

CS164: Curve and Surface Reconstruction from Samples



Leonidas Guibas
Computer Science Dept.
Stanford University



[Many slides: thanks Mario Botsch, Nina Amenta]

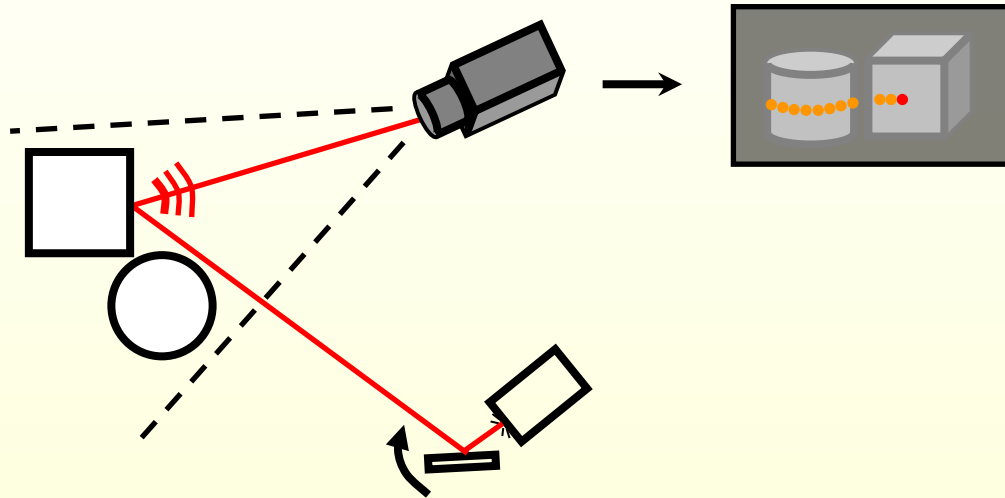
Surface Samples: Inputs and Applications

- Laser range data
- Hand digitizers
- Computer vision
- Medical imaging
- Entertainment
- Industrial design
- Reverse engineering
- Paleontology
- Archeology
- Medical devices

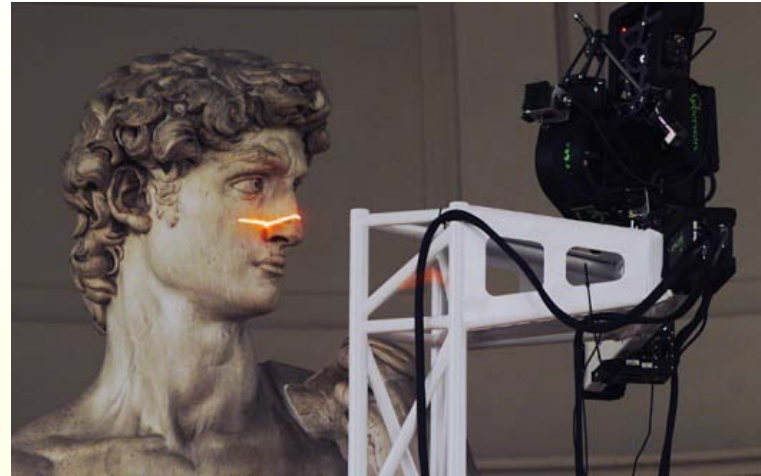
Important topic in computational geometry.

Range Scanning Systems

● Active: Laser Scanning

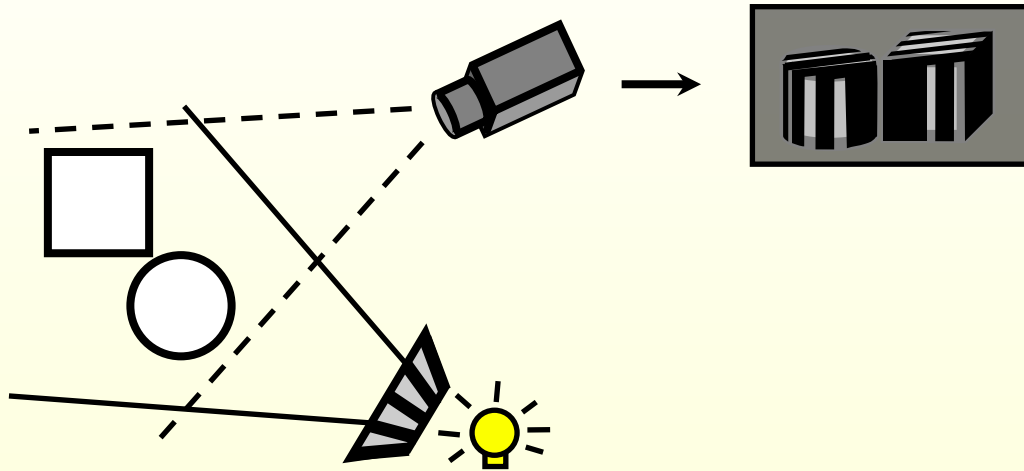


Acquisition Technologies



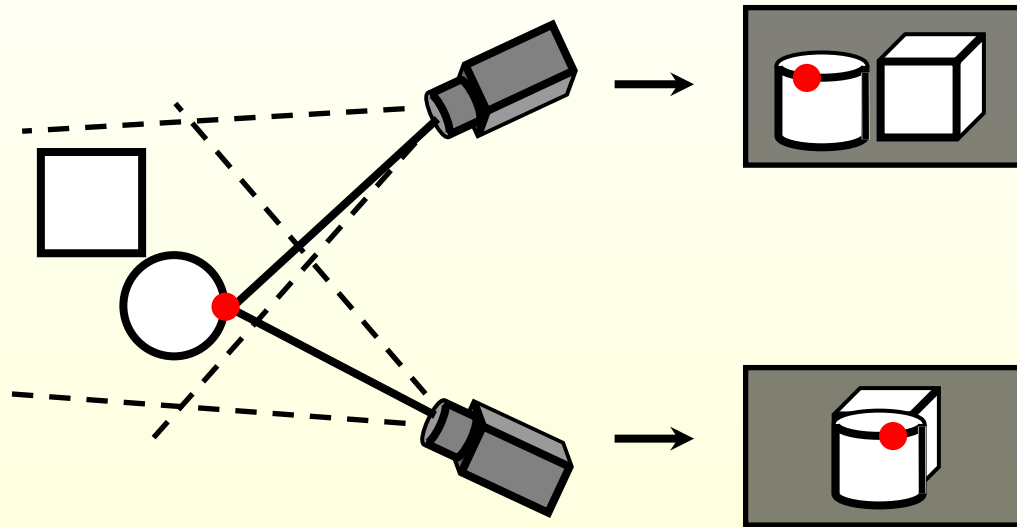
Range Scanning Systems

● **Active:** Structured Light Acquisition



Range Scanning Systems

● **Passive:** Stereo Matching



Range Scanning

- Active systems are superior in data quality
- Multiple scans required for complex objects
 - Scan path planning
 - Scan registration
- Scans are incomplete and noisy
 - Model repair, hole filling
 - Smoothing for noise removal

From Scans to Models

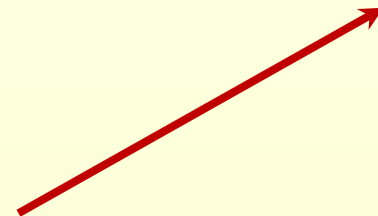


set of raw partial scans

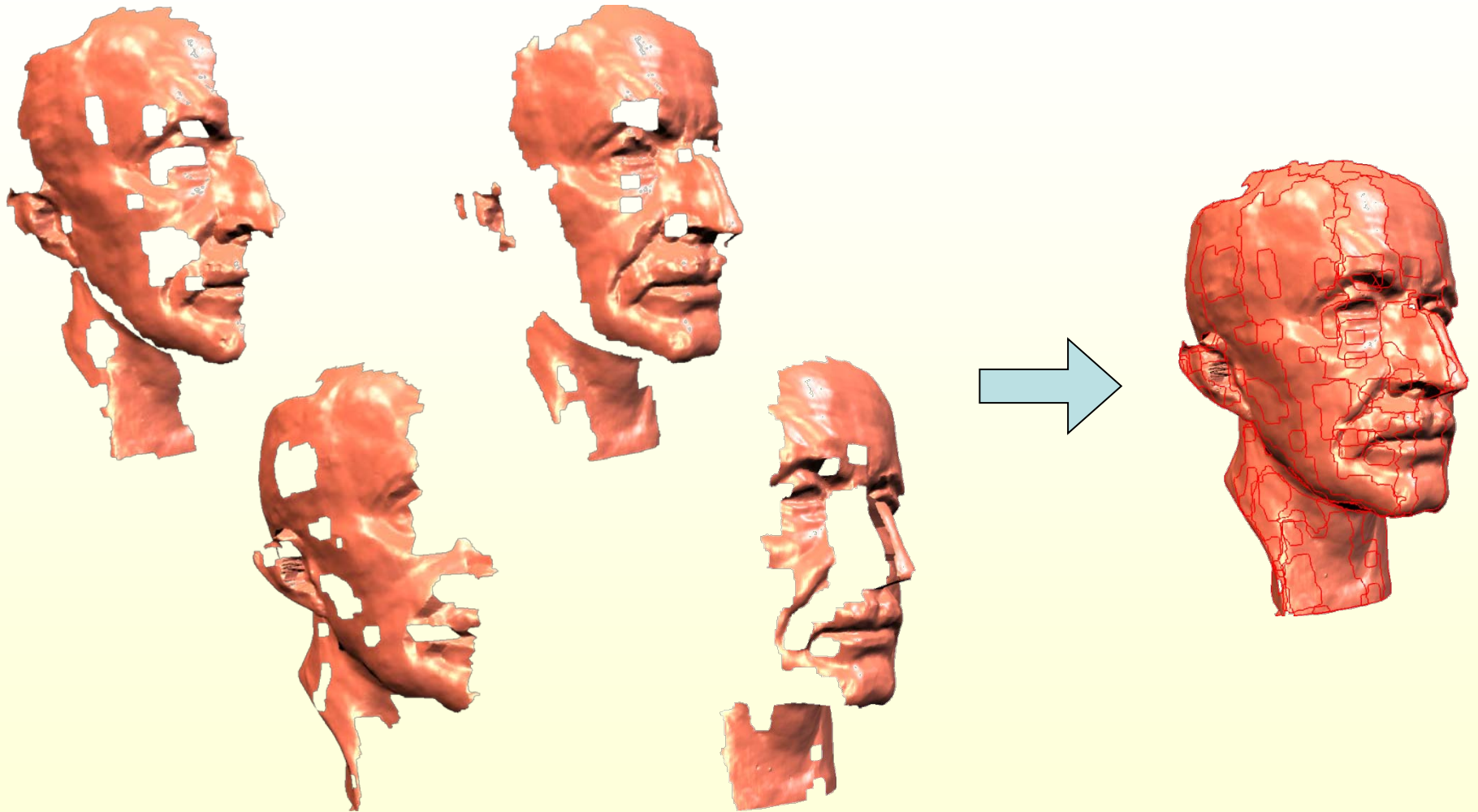
reconstructed model



scan registration

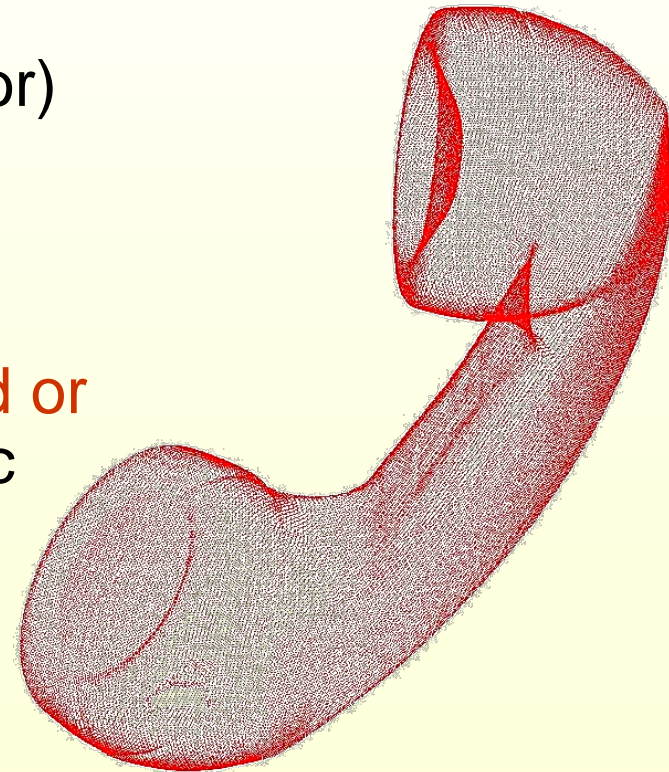


Scan Registration

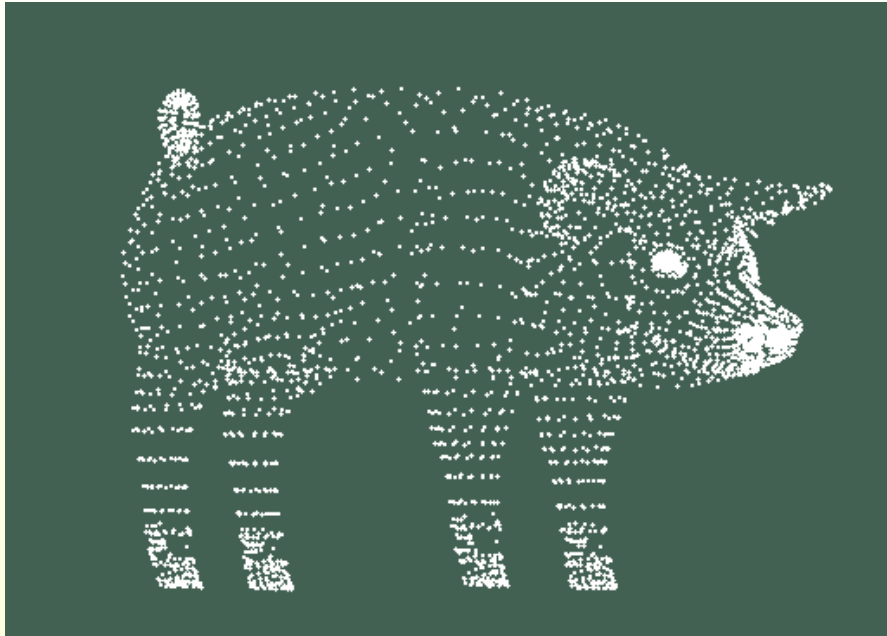


Point Cloud Data (PCD) Challenges

- PCD = “point cloud data”
 - unorganized collection of points sampled from the surface (or interior) of an object, with noise added
 - typical output of a 3-D scanning process
- **no connectivity information or manifold or mesh structure** □ hard to use geometric methods directly
- **no regular sampling**
□ hard to use signal processing tools



Reconstruction from Point Samples



Input: *Samples* taken from object surface



Output: Well-formed polygonal model

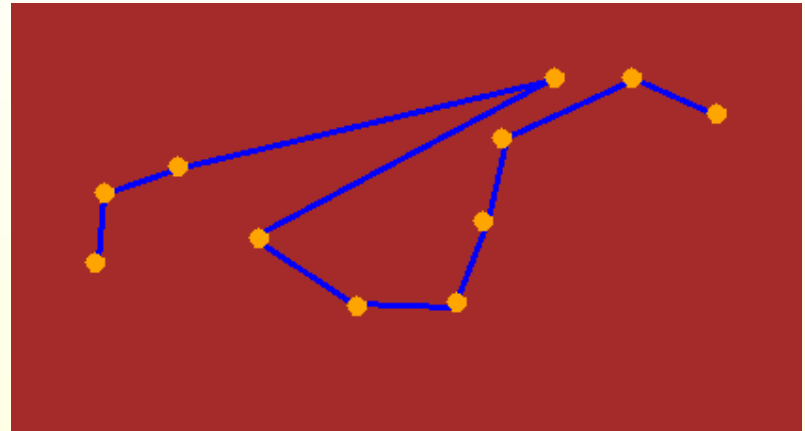
Correctness of Reconstruction

- Is it the boundary that of a solid (watertight)?
- Is it geometrically close to original surface?
- Is it topologically homeomorphic to original surface?



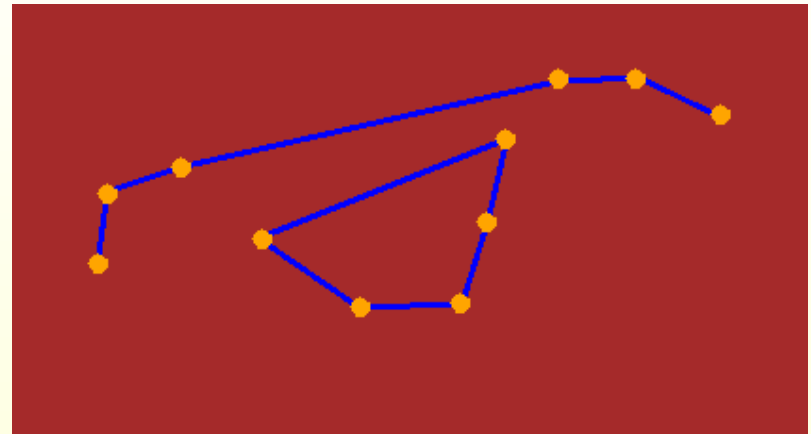
Correctness of Reconstruction

- Is it the boundary of a solid (watertight)?
- Is it geometrically close to original surface?
- Is it topologically homeomorphic to original surface?



