Surface Reconstruction I

USTC, 2024 Spring

Qing Fang, fq1208@mail.ustc.edu.cn

https://qingfang1208.github.io/

Introduction

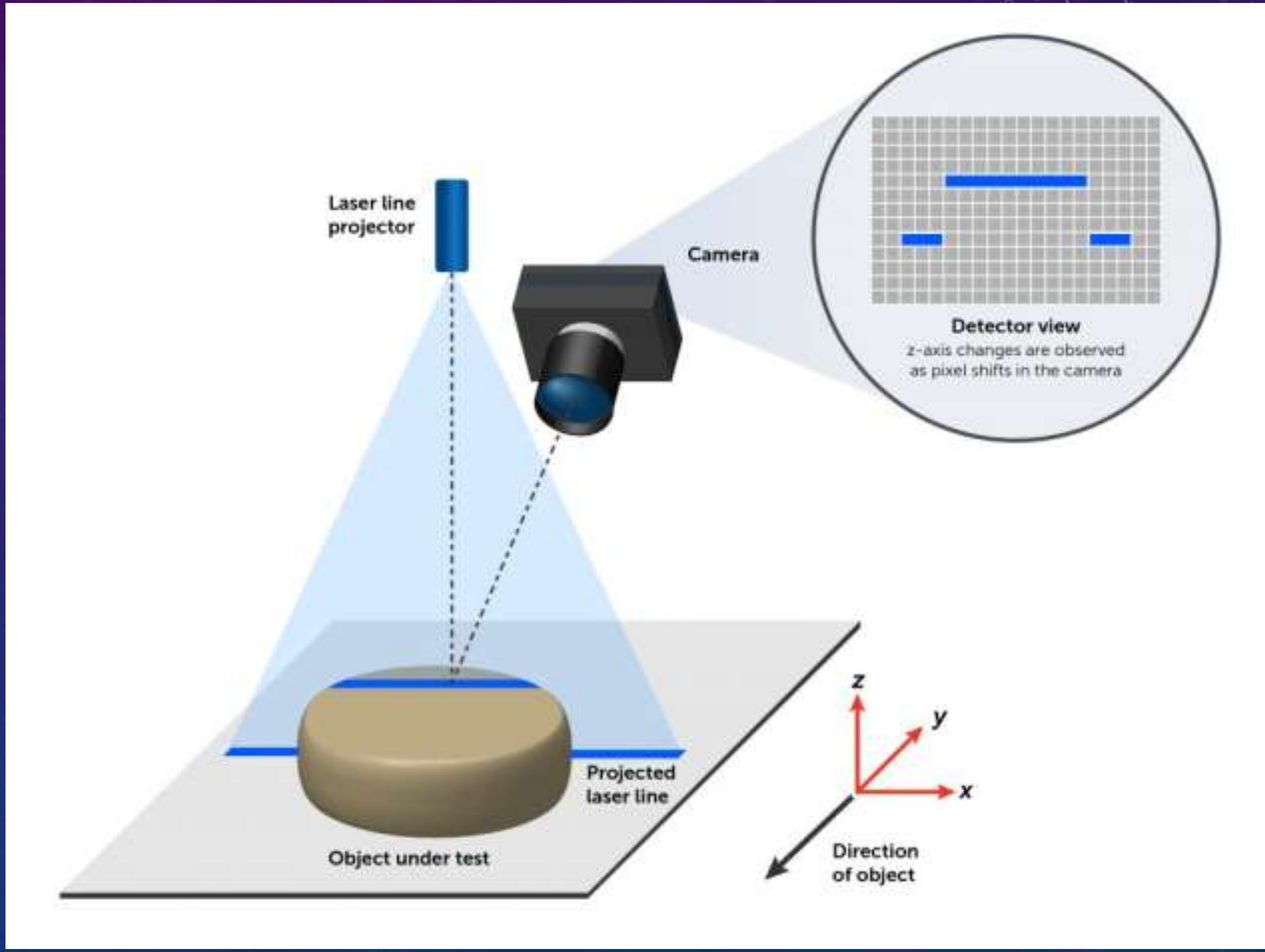
Surface Reconstruction

- Rendering

- Reconstruction

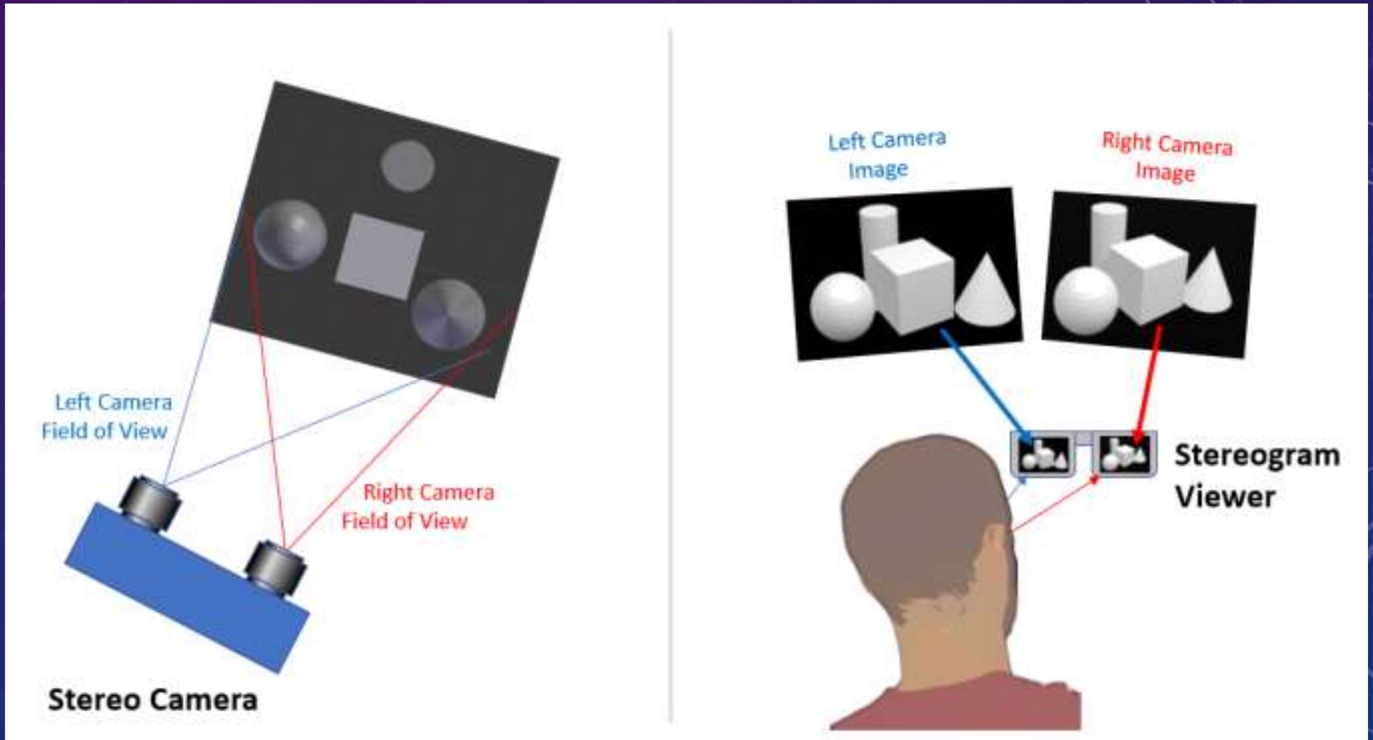
Shape from …

- Laser triangulation



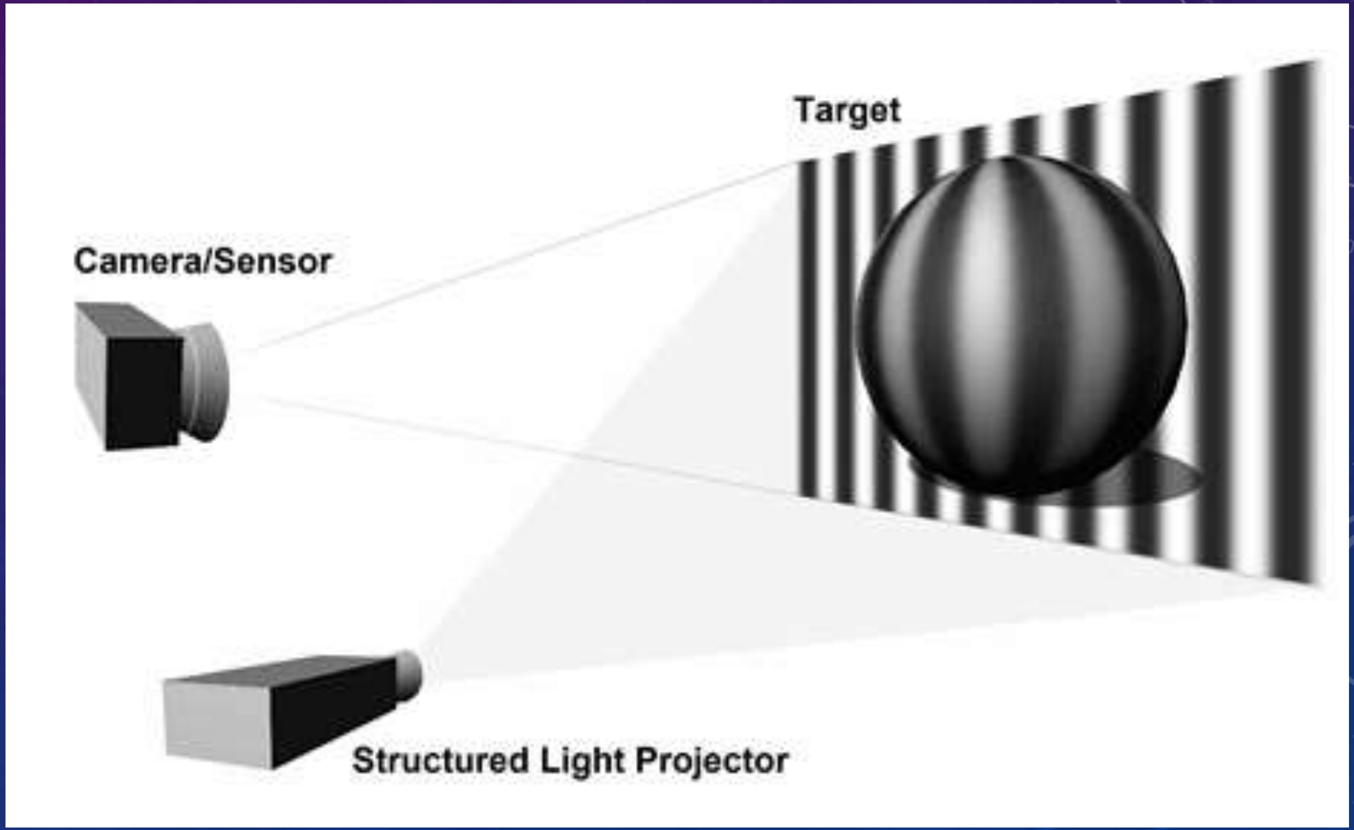
Shape from …

- Laser triangulation
- Stereo

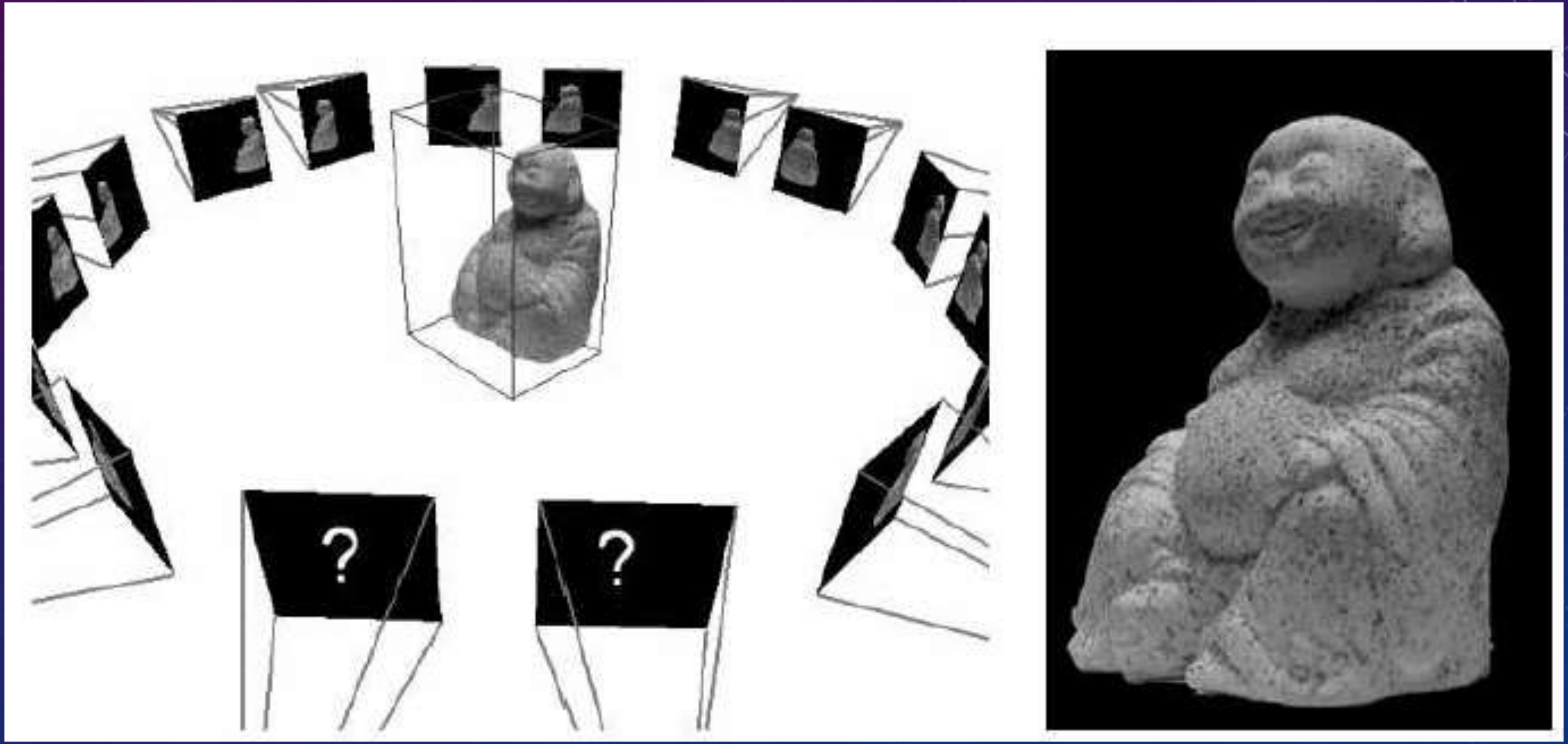


Shape from …

- Laser triangulation
- Stereo
- Structured Light
- …

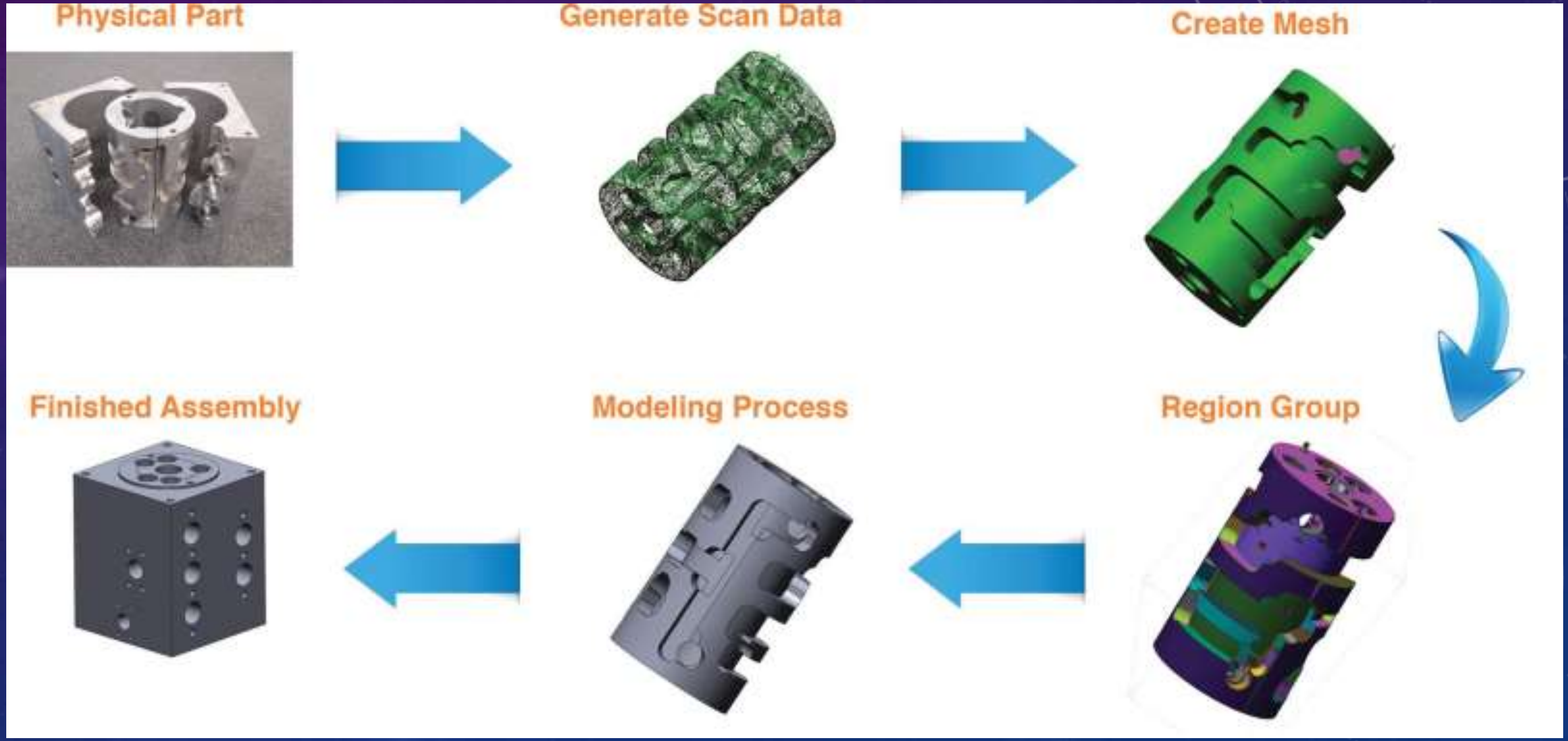


Shape from data



Applications

- Reverse engineering

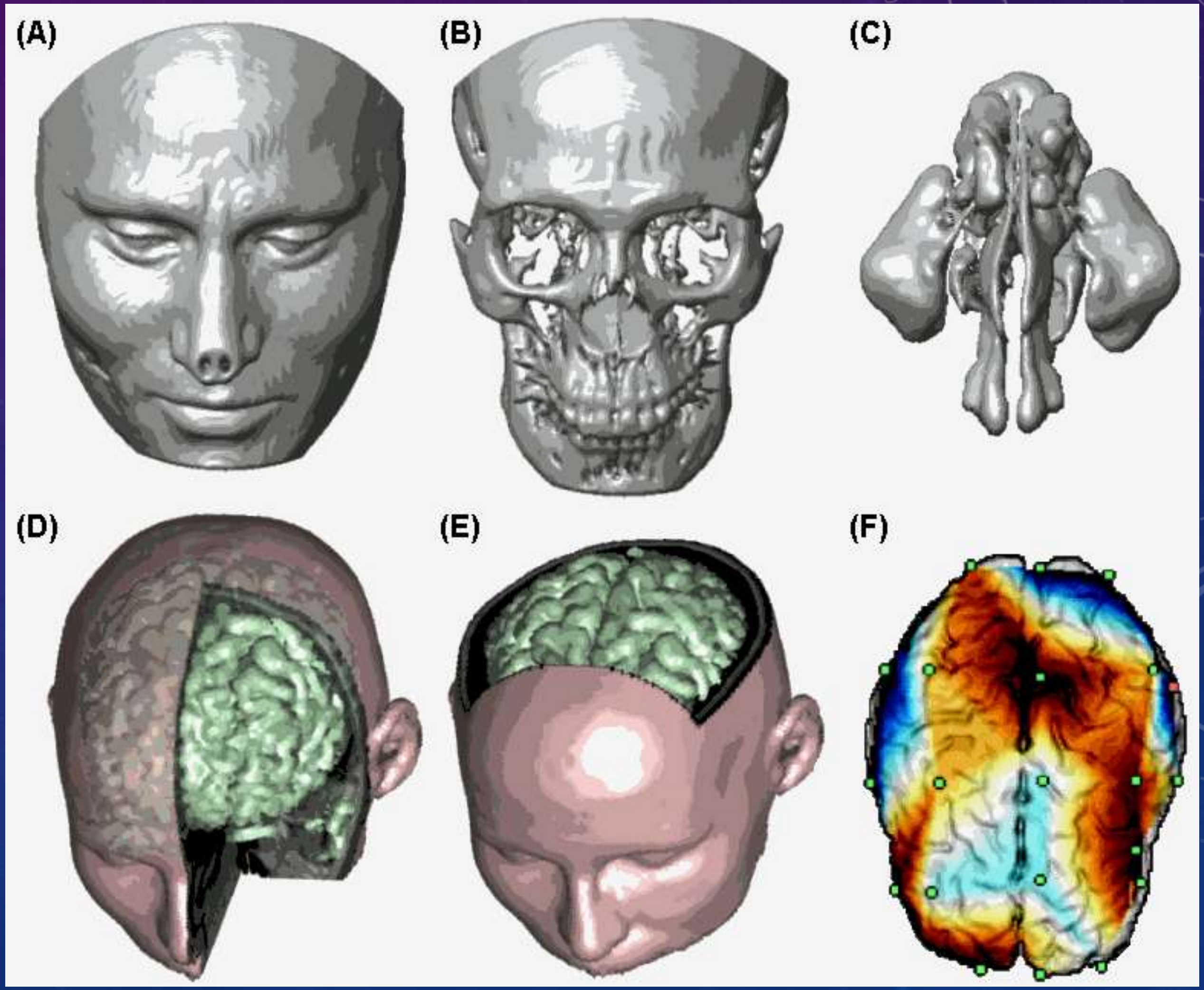


Applications

- Reverse engineering
- Augmented reality

Applications

- Reverse engineering
- Augmented reality
- Medical Imaging

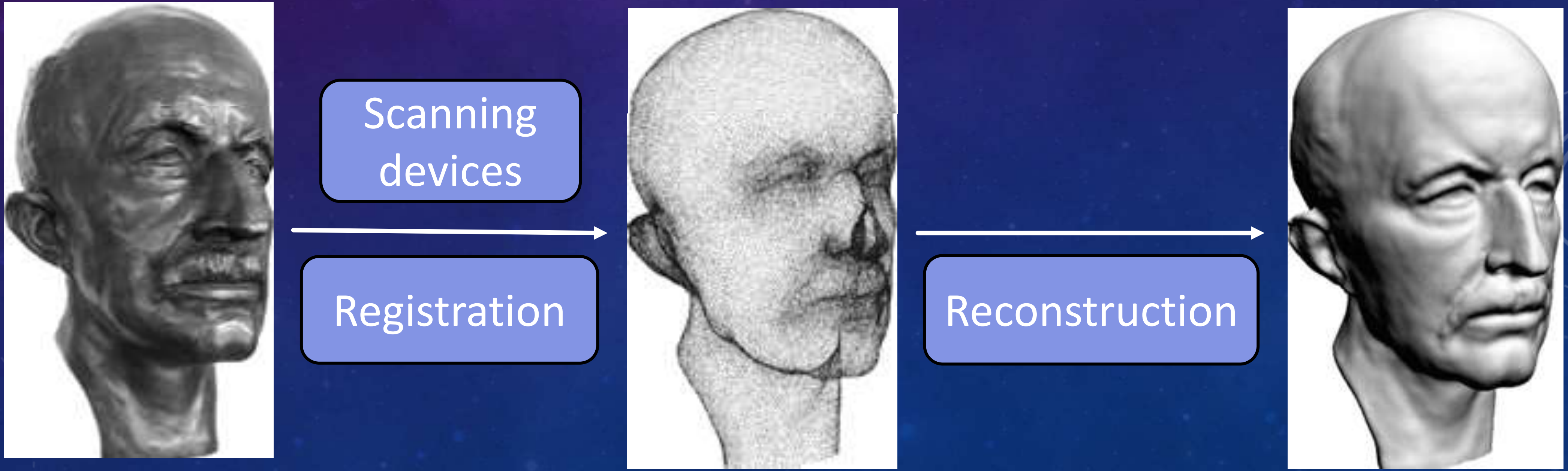


Applications

- Reverse engineering
- Augmented reality
- Medical Imaging
- Digital preservation
