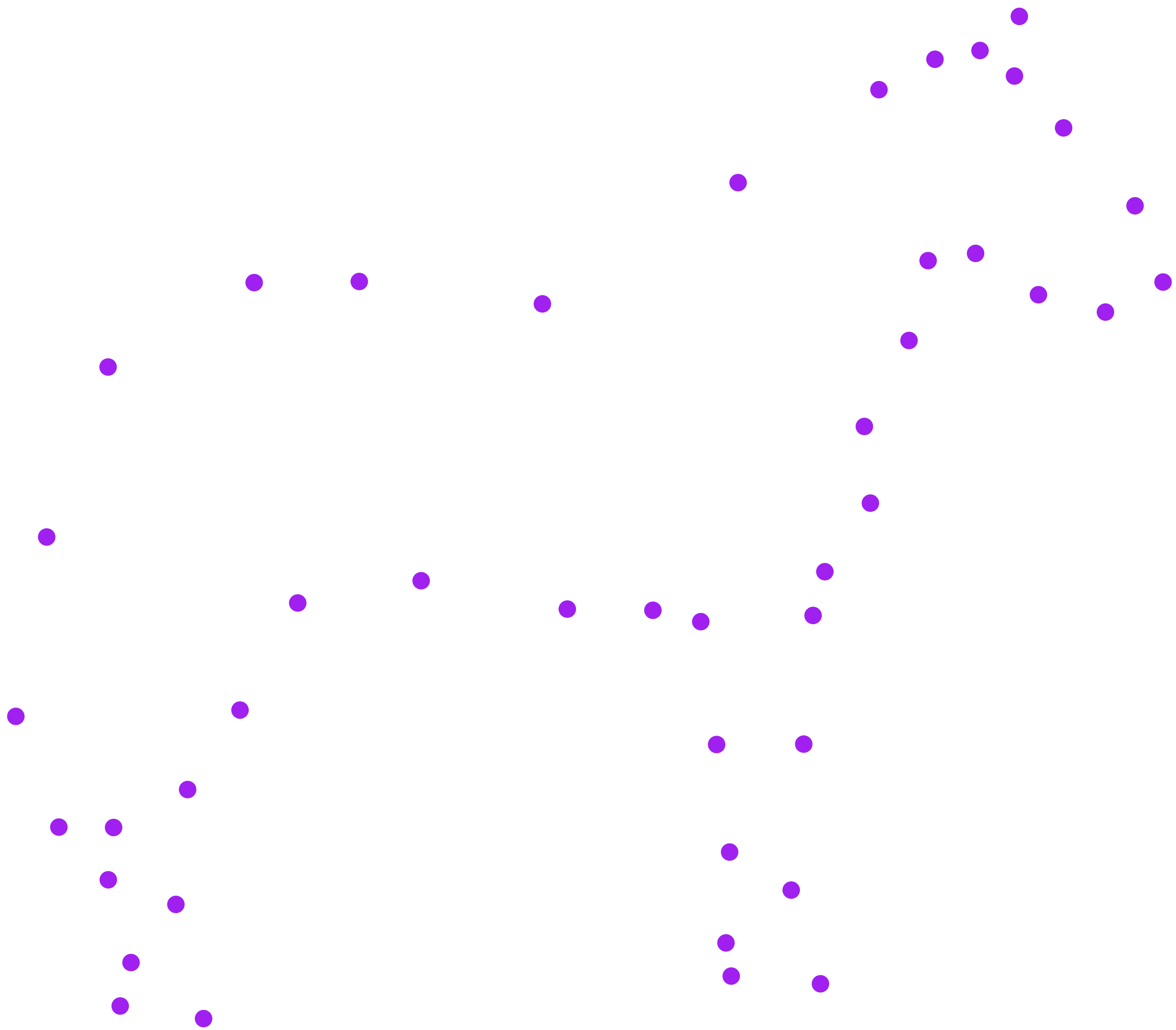


Reconstruction



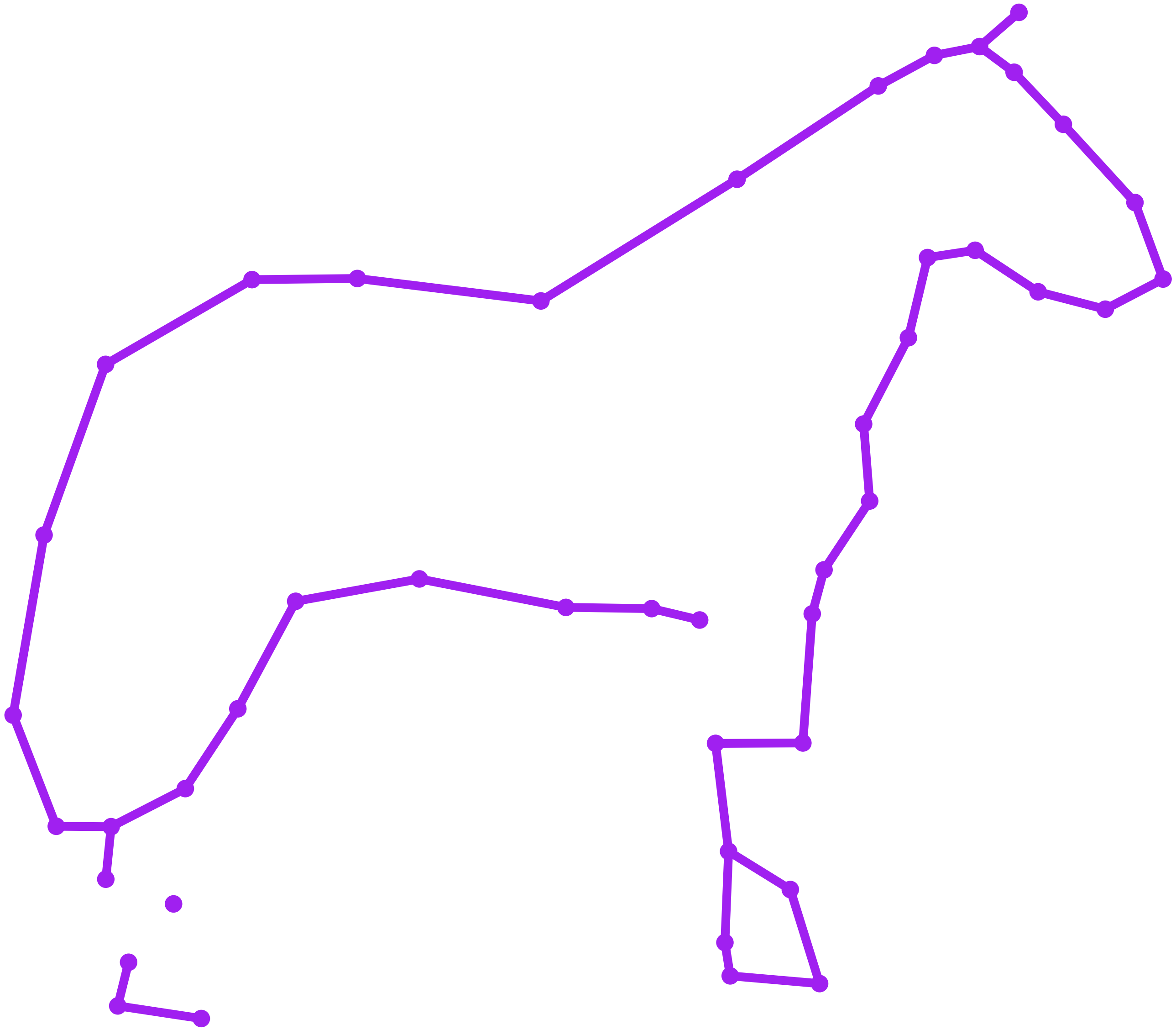


Reconstruction   Crust 2D   Algorithm



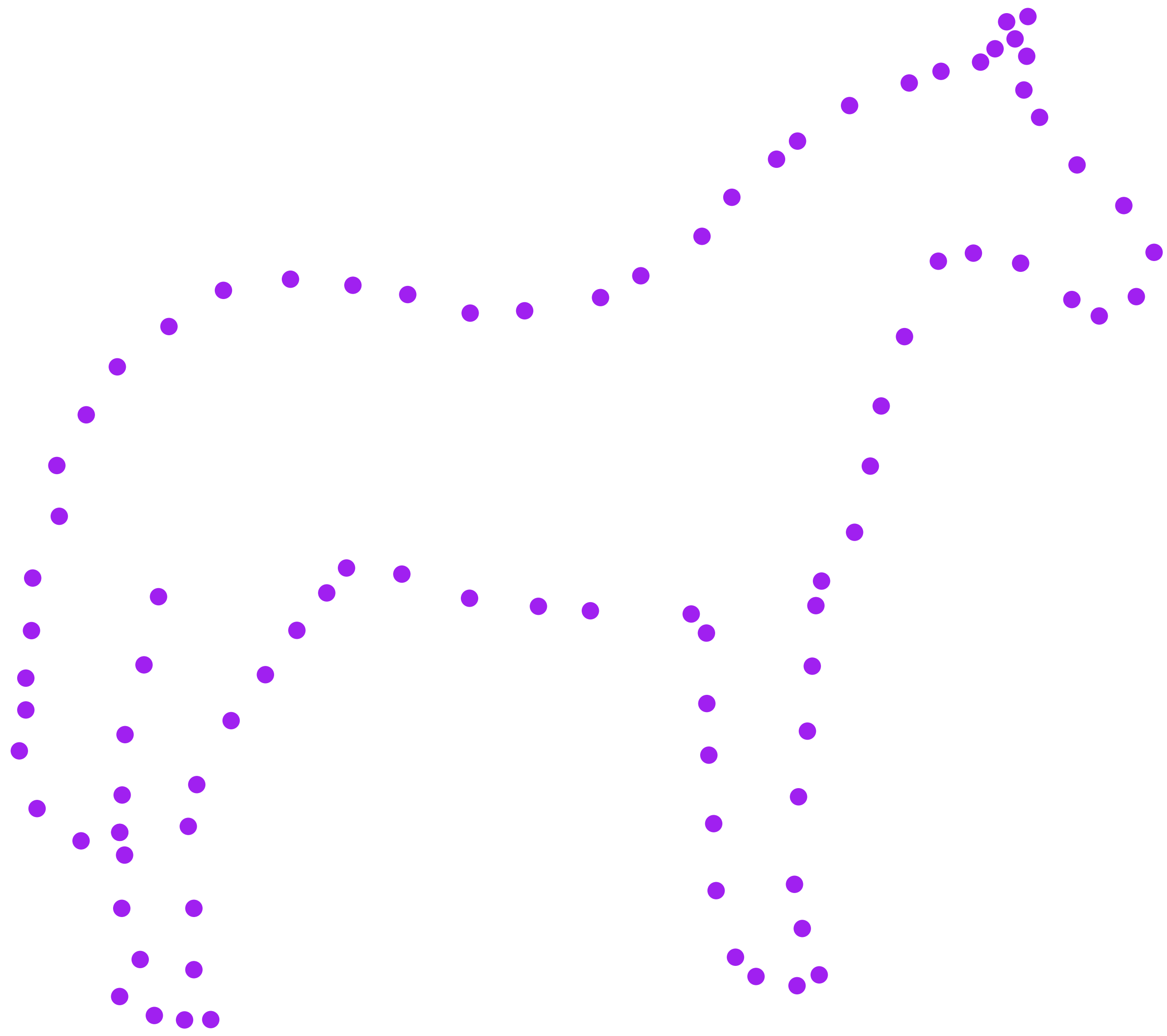
Reconstruction

Crust 2D Algorithm

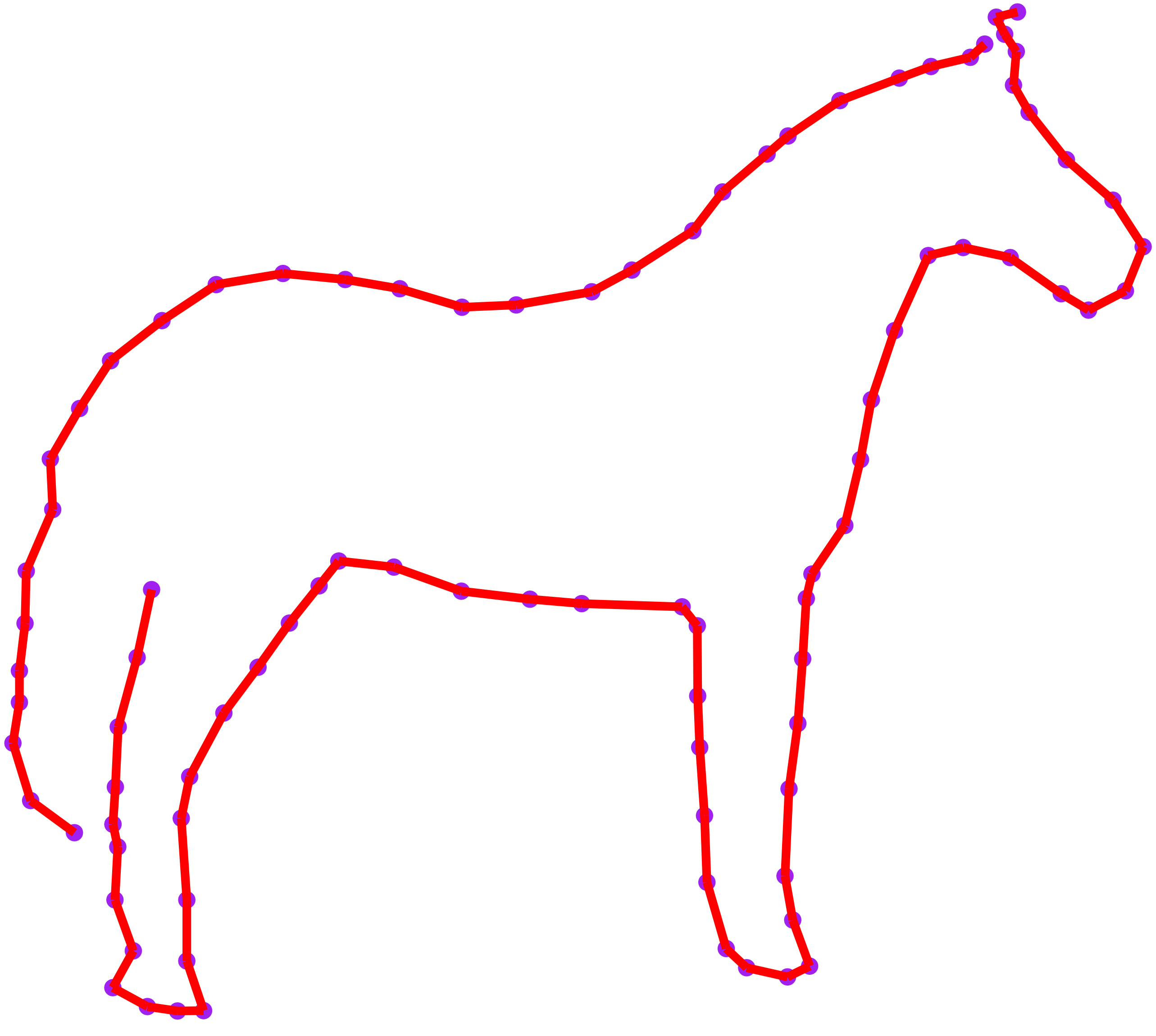


Reconstruction

Reconstruction Crust 2D Algorithm



Reconstruction Crust 2D Algorithm



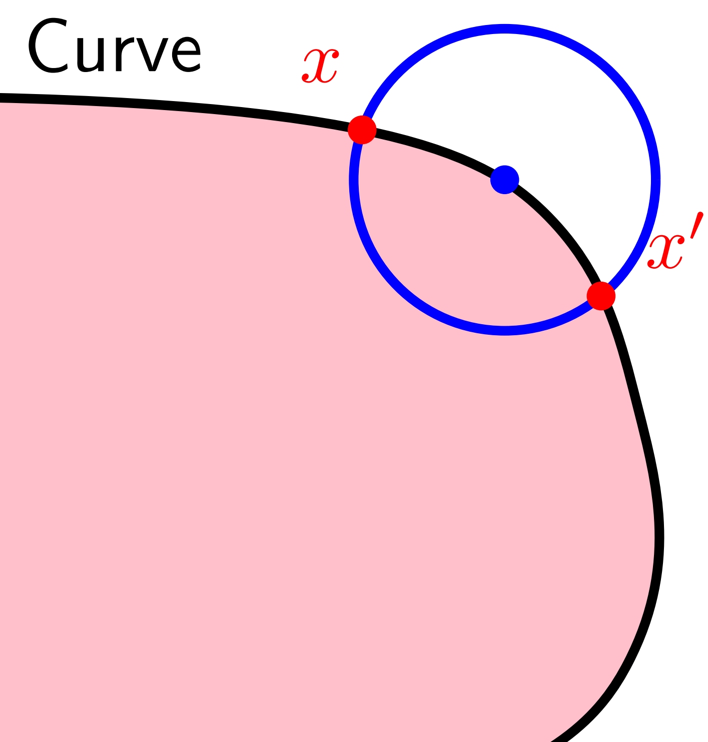
Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

Reconstruction

Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

x, x' two neighboring points on Curve

Circle thru x and x' centered on Curve

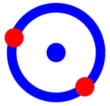


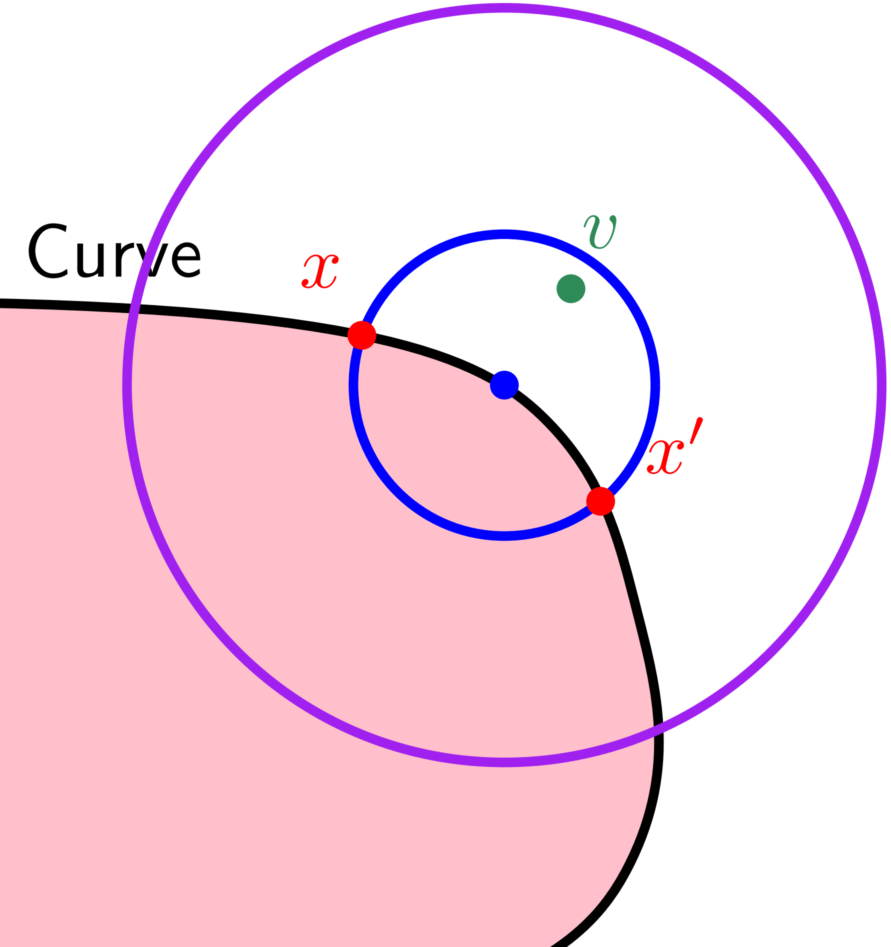
Reconstruction

Theorem: 0.4 sample ⇒ wanted result ⊂ crust

x, x' two neighboring points on Curve

Circle thru x and x' centered on Curve

By contradiction assume $v \in$ 



Reconstruction

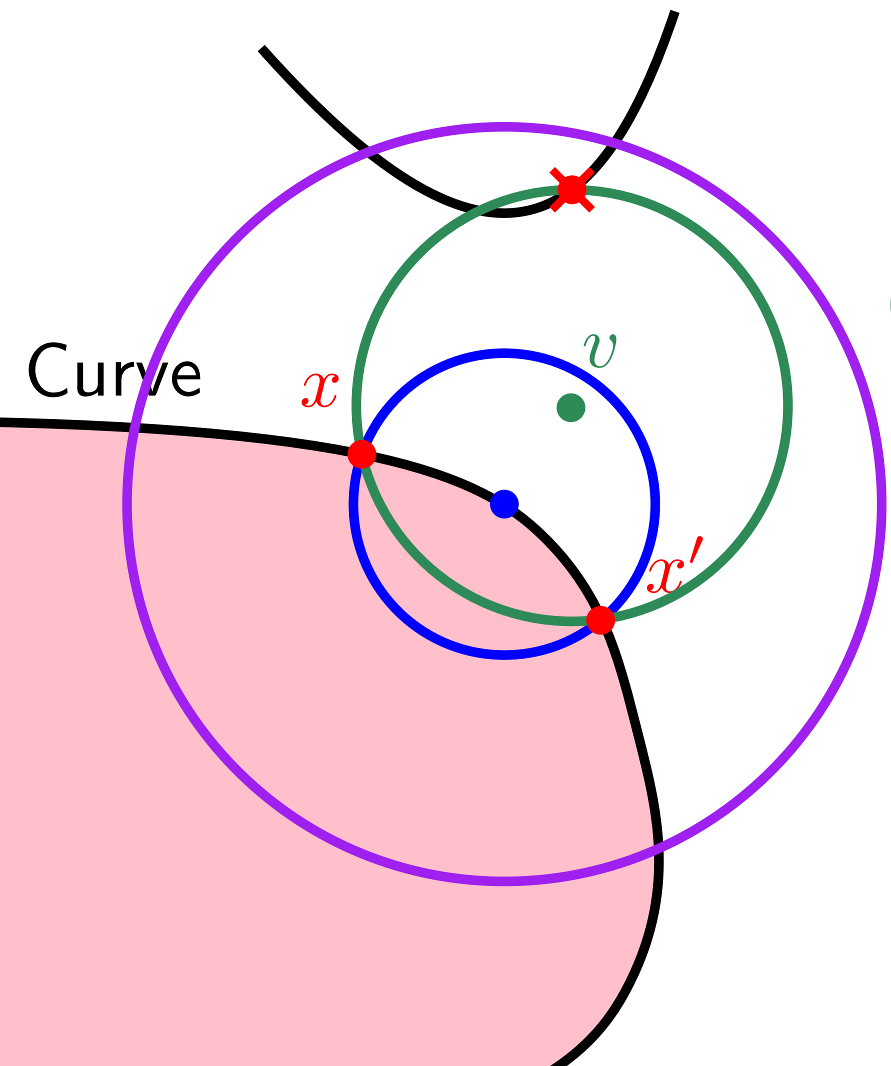
Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

x, x' two neighboring points on Curve

Circle thru x and x' centered on Curve

By contradiction assume $v \in$ ⊙

⊙ intersects another cc of curve
(by Lemma)

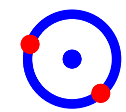


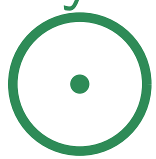
Reconstruction

Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

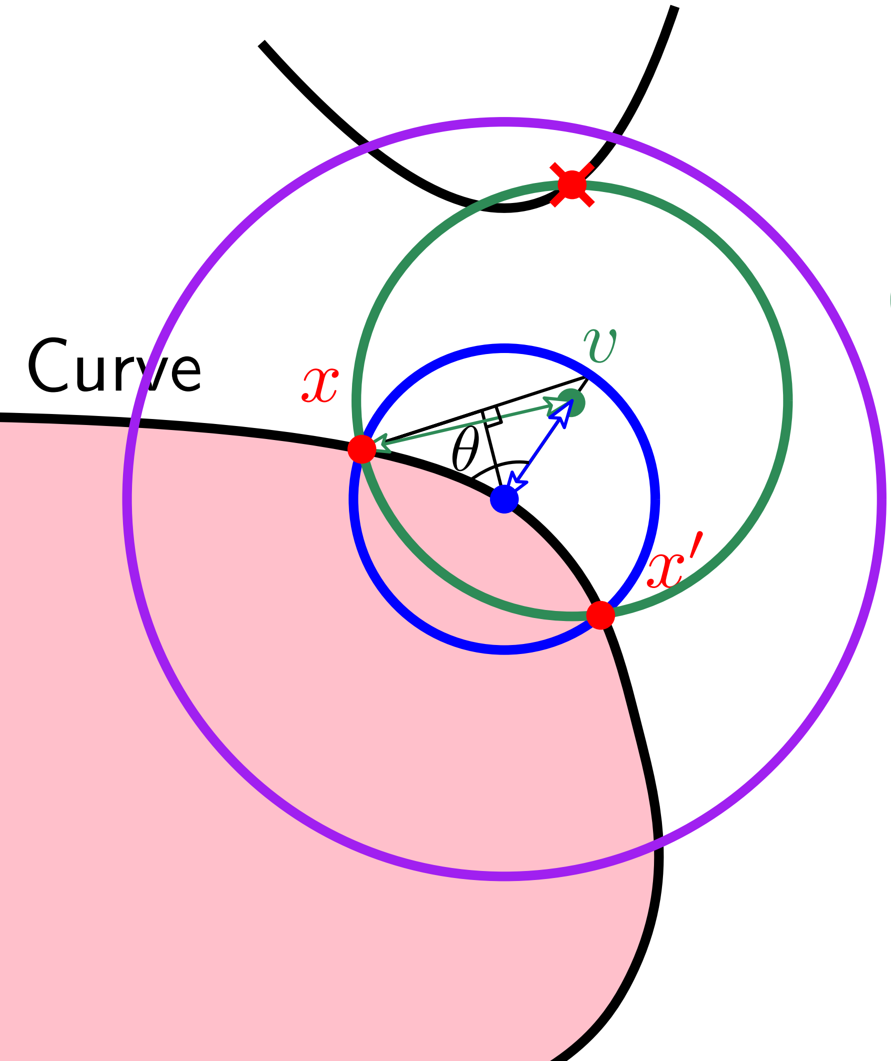
x, x' two neighboring points on Curve

Circle thru x and x' centered on Curve

By contradiction assume $v \in$ 

 intersects another cc of curve (by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

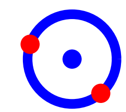


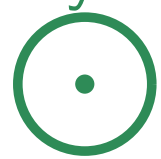
Reconstruction

Theorem: 0.4 sample ⇒ wanted result ⊂ crust

x, x' two neighboring points on Curve

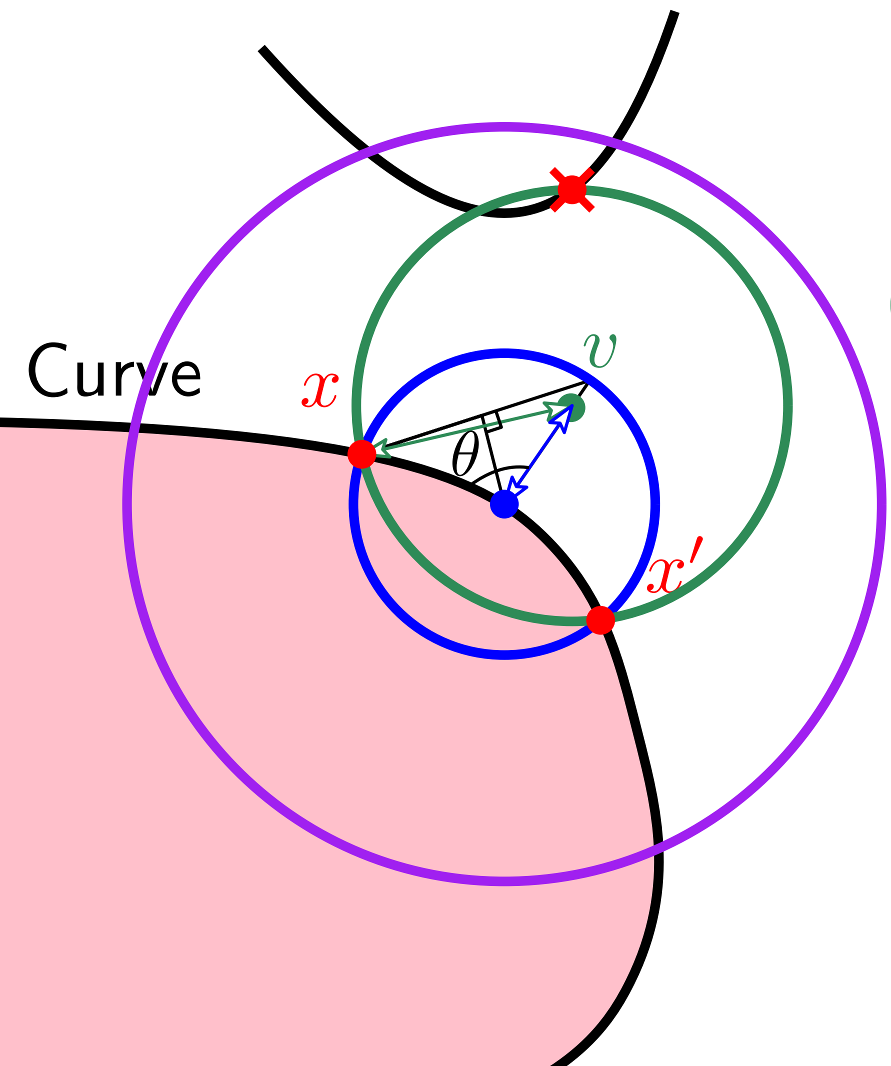
Circle thru x and x' centered on Curve

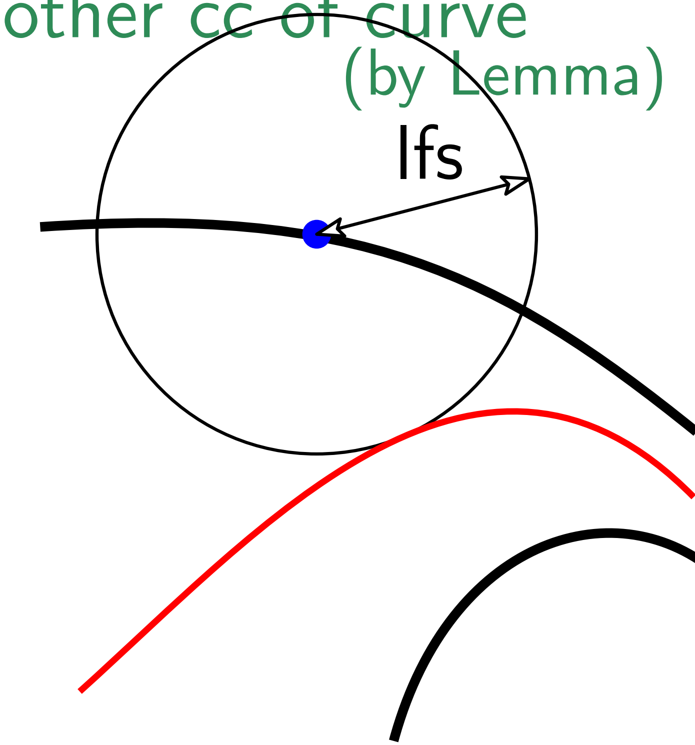
By contradiction assume $v \in$ 

 intersects another cc of curve (by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq$





Reconstruction

Crust 2D

0.4 sample ⇒ wanted result ⊂ crust

Theorem: 0.4 sample ⇒ wanted result ⊂ crust

x, x' two neighboring points on Curve

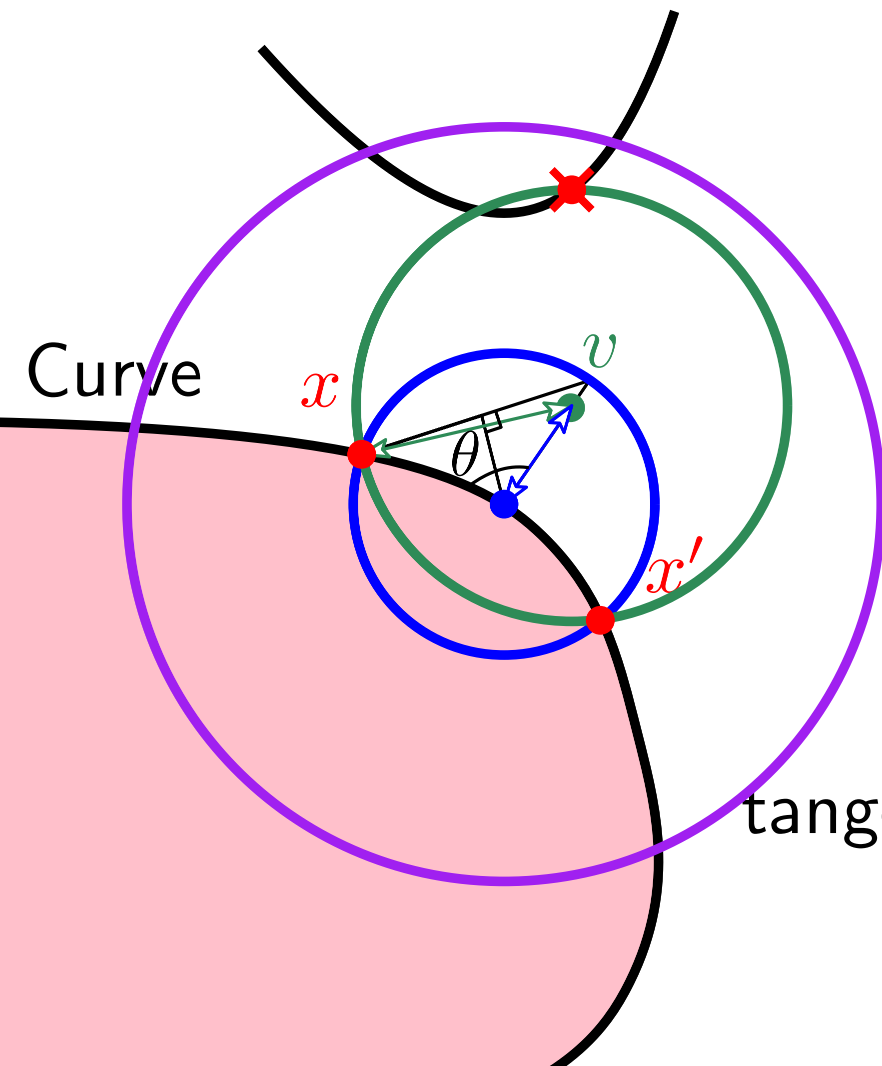
Circle thru x and x' centered on Curve

By contradiction assume $v \in$ (blue circle)

