4 - Surface Reconstruction

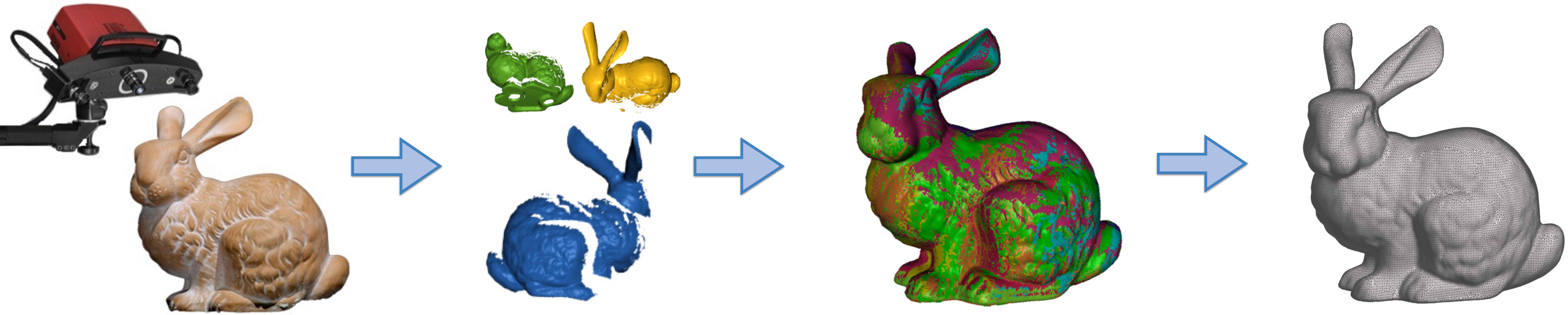
Xifeng Gao

Contents and Goals

- Contents:
  - Definition and variants of the surface reconstruction problem
  - Implicit surface reconstruction pipeline
- Goals:
  - Understand the implicit surface reconstruction pipeline
  - Be able to implement the 3D version of the signed distance function (SDF) based surface reconstruction in three weeks

3D Surface Representation

3D Surface Reconstruction



Physical Object

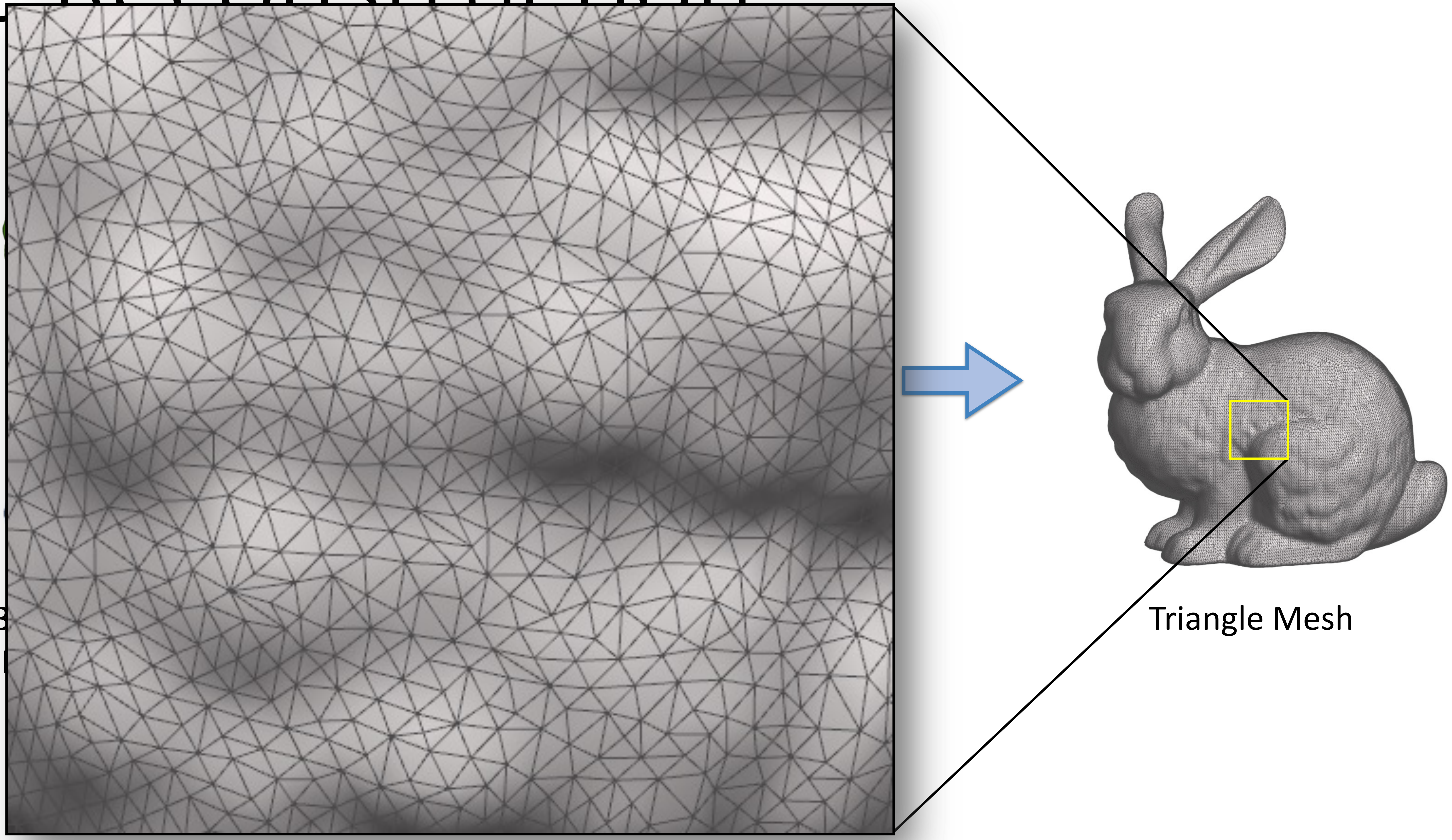
3D Points,
Range Images

Registered Point Set

Triangle Mesh