Correctness of Reconstruction

- Is it the boundary of a solid (watertight)?
- Is it geometrically close to original surface?
- Is it topologically homeomorphic to original surface?



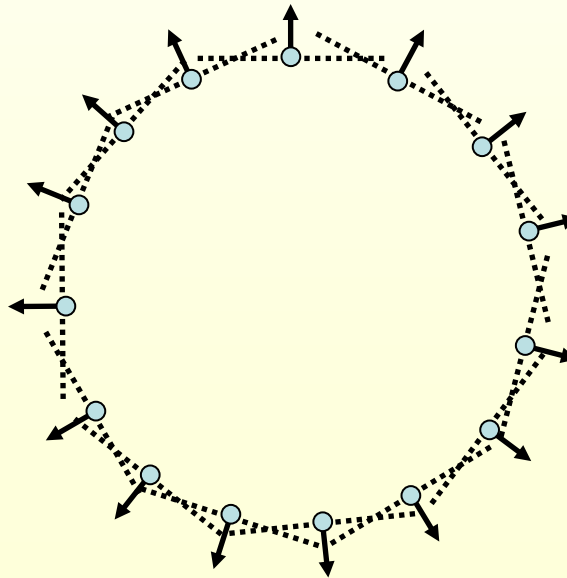
Many Reconstruction Methods

- **Implicit surface methods**
 - Voxel-based methods and level sets
 - Basis functions (e.g., radial)
 - Poisson indicator functions
 - Moving least squares (MLS)
- **Voronoi-based combinatorial methods**
 - Crust
 - Cocone
 - Wrap

Implicit Function Methods

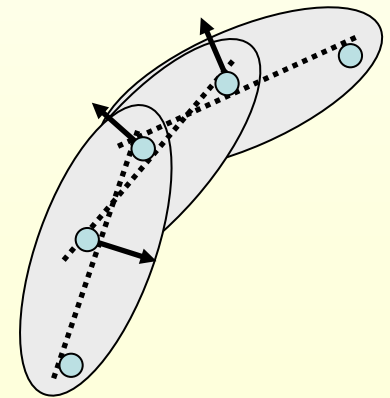
Signed Distance Function (SDF)

- Construct SDF from point samples
 - Distance to points is not enough
 - Need inside/outside information
 - Reconstruct normal vectors first



Normal Estimation

- Find normal \mathbf{n}_i for each sample point \mathbf{p}_i
 1. Examine local neighborhood for each point
 - Set of k nearest neighbors
 2. Compute best approximating tangent plane
 - Covariance analysis
 3. Determine normal orientation
 - MST propagation



Normal Estimation

• Find normal \mathbf{n}_i for each sample point \mathbf{p}_i

1. Examine local neighborhood for each point

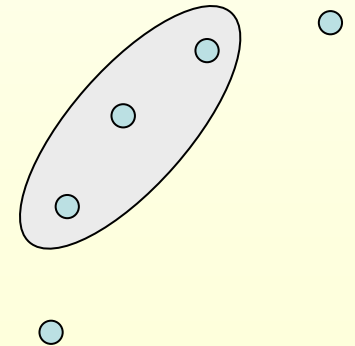
• Set of k nearest neighbors

2. Compute best approximating tangent plane

• Covariance analysis

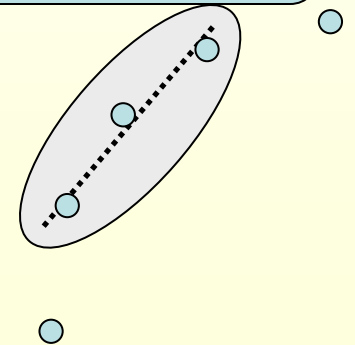
3. Determine normal orientation

• MST propagation



Normal Estimation

- Find normal \mathbf{n}_i for each sample point \mathbf{p}_i
 1. Examine local neighborhood for each point
 - Set of k nearest neighbors
 2. Compute best approximating tangent plane
 - Covariance analysis
 3. Determine normal orientation
 - MST propagation



Principal Component Analysis

- Fit a plane with center \mathbf{c} and normal \mathbf{n} to a set of points $\{\mathbf{p}_1, \dots, \mathbf{p}_m\}$

- Minimize least squares error

$$\min_{\mathbf{c}, \mathbf{n}} \sum_{j=1}^m (\mathbf{n}^T (\mathbf{p}_j - \mathbf{c}))^2$$

- Subject to non-linear constraint

$$\|\mathbf{n}\| = 1$$

Principal Component Analysis

- Plane center is barycenter of points

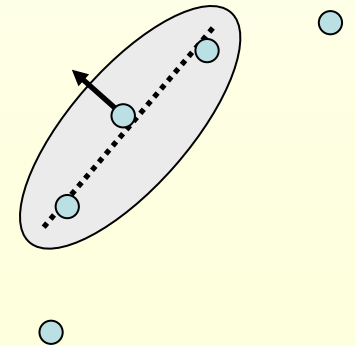
$$\mathbf{c} = \bar{\mathbf{p}} =: \frac{1}{m} \sum_{j=1}^m \mathbf{p}_j$$

- Normal is eigenvector corresponding to the smallest eigenvalue of the point covariance matrix

$$\sum_{j=1}^m (\mathbf{p}_j - \bar{\mathbf{p}}) (\mathbf{p}_j - \bar{\mathbf{p}})^T \in \mathbb{R}^{3 \times 3}$$

Normal Estimation

- Find normal \mathbf{n}_i for each sample point \mathbf{p}_i
 1. Examine local neighborhood for each point
 - Set of k nearest neighbors
 2. Compute best approximating tangent plane
 - Covariance analysis
 3. Determine normal orientation
 - MST propagation

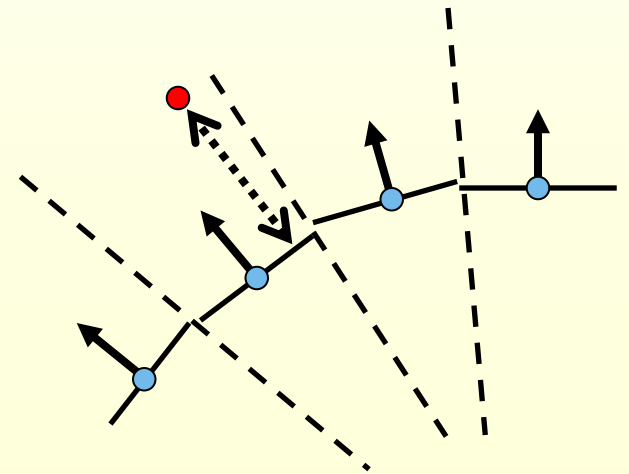


Normal Orientation

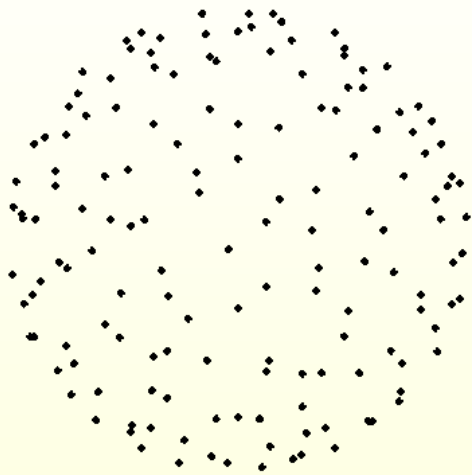
- Build neighborhood graph connecting nearby points
 - Edge (ij) exists if $\mathbf{p}_i \in \text{kNN}(\mathbf{p}_j)$ or $\mathbf{p}_j \in \text{kNN}(\mathbf{p}_i)$
- Propagate normal orientation through graph
 - For neighbors $\mathbf{p}_i, \mathbf{p}_j$: Flip \mathbf{n}_j if $\mathbf{n}_i^T \mathbf{n}_j < 0$
 - Fails at sharp edges/corners
- Propagate along “safe” paths (parallel normals)
 - Minimum spanning tree with angle-based edge weights $w_{ij} = 1 - |\mathbf{n}_i^T \mathbf{n}_j|$

SDF from Point & Normals

- Distance from tangent planes [Hoppe 92]
 - Points + normals determine local tangent planes
 - Use distance from closest point's tangent plane
 - Linear approximation in Voronoi cell
 - Simple and efficient, but SDF is only C^0



Hoppe '92 Reconstruction



150 samples

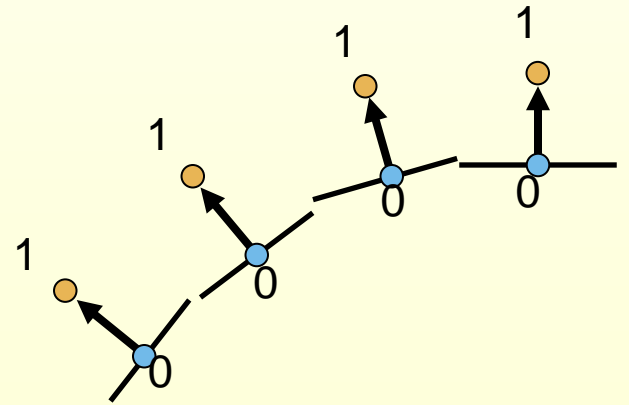


reconstruction
on 50^3 grid

Smooth SDF Approximation

Scattered data interpolation problem

- On-surface constraints $\text{dist}(\mathbf{p}_i) = 0$
- Avoid trivial solution $\text{dist} \equiv 0$
- Off-surface constraints $\text{dist}(\mathbf{p}_i + \mathbf{n}_i) = 1$



Smooth SDF Approximation

• Scattered data interpolation problem

- On-surface constraints $\text{dist}(\mathbf{p}_i) = 0$
- Avoid trivial solution $\text{dist} \equiv 0$
- Off-surface constraints $\text{dist}(\mathbf{p}_i + \mathbf{n}_i) = 1$

• Radial basis functions

- Well suited for smooth interpolation
- Sum of shifted, weighted kernel functions

$$\text{dist}(\mathbf{x}) = \sum_i w_i \cdot \varphi(\|\mathbf{x} - \mathbf{c}_i\|) = \sum_i w_i \cdot \varphi_i(\mathbf{x})$$

RBF Interpolation

- Interpolate on- and off-surface constraints

$$\text{dist}(\mathbf{x}_j) = \sum_{i=1}^n w_i \cdot \varphi(\|\mathbf{x}_j - \mathbf{c}_i\|) \stackrel{!}{=} d_j, \quad j = 1, \dots, n$$

- Choose centers \mathbf{c}_i as constrained points \mathbf{x}_i
- Solve symmetric linear system for weights

w_i

$$\begin{pmatrix} \varphi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \cdots & \varphi(\|\mathbf{x}_1 - \mathbf{x}_n\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|\mathbf{x}_n - \mathbf{x}_1\|) & \cdots & \varphi(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix}$$

RBF Basis Functions

• Wendland basis functions

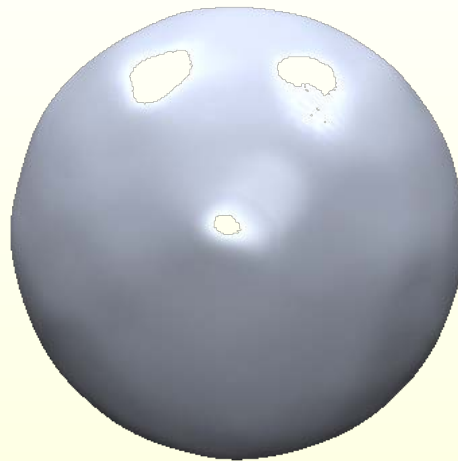
$$\varphi(r) = \left(1 - \frac{r}{\sigma}\right)_+^4 \left(4\frac{r}{\sigma} + 1\right)$$

- Compactly supported in $[0, \sigma]$
- Leads to sparse, symm. pos. def. linear system
- C^2 smoothness
- But not necessarily fair
- Not suited for highly irregular sampling

Comparison



Hoppe '92



Compact RBF
Wendland C^2

RBF Basis Functions

• Triharmonic basis functions

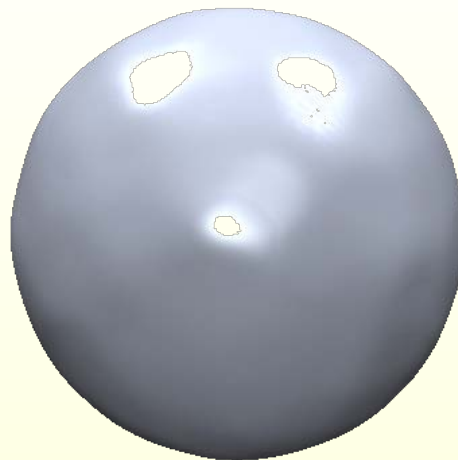
$$\phi(r) = r^3$$

- Globally supported
- Leads to dense symm. linear system
- C^2 smoothness
- Provably optimal fairness
- Works well for highly irregular sampling

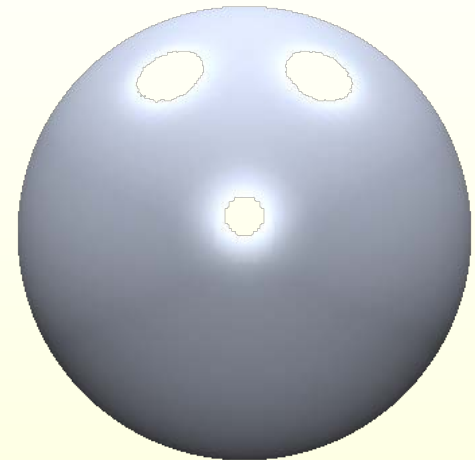
Comparison



Hoppe '92



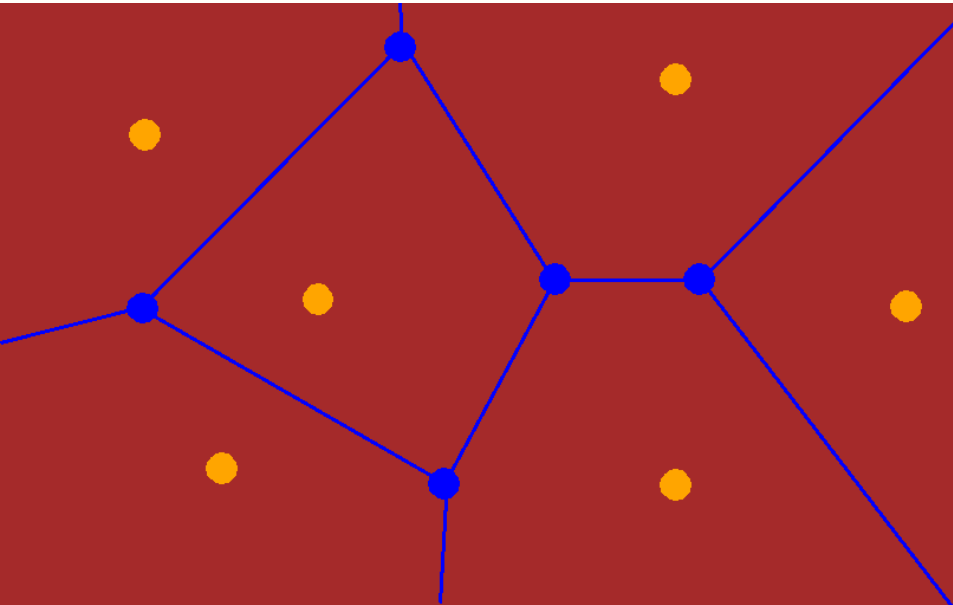
Compact RBF
Wendland C^2



Global RBF
Triharmonic

Voronoi Reconstruction

Tools - Voronoi Diagram



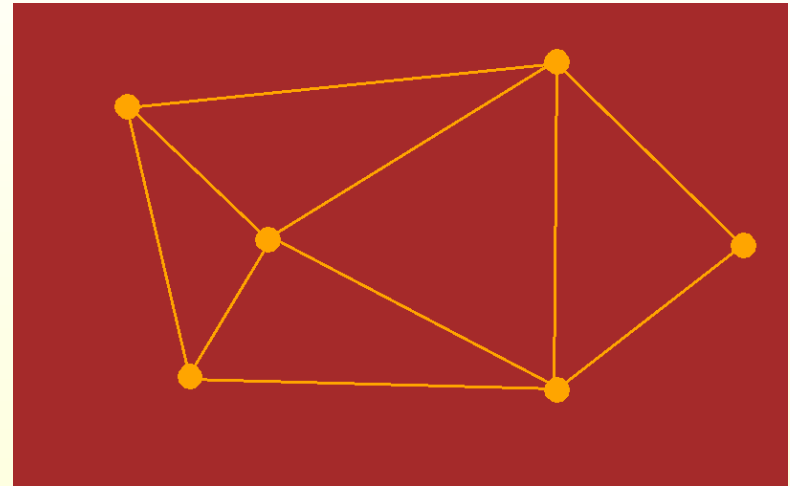
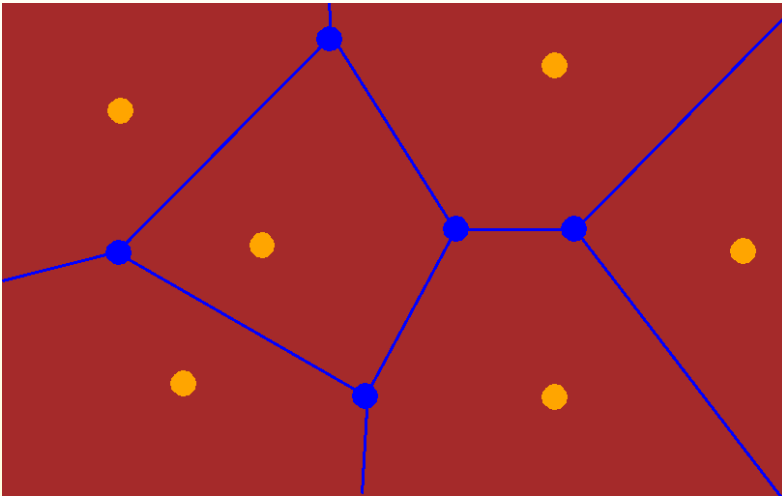
Points closest to each sample form cells