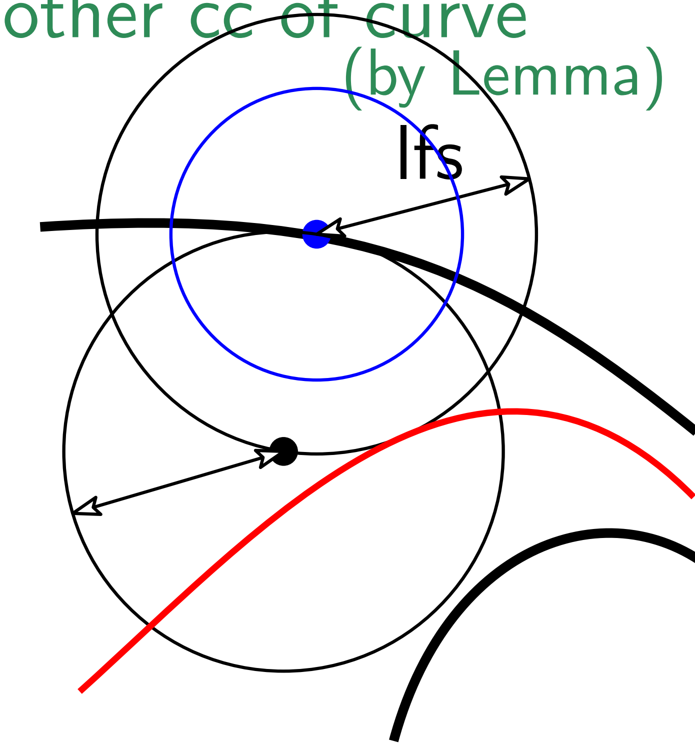
(green circle) intersects another cc of curve (by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq$



tangent disk is empty



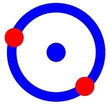
Reconstruction

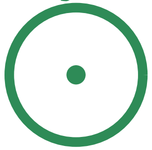
Crust 2D 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

x, x' two neighboring points on Curve

Circle thru x and x' centered on Curve

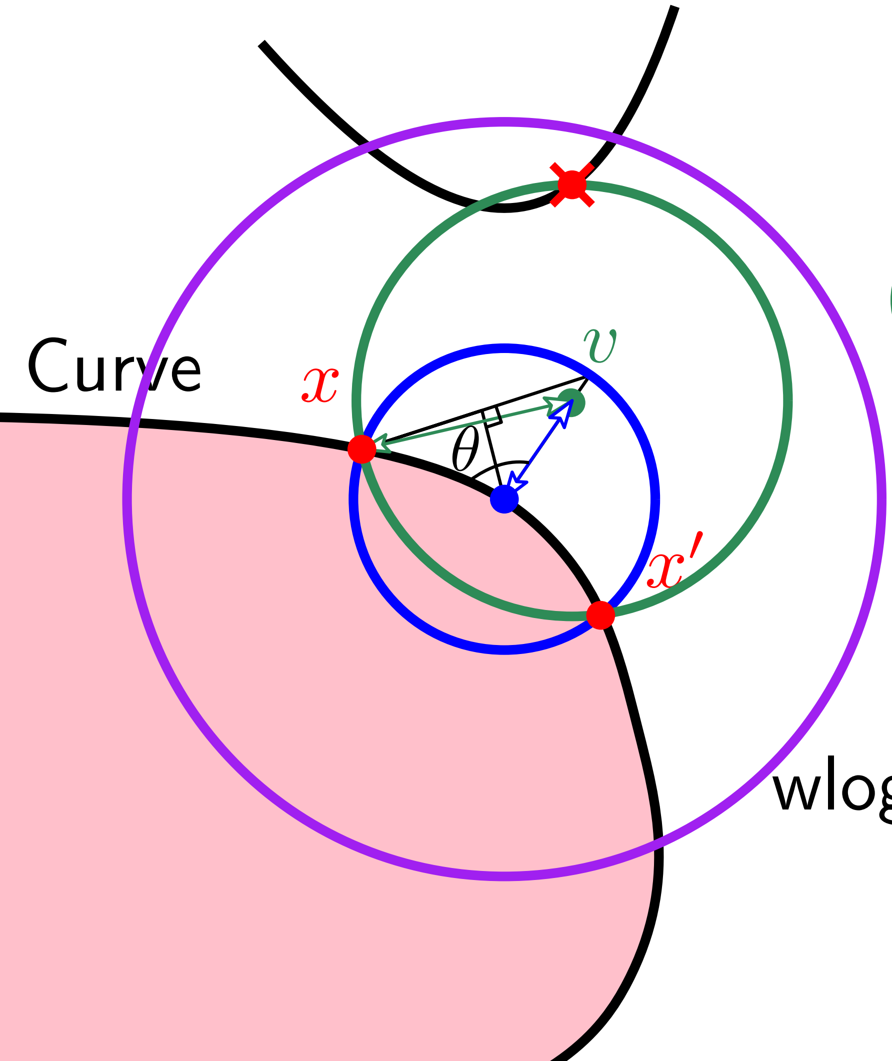
By contradiction assume $v \in$ 

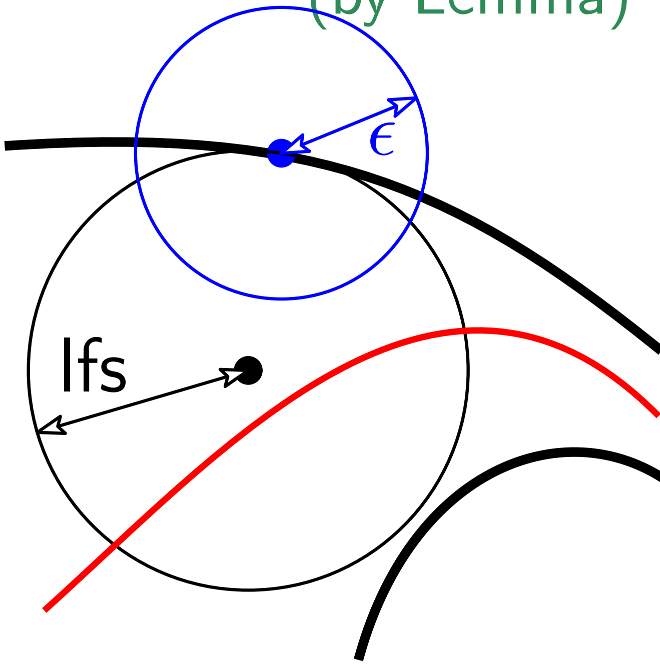
 intersects another cc of curve (by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq$

wlog lfs=1 and $r \leq \epsilon$





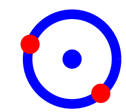
Reconstruction

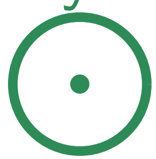
Crust 2D

Theorem: **0.4 sample $\Rightarrow$ wanted result $\subset$ crust**

x, x' two neighboring points on Curve

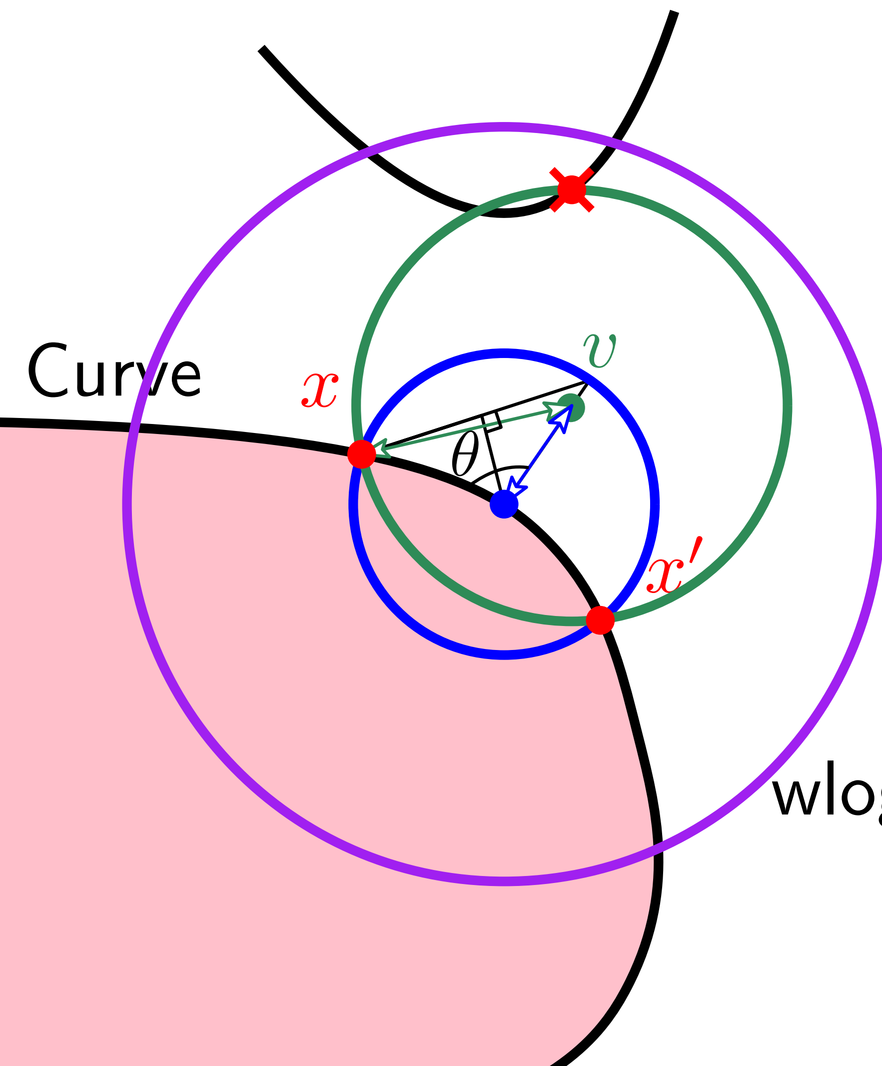
Circle thru x and x' centered on Curve

By contradiction assume $v \in$ 

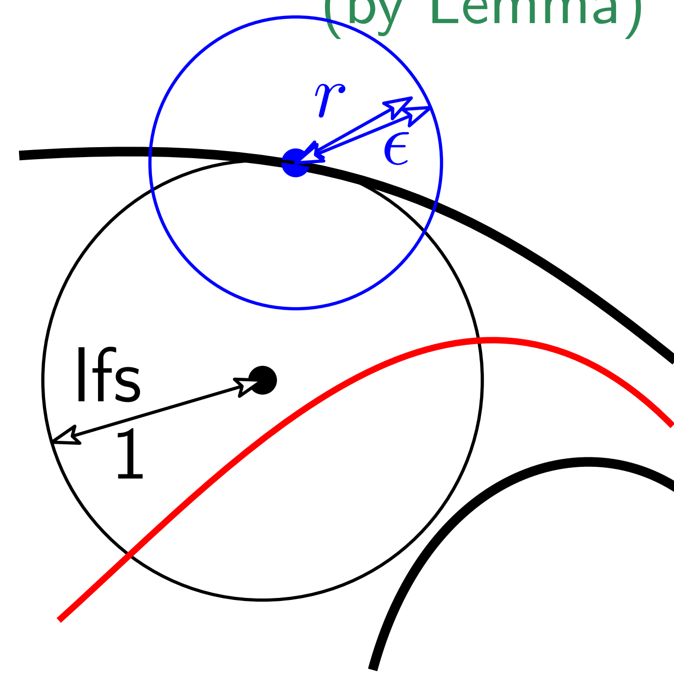
 intersects another cc of curve (by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq$



wlog lfs=1 and $r \leq \epsilon$



Reconstruction

Crust 2D   0.4 sample ⇒ wanted result ⊂ crust

Theorem: 0.4 sample ⇒ wanted result ⊂ crust

x, x' two neighboring points on Curve

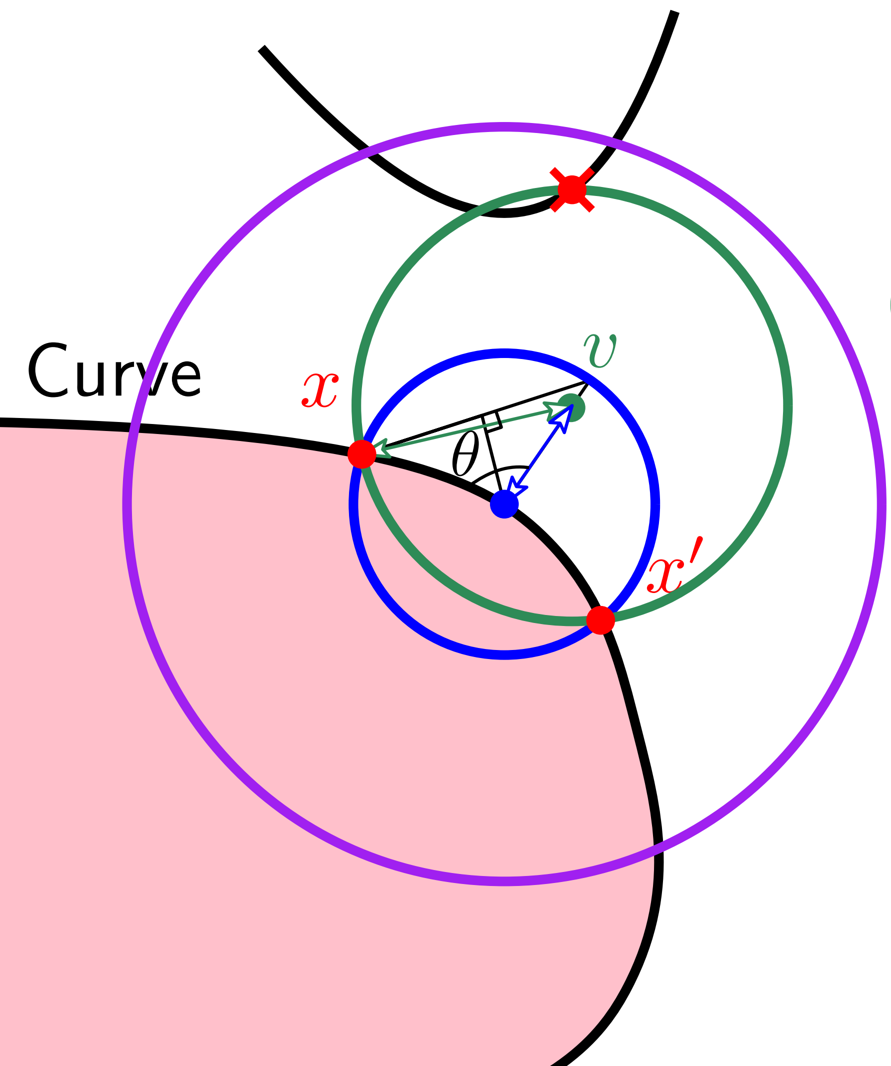
Circle thru x and x' centered on Curve

By contradiction assume $v \in$ ⊙

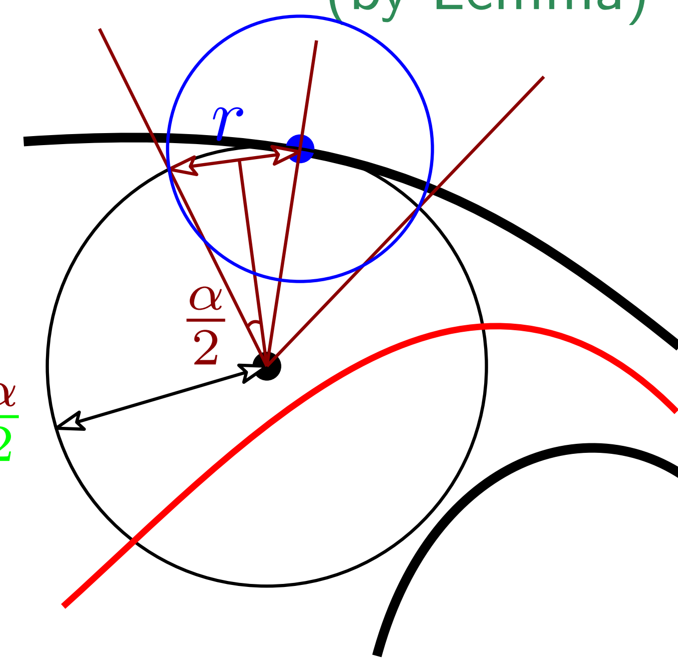
⊙ intersects another cc of curve (by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq$



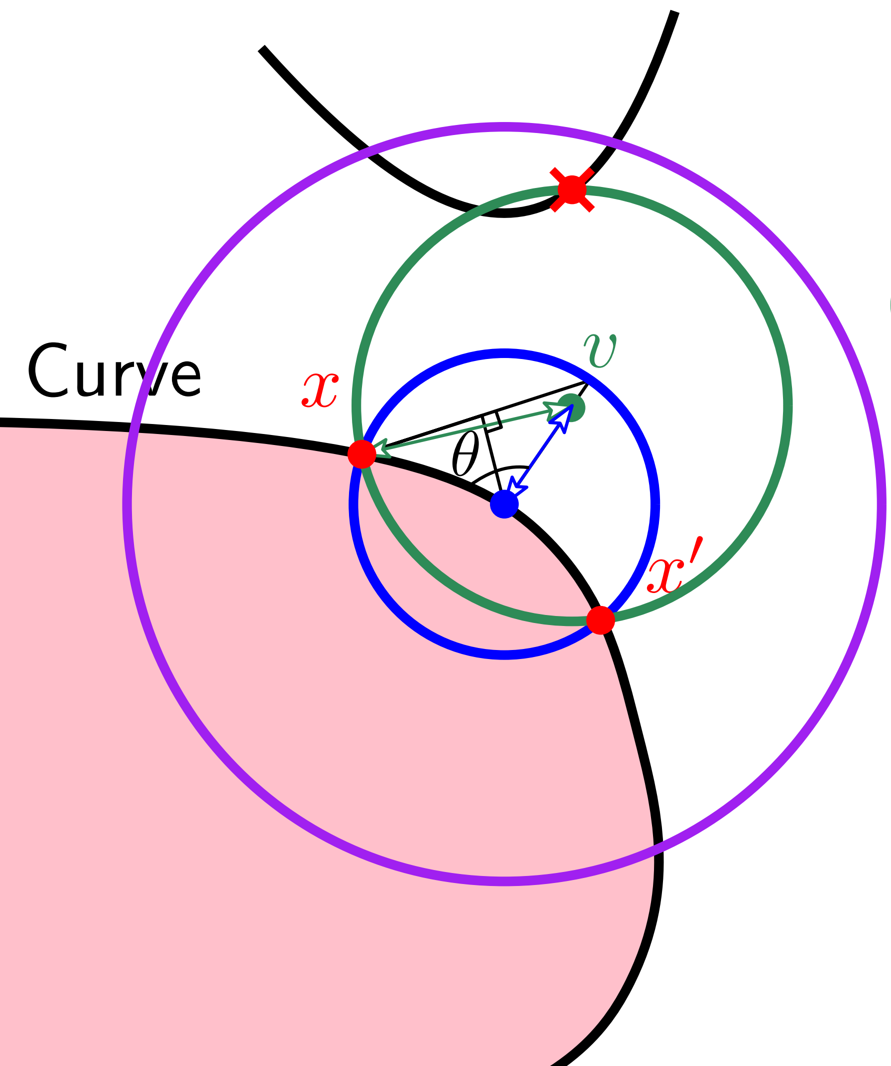
$r = 2 \sin \frac{\alpha}{2}$



Reconstruction

Crust 2D 0.4 sample ⇒ wanted result ⊂ crust

Theorem: 0.4 sample ⇒ wanted result ⊂ crust



x, x' two neighboring points on Curve

Circle thru x and x' centered on Curve

By contradiction assume $v \in$

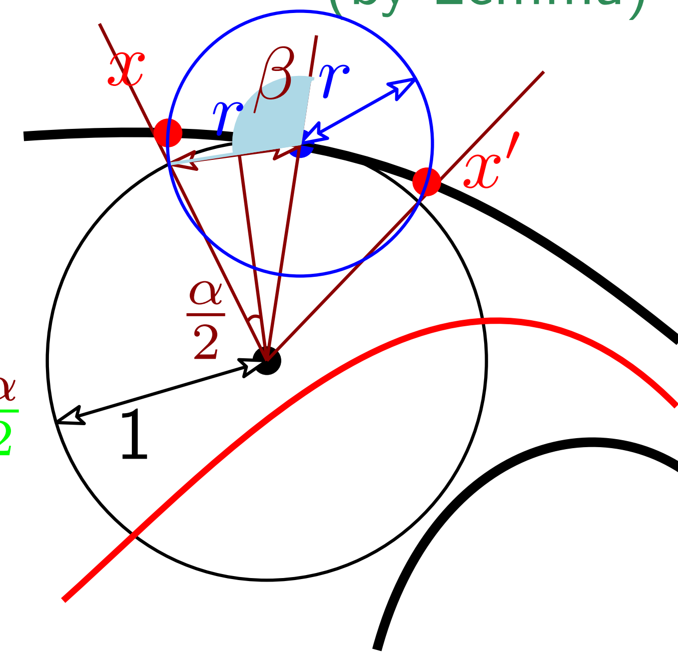
intersects another cc of curve

(by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq \beta = \pi - \frac{\pi - \alpha}{2}$

$r = 2 \sin \frac{\alpha}{2}$



Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

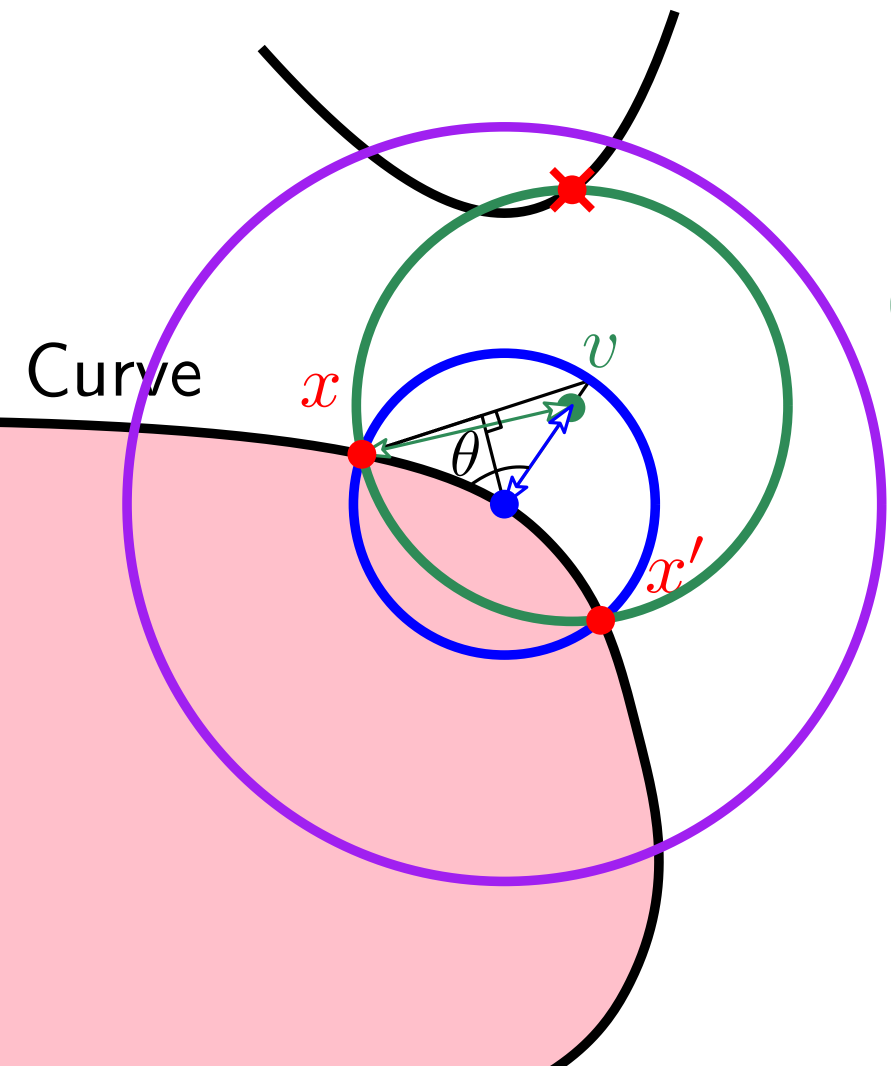
x, x' two neighboring points on Curve

Circle thru x and x' centered on Curve

By contradiction assume $v \in$

intersects another cc of curve

(by Lemma)

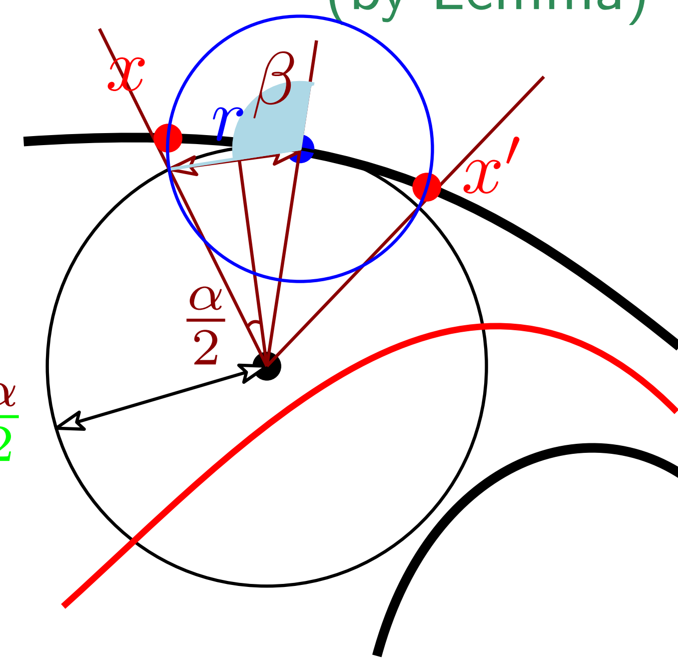


$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq \beta = \pi - \frac{\pi-\alpha}{2}$

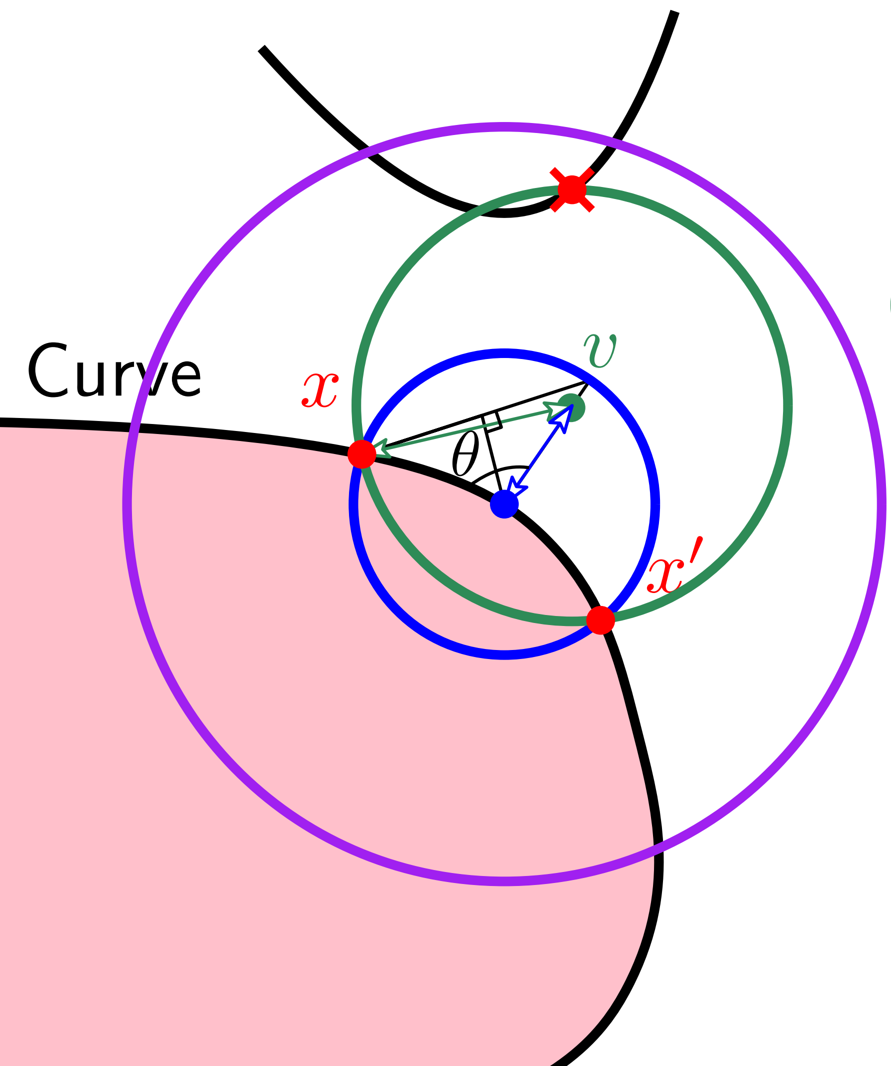
$\leq \frac{\pi}{2} + \text{arcsin} \frac{r}{2}$

$r = 2 \sin \frac{\alpha}{2}$



Reconstruction

Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust



x, x' two neighboring points on Curve

Circle thru x and x' centered on Curve

By contradiction assume $v \in \odot$

$\odot$ intersects another cc of curve (by Lemma)

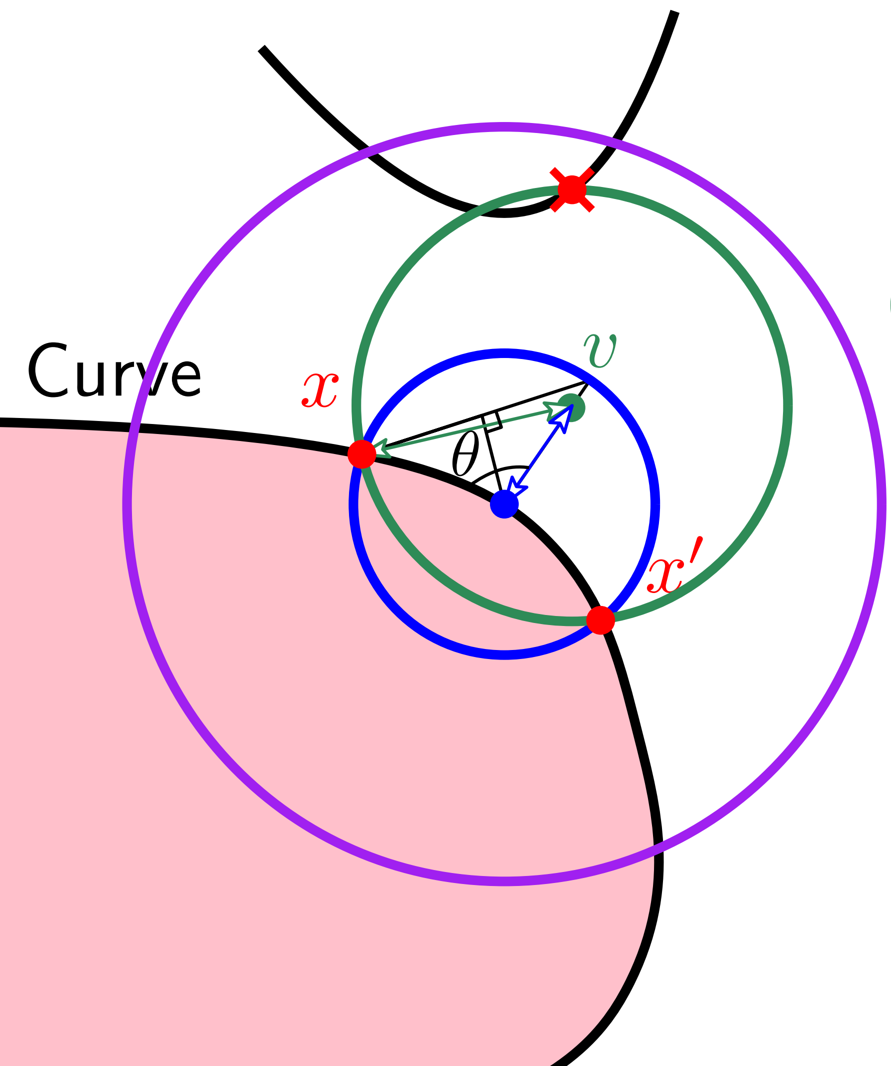
$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq \frac{\pi}{2} + \arcsin \frac{r}{2}$

$\| \bullet \times \| \leq \| \bullet \; \bullet \| + \| \bullet \times \|$

Reconstruction

Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust



x, x' two neighboring points on Curve

Circle thru x and x' centered on Curve

By contradiction assume $v \in$

intersects another cc of curve

(by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

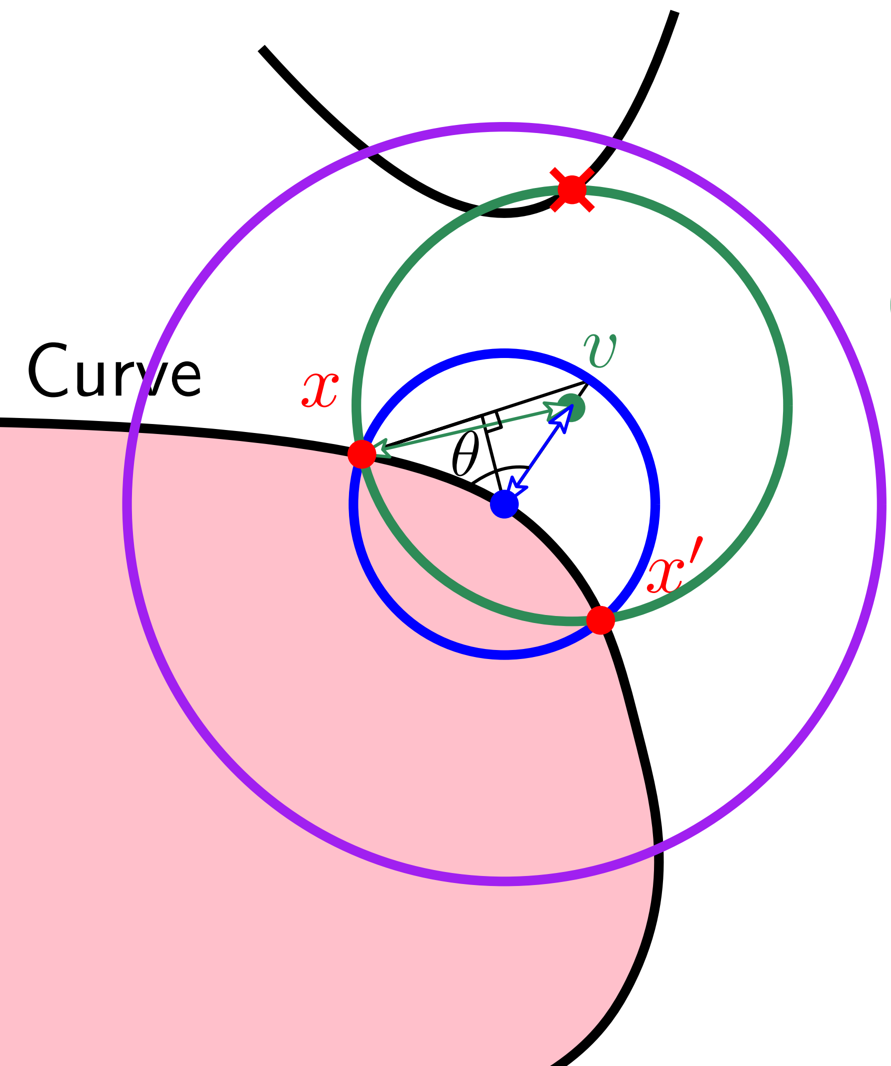
$\theta \leq \frac{\pi}{2} + \text{arcsin}\frac{r}{2}$

$\|\bullet \times\| \leq \|\bullet\ \bullet\| + \|\bullet \times\|$

$\leq r + 2r \sin\left(\frac{\pi}{4} + \frac{1}{2}\text{arcsin}\,\frac{r}{2}\right)$

Reconstruction

Theorem: 0.4 sample ⇒ wanted result ⊂ crust



x, x' two neighboring points on Curve

Circle thru x and x' centered on Curve

By contradiction assume $v \in$ ⊙

⊙ intersects another cc of curve

(by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq \frac{\pi}{2} + \text{arcsin} \frac{r}{2}$

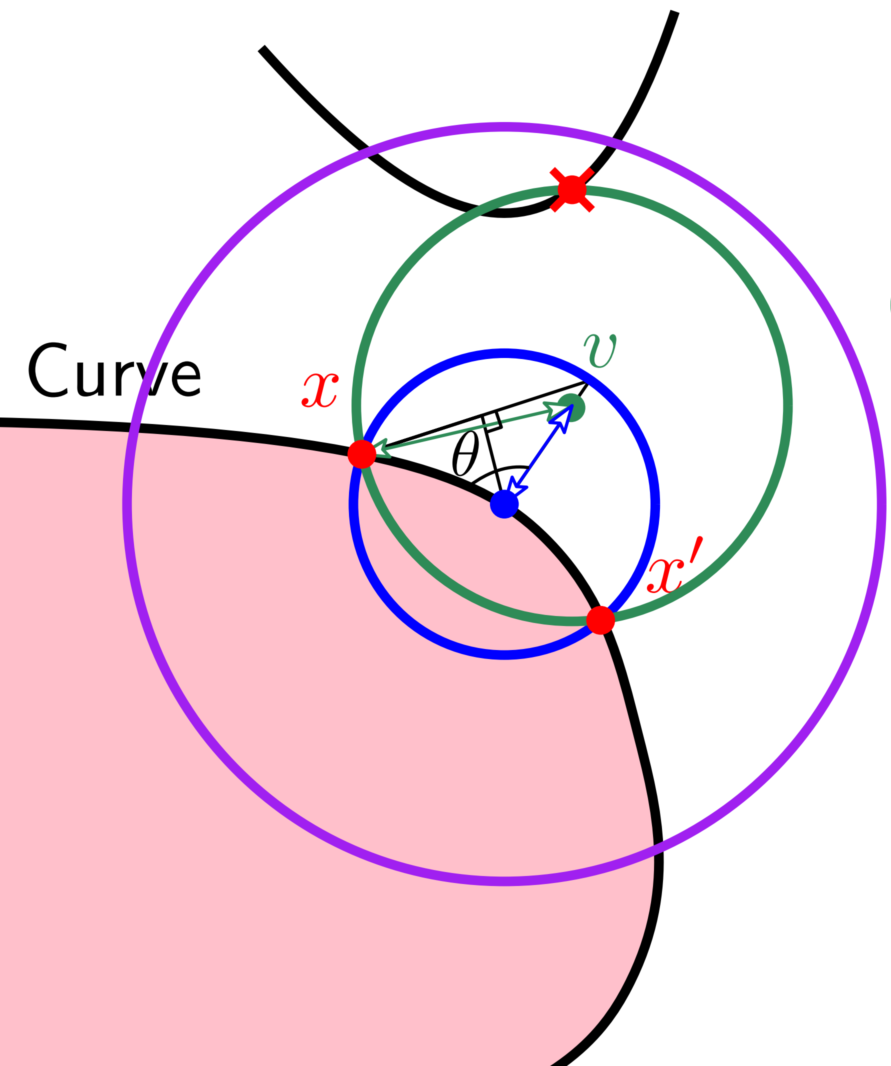
$\|$● ✖$\| \leq \|$● ●$\| + \|$● ✖$\|$

$\leq r + 2r \sin\left(\frac{\pi}{4} + \frac{1}{2} \arcsin \frac{r}{2}\right)$

if $\|$● ✖$\|$ $\leq$lfs$= 1$ contradiction is reached

Reconstruction

Theorem: 0.4 sample ⇒ wanted result ⊂ crust



x, x' two nei

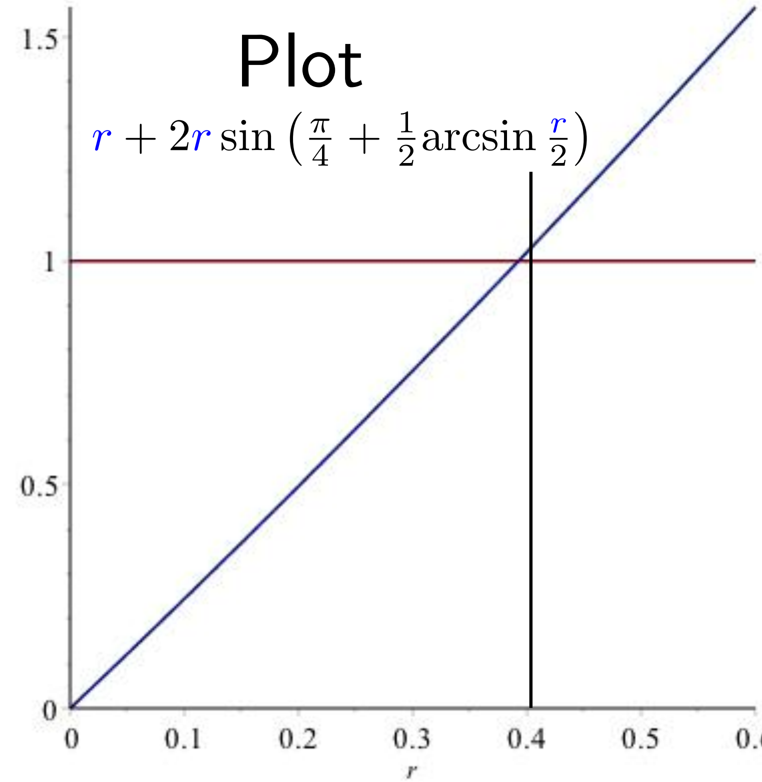
Circle thru x

By contradict

intersect

$R \leq 2r \sin\frac{\theta}{2}$

$\theta \leq \frac{\pi}{2} + \text{arcs}$



‖•✖‖ ≤ ‖• •‖ + ‖•✖‖

$\leq r + 2r\sin\left(\frac{\pi}{4} + \frac{1}{2}\arcsin\frac{r}{2}\right)$

if ‖•✖‖ $\leq$ lfs $= 1$ contradiction is reached

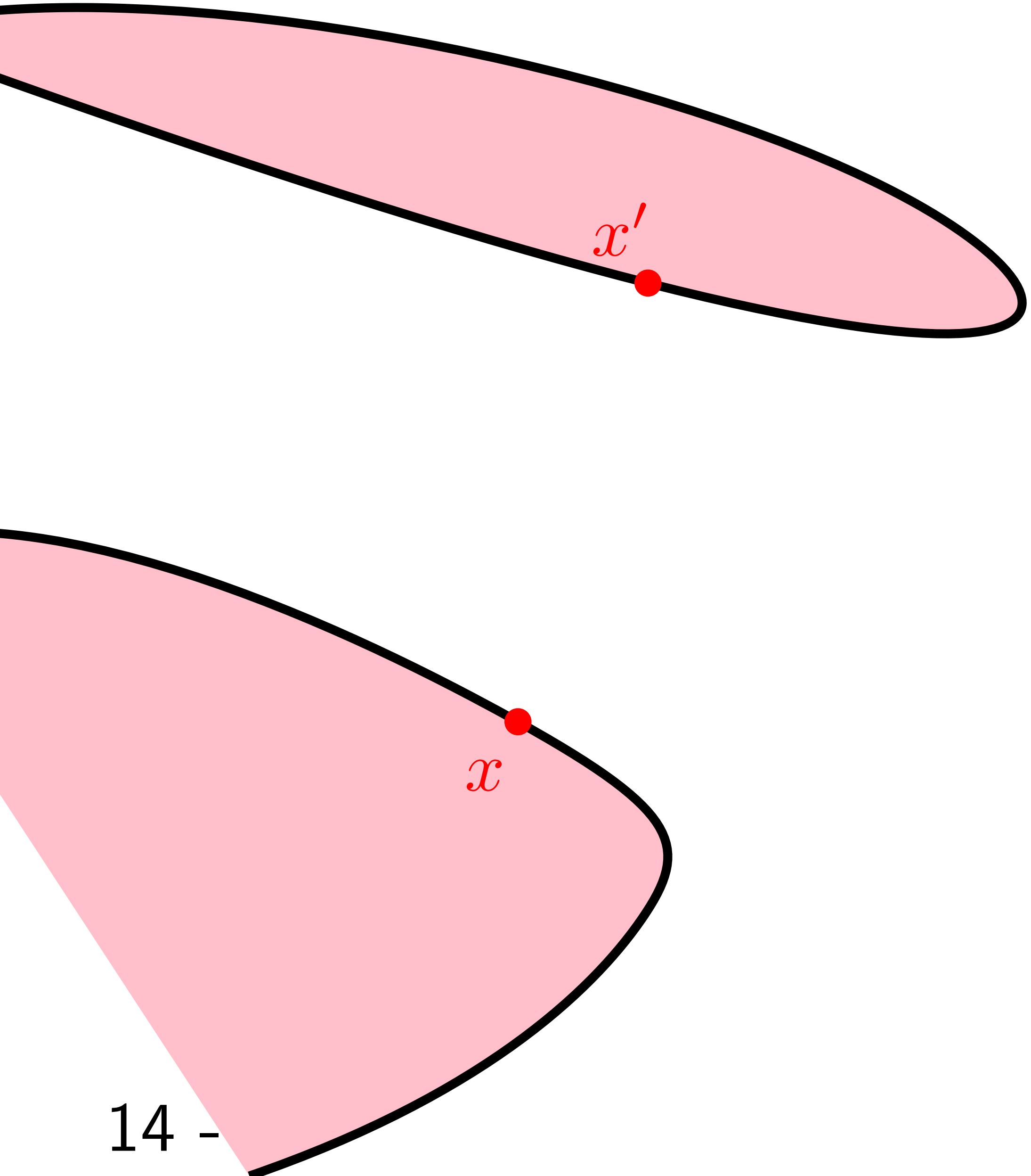
Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

