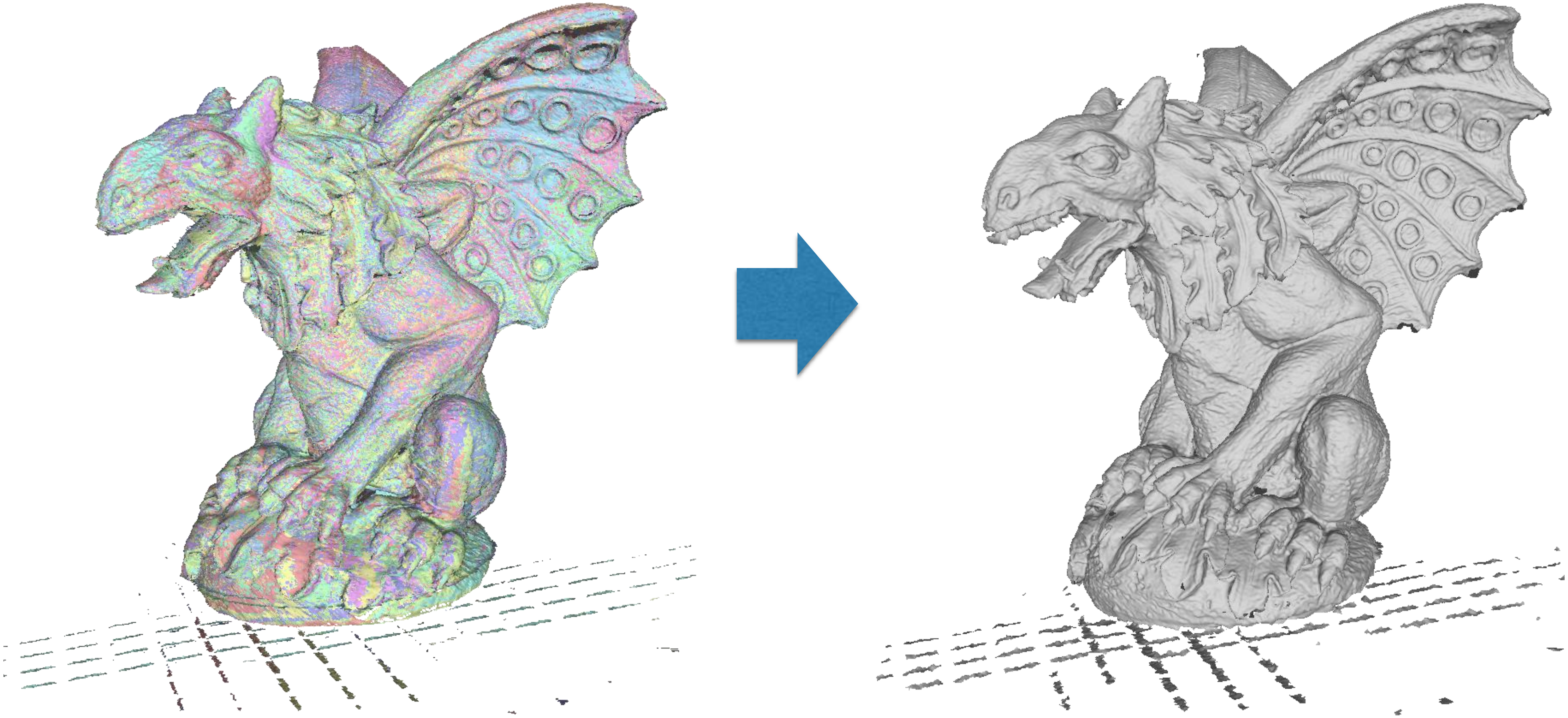
Surface Reconstruction

Gianpaolo Palma

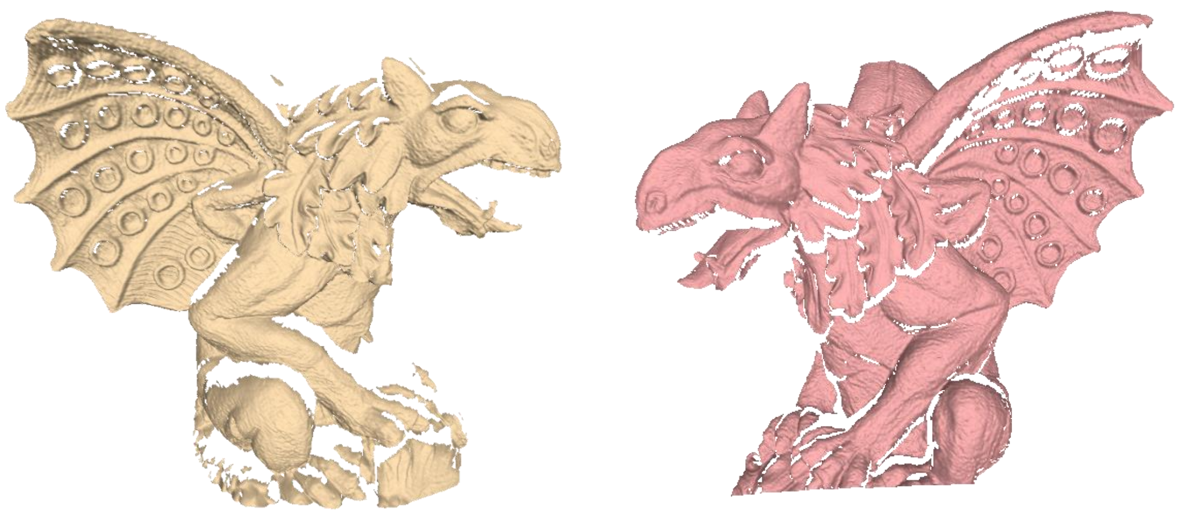
Surface reconstruction



Input

- **Point cloud**
  - With or without normals
  - Examples: multi-view stereo, union of range scan vertices

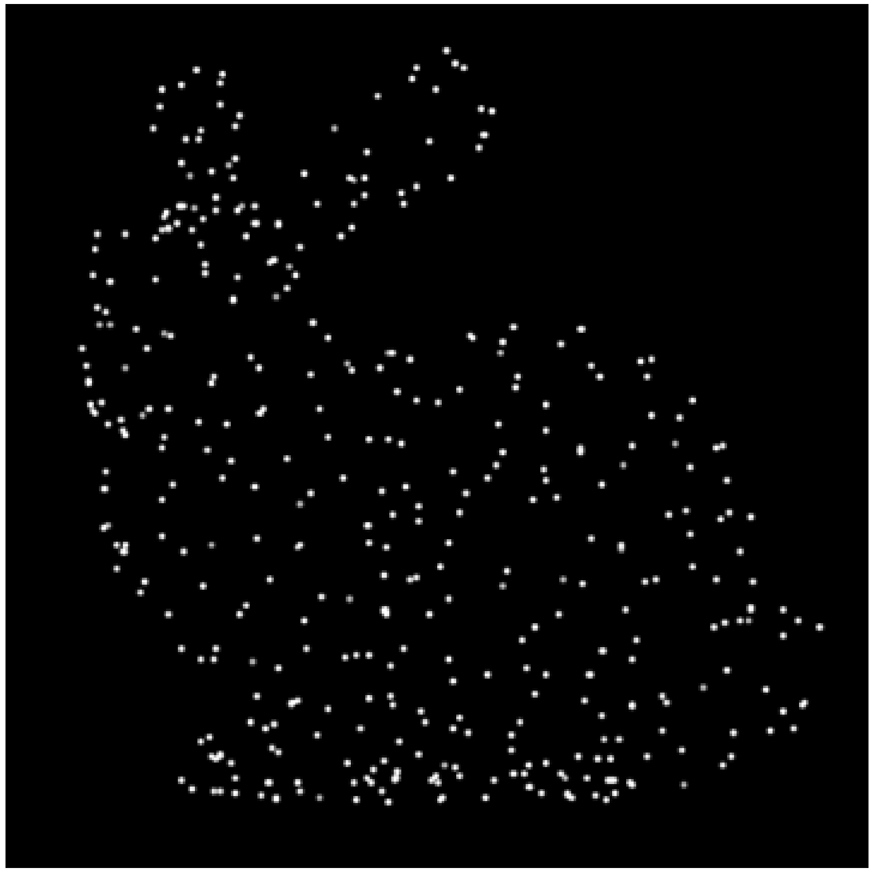
- **Range scans**
  - Each scan is a triangular mesh
  - Normal vectors derived by local connectivity
  - All the scans in the same coordinate system



Problem

Given a set of points $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ with $\mathbf{p}_i \in \mathbb{R}^3$

Find a manifold surface $S \subset \mathbb{R}^3$ which approximates P



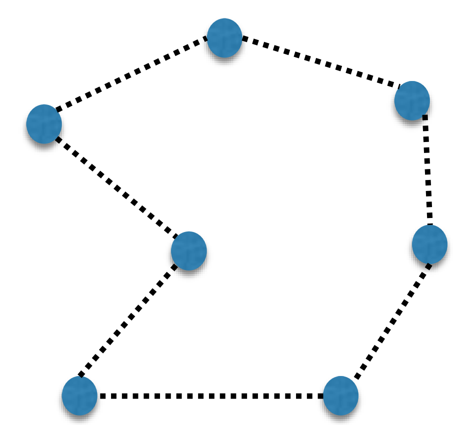
Surface Reconstruction

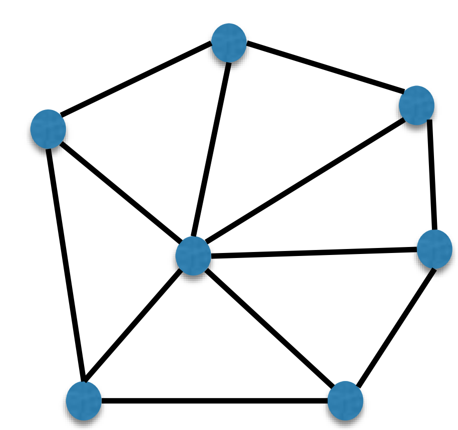
- Explicit approach
  - Delaunay Triangulation
  - Ball Pivoting
  - Zippering

- Implicit approach
  - Radial Basis Function
  - Signed distance field from range scan
  - Moving Least Square
  - Smoothed Signed Distance Surface Reconstruction
  - Poisson Surface Reconstruction

Delaunay Algorithm

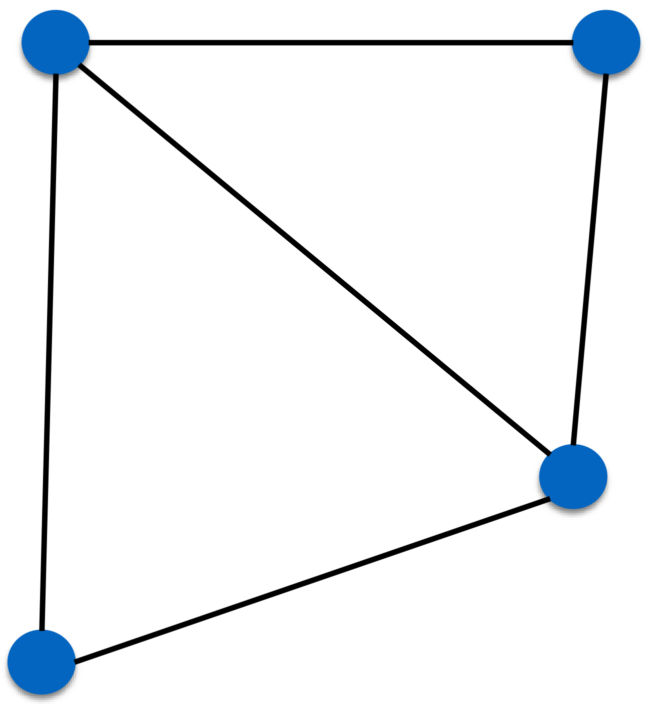
- General triangulation on n points in d-dimensional space by partition of the covex-hull with d-simplex
  - A triangulation such that for each d-simplex the circum-hypersphere doesn’t contains any other points
  - The triangulation covers all the covex-hull defined by the input points

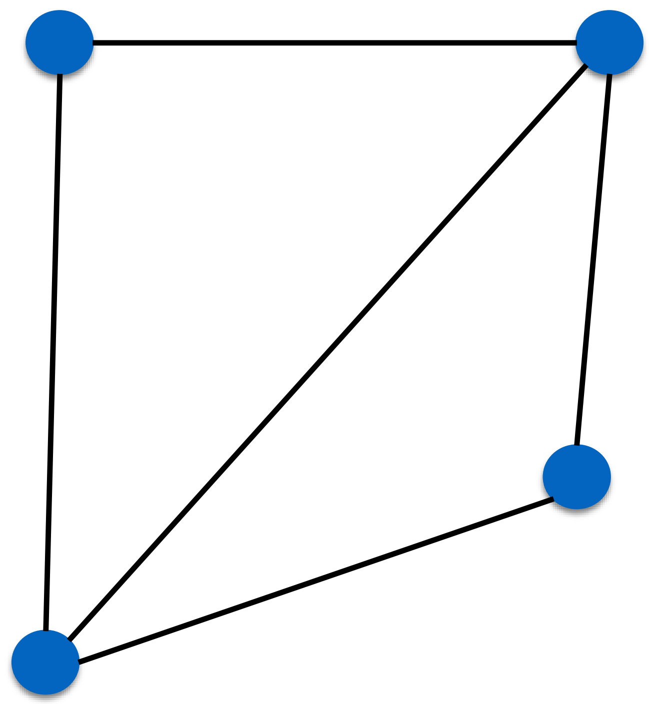




Delaunay Algorithm

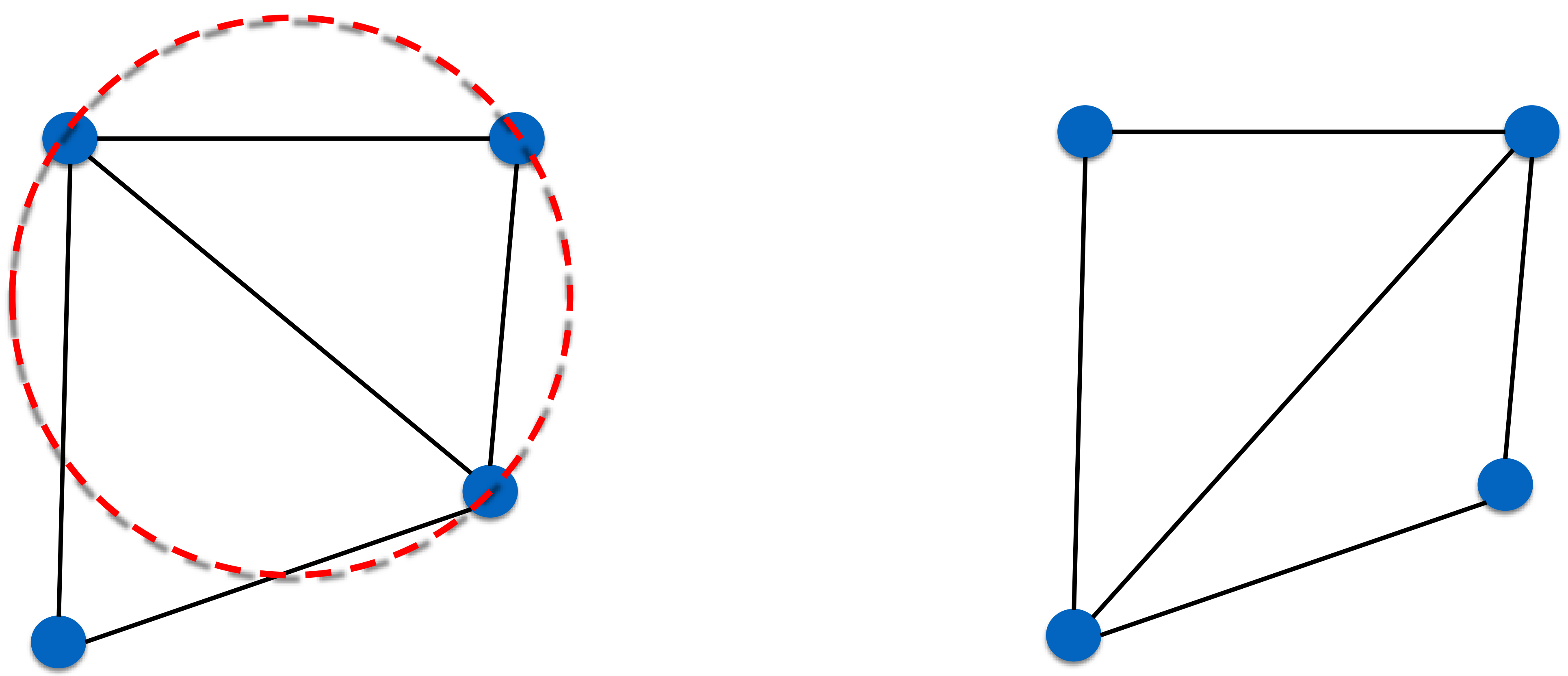
- 2D case





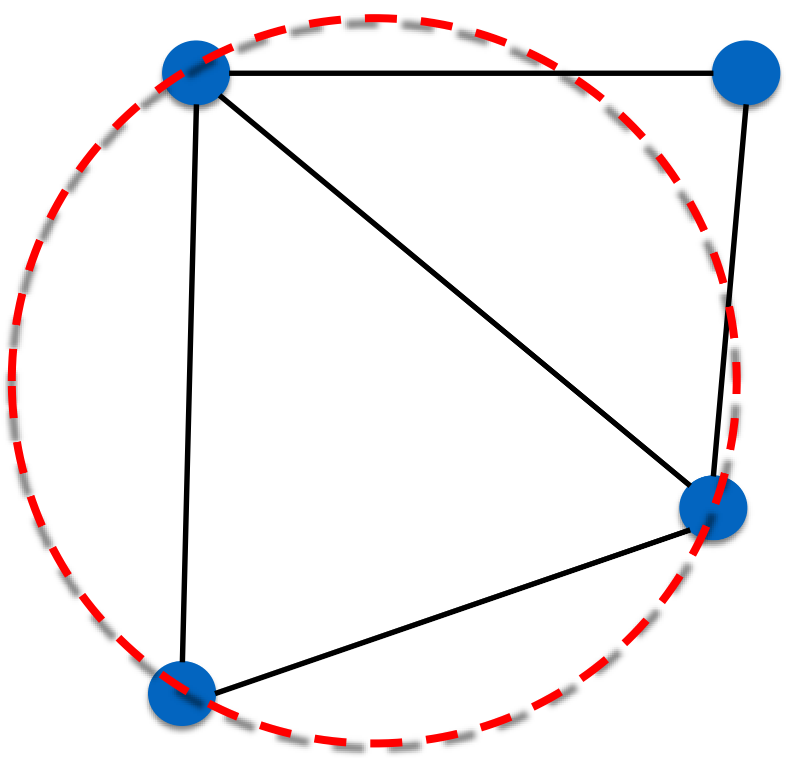
Delaunay Algorithm

- 2D case

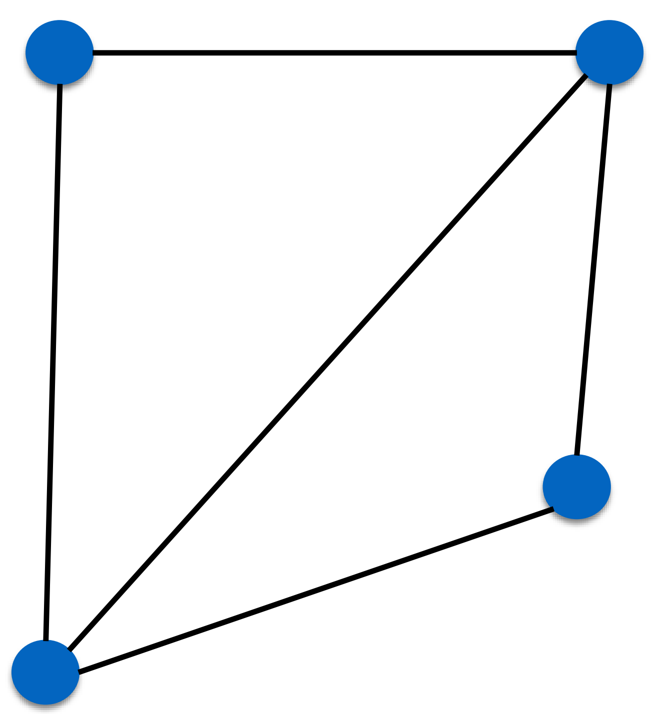


Delaunay Algorithm

- 2D case

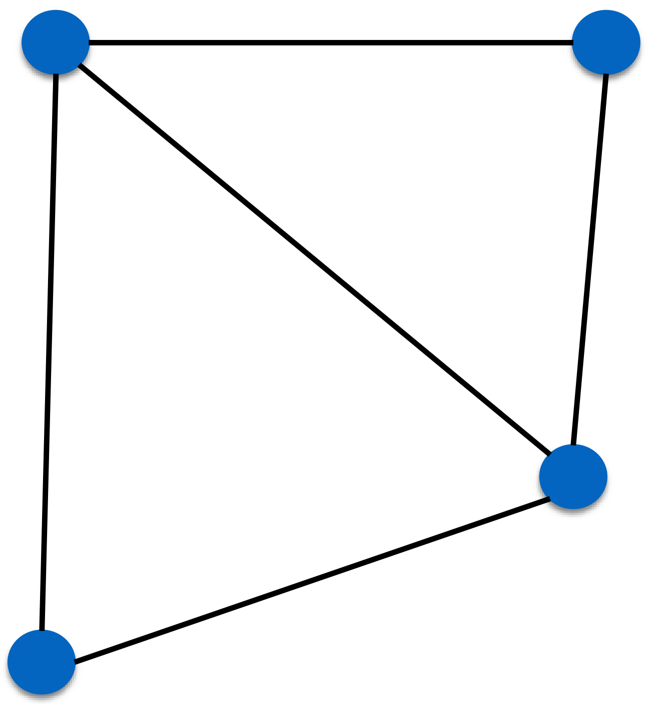


VALID DELAUNAY

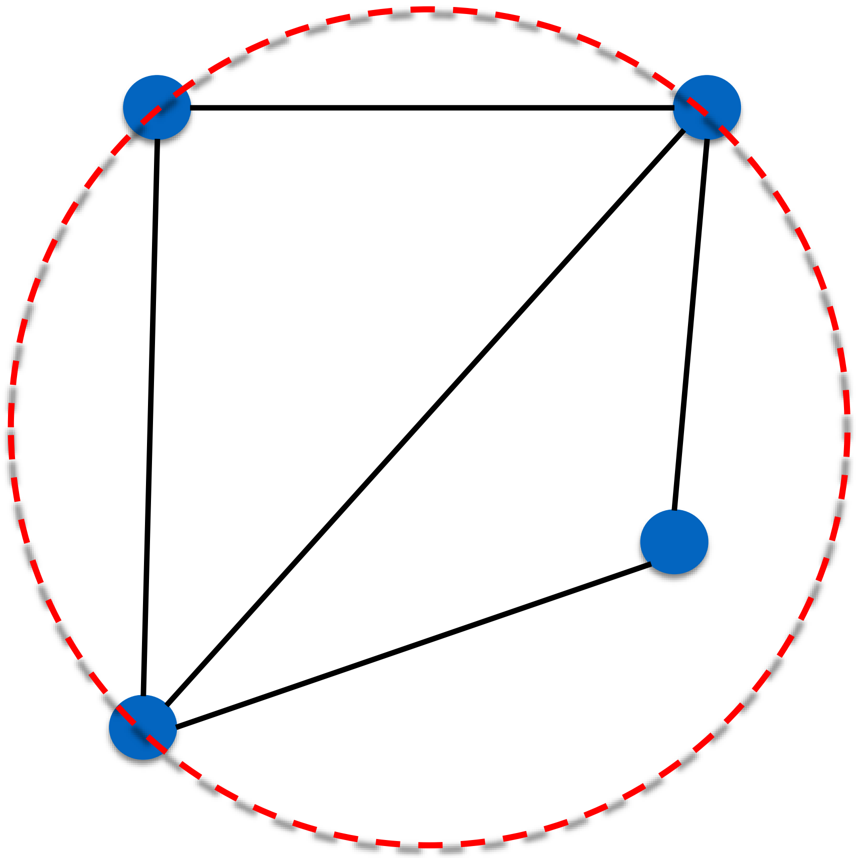


Delaunay Algorithm

- 2D case



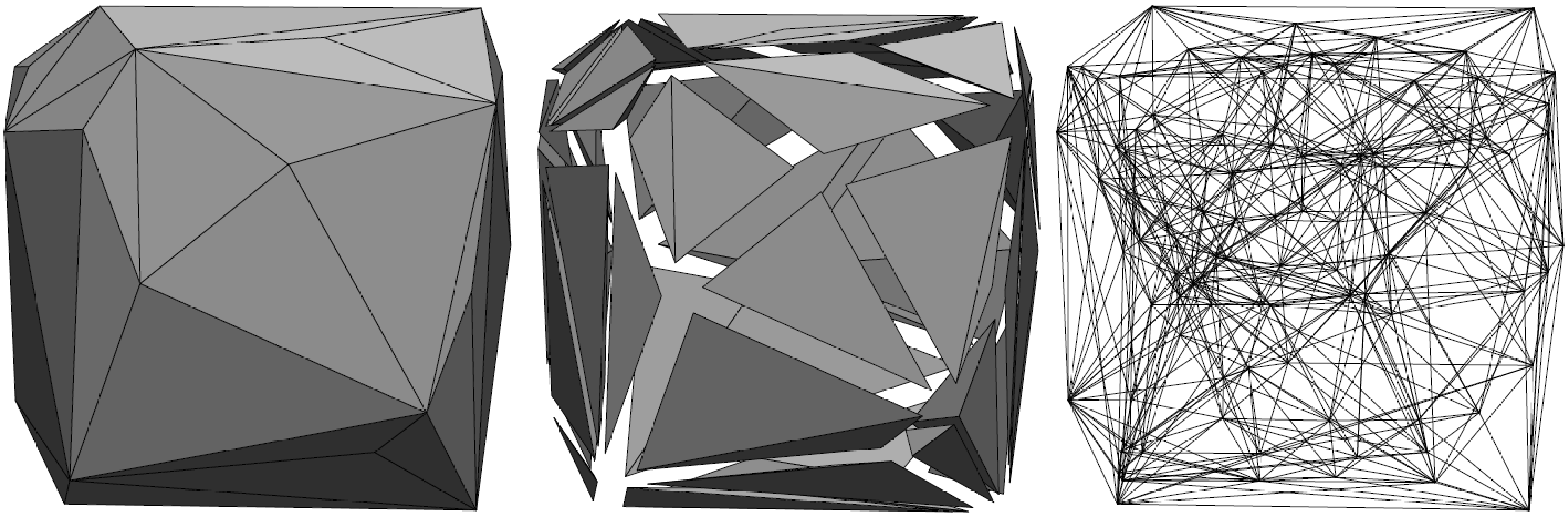
VALID DELAUNAY



NO VALID DELAUNAY

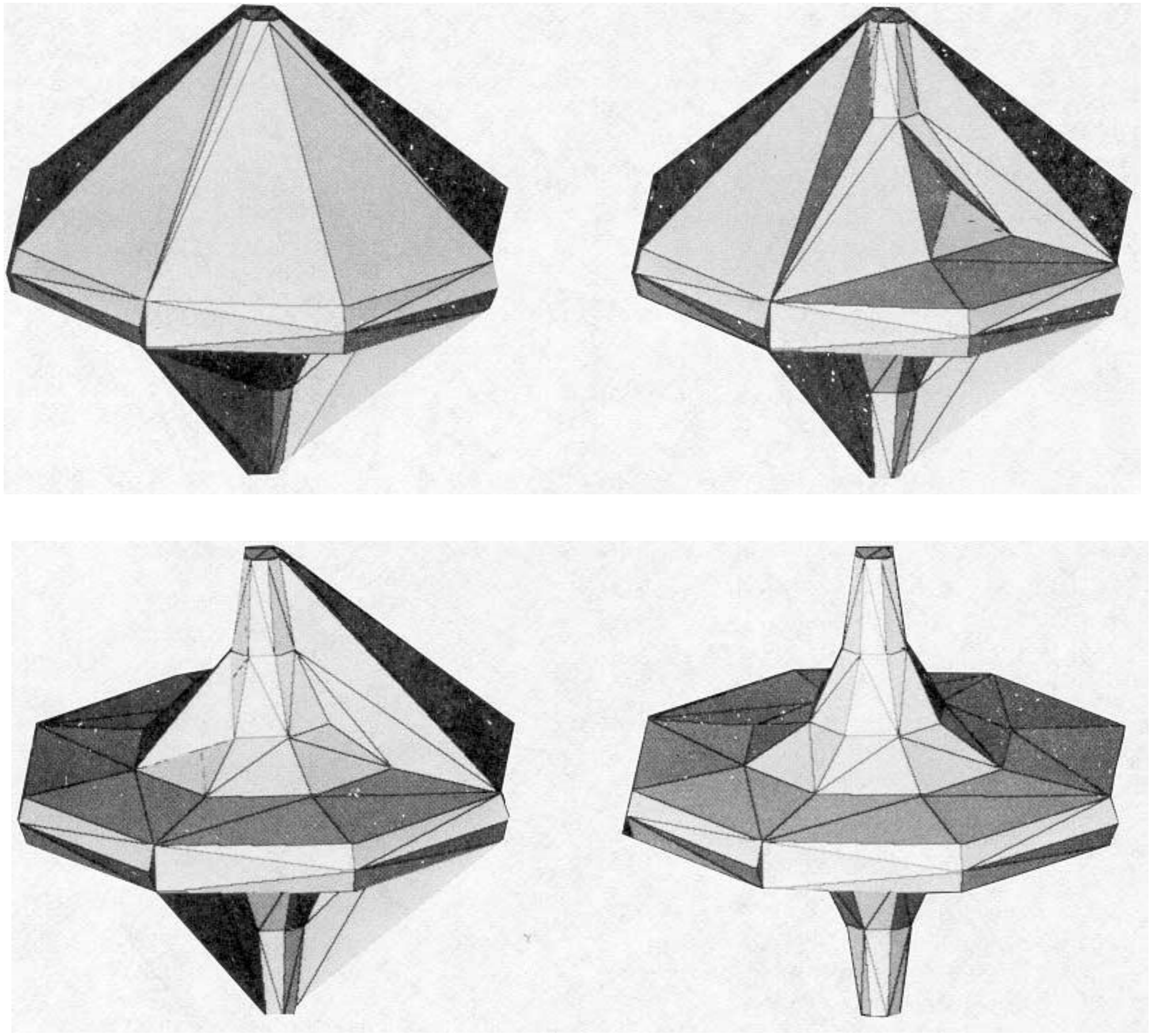
Delaunay Algorithm

- 3D case (triangle -> tetrahedron, circle->sphere)