- …

Problem

- Input: a multi-view set of points in 3D that sampled from a model surface
- Output: a 2D manifold mesh surface that closely approximates the model



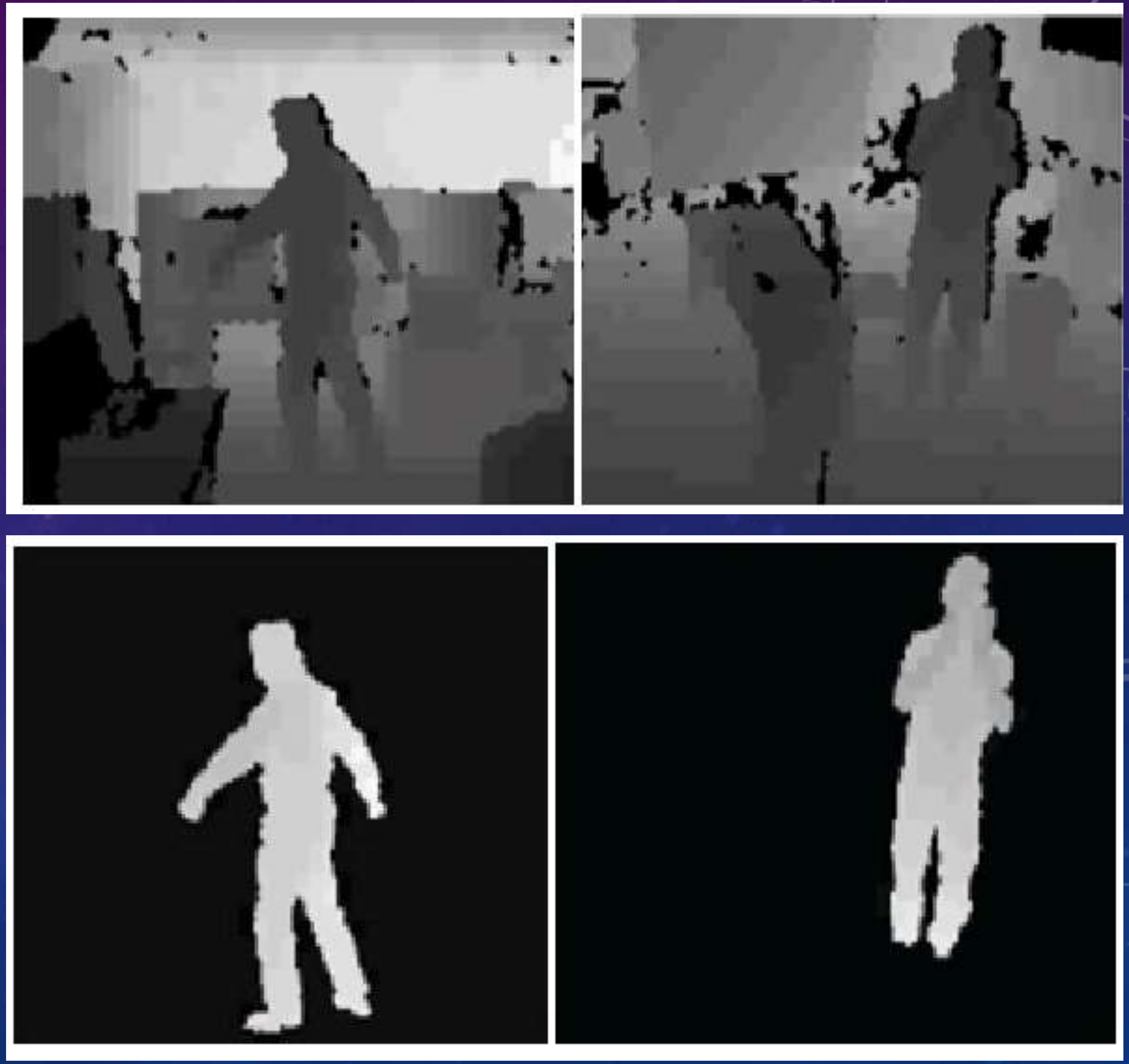
Registration

Depth Image

- Resolution: width × height
- Pixels: depth value
  - Nearer is darker

Point clouds

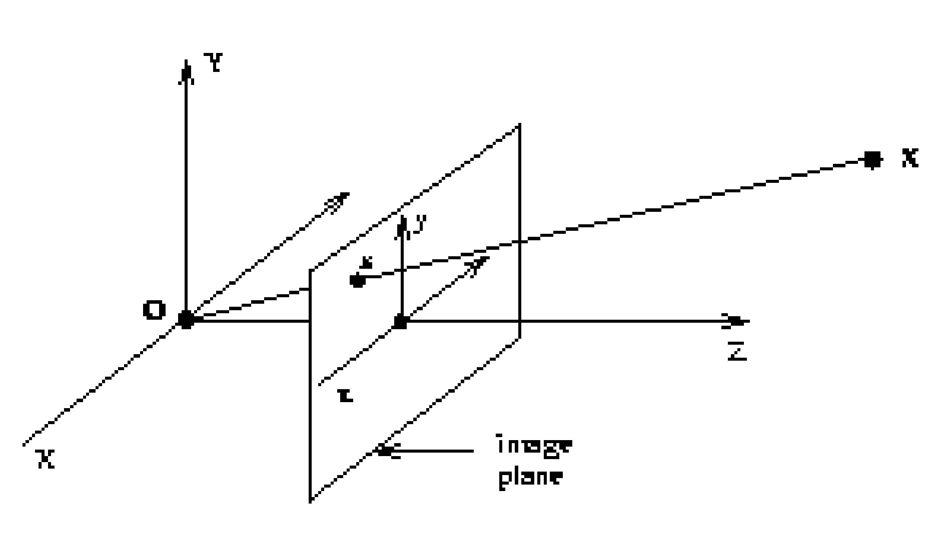
- Preprocessing
  - Segmentation

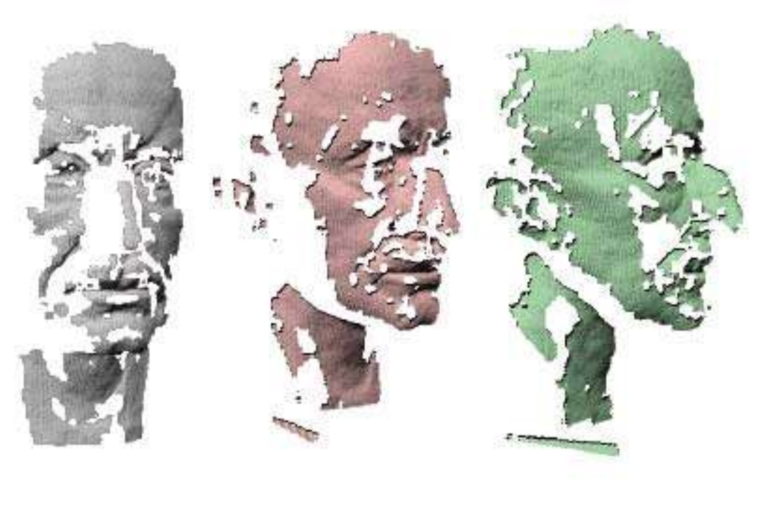


Point clouds

- Preprocessing
  - Segmentation
  - Camera matrix

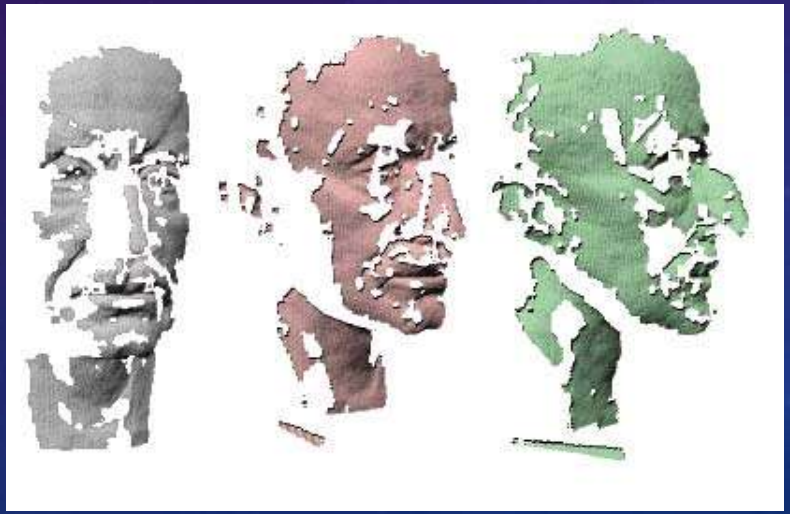
$$Z\begin{pmatrix}u\\v\\1\end{pmatrix}=\begin{pmatrix}f_x & 0 & c_x\\0 & f_y & c_y\\0 & 0 & 1\end{pmatrix}\begin{pmatrix}X\\Y\\Z\end{pmatrix}$$

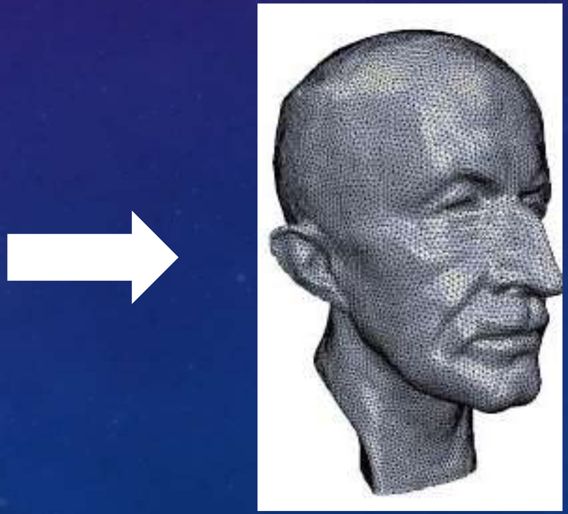




Registration

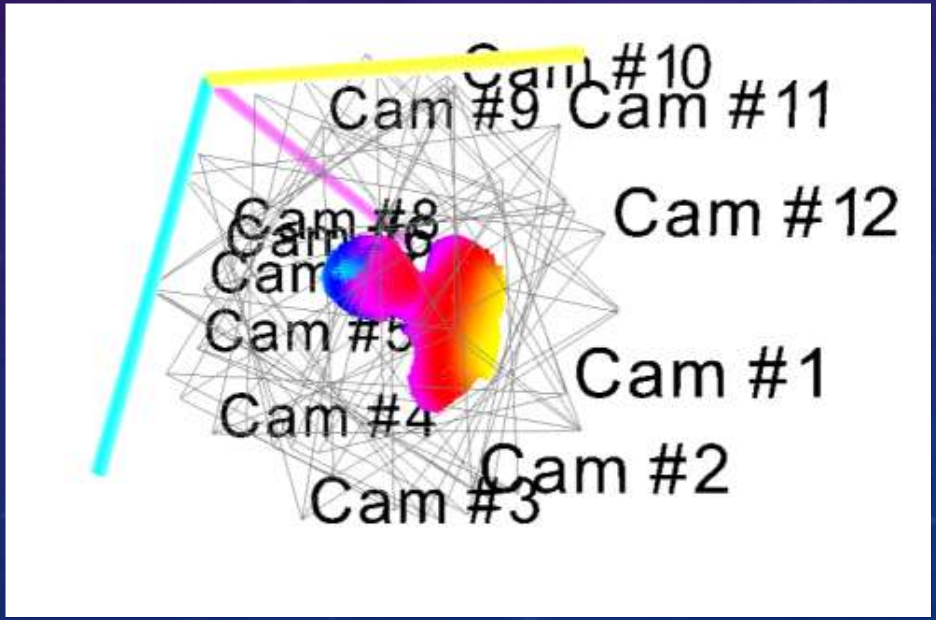
- Any surface reconstruction algorithm should strive to use all of the detail in all the available range data.