Cell boundaries have more than one closest sample

Adjacent cells connect adjacent samples

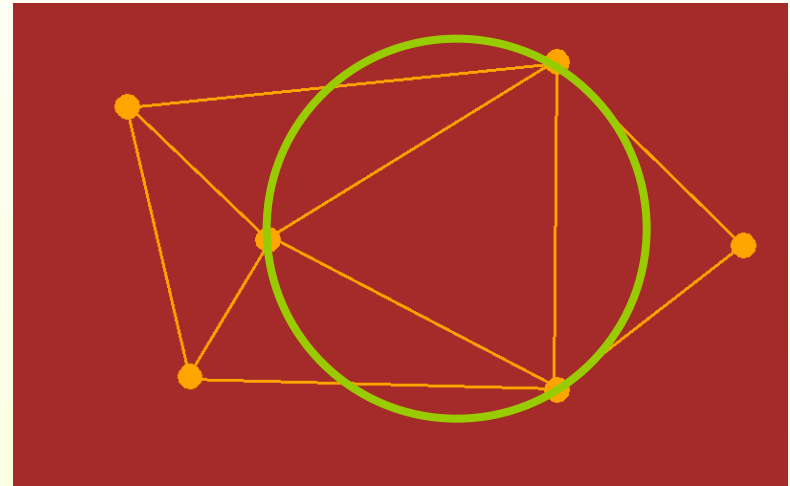
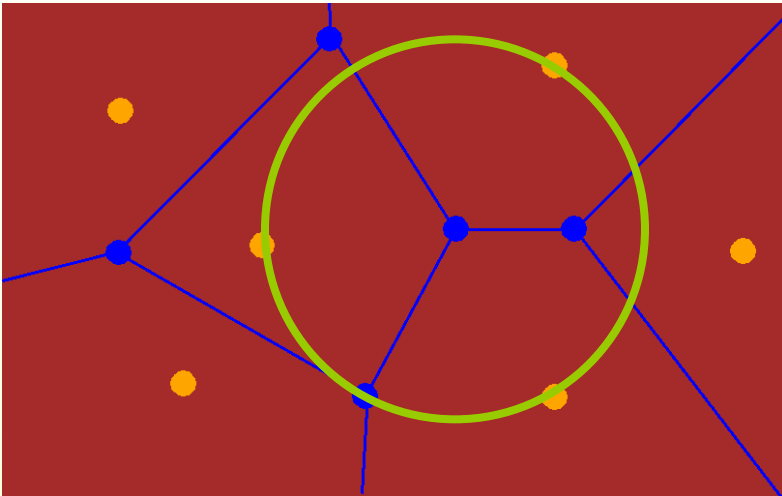
Delaunay Triangulation

Delaunay edges connect adjacent samples



Delaunay Triangulation

Delaunay edges connect adjacent samples



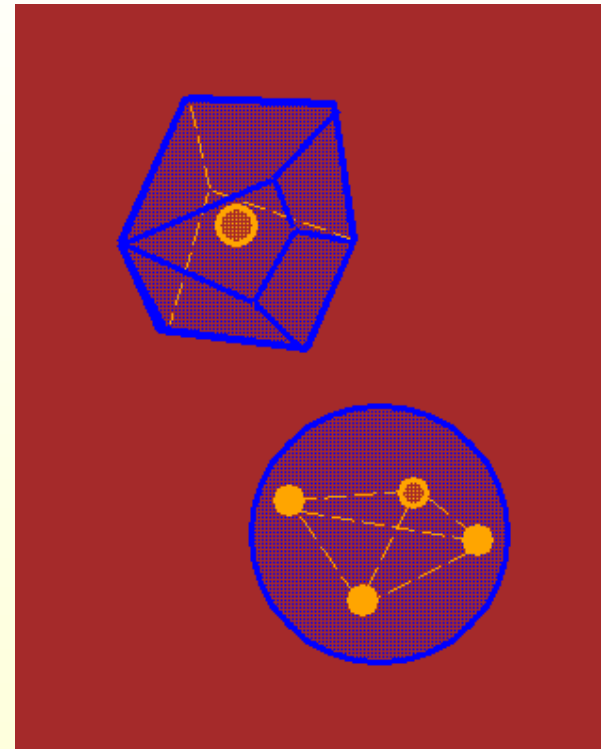
Empty Voronoi balls centered at Voronoi vertices pass through closest samples

3D Voronoi/Delaunay

Voronoi cells are convex polyhedra

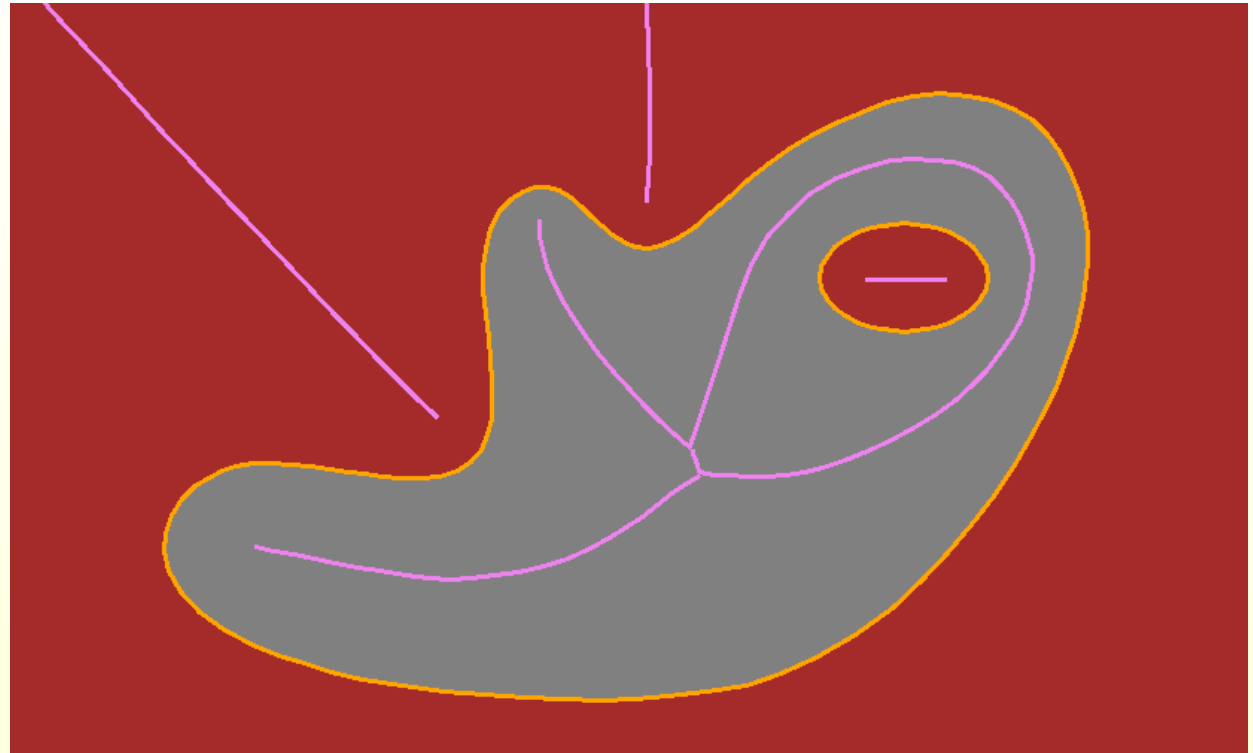
Voronoi balls pass through 4 samples

Delaunay tetrahedra



Another Concept: Medial Axis

Think of object surface as infinite set of samples



Medial axis is set of points with more than one closest sample

In 3D As Well



2-D medial axis

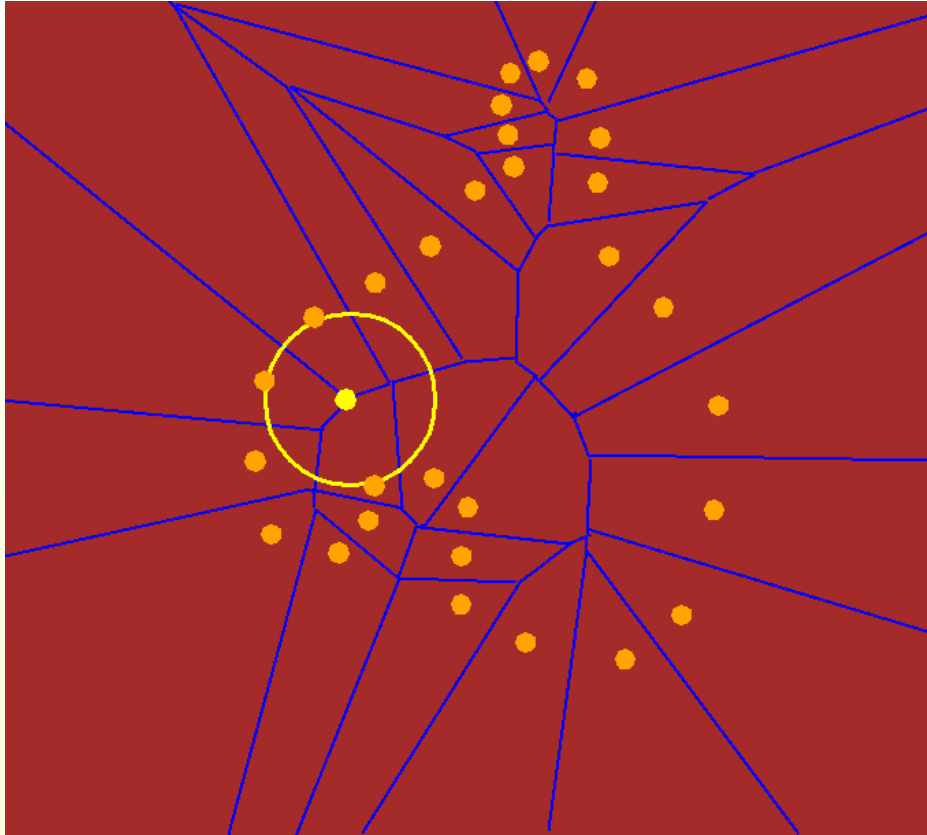
Medial Axis



Maximal ball avoiding surface is a
medial ball

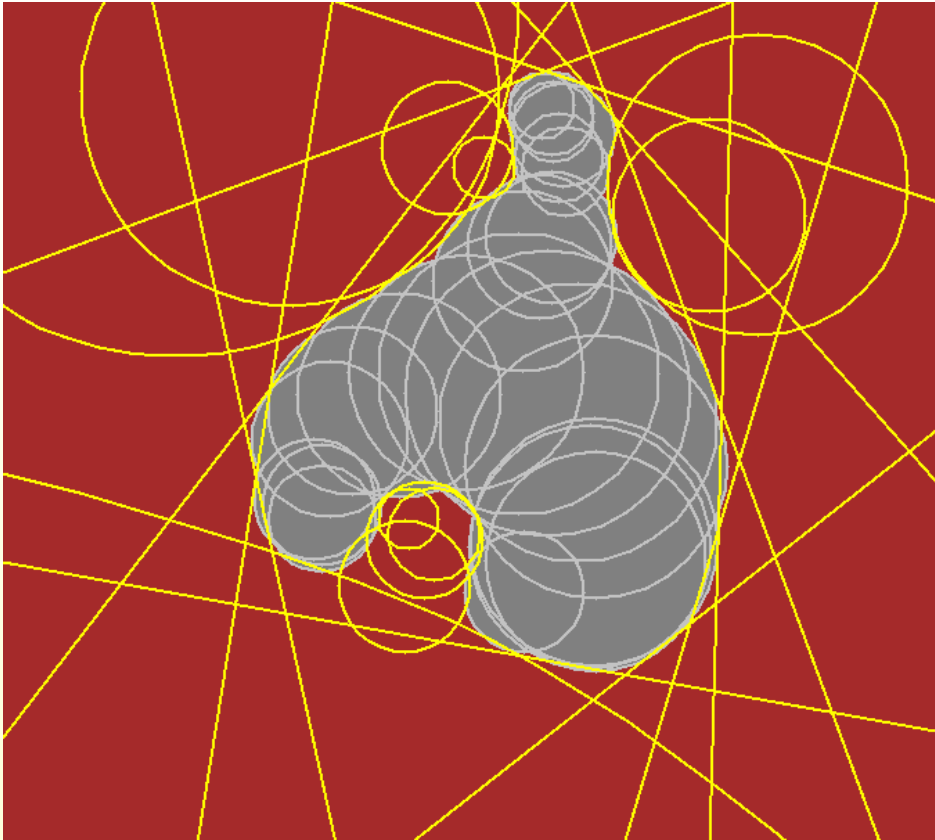
Every solid is a union of balls !

2D Medial Axis Approximation



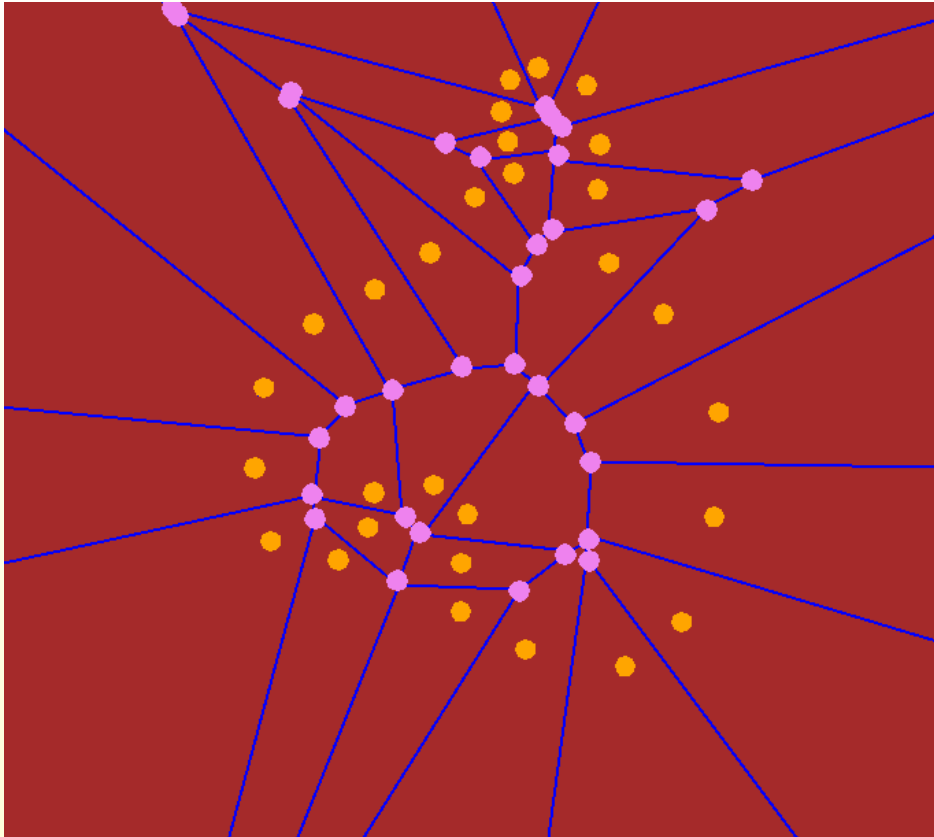
Voronoi balls approximate medial balls

2D Medial Axis Approximation



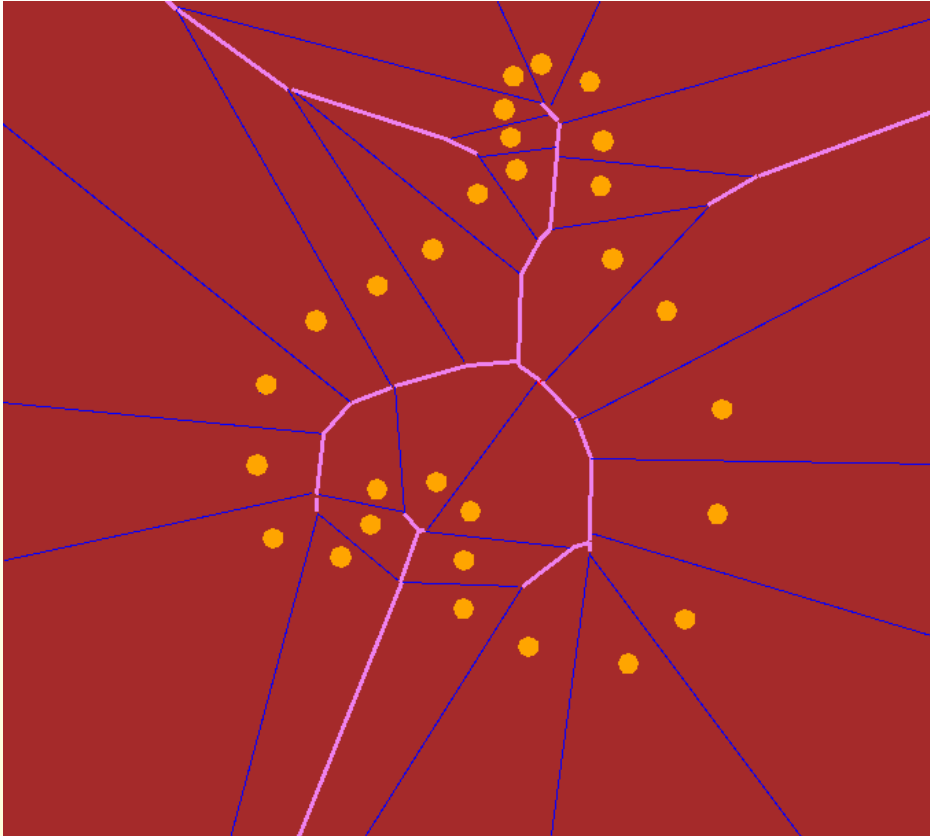
The Voronoi balls approximate the object and its complement