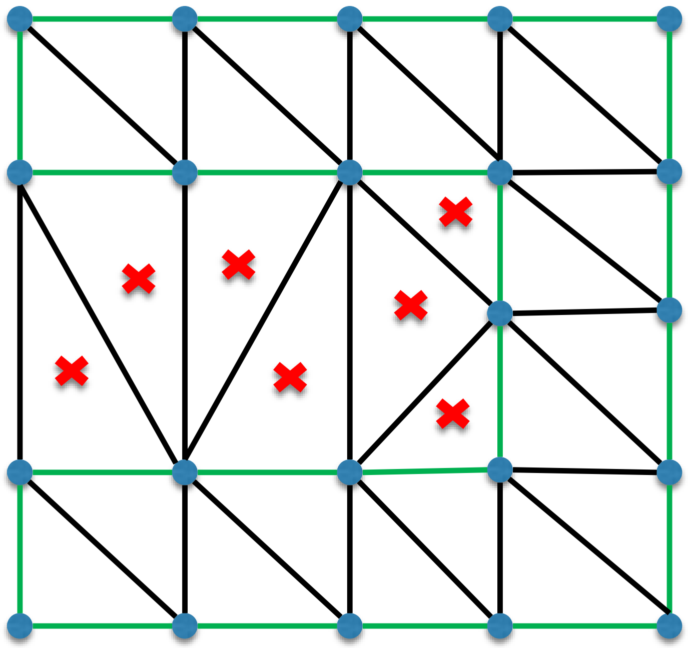


Delaunay Algorithm

- Need a sculpting operation to extract the limit surface



[Boissonnat, TOG 84]



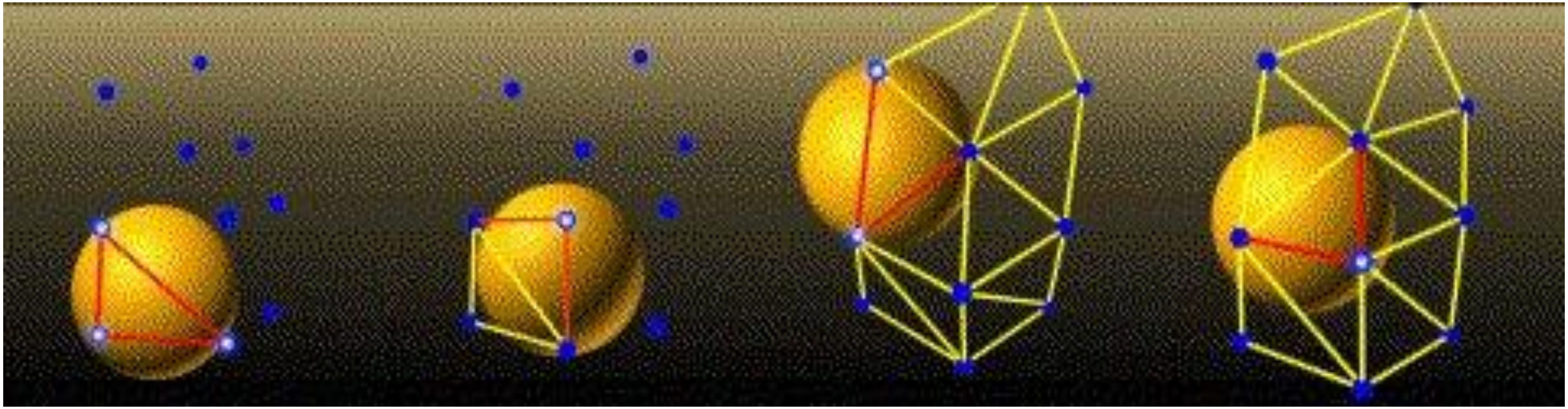
Delaunay Algorithm

- Problems
  - Need clean data
  - Slow
  - Not always exist for d >= 3

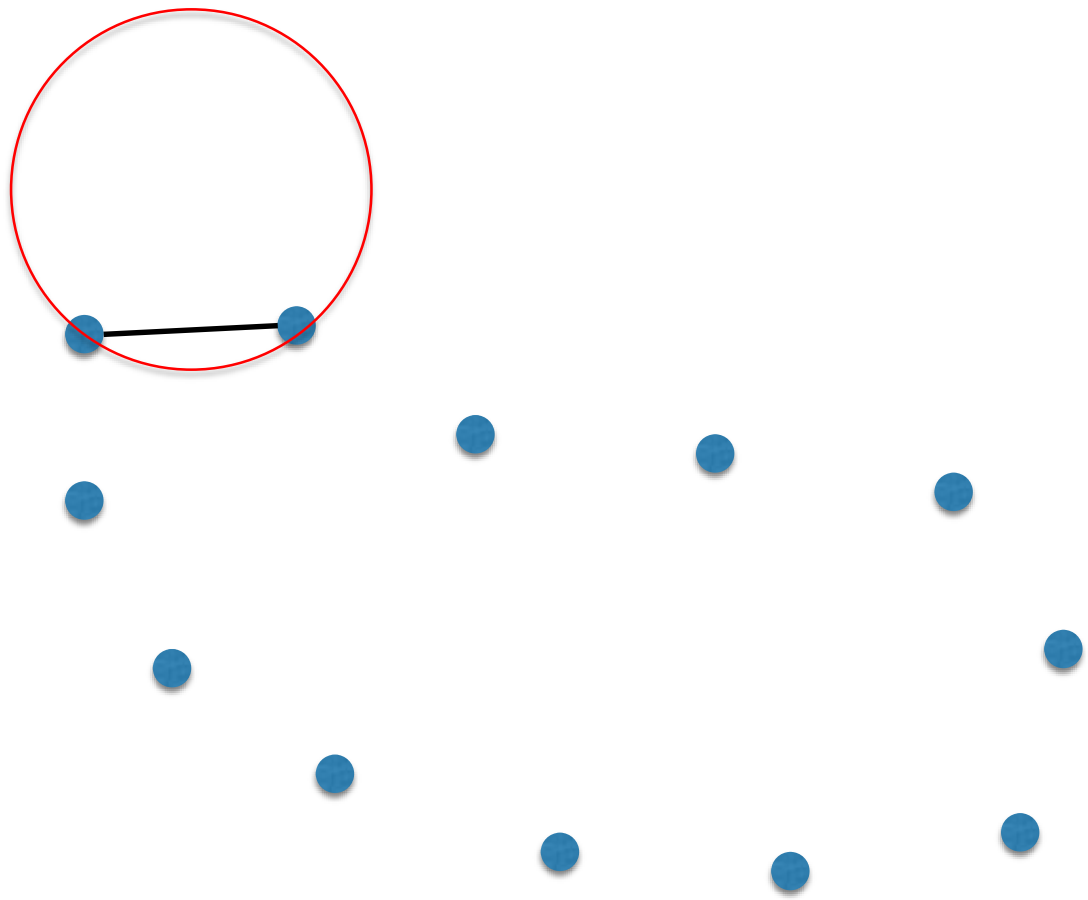
Ball Pivoting

[Bernardini et al., TVCG 99]

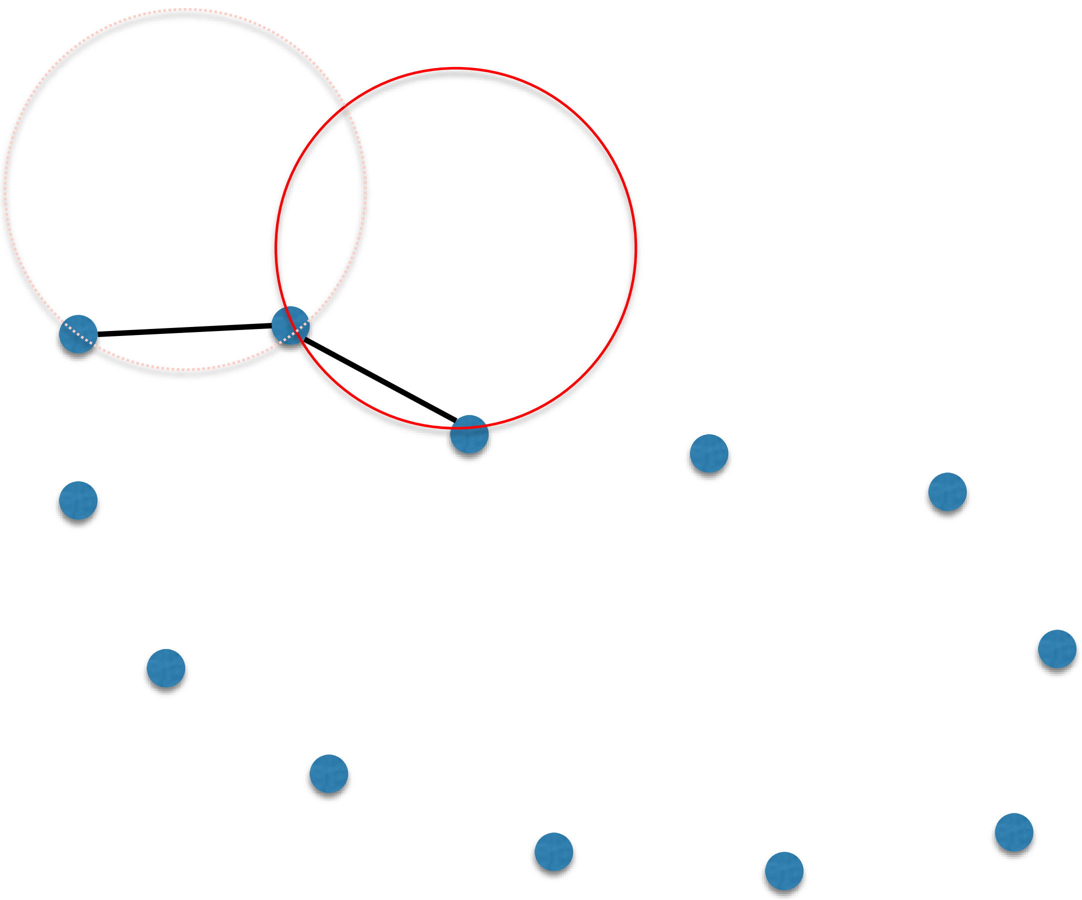
- Pick a ball radius, roll ball around surface, connect what it hits
- Pivoting of a ball of fixed radius around an edge of the current front adds a new triangles to the mesh



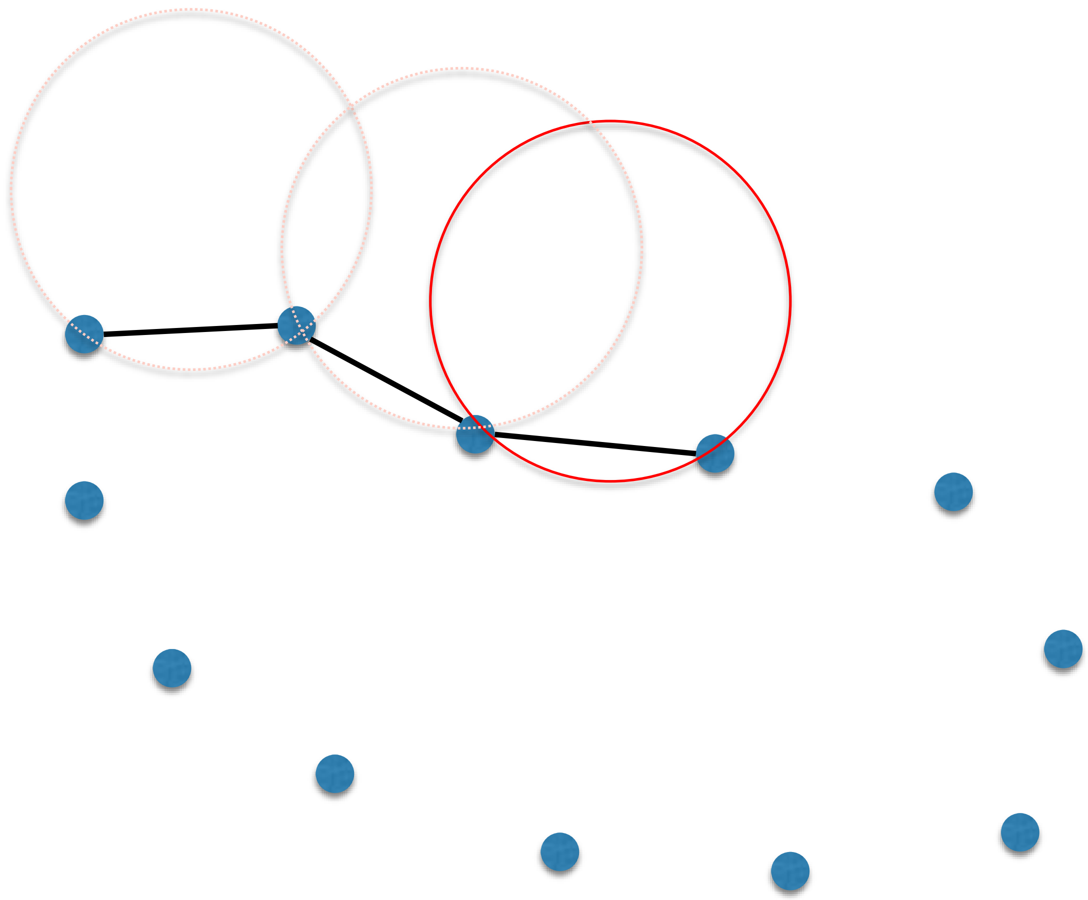
Ball Pivoting



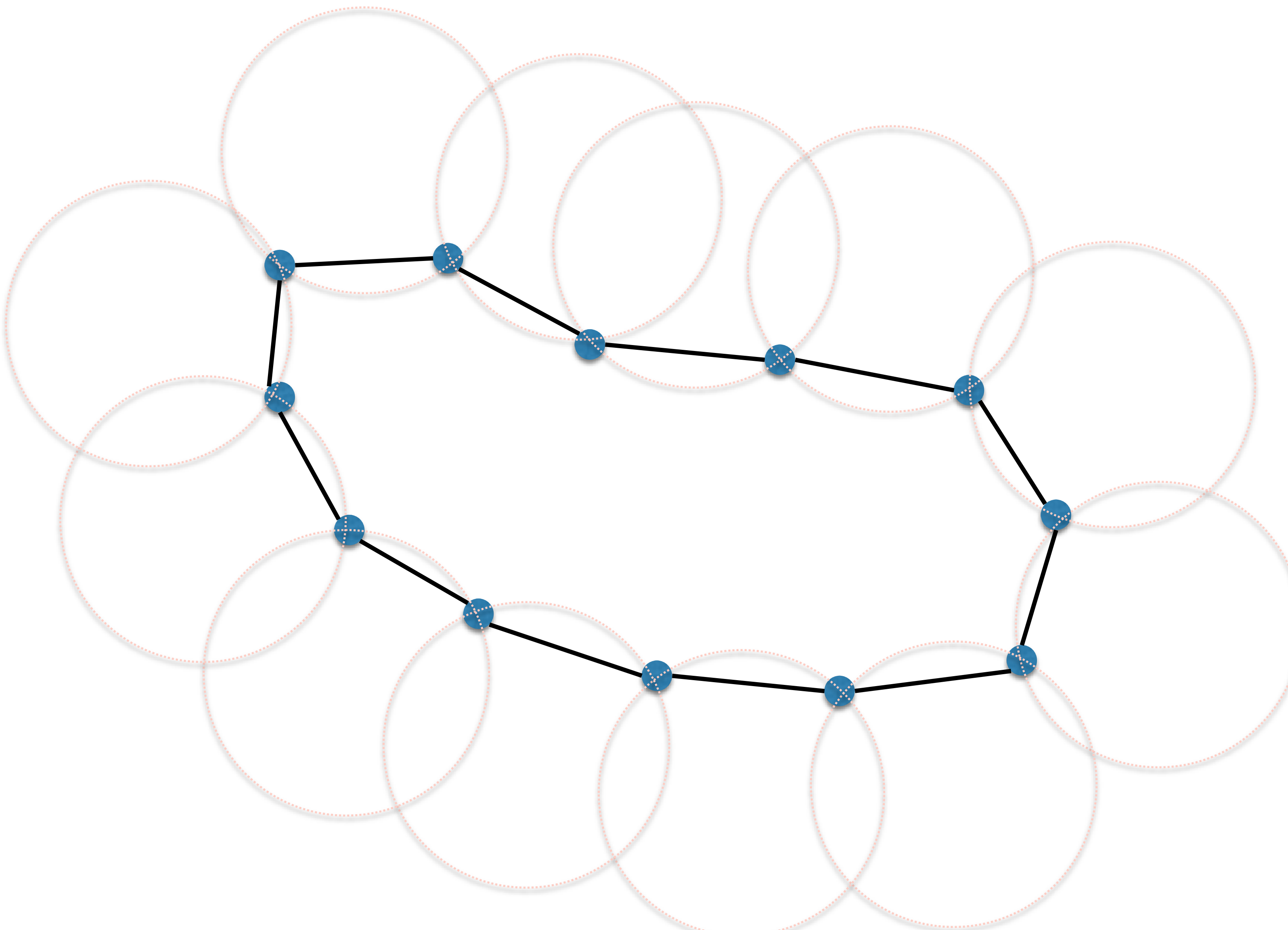
Ball Pivoting



Ball Pivoting



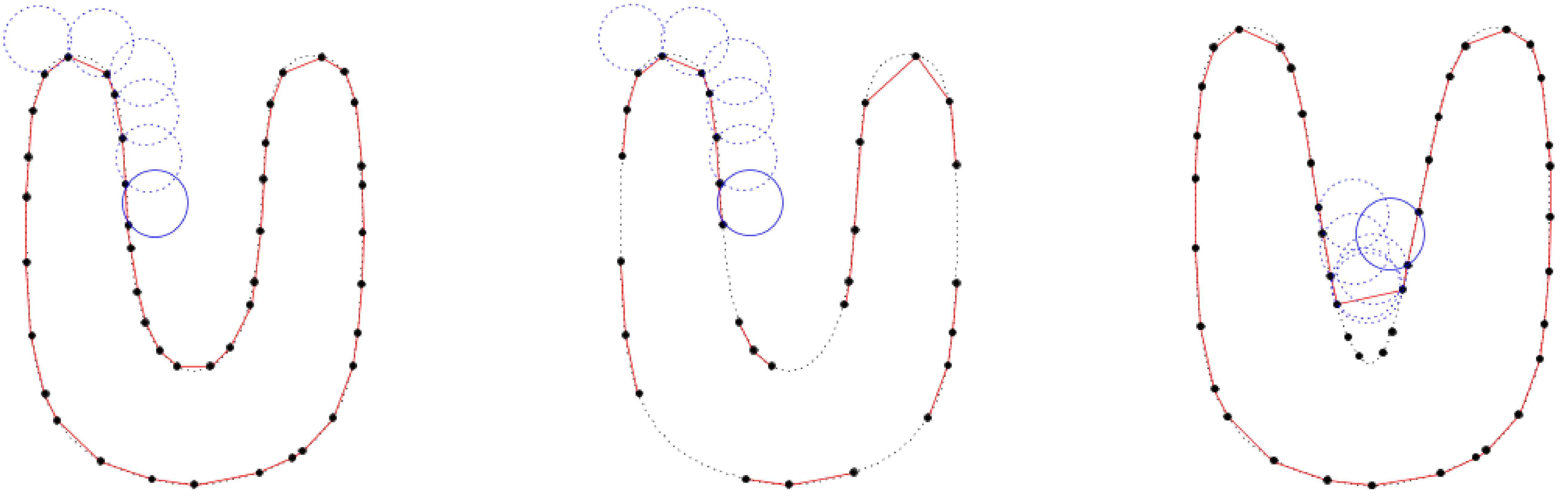
Ball Pivoting



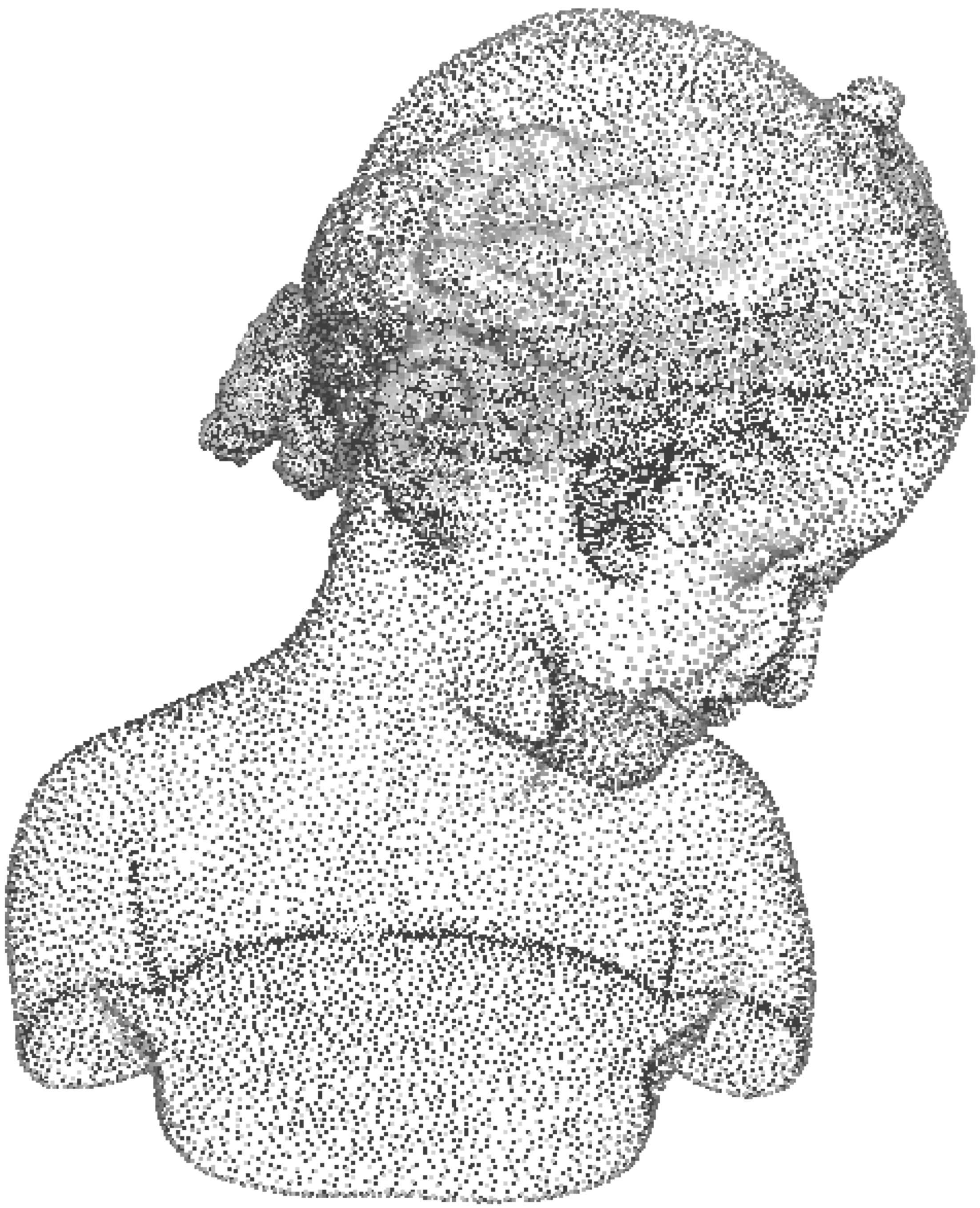
Ball Pivoting

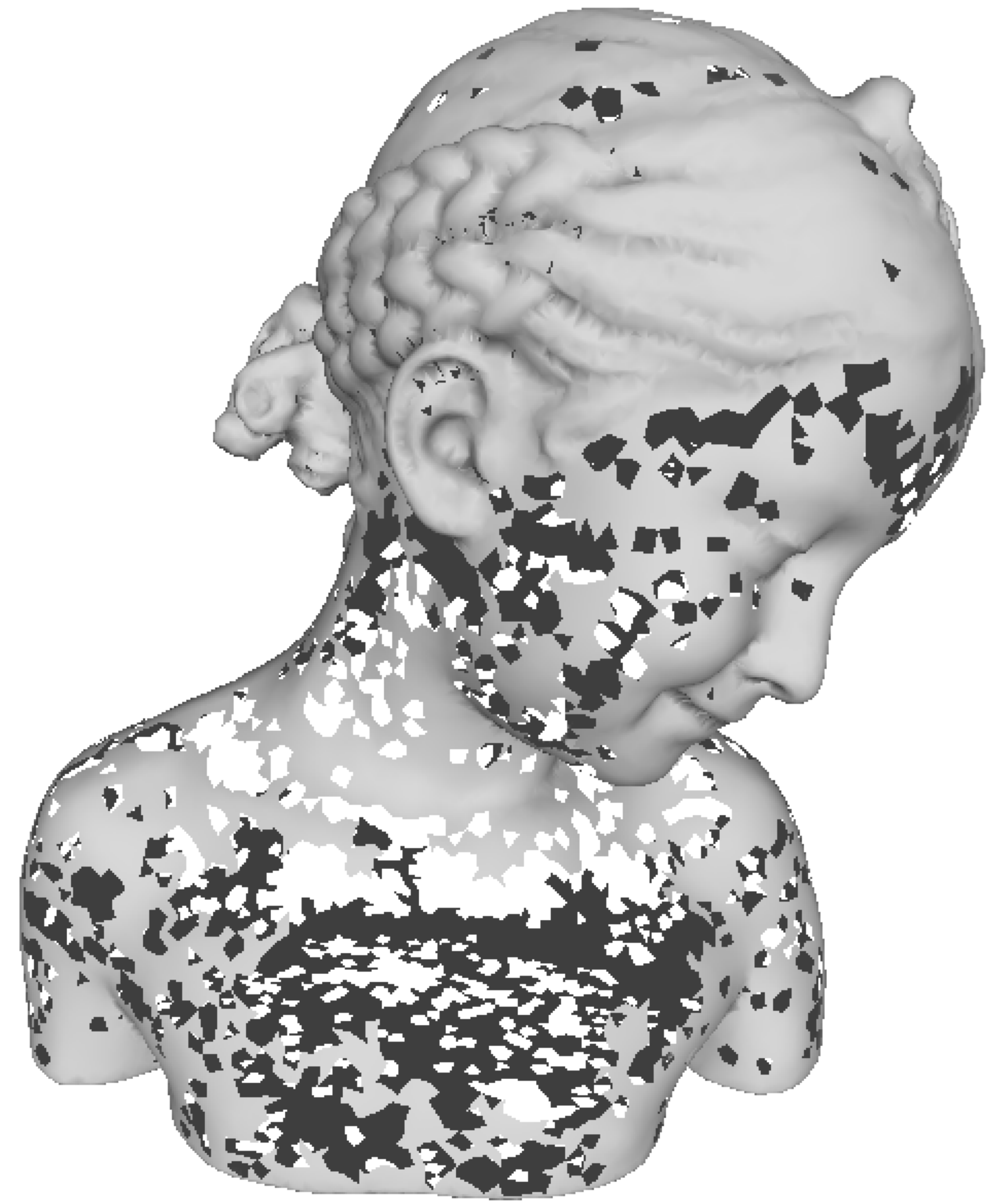
- Problem with different sampling density, but we can use ball of increasing radius