Reconstruction

Theorem: 0.25 sample $\Rightarrow$ crust $\subset$ wanted result

Reconstruction

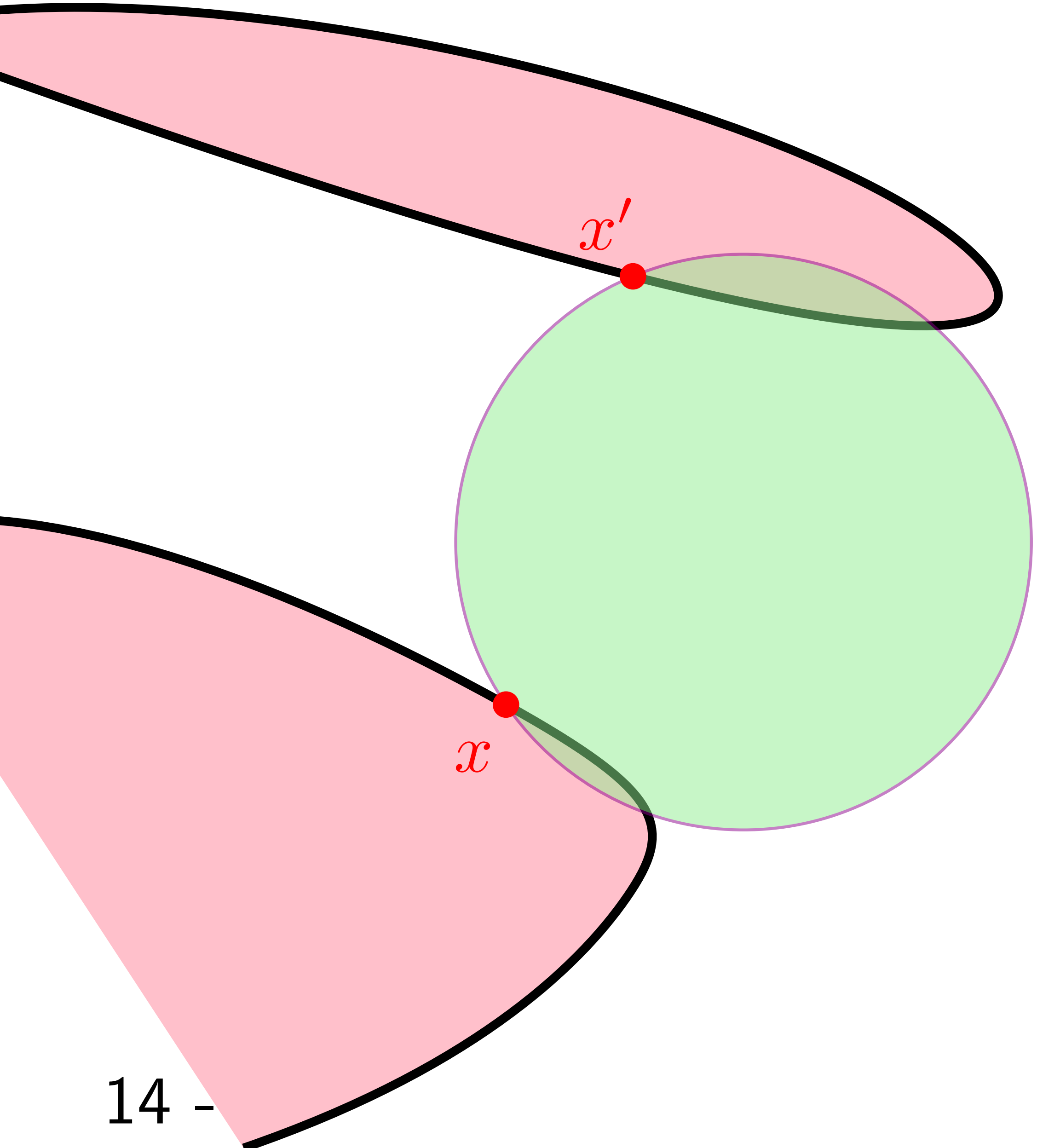
Theorem: 0.25 sample ⇒ crust ⊂ wanted result



Reconstruction

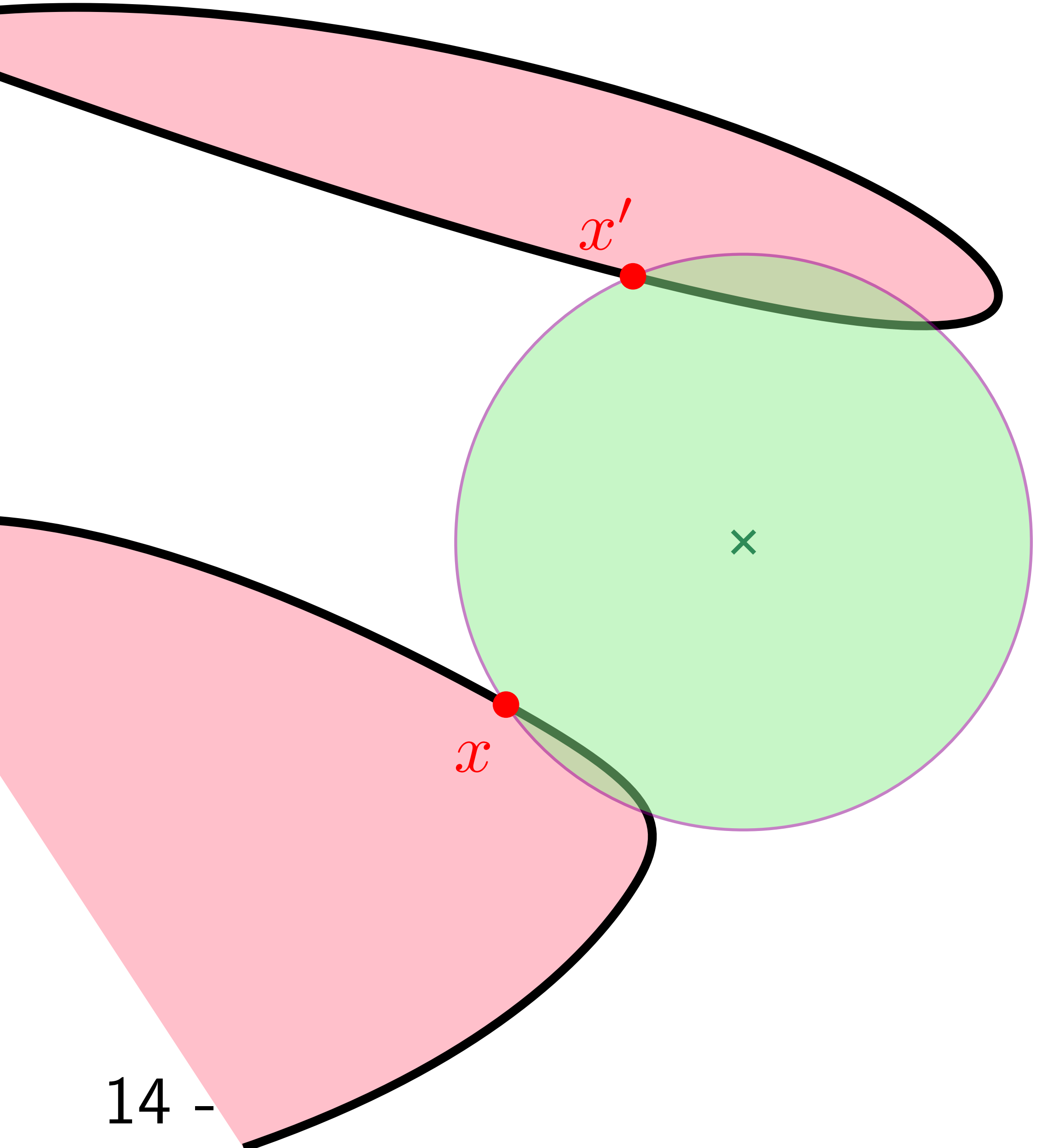
Theorem: 0.25 sample ⇒ crust ⊂ wanted result

Assume empty circle



Reconstruction

Theorem: 0.25 sample ⇒ crust ⊂ wanted result



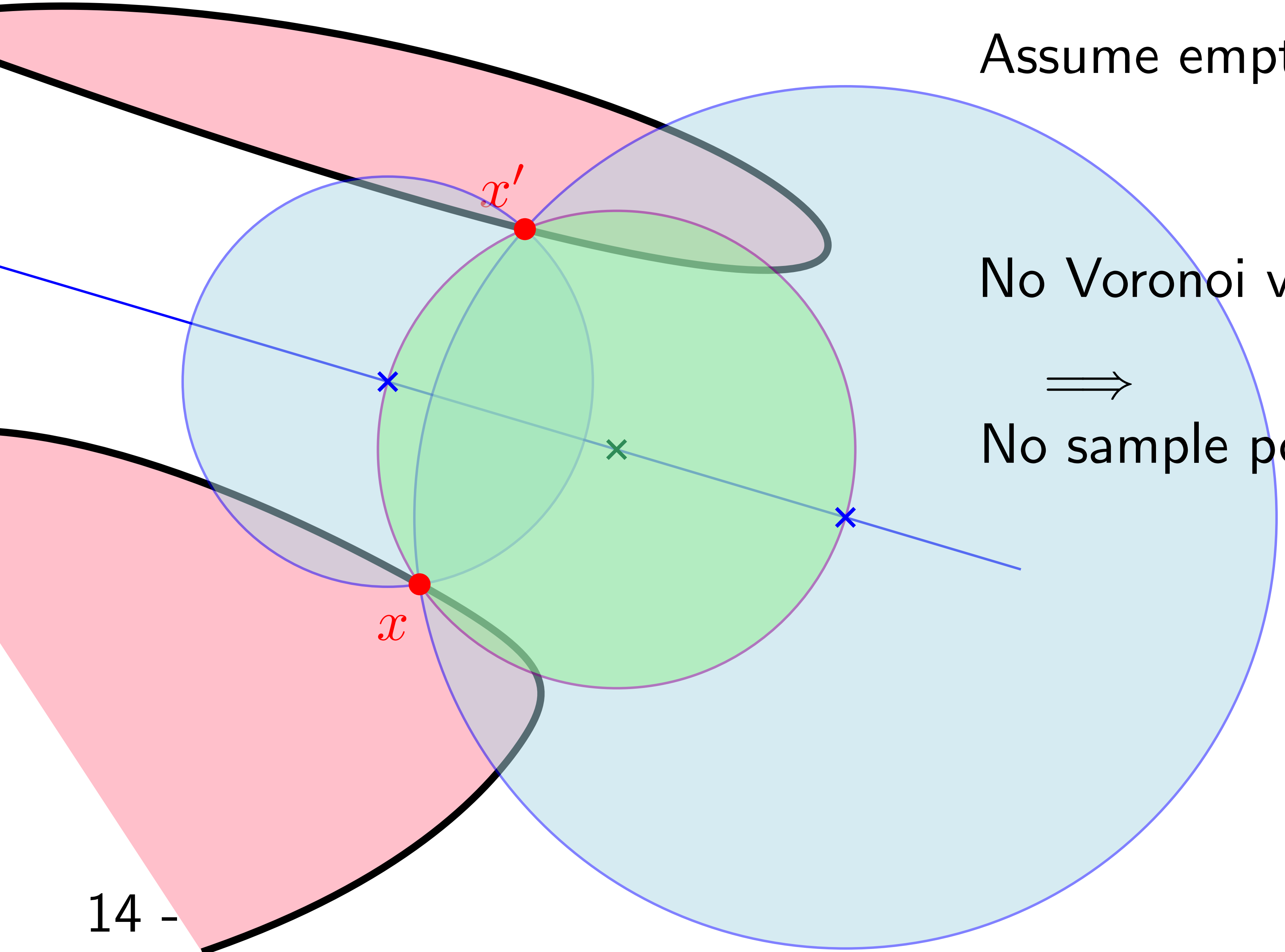
Assume empty circle

No Voronoi vertices there

$\Longrightarrow$

Reconstruction

Theorem: 0.25 sample ⇒ crust ⊂ wanted result



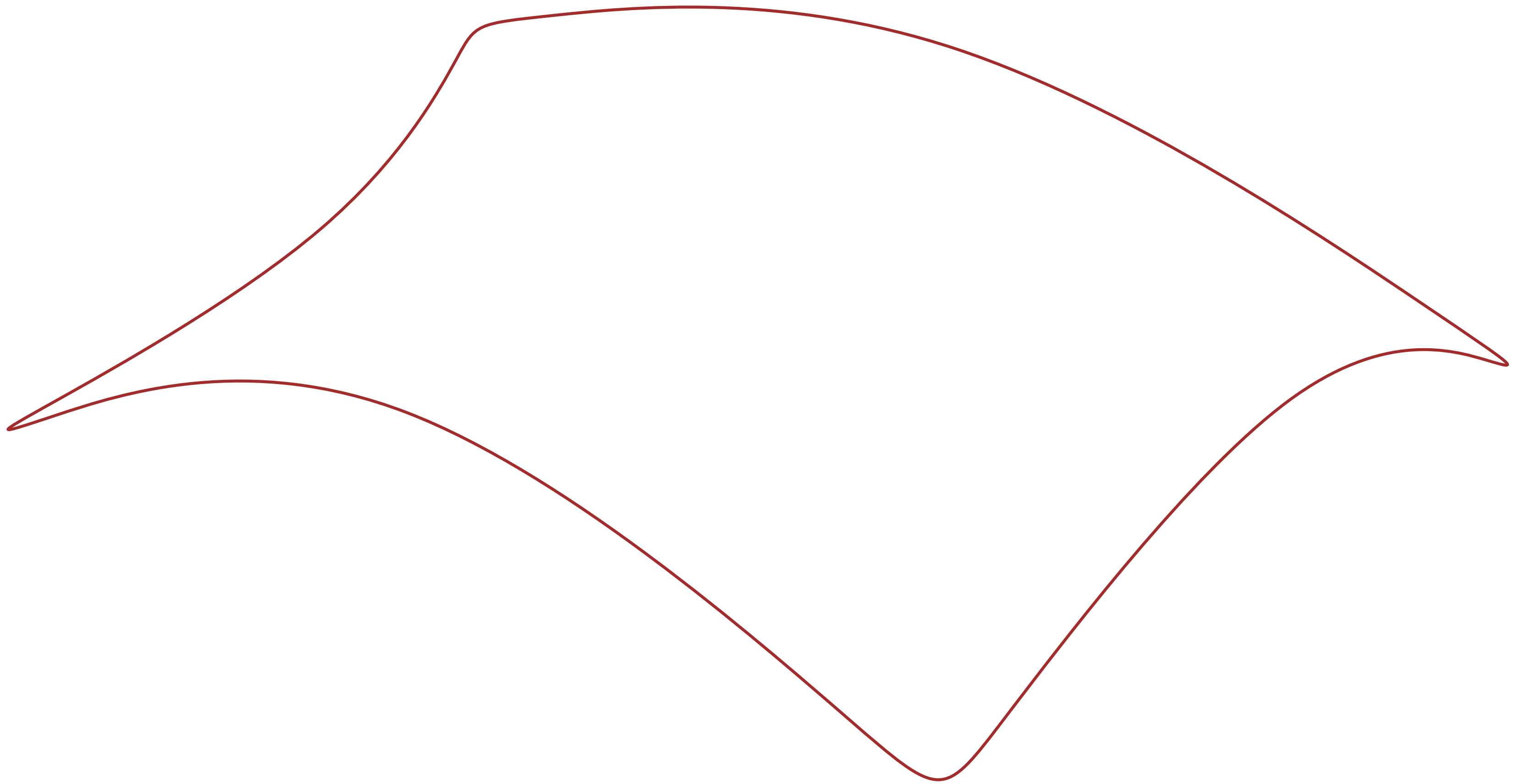
Assume empty circle

No Voronoi vertices there

$\Longrightarrow$

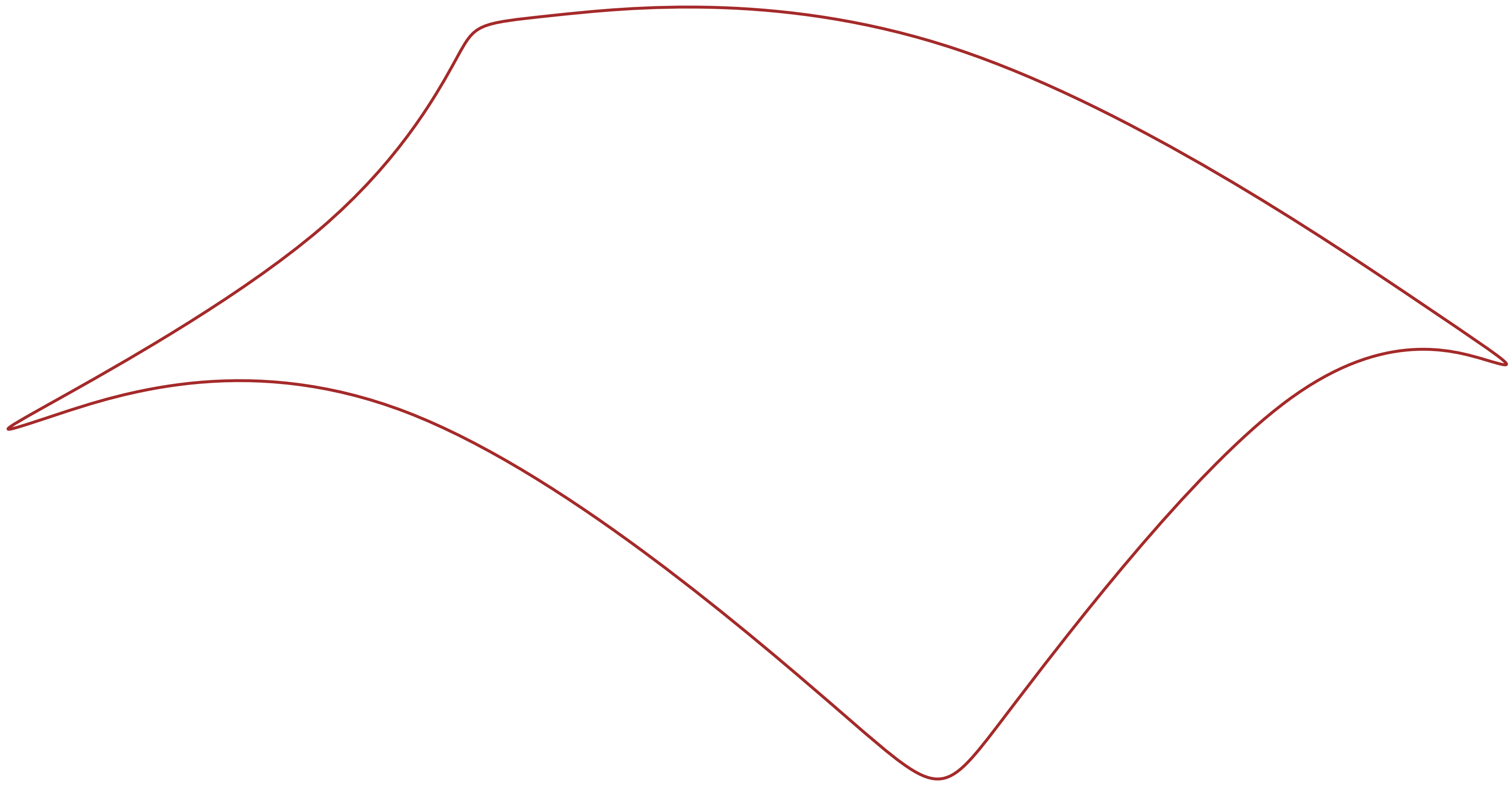
No sample points there

Reconstruction 3D



Reconstruction 3D

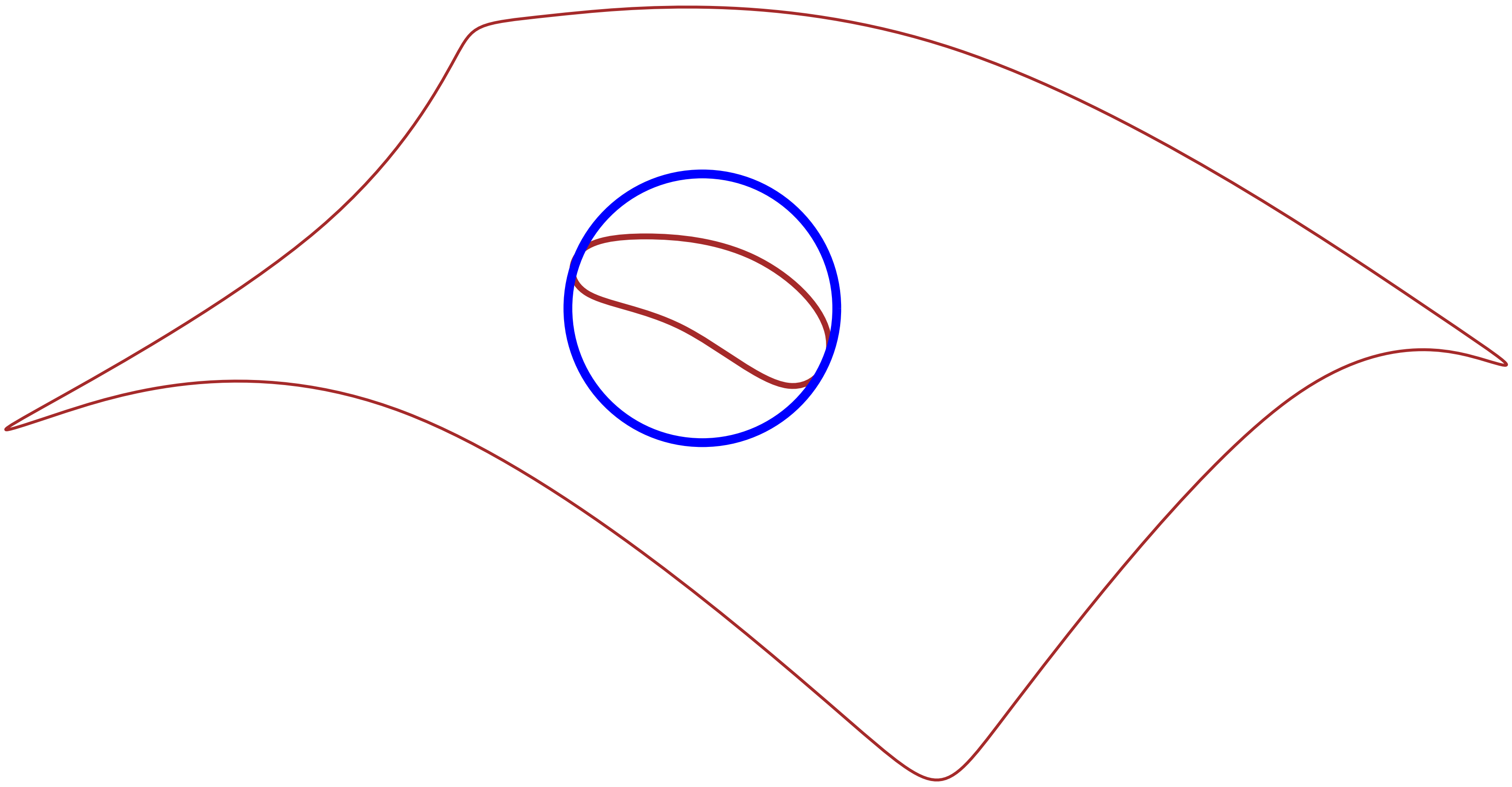
Difficulty: sliver



Reconstruction 3D

Difficulty: sliver

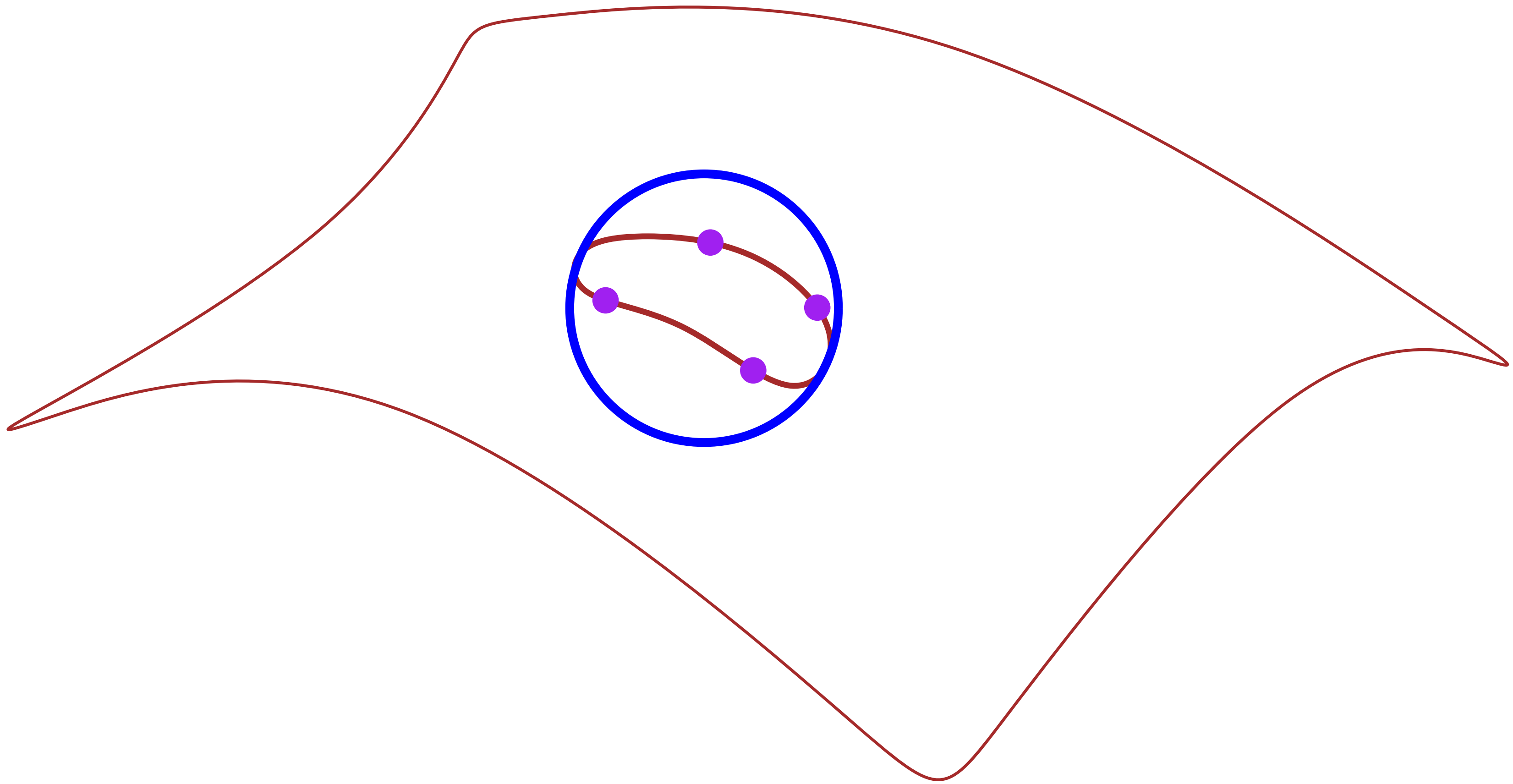
small sphere



Reconstruction 3D

Difficulty: sliver

small sphere four sample points

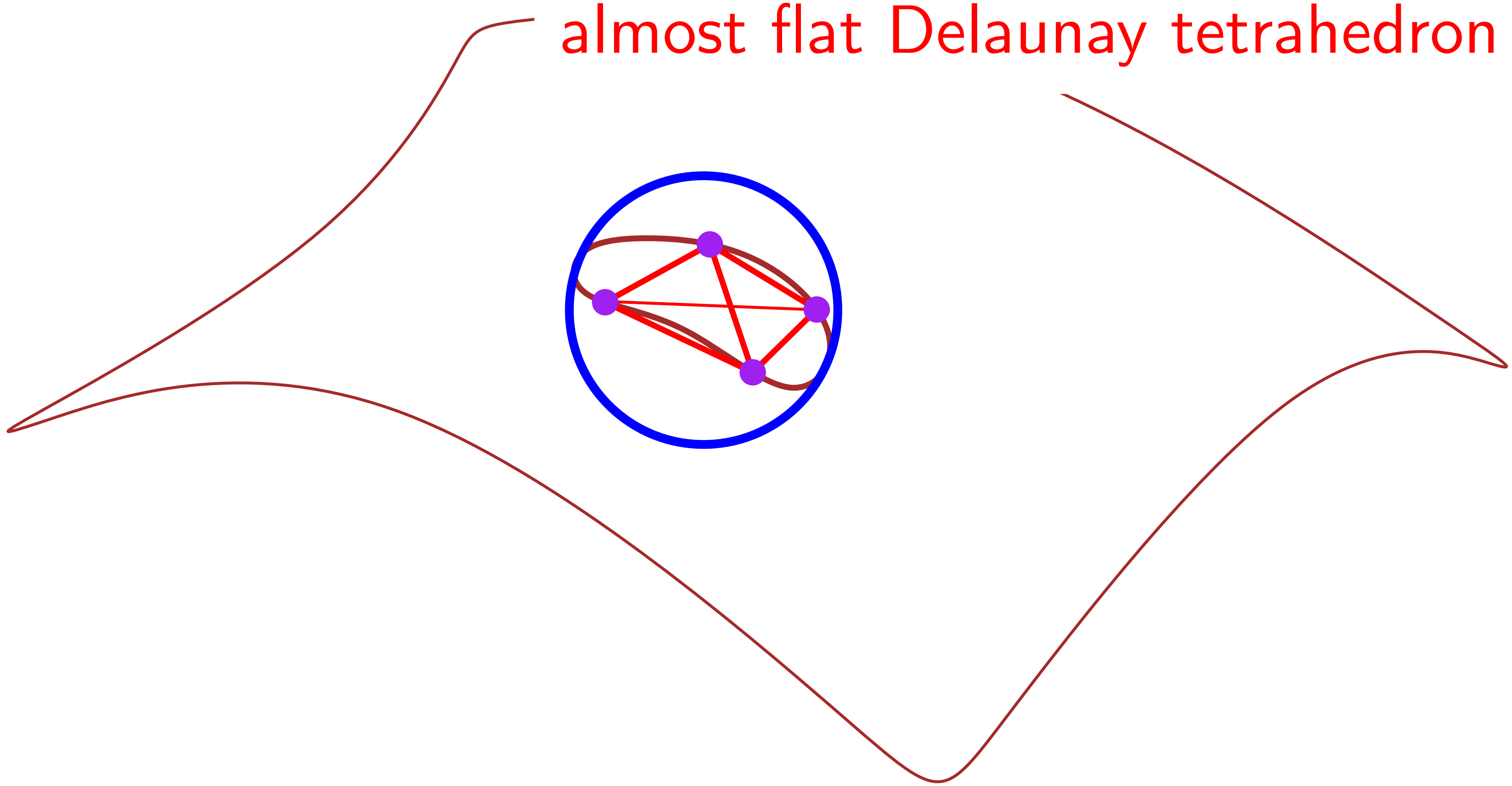


Reconstruction 3D

Difficulty: sliver

small sphere four sample points

almost flat Delaunay tetrahedron

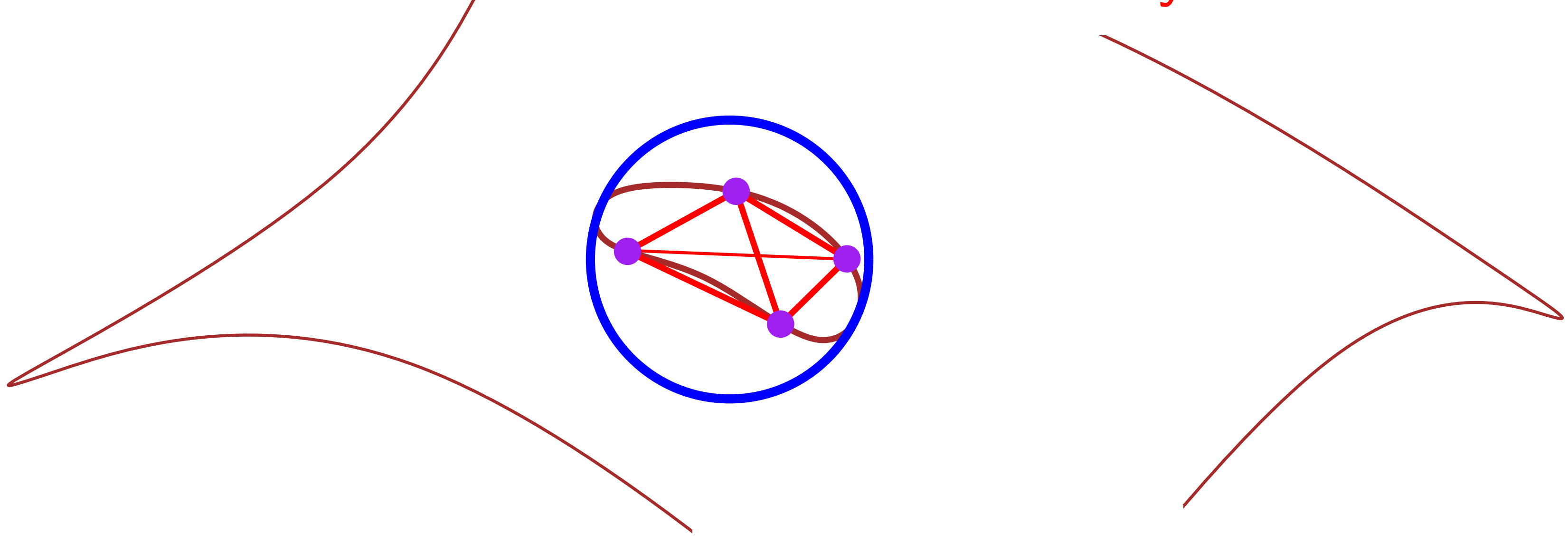


Reconstruction 3D

Difficulty: sliver

small sphere four sample points

almost flat Delaunay tetrahedron



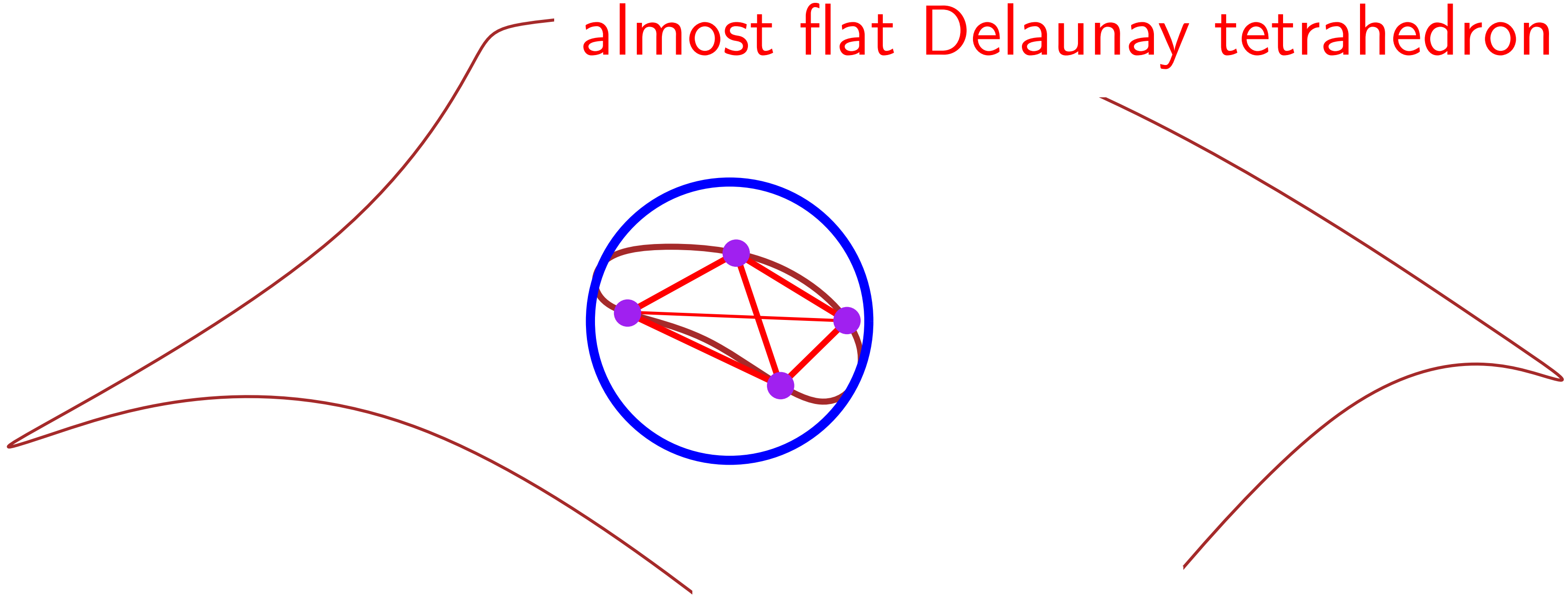
Which triangle belongs to reconstruction ?

Reconstruction 3D

Difficulty: sliver

small sphere four sample points

almost flat Delaunay tetrahedron

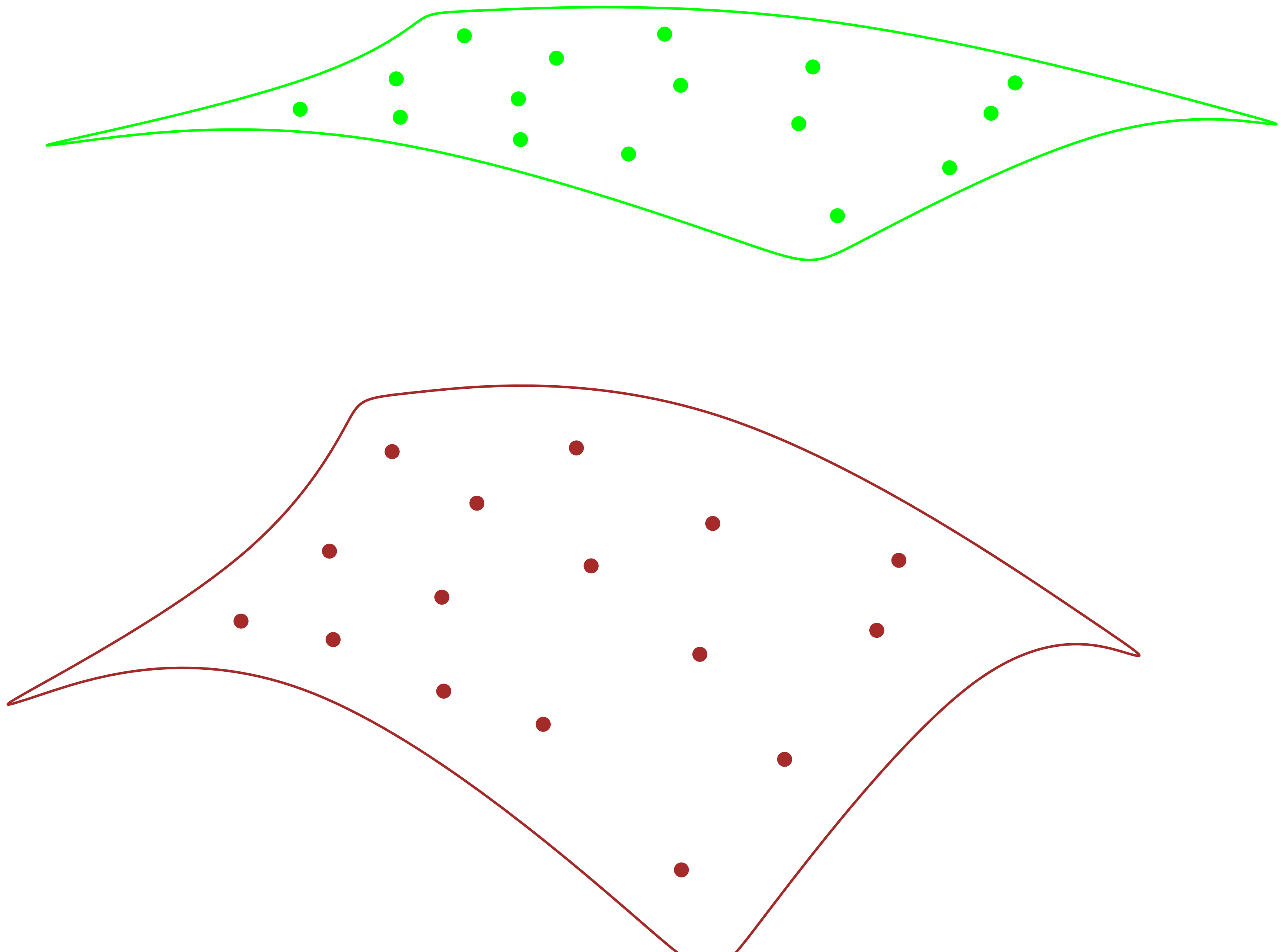


Which triangle belongs to reconstruction ?