2D Medial Axis Approximation



The set of Voronoi vertices
approximates the medial axis

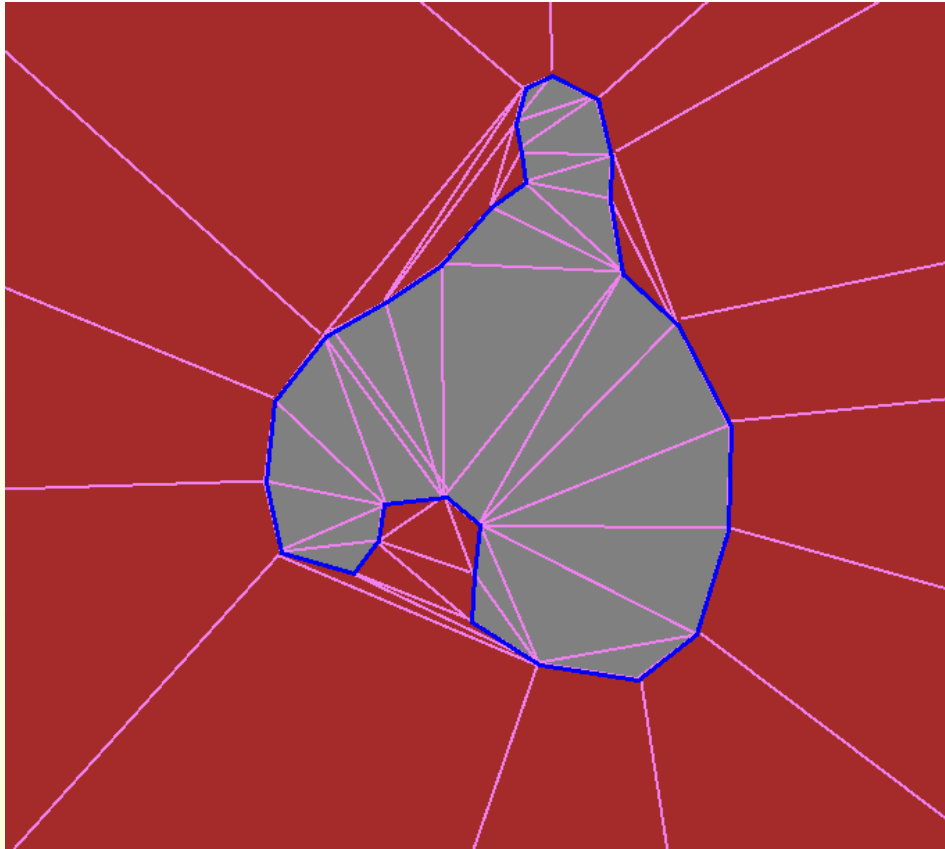
2D Curve Reconstruction



Voronoi edges can be divided into **pink**, approximating the medial axis and **blue**, separating adjacent samples



2D Curve Reconstruction

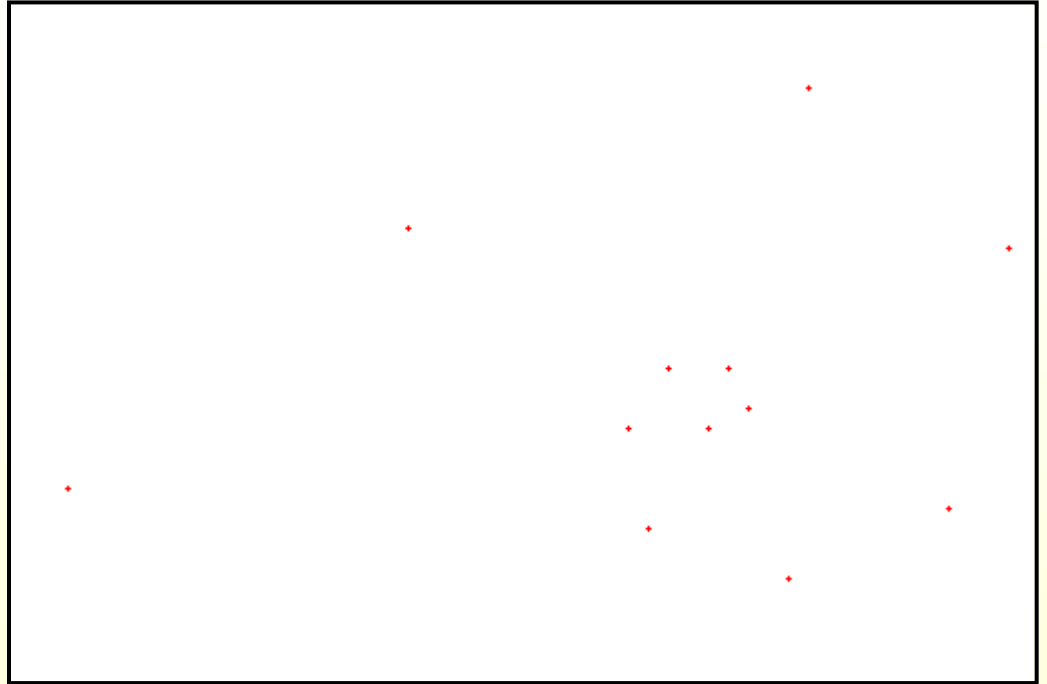
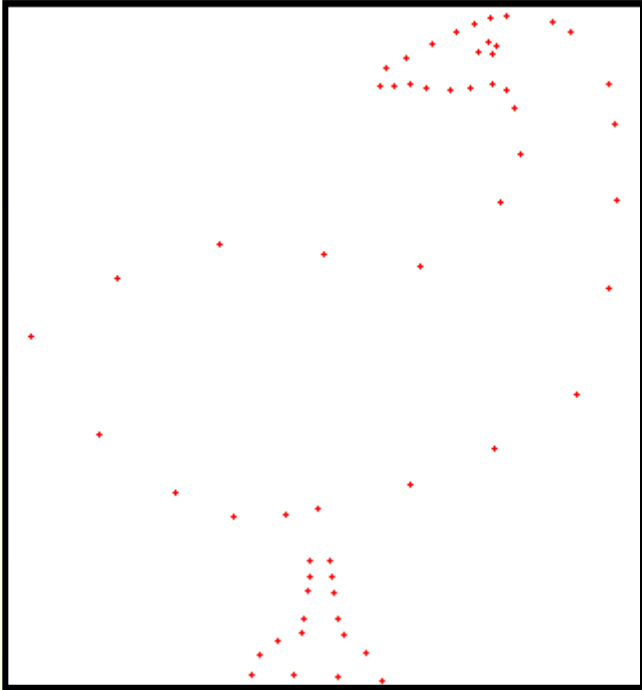


In the dual picture, **blue** Delaunay edges reconstruct the curve

Many algorithms, with proofs, for coloring edges

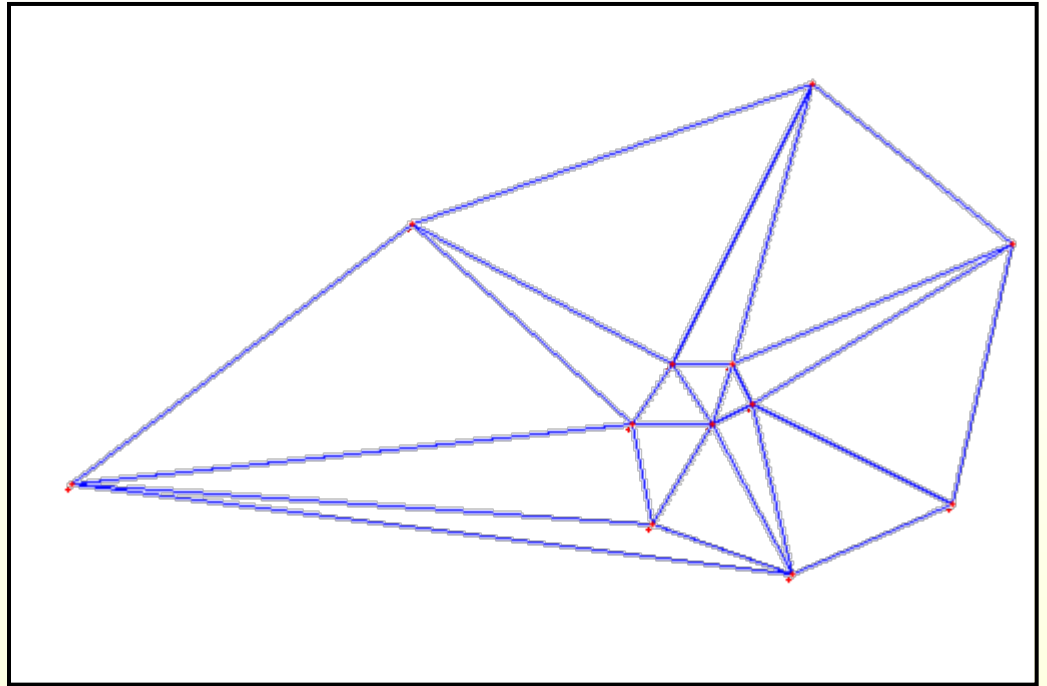
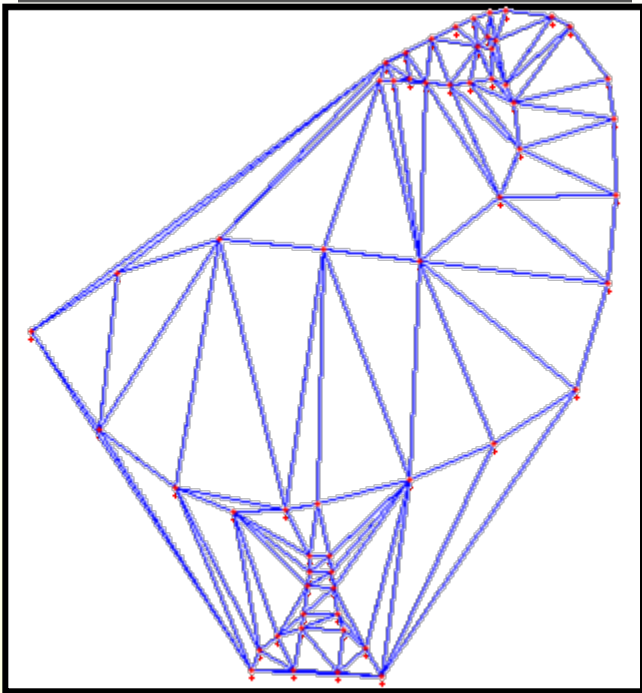
Not trivial to generalize to 3D !!

Curve From Points – Connect The Dots



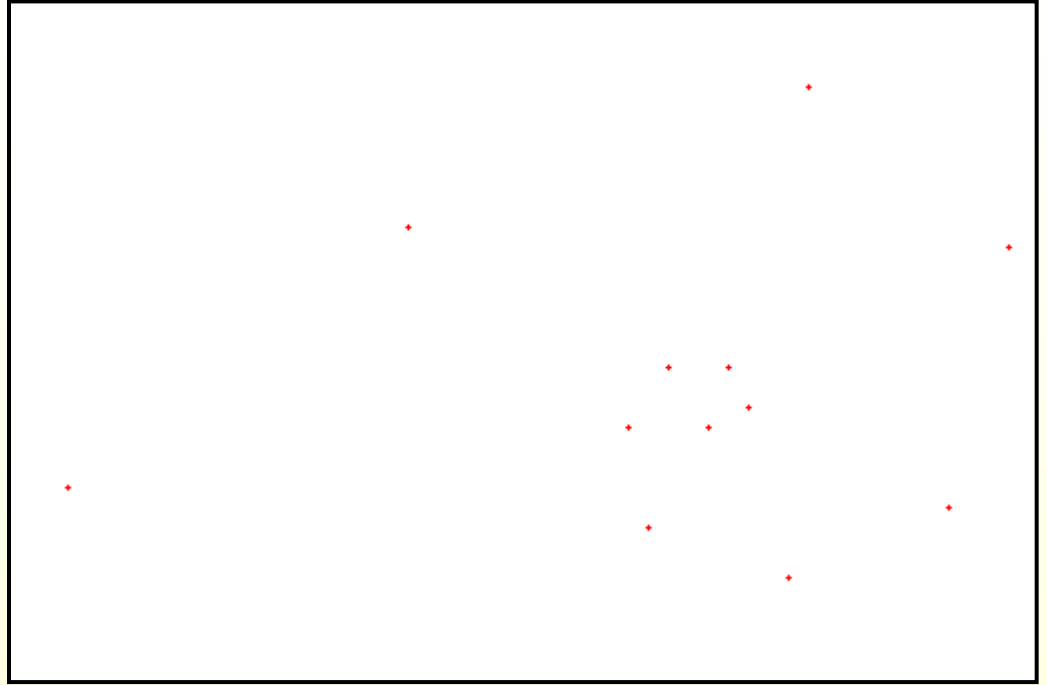
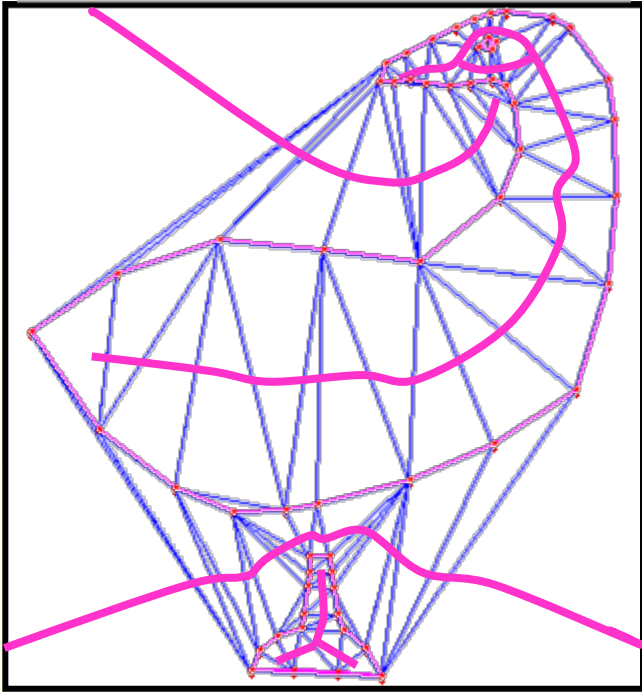
● How to find “correct” connectivity?