- Problem with concavities



Ball Pivoting





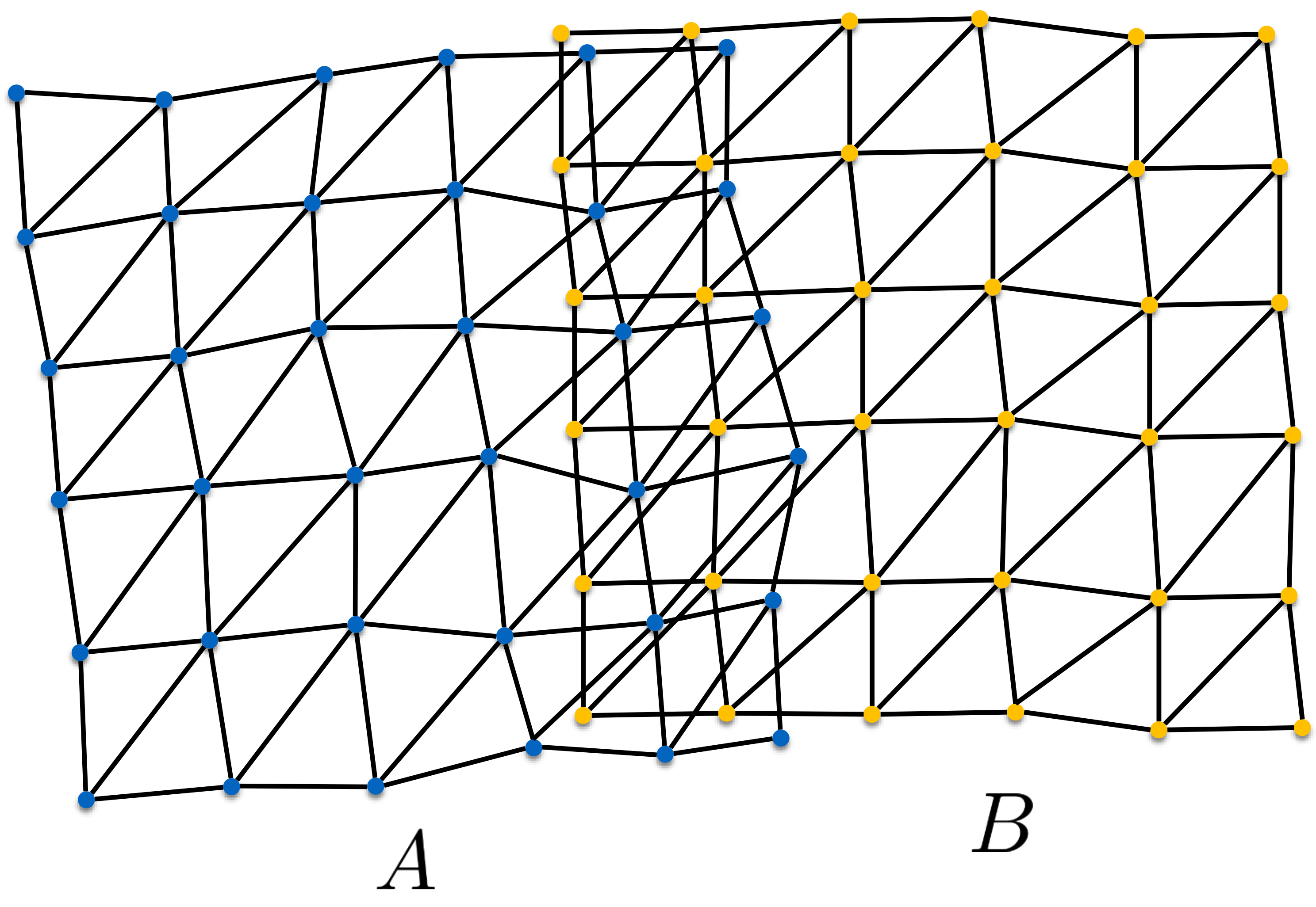
Ball Pivoting

- Iterative approach
  - Small Radius, capture high frequencies
  - Large Radius, close holes (keeping mesh from previous pass)

Zippering

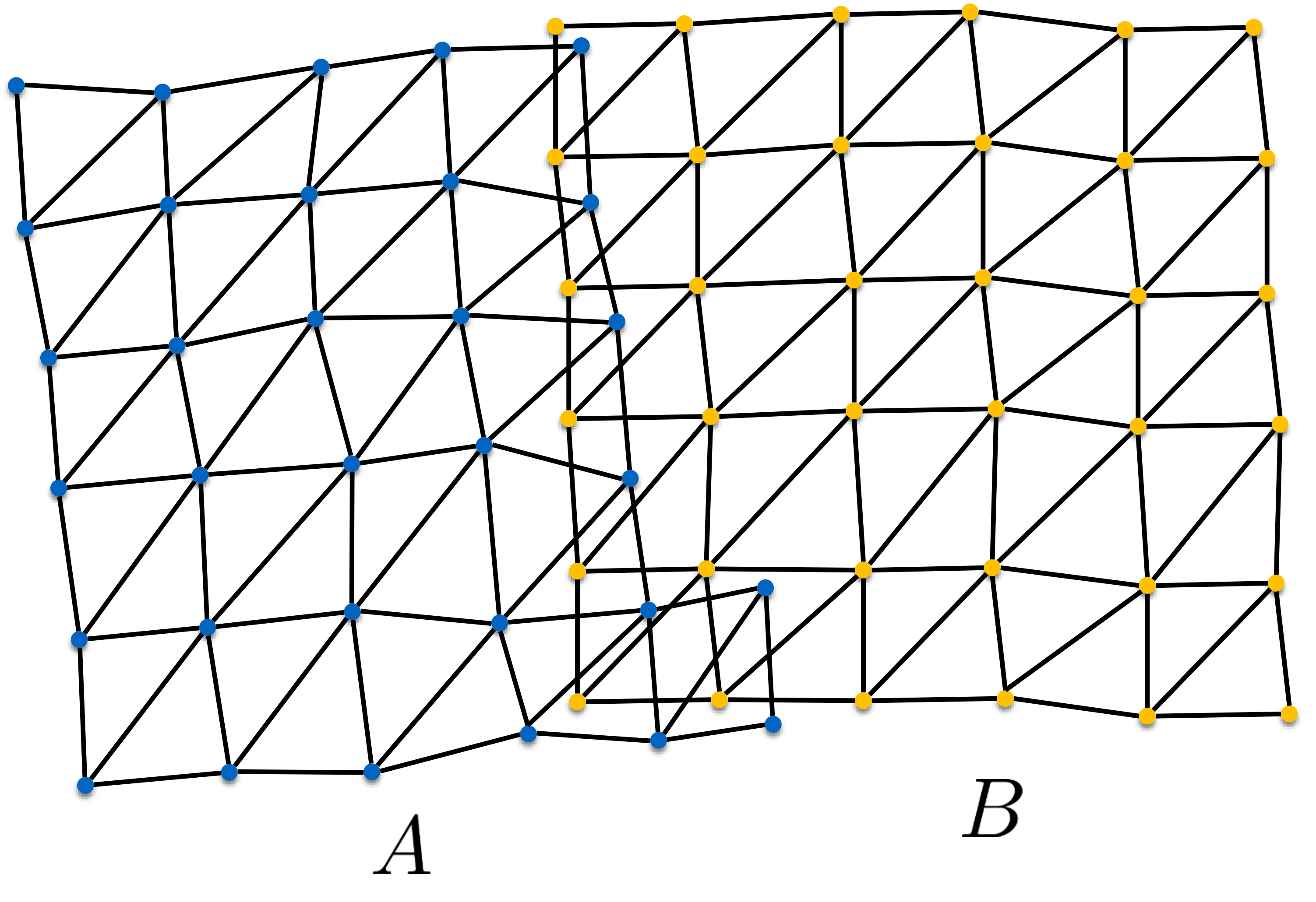
[Turk et al., SIGGRAPH 94]

- “Zipper” several scans to one single model



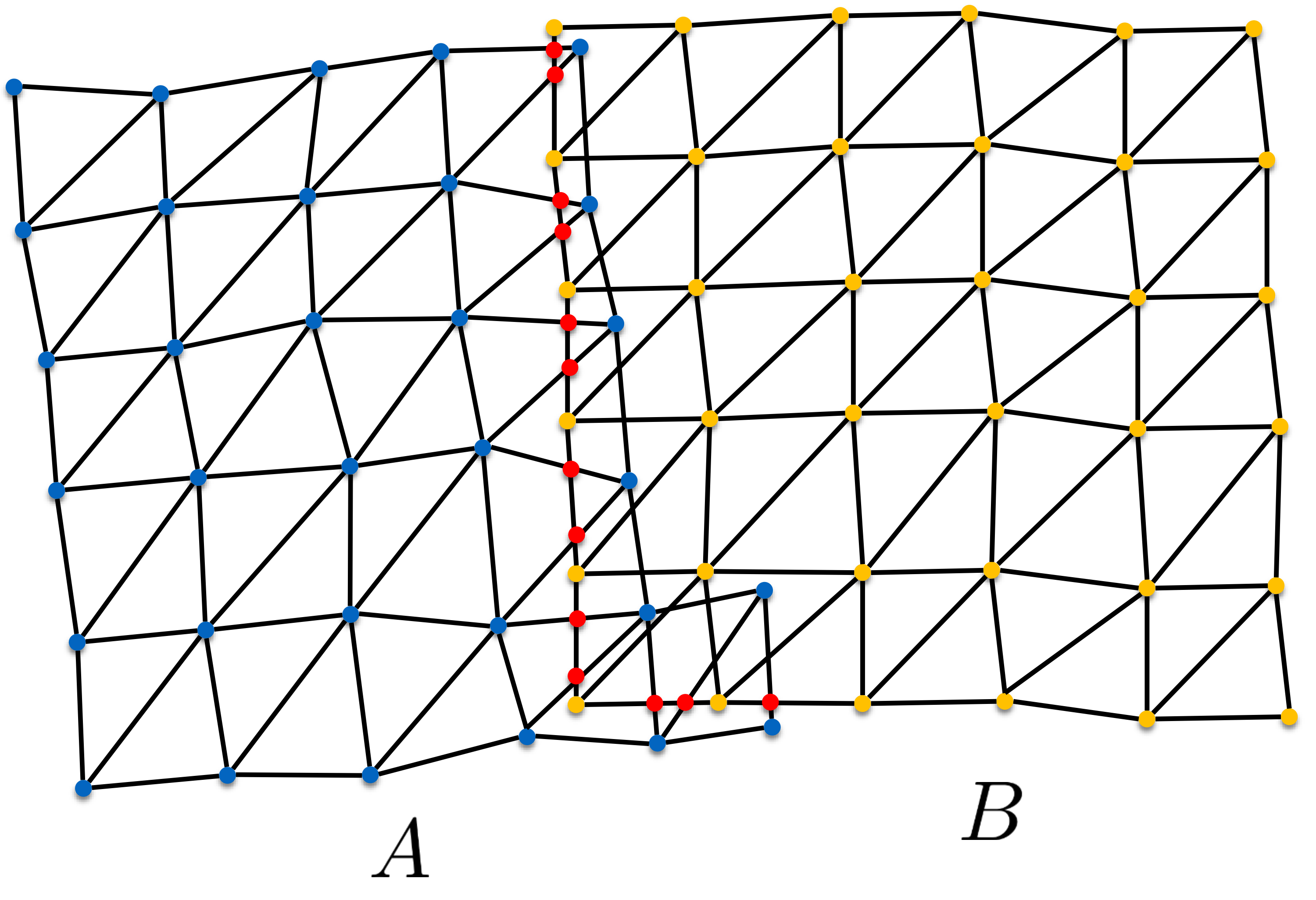
Zippering

- Remove overlap regions (all the vertices of the triangle have as neighbor a no-border vertex)



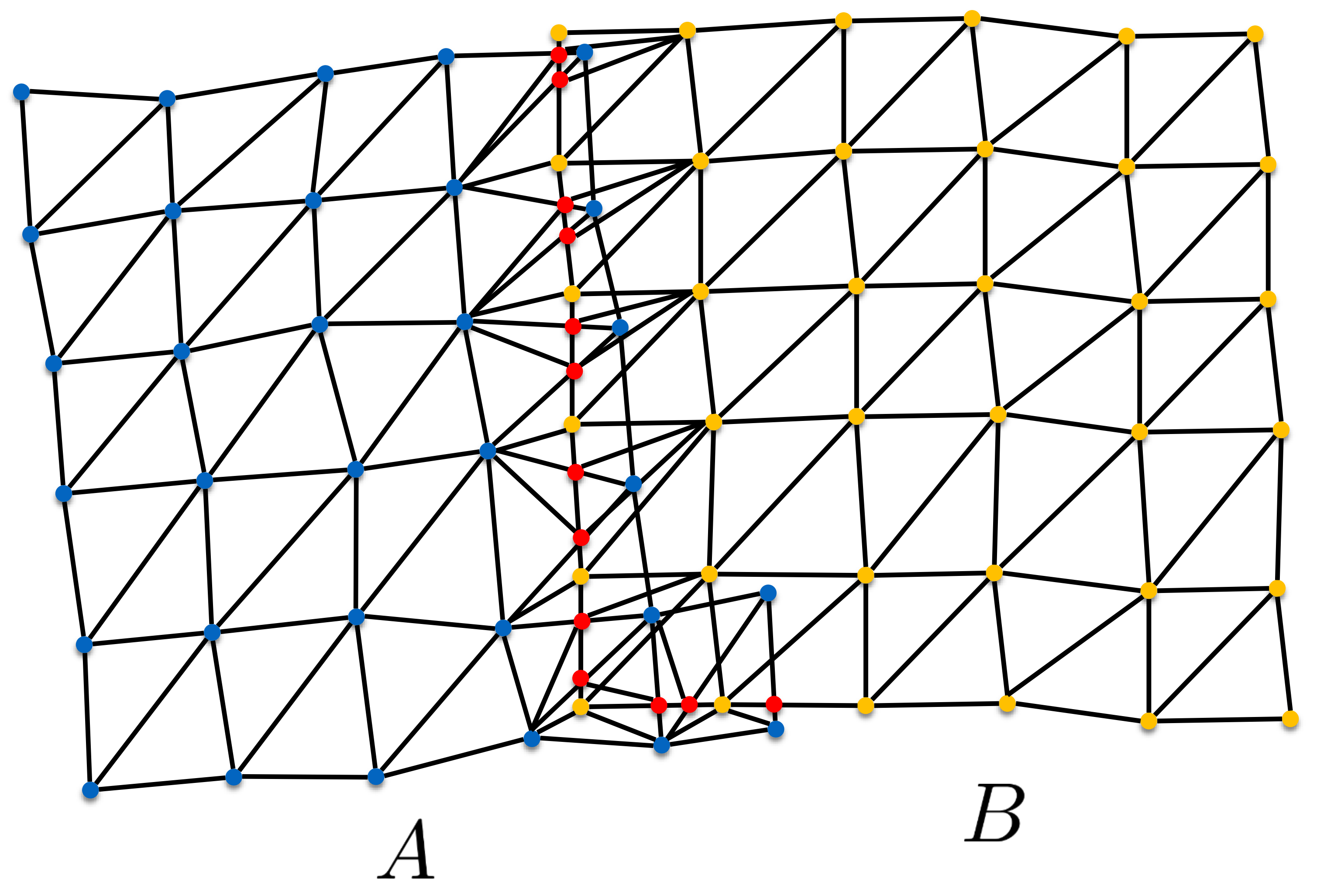
Zippering

- Project and intersect the boundary of B



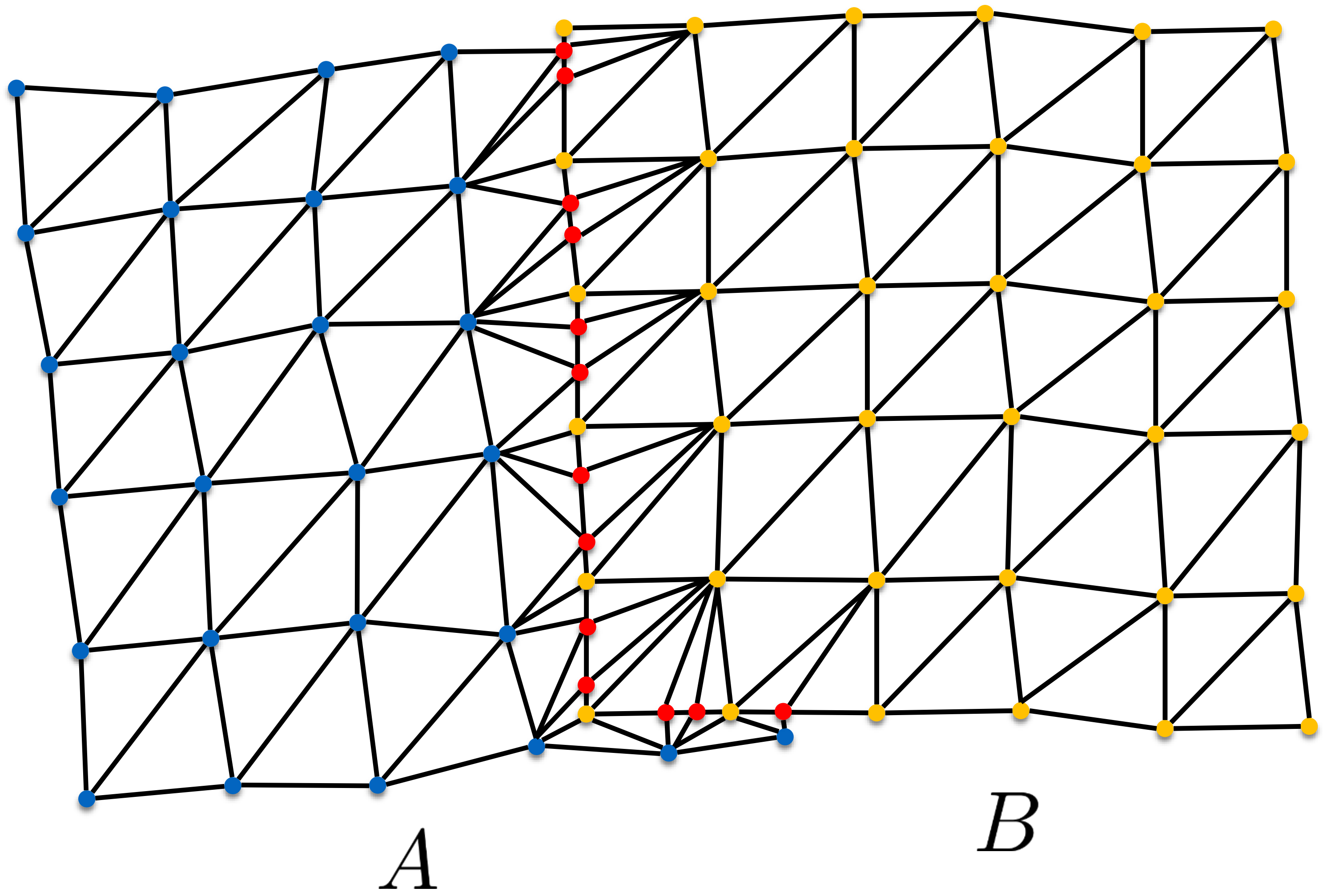
Zippering

- Incorporate the new points in the triangulation



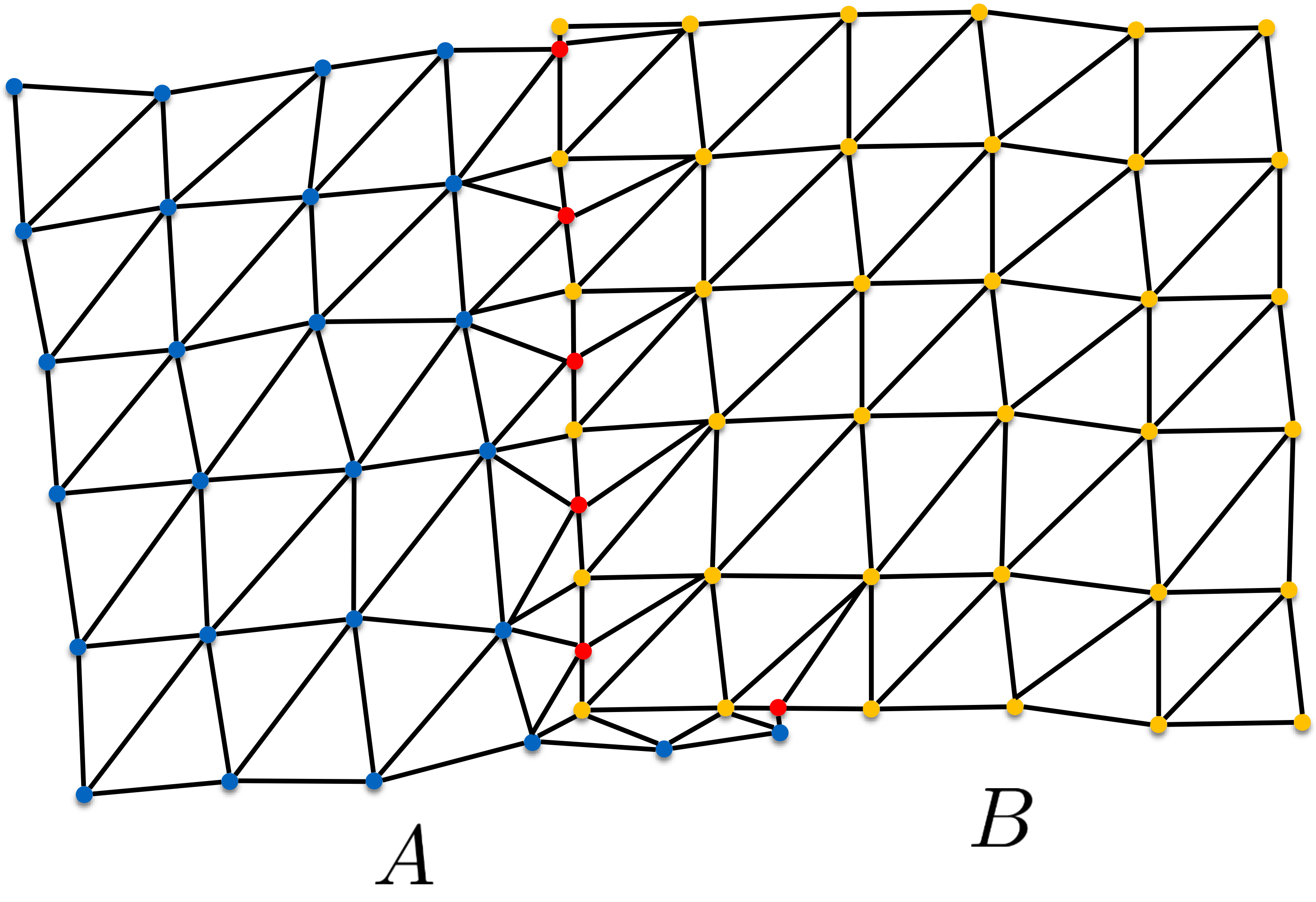
Zippering

- Remove overlap regions of A



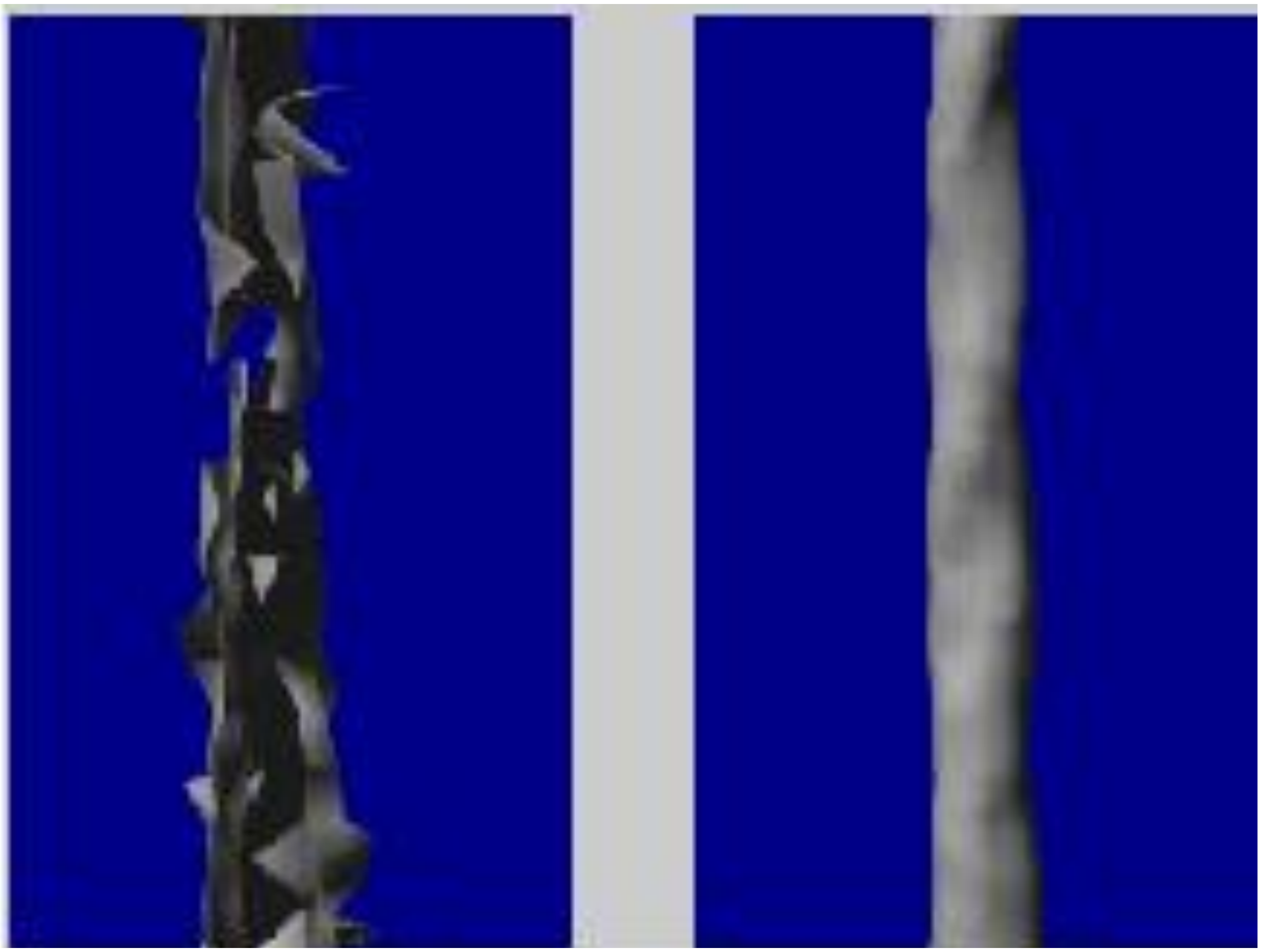
Zippering

- Optimize triangulation



Zippering

- Preserve regular structure of each scan but problems with intricate geometry, noise and small misalignment