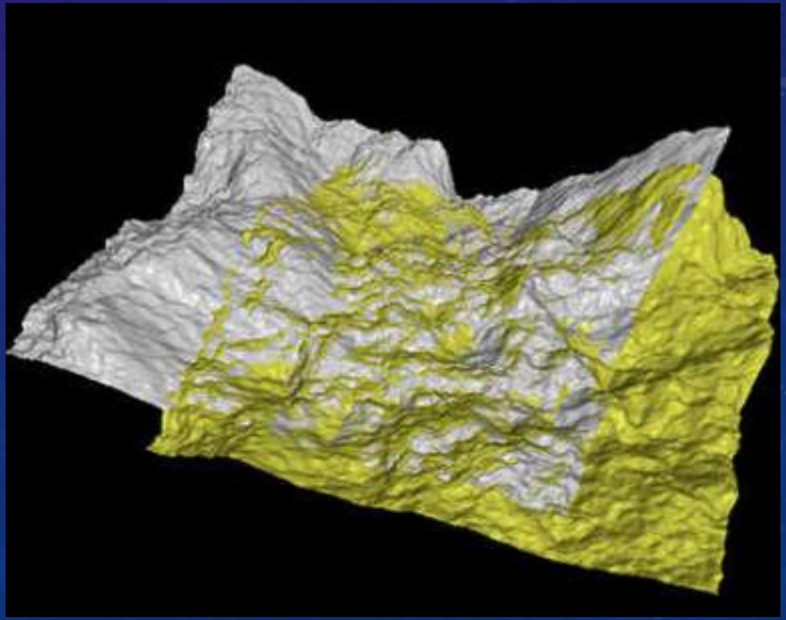




Accurate registration may require

- Calibrated scanner/object positioning
- Software-based optimization

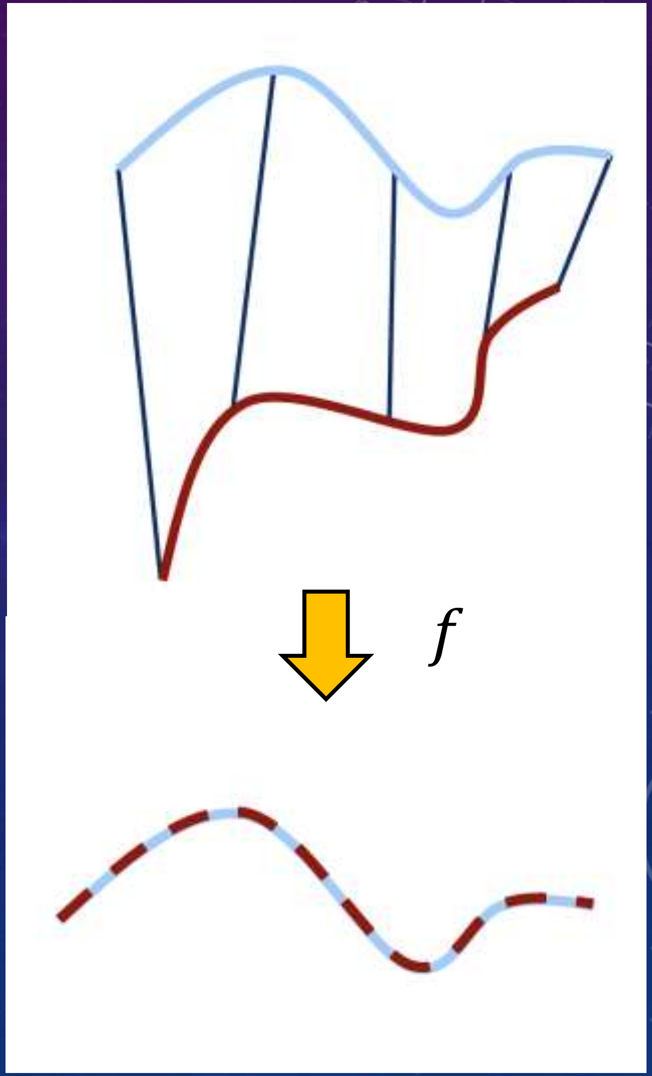




Pairwise registration

- Source point sets: $P = \{p_1, \ldots, p_m\}$
- Target point sets: $Q = \{q_1, \ldots, q_n\}$
- Find function f, s.t. minimize

$$E = dist^2(f(P), Q)$$

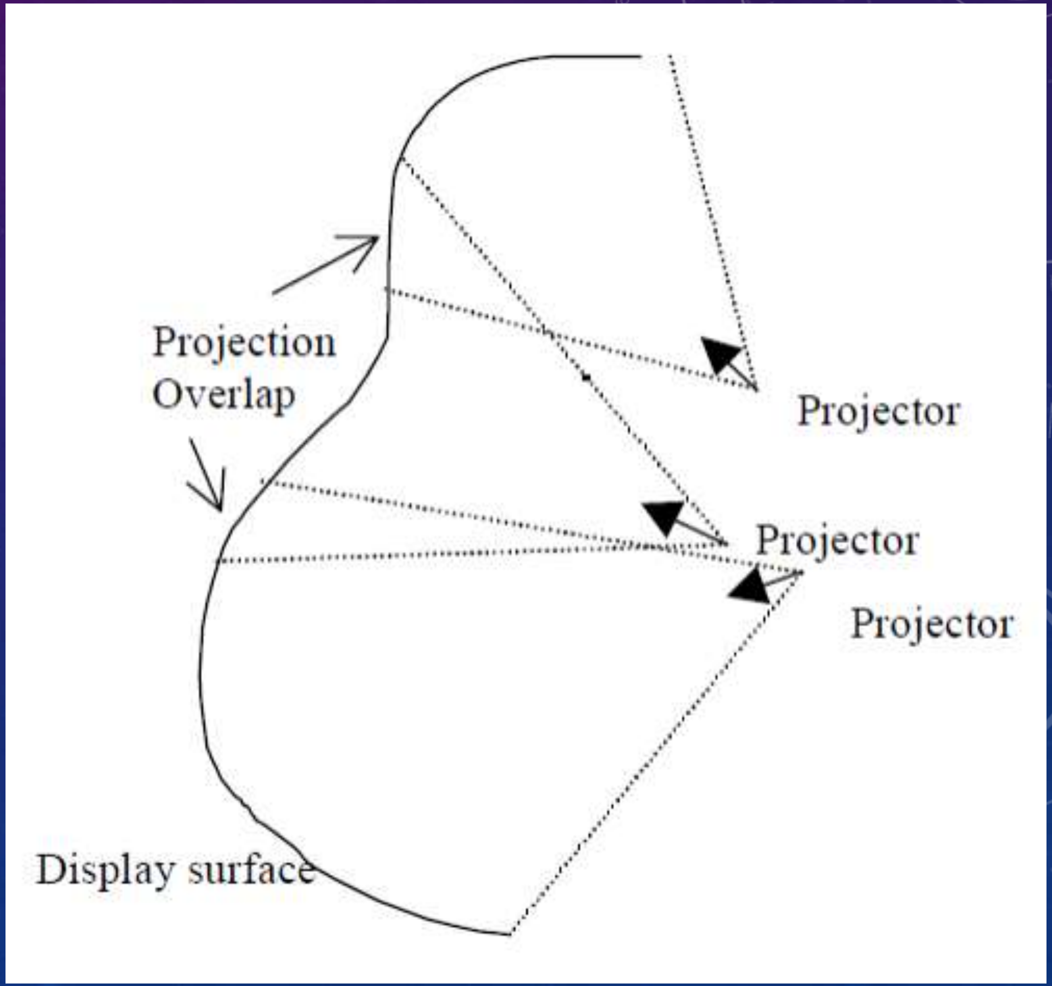


Rigid registration

- f is rigid transformation.
- Special case: $\{p_i \to q_i, i = 1, \ldots n\}$

$$E(R,t) = \sum_{i=1}^{n} \|q_i - Rp_i - t\|^2$$

There is a close-form solution.



Rigid registration

$$E(R,t) = \sum_{i=1}^{n} \|q_i - Rp_i - t\|^2$$

$$\frac{\partial E}{\partial t} = 2\sum_{i=1}^{n} (q_i - Rp_i - t) = 0$$

$$\Rightarrow t = \frac{1}{n}\sum_{i=1}^{n} (q_i - Rp_i)$$

Let $\bar{p}_i = p_i - \frac{1}{n}\sum_{i=1}^{n} p_i$

$$\bar{q}_i = q_i - \frac{1}{n}\sum_{i=1}^{n} q_i$$

$$E(R) = \sum_{i=1}^{n} \|\bar{q}_i - R\bar{p}_i\|^2$$

$$= \|\bar{Q} - R\bar{P}\|_F^2$$

$$= tr\big((\bar{Q} - R\bar{P})^T(\bar{Q} - R\bar{P})\big)$$

$$= C - 2tr(R\bar{P}\bar{Q}^T)$$

Rigid registration

Let $\bar{P}\bar{Q}^T = USV^T$, as

$$tr(A^TB)^2 \leq tr(A^TA)tr(B^TB)$$

Then

$$tr(RUSV^T)^2 = tr(SV^TRU)^2$$

$$\leq tr(SS^T) \leq tr(S)^2$$

Minimizer $R = VU^T$

Let $\bar{p}_i = p_i - \frac{1}{n}\sum_{i=1}^{n} p_i$

$$\bar{q}_i = q_i - \frac{1}{n}\sum_{i=1}^{n} q_i$$

$$E(R) = \sum_{i=1}^{n} \|\bar{q}_i - R\bar{p}_i\|^2$$

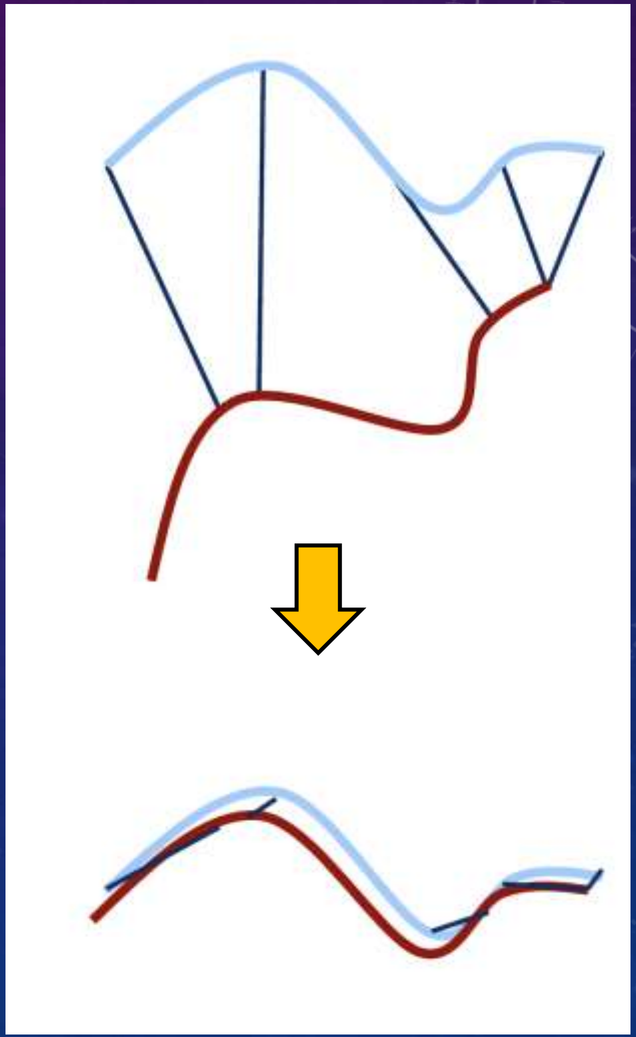
$$= \|\bar{Q} - R\bar{P}\|_F^2$$

$$= tr\left((\bar{Q} - R\bar{P})^T(\bar{Q} - R\bar{P})\right)$$

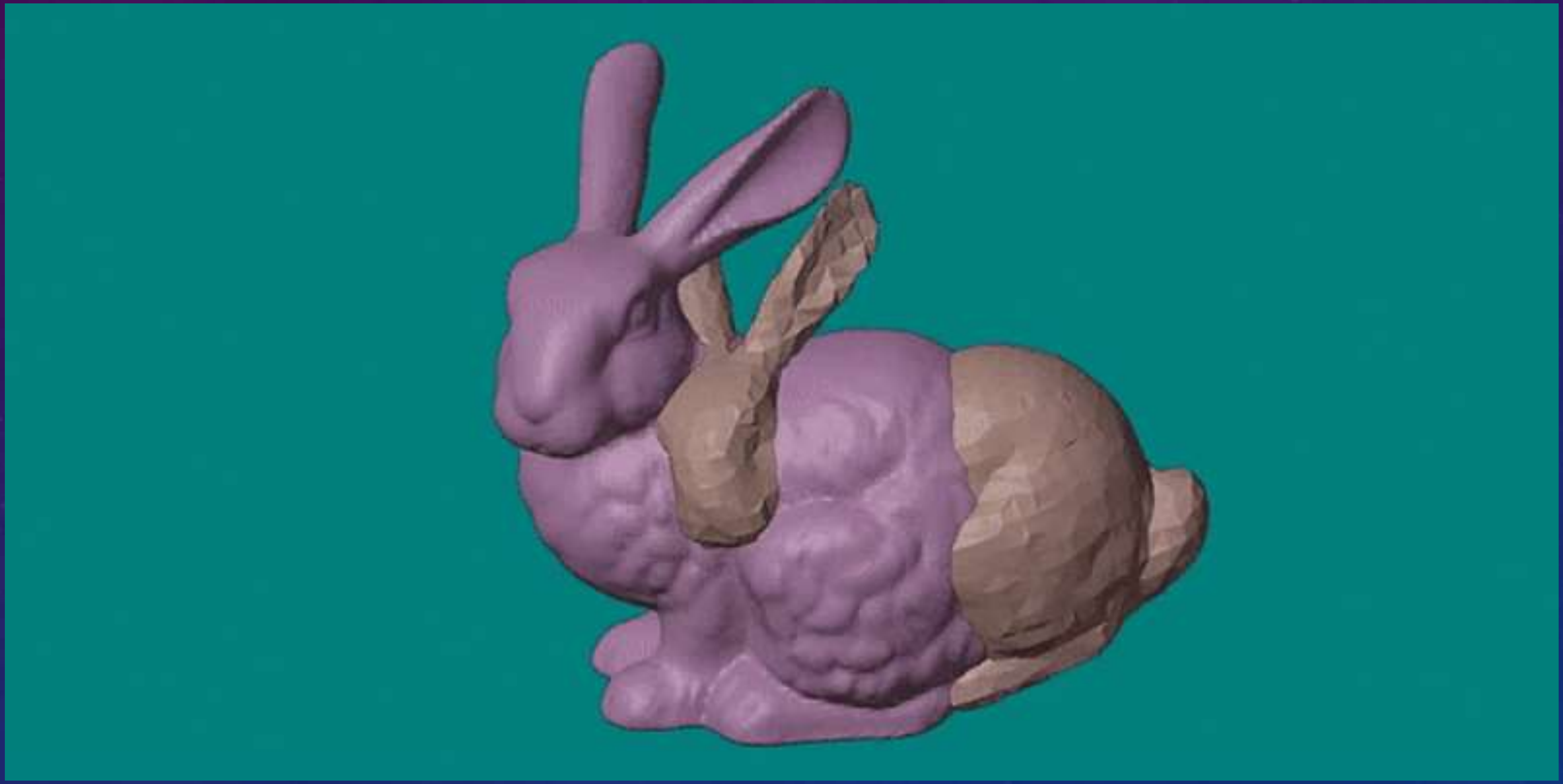
$$= C - 2tr(R\bar{P}\bar{Q}^T)$$

Rigid registration

- f is rigid transformation.
- Iterative close-point (ICP):
  - Identify nearest points
  - Compute the optimal (R, t)
  - Repeat until E is small