Use Delaunay to Connect



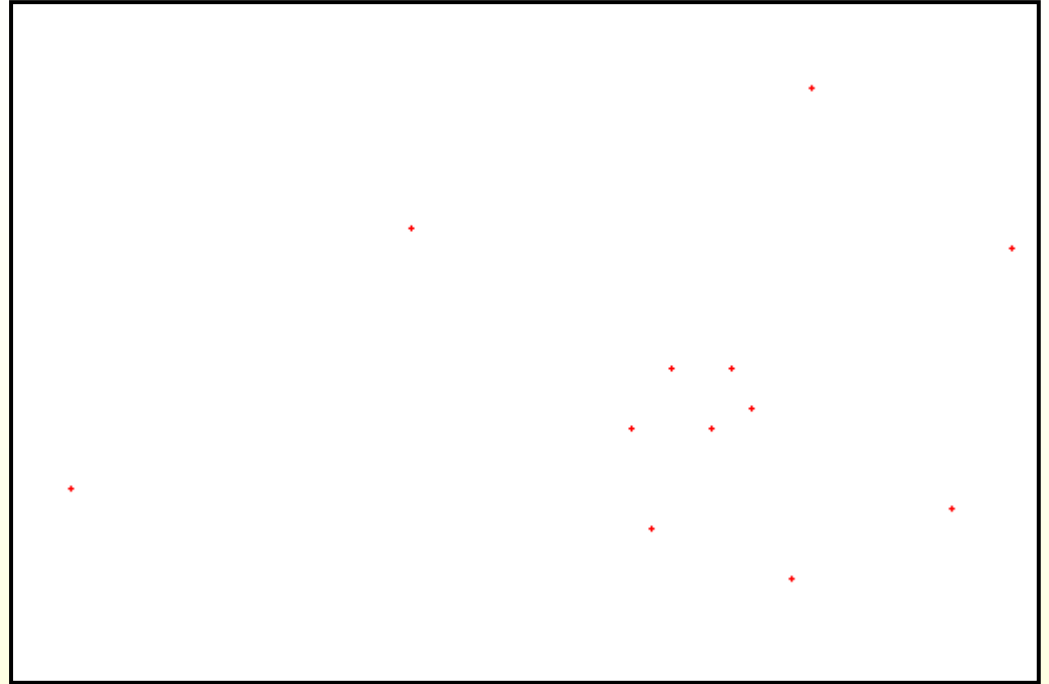
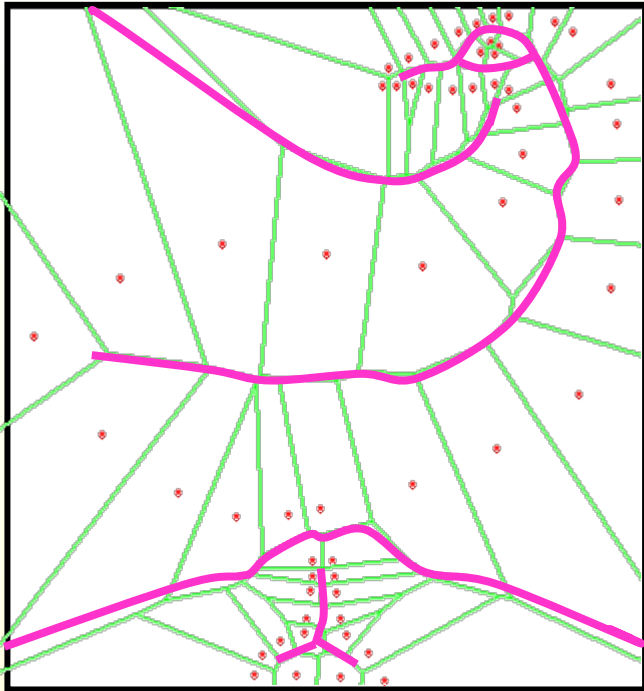
- Too many connections
- Need shape “structure” information - skeleton

Exploit Medial Axis



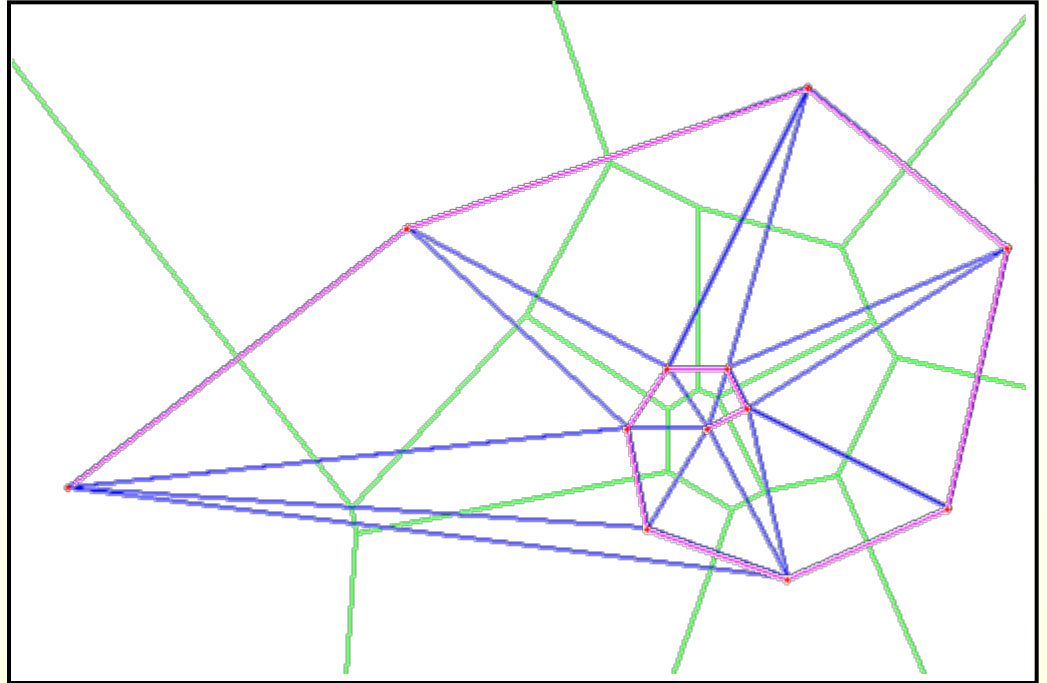
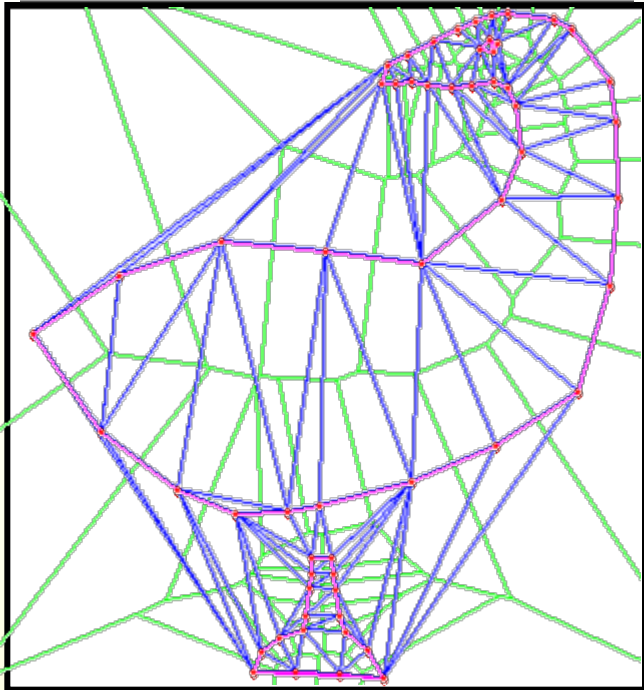
- Only accept edges far from the medial axis
- But medial axis is unknown

Use Voronoi!



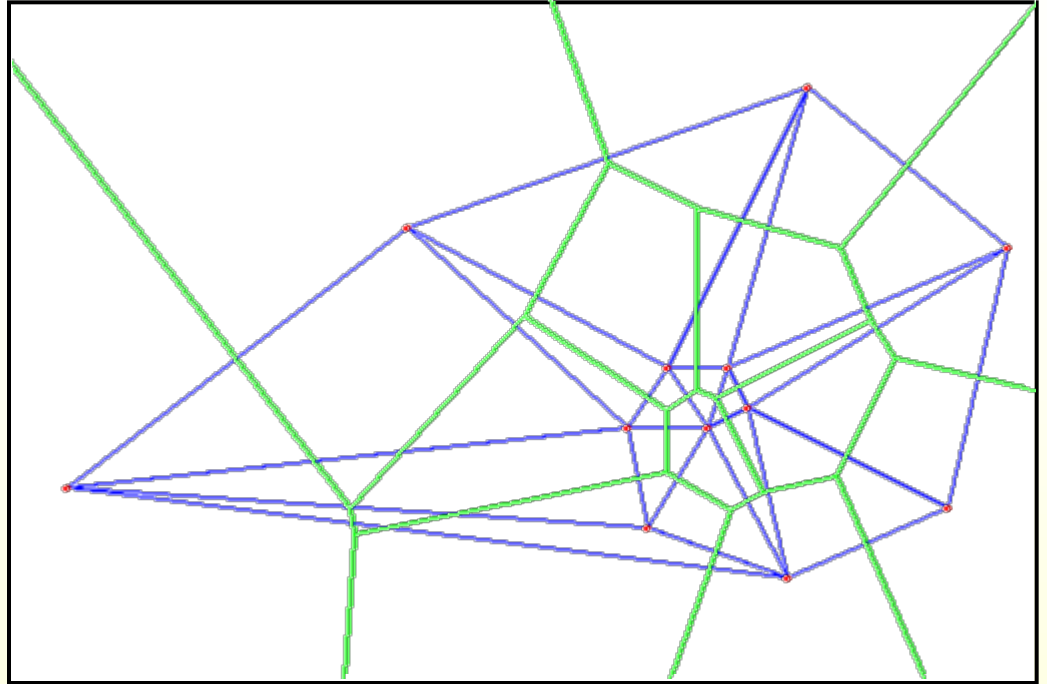
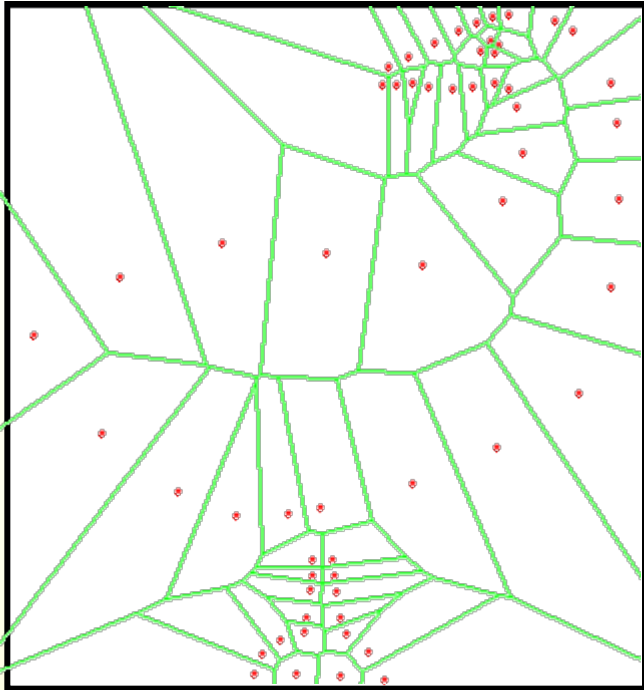
- Voronoi diagram of set of points on curve approximates the medial axis – if points sampled densely enough

The CRUST Algorithm



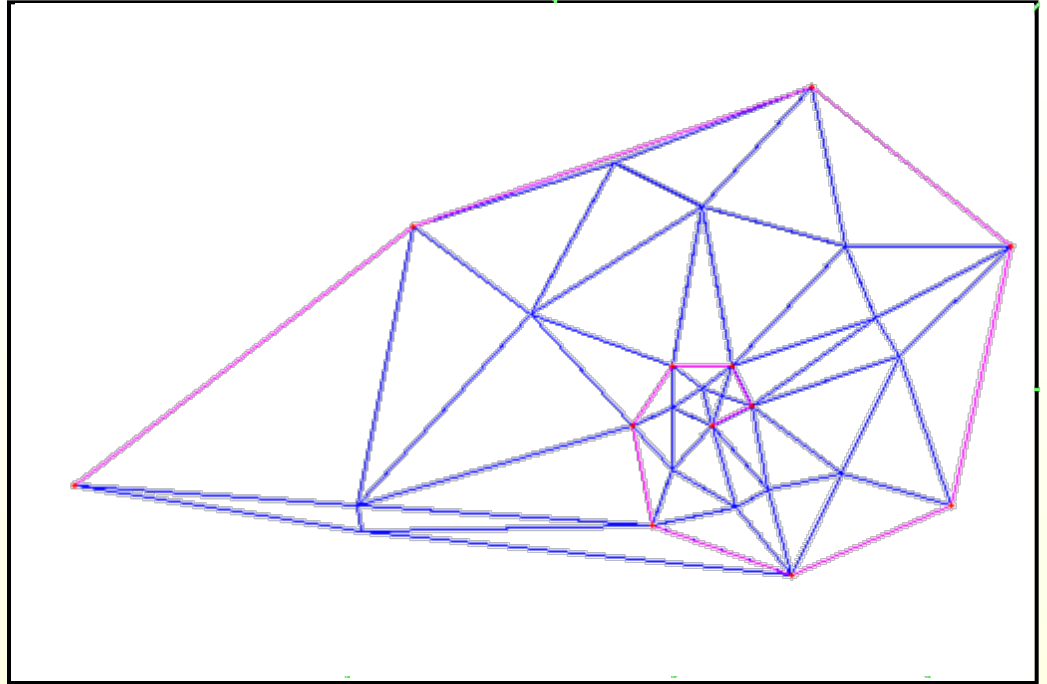
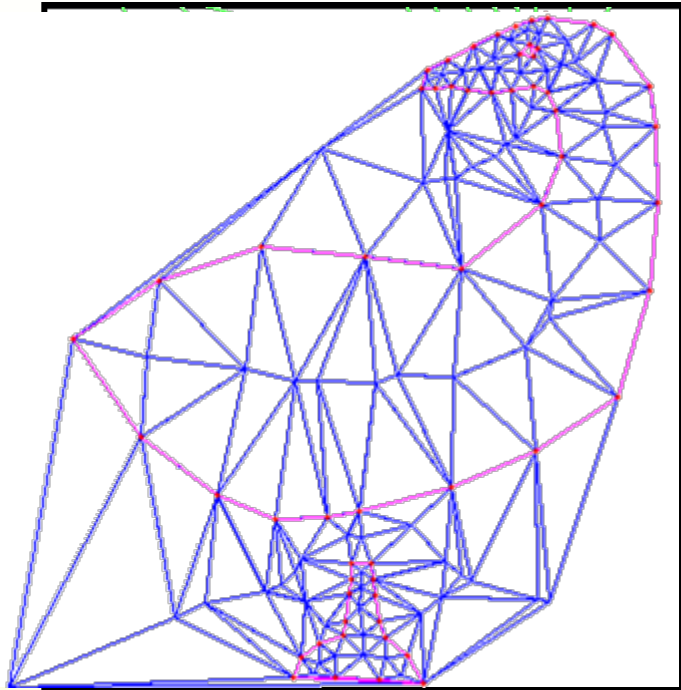
- Use Voronoi vertices to represent MA
- Edge e in **crust** \Leftrightarrow a circumcircle of e contains no other sample points or Voronoi vertices of S

The CRUST Algorithm



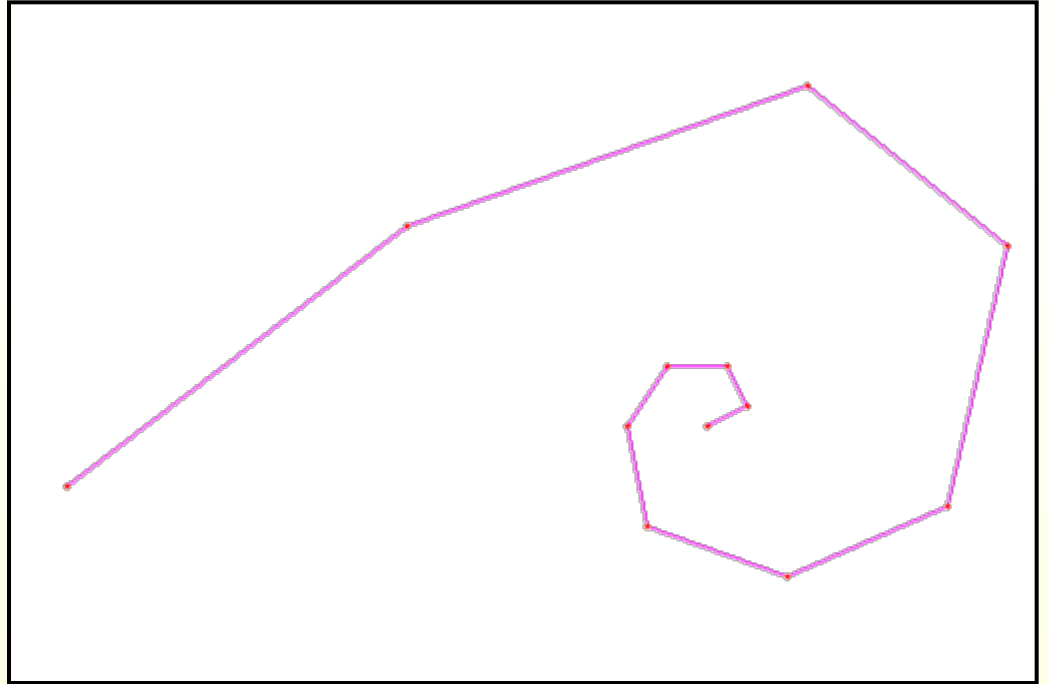
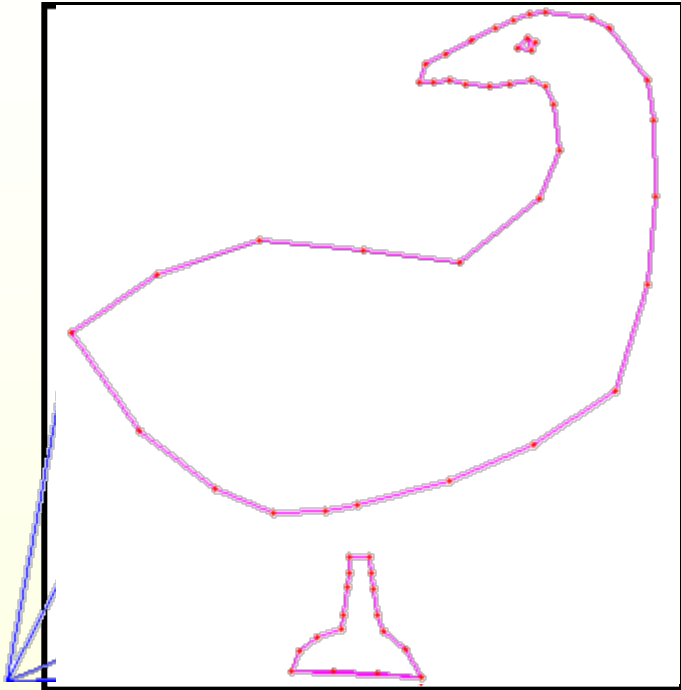
- Compute Voronoi diagram of S
- Let V be set of Voronoi vertices