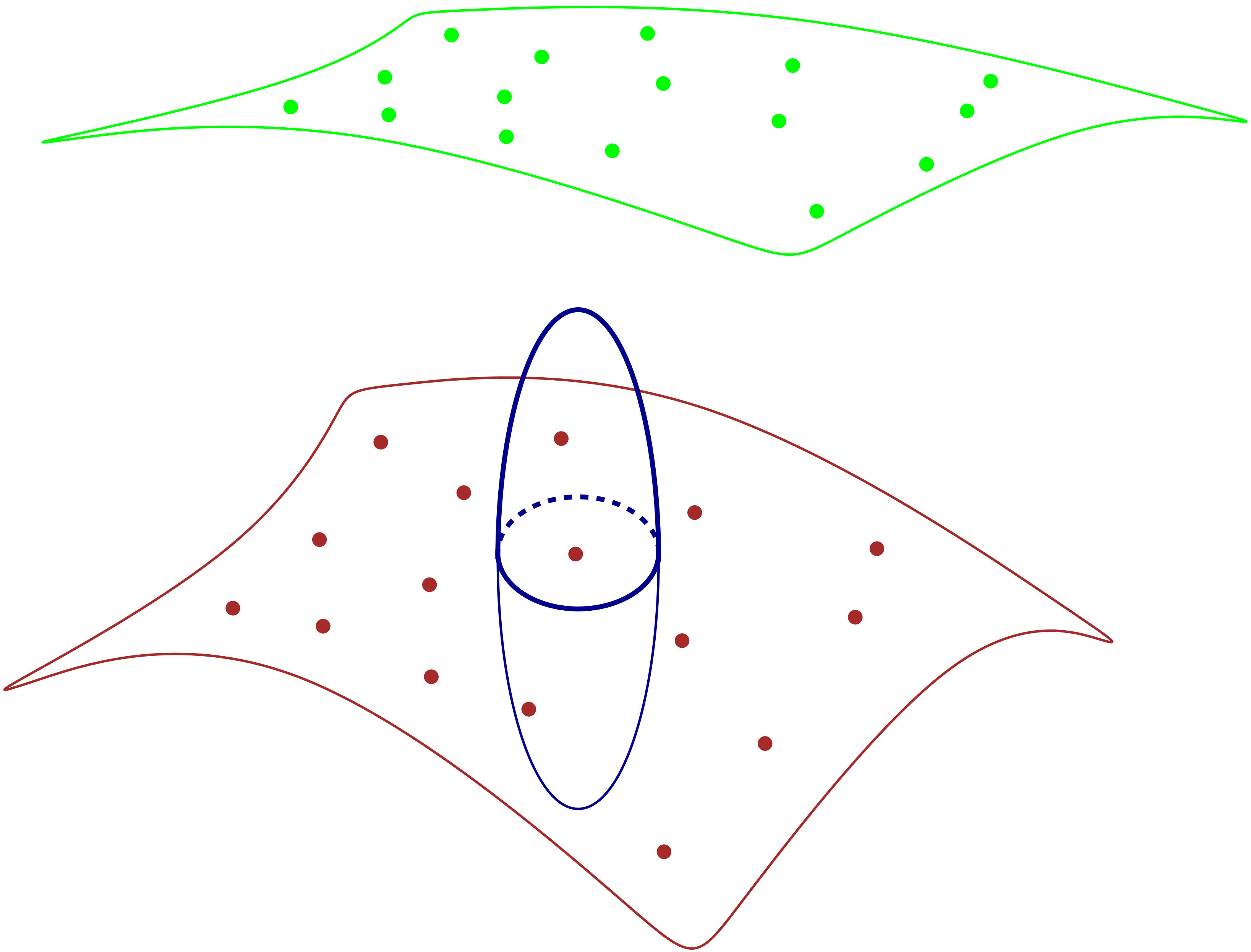
Crust: Voronoi vertices may kill useful triangles

Reconstruction

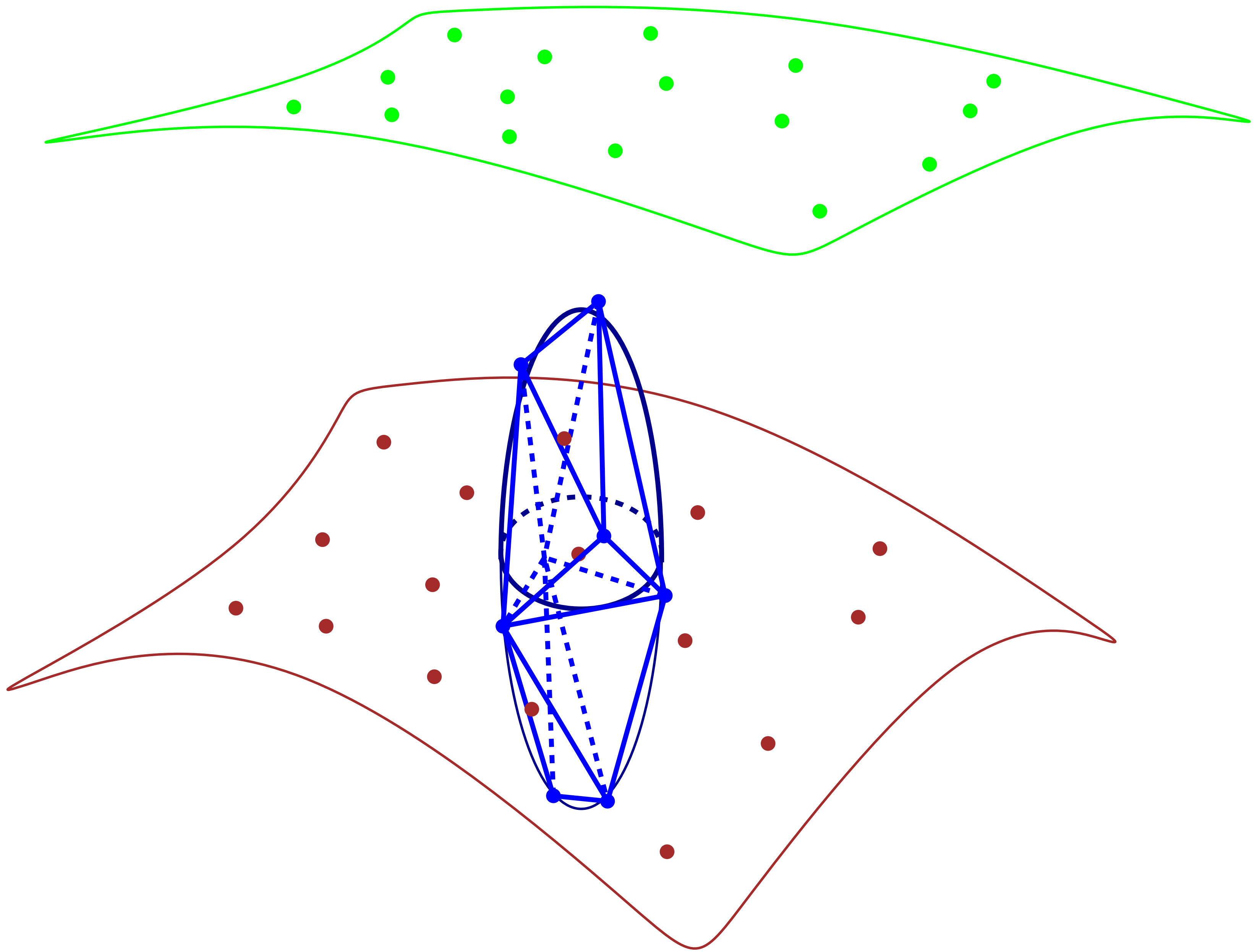
3D



Reconstruction 3D

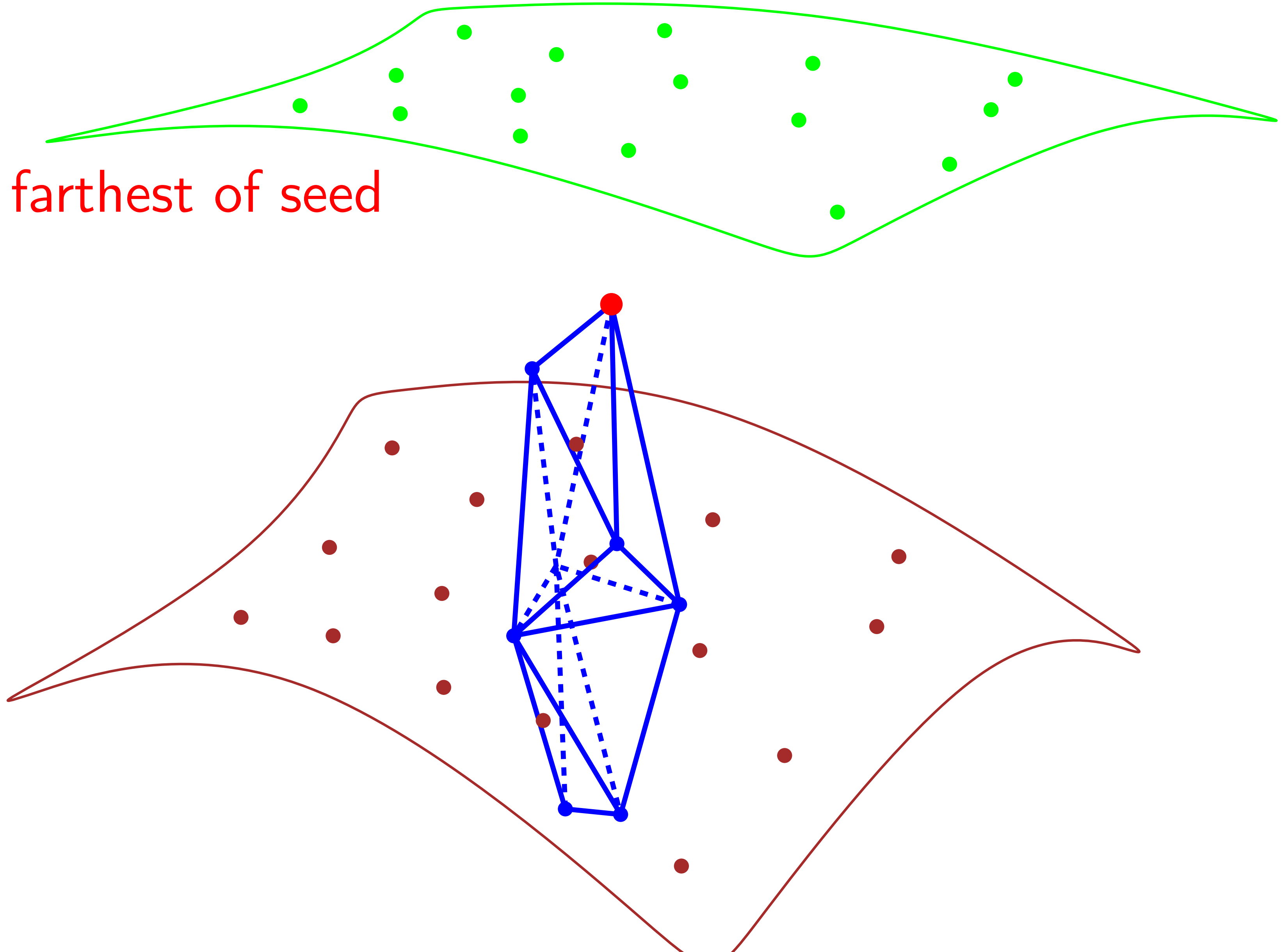


Reconstruction 3D



Reconstruction

3D

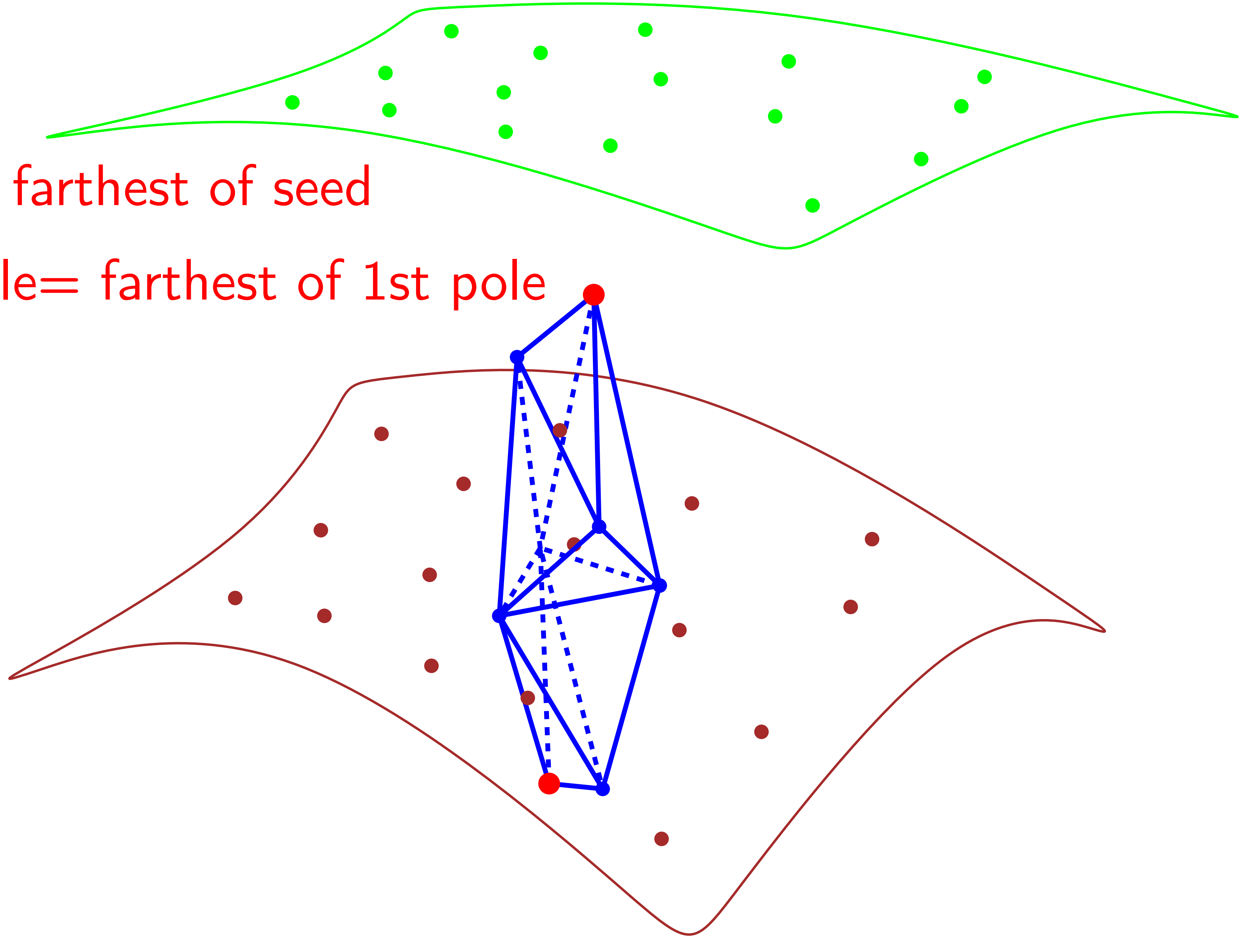


Pole = farthest of seed

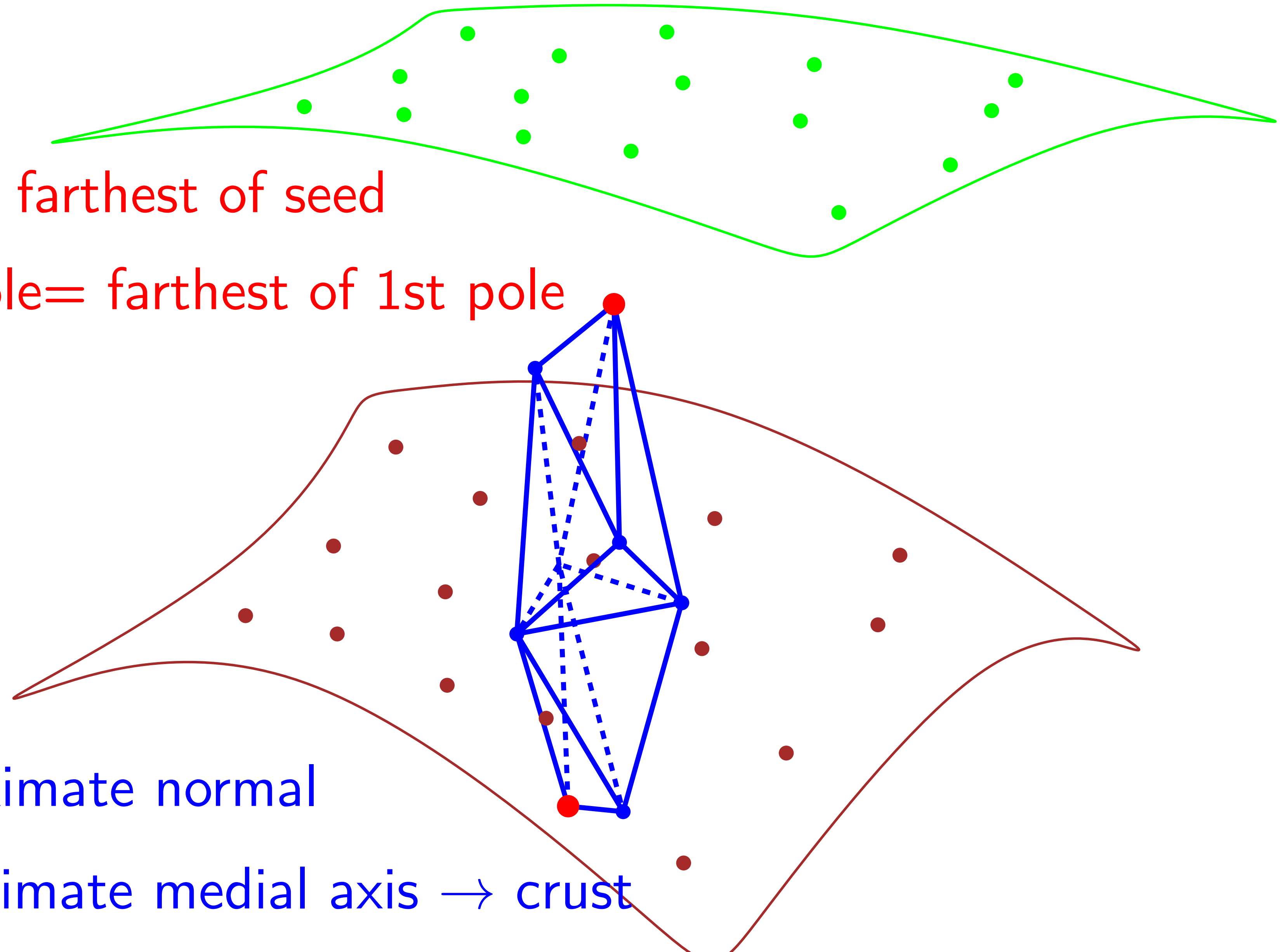
Reconstruction 3D

Pole = farthest of seed

2nd pole= farthest of 1st pole



Reconstruction 3D



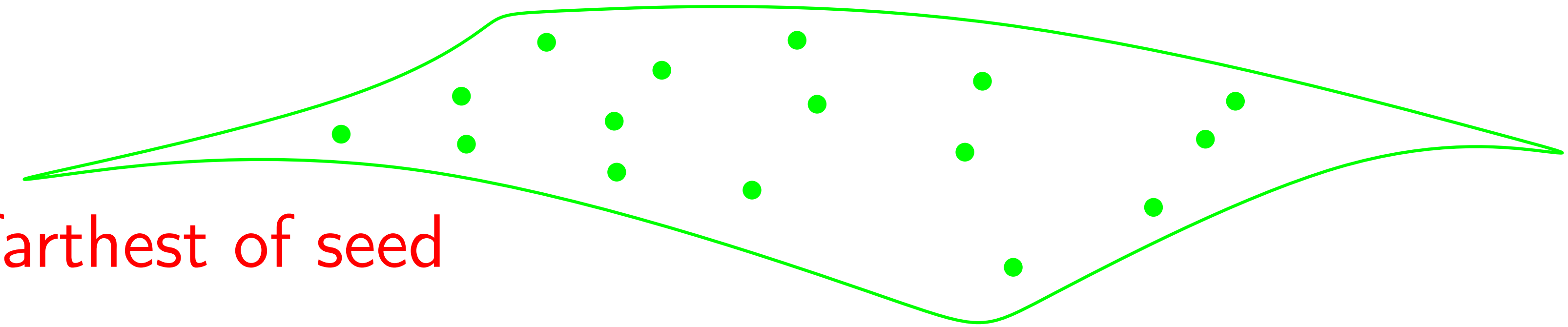
Pole = farthest of seed

2nd pole= farthest of 1st pole

Approximate normal

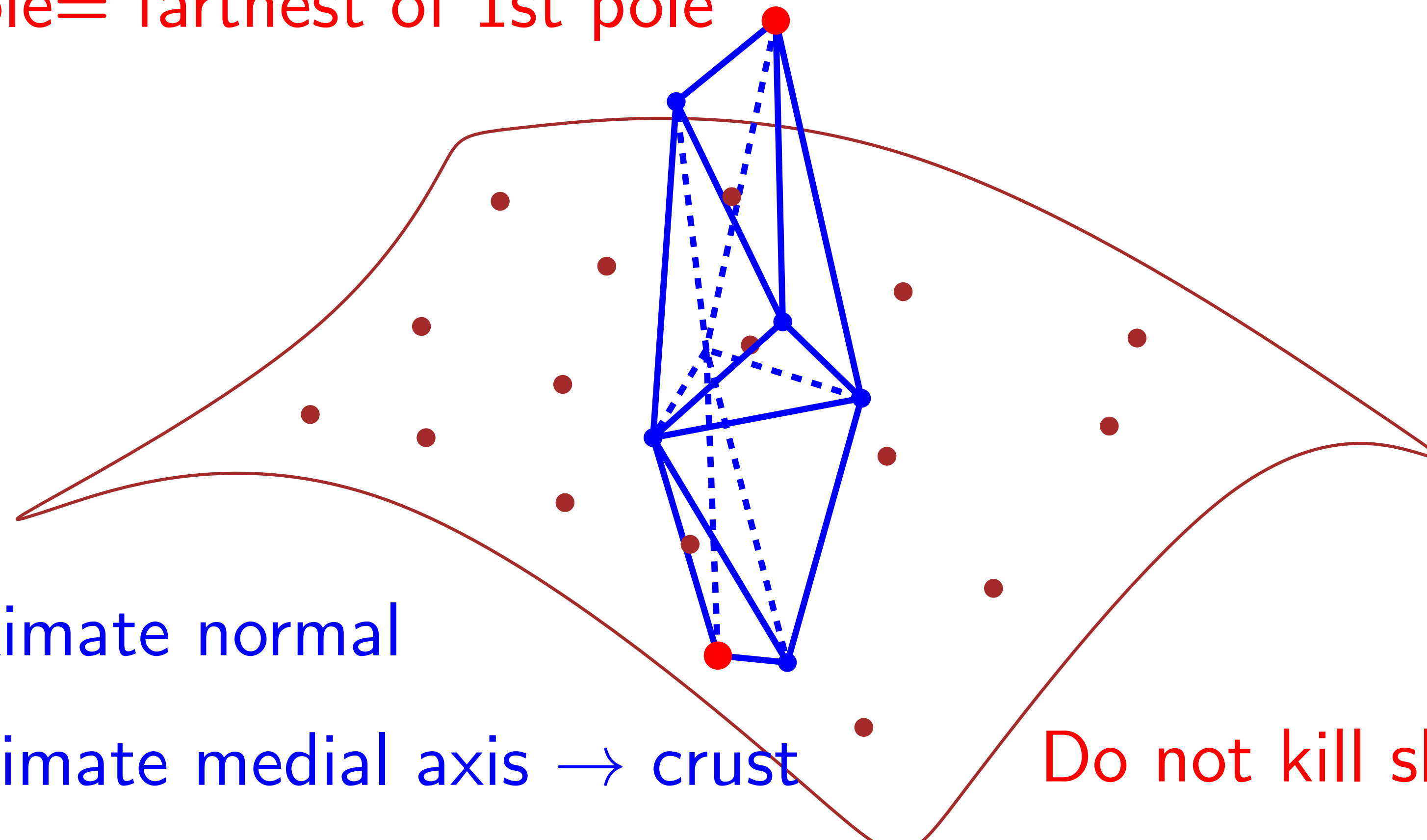
Approximate medial axis → crust

Reconstruction 3D



Pole = farthest of seed

2nd pole= farthest of 1st pole



Approximate normal

Approximate medial axis → crust

Do not kill slivers

Meshing

Meshing

Discretize space to solve (differential) equations

- Finite elements
- Finite differences

Meshing

Discretize space to solve (differential) equations

- Finite elements
- Finite differences

Good mesh:

- Control shape of elements (no small angles)
- Control size of elements (adjust to function variability)
- Minimize number of elements

Meshing

Gallery

Structured meshes (advancing front, deformation)