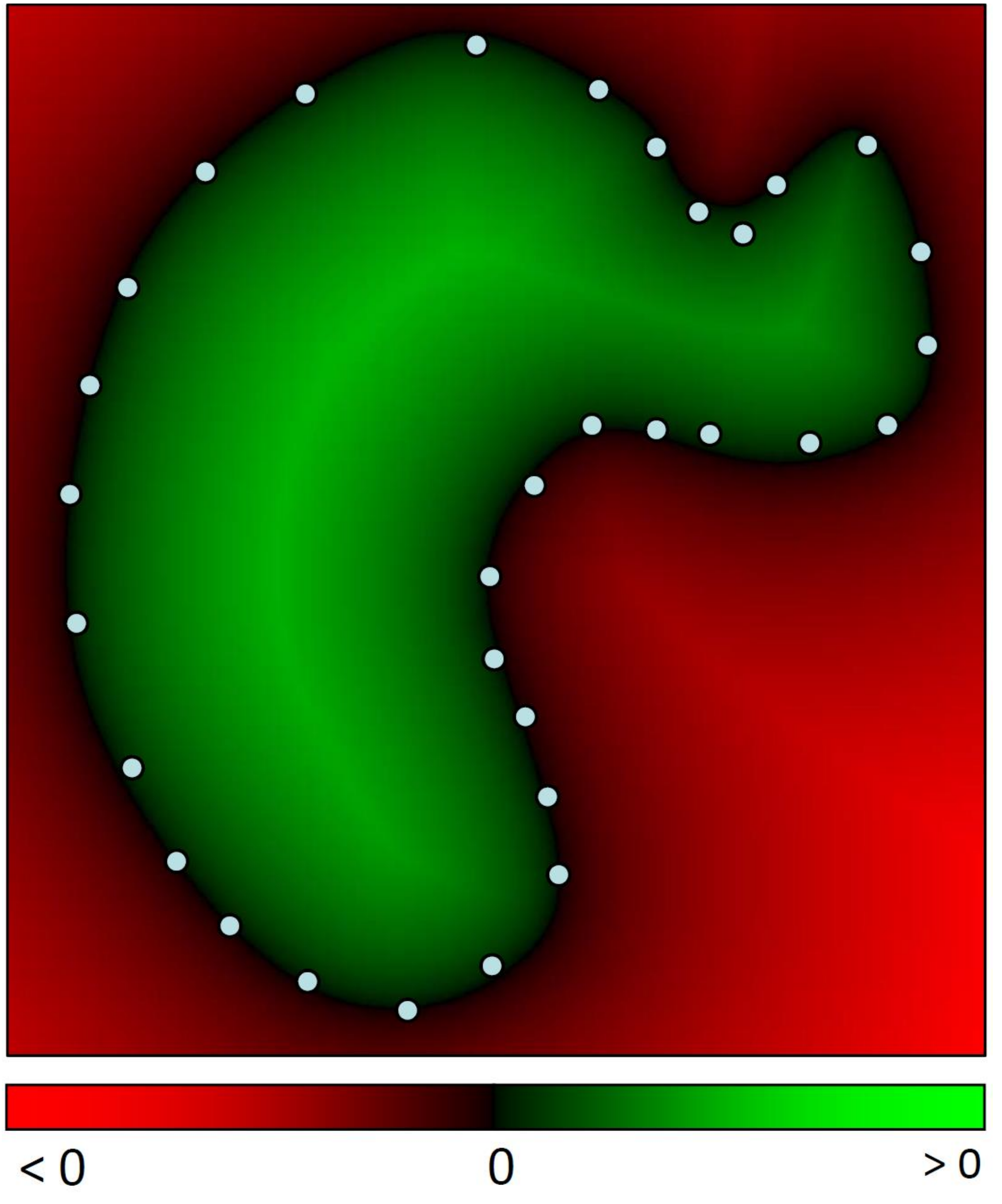
Implicit Reconstruction

- Define a distance function f with value < 0 outside the shape and > 0 inside the shape

$$f : \mathbb{R}^3 \leftarrow \mathbb{R}$$

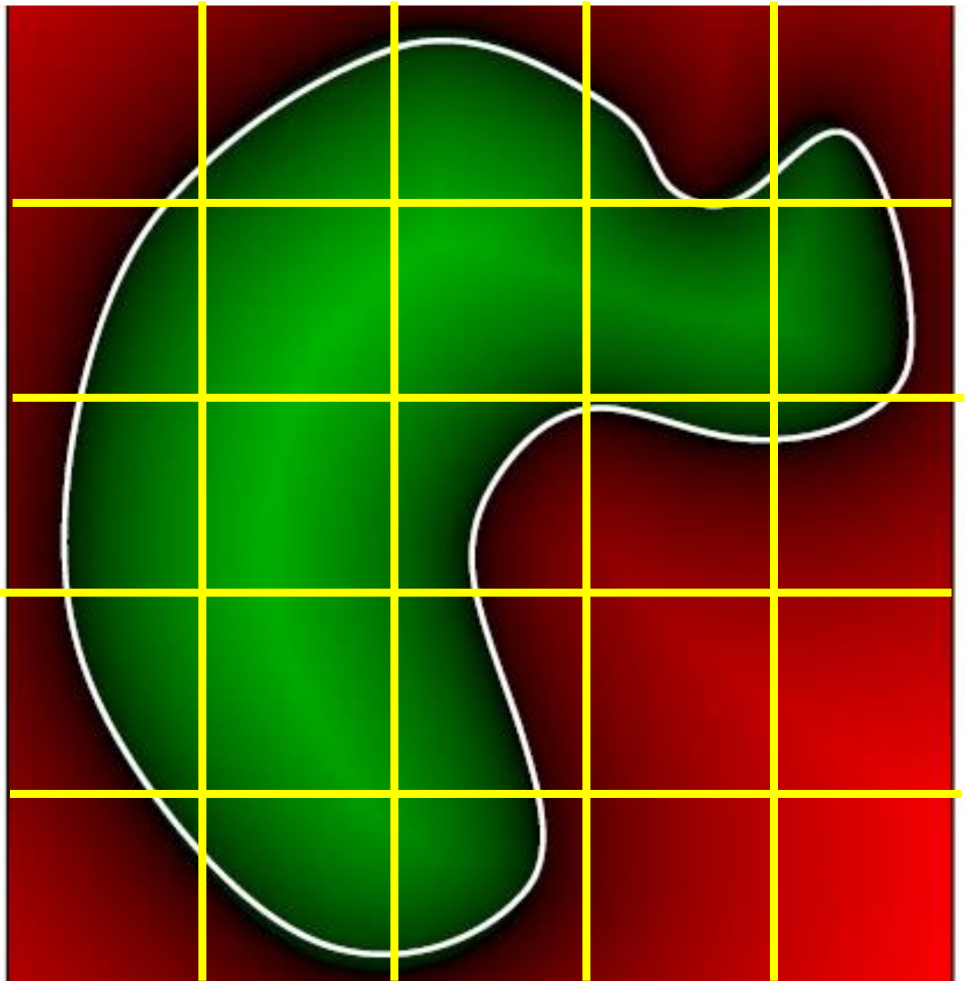
- Extract the zero-set

$$S = \{\mathbf{x} \in \mathbb{R}^3 : f(\mathbf{x}) = 0\}$$



Implicit Reconstruction Algorithm

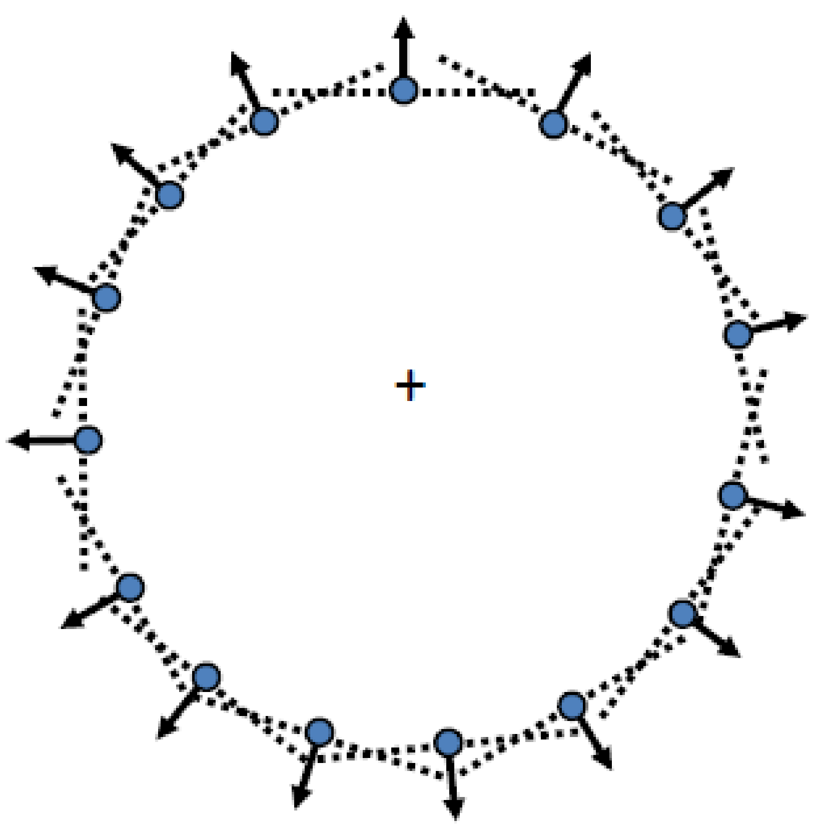
- Input: Point cloud or range map
  1. Estimation of the signed distance field
  2. Evaluation of the function on an uniform grid
  3. Mesh extraction via Marching Cubes
- Output: Triangular Mesh



- The existing algorithms differ on the method used to compute the signed distance filed

Signed Distance Function

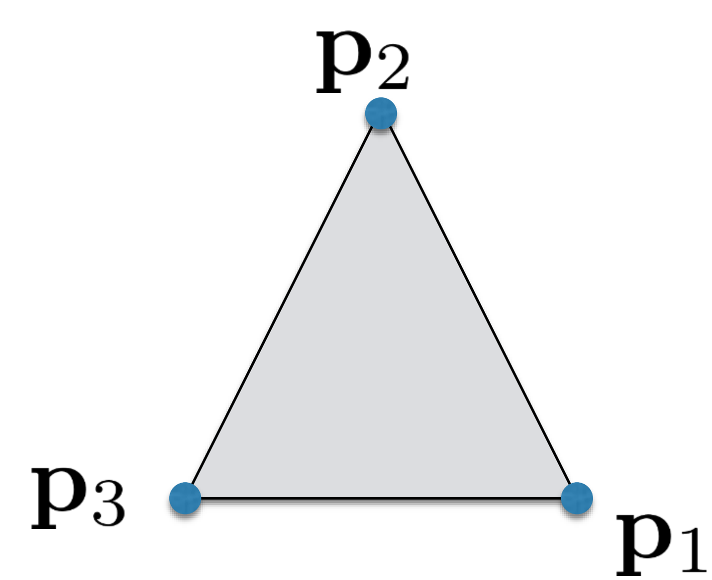
- Construct SDF from point samples
  - Distance to points is not enough
  - Need inside/outside information
  - Requires normal vectors



Normal Estimation for Range Map

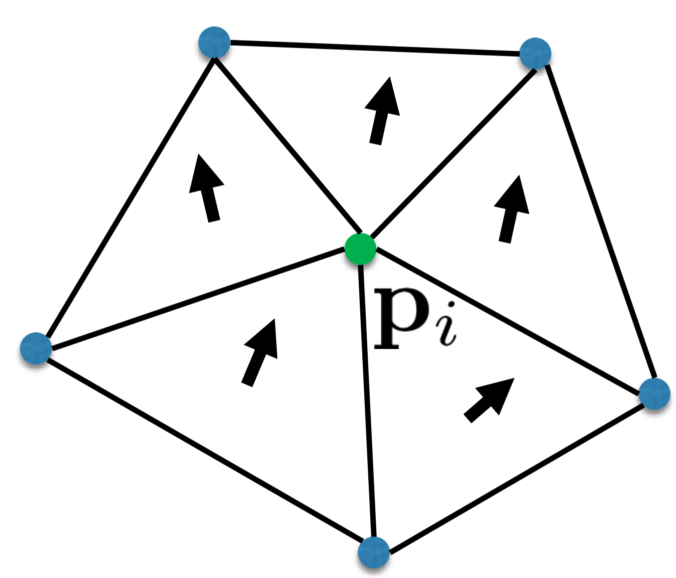
- Per-face normal

$$\vec{\mathbf{n}} = \frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)}{\|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\|}$$



- Per-vertex normal

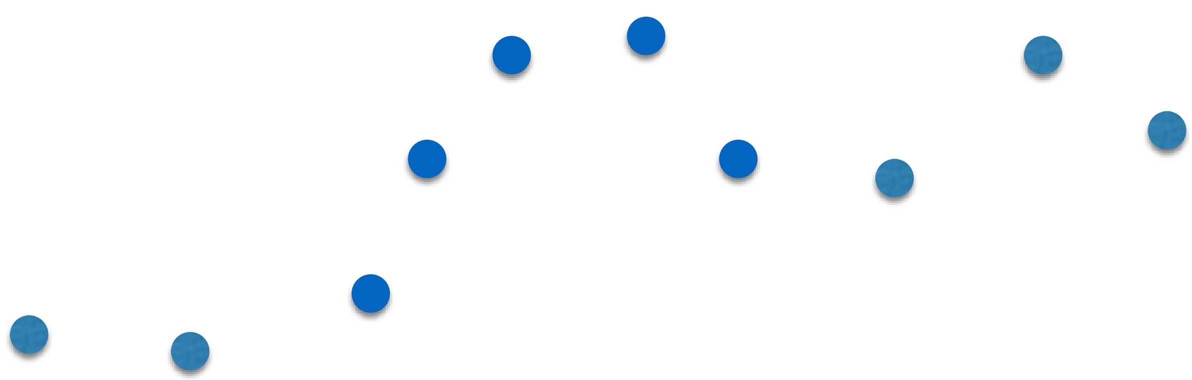
$$\vec{\mathbf{n}}(\mathbf{p}_i) = \sum_{j \in T_i} \vec{\mathbf{n}}_j / \| \sum_{j \in T_i} \vec{\mathbf{n}}_j \|$$



Normal Estimation for Point Cloud

[Hoppe et al., SIGGRAPH 92]

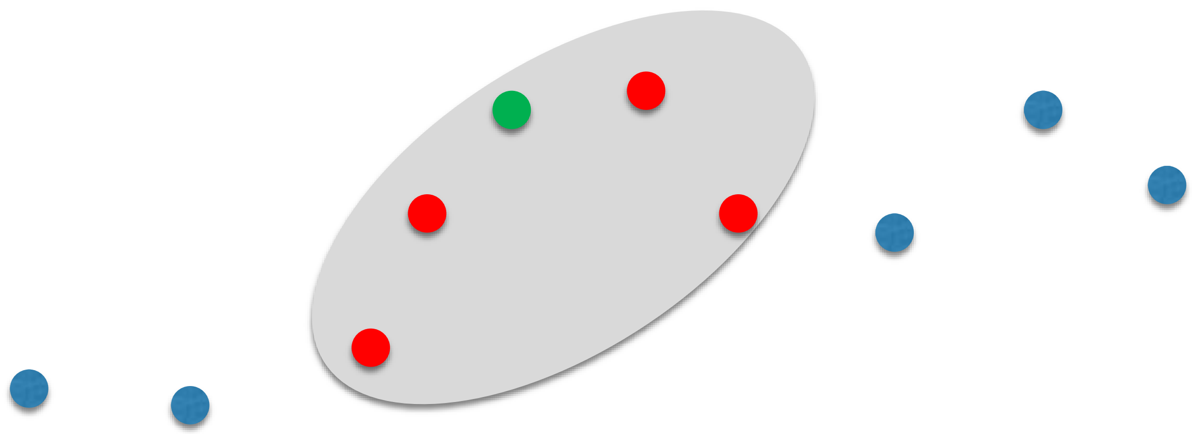
- Estimate the normal vector for each point
  1. Extract the k-nearest neighbor point
  2. Compute the best approximating tangent plane by covariance analysis
  3. Compute the normal orientation



Normal Estimation for Point Cloud

[Hoppe et al., SIGGRAPH 92]

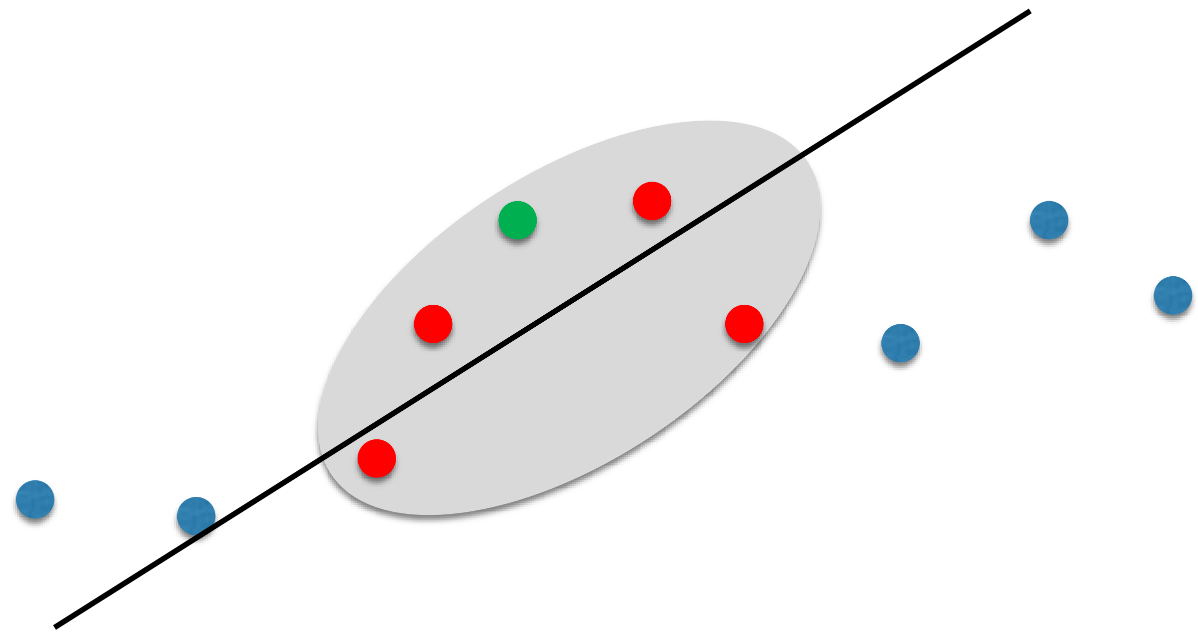
- Estimate the normal vector for each point
  1. **Extract the k-nearest neighbor point**
  2. Compute the best approximating tangent plane by covariance analysis
  3. Compute the normal orientation



Normal Estimation for Point Cloud

[Hoppe et al., SIGGRAPH 92]

- Estimate the normal vector for each point
  1. Extract the k-nearest neighbor point
  2. **Compute the best approximating tangent plane by covariance analysis**
  3. Compute the normal orientation



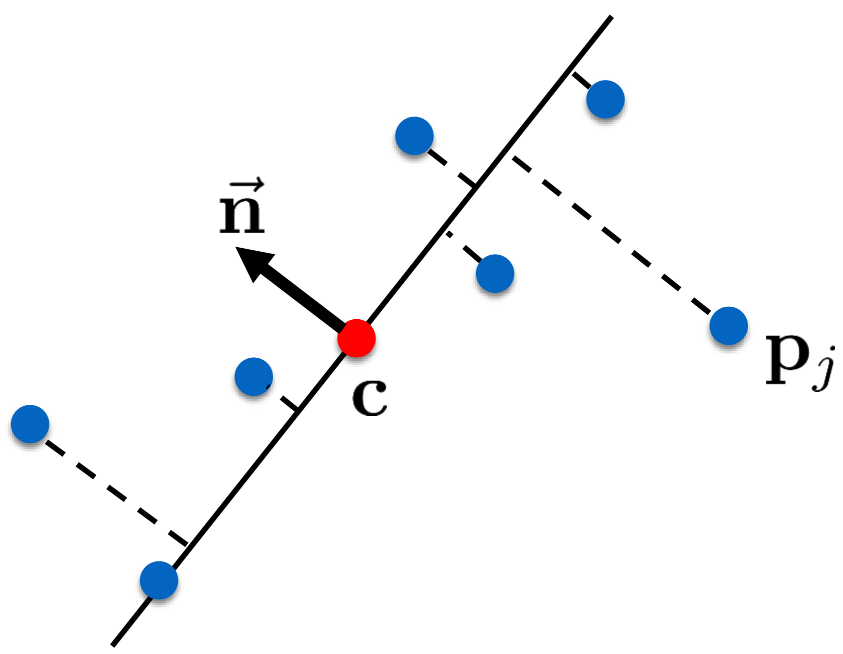
Principal Component Analysis

- Fit a plane with center $\mathbf{c}$ and normal $\vec{\mathbf{n}}$ to a set of points $\{\mathbf{p}_1, \ldots, \mathbf{p}_k\}$
  - Minimize least squares error

$$\min_{\mathbf{c},\vec{\mathbf{n}}} \sum_{j=0}^{k} (\vec{\mathbf{n}}^T(\mathbf{p}_j - \mathbf{c}))^2$$

  - Subject non-linear constraint

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



Principal Component Analysis

1. Compute barycenter (plane center)

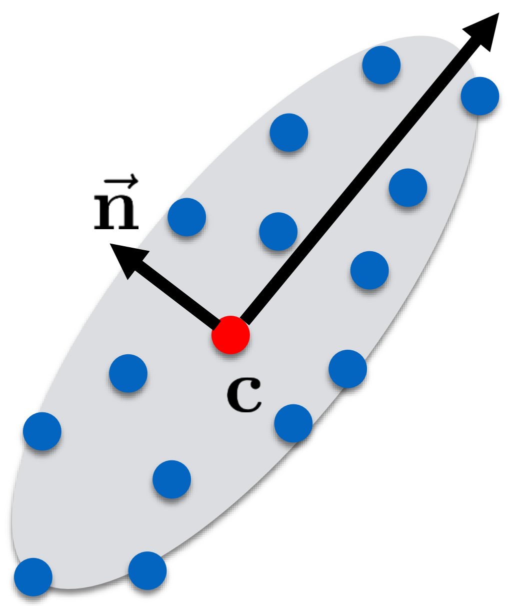
$$\mathbf{c} = \frac{1}{k} \sum_{j=0}^{k} \mathbf{p}_j$$

2. Compute covariance matrix

$$\mathbf{C} = \mathbf{M}\mathbf{M}^T \in \mathbb{R}^{3\times 3}$$

with

$$\mathbf{M} = [(\mathbf{p}_1 - \mathbf{c}), \ldots, (\mathbf{p}_k - \mathbf{c})] \in \mathbb{R}^{3\times k}$$

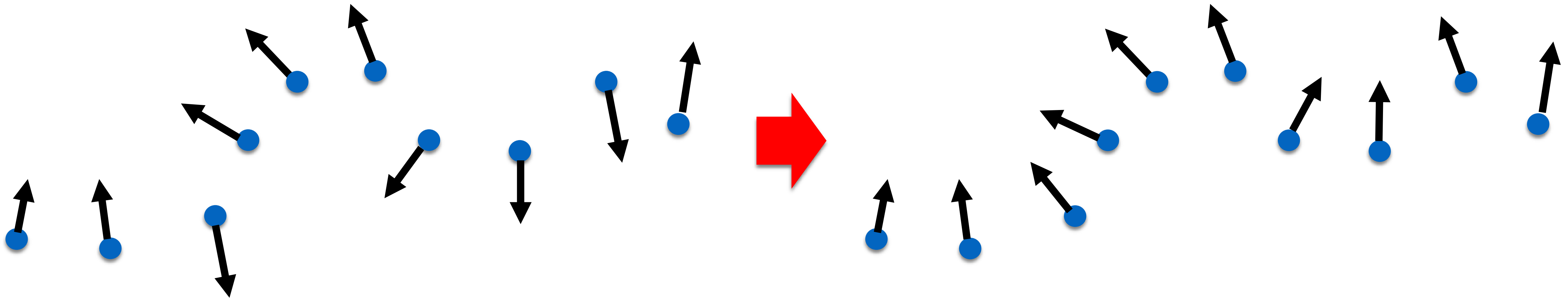


3. Select as normal the eigenvector of the covariance matrix with the smallest eigenvalue

Normal Estimation for Point Cloud

[Hoppe et al., SIGGRAPH 92]