The CRUST Algorithm



- Compute Voronoi diagram of S
- Let V be set of Voronoi vertices
- Compute Delaunay Triangulation of $S \cup V$

The CRUST Algorithm



- Compute Voronoi diagram of S
 - Let V be set of Voronoi vertices
- Compute Delaunay Triangulation of $S \cup V$
- **Crust** = all edges between points of S

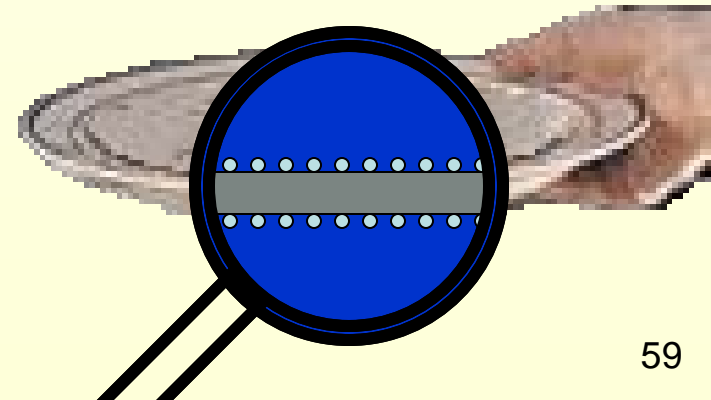
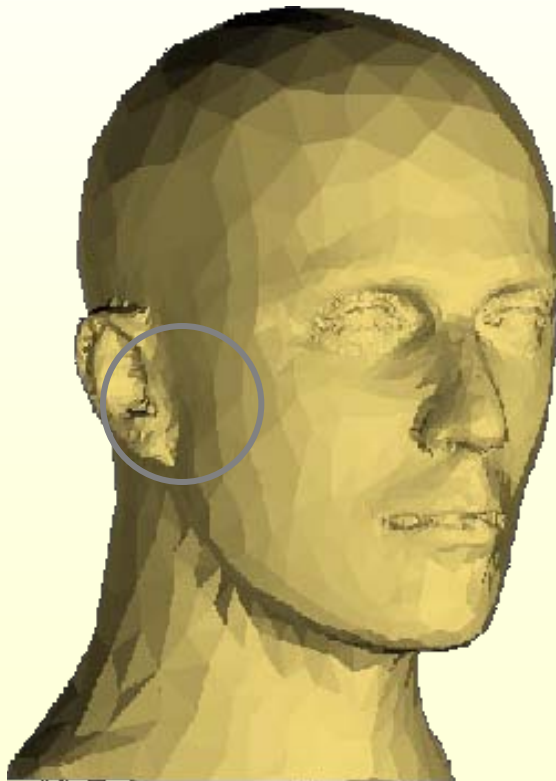
CRUST Applet in 2D

● Nice Applet:

[http://valis.cs.uiuc.edu/~sariel/research/CG/
applets/Crust/Crust.html](http://valis.cs.uiuc.edu/~sariel/research/CG/applets/Crust/Crust.html)

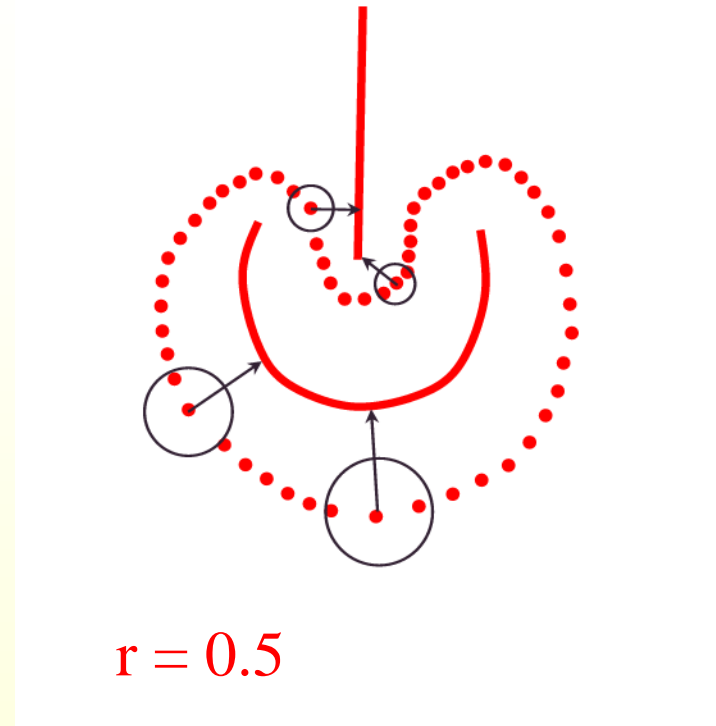
Sampling Conditions

- Must take into account
 - *curvature* of the surface
 - *proximity* of other parts of the surface



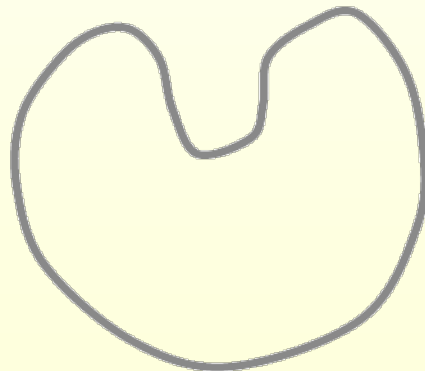
Sampling Conditions

- Good sample - sampling density (at least) inversely proportional to distance from medial axis
- ***r-sample***: distance from any point p on surface to nearest sample point $\leq rd$
 - d - distance from point p to medial axis
- In general, $r \in (0, 1]$
- $r=0.252\dots$ is good enough



CRUST Algorithm – Theory

- Basic Theorem: The crust of an r -sample from a smooth curve F , for $r \leq 0.252 \dots$ connects exactly the adjacent samples of F

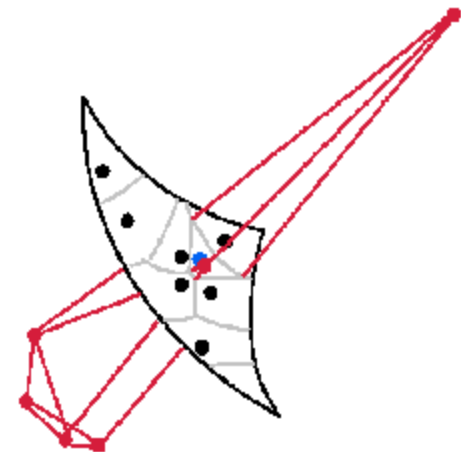
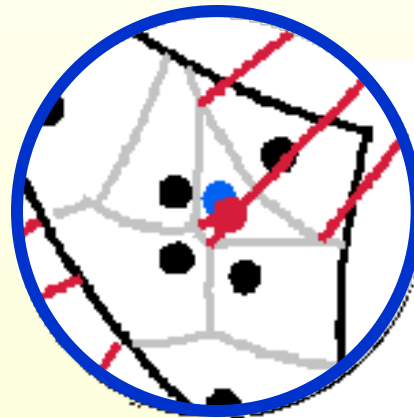


Smooth curve F

- The algorithm may fail when r is too big

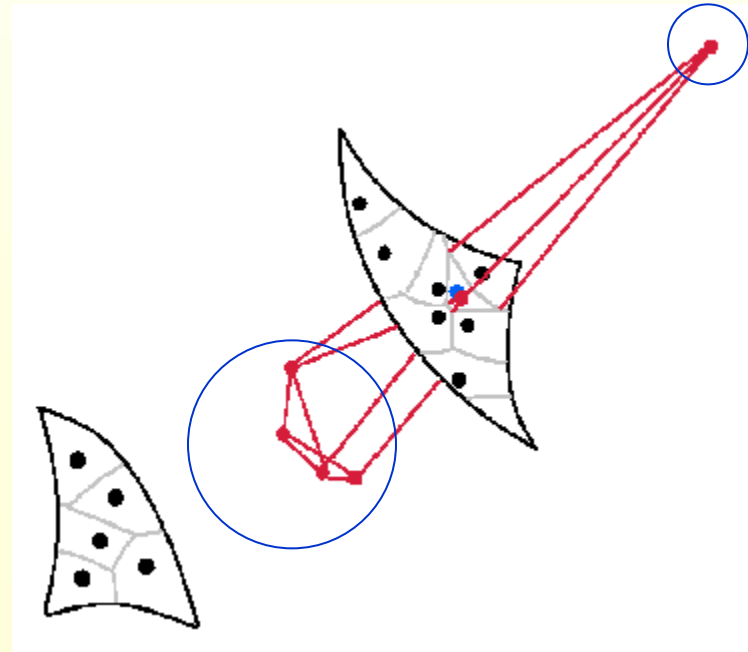
3D CRUST Algorithm

- Extend 2D approach
- Voronoi vertex is equidistant from 4 sample points
- **BUT in 3D not all Voronoi vertices are near medial axis**
(regardless of sampling density)



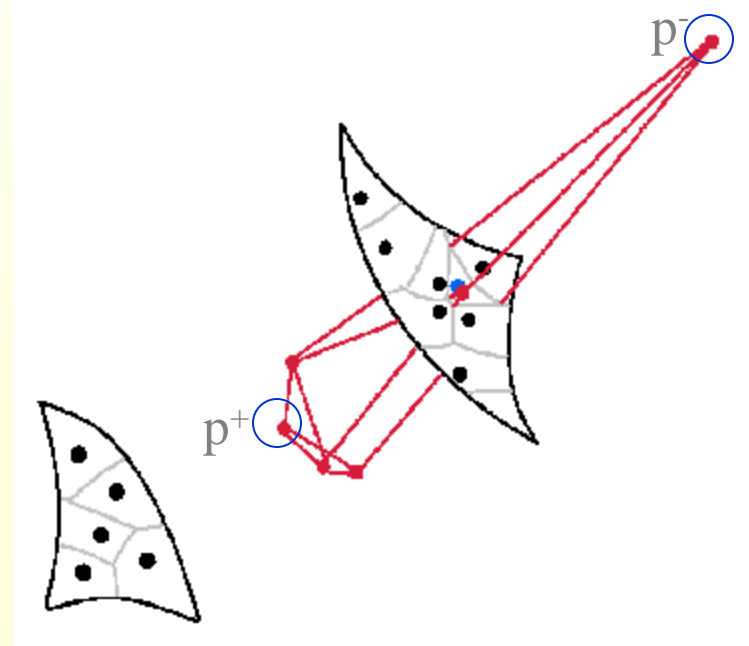
3D CRUST Algorithm

- But **some** vertices of the Voronoi cell are near the medial axis
- Intuitively – cell is closed not just from the sides but also from “top” & “bottom”



3D CRUST Algorithm

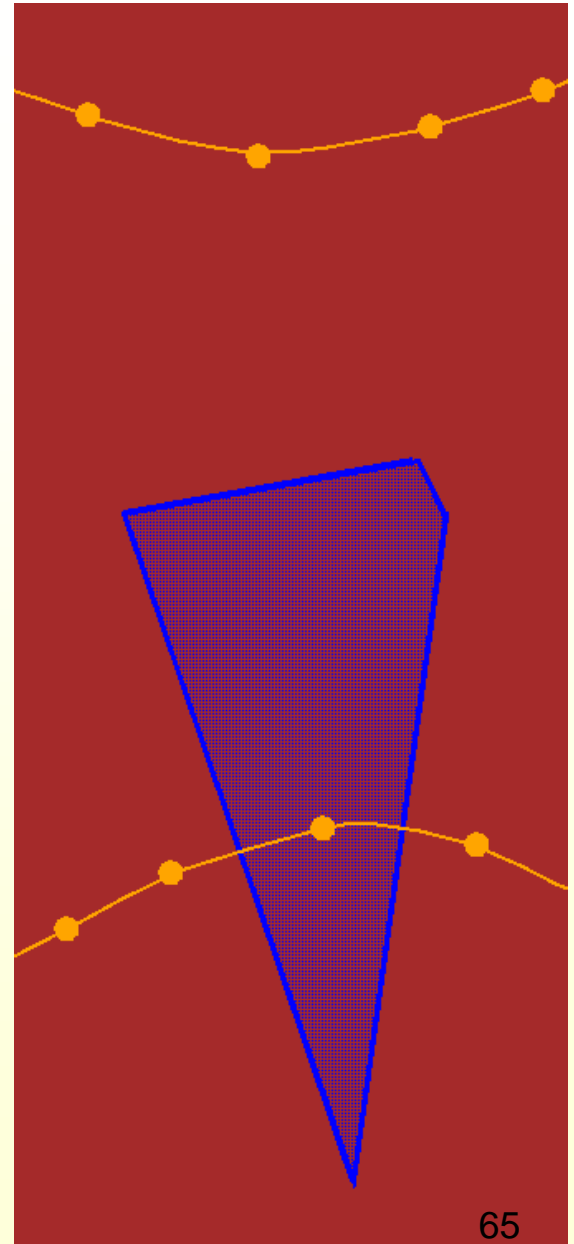
- Solution: use only two farthest vertices of V_s -
- one on each side of the surface
- Call vertices **poles** of s (p^+ , p^-)



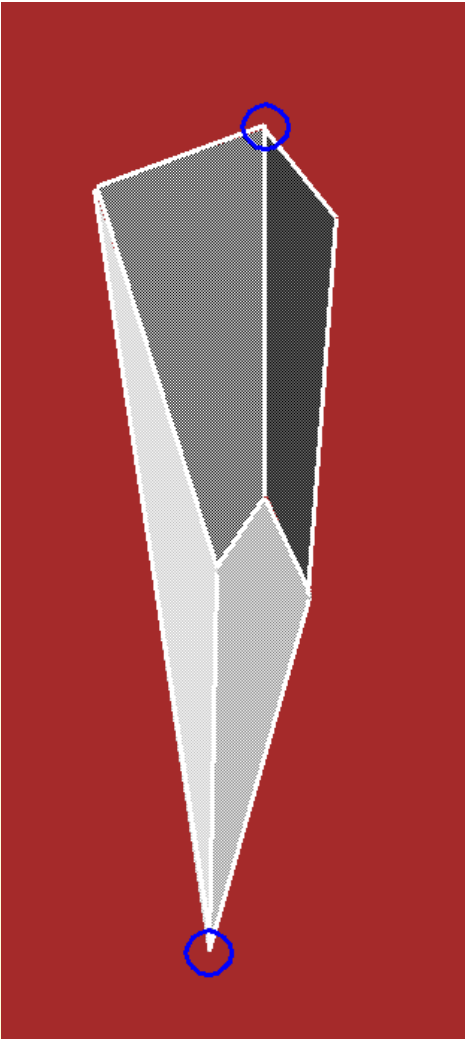
Poles

For dense surface samples, Voronoi cells are:

- long and skinny
- perpendicular to surface
- with ends near the medial axis



Poles



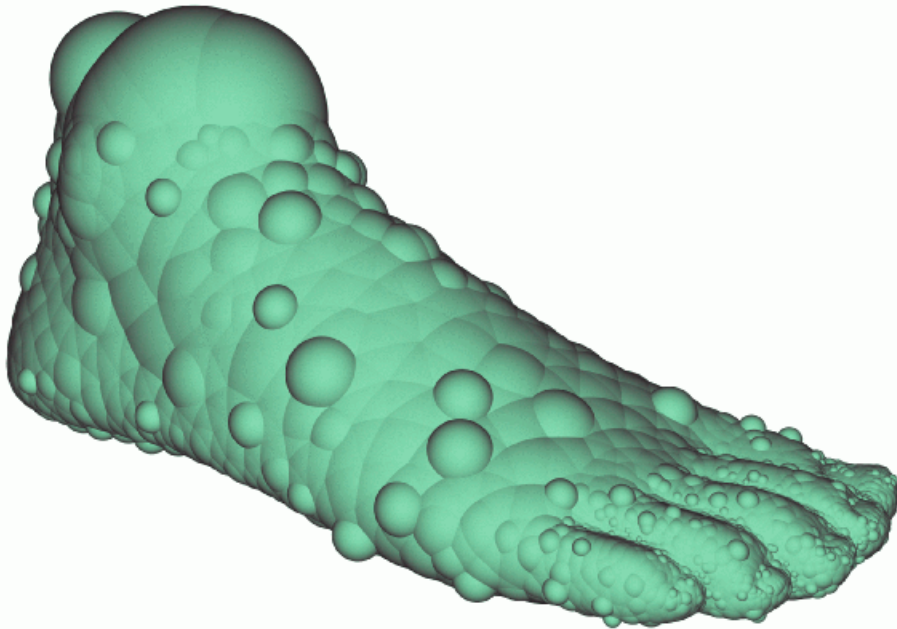
Poles are Voronoi vertices at opposite ends