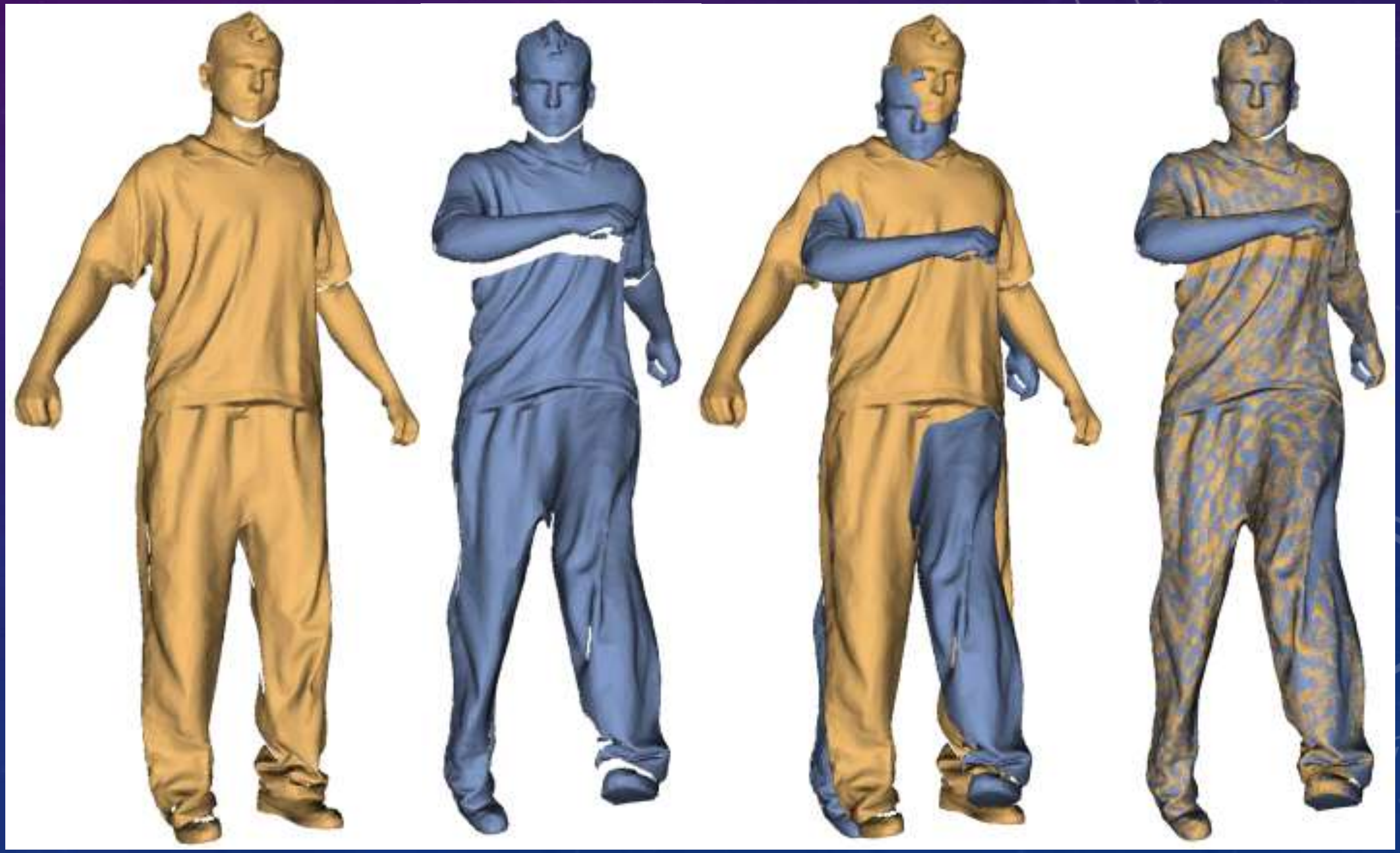


Rigid registration



Non-rigid registration

- f is non-rigid.
- Deformation fields:
  - Rigid locally



Non-rigid registration

- f is non-rigid.
- Deformation fields:
  - Rigid locally
  - Interpolation

Deformation graph

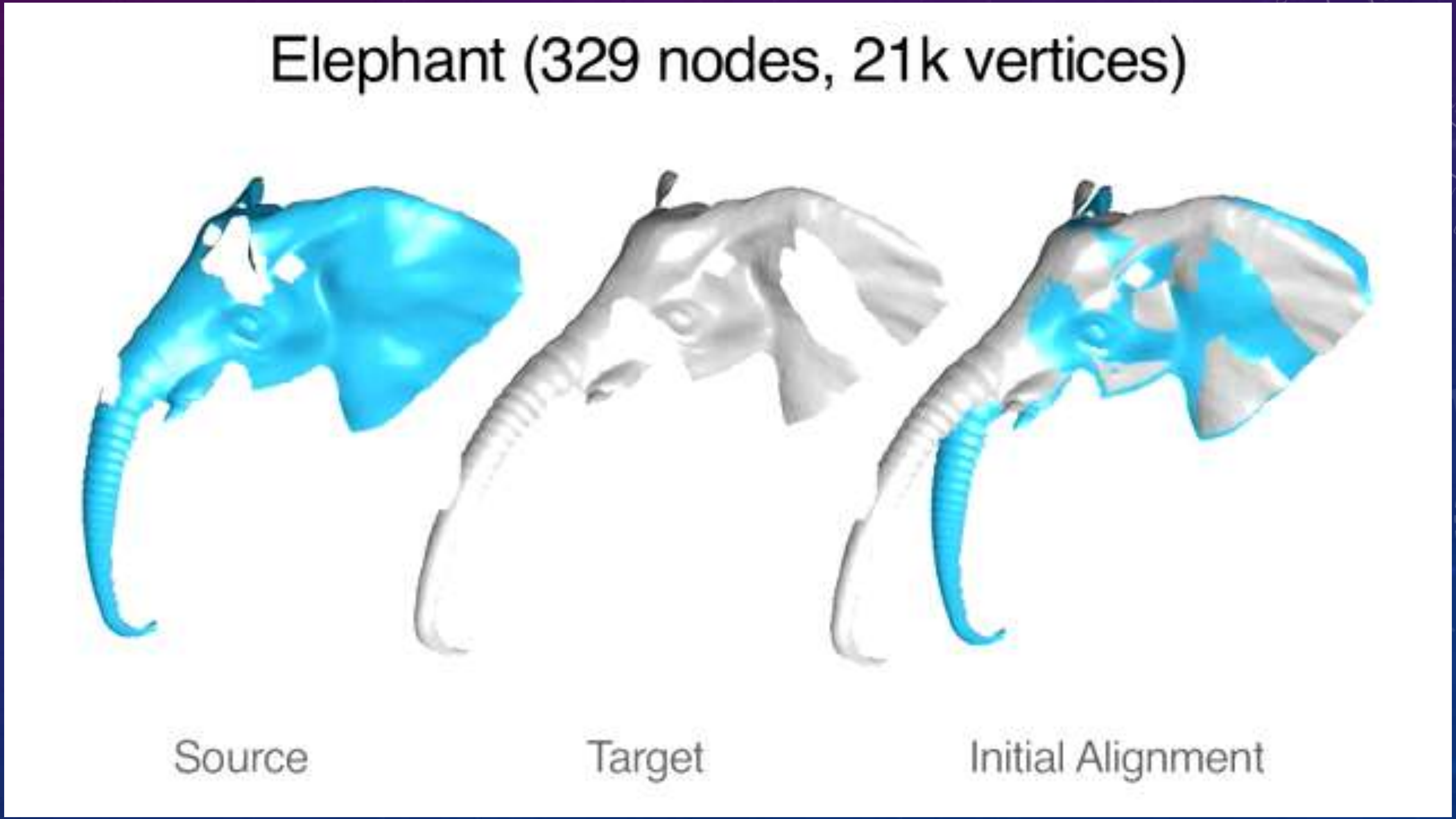
Graph node:

$$\hat{x} - \widehat{p_i} = A_i(x - p_i) + t_i,$$

affine matrix $A_i \in \mathbb{R}^{3\times3}$.

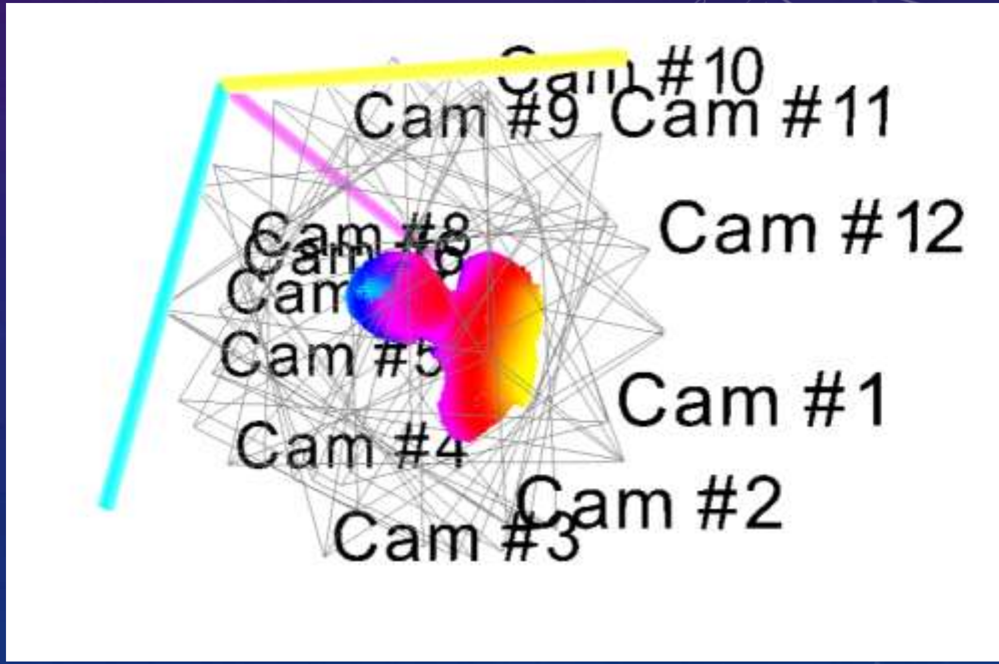
$$x = \sum_{p_i \in \mathcal{N}(x)} w_i(x) \times (A_i(x - p_i) + t_i + \widehat{p_i})$$

Non-rigid registration



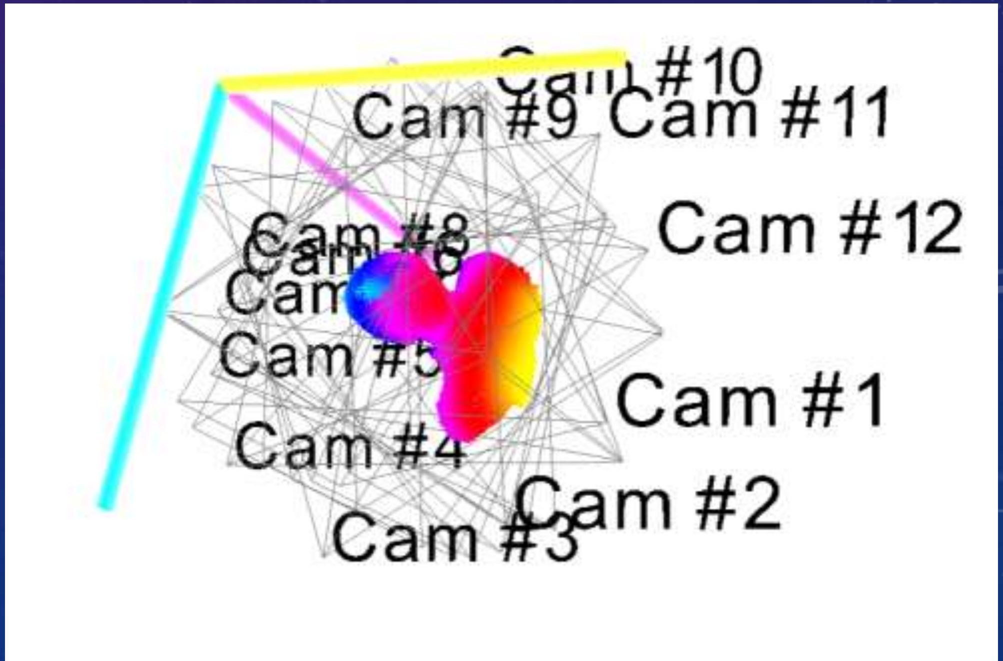
Global Registration

- Given: n scans around an object
- Goal: align them all
- First attempt: ICP each scan to one other



Global Registration

- Want method for distributing accumulated error among all scans
- Methods:
  - Set "anchor" scan - one scan covers most of surface
  - Align each new scan to all previous scans



Global Registration

- Want method for distributing accumulated error among all scans
- Methods:
  - Brute-Force Solution

While not converged:

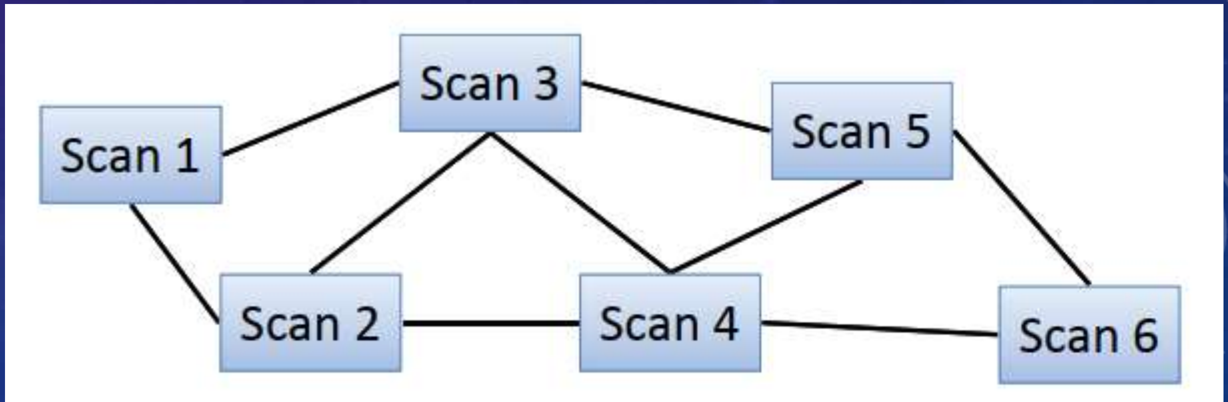
- For each scan:
  - For each point:
    - For every other scan
      - » Find closest point
- Minimize error w.r.t. transforms of all scans

this scan

Global Registration

- Want method for distributing accumulated error among all scans
- Methods:
  - Brute-Force Solution
  - Graph Methods

Find transformations consistent as possible with all pairwise ICP



Reconstruction

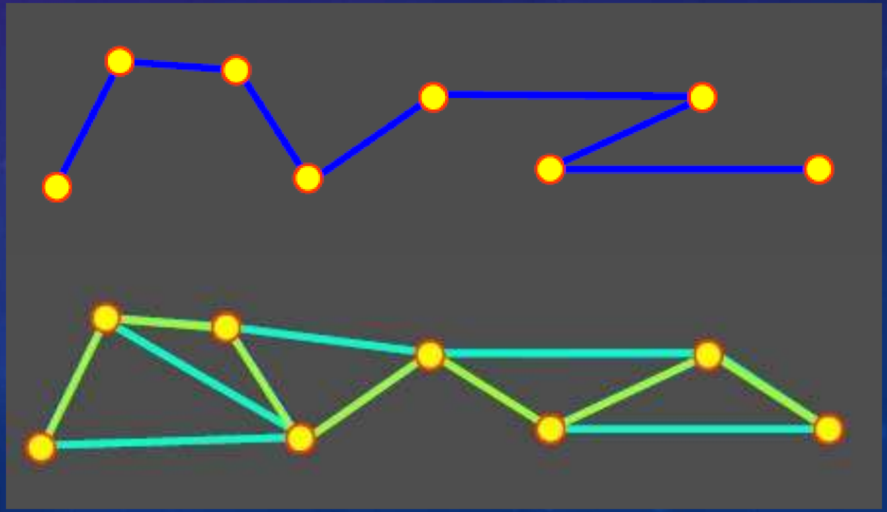
Reconstruction methods

- Explicit methods
  - VD and DT
  - …
- Implicit methods

A new Voronoi-based surface reconstruction algorithm

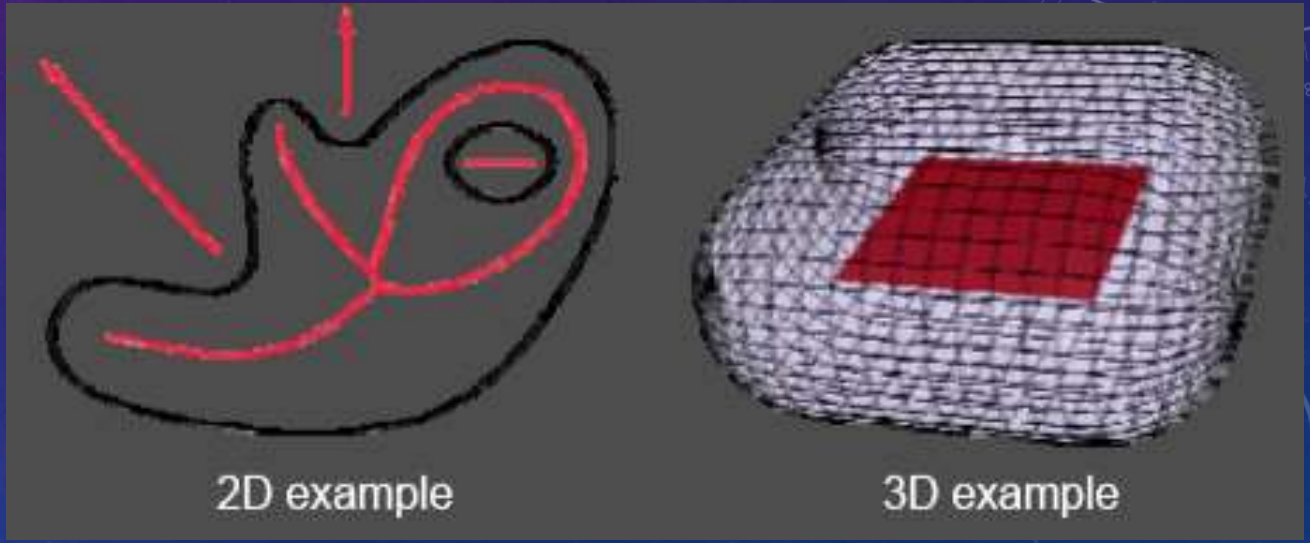
Delaunay triangulation

- 2D case
  - Curve from Points
  - Which edges to choose?