Delaunay mesh refinement

- [Ruppert]
- protecting small angles
- off-centers

Delaunay mesh optimization

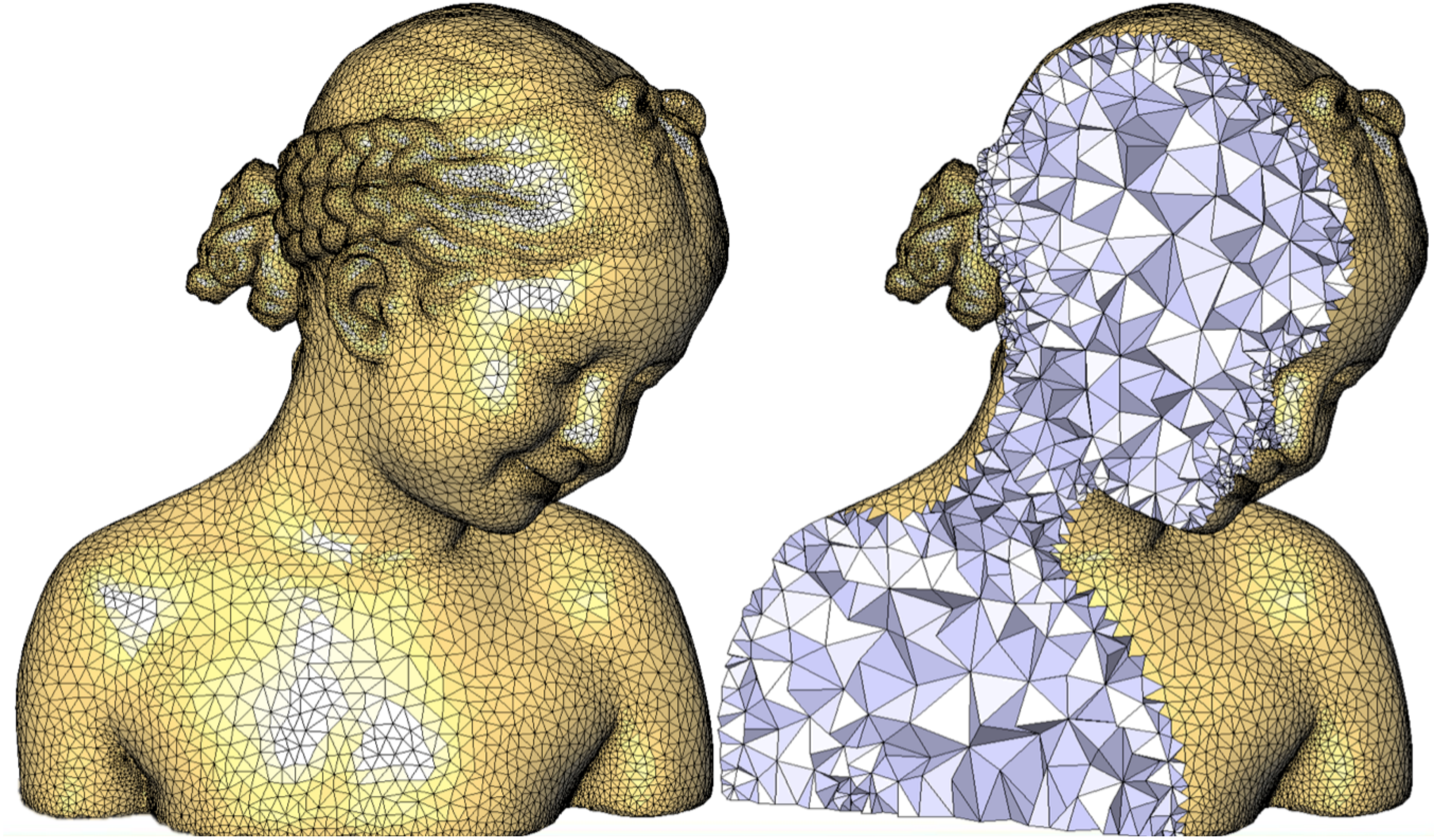
3D

Meshing

Gallery

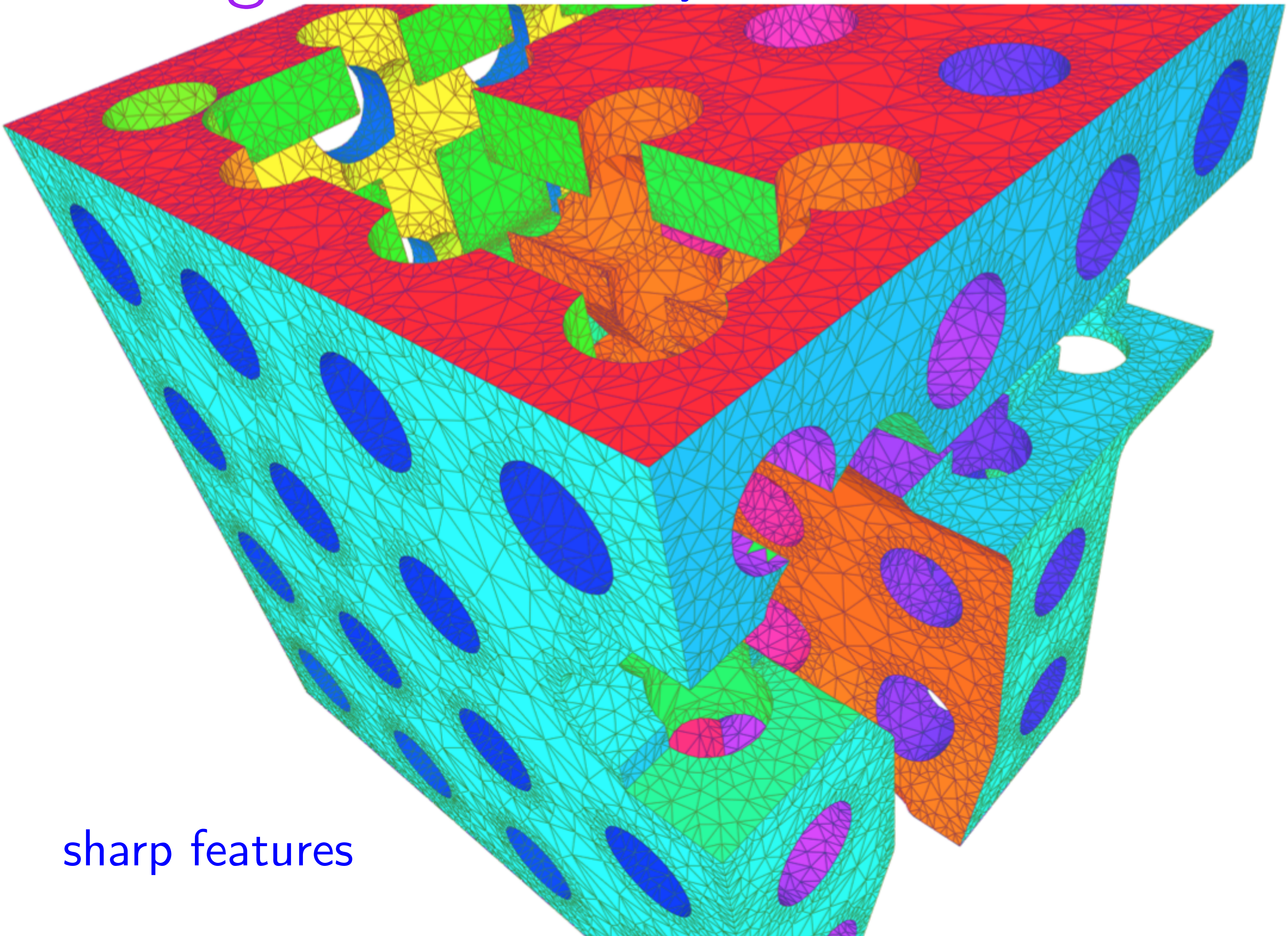
Meshing

Gallery



Meshing

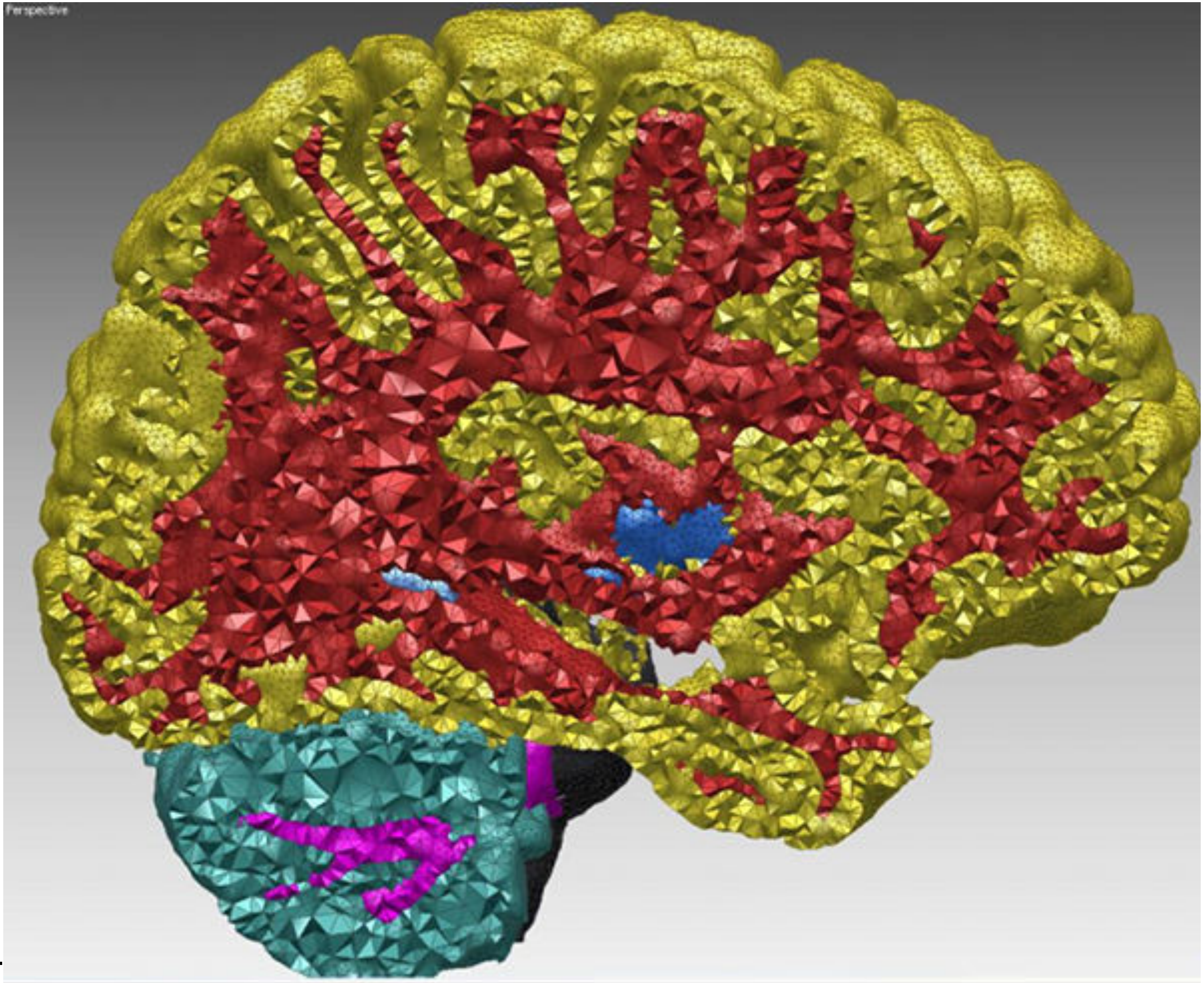
Gallery



sharp features

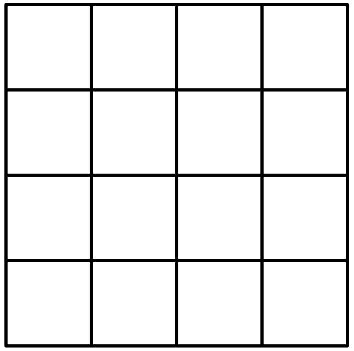
Meshing

Gallery



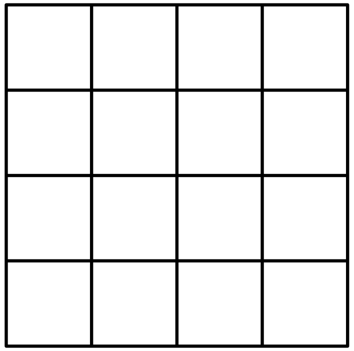
Meshing

Regular grid

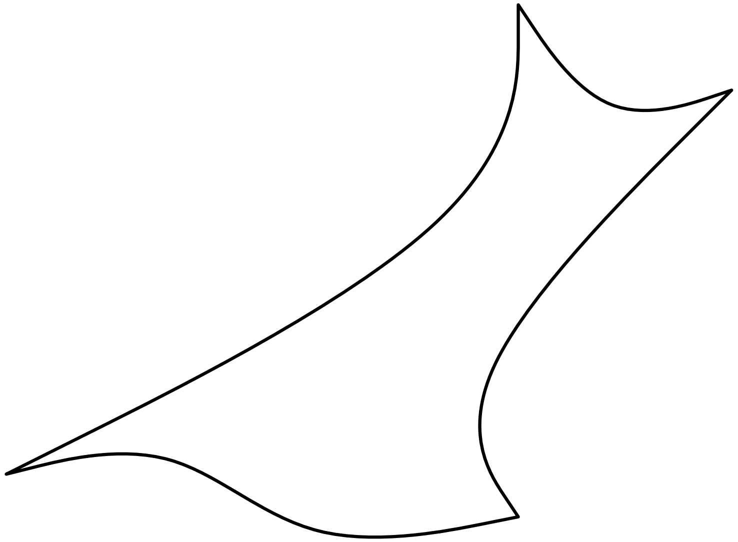


Meshing

Regular grid

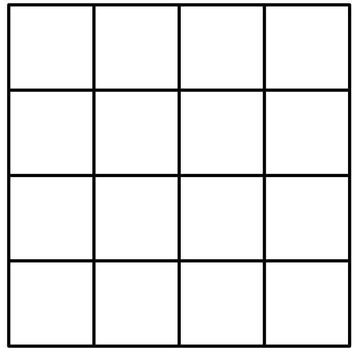


Shape

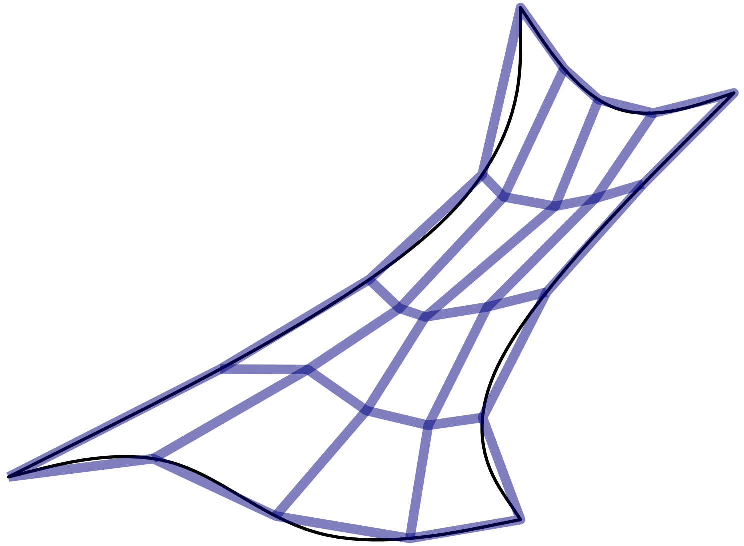


Meshing

Regular grid



Shape



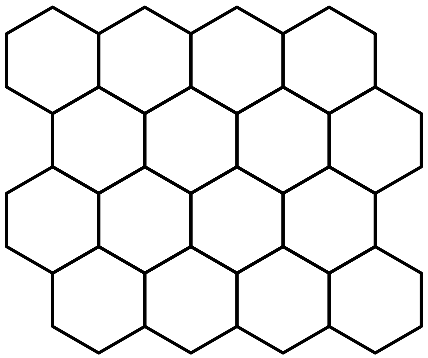
Deform

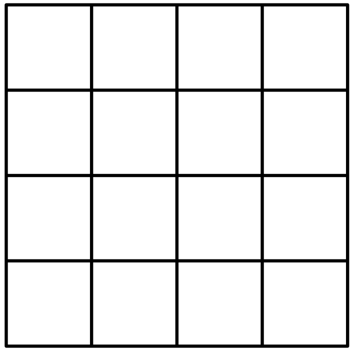
to fit the grid in the shape

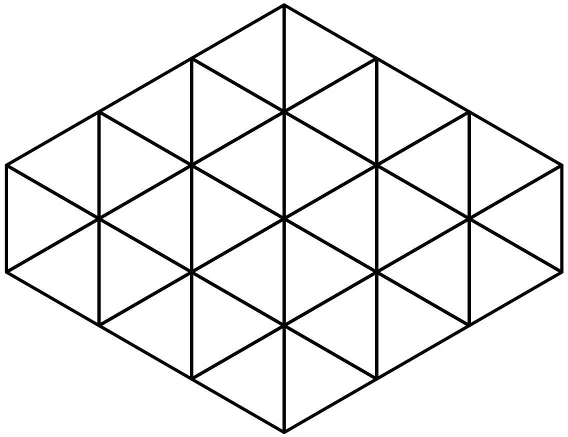
Meshing

Structured meshes

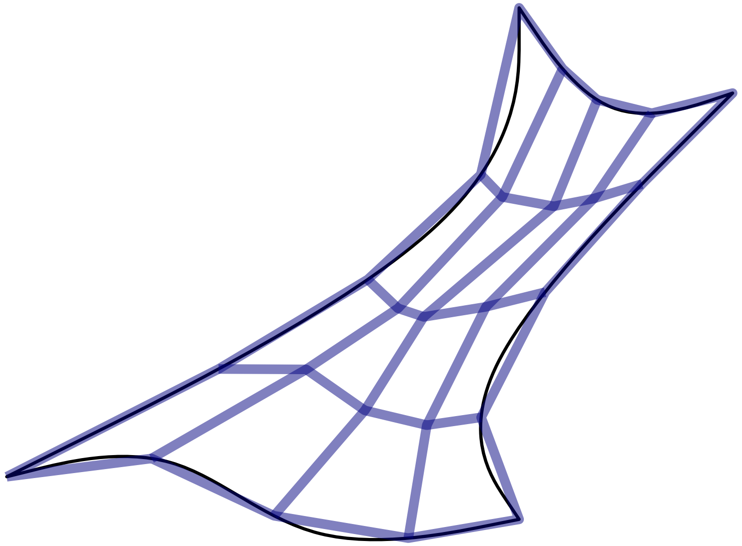
Regular grid







Shape

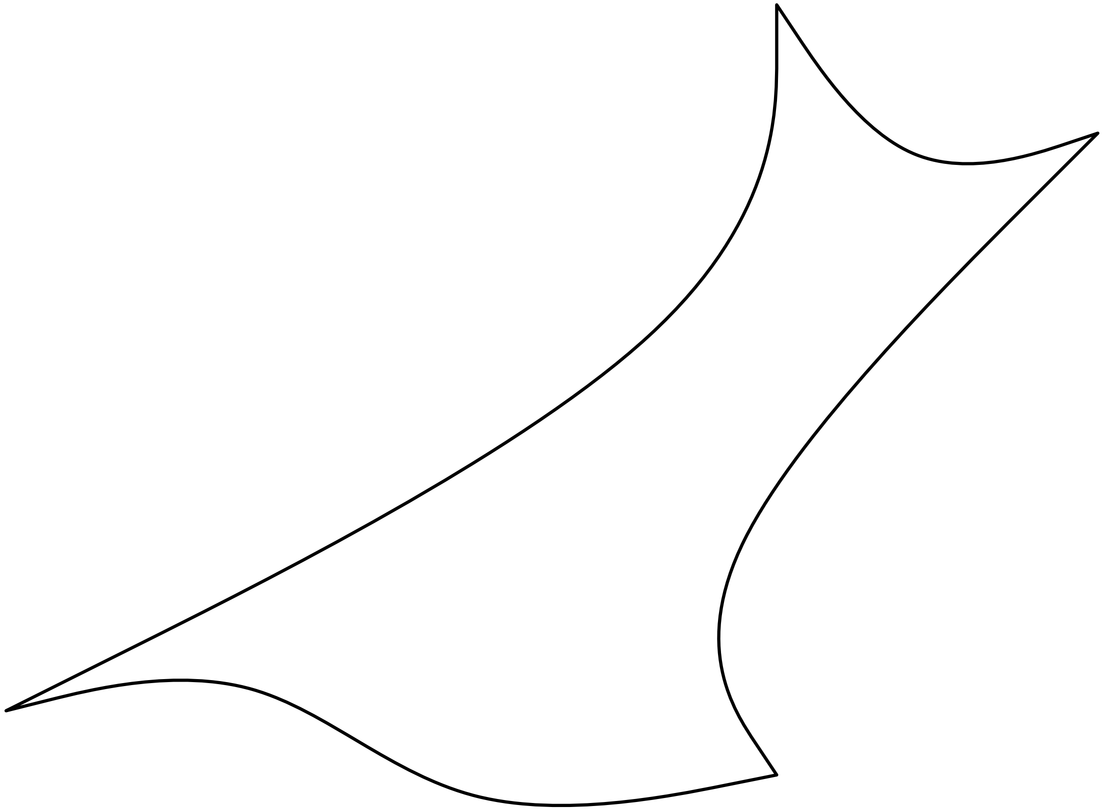


Deform

to fit the grid in the shape

Meshing

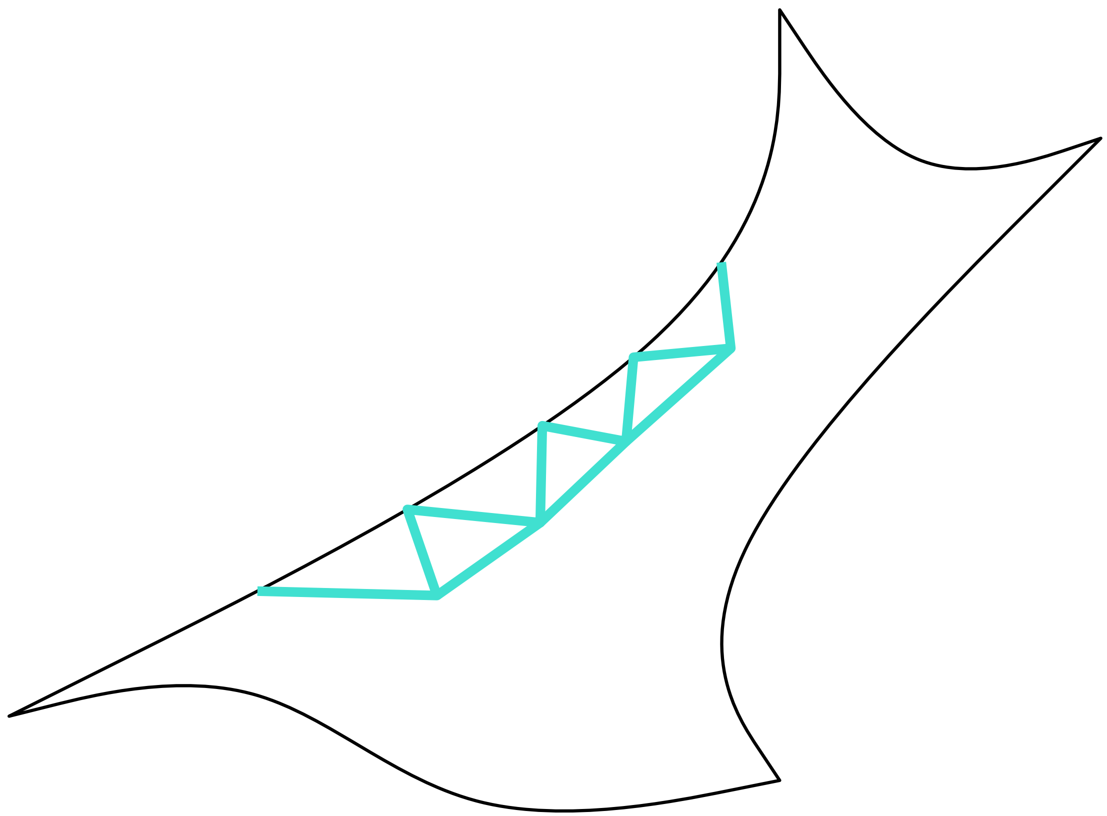
Shape



Meshing

Shape

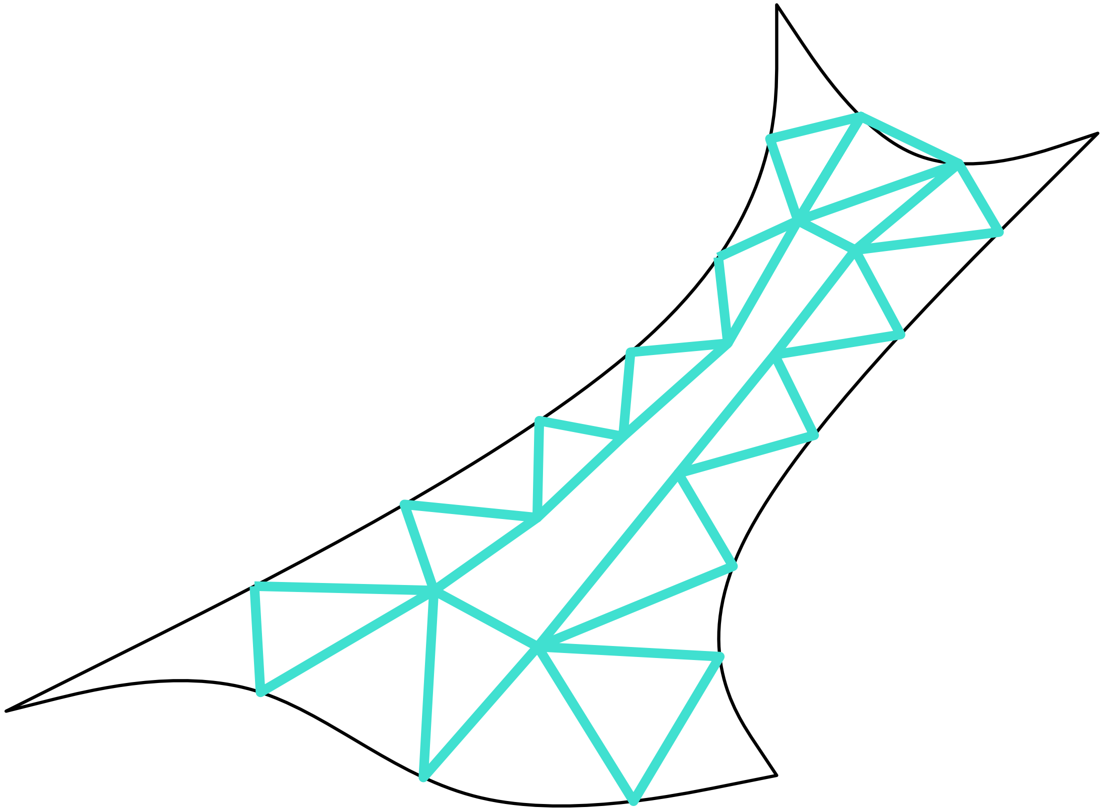
Advancing front



Meshing

Shape

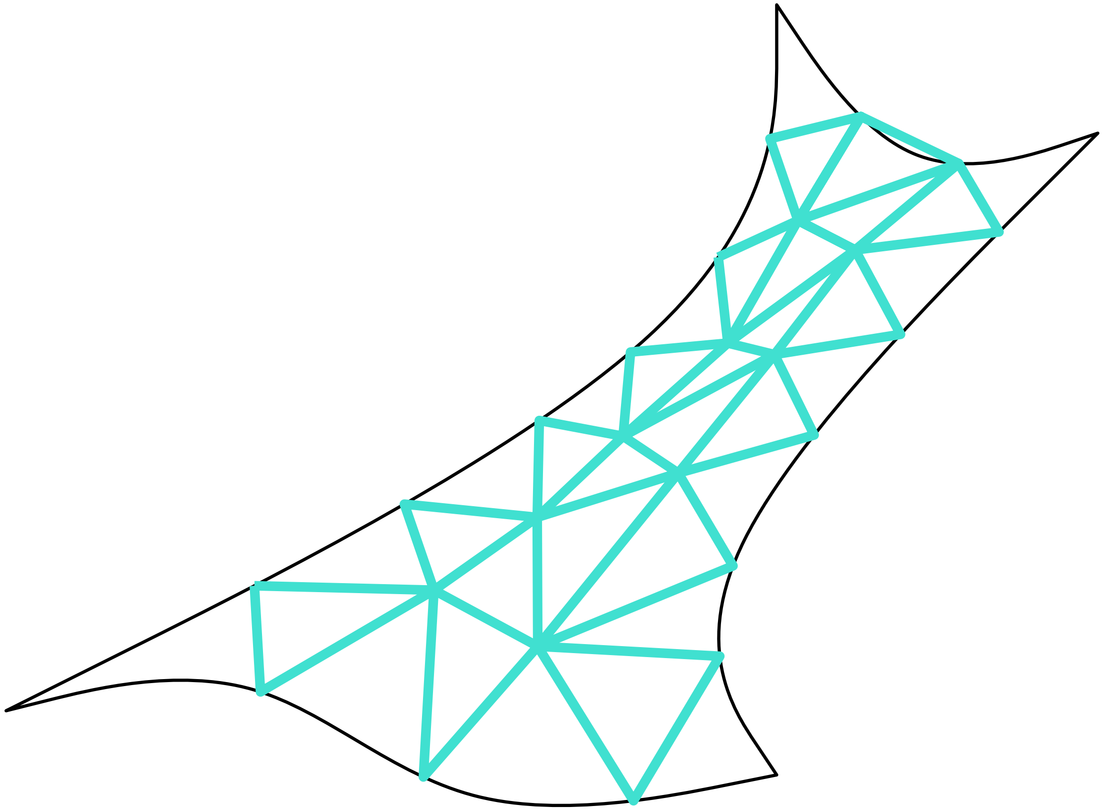
Advancing front



Meshing

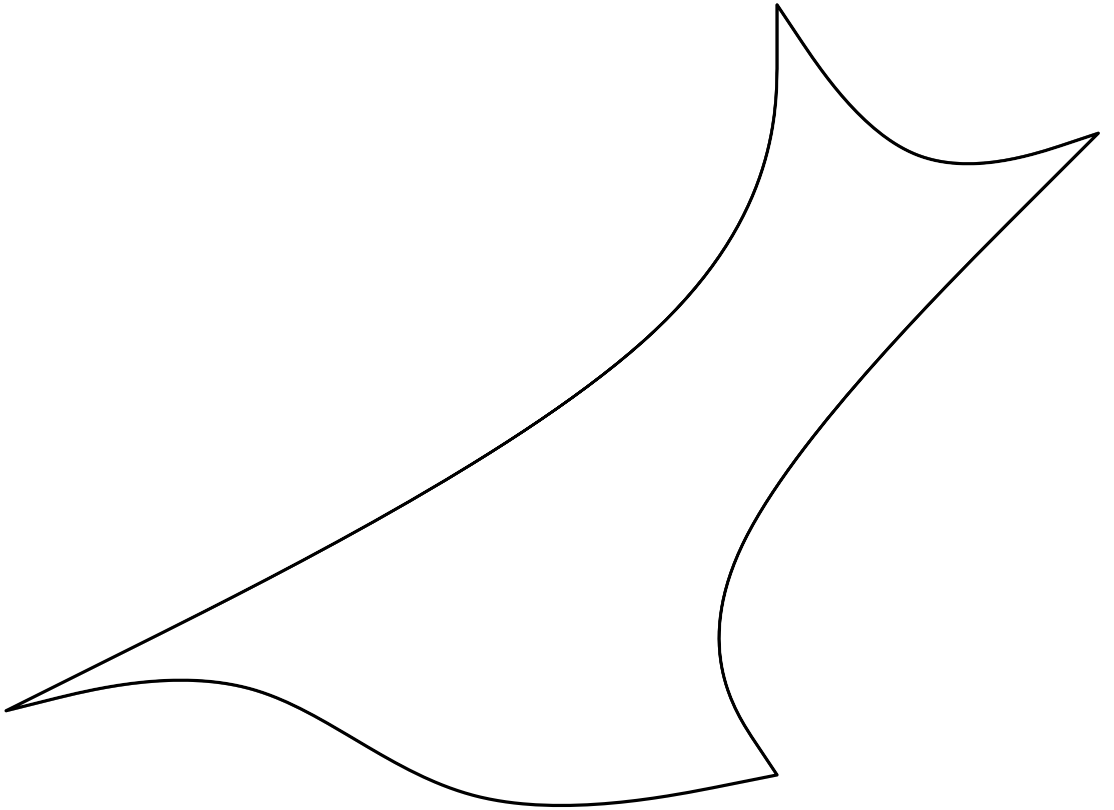
Shape

Advancing front



Meshing

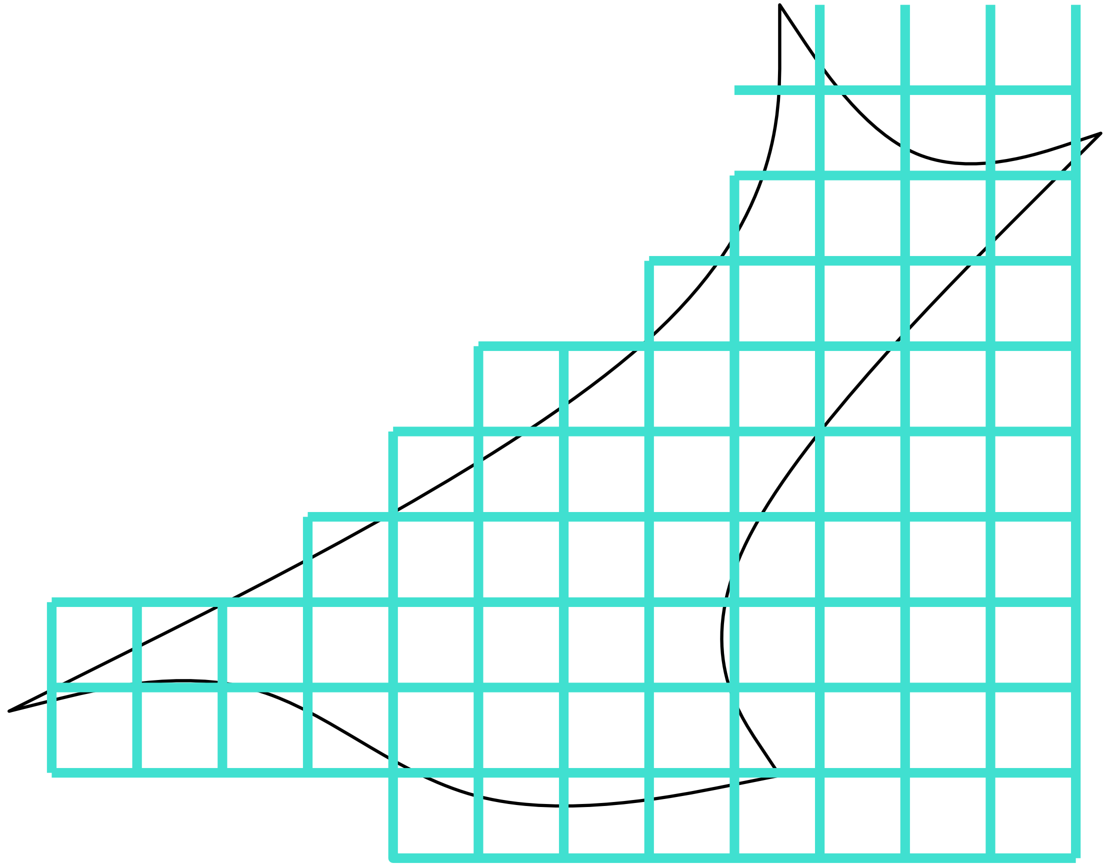
Shape



Meshing

Shape

Add grid

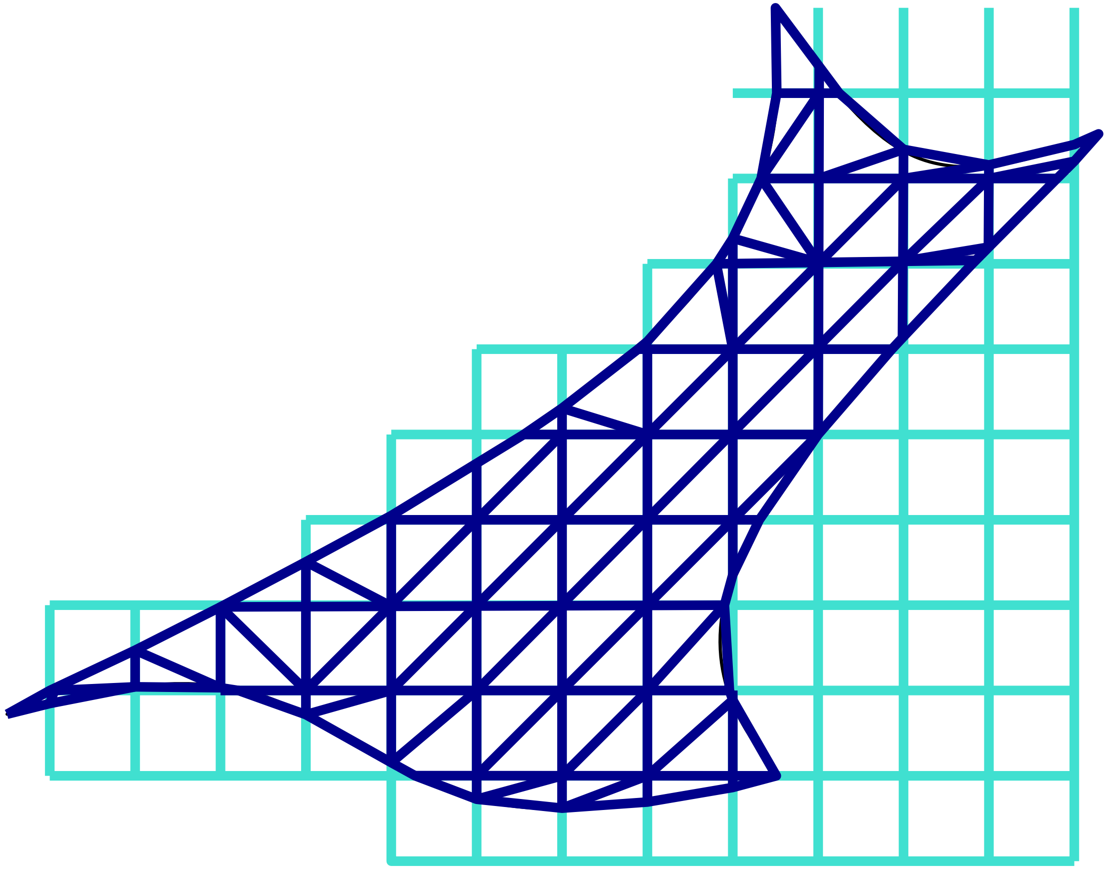


Meshing

Shape

Add grid

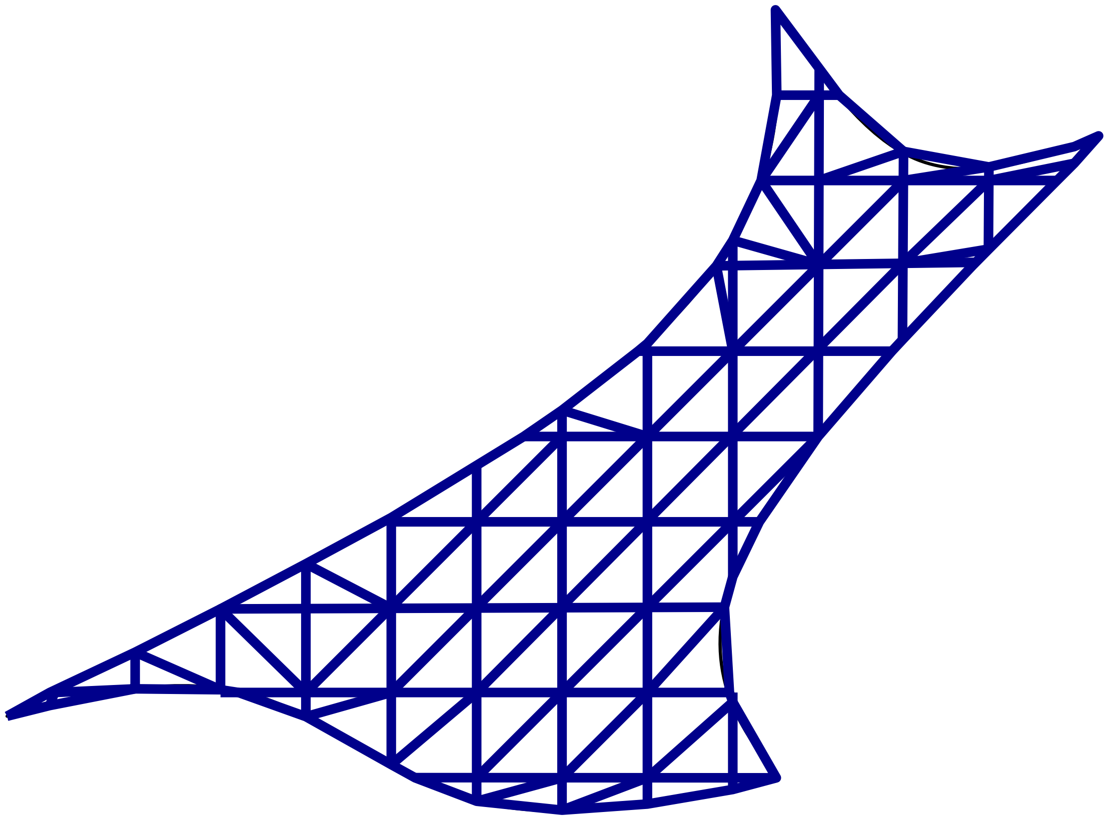
Triangulate



Meshing

Structured meshes

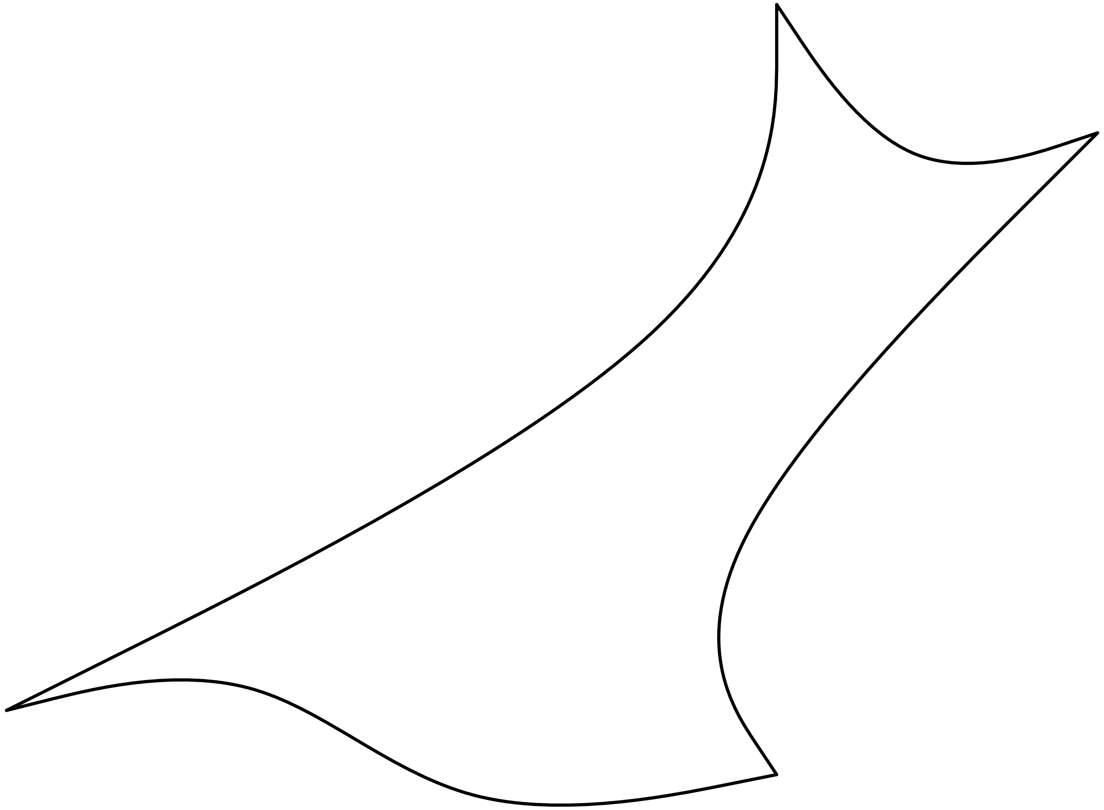
Shape



Triangulate

Uniform mesh

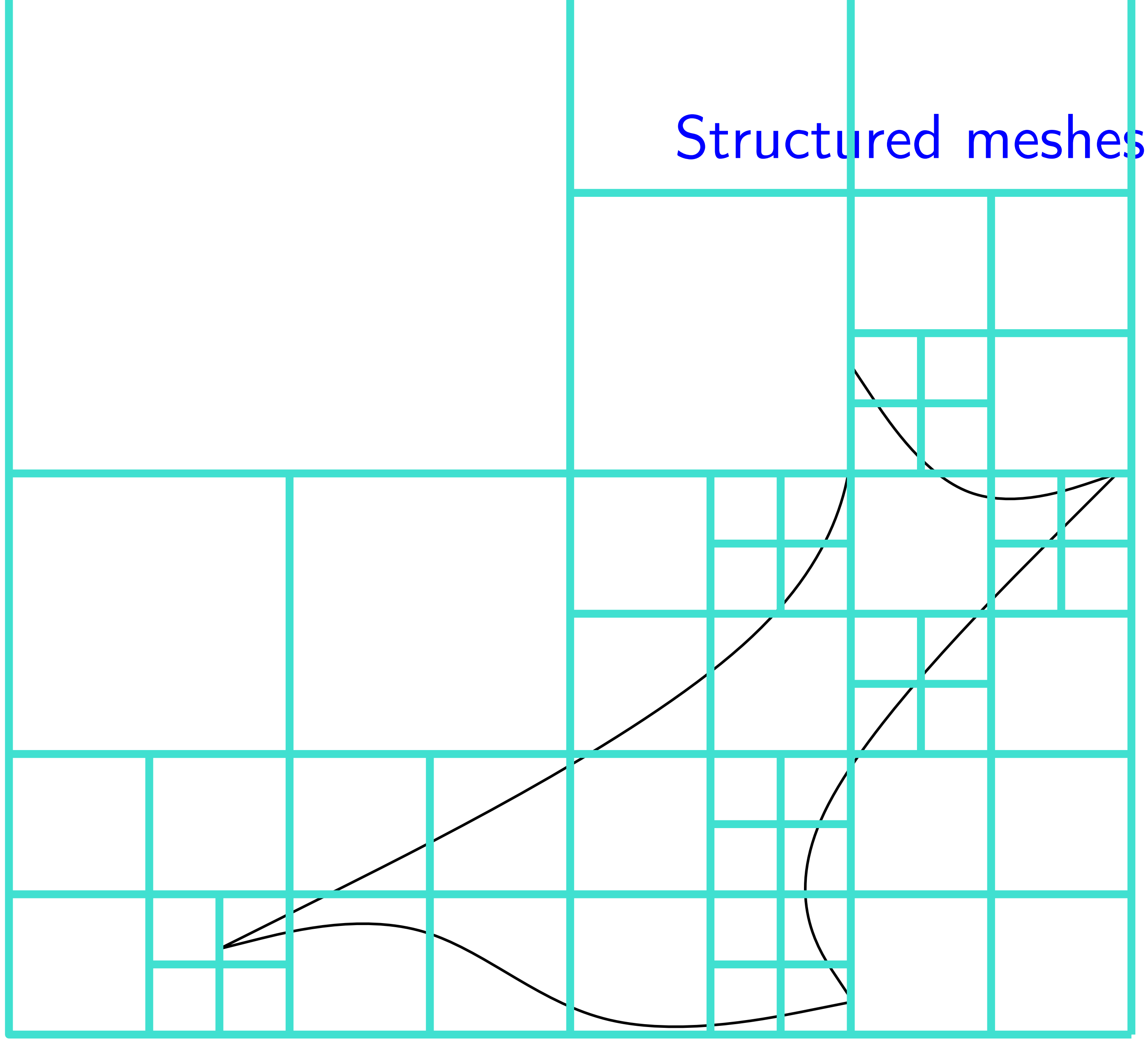
Shape



Meshing

Structured meshes

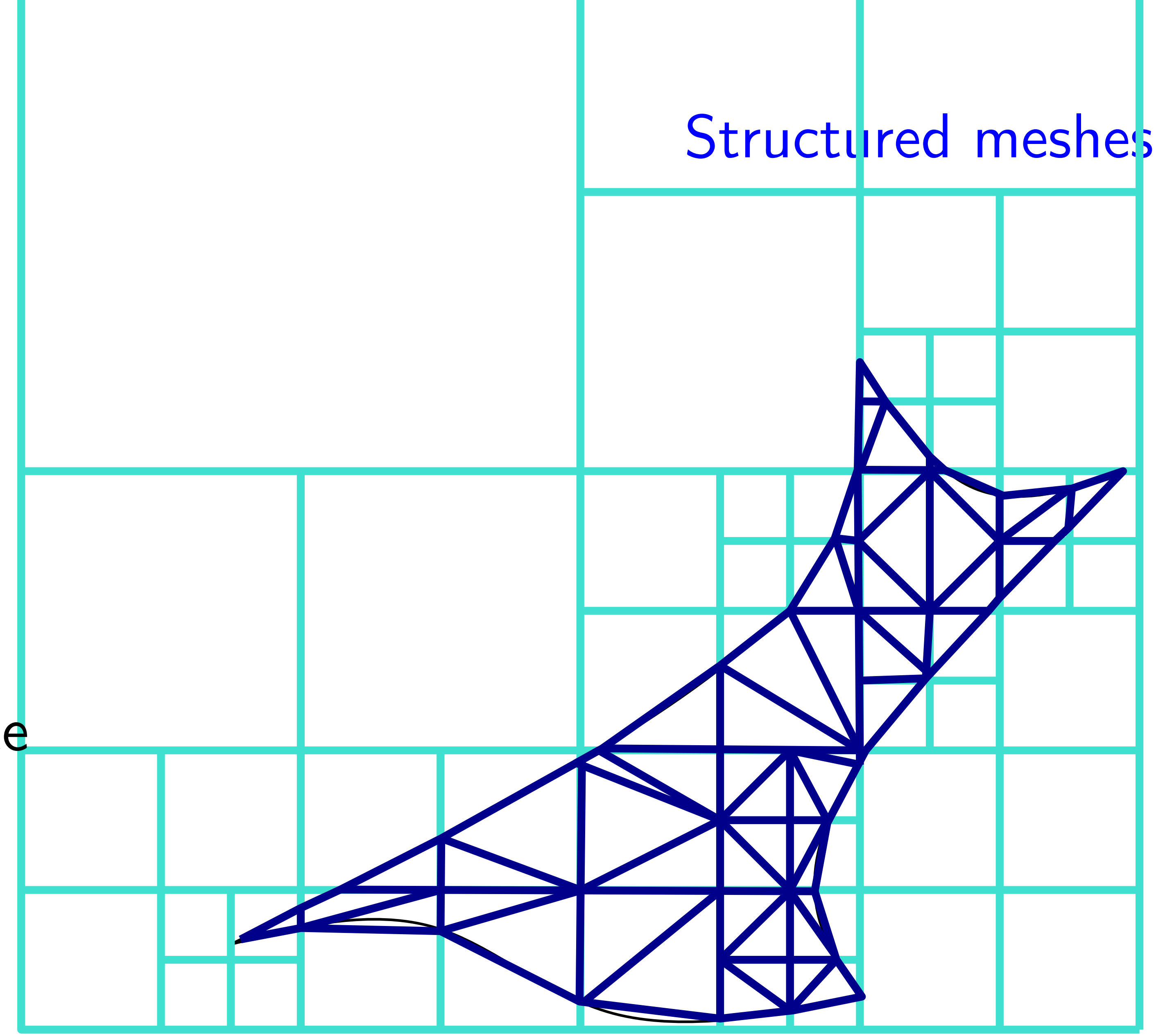
Shape



Meshing

Shape

Triangulate

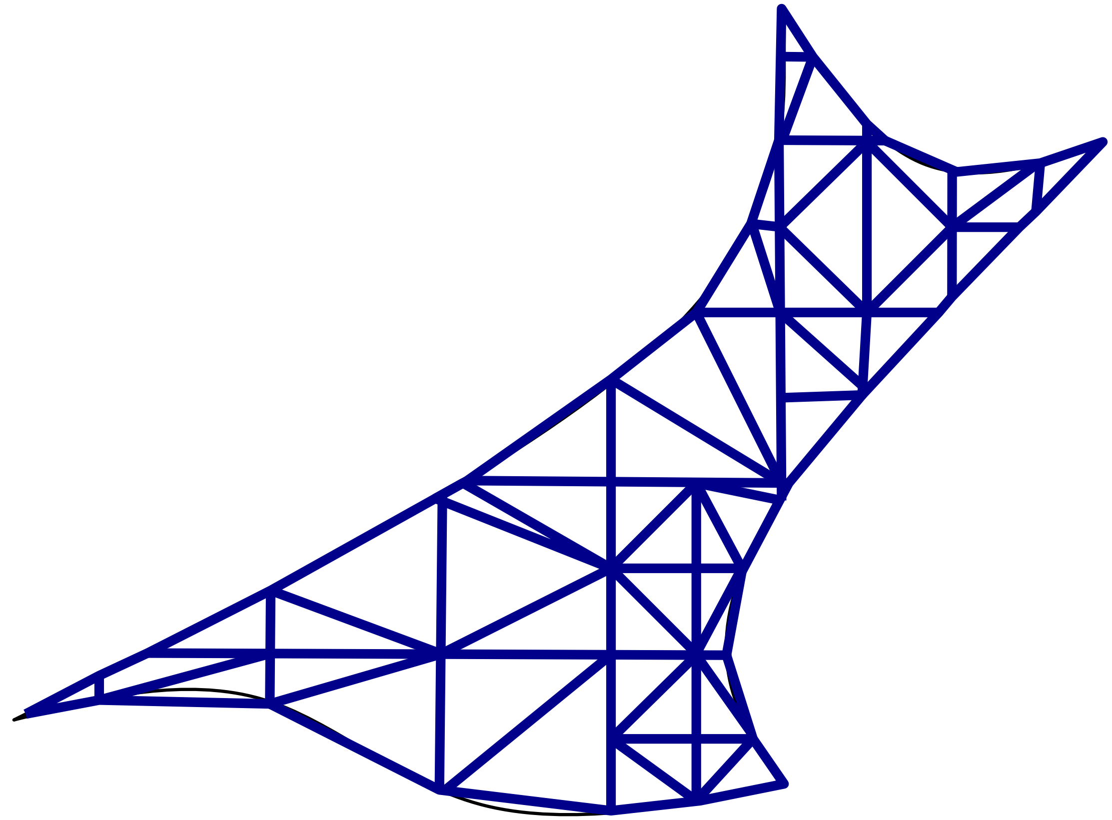


Meshing

Structured meshes

Shape

Triangulate



Adaptive mesh

Meshing

Delaunay mesh refinement [Ruppert]