To find: farthest Voronoi vertex from sample,
farthest on opposite side

Poles

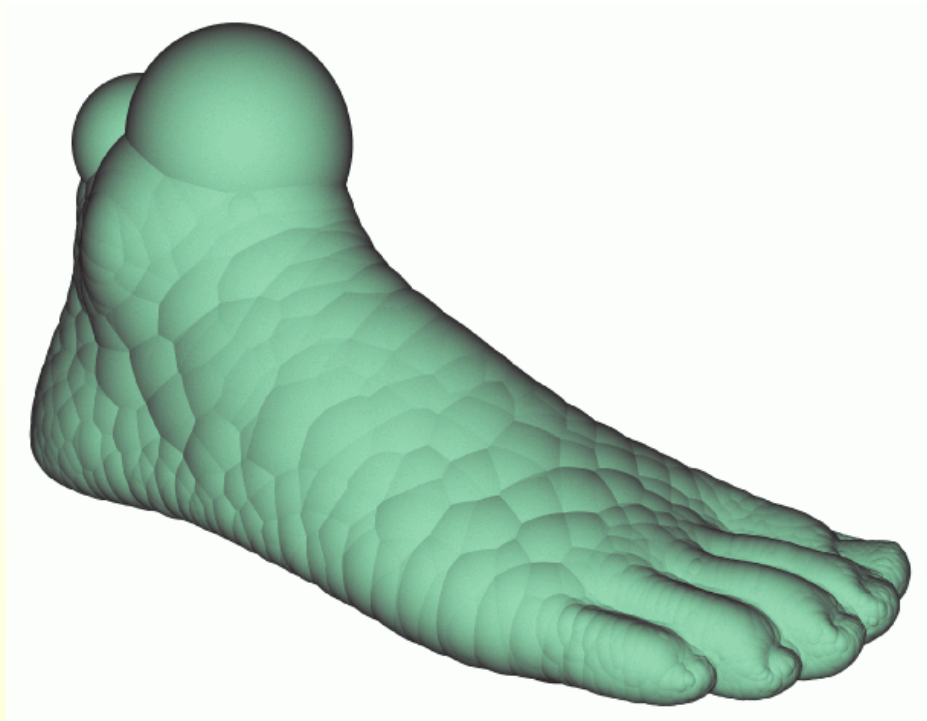
Problem in 3D:

Not all Voronoi vertices are near medial axis, even when samples are arbitrarily dense



Interior Voronoi balls

Poles



Subset of Voronoi vertices, the poles, approximate medial axis

Interior *polar* balls

3D CRUST Algorithm

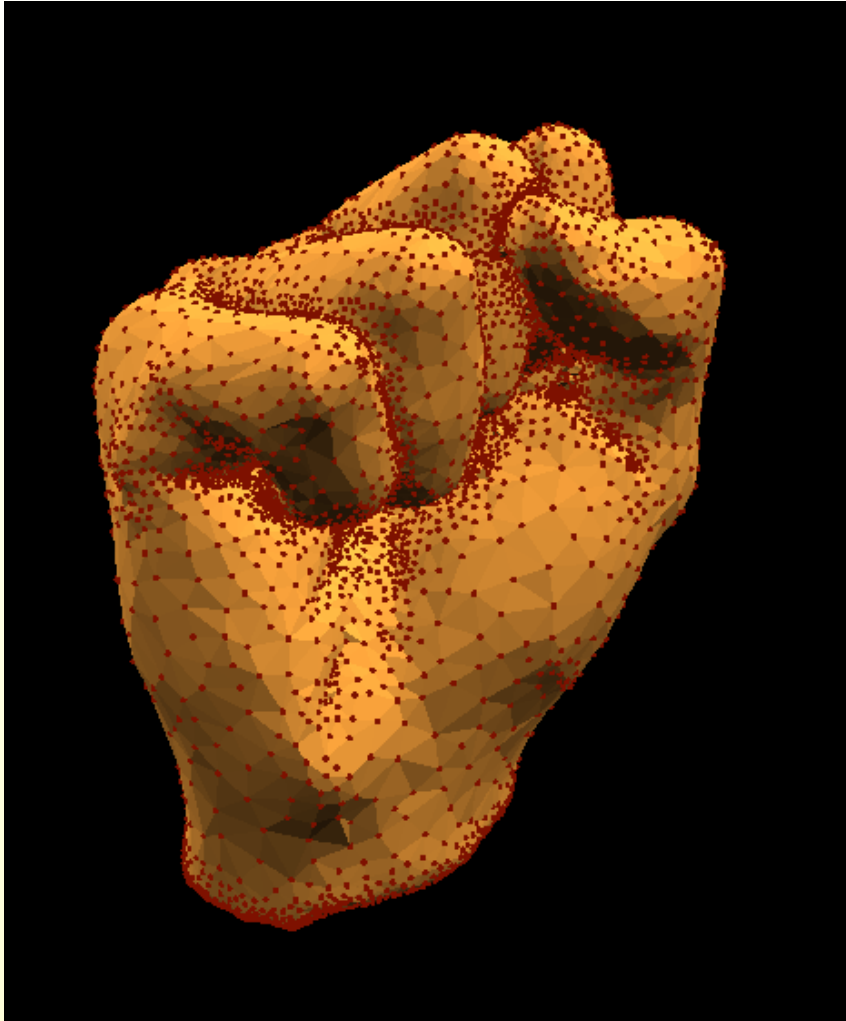
- Compute Voronoi diagram of S
- For each s in S find (p^+, p^-)
- Let P be the set of all poles p^+ and p^-
- Compute Delaunay triangulation T of $S \cup P$
- Add to crust all triangles in T with vertices in S

Time Complexity

- Time complexity is $O(n^2)$ where $n = |S|$
- Upper bound on 3D Delaunay triangulation
- Almost never arises in practice
- Other steps are linear (but expensive)



Summary: CRUST Algorithm



Surface reconstruction with theoretical guarantees

Uses poles to find Delaunay triangles eligible for surface

Amenta & Bern, 98

Amenta, Bern and Kamvysselis, 98

More Reconstruction Examples

Golf club

16864 points

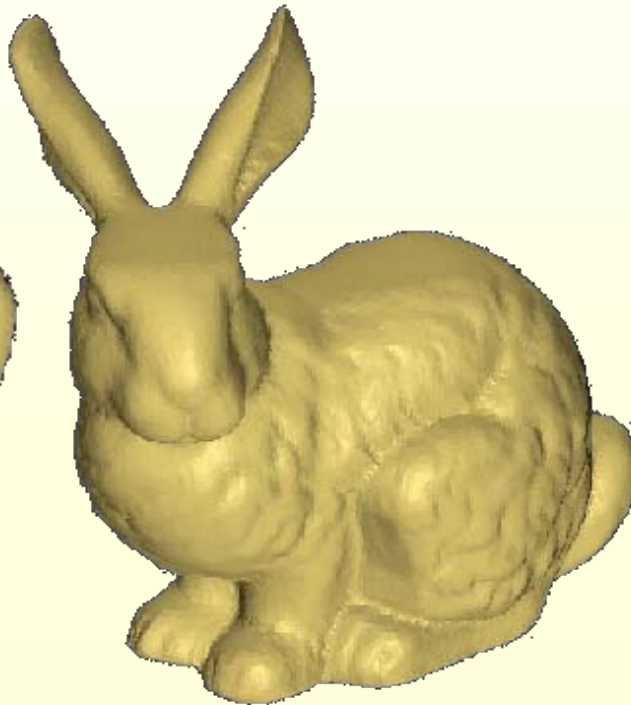
12 minutes



Bunny

35947 points

23 minutes



Foot

20021 points

15 minutes



Improvements on Crust

Amenta et al, '00: Simpler algorithm, simpler proof, topological guarantees.

Dey and Giesen, '01: Sharp corners and boundaries.

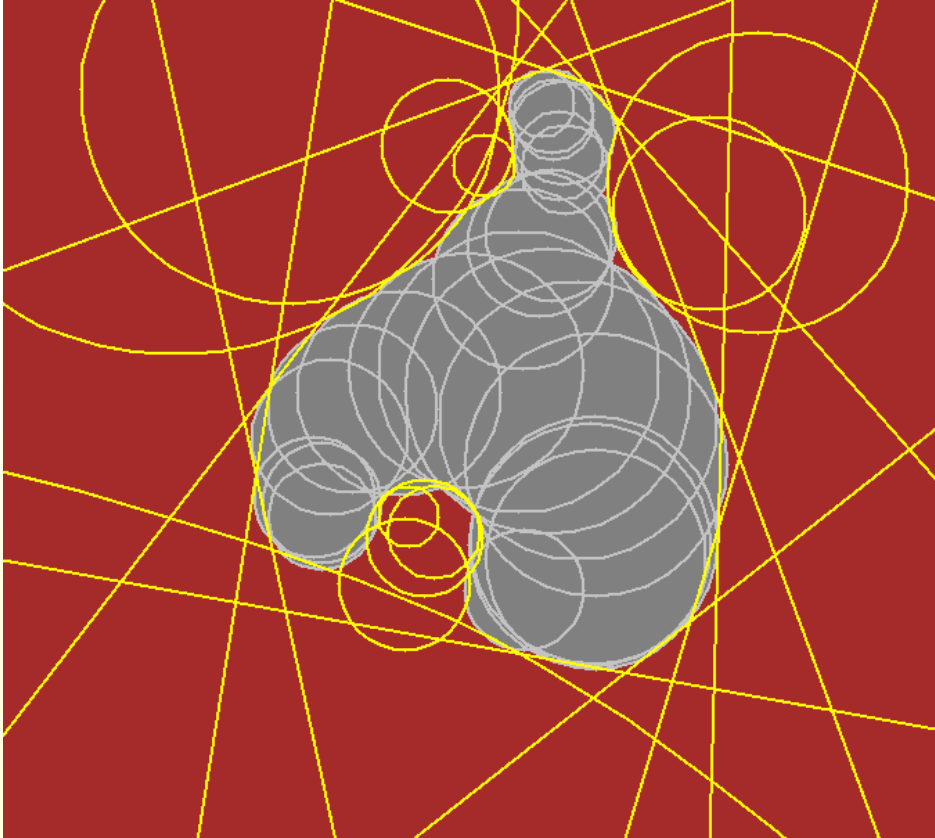
Ramos, '01: $O(n \lg n)$ algorithm, replacing Delaunay with well-separated pair decomposition.

Practical Crust Drawbacks

- Fails when sample is not sufficiently dense: holes in surface, errors at sharp corners
- Need to select surface from set of eligible triangles. Hard to do in a way that is provably correct, makes nice surface, etc

Still needed: Algorithm which is robust, has no post-processing, and is still correct

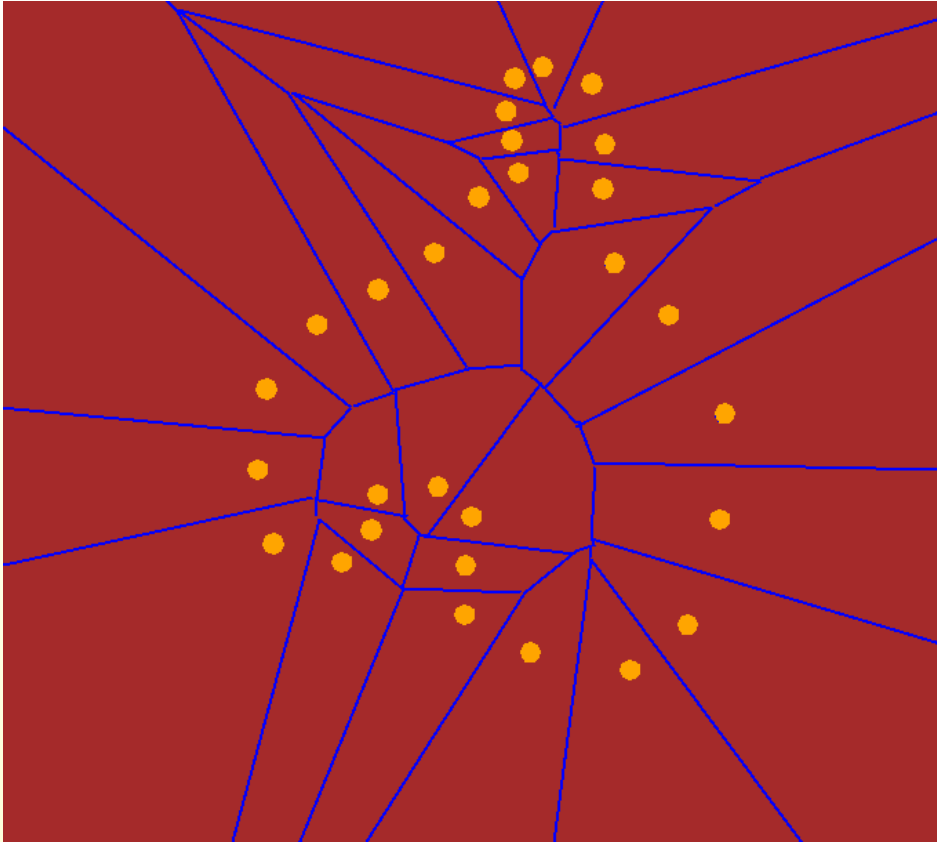
Power Crust



Amenta, Choi and Kolluri, 01

Idea: Approximate object as union of balls, compute polygonal surface from balls

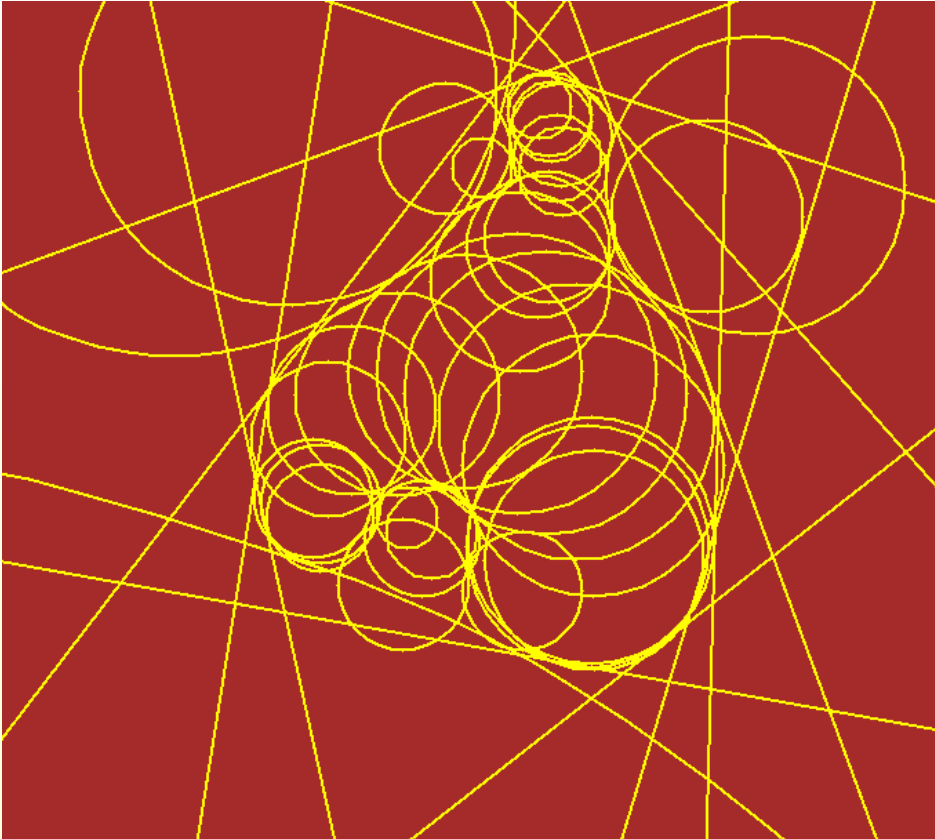
Power Crust



Compute Voronoi diagram of samples

Select poles to approximate object and its complement by finite unions of balls

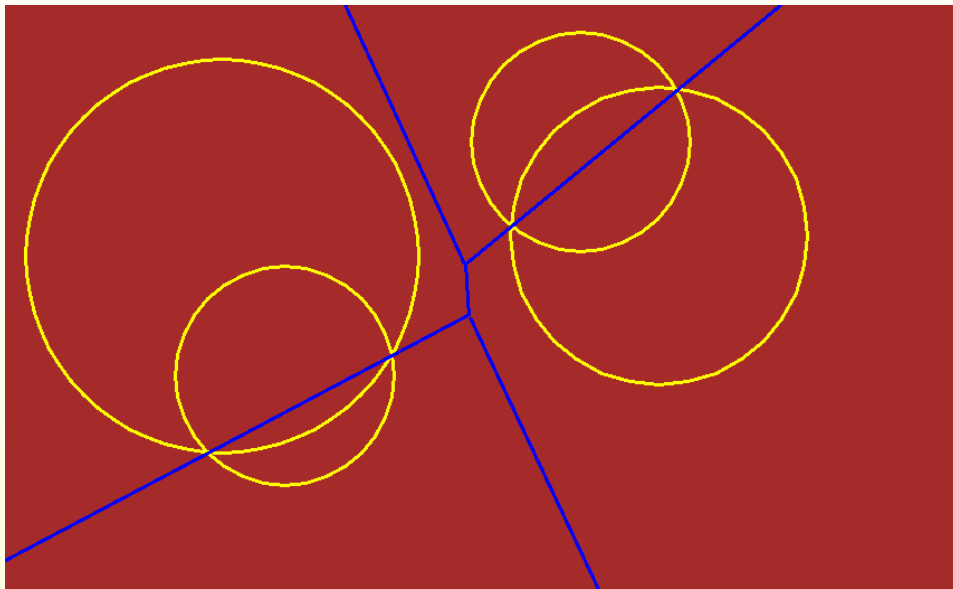
Power Crust



Compute polygonal surface from
polar balls using the **power diagram**

Power Diagram

Power diagram is Voronoi diagram of balls.