- Estimate the normal vector for each point
  1. Extract the k-nearest neighbor point
  2. Compute the best approximating tangent plane by covariance analysis
  3. **Compute a coherent normal orientation**



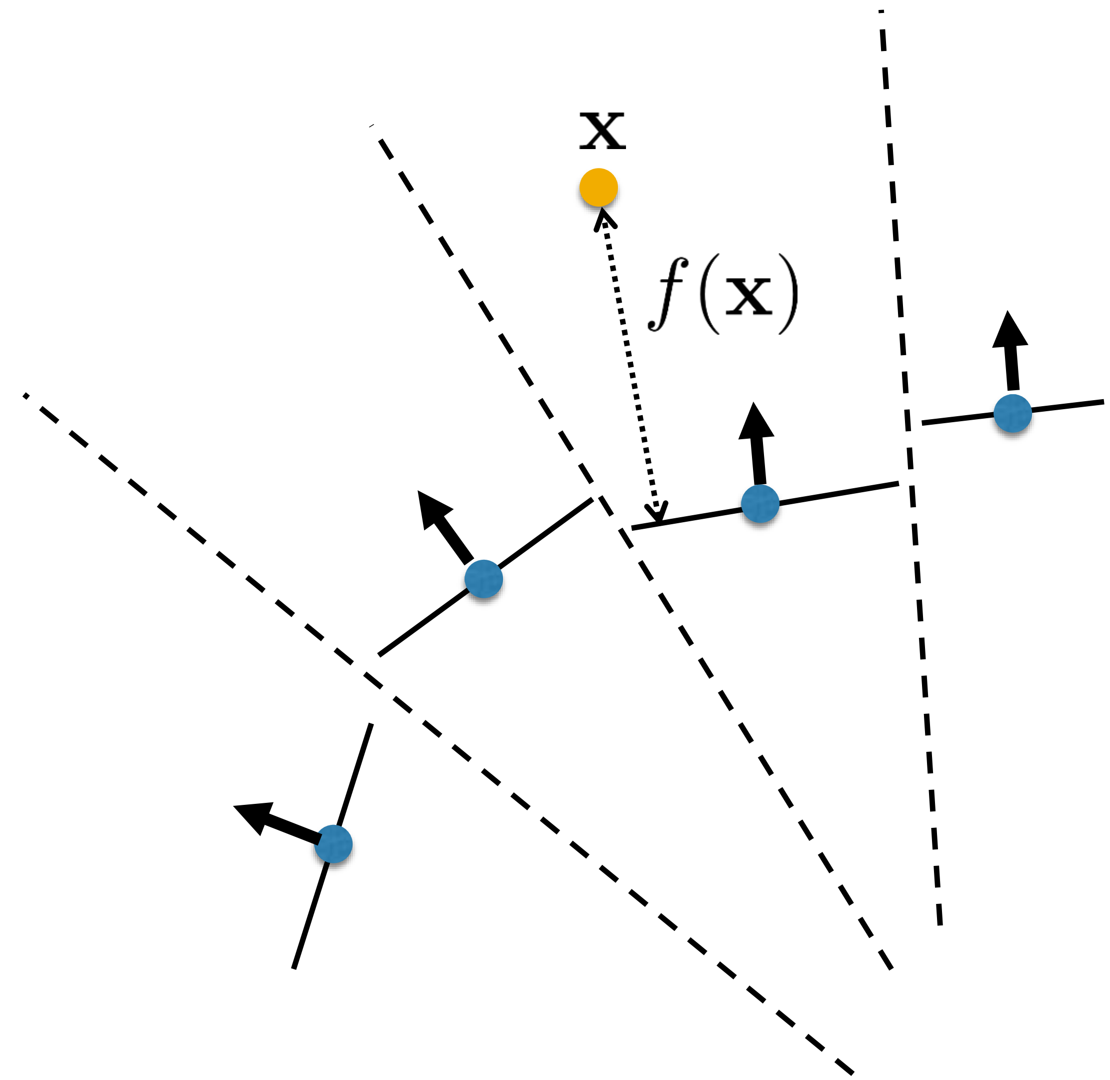
Normal Orientation

- Build graph connecting neighboring points
  - Edge (ij) exists if $\mathbf{p}_i \in \mathrm{kNN}(\mathbf{p}_j)$ or $\mathbf{p}_j \in \mathrm{kNN}(\mathbf{p}_i)$
- Propagate normal orientation through graph
  - For edge (ij) flip $\vec{\mathbf{n}_j}$ if $\vec{\mathbf{n}_j}^T \vec{\mathbf{n}_i} < 0$
  - Fails at sharp edges/corners
- Propagate along “safe” paths
  - Build a minimum spanning tree with angle-based edge weights $w_{ij} = 1 - \vec{\mathbf{n}_j}^T \vec{\mathbf{n}_i}$

SDF from tangent plane

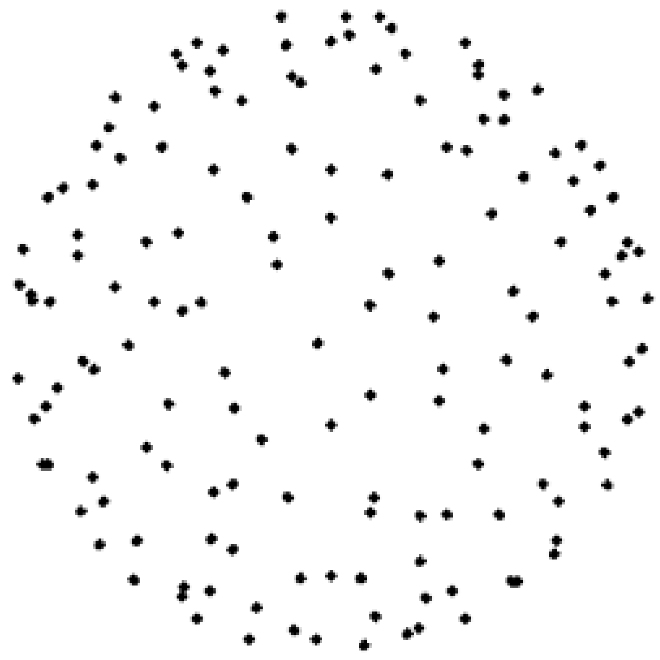
[Hoppe et al., SIGGRAPH 92]

- Signed distance from tangent planes
  - Points and normals determine local tangent planes
  - Use distance from closest point's tangent plane
  - Simple and efficient, but SDF is not continuous



SDF from tangent plane

[Hoppe et al., SIGGRAPH 92]



150 SAMPLES

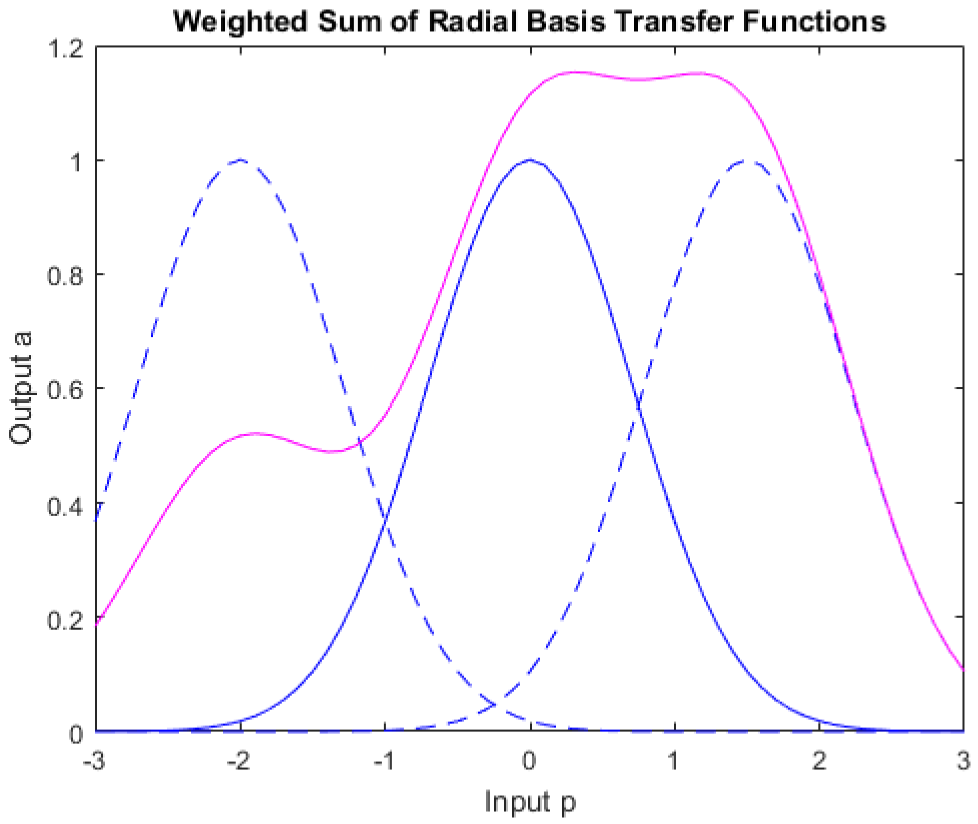
RECONSTRUCTION
WITH A 50^3 GRID

Smooth SDF Approximation

- Use radial basis functions (RBFs) to implicitly represent surface
  - Function such that the value depends only on the distance from the origin or from a center
  - Sum of radial basis functions used to approximate a function

$$\phi(\mathbf{x}) = \phi(\|\mathbf{x}\|)$$

$$\phi(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{c}\|)$$



Smooth SDF Approximation

[Carr et al., SIGGRAPH 01]

- Give the n input points $\{\mathbf{x}_i : f(\mathbf{x}_i) = 0\}$
- Approximate distance field with a shifted weighted sum of radial basis functions

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i \phi(\|\mathbf{x} - \mathbf{x}_i\|)$$

- Use the input points as centers of the radial functions
- Constrain:
  - The approximated SDF must be continuous and smooth

Estimate the RBF weight

[Carr et al., SIGGRAPH 01]

- Set a system of n equations

$$\forall j \;\; f(\mathbf{x}_j) = \sum_{i=1}^{n} w_i \phi(\|\mathbf{x}_j - \mathbf{x}_i\|)$$

$$f(\mathbf{x}_j) = d_j$$

- Solve a linear system

$$\mathbf{Aw} = \mathbf{d} \Rightarrow \begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \dots & \phi(\|\mathbf{x}_1 - \mathbf{x}_n\|) \\ \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}_n - \mathbf{x}_1\|) & \dots & \phi(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix}$$

Estimate the RBF weight

[Carr et al., SIGGRAPH 01]

- For the input point we have $f(\mathbf{x}_j) = d_j = 0$
- The RBF system is $\mathbf{A}\mathbf{w} = 0$
- Problem: It gets the trivial solution $f(\mathbf{x}) = 0$
- We need additional constrains
  - Off-surface point

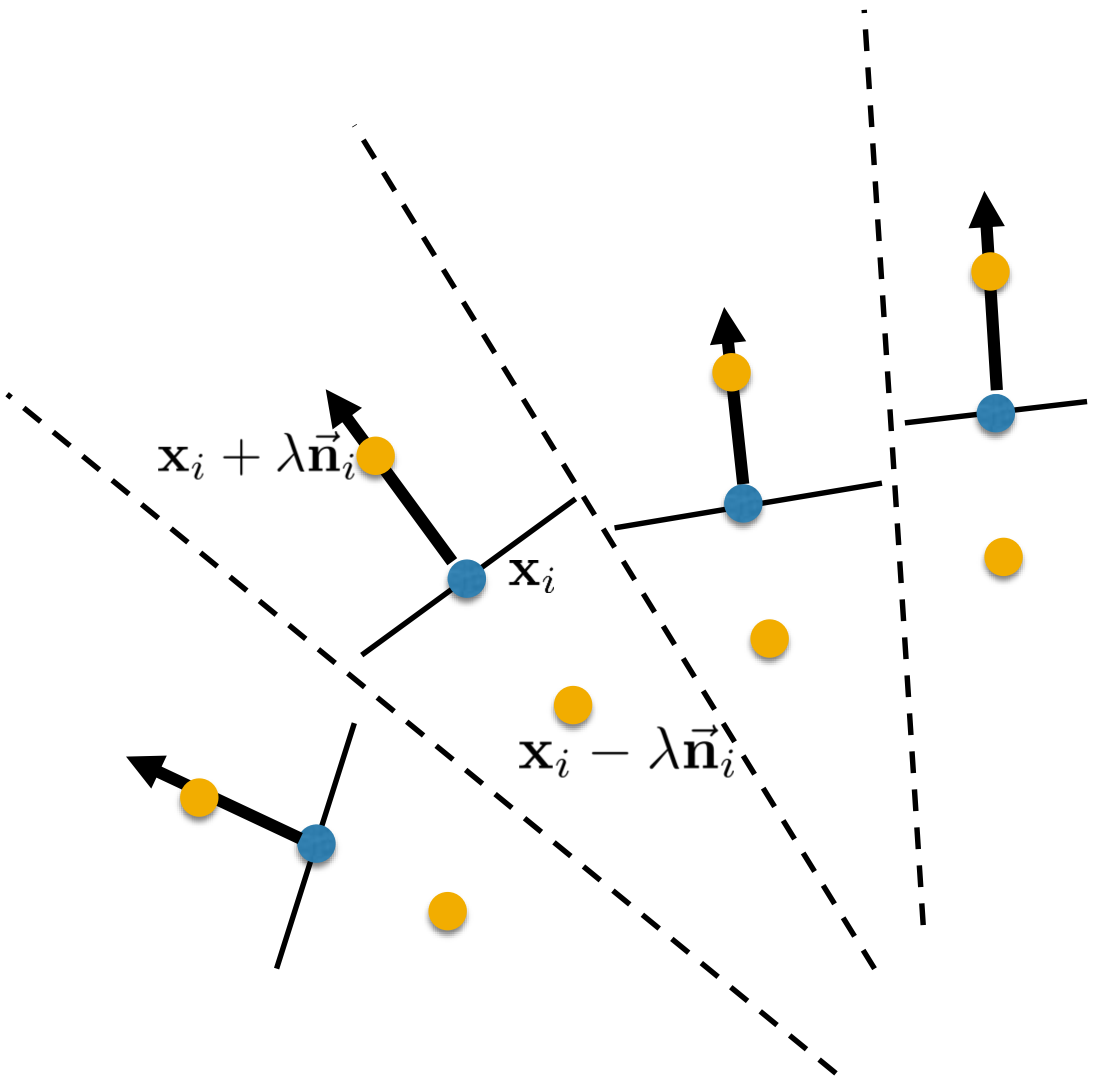
Off-surface Points

[Carr et al., SIGGRAPH 01]

- For each point in data add 2 off-surface points on both sides of surface

- Use normal data to find off-surface points

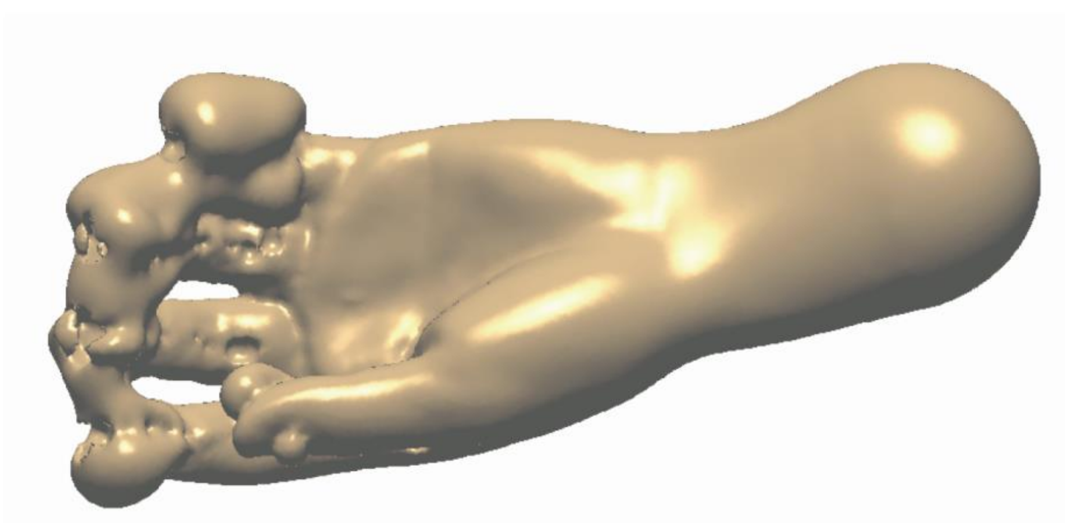
$$f(\mathbf{x}_i) = 0$$
$$f(\mathbf{x}_i + \lambda\vec{\mathbf{n}}_i) = \lambda$$
$$f(\mathbf{x}_i - \lambda\vec{\mathbf{n}}_i) = -\lambda$$



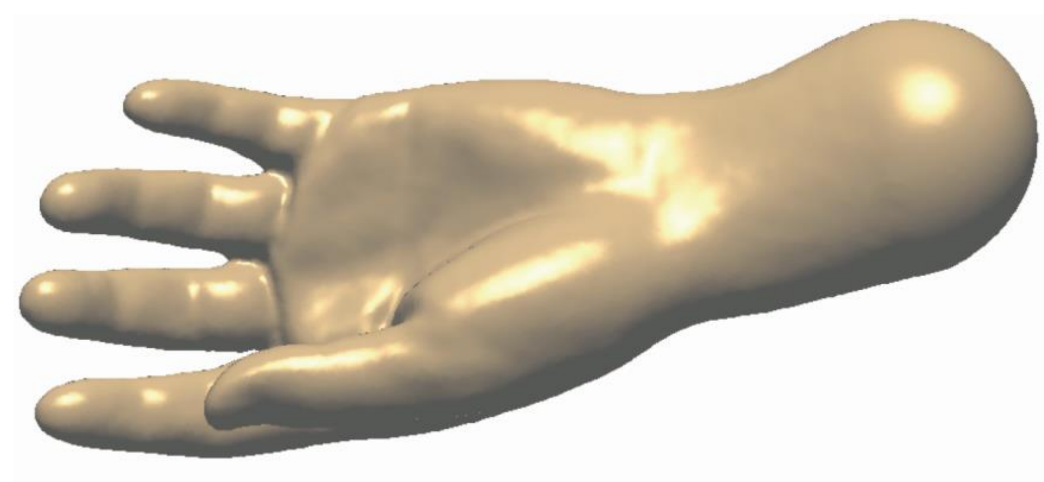
Off-surface Points

[Carr et al., SIGGRAPH 01]

- Select an offset such that off-surface points do not intersect other parts of the surface
- Adaptive offset: the off-surface point is constructed so that the closest point is the surface point that generated it



FIXED OFFSET



ADAPTIVE OFFSET

Radial Basis Function

- **Wendland basis functions**

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

- Compactly supported in $[0, \sigma]$
- Leads to sparse, symmetric positive-definite linear system
- Resulting SDF C^2 is smooth
- But surface is not necessarily fair
- Not suited for highly irregular sampling

Radial Basis Function

- **Triharmonic basis functions**

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

- Globally supported function
- Leads to dense linear system
- SDF C^2 is smooth
- Provably optimal fairness
- Works well for irregular sampling

Radial Basis Function

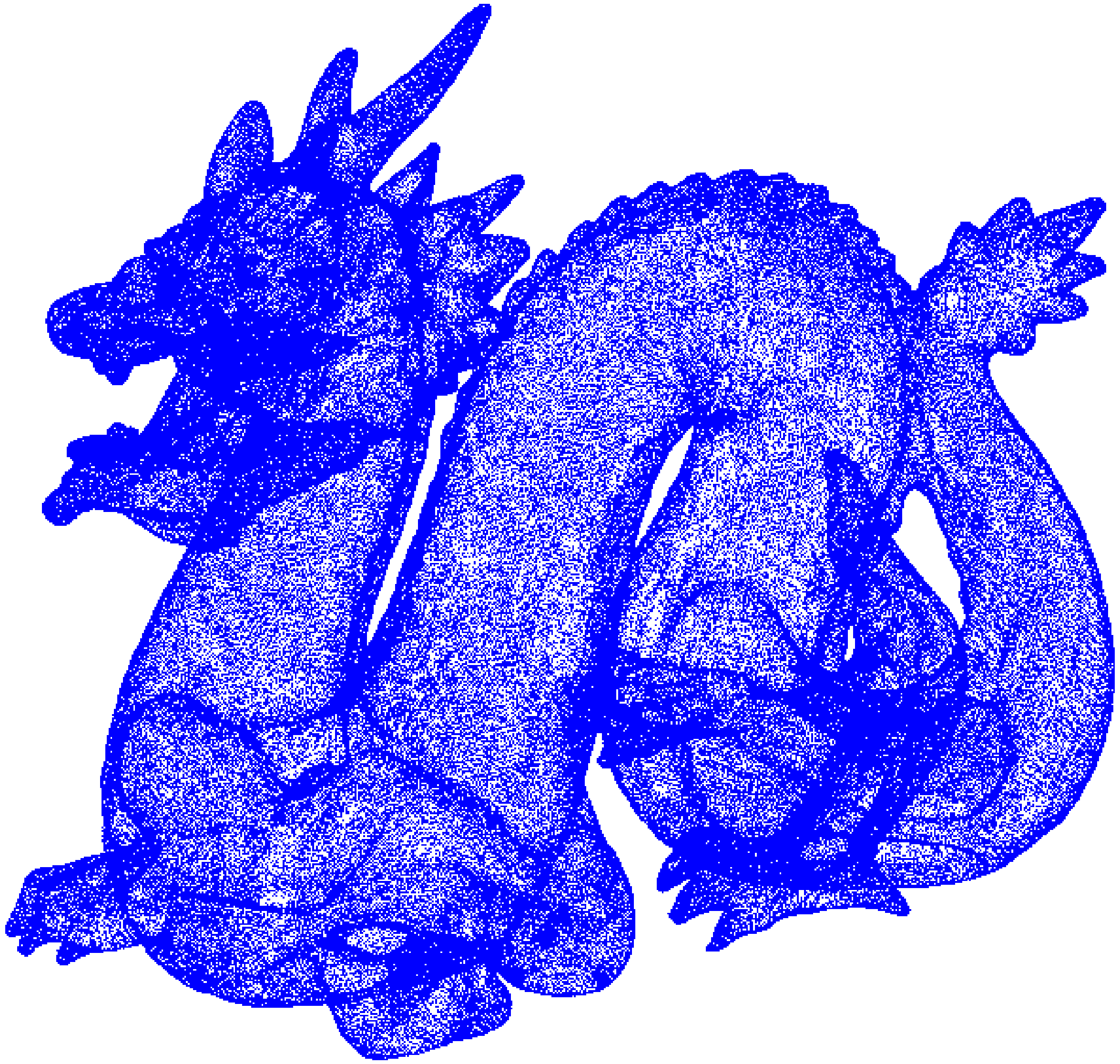
SDF FROM
TANGENT PLANE

RBF
WENDLAND

RBF
TRIHARMONIC

RBF Reconstruction Example

[Carr et al., SIGGRAPH 01]



SDF from Range Scan

[Curless et al., SIGGRAPH 96]

- Compute the SDF for each range scan $f(\mathbf{x}_i)$
  - Distance along scanner's line of sight
- Compute a weighting function for each scan $w(\mathbf{x}_i)$
  - Use of different weights
- Compute global SDF by weighted average

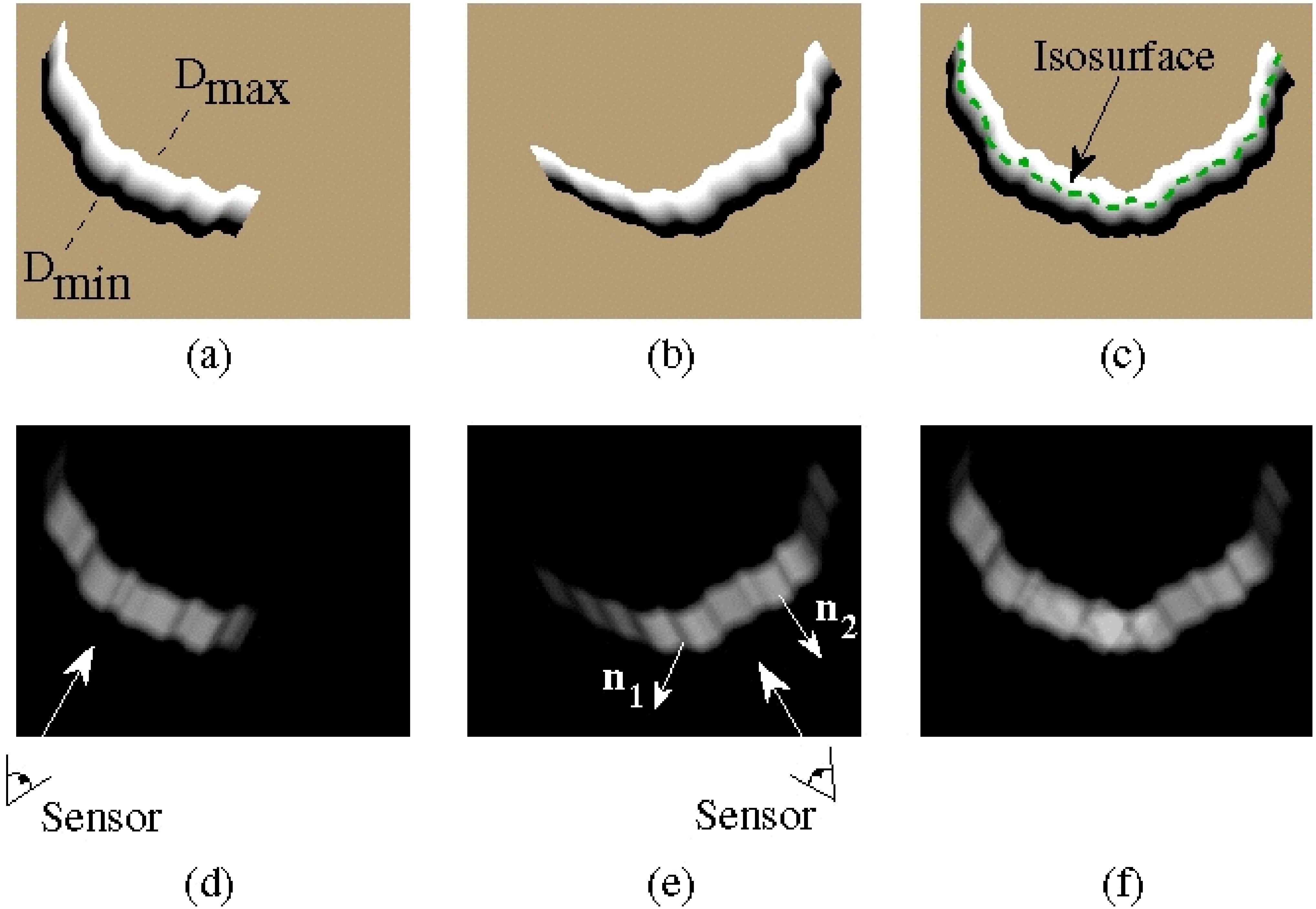
$$F(\mathbf{x}) = \frac{\sum_i w_i f_i(\mathbf{x})}{\sum_i w_i}$$