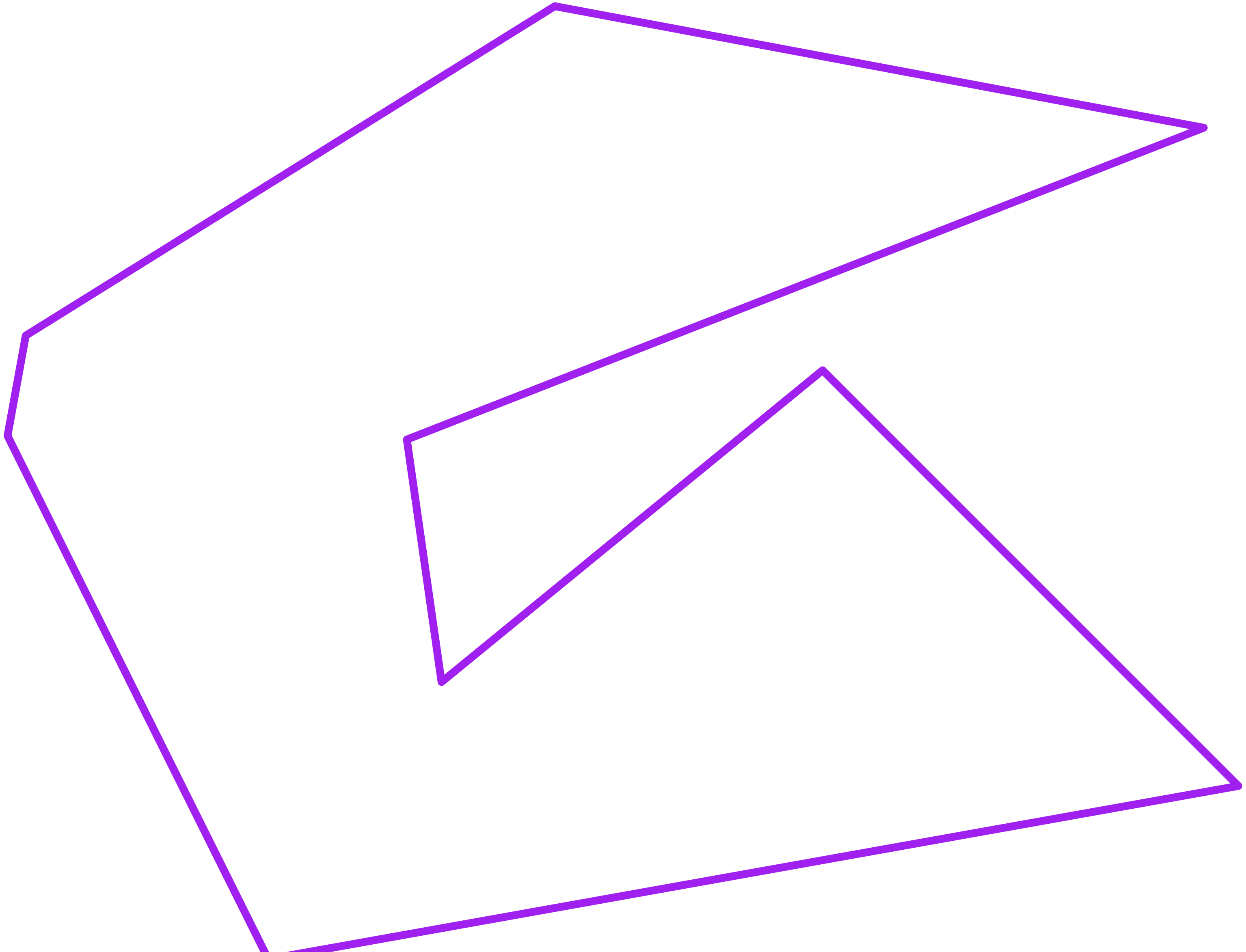
Unstructured mesh

- Use Delaunay (good angles property)
- Add vertices

Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

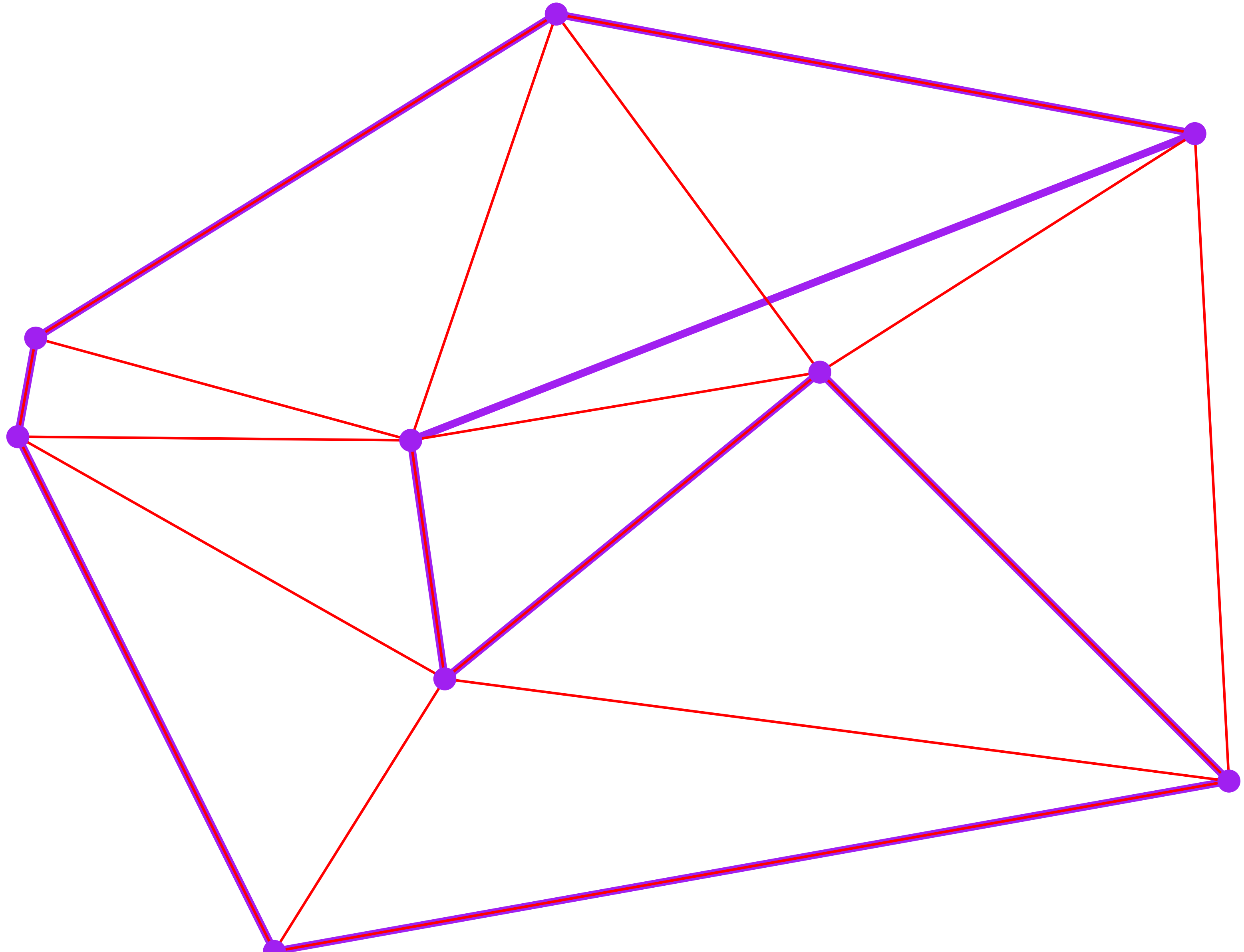


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay

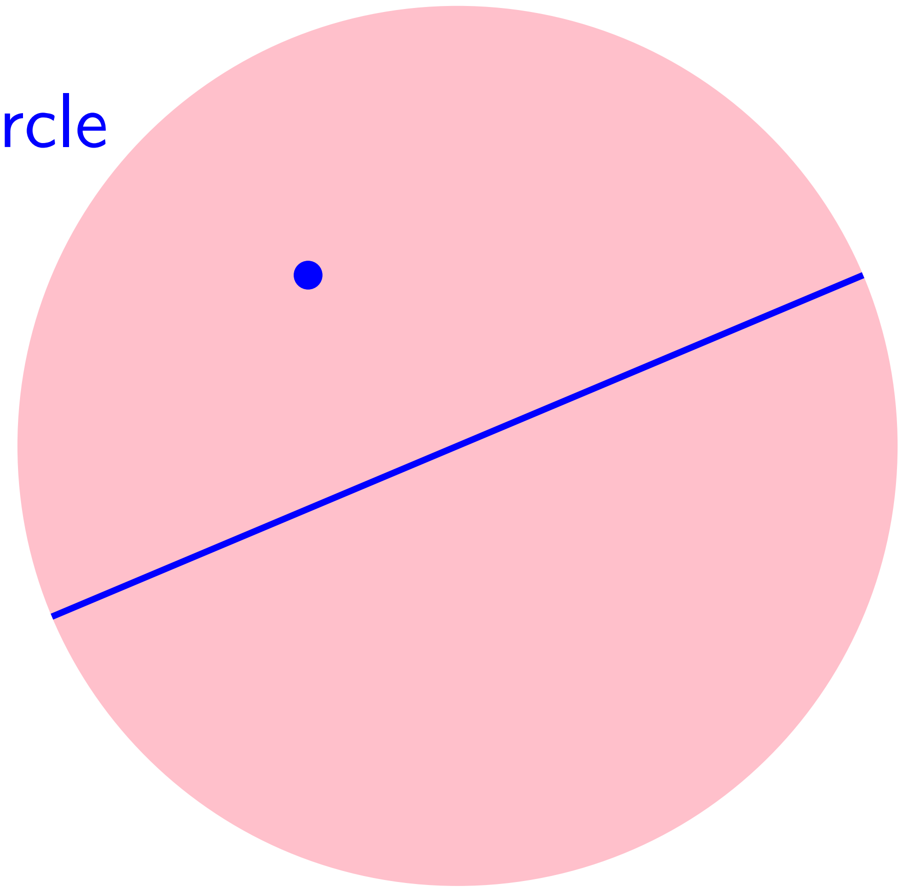


Meshing

Delaunay mesh refinement [Ruppert]

Def: Edge encroached by vertex

if inside diametral circle



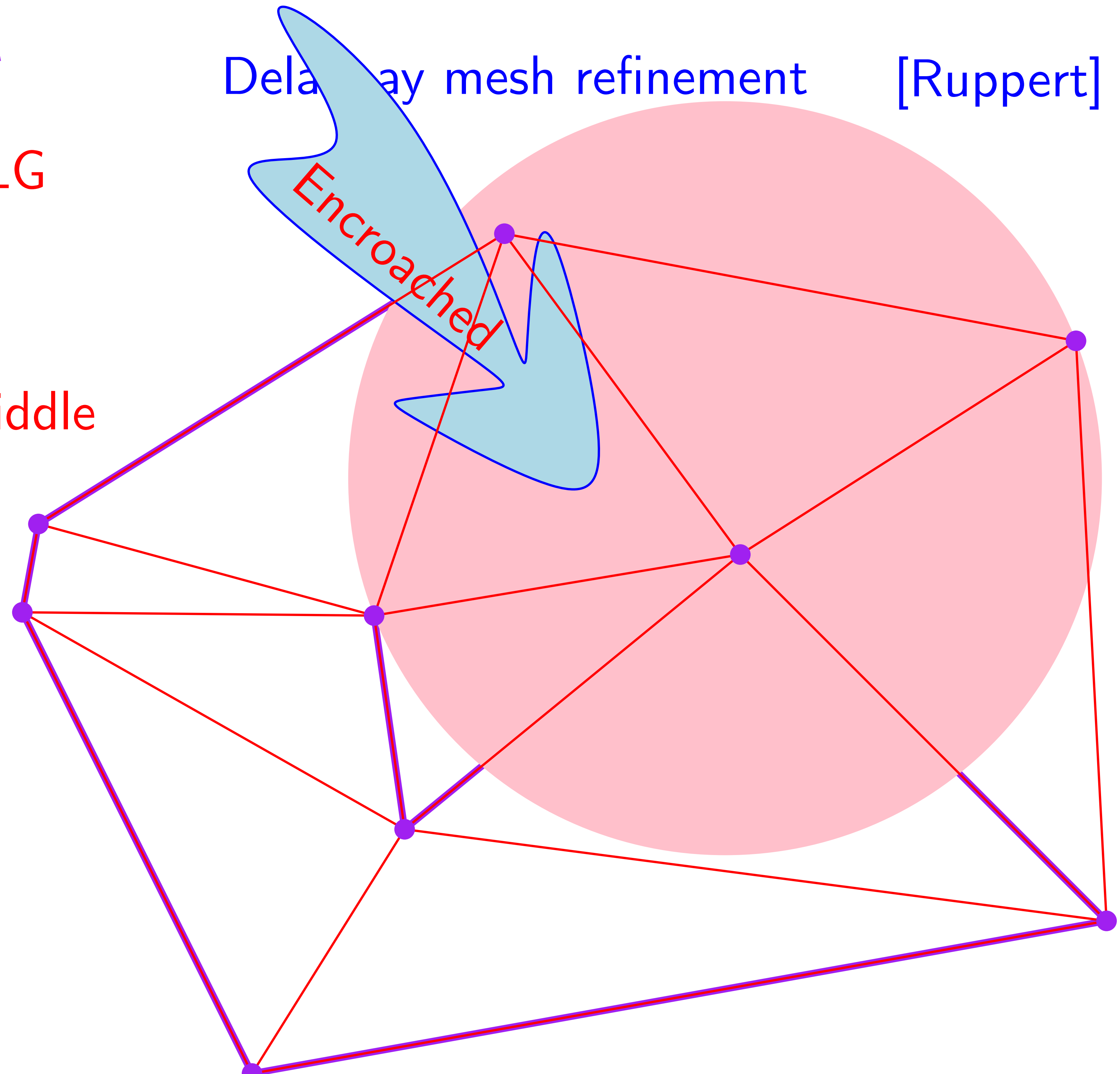
Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay

Split at middle

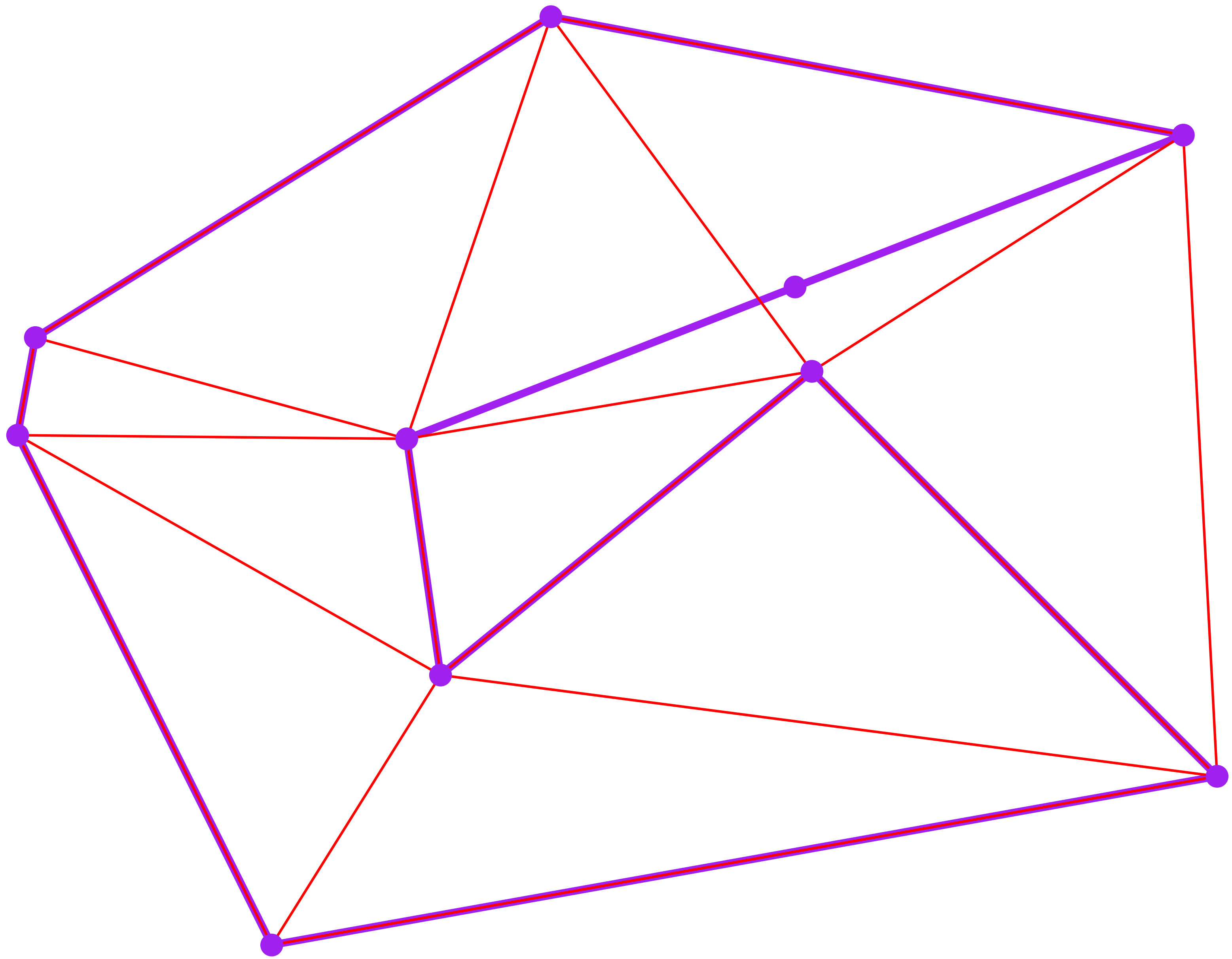


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay

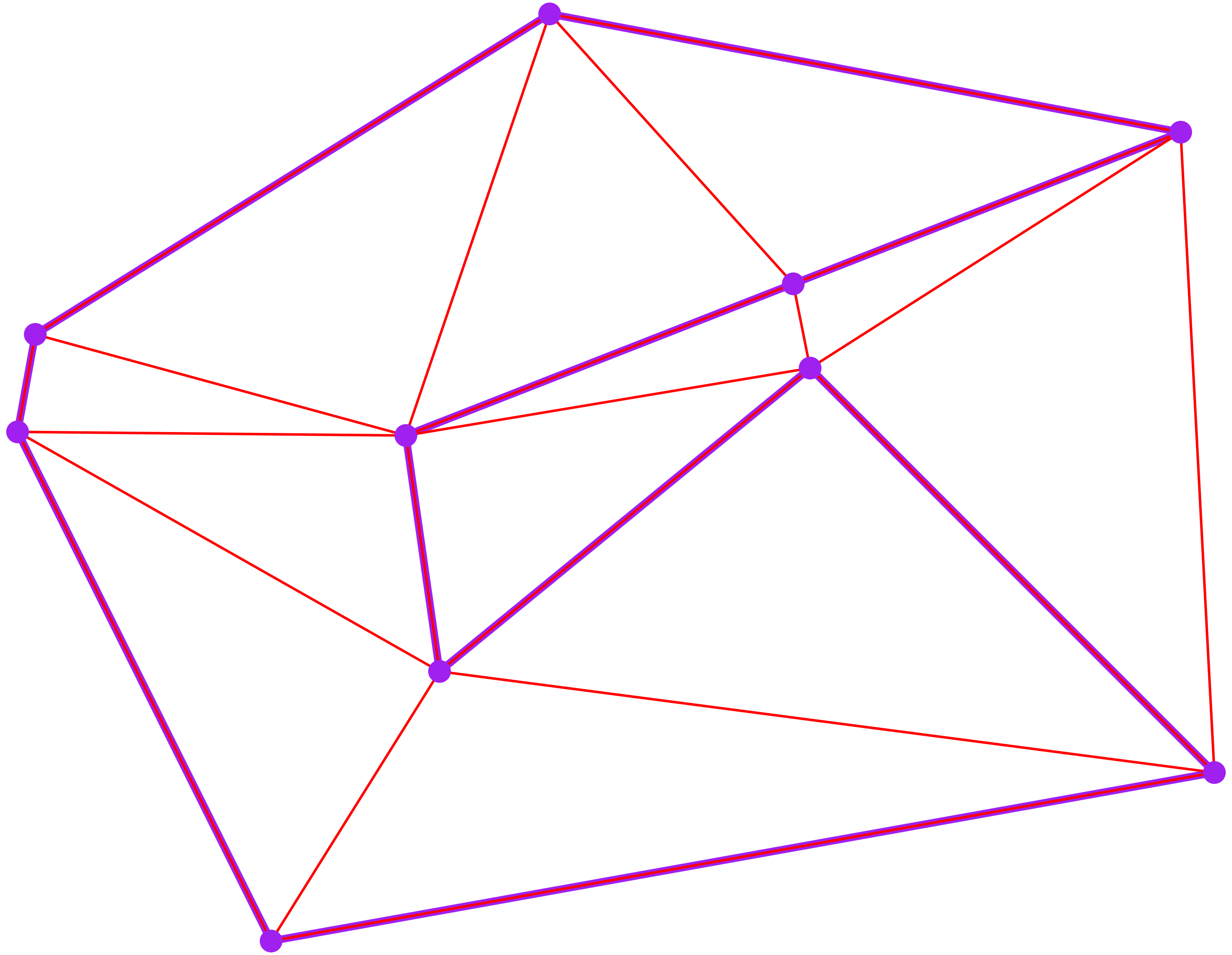


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay

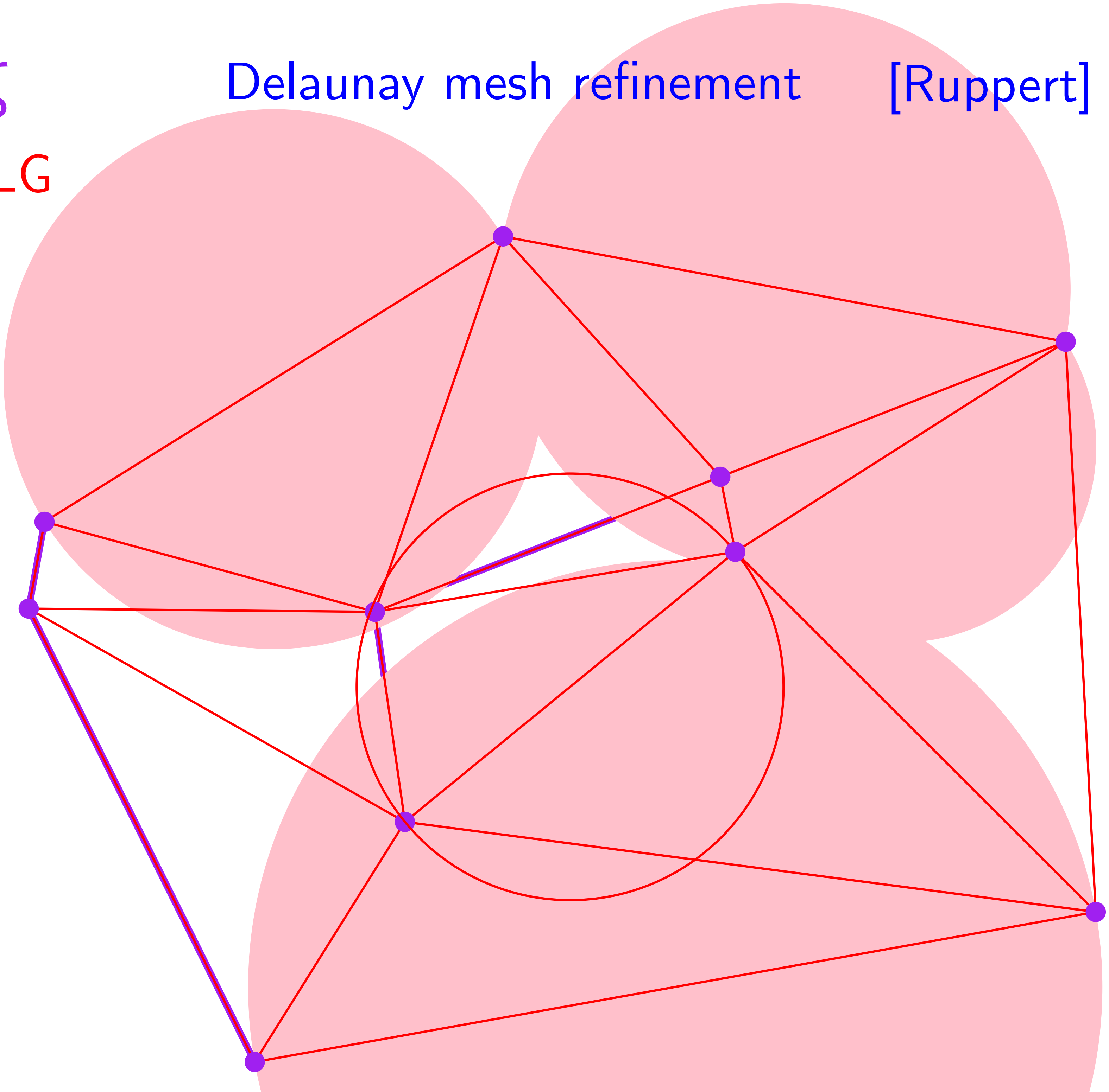


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay

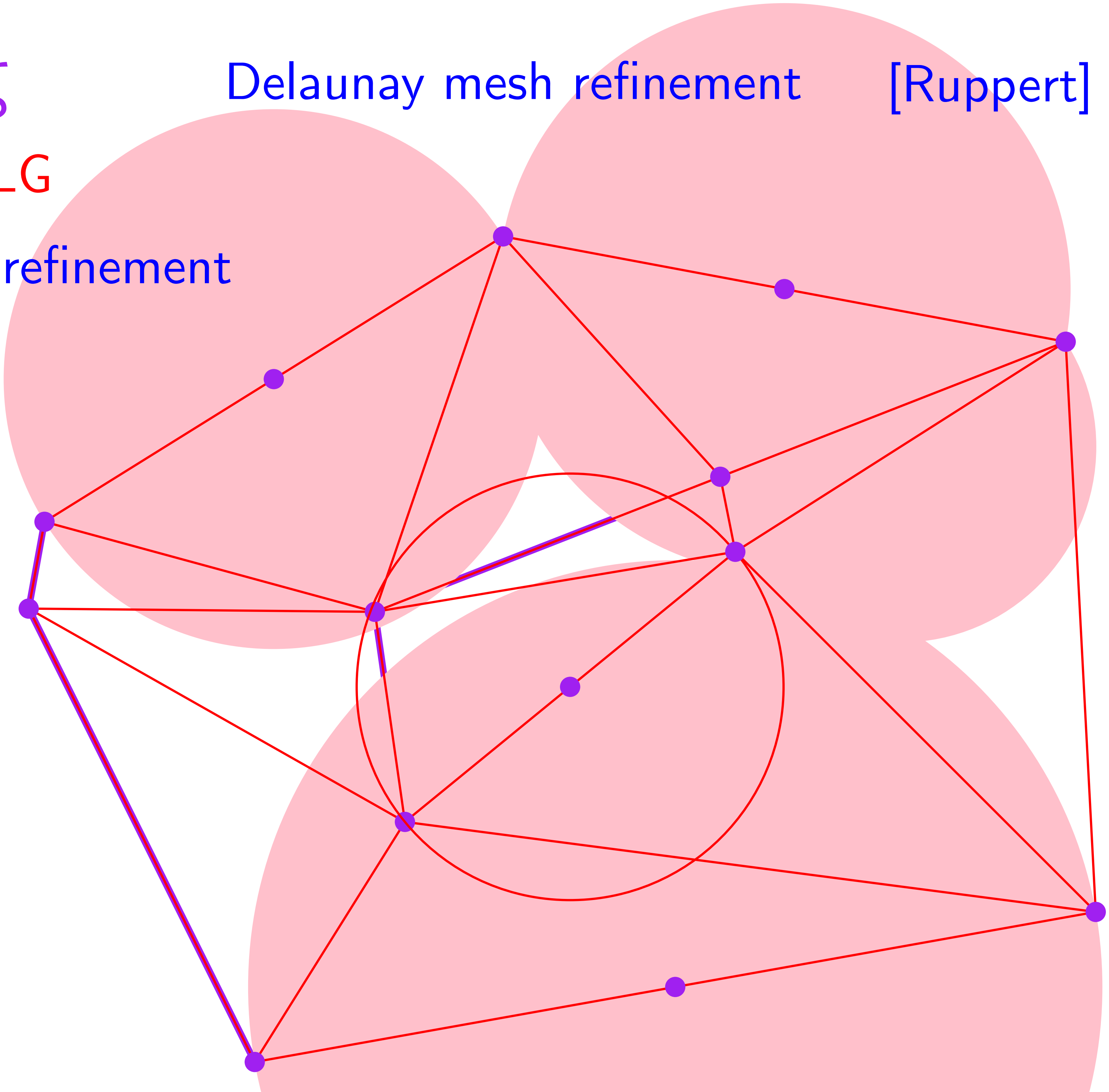


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

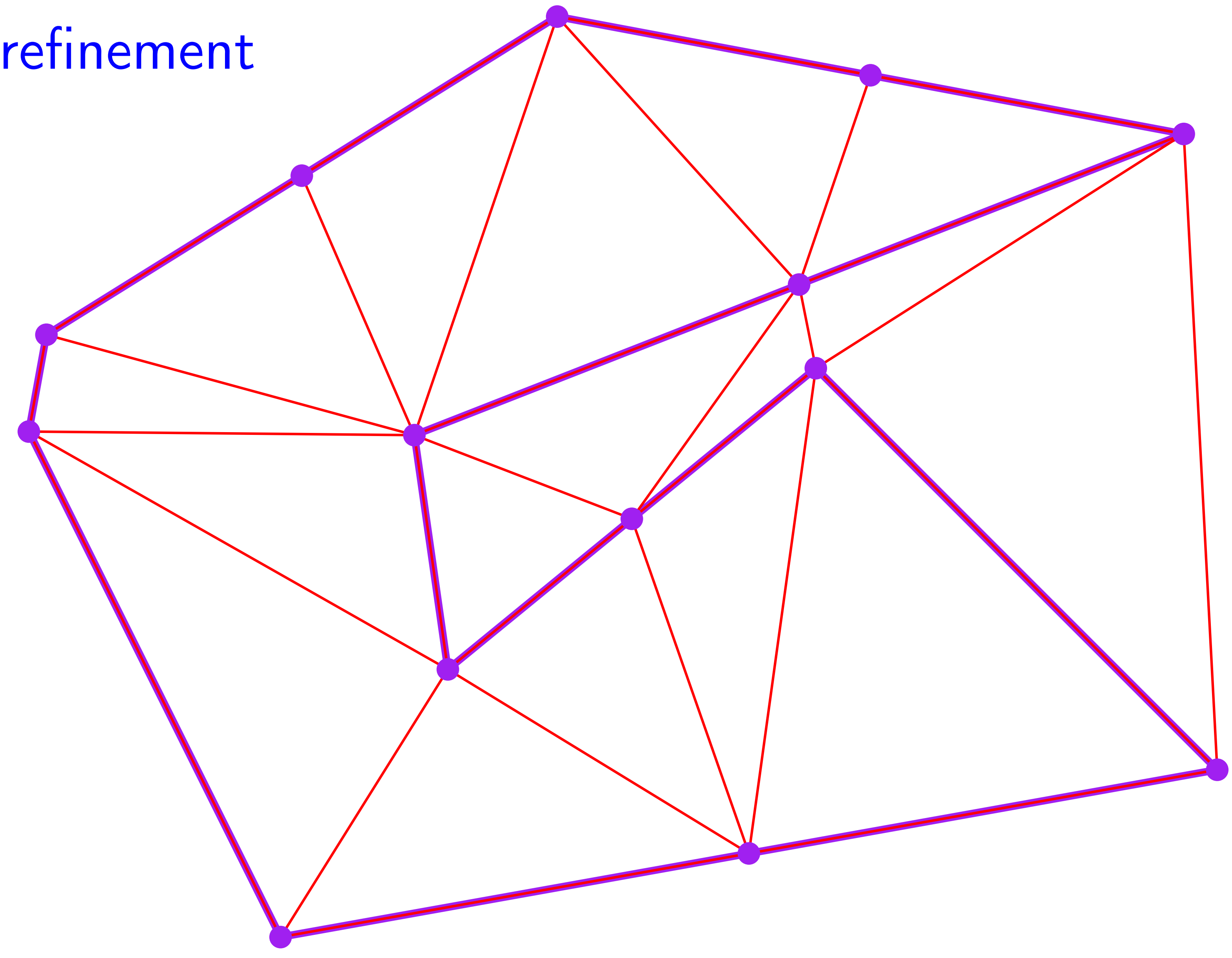


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

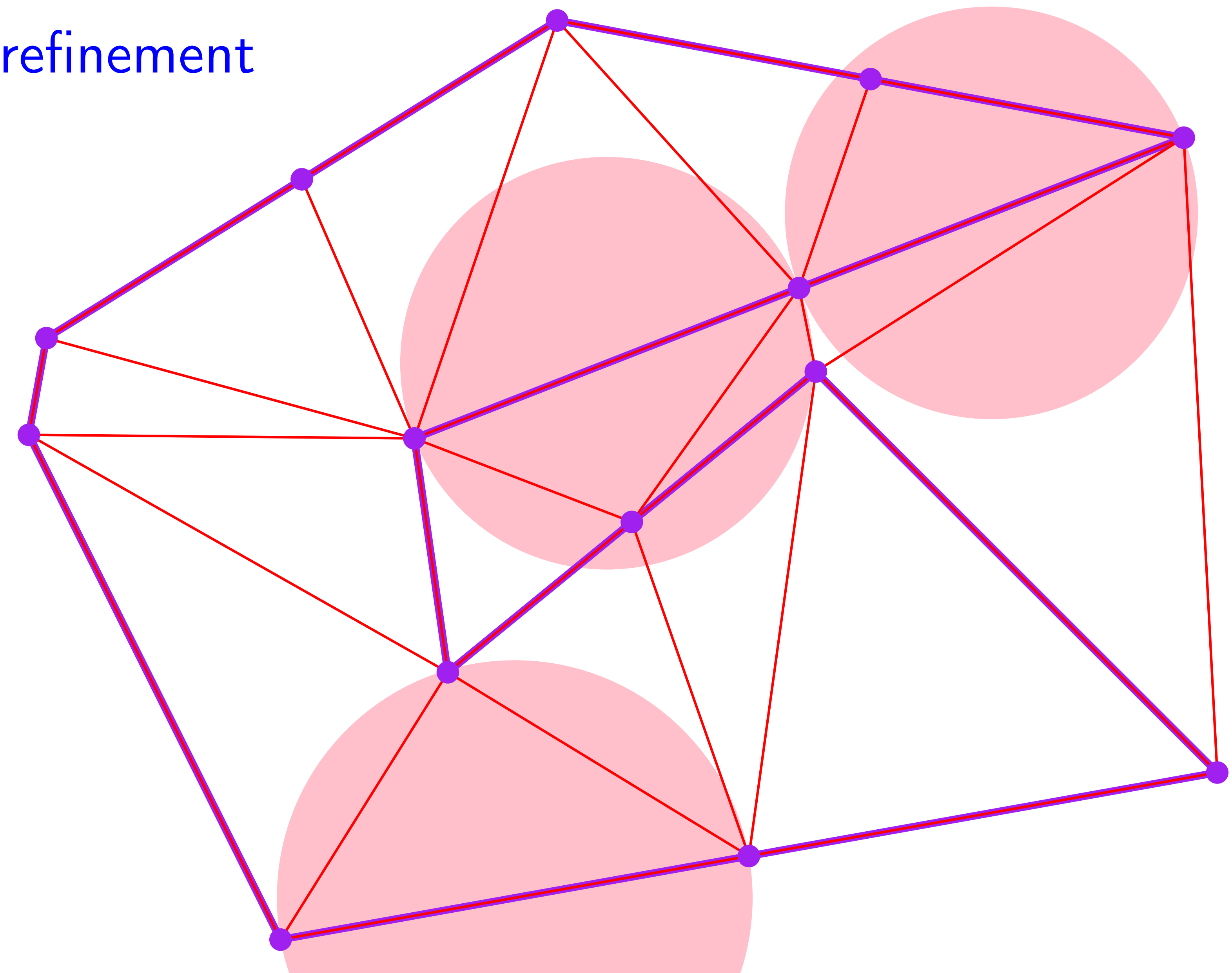


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

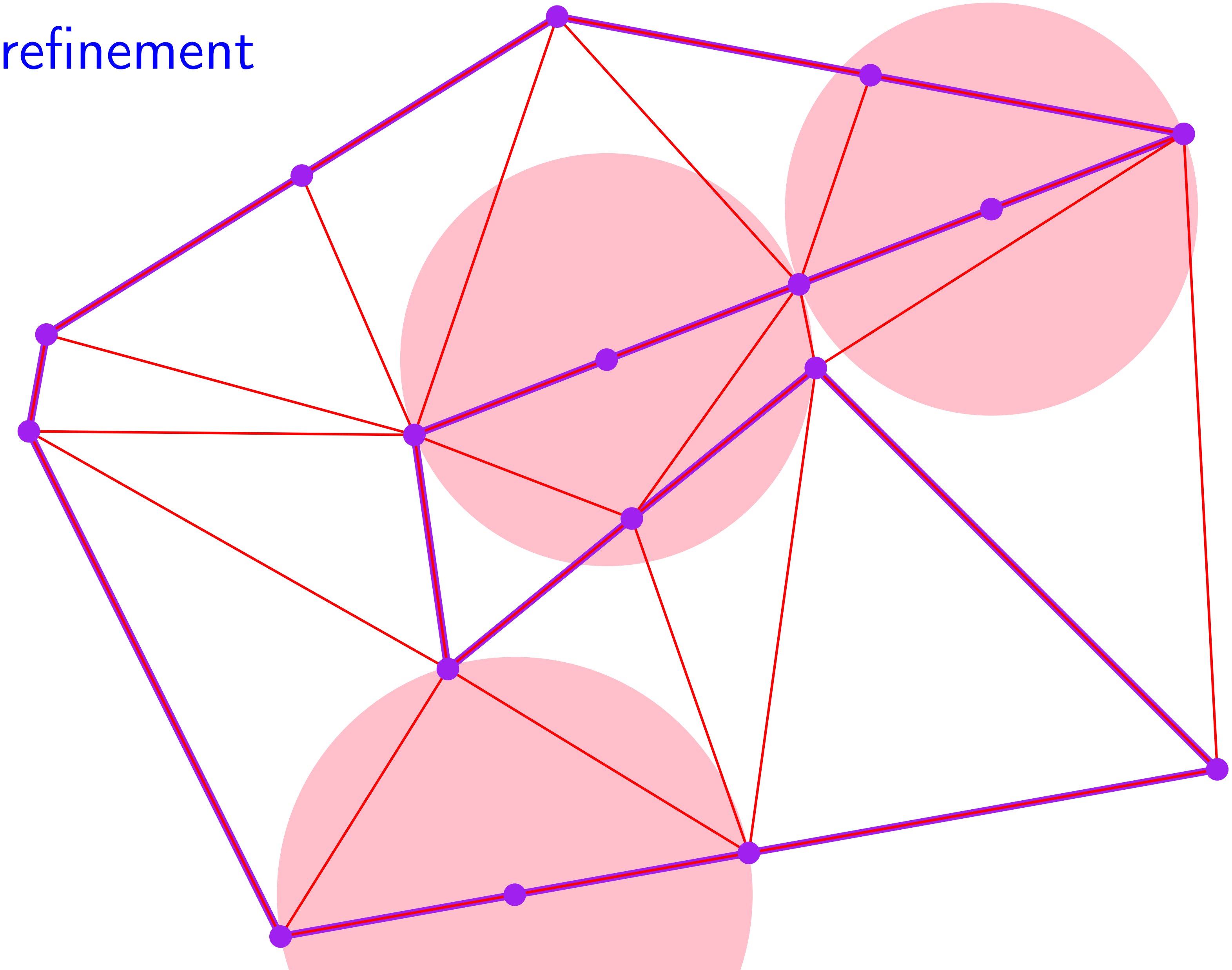


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

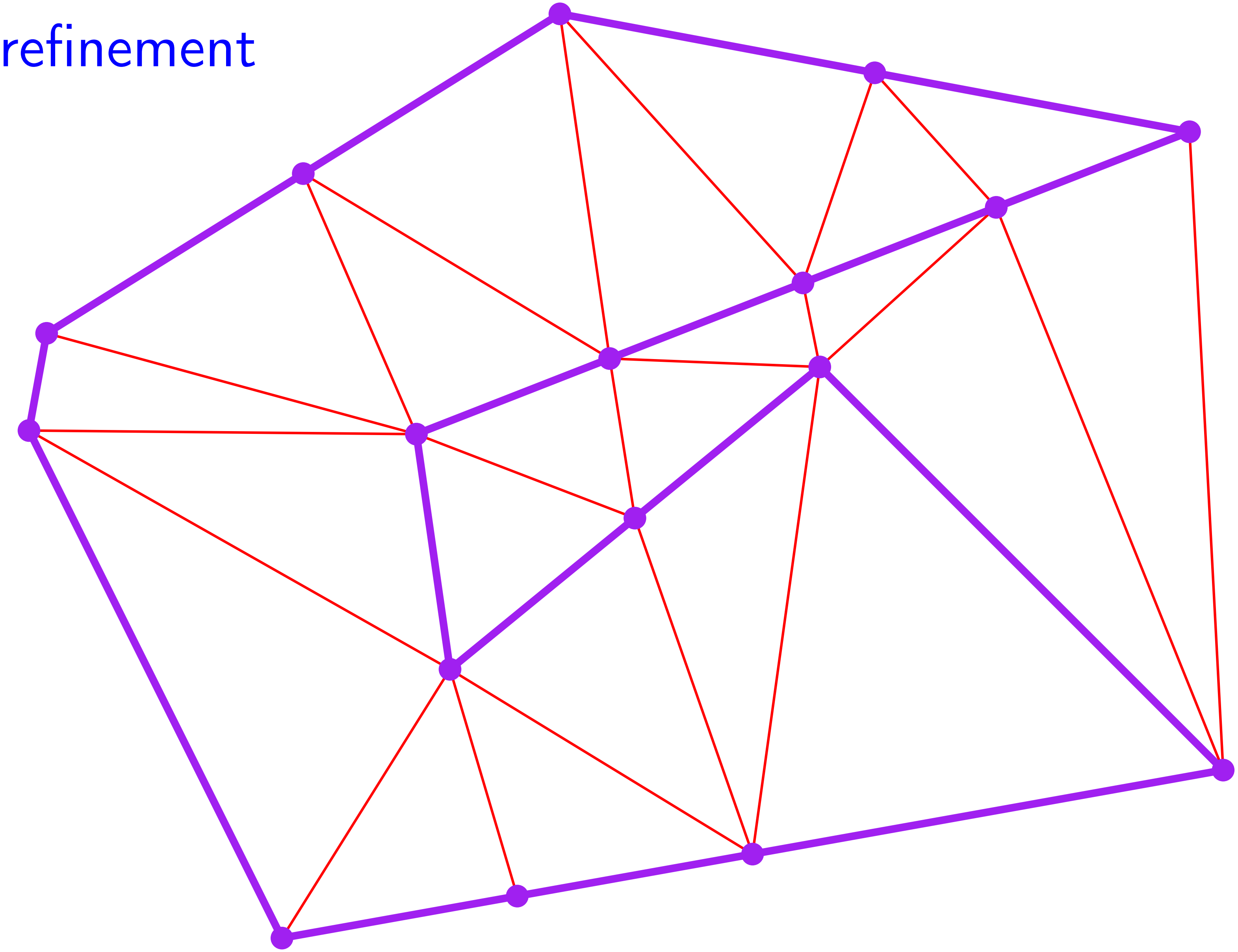


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement



Meshing

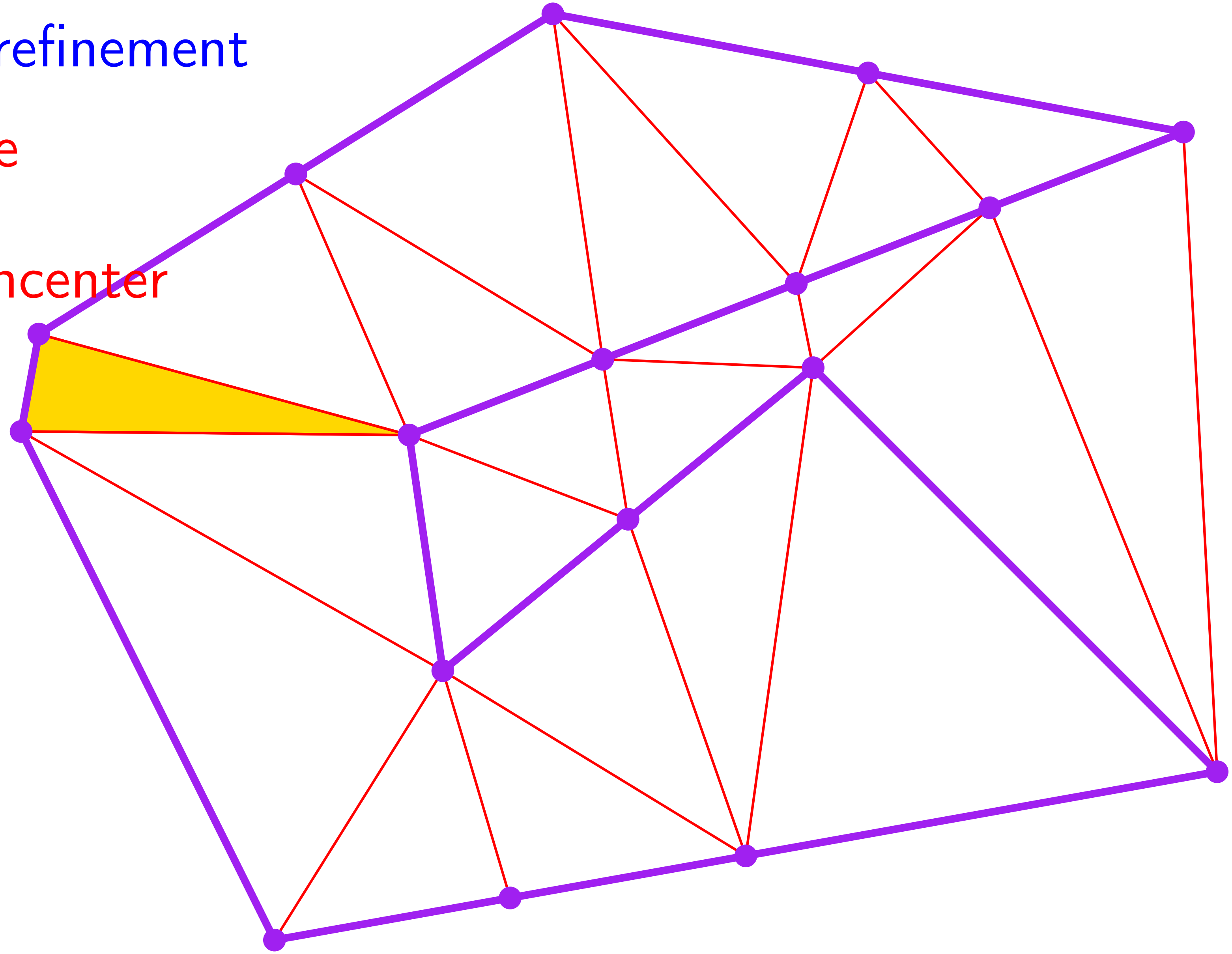
Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