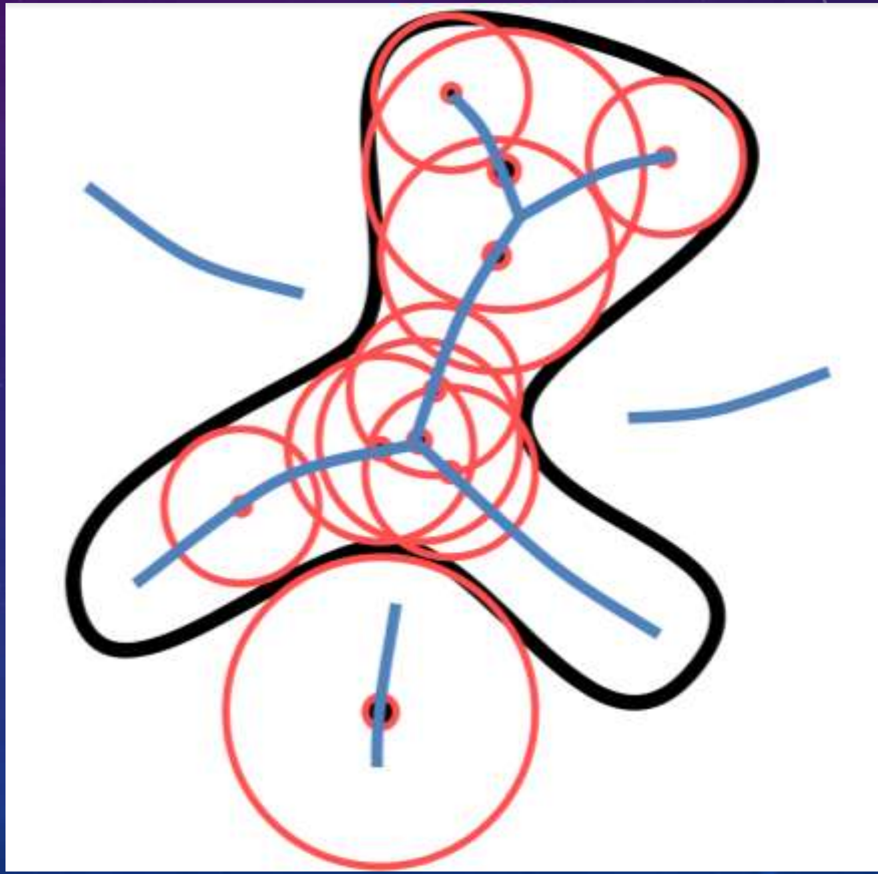
Medial Axis

- Set of points with more than one closest point on the surface.



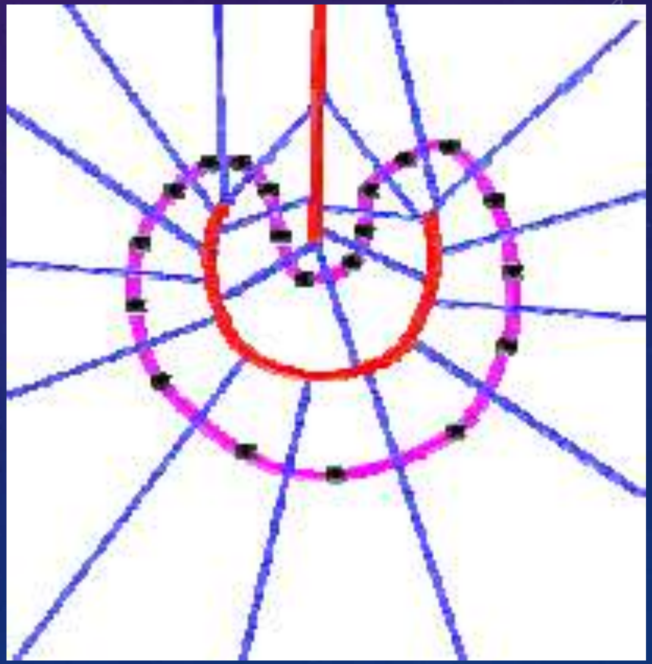
Medial Axis

- Set of points with more than one closest point on the surface.
- Locus of centers of tangentially touch the curve in at least 2 points.



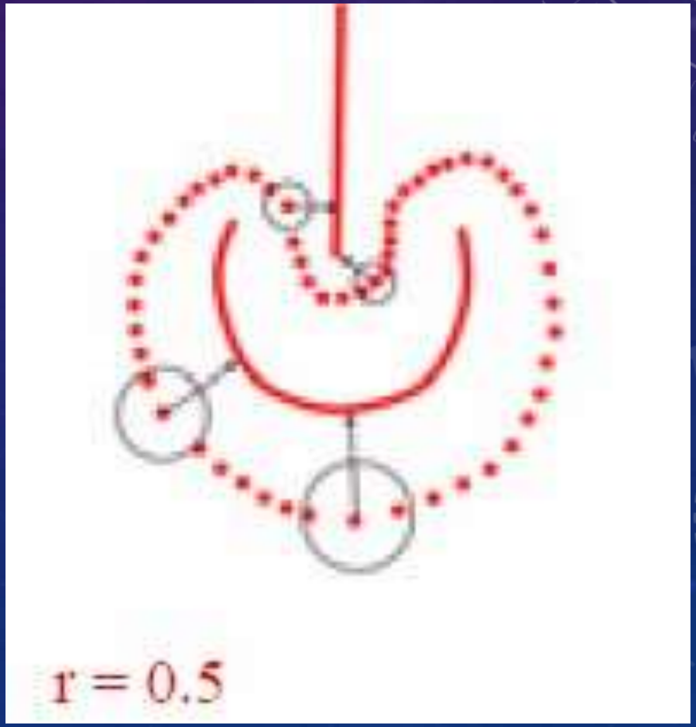
Medial Axis and VD

- Voronoi diagram of set of points on curve approximates Medial if points sampled densely enough.



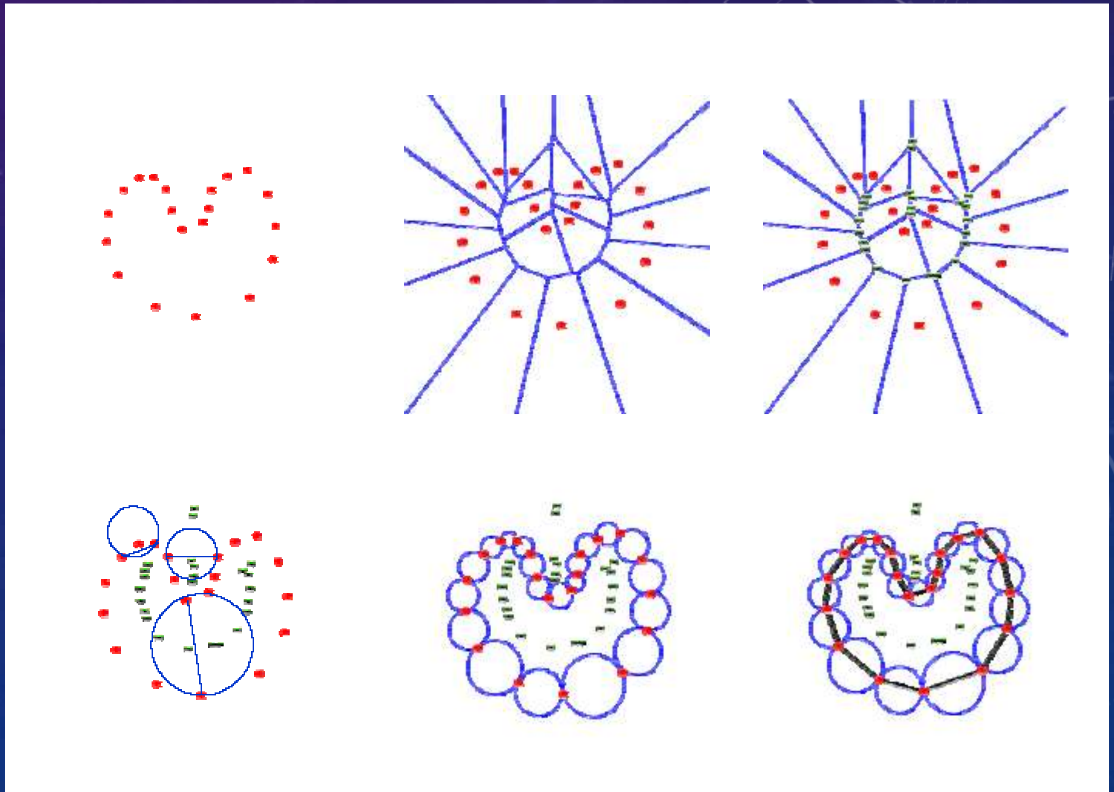
Medial Axis and VD

- Voronoi diagram of set of points on curve approximates Medial if points sampled densely enough.
- r-sample : distance from any point on surface to nearest sample point ≤ r × distance from point to medial axis



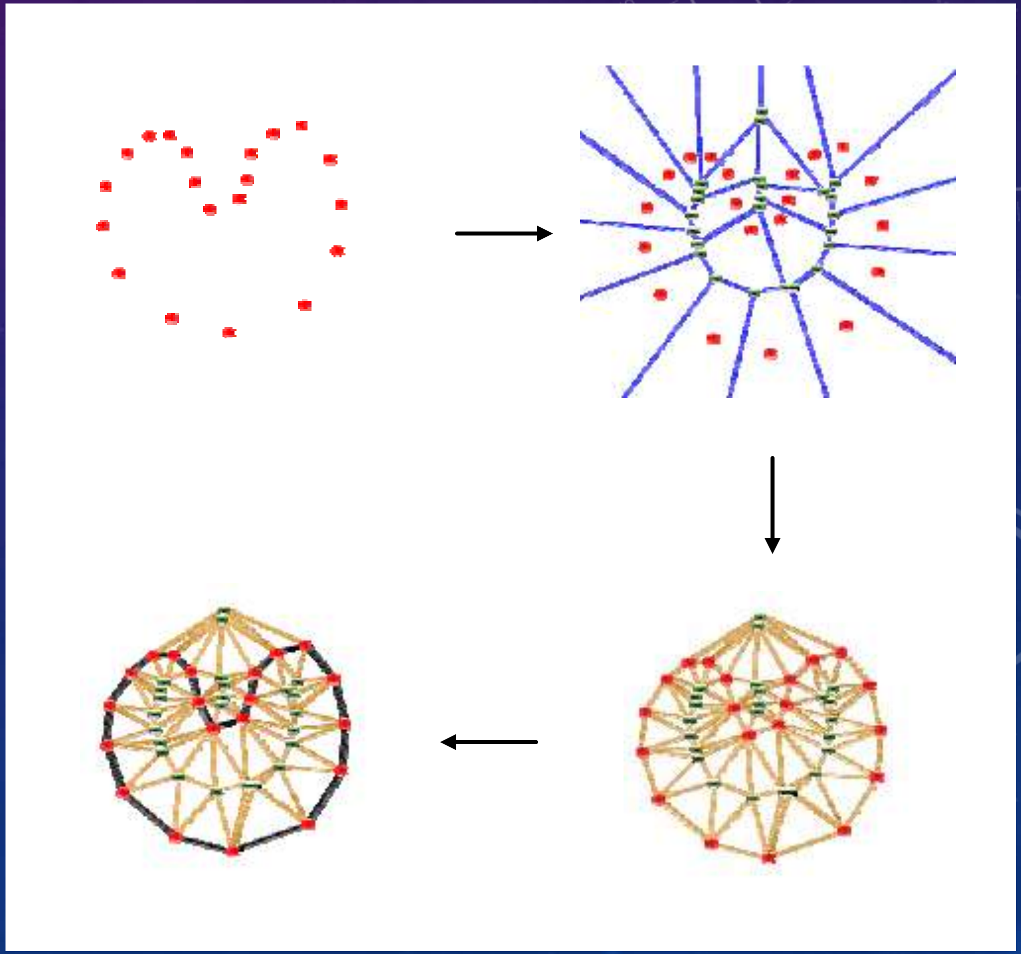
Idea

- Adopt Delaunay edges which are “far” from Media Axis
- To represent Media Axis use Voronoi vertices
- Edge e in crust <=> circumcircle of e contains no other sample points or Voronoi vertices of S



2D Crust algorithm

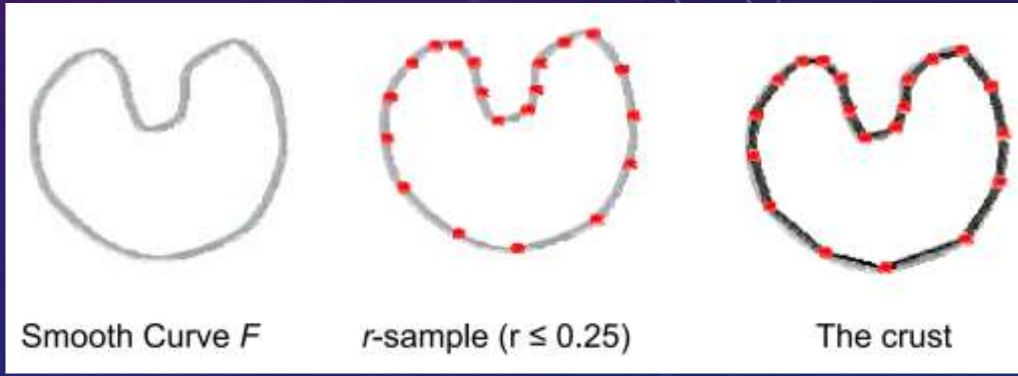
- Compute Voronoi diagram of S and V is the set of Voronoi vertices.
- Compute Delaunay triangulation of SUV.
- Return all Delaunay edges between points of S.



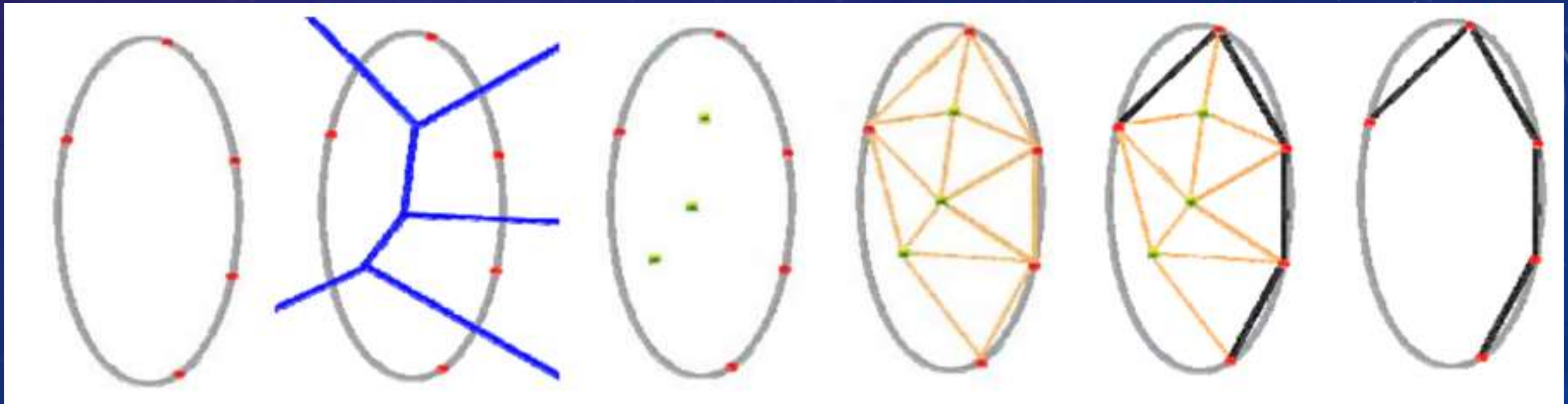
Theory

- Theorem:

The crust of an r-sample from a smooth curve F, for r ≤ 0.25 connects only adjacent samples of F.