SDF from Range Scan

[Curless et al., SIGGRAPH 96]

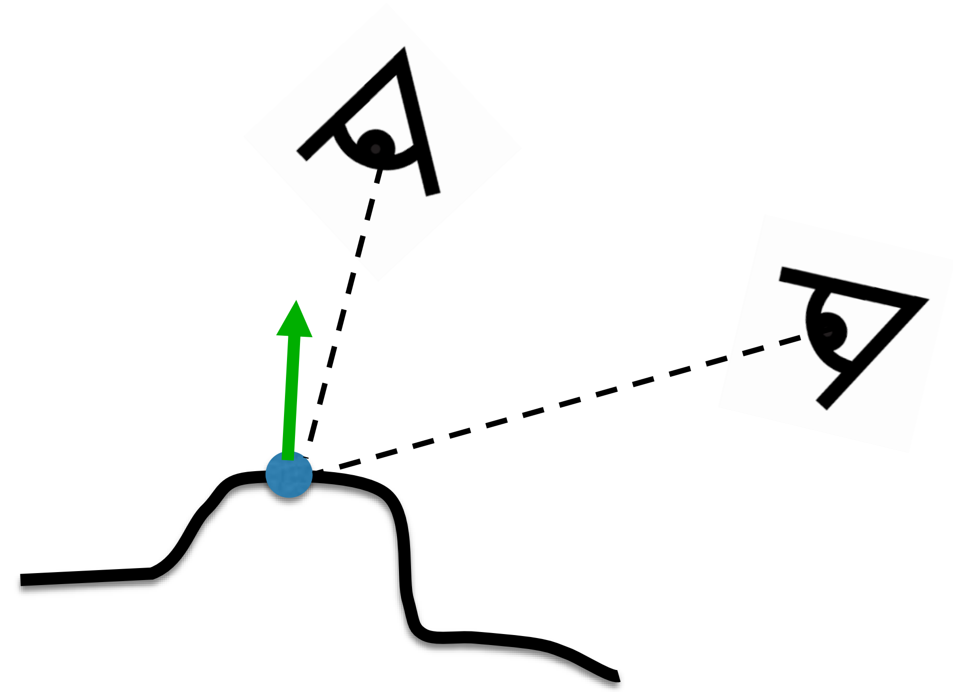
SDF

WEIGHTS

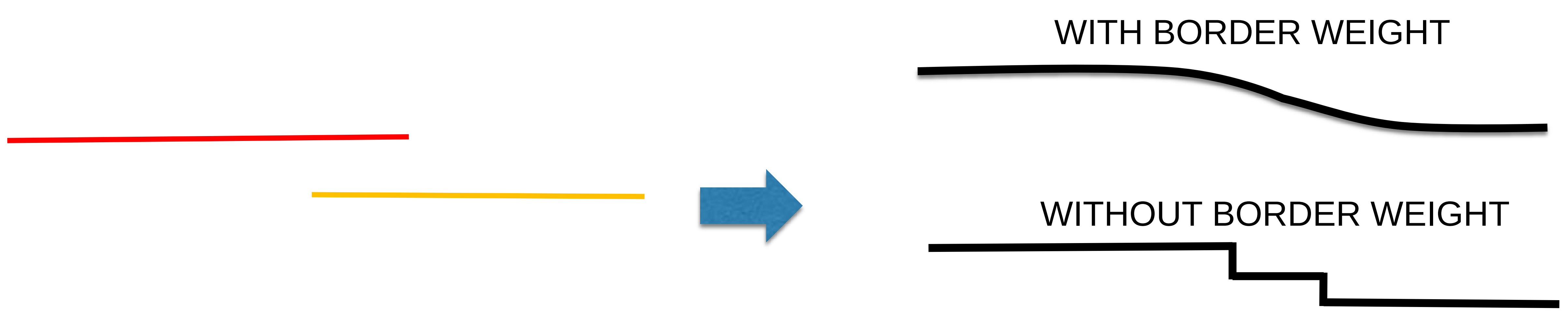


SDF from Range Scan

- Weighting functions
  - Scanning angle

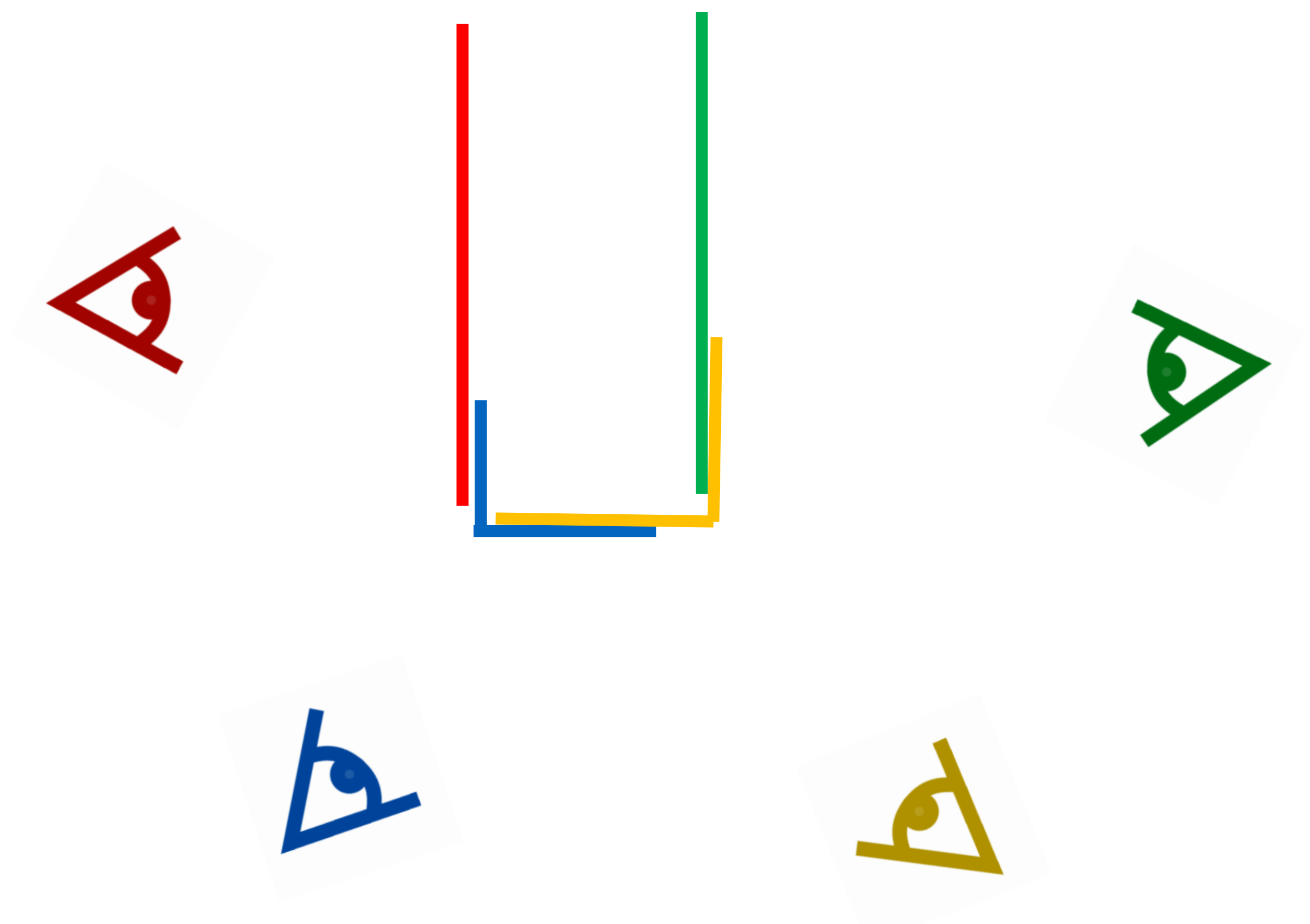


  - Distance from the border of the scan



SDF from Range Scan

- Restrict the function near the surface to avoid interference with other scans



Moving Least Square

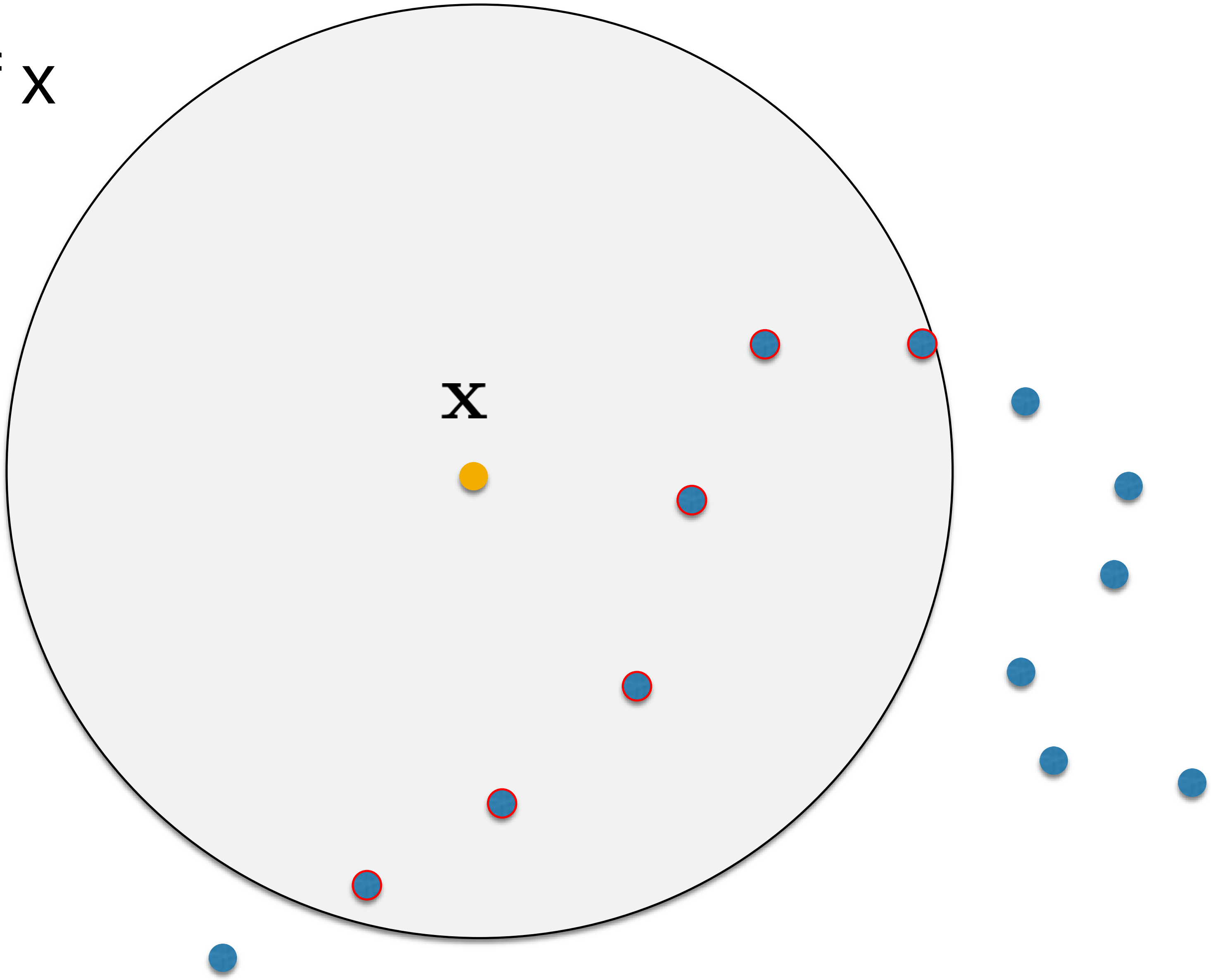
[Alexa et al., VIS 01]

- Approximates a smooth surface from irregularly sampled points
- Create a local estimate of the surface at every point in space
- Implicit function is computed by local approximations
- Projection operator that projects points onto the MSL surface

Moving Least Square

[Alexa et al., VIS 01]

- How to project x on the surface defined by the input points
  1. Get Neighborhood of x



Moving Least Square

[Alexa et al., VIS 01]

- How to project x on the surface defined by the input points

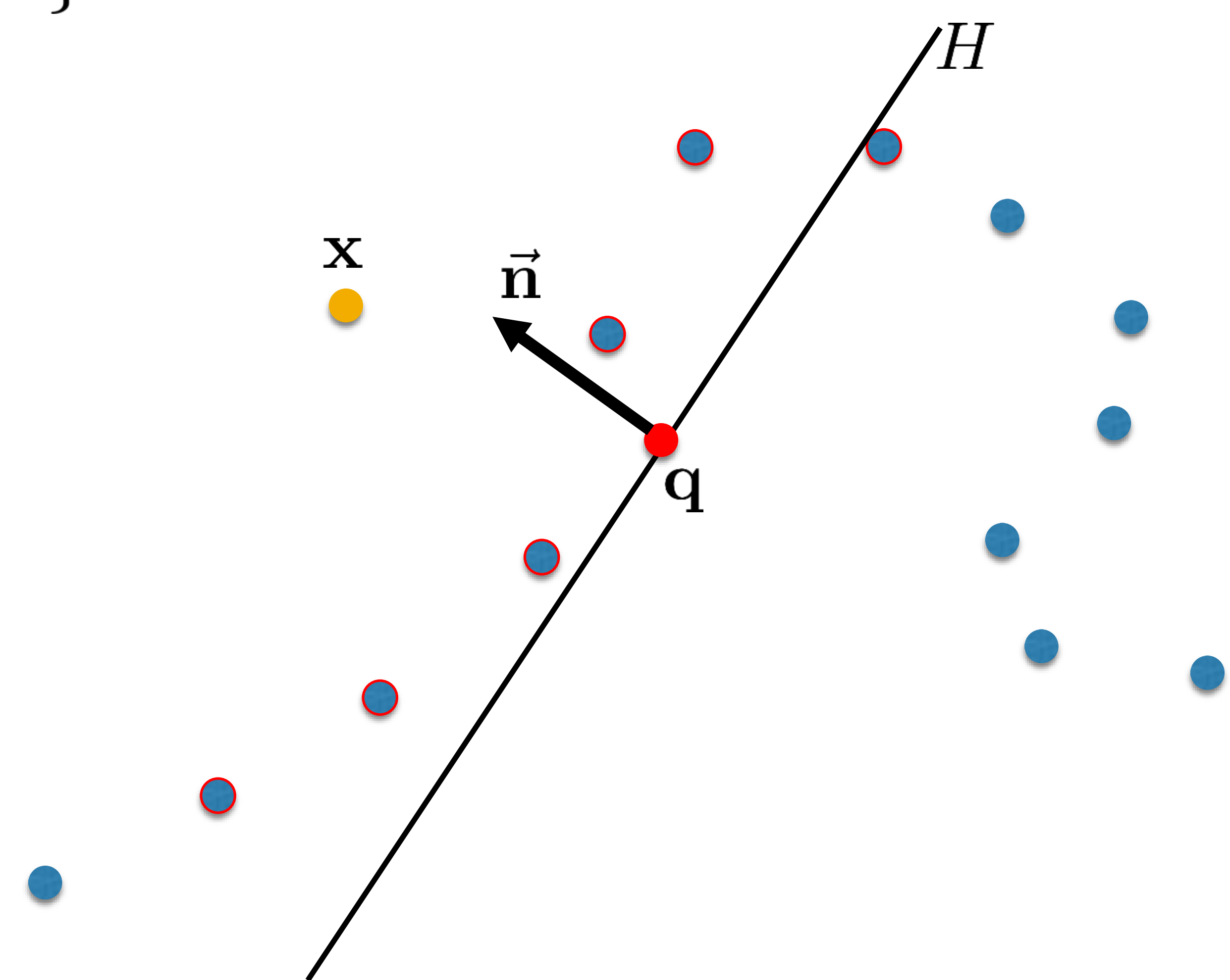
2. Find a local reference plane

$$H = \{\mathbf{x} \in \mathbb{R}^3 | \vec{\mathbf{n}}^T(\mathbf{x} - \mathbf{q}) = 0\}$$

minimizing the energy

$$\sum_i (\vec{\mathbf{n}}^T(\mathbf{p_i} - \mathbf{q}))^2 \theta(\|\mathbf{p_i} - \mathbf{q}\|)$$

Smooth, positive, and monotonically decreasing weight function



Moving Least Square

[Alexa et al., VIS 01]

- How to project x on the surface defined by the input points

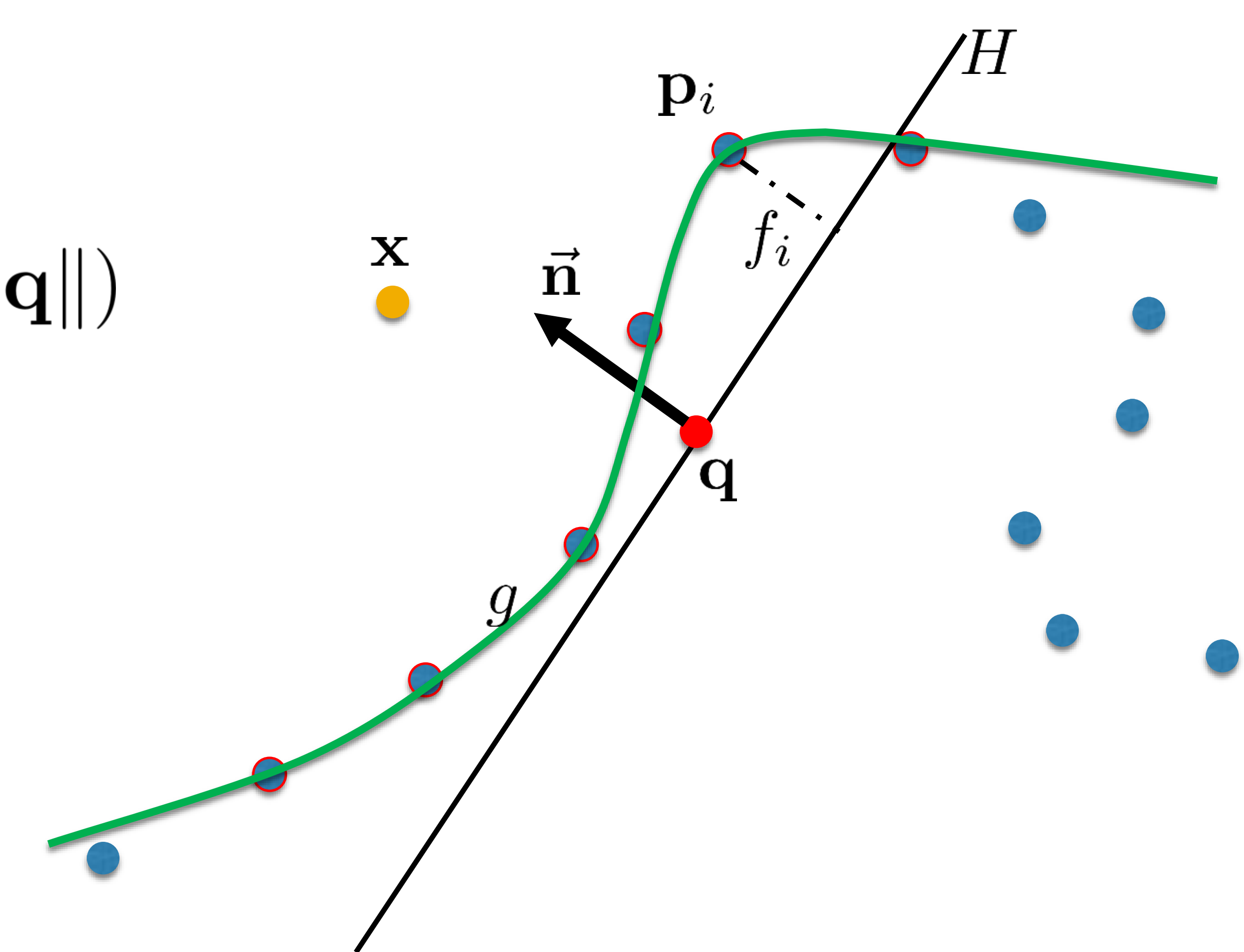
3. Find a polynomial approximation

$$g : H \to \mathbb{R}^3$$

minimizing the energy

$$\sum_i (g(x_i, y_i) - f_i)^2 \theta(\|\mathbf{p_i} - \mathbf{q}\|)$$

2D coordinate of the projection on H



Moving Least Square

[Alexa et al., VIS 01]

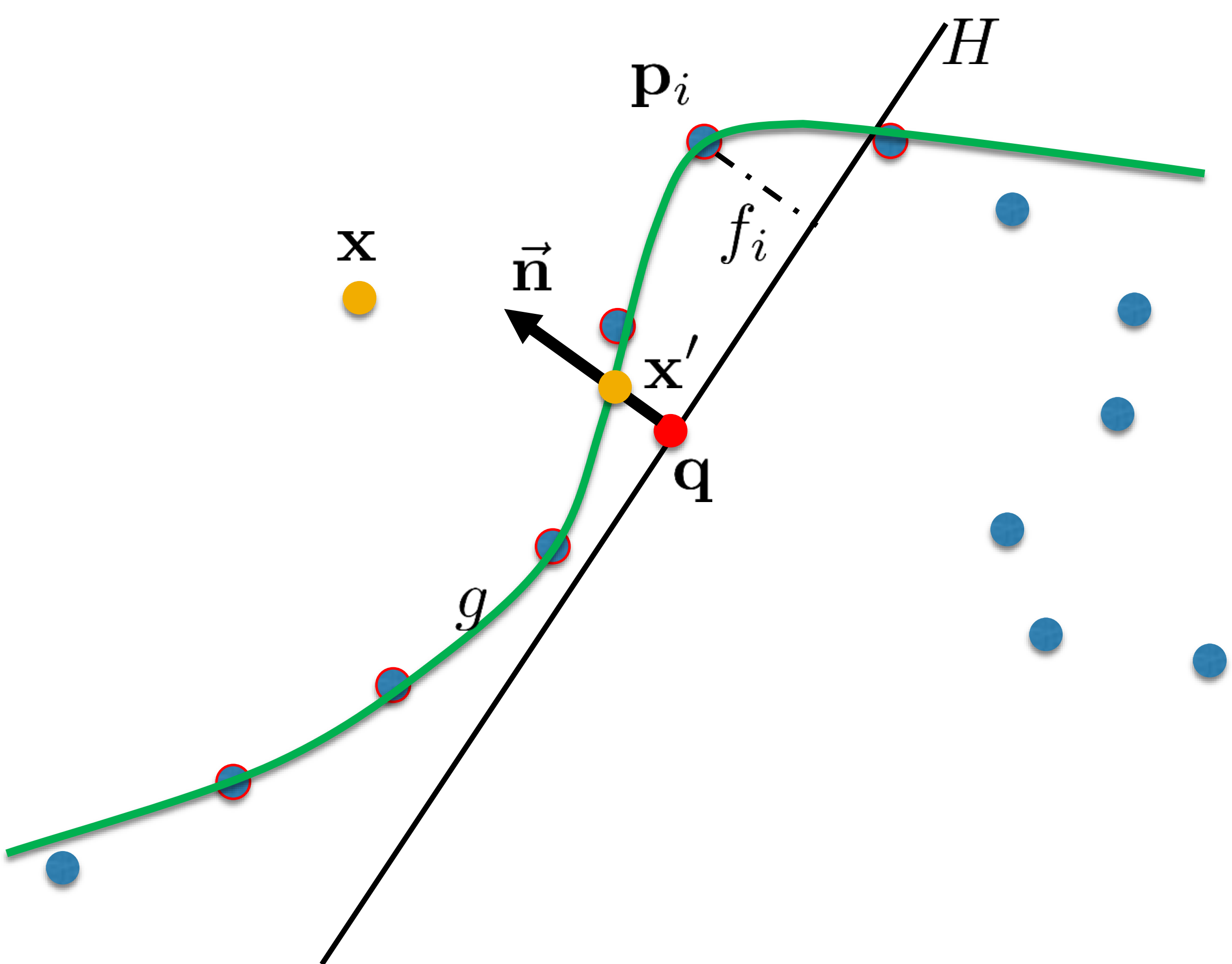
- How to project x on the surface defined by the input points

4. Projection of x

$$\mathbf{x}' = \mathbf{q} + g(0,0)\vec{\mathbf{n}}$$

5. Iterate if

$$g(0,0) > \epsilon$$

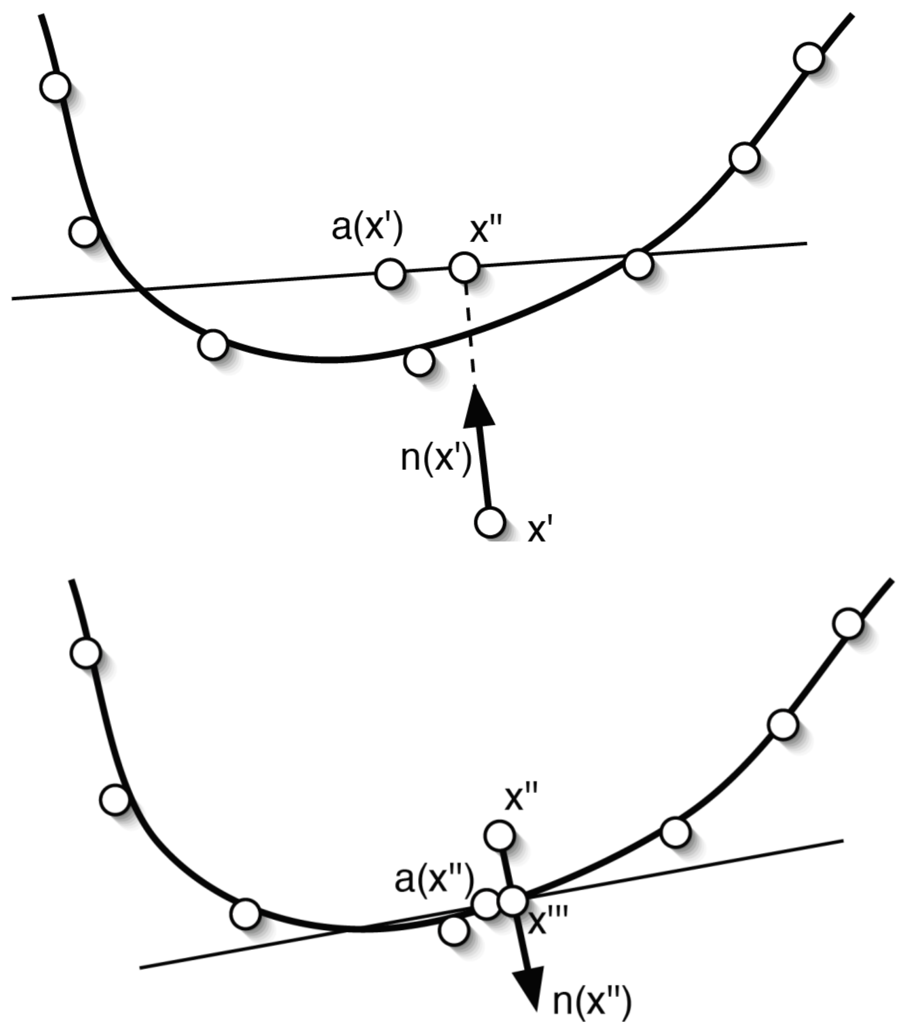


Moving Least Square

- Simpler projection approach using weighted average position and normal

$$\mathbf{a}(\mathbf{x}) = \frac{\sum_i \theta(\|\mathbf{x}-\mathbf{p}_i\|)\mathbf{p}_i}{\sum_i \theta(\|\mathbf{x}-\mathbf{p}_i\|)}$$

$$\vec{\mathbf{n}}(\mathbf{x}) = \frac{\sum_i \theta(\|\mathbf{x}-\mathbf{p}_i\|)\vec{\mathbf{n}}_i}{\|\sum_i \theta(\|\mathbf{x}-\mathbf{p}_i\|)\vec{\mathbf{n}}_i\|}$$