Small angle

Add circumcenter



Meshing

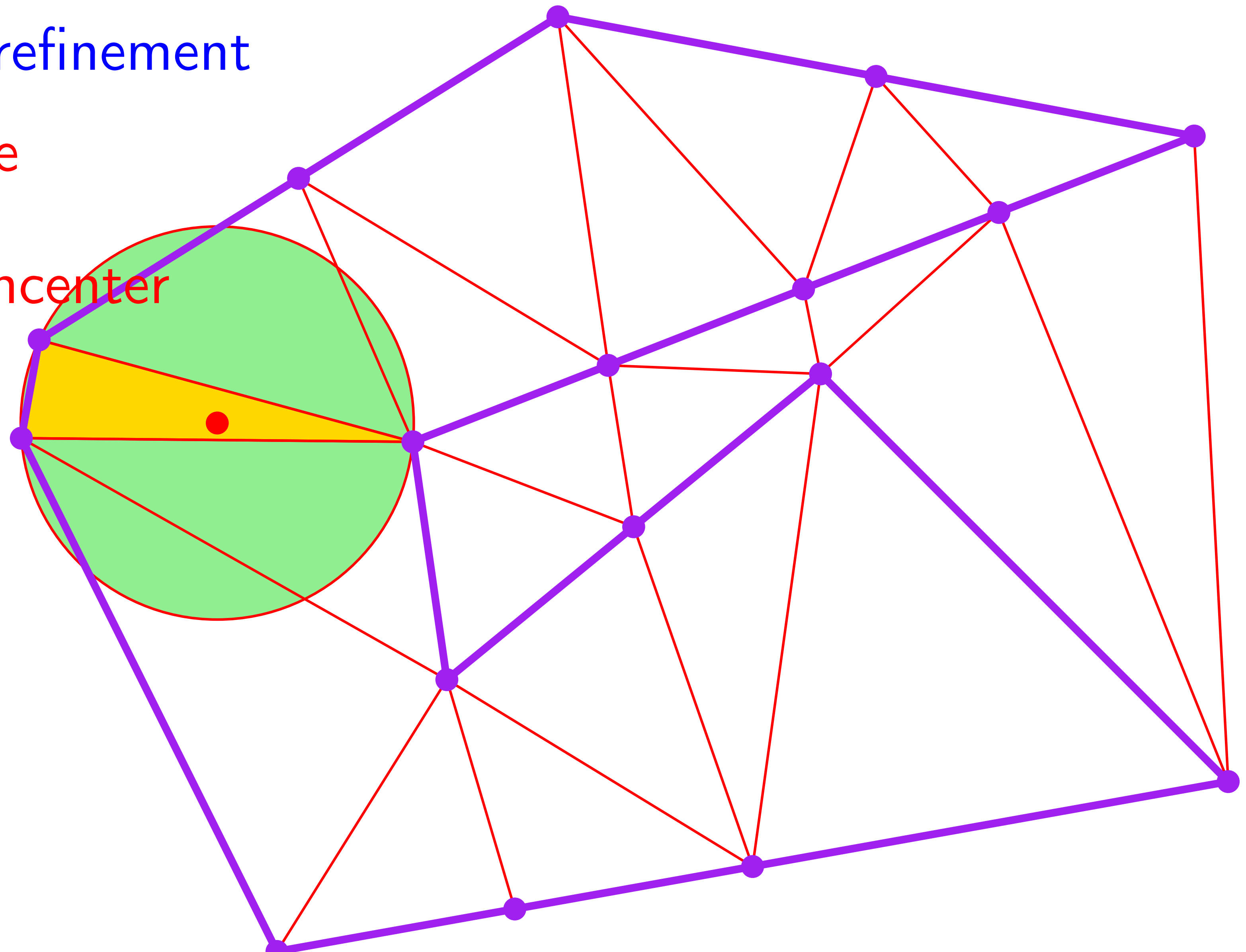
Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

Small angle

Add circumcenter

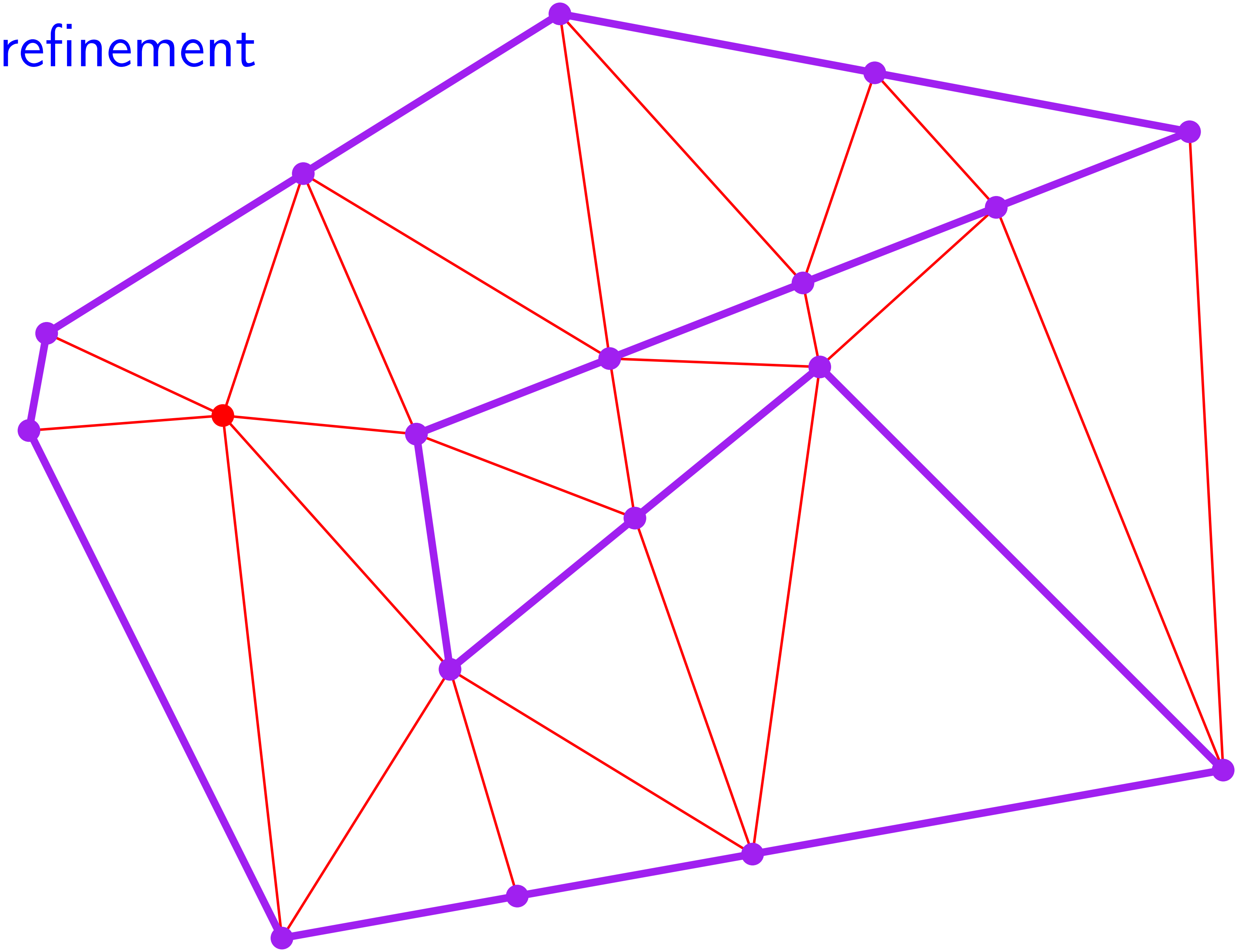


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement



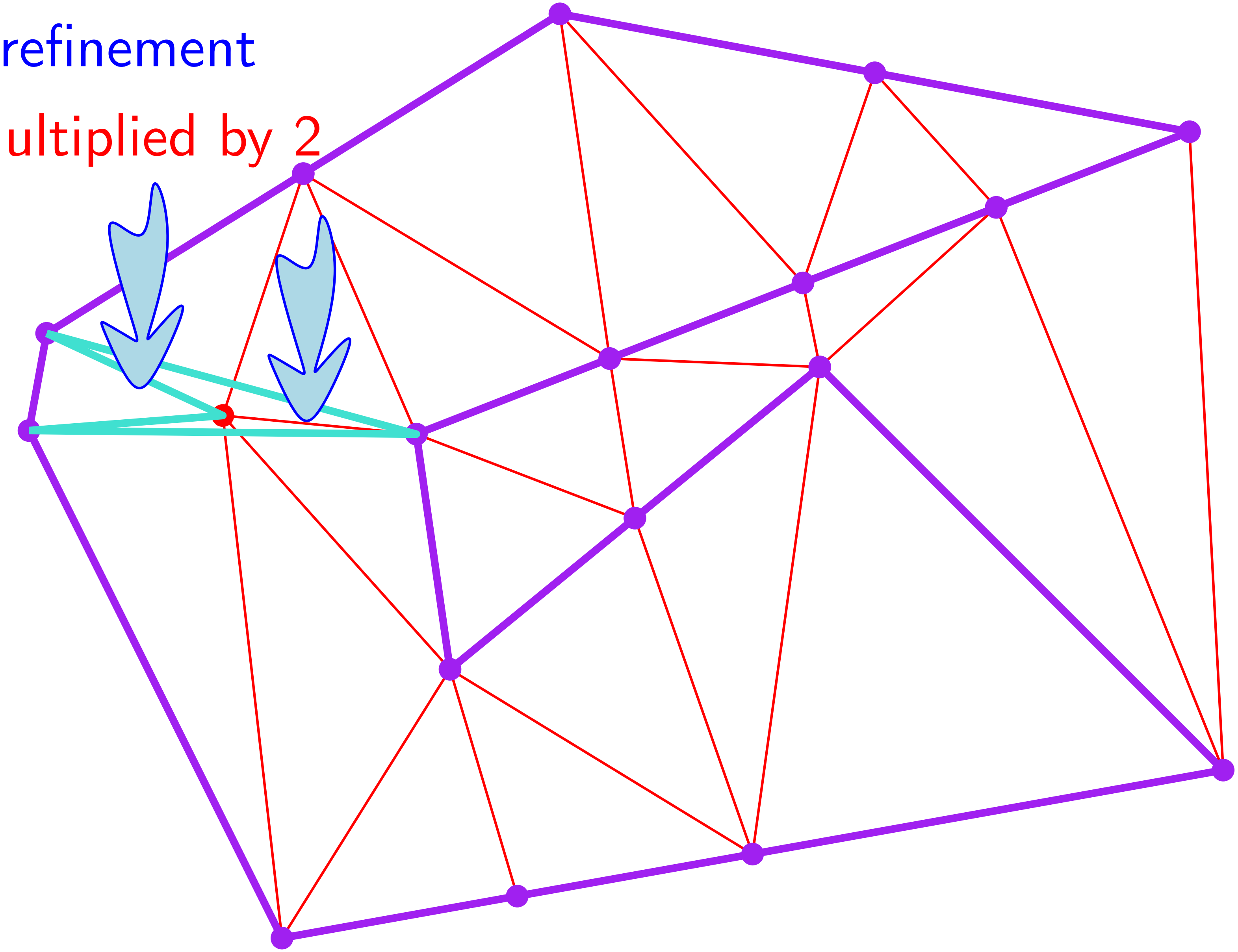
Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

Angle is multiplied by 2

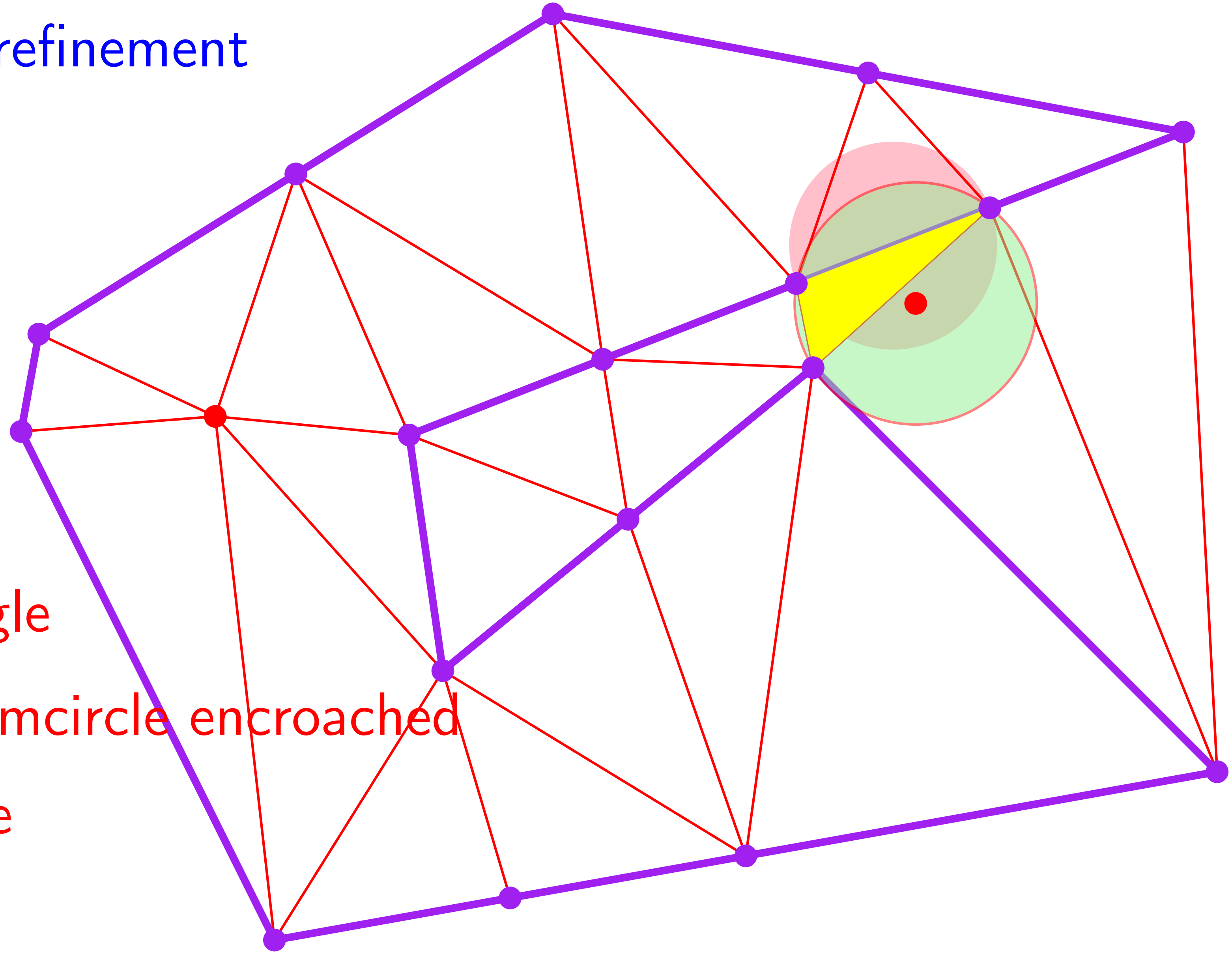


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement



Small angle

But circumcircle encroached

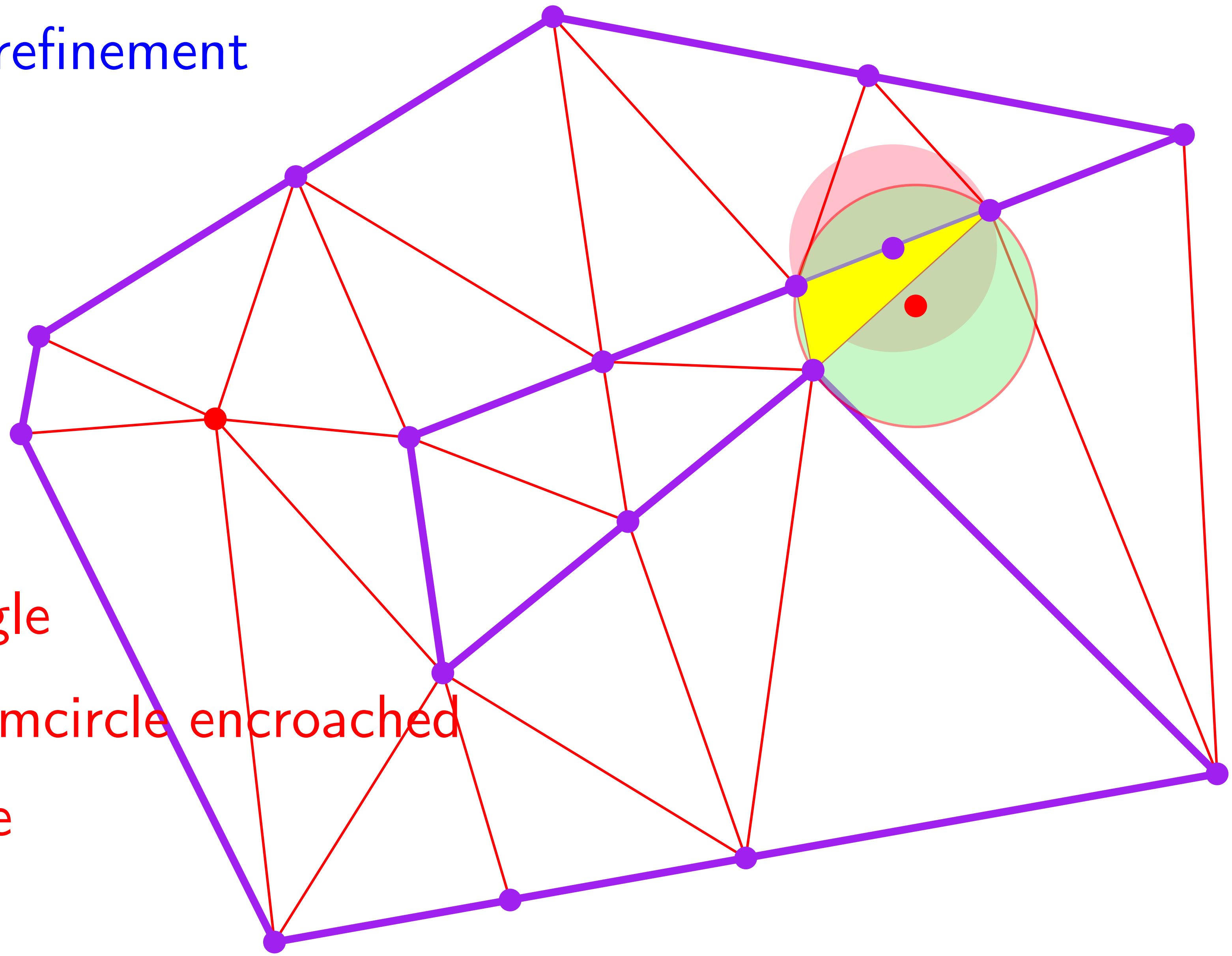
Split edge

Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement



Small angle

But circumcircle encroached

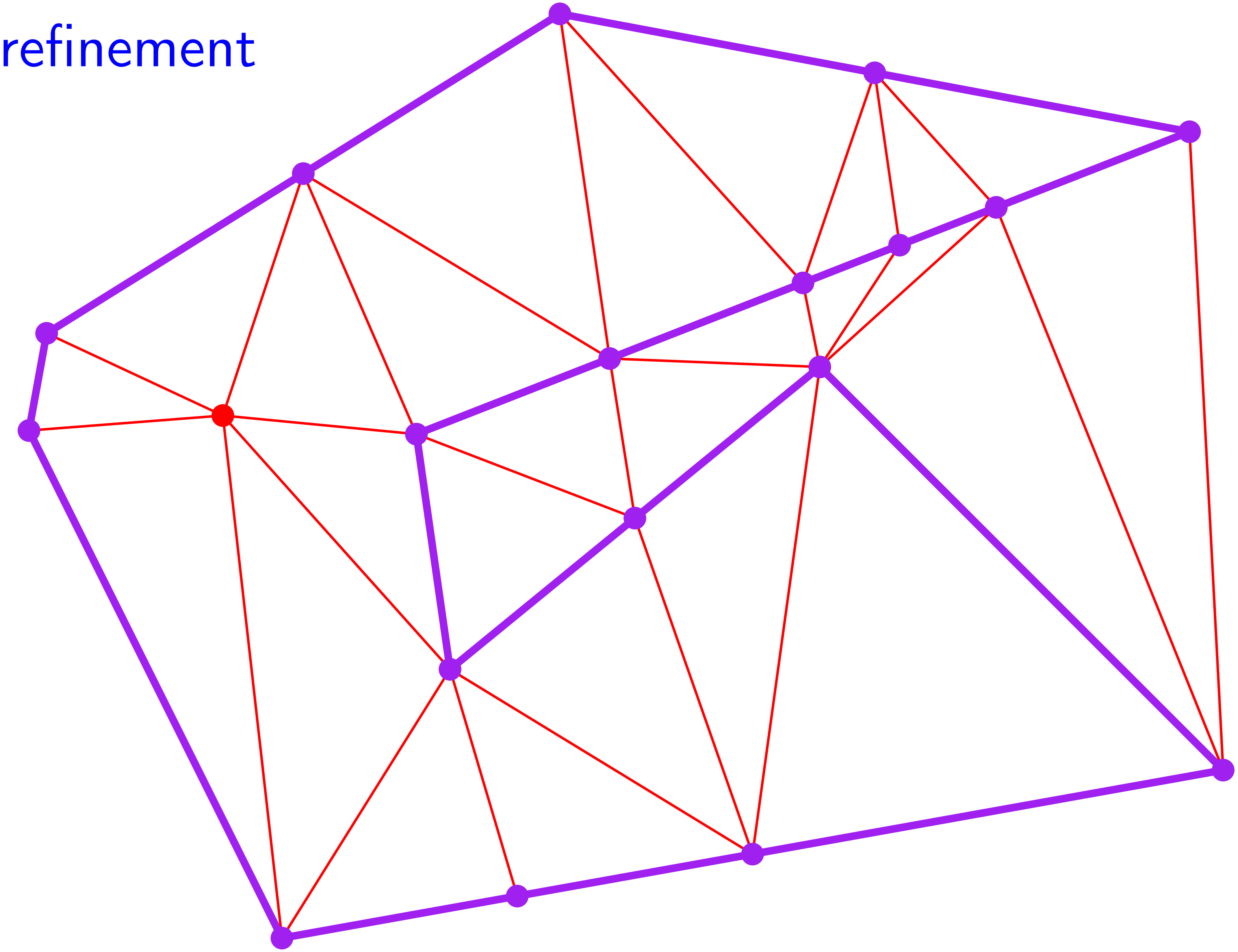
Split edge

Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

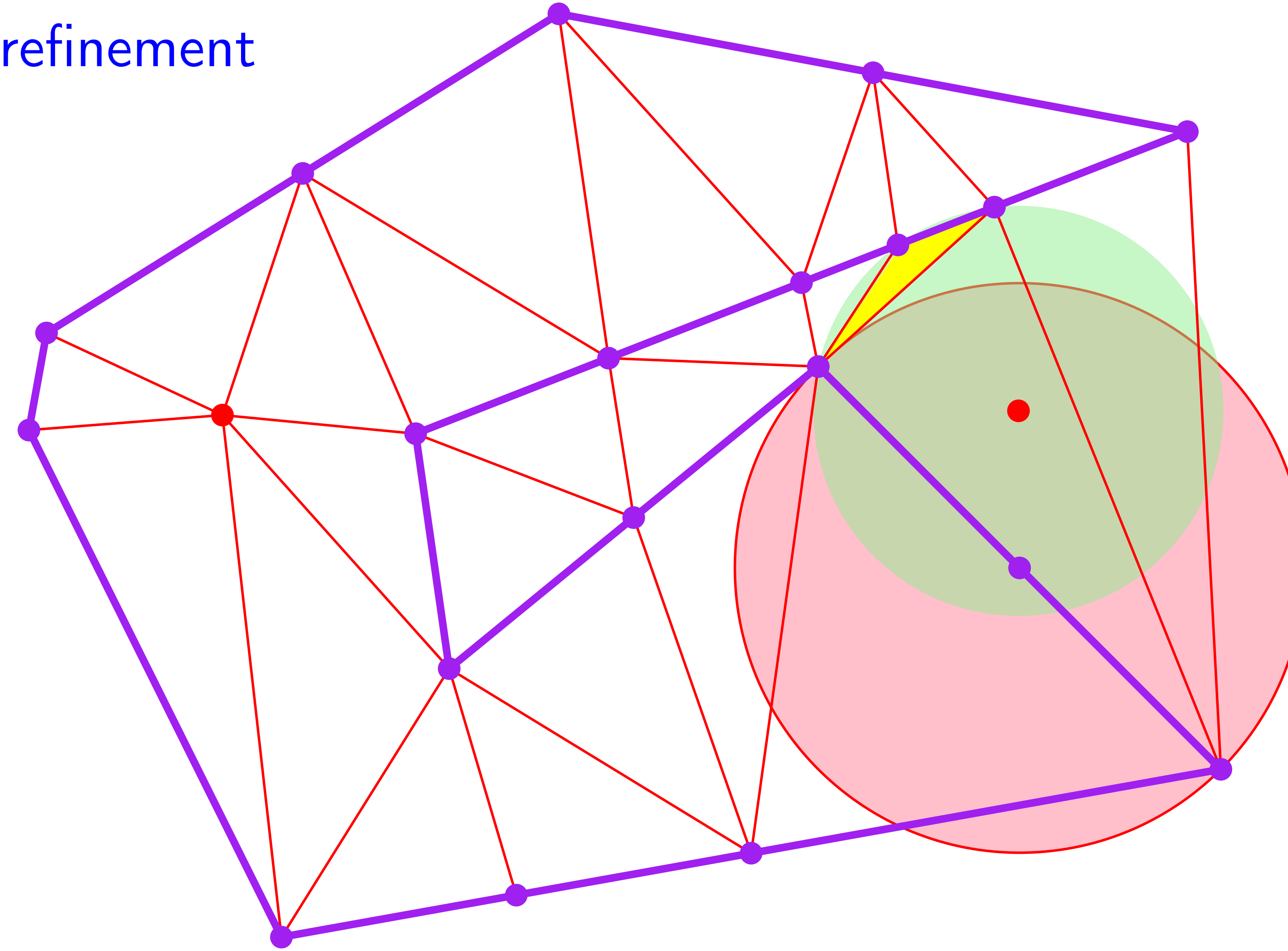


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

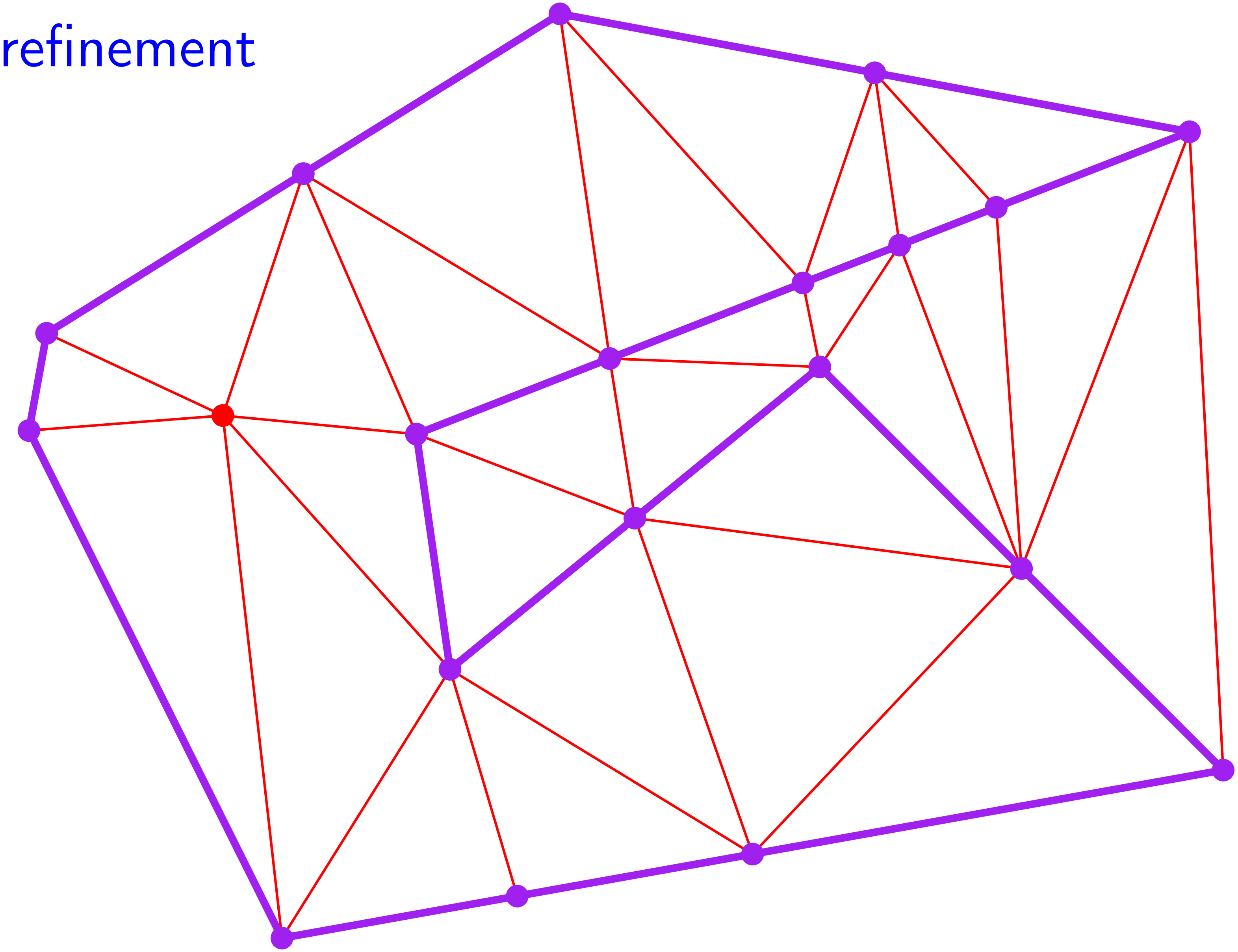


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

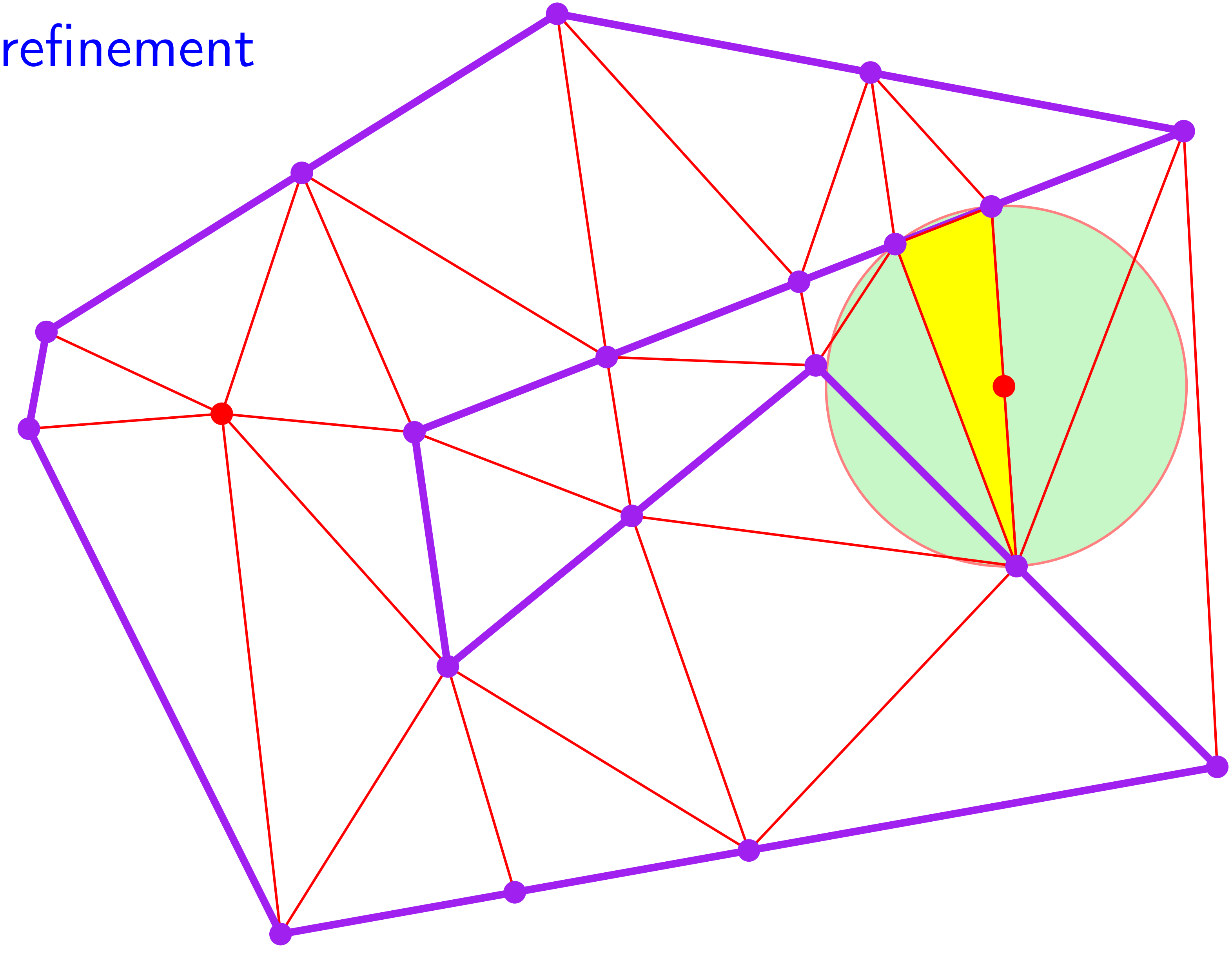


Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement



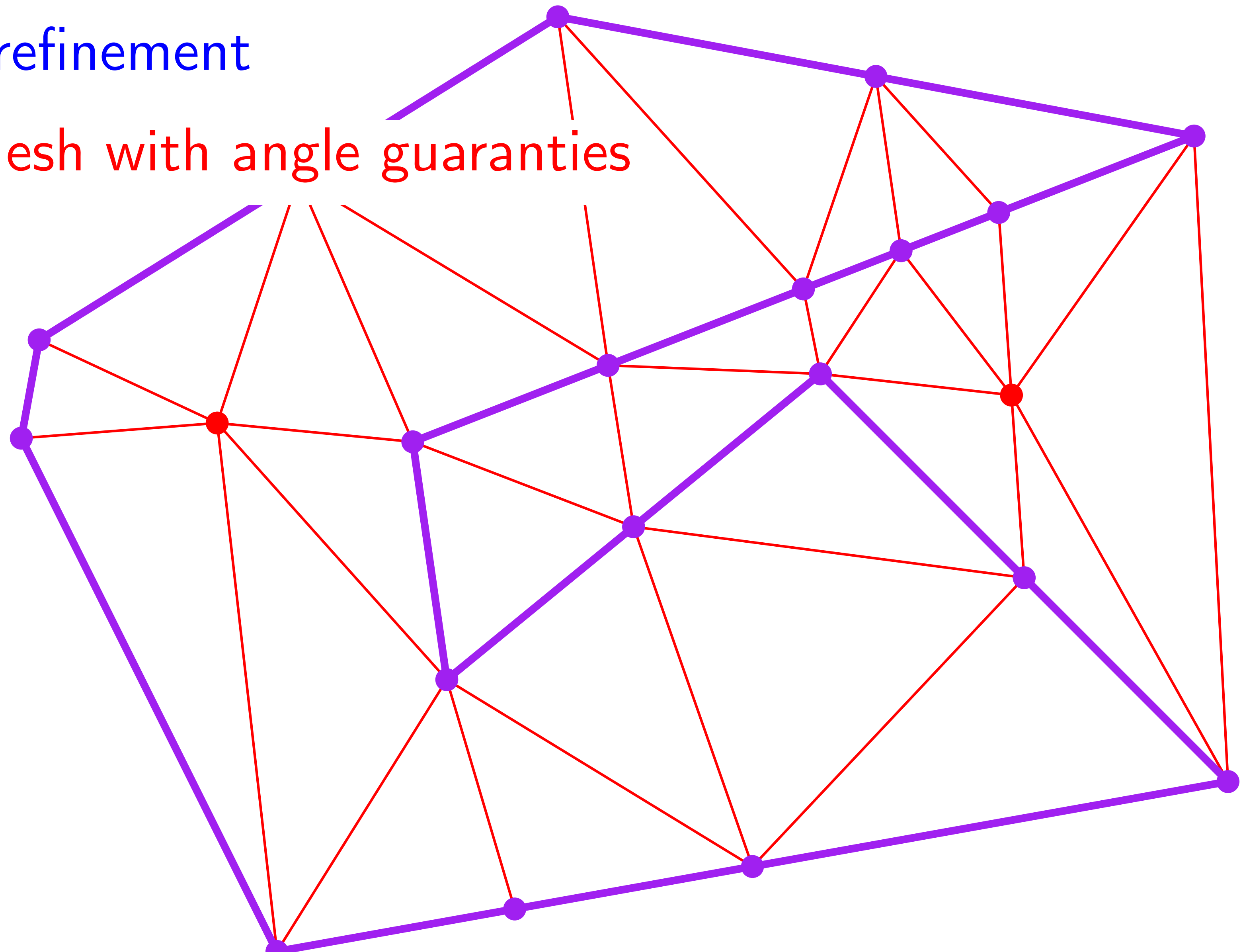
Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