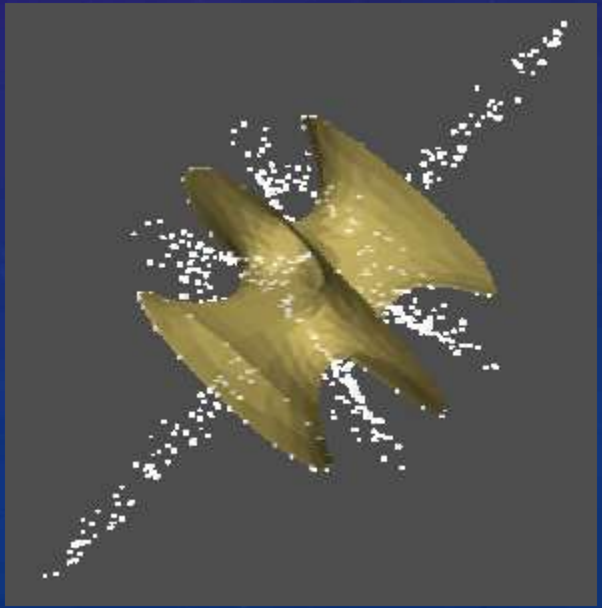


- If r is large



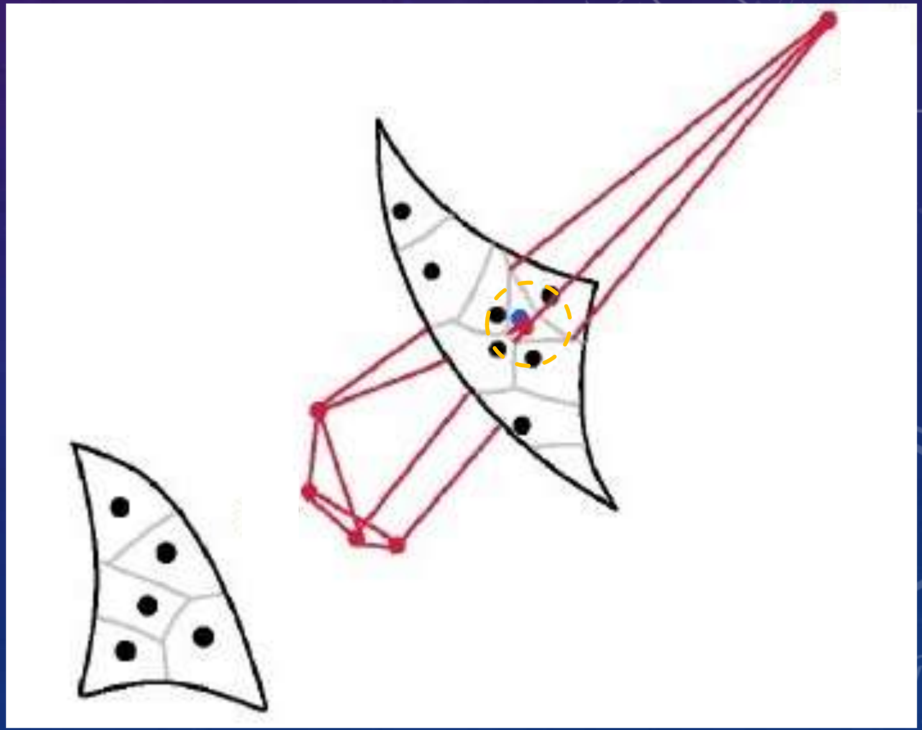
Delaunay triangulation

- 2D case
  - Curve from Points
  - Which edges to choose?
- 3D case
  - Shell from points



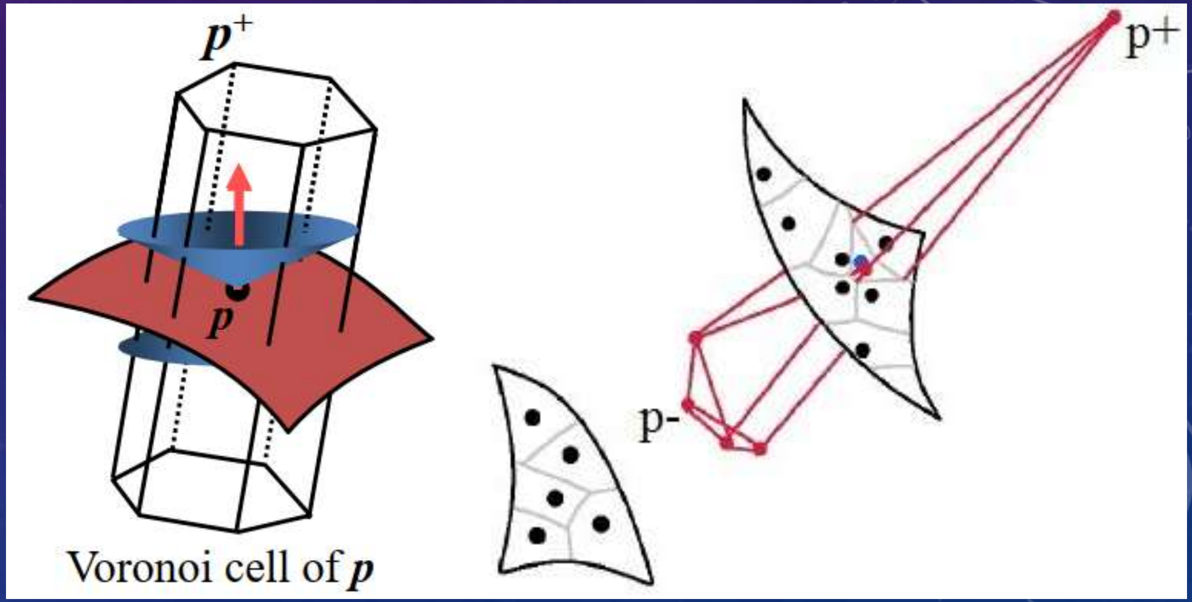
Differences between 2D and 3D

- In 3D Voronoi cells are polyhedral
- In 3D Voronoi vertex is equidistant from 4 sample points.
- In 3D not all Voronoi vertices are near medial axis (regardless of sampling density)



Observation

- Some vertices of the Voronoi cell are near medial axis.
- Poles-two farthest vertices of Vs $(p^+(s), p^-(s))$ - one on each side of the surface.

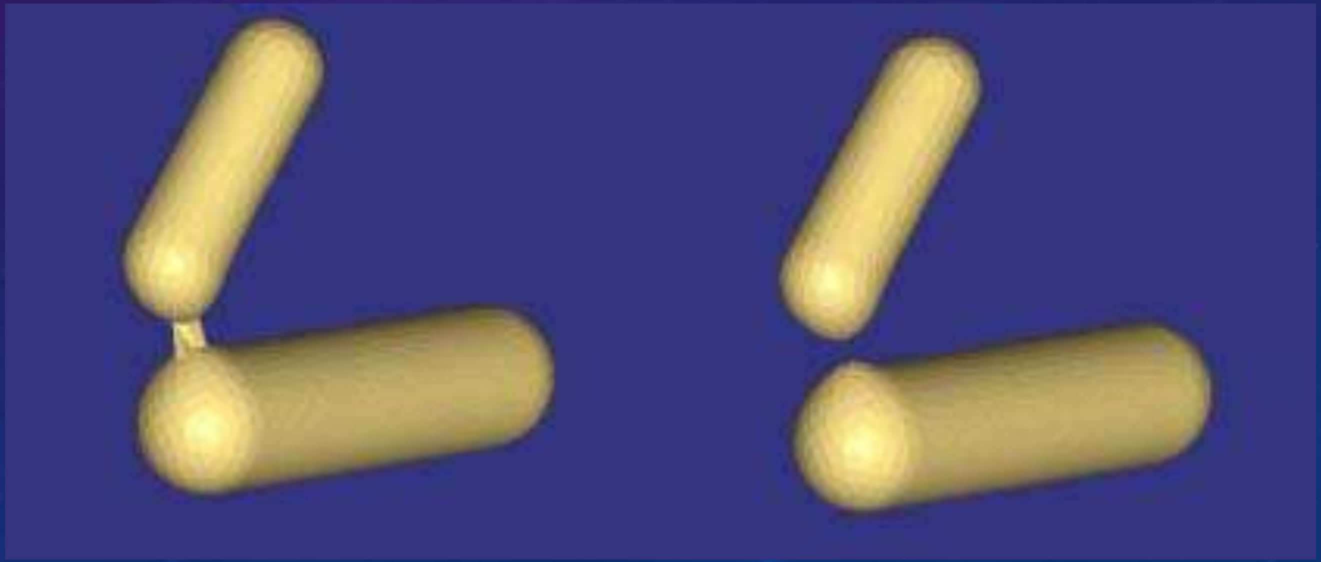


3D Crust algorithm

- Compute Voronoi diagram of S
- For each s ∈ S, identify the poles $p^+(s)$ and $p^-(s)$
  - $p^+(s)$ is the vertex of Vs most distant from s
  - $p^-(s)$ is the vertex of Vs most distant from s in the opposite direction
- Let P be the set of all poles and compute Delaunay triangulation T of S U P
- Add to crust all triangles in T with vertices only in S

Post-processing

- Delete triangles whose normals differ too much from the direction vectors from the triangle vertices to their poles



Problems & Limitations

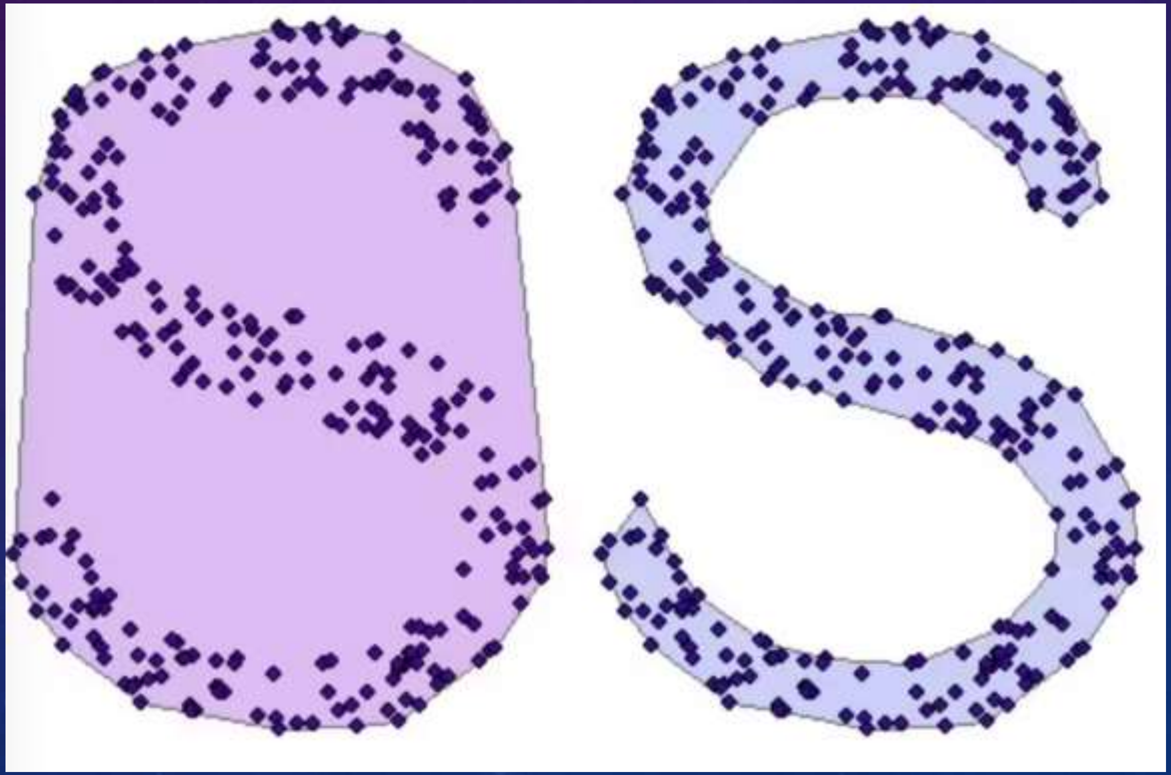
- Sampling of points needs to be dense –Undersampling causes holes
- Problems at sharp corners
- Heuristically choosing poles
- Algorithm is slow

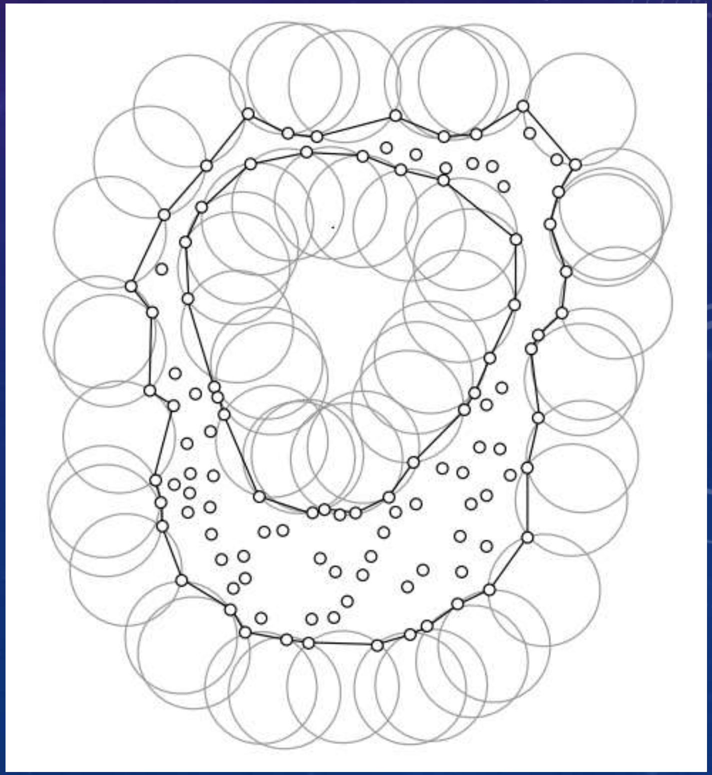
Reconstruction methods

- Explicit methods
  - VD and DT
  - Alpha shape
  - …
- Implicit methods

Alpha shape

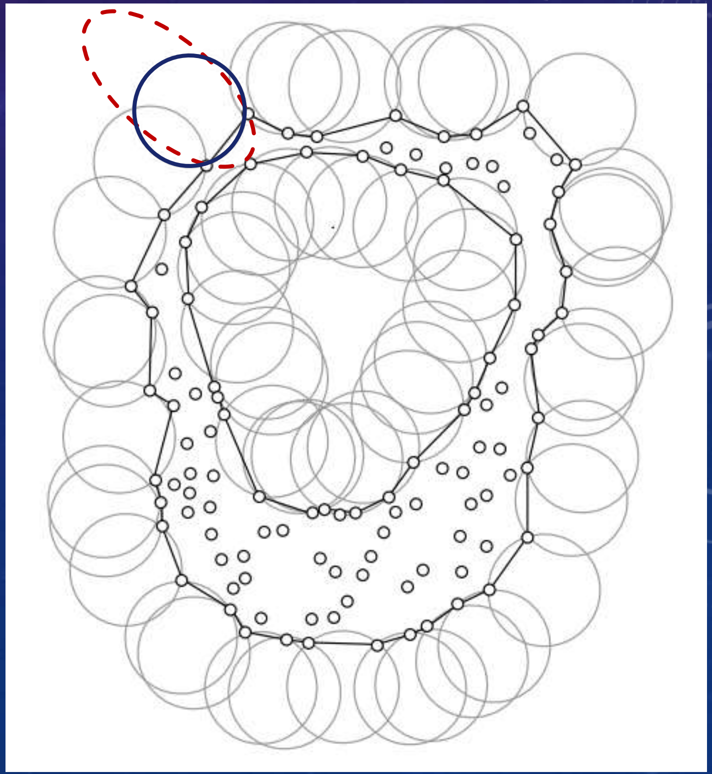
- Convex hull V.S. alpha shape





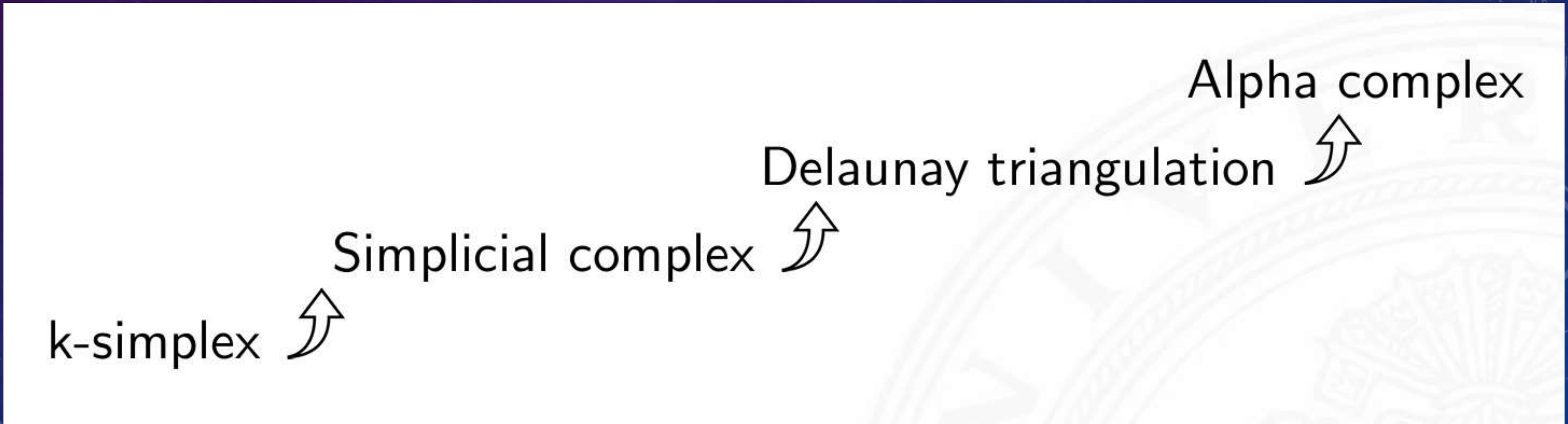
Alpha shape

- Ice cream with solid chocolate chips
- Spherical ice spoon
- Curve out all parts of the ice cream with out touching the chocolate chips
- Straighten all curvatures



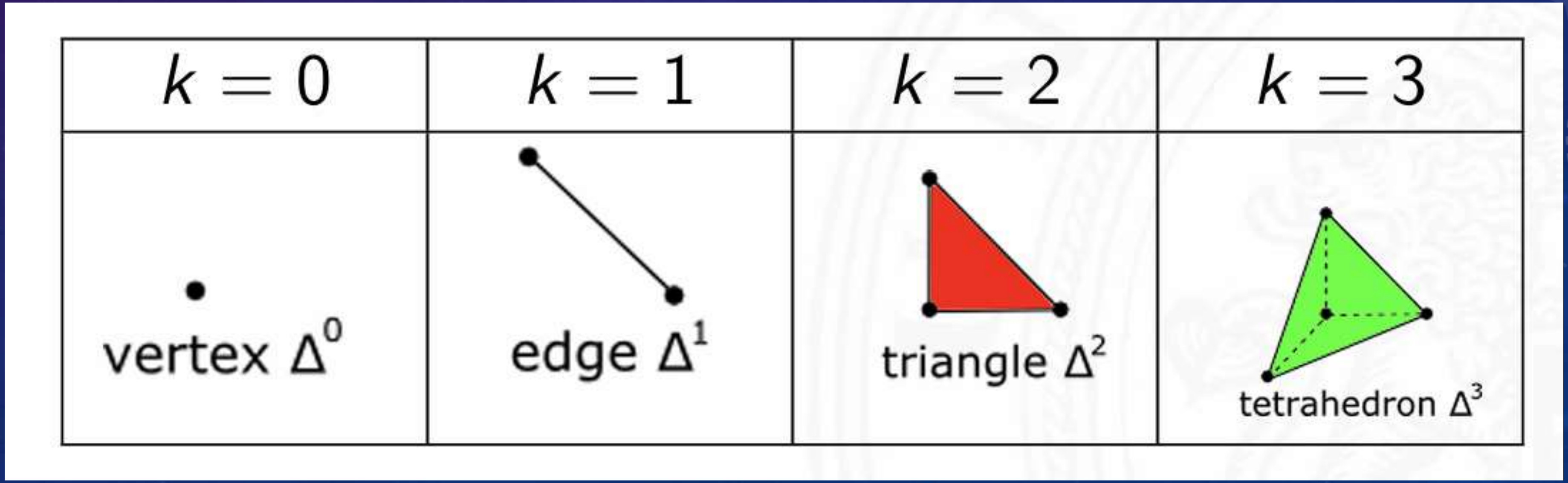
Alpha shape

- 2D case -> 3D case



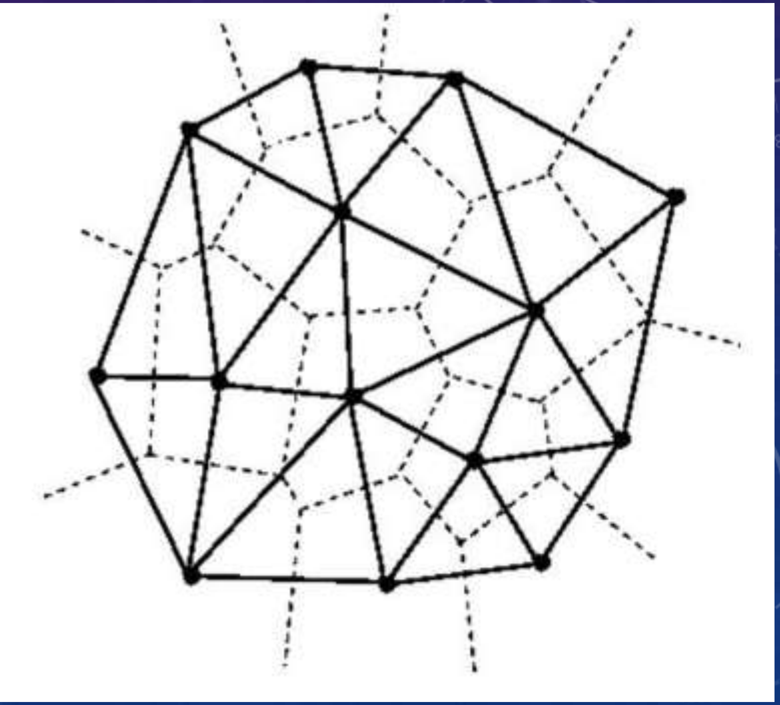
k – simplex

- k – simplex Δ_S : the convex hull of S , for any subset $S \subseteq P$ of size $|S| = k + 1$
- The general position assumption : k – simplex Δ_S has exactly dimension k



Simplicial complex

- A collection C of simplices forms a simplicial complex if it satisfies the following conditions :
  - For a simplex Δ_S of C, the boundary simplices of Δ_S are in C.
  - For two simplices of C, their intersection is either $\emptyset$ or a simplex in C



Alpha shape

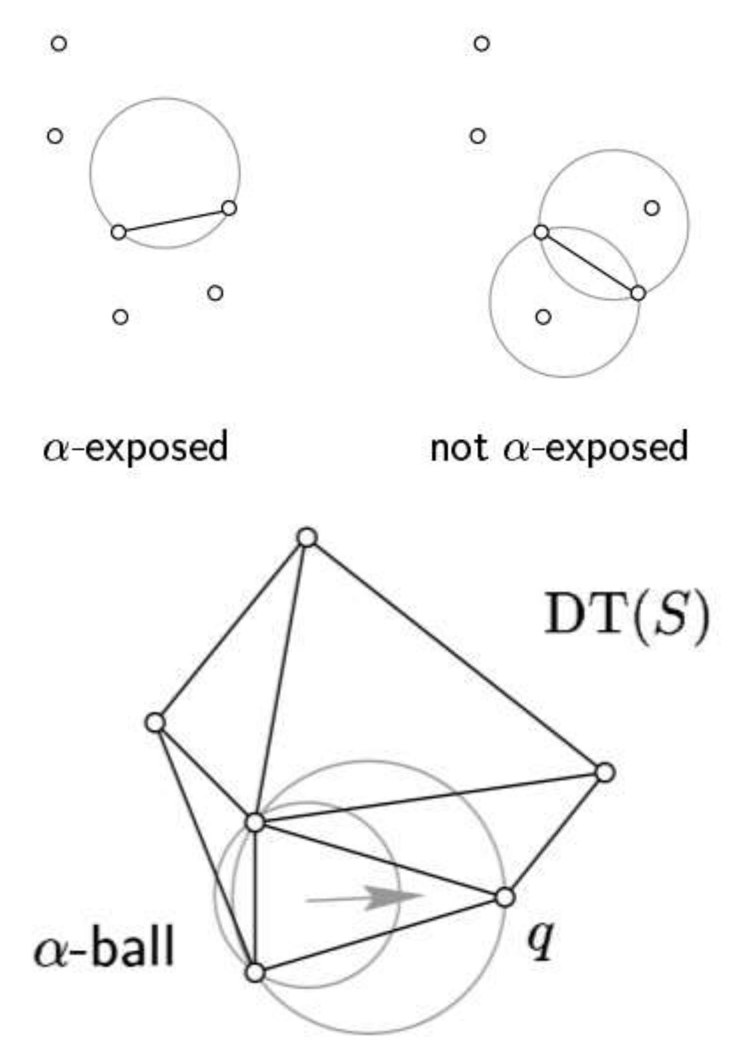
- r-ball : an open ball with radius r
  - 0-ball : point
  - ∞-ball : open half-space
- For given point set P, r-ball b is empty if $b \cap P = \emptyset$

α-exposed

- A $k-$simplex Δ_S is α-exposed if there exists an empty α-ball b with $S = \partial b \cap P$
- If Δ_S is an α-exposed simplex of P, then $\Delta_S \in DT(P)$.

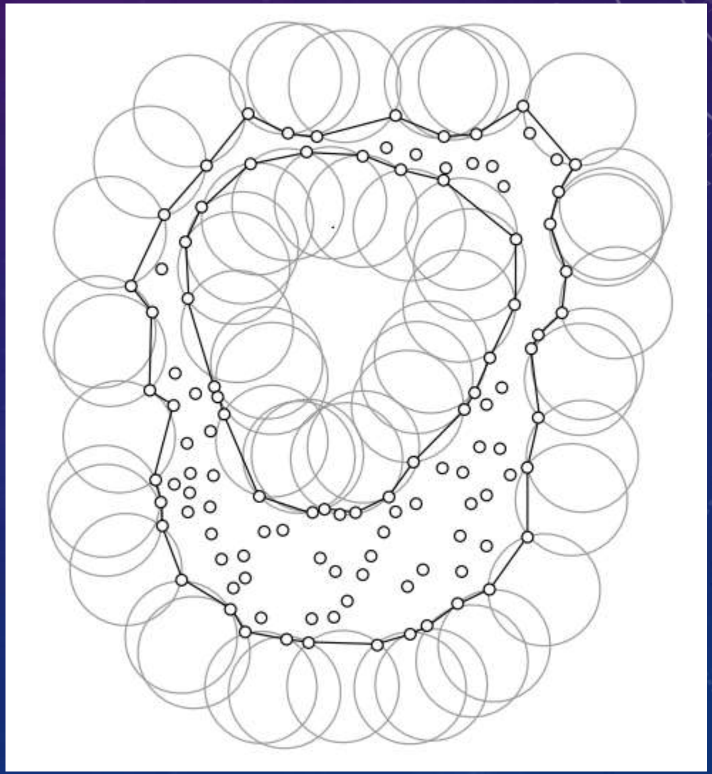
For $d = 2$, circumsphere of S

For $d < 2$, increase α until meet other point.



α-exposed

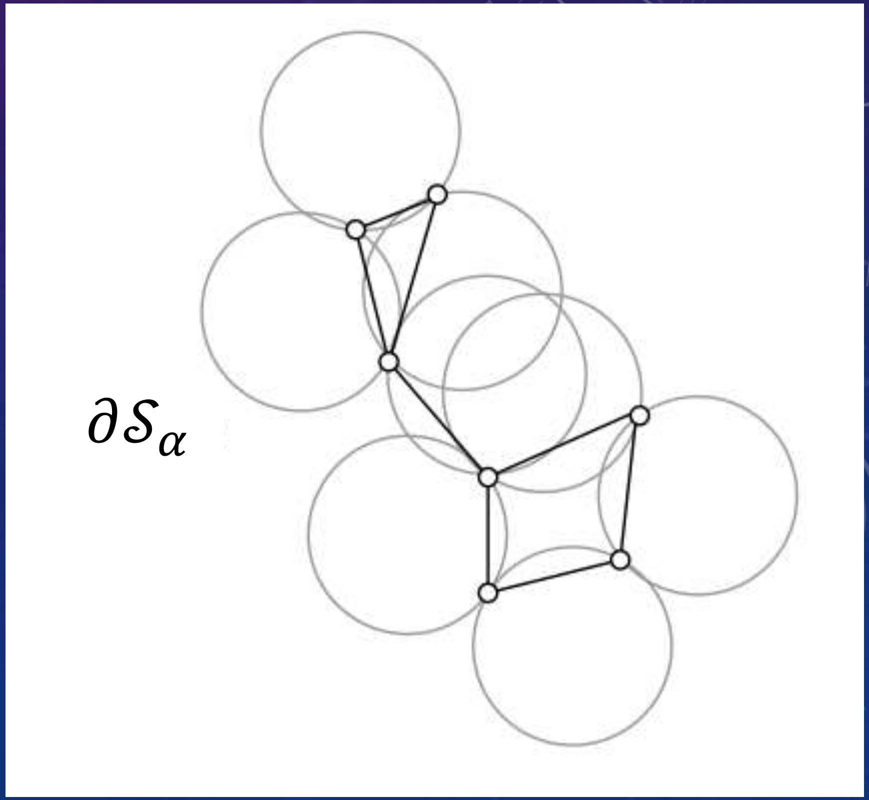
- Ice-cream spoon hits against one or more of the points in P $\rightarrow$ the simplex spanned by these points is α-exposed



α-shape

- The boundary $\partial\mathcal{S}_\alpha$ of the α-shape of the point set P consists of all k– simplex of P for $0 \ \leq \ k \ < d$ which are α-exposed

$$\partial\mathcal{S}_\alpha = \{\Delta_S | S \subseteq P, |S| \leq d \text{ and } \Delta_S \ \alpha\text{–exposed}\}$$

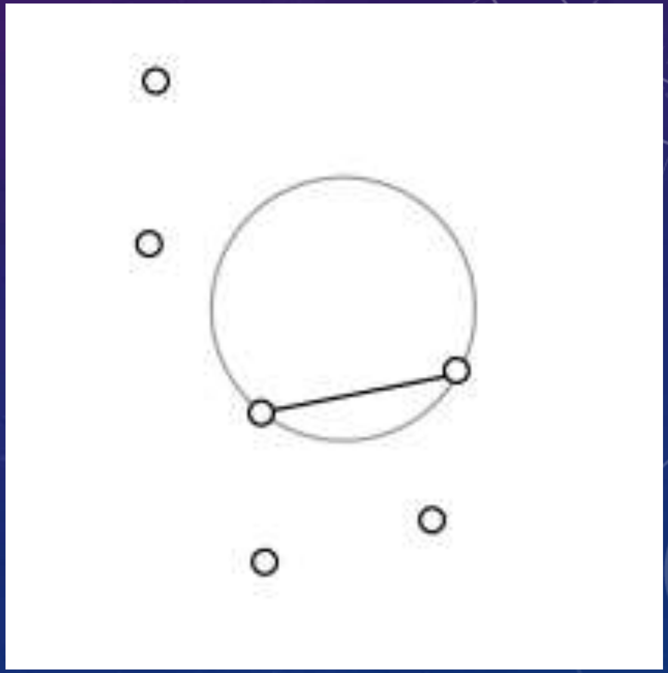


Property

- $\lim_{\alpha \to 0} \partial \mathcal{S}_\alpha = P, \quad \lim_{\alpha \to \infty} \partial \mathcal{S}_\alpha = \partial conv(P)$

  $\Longrightarrow \lim_{\alpha \to 0} \mathcal{S}_\alpha = P, \lim_{\alpha \to \infty} \mathcal{S}_\alpha = conv(P)$

- For any $0 \leq \alpha \leq \infty$, we have $\partial \mathcal{S}_\alpha \subset DT(P)$



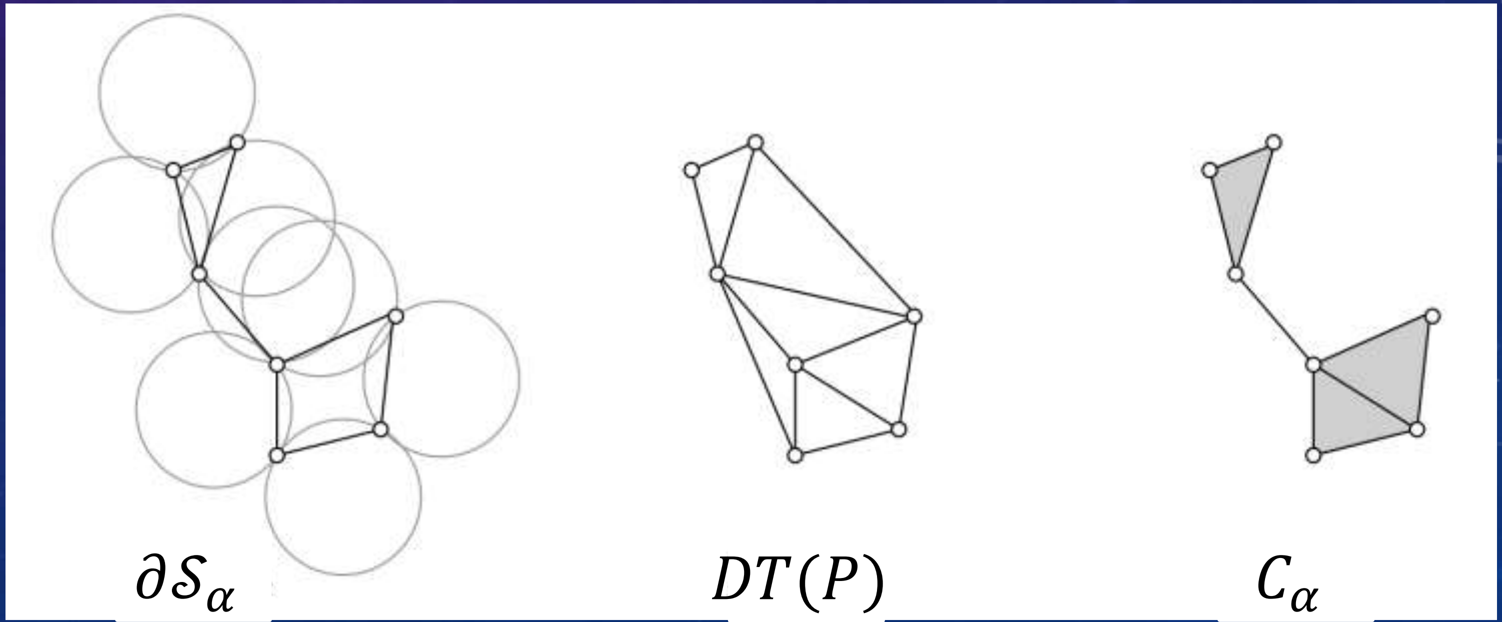
α-complex

- A simplex $\Delta_S \in DT(P)$ is in C_α if

  a) the circumcircle of S with radius $r < \alpha$ is empty or

  b) it is a boundary simplex of a simplex of a)

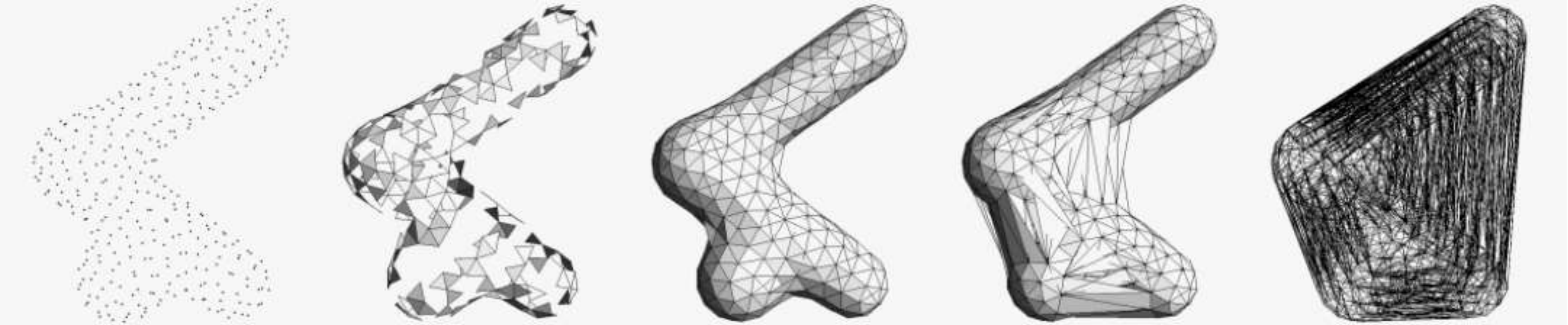
$$\partial\mathcal{S}_\alpha = \partial C_\alpha$$



Algorithm

- Computing the Delaunay triangulation of P, knowing the boundary of α-shape is contained in it.
- Determine C_α by inspecting all simplices $\Delta_S \in DT(P)$. If the circumcircle of S with radius $r < \alpha$ is empty, we accept Δ_S as a member of C_α, together with all its faces.
- All d-simplices of C_α make up the interior of $\mathcal{S}_\alpha(P)$ and all simplices on the boundary of ∂C_α form $\partial \mathcal{S}_\alpha$

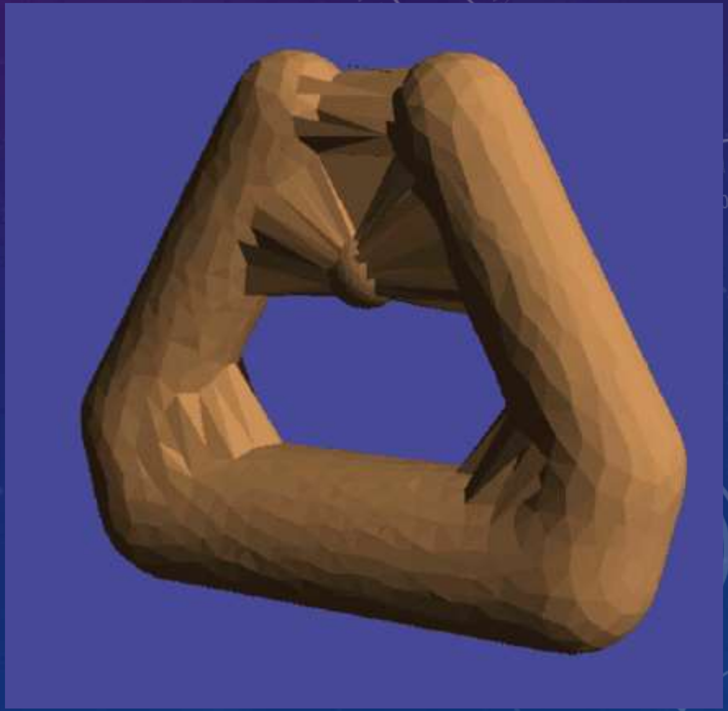
Family of α



$$\alpha = \{0, 0.19, 0.25, 0.75, \infty\}$$

Problems & Limitations

- Choosing the ”best” α value is not trivial → some heuristical methods
- Not for all object’s surfaces there is a good α value due to non-uniformly sampled data
  - Interstices might be covered
  - Neighboring objects might be connected
  - Joints or sharp turns might not be sharp anymore



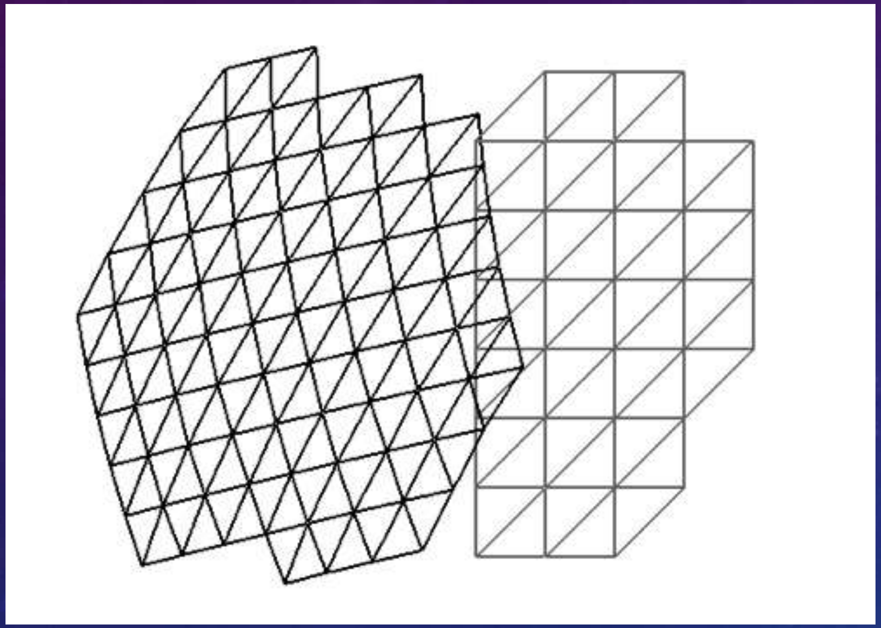
Reconstruction methods

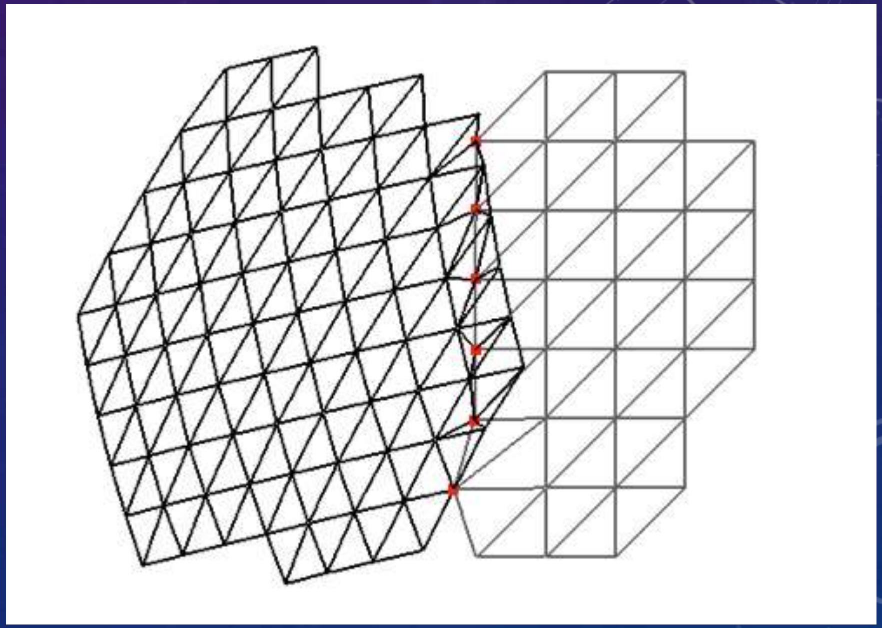
- Explicit methods
  - VD and DT
  - Alpha shape
  - Zippering range scans
  - …
- Implicit methods

Idea

- Use range scanner properties for reconstruction
- Single scan from given direction produces regular lattice of points in X and Y with changing depth (Z).
- Take multiple scans to create complete model

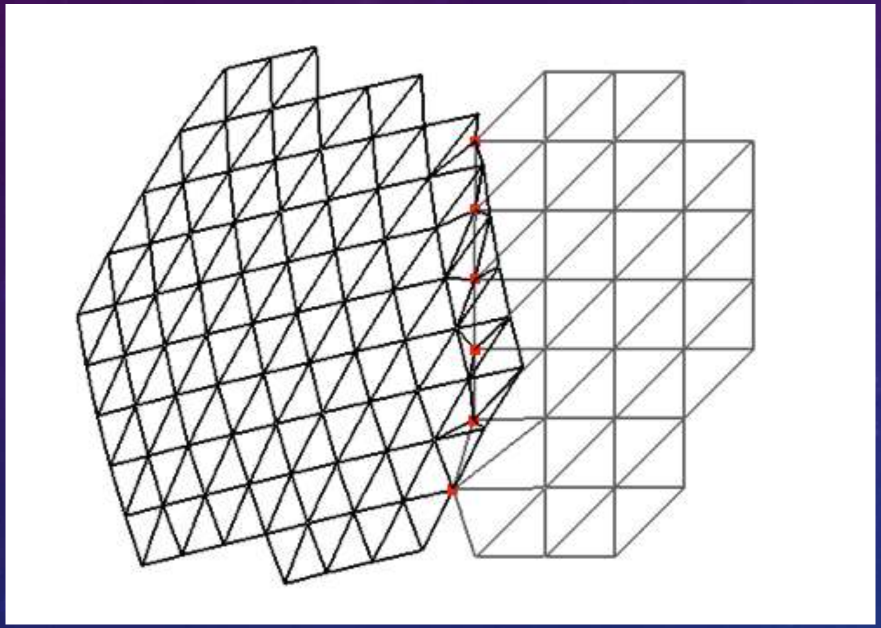
Zippering range scans

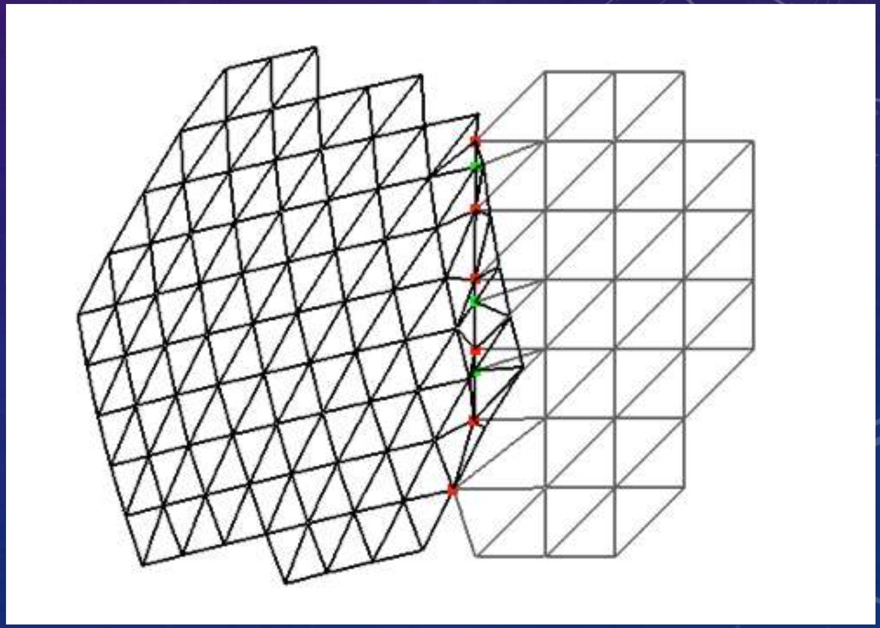




Project & insert boundary vertices

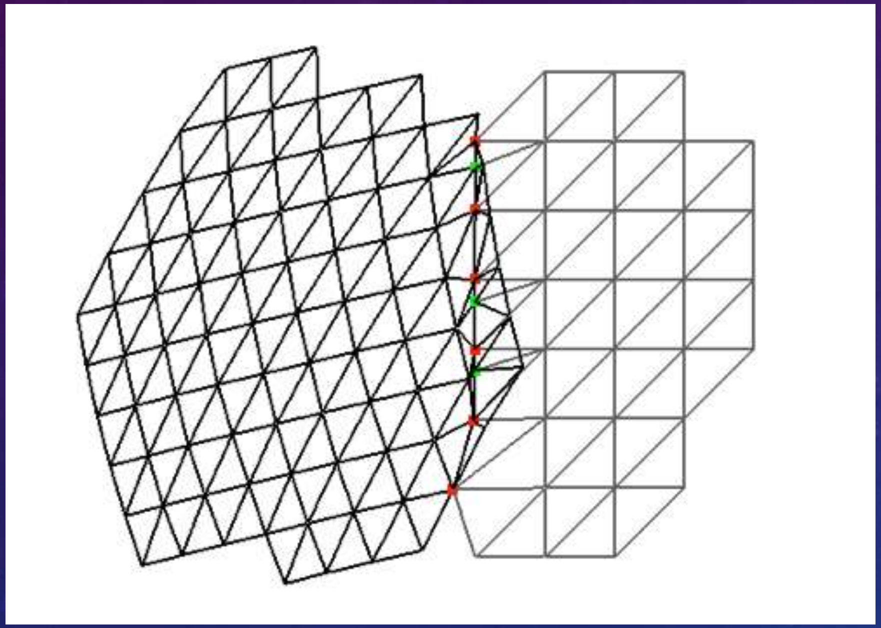
Zippering range scans

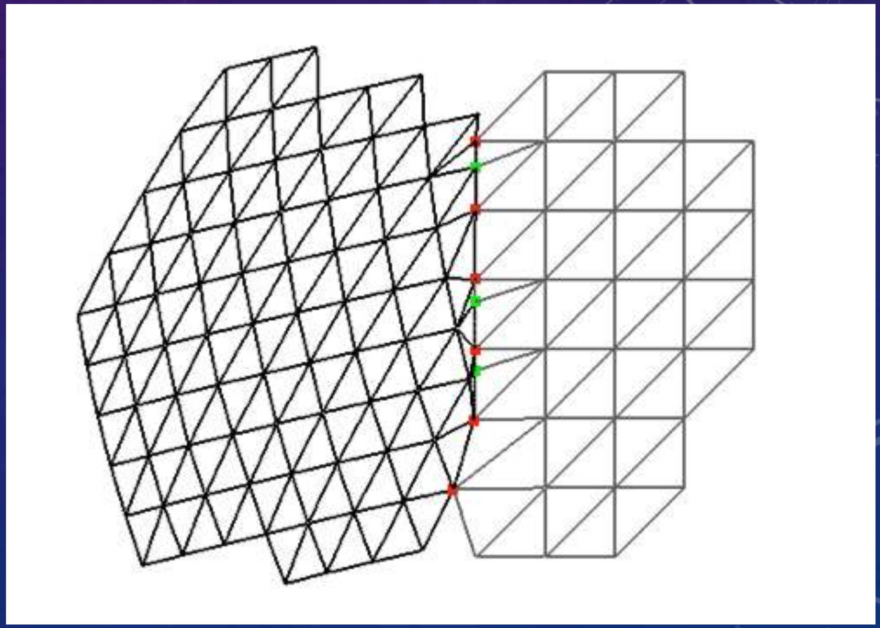




Intersect boundary edges

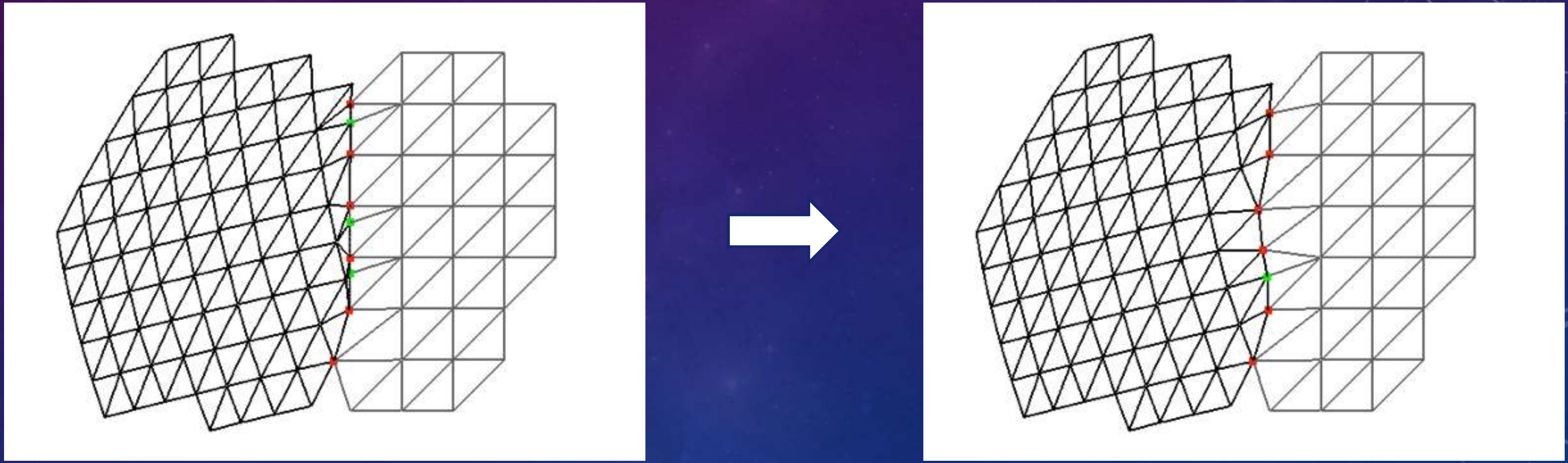
Zippering range scans





Discard overlap region

Zippering range scans



Locally optimize triangulation

Problems & Limitations

- Pros:
  - Preserves regular structure of each scan
  - Fast, no additional data structures
- Cons:
  - Lot of small “fixes” / “tricks”
  - Problems with complex, noisy, incomplete data