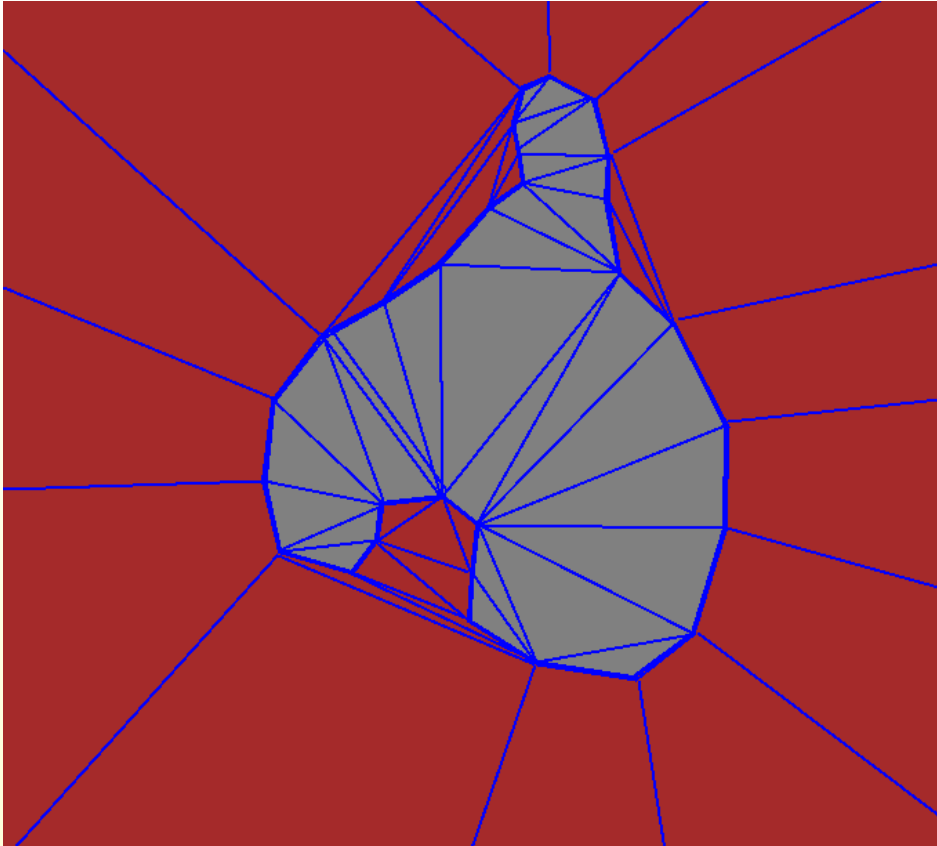
Voronoi diagram program can be easily modified to produce power diagrams

Using power distance function

$$p(x, B) = d^2(x, c) - r^2$$

Has polyhedral cells

Power Crust



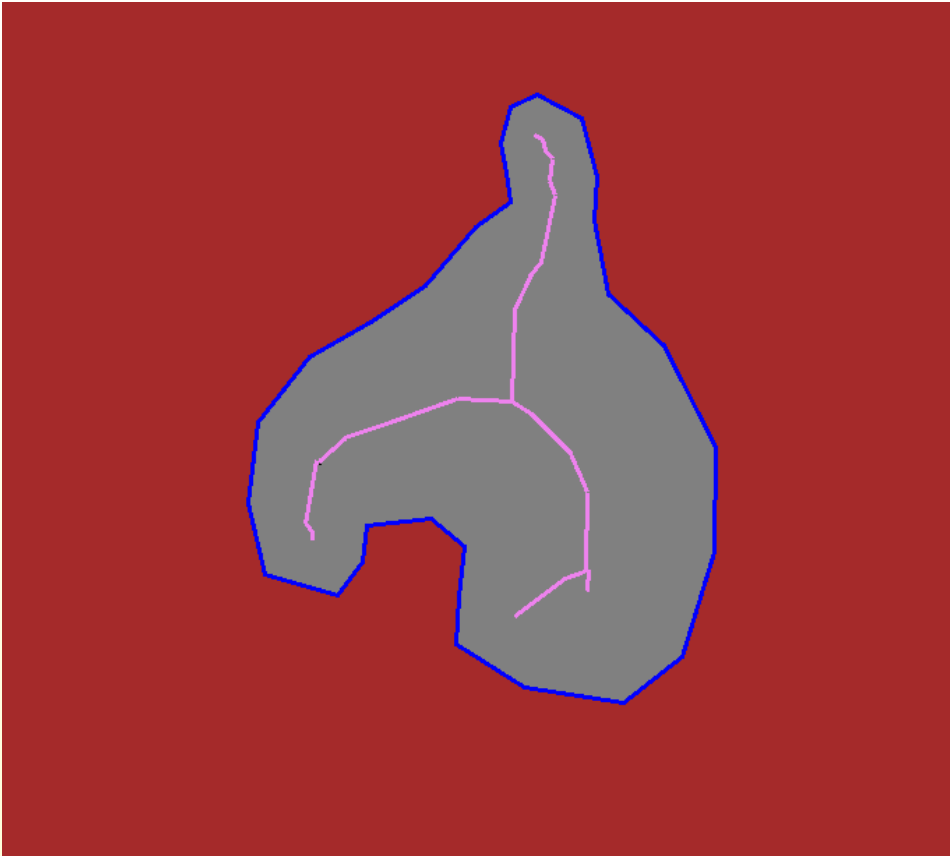
Label power diagram cells inside or outside object (skipping details)

Inside cells form polyhedral solid

Power Crust

Boundary of solid approximates
surface: power crust

Connect inner poles with adjacent
power diagram cells: power shape
approximates medial axis



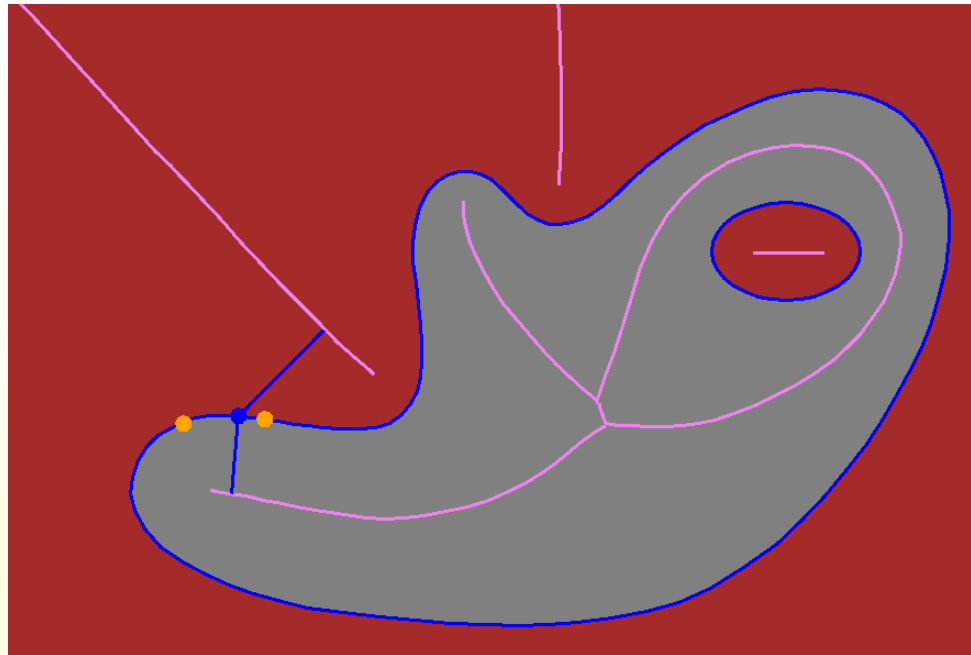
Power Crust

Robust: Always boundary of a solid

Simple: No surface extraction or hole-filling steps required

Correct: Theoretical results relate geometric and topological quality of approximation to quality of sample

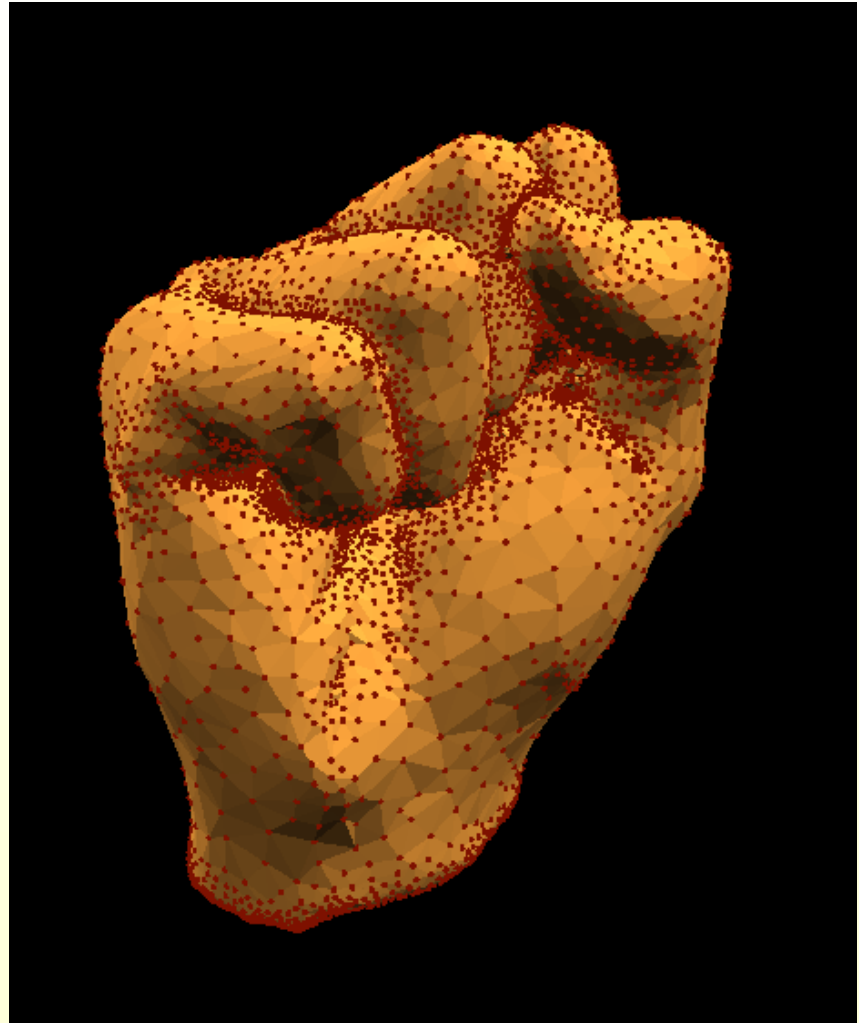
Sampling Requirement



Sample is **sufficiently dense** when distance from any surface point x to nearest sample is at most small constant r times distance to medial axis.

Sampling Requirement

Captures intuition that we need dense sampling where curvature is high or where there are nearby features.



Theoretical Results

Amenta, Choi, Kolluri, CGTA 01.

Assume sufficiently dense sampling, smooth surface

- Power crust approaches object surface linearly as sampling density increases
- Power crust normals converge to surface normals linearly

- Power crust is homeomorphic to surface for dense enough samples

Theoretical Results

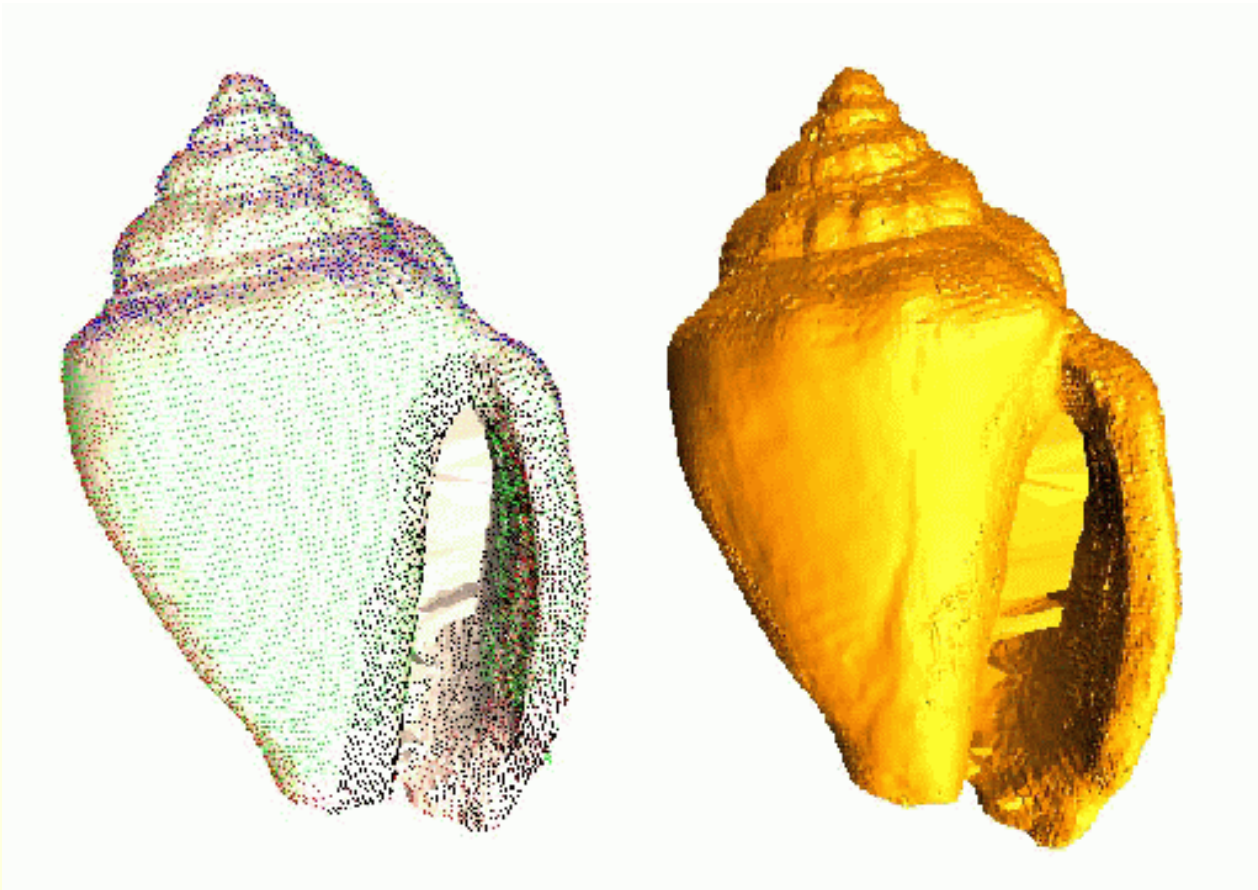
- Similar results for union of balls
- Power shape (union of balls) is homotopy equivalent to the underlying solid object
- Set of poles converges to medial axis, faster in some places than in others
 - also Boissonnat and Cazals, 01; and Dey, 02 gives polygonal MA approximation.

Results



Laser range data, power crust, approximate medial axis.

Results



Four laser range scans merged. Hole deep inside object filled.

Robust



Good reconstruction even with lots of added noise.

Medial Axis Simplification

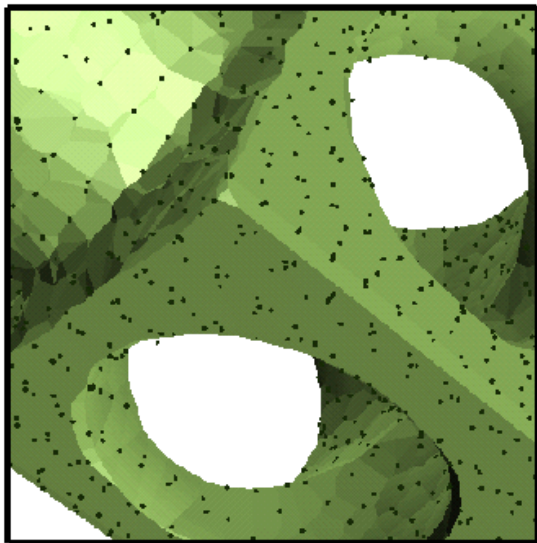
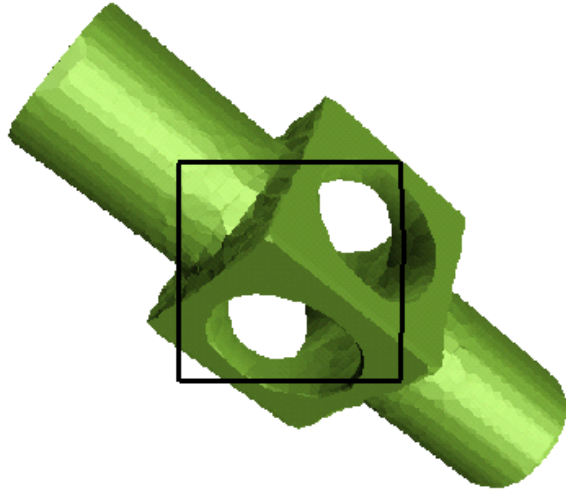


Simplification of original power shape



Simplification of power shape of noisy hand

Sharp Corners



With additional hack, sharp edges can be resolved when they are far enough away from other features.

No need to have samples on the edge

The End