Output: Mesh with angle guaranties



Meshing

Delaunay mesh refinement [Ruppert]

Small angles means $< \alpha < 20^\circ$

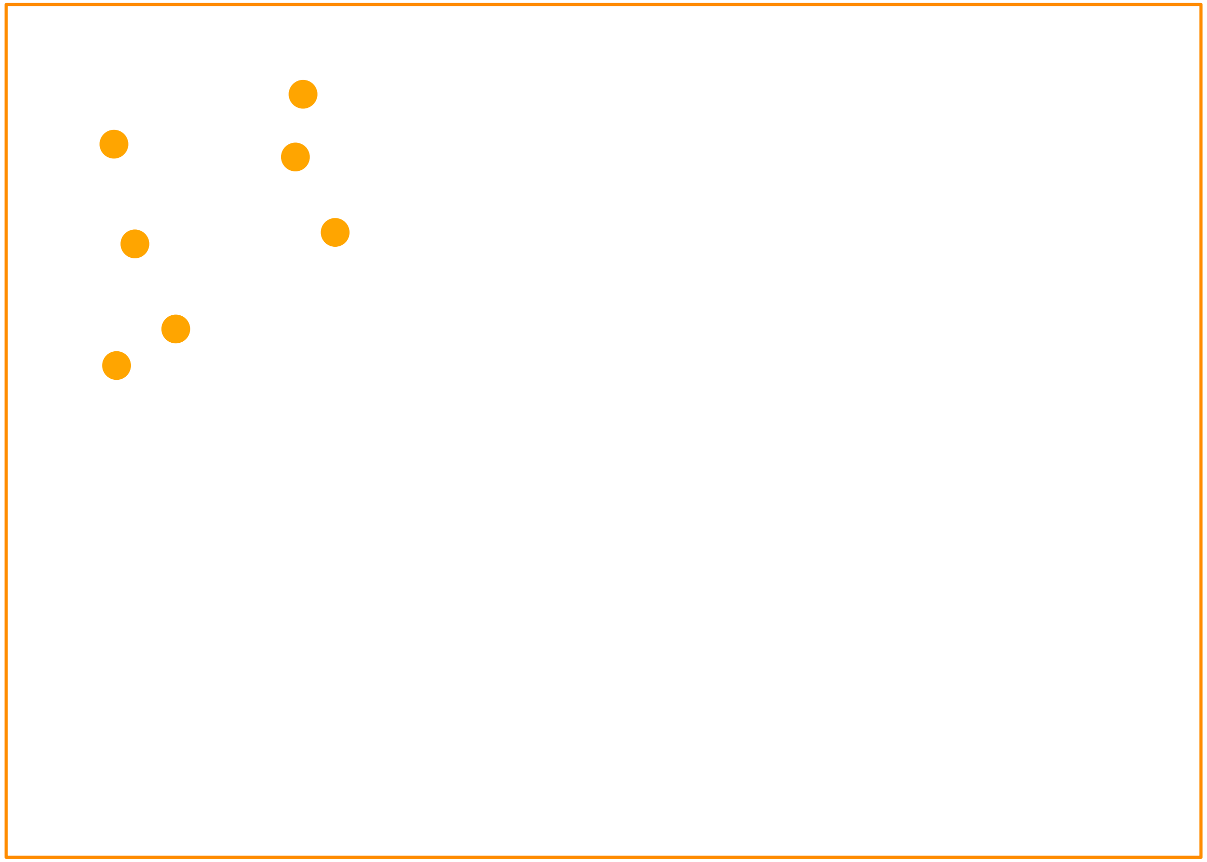
Theorem: algorithm terminates with mesh of size O(optimal)

Meshing

LLoyd iteration

Meshing

LLoyd iteration



Meshing

LLoyd iteration

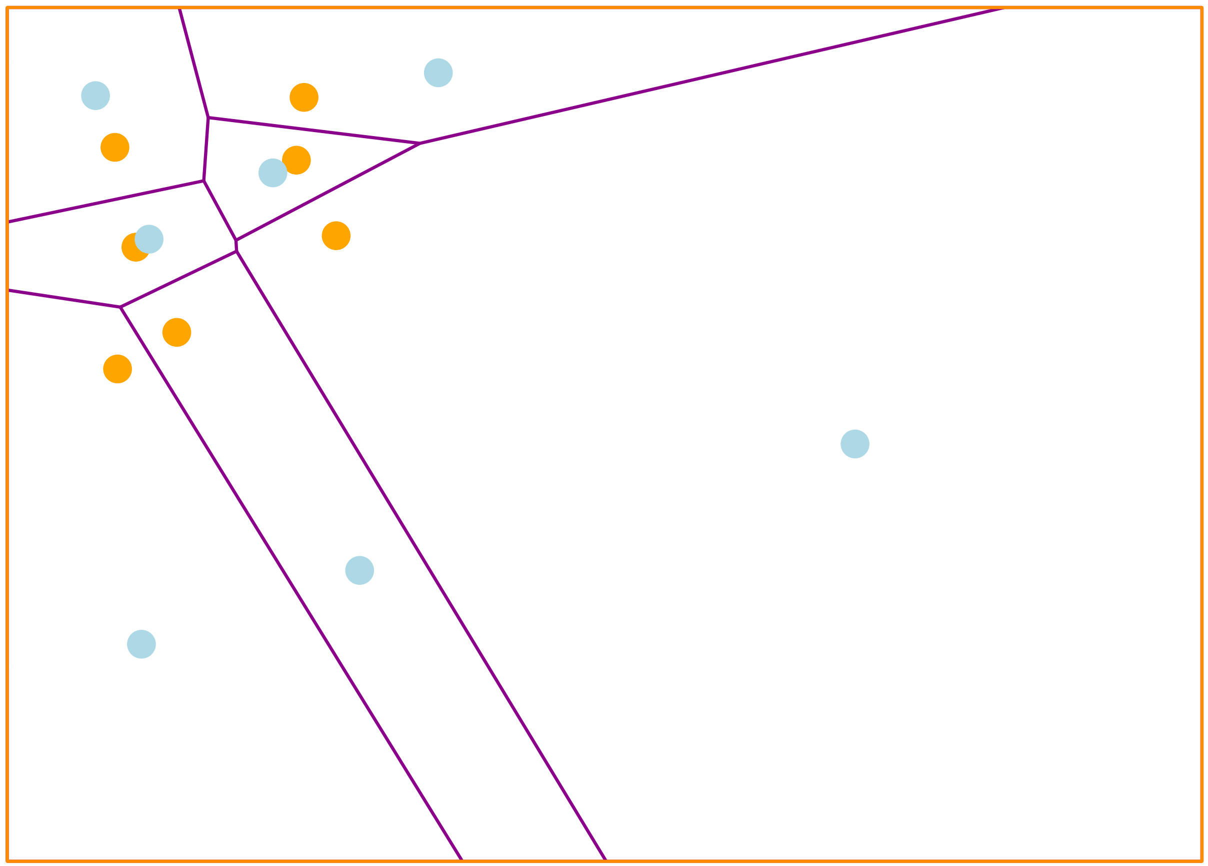
Meshing

Delaunay mesh optimization

LLoyd iteration

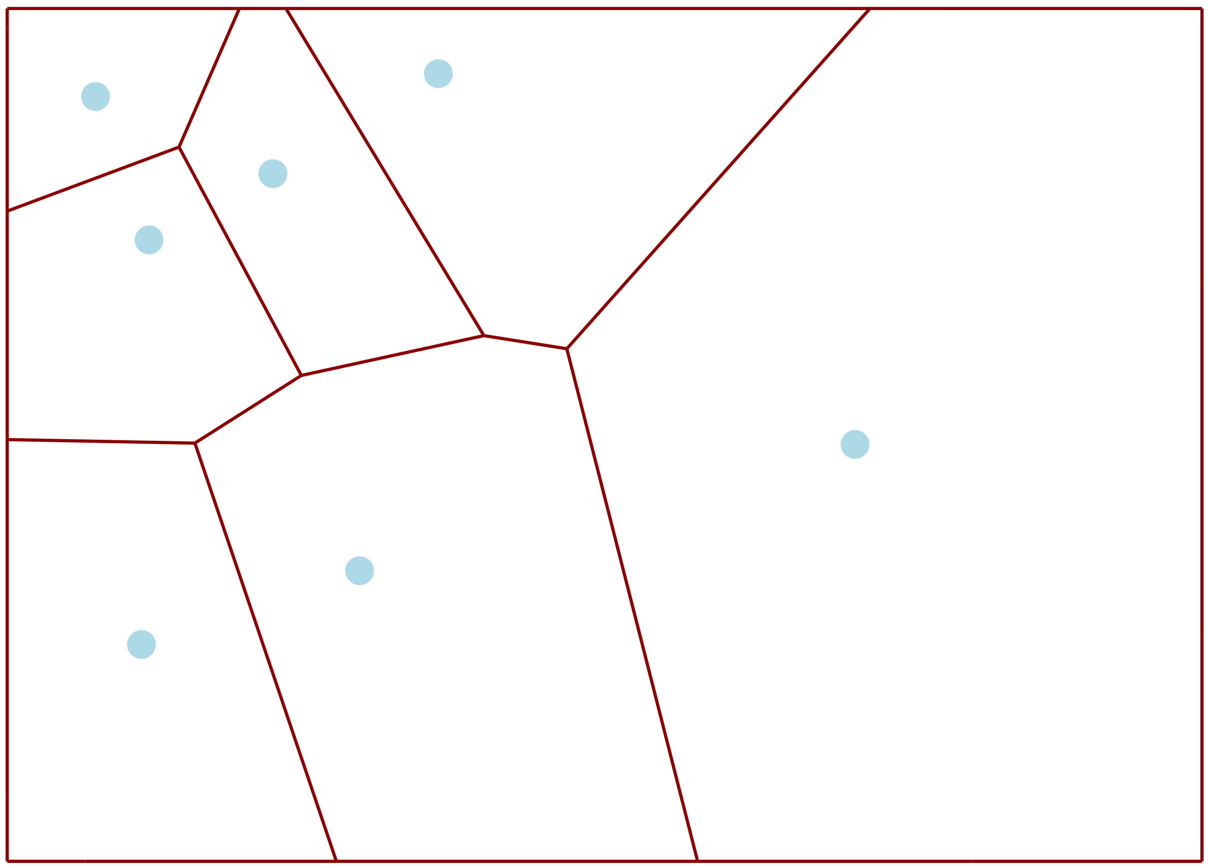
Move to barycenter

Clip by some boundary



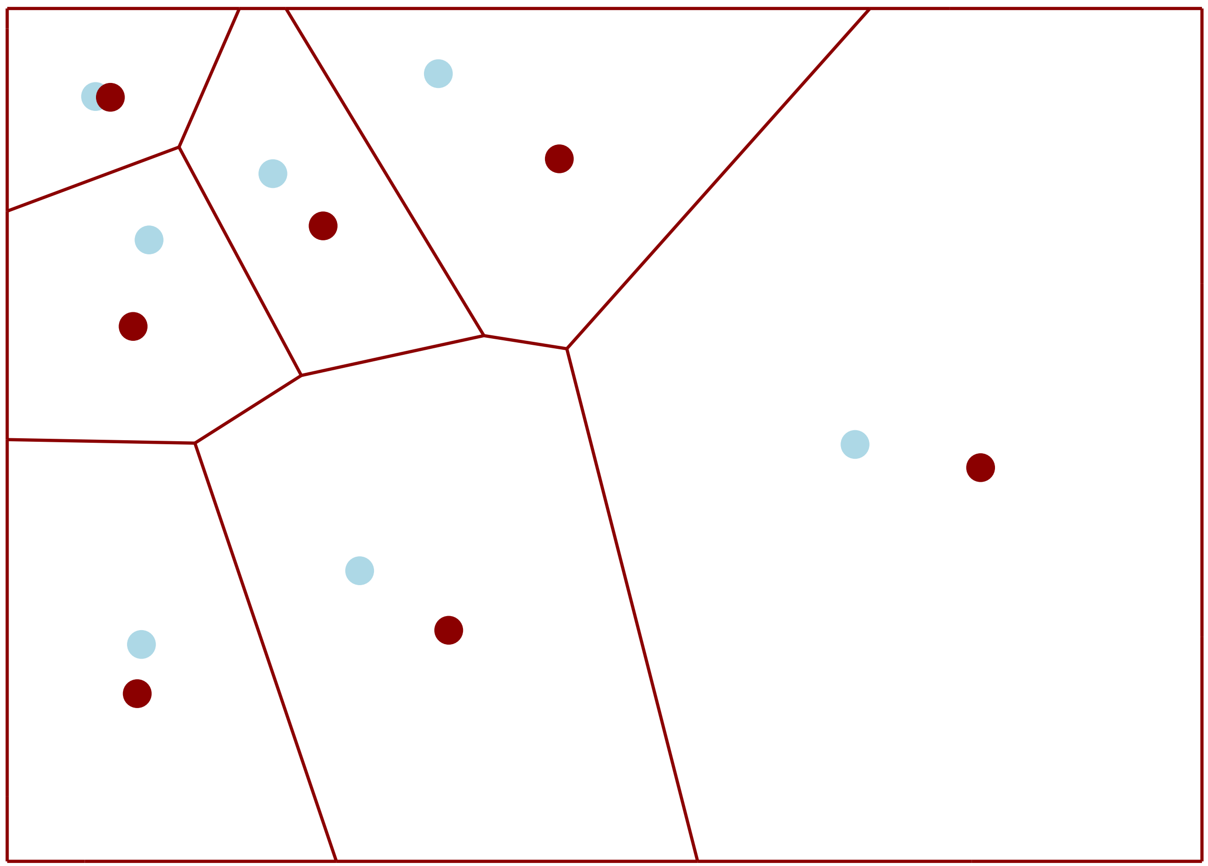
Meshing

LLoyd iteration

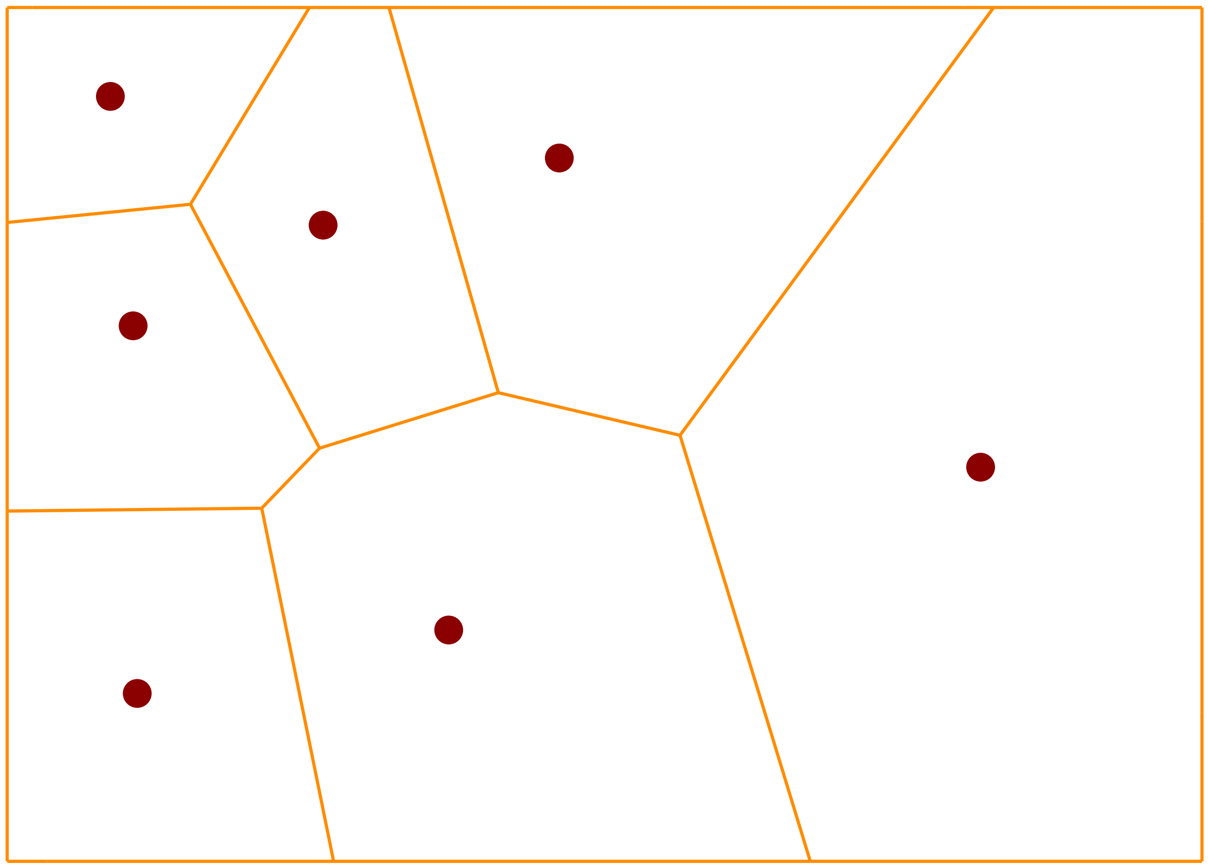


Meshing

LLoyd iteration



LLoyd iteration

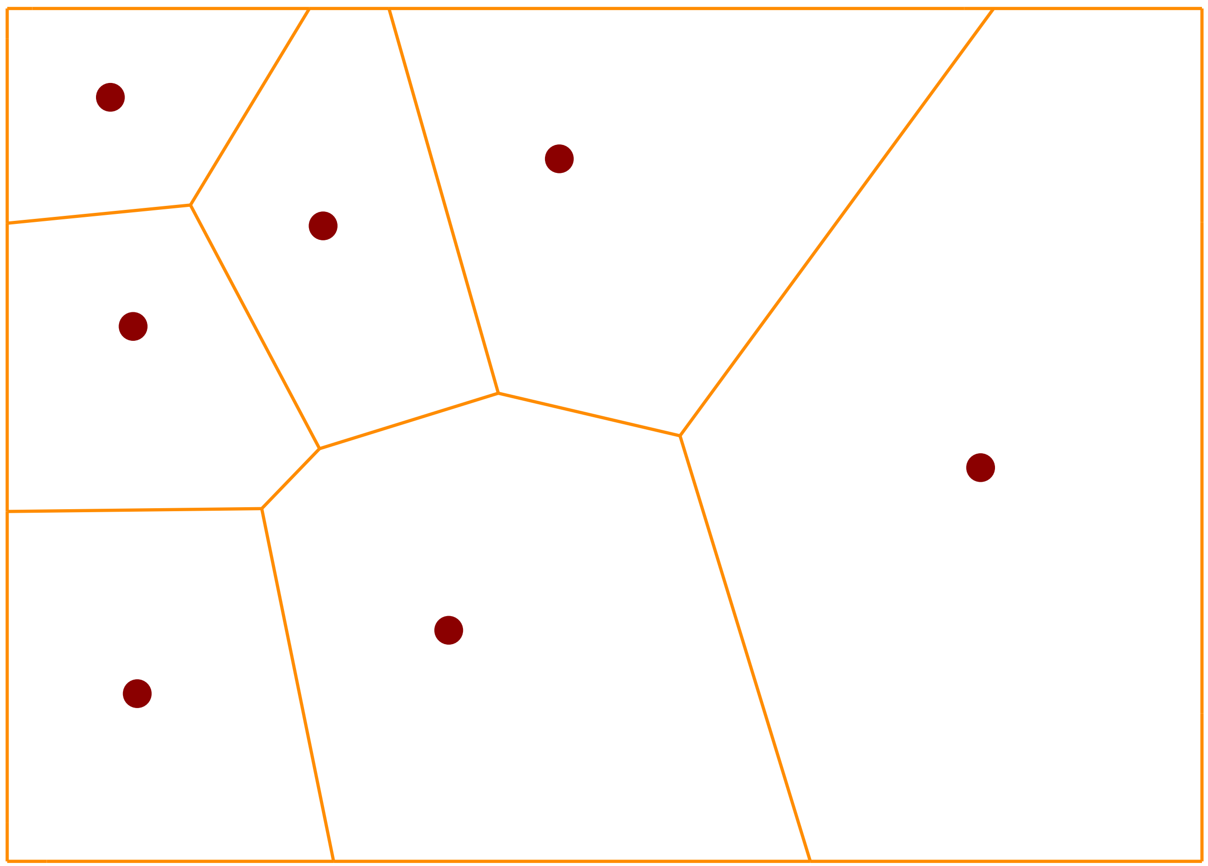


Meshing

Delaunay mesh optimization

LLoyd iteration

Reach a nice point distribution



Meshing

Alternate

Delaunay mesh refinement

Lloyd smooting   or different kind of smoothing

Meshing

Delaunay mesh optimization

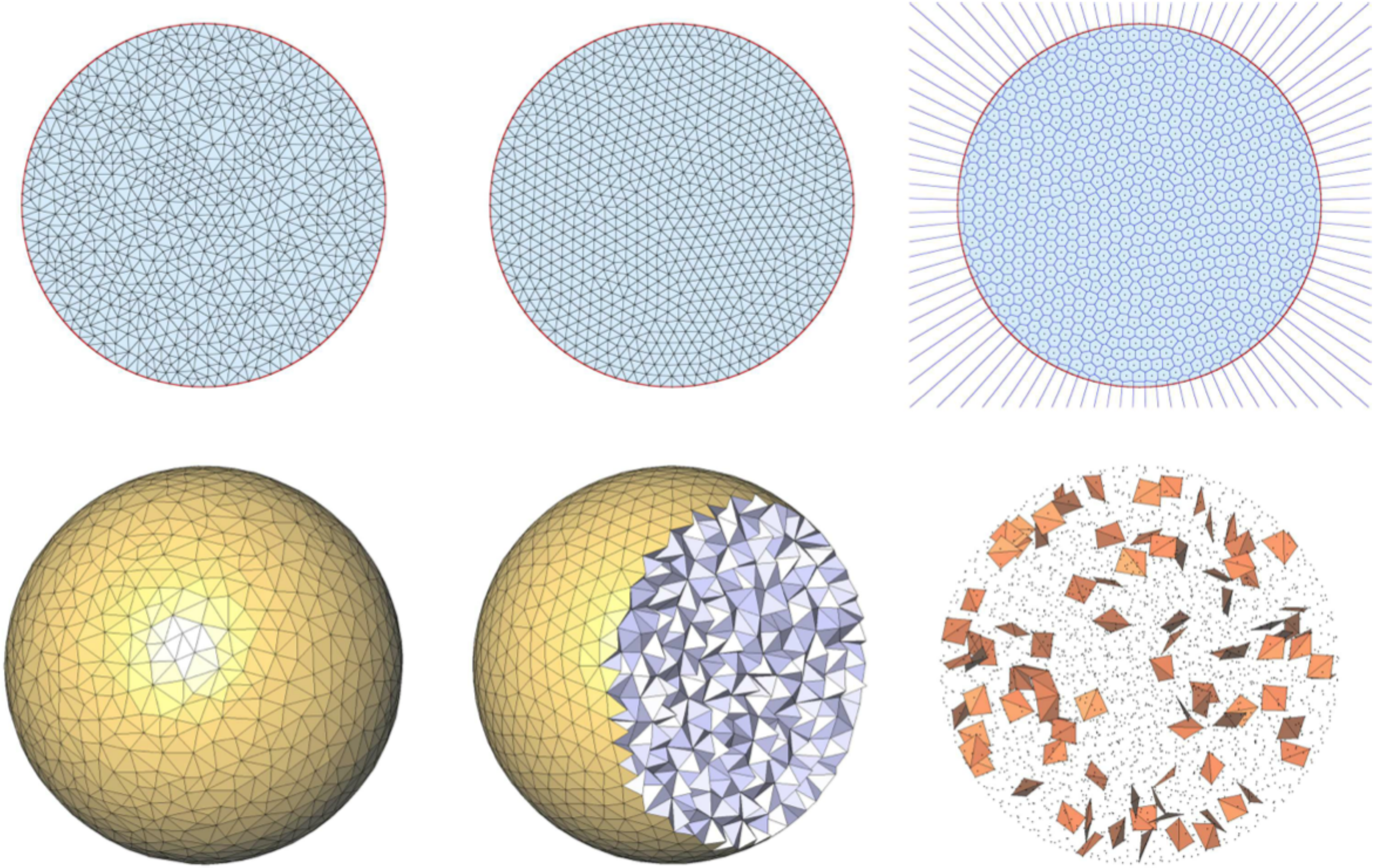
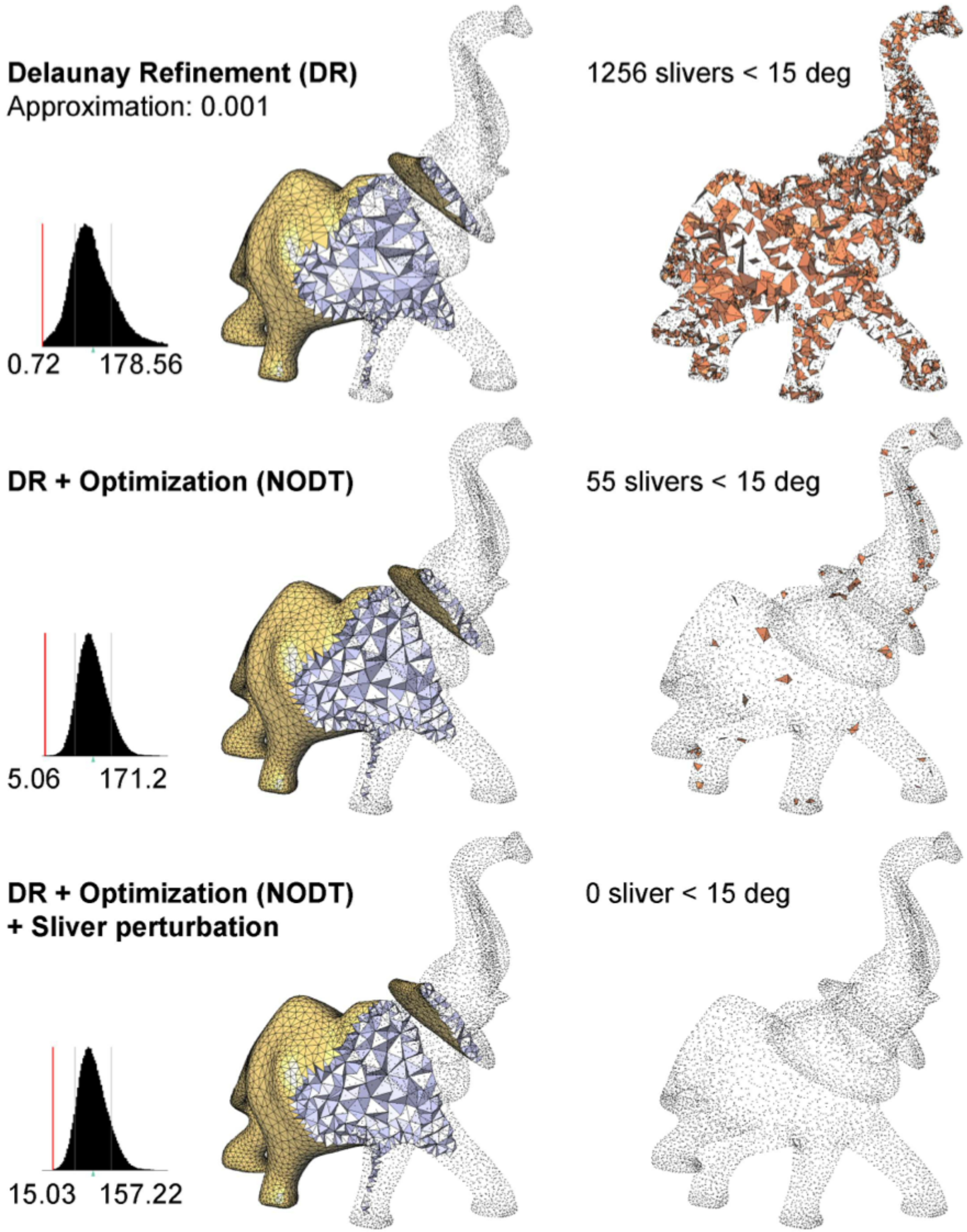


Figure 1.6: CVT mesh optimization. In 2D (top), (left) a 2D Delaunay mesh M_2 generated by Delaunay refinement, (center) M_2 optimized with CVT, and (right) M_2's Voronoi diagram. In 3D (bottom), (left) a 3D Delaunay mesh M_3 generated by Delaunay refinement, (center) M_3 optimized with CVT, and (right) M_3's slivers (tetrahedra with dihedral angles smaller than 5°).

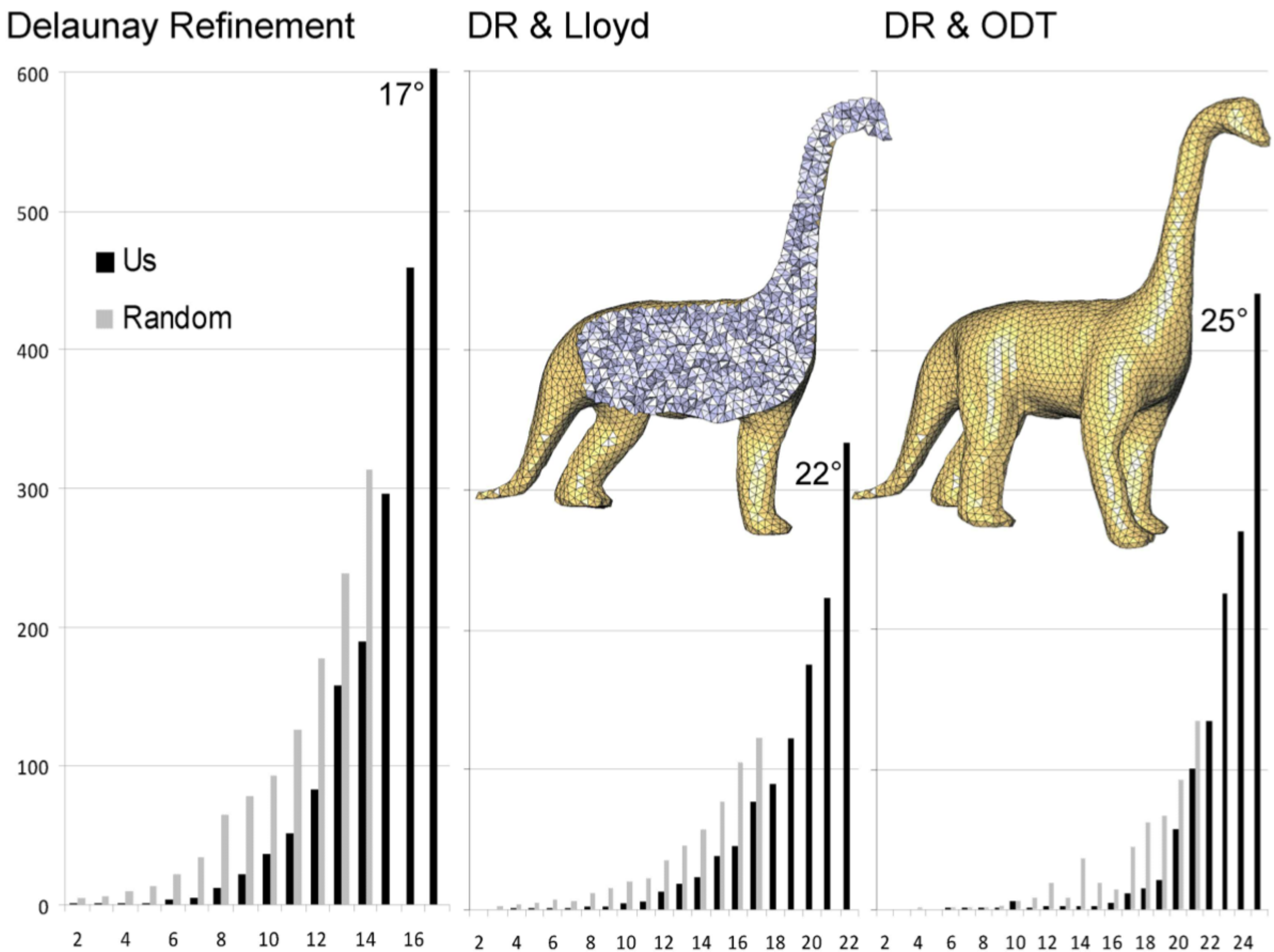
Meshing

Delaunay mesh optimization



Meshing

Delaunay mesh optimization

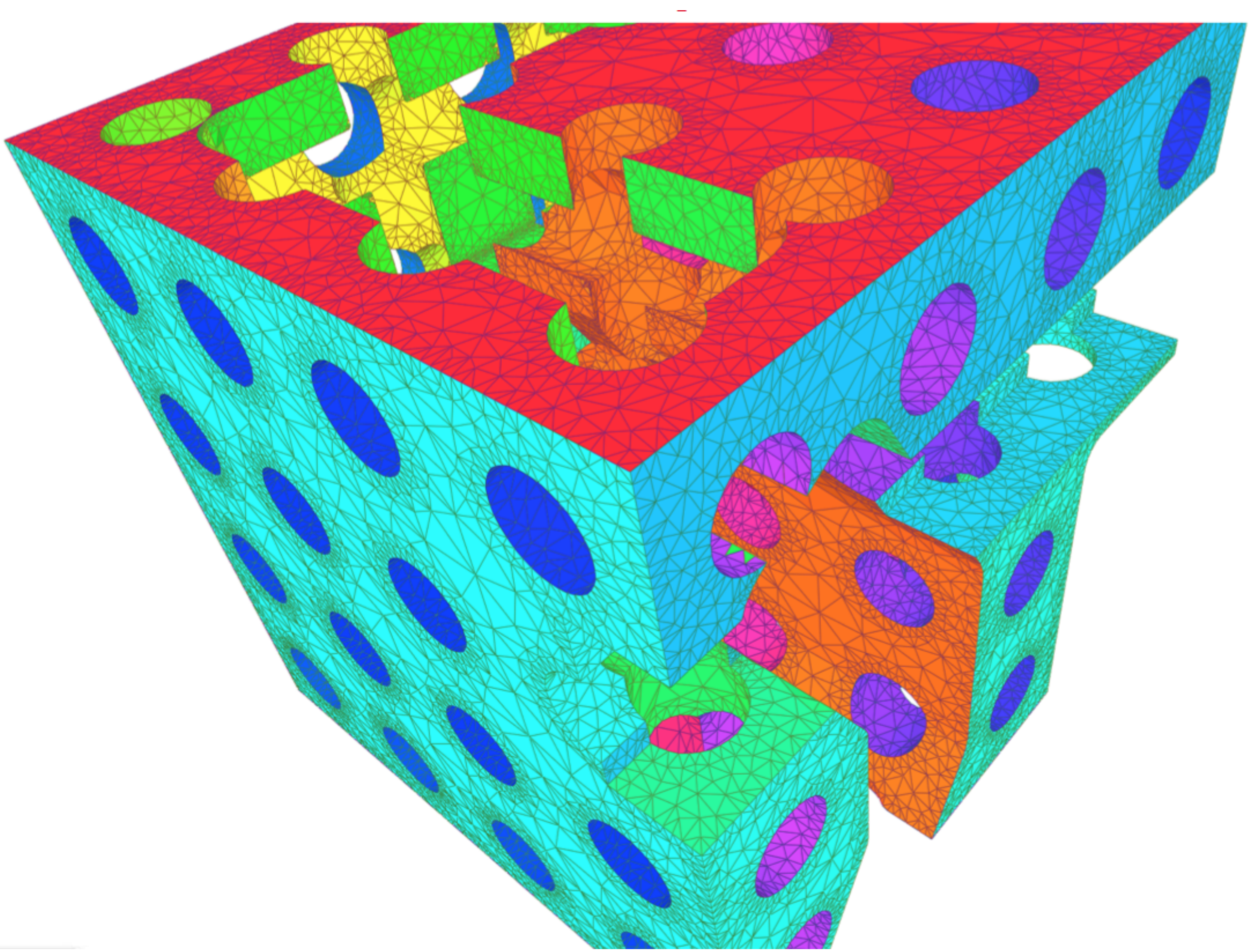


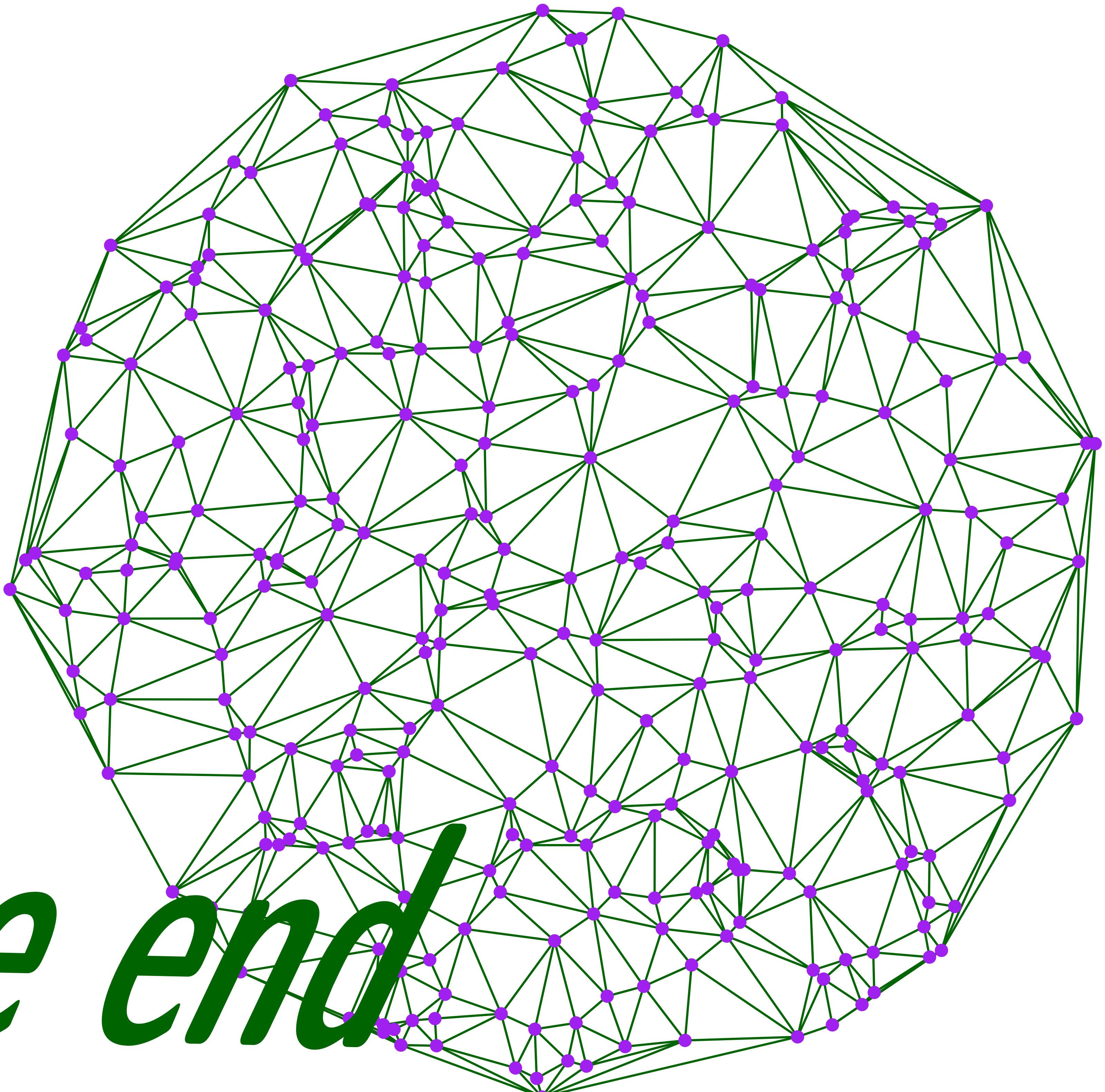
Meshing

3D

Constraints: edges and faces

Point to insert may be encroached by edges or faces





The end