1) $\mathbf{x}' \leftarrow \mathbf{a}(\mathbf{x})$

2) $\vec{\mathbf{n}} \leftarrow \vec{\mathbf{n}}(\mathbf{x}')$

3) $\mathbf{a} \leftarrow \mathbf{a}(\mathbf{x}')$

4) if $\vec{\mathbf{n}}^T(\mathbf{a}-\mathbf{x}') < \epsilon$ return $\mathbf{x}'$

5) else $\mathbf{x}' \leftarrow \mathbf{x}' + \vec{\mathbf{n}}\vec{\mathbf{n}}^T(\mathbf{a}-\mathbf{x}')$ go to 2)

[Alexa et al., SPBG 04]

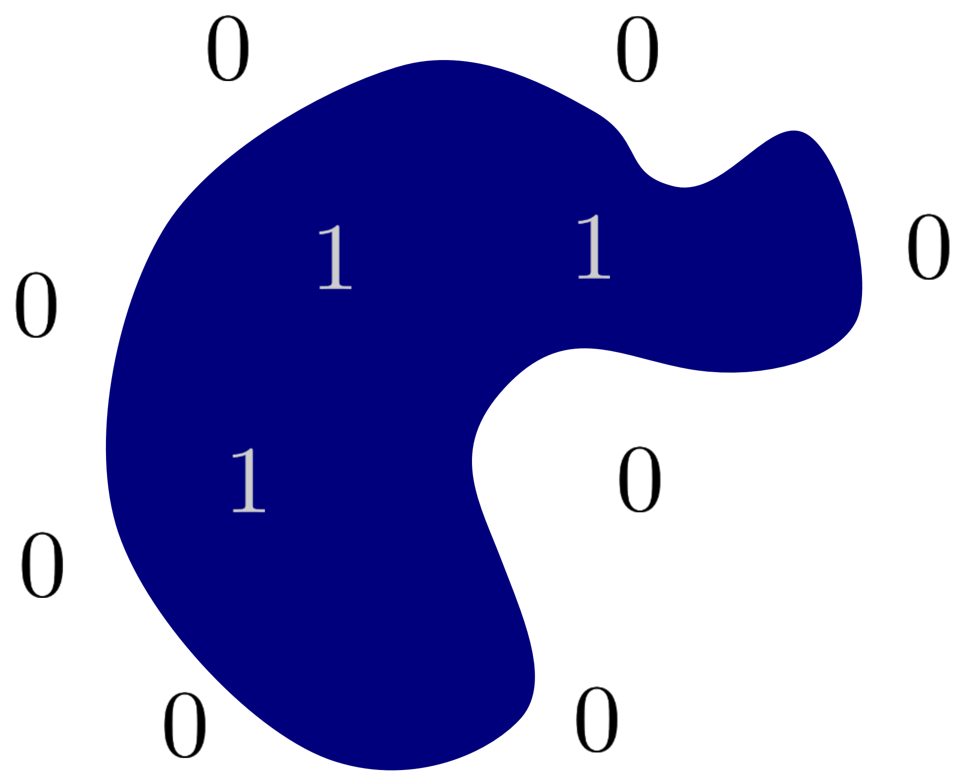
Moving Least Square

Poisson Surface Reconstruction

[Kazhdan et al., SGP 06]

- Reconstruct the surface of the model by solving for the indicator function of the shape

$$\chi_M(p) = \begin{cases} 1 & \text{if } p \in M \\ 0 & \text{if } p \notin M \end{cases}$$

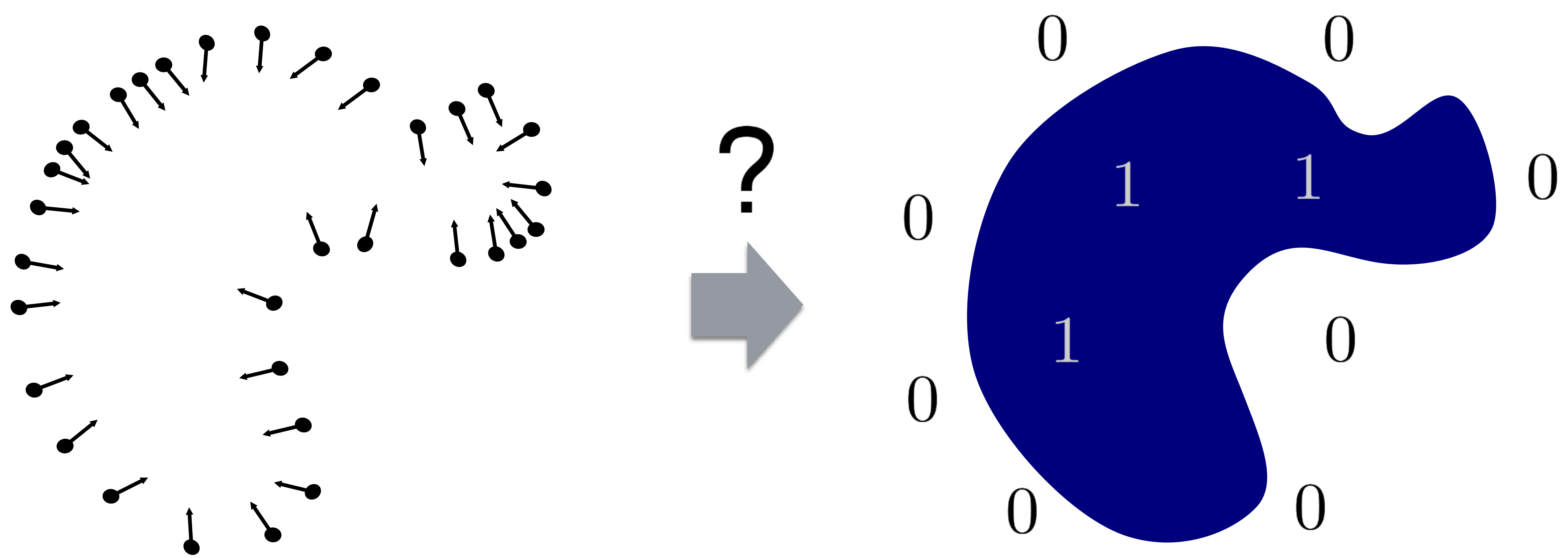


Indicator function χ_M

Indicator Function

[Kazhdan et al., SGP 06]

- How to compute the indicator function?



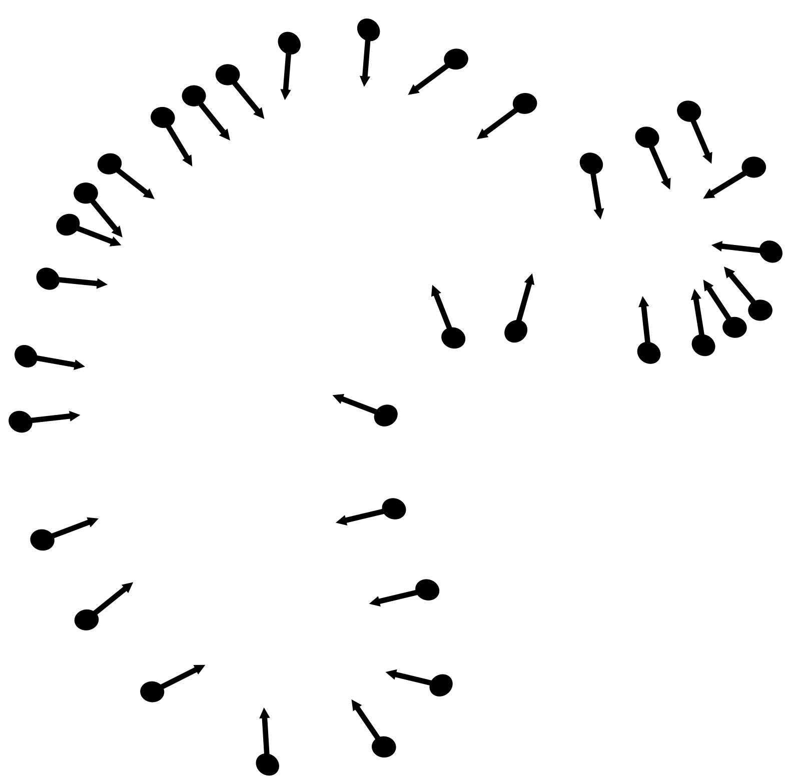
Oriented points

Indicator function χ_M

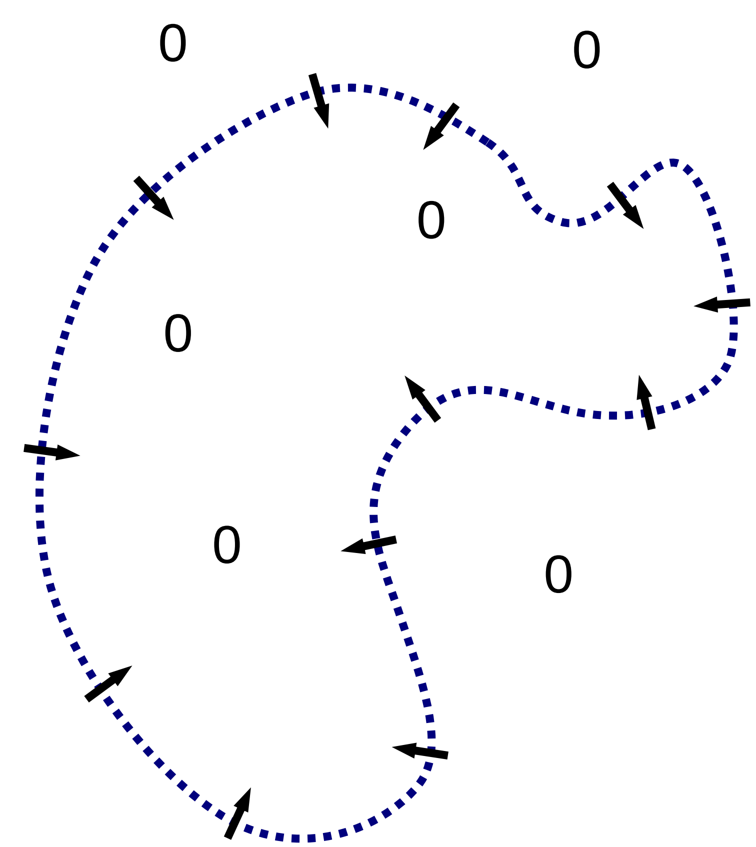
Indicator Function

[Kazhdan et al., SGP 06]

- The gradient of the indicator function is a vector field that is zero almost everywhere except at points near the surface, where it is equal to the inward surface normal



Oriented points



Indicator gradient $\nabla\chi_M$

Integration as a Poisson Problem

[Kazhdan et al., SGP 06]

- Represent the points by a vector field $\vec{V}$
- Find the function χ whose gradient best approximates $\vec{V}$:

$$\min_{\chi} \|\nabla\chi - \vec{V}\|$$

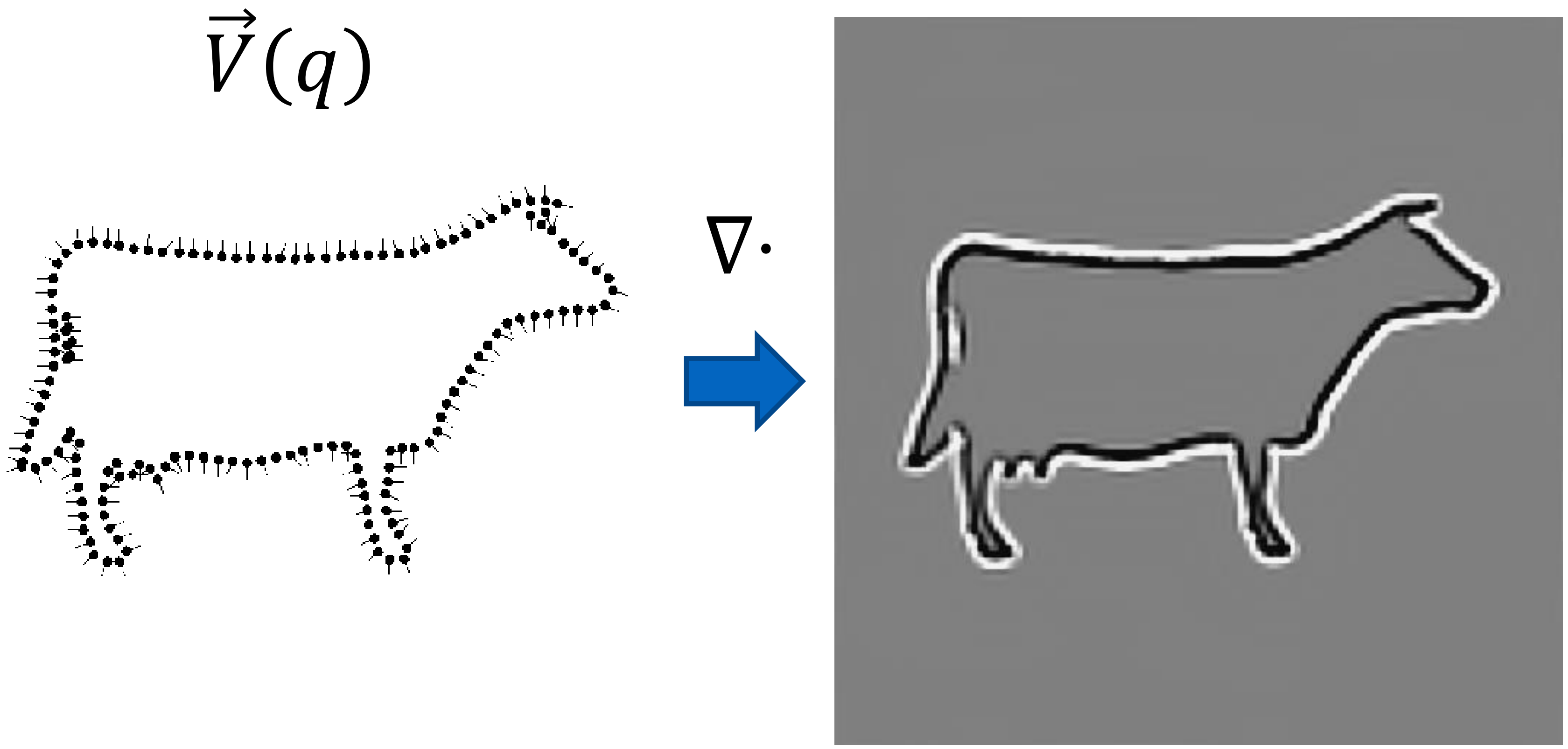
- Applying the divergence operator, we can transform this into a Poisson problem:

$$\nabla \cdot (\nabla\chi) = \nabla \cdot \vec{V} \quad \Leftrightarrow \quad \Delta\chi = \nabla \cdot \vec{V}$$

Poisson Surface Reconstruction

[Kazhdan et al., SGP 06]

1. Compute the divergence



Poisson Surface Reconstruction

[Kazhdan et al., SGP 06]

1. Compute the divergence

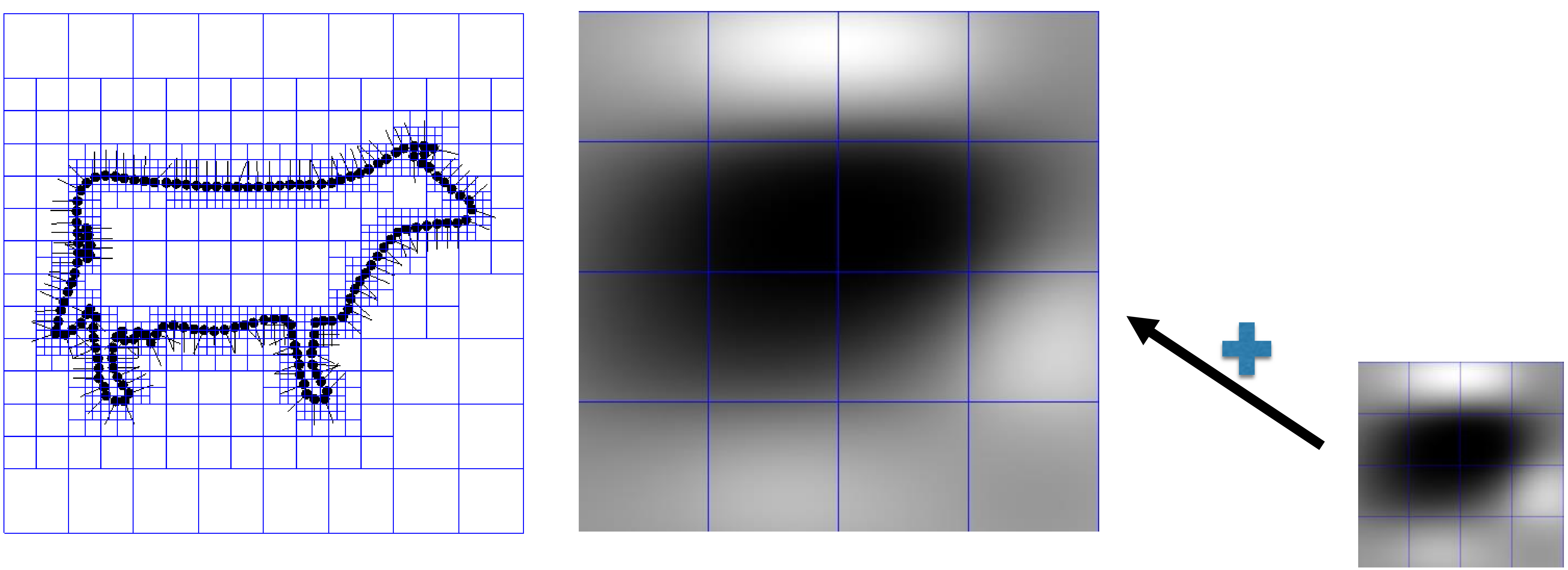
2. Solve the Poisson equation

Poisson Surface Reconstruction

[Kazhdan et al., SGP 06]

Solve the Poisson equation

- Discretize over an octree
- Update coarse → fine

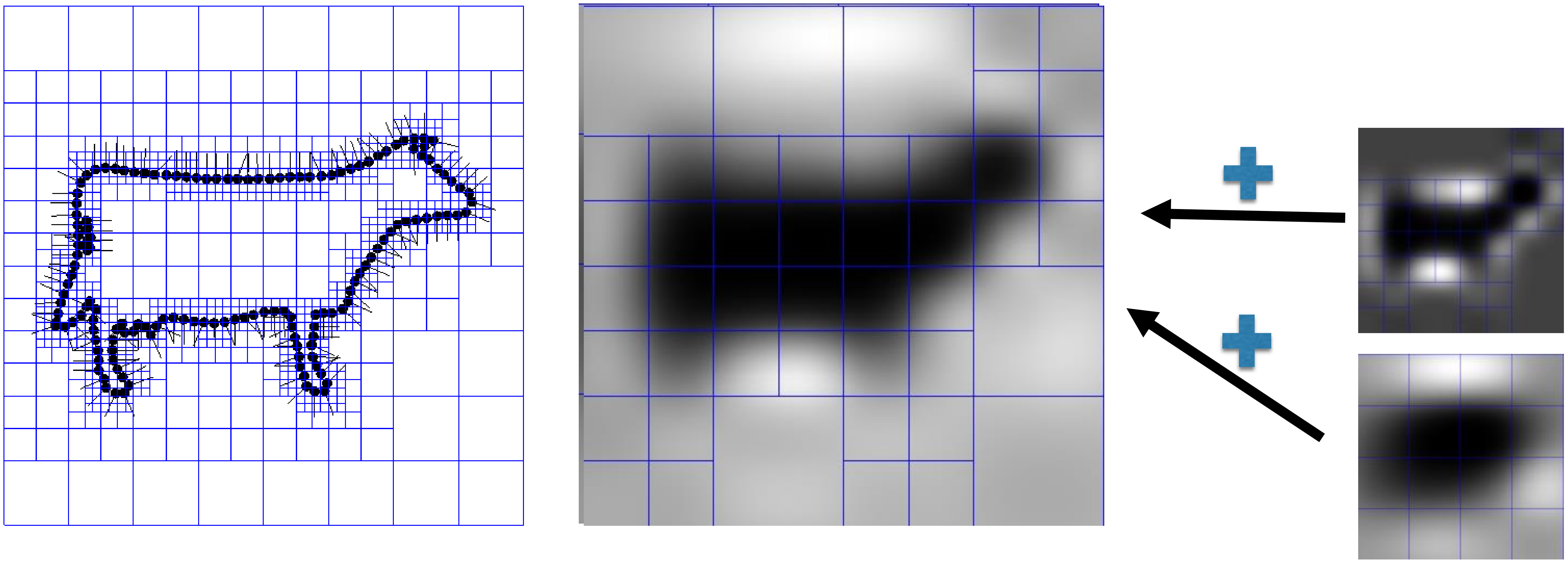


Poisson Surface Reconstruction

[Kazhdan et al., SGP 06]

Solve the Poisson equation

- Discretize over an octree
- Update coarse → fine

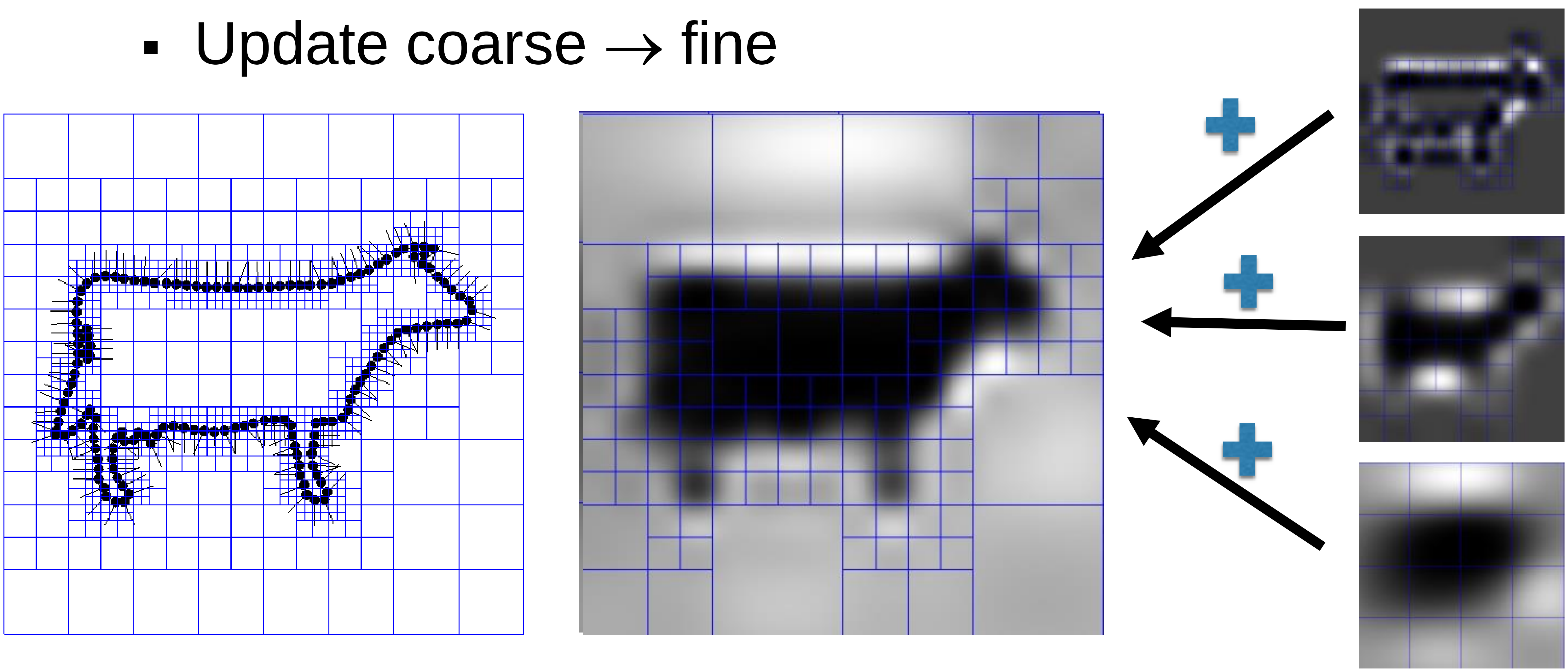


Poisson Surface Reconstruction

[Kazhdan et al., SGP 06]

Solve the Poisson equation

- Discretize over an octree
- Update coarse $\rightarrow$ fine

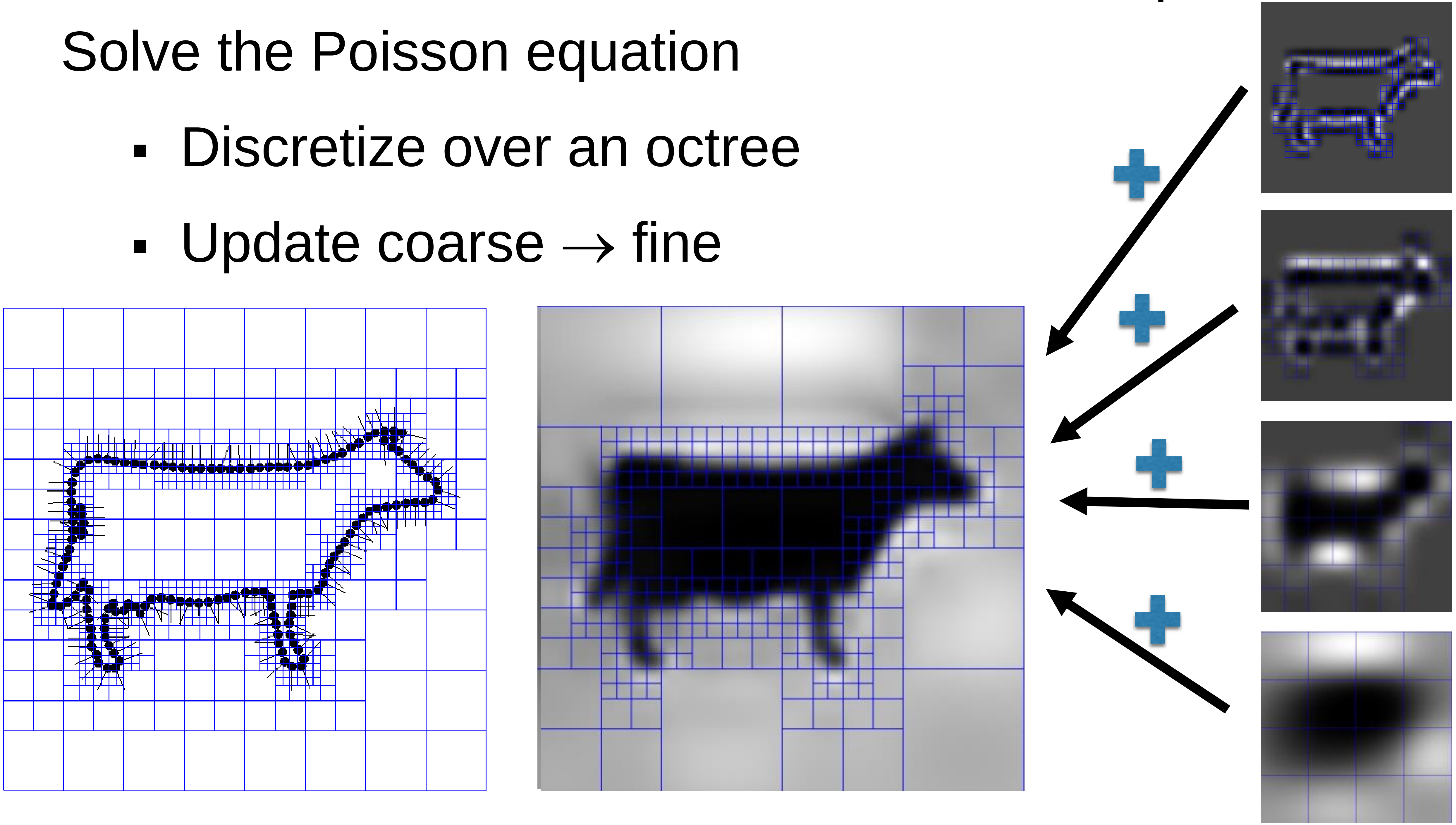


Poisson Surface Reconstruction

[Kazhdan et al., SGP 06]

Solve the Poisson equation

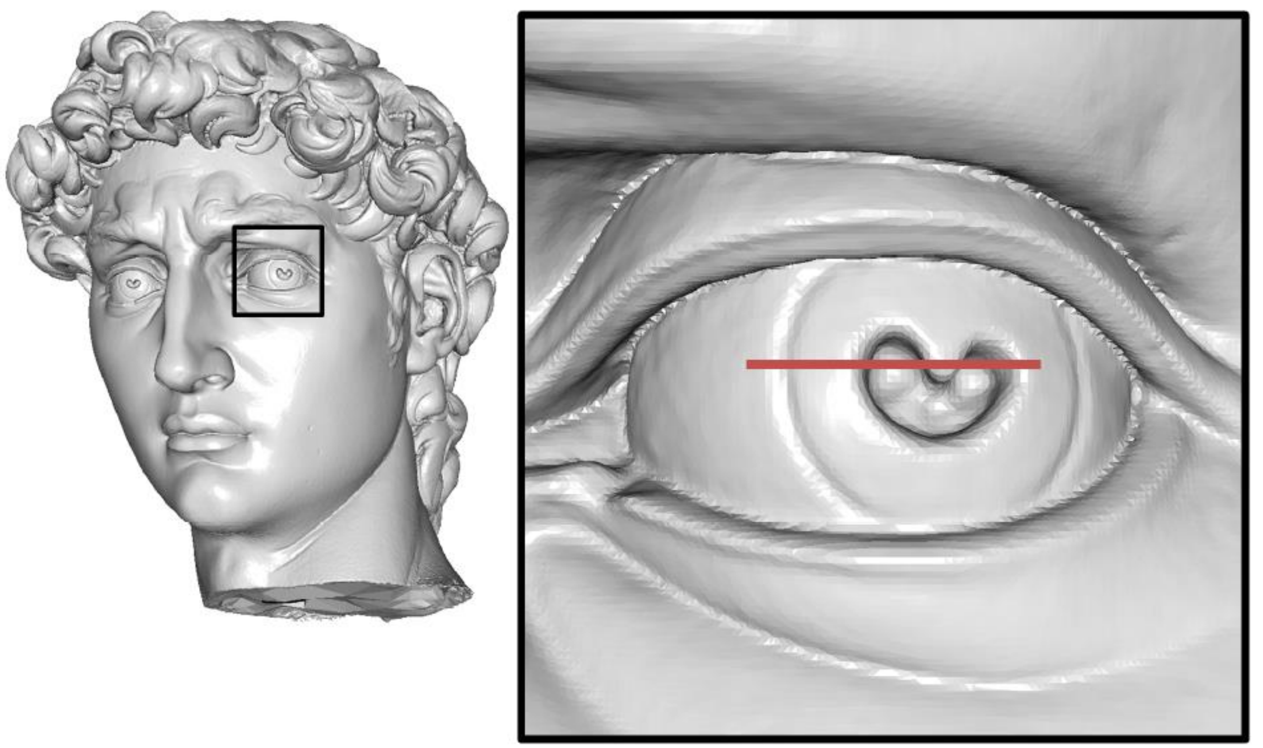
- Discretize over an octree
- Update coarse $\rightarrow$ fine

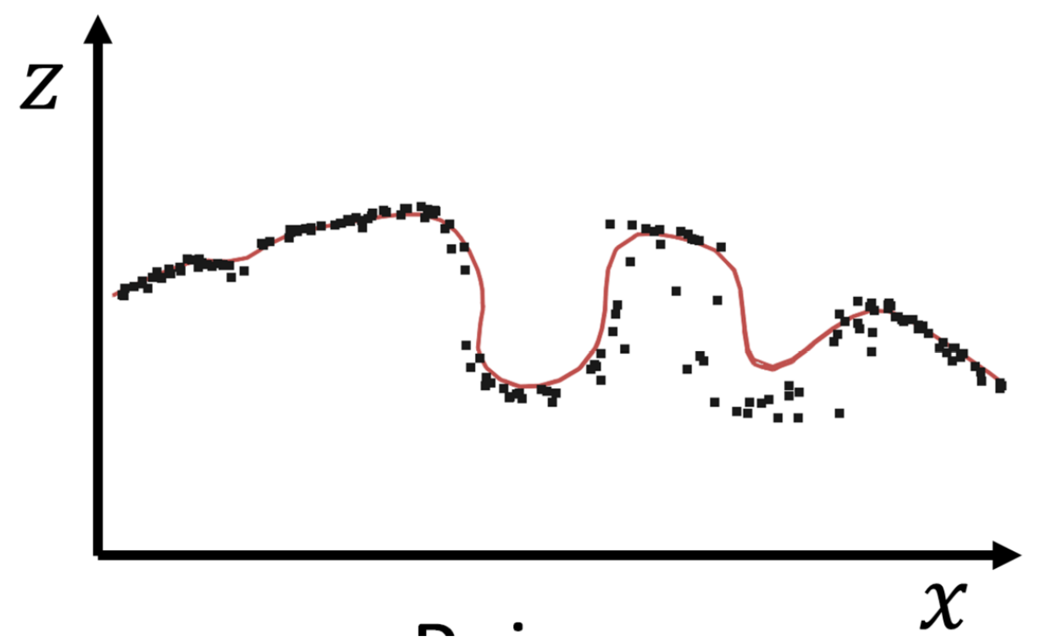


Poisson Surface Reconstruction

[Kazhdan et al., SGP 06]

- Advantages:
  - Robust to noise
  - Adapt to the sampling density
  - Can handle large models
- Disadvantages
  - Over-smoothing





Smooth Signed Distance Surface Reconstruction

[Calakli et al., PG 11]

- Oriented point set $D = \{(\mathbf{p}_i, \vec{\mathbf{n}_i})\}$
- Implicit surface $S = \{\mathbf{x} \mid f(\mathbf{x}) = 0\}$

$$\forall(\mathbf{p}_i, \vec{\mathbf{n}_i}) \ f(\mathbf{p}_i) = 0 \text{ and } \nabla f(\mathbf{p}_i) = \vec{\mathbf{n}_i}$$

- Least square energy (data term and regularization term)

$$E(f) = E_D(f) + E_R(f)$$

$$E_D(f) = \sum f(\mathbf{p}_i)^2 + \lambda_1 \sum \|\nabla f(\mathbf{p}_i) - \vec{\mathbf{n}}_i\|^2$$

$$E_R(f) = \lambda_2 \int_V \|\mathbf{H} f(\mathbf{x})\|^2 d\mathbf{x}$$

Smooth Signed Distance Surface Reconstruction

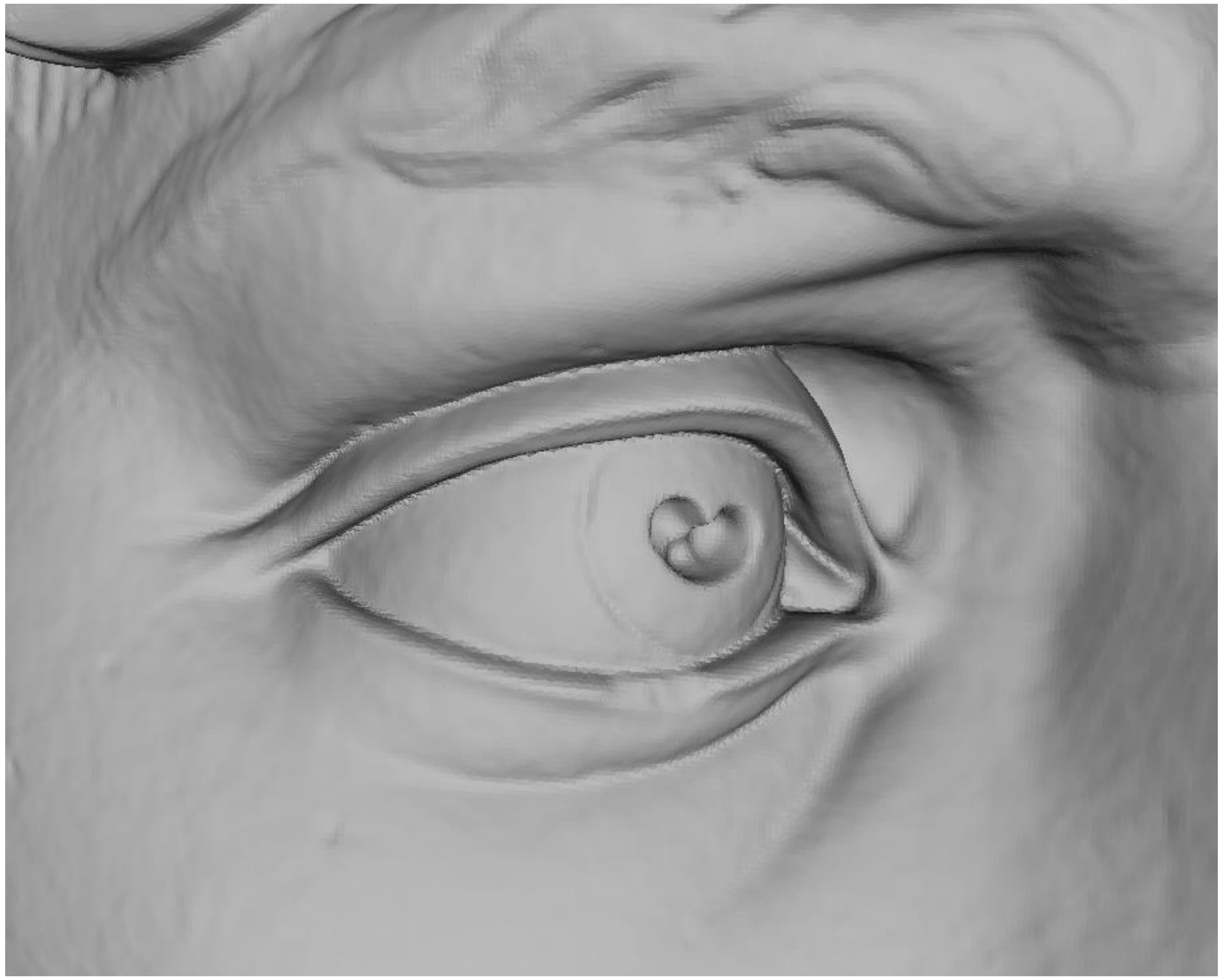
[Calakli et al., PG 11]

$$E(f) = \sum f(\mathbf{p}_i)^2 + \lambda_1 \sum \|\nabla f(\mathbf{p}_i) - \vec{\mathbf{n}}_i\|^2 + \lambda_2 \int_V \|\mathbf{H} f(\mathbf{x})\|^2 d\mathbf{x}$$

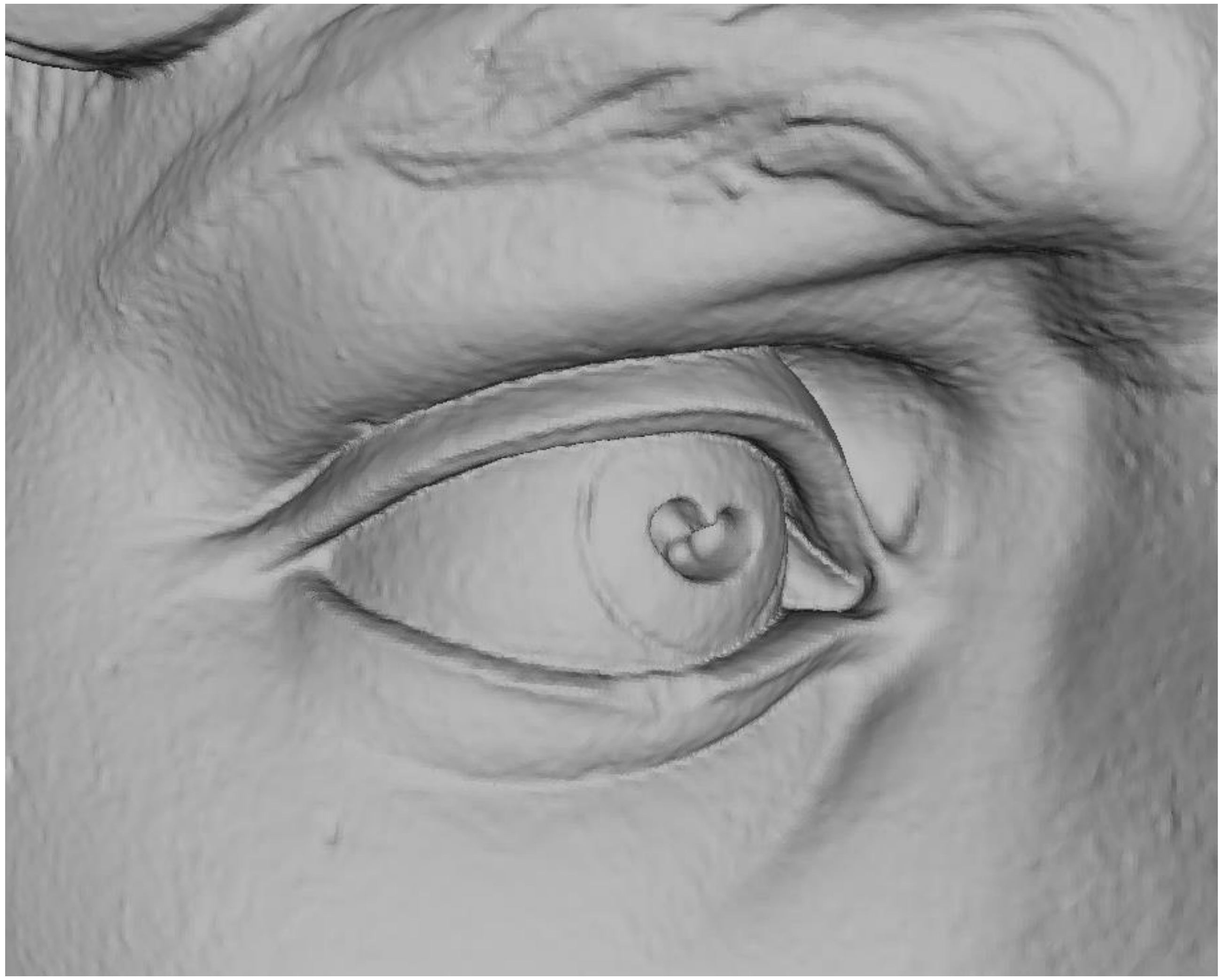
- Near the point data dominates the energy
  - Make the function approximate the signed distance function
- Away from the point data dominates the regularization energy
  - Tend to make the gradient vector field constant

Smooth Signed Distance Surface Reconstruction

[Calakli et al., PG 11]



POISSON



SSD

Screened Poisson Surface Reconstruction

[Kazhdan et al., TOG 13]

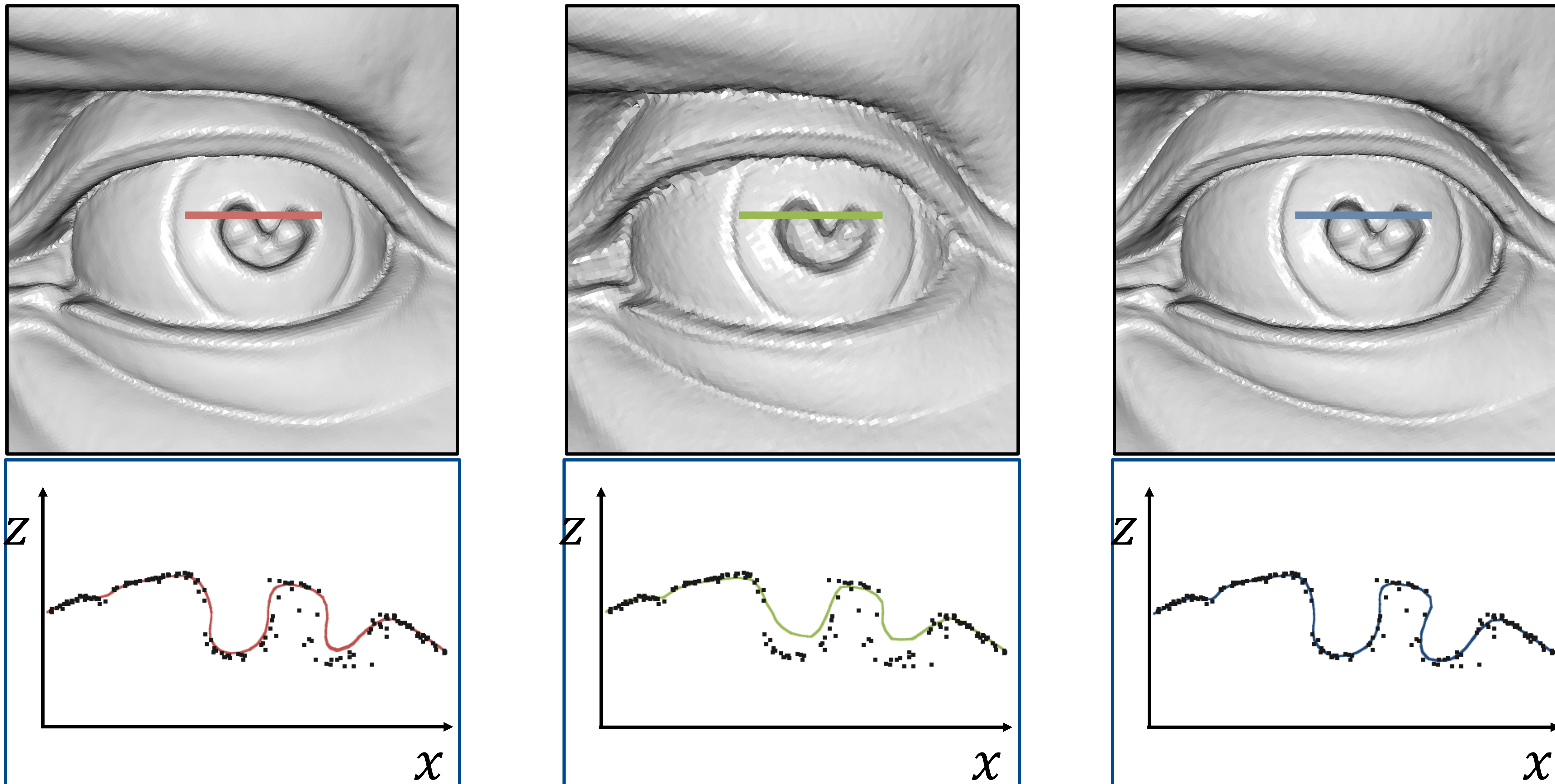
- Add discrete interpolation to the energy

$$E(\chi) = \int \|\nabla\chi(\mathbf{p}) - V(\mathbf{p})\|^2 d\mathbf{p} + \lambda \sum_{\mathbf{p}\in D} \chi^2(\mathbf{p})$$

- Encourage indicator function to be zero at samples

Screened Poisson Surface Reconstruction

[Kazhdan et al., TOG 13]

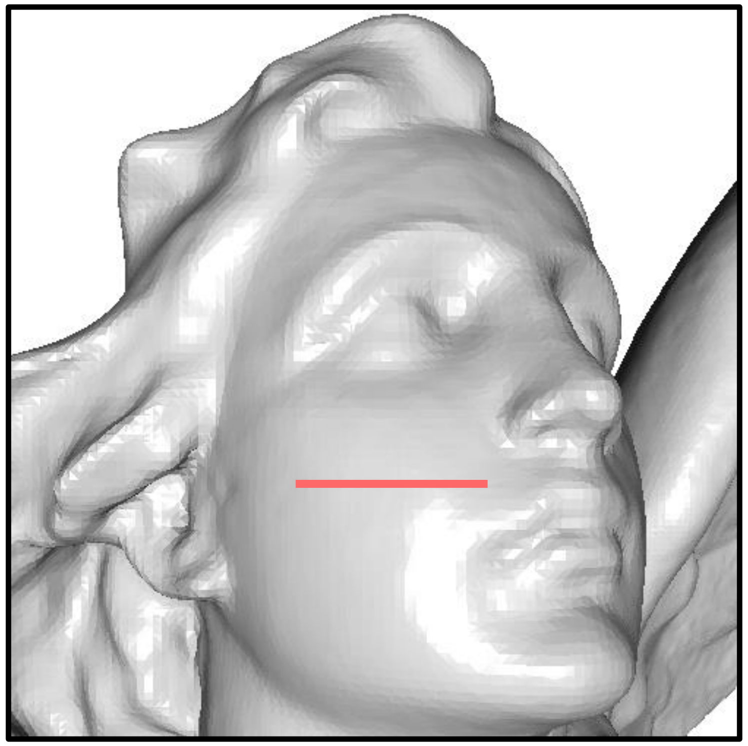


POISSON SSD SCREENED POISSON

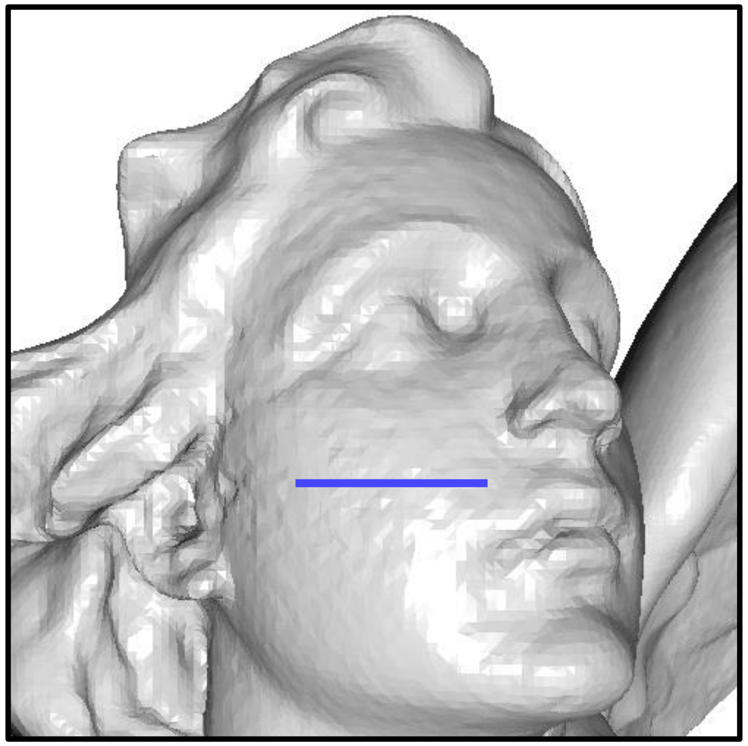
Screened Poisson Surface Reconstruction

[Kazhdan et al., TOG 13]

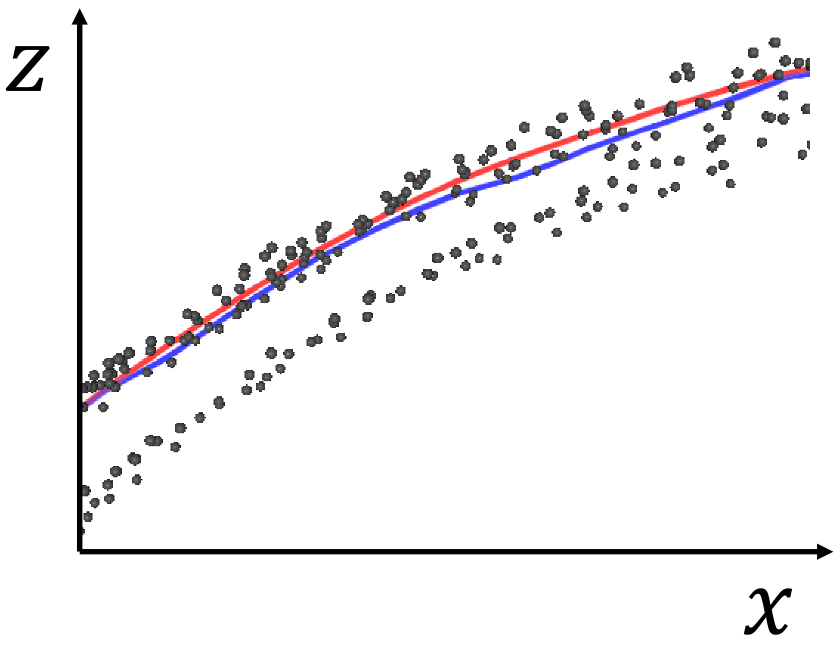
- Sharper reconstruction
- Fast method (linear solver)
- But it assumes clean data



POISSON



SCREENED POISSON



References

- Boissonnat, Jean-Daniel. "Geometric structures for three-dimensional shape representation." *ACM Transactions on Graphics (TOG)* 3.4 (1984): 266-286.
- Bernardini, Fausto, et al. "The ball-pivoting algorithm for surface reconstruction." *IEEE transactions on visualization and computer graphics* 5.4 (1999): 349-359.
- Turk, Greg, and Marc Levoy. "Zippered polygon meshes from range images." *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, 1994.
- Hoppe, Hugues, et al. *Surface reconstruction from unorganized points*. Vol. 26. No. 2. ACM, 1992.
- Carr, Jonathan C., et al. "Reconstruction and representation of 3D objects with radial basis functions." *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001.
- Curless, Brian, and Marc Levoy. "A volumetric method for building complex models from range images." *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996.
- Alexa, Marc et al. “Point set surfaces”. In: *Proc. of IEEE Visualization 01*.
- Alexa, Marc, and Anders Adamson. "On Normals and Projection Operators for Surfaces Defined by Point Sets." *SPBG* 4 (2004): 149-155.
- Kazhdan, Michael, Matthew Bolitho, and Hugues Hoppe. "Poisson surface reconstruction." *Proceedings of the fourth Eurographics symposium on Geometry processing*. Eurographics Association, 2006.
- Calakli, Fatih, and Gabriel Taubin. "SSD: Smooth signed distance surface reconstruction." *Computer Graphics Forum*. Vol. 30. No. 7. Blackwell Publishing Ltd, 2011.
- Kazhdan, Michael, and Hugues Hoppe. "Screened poisson surface reconstruction." *ACM Transactions on Graphics (TOG)* 32.3 (2013): 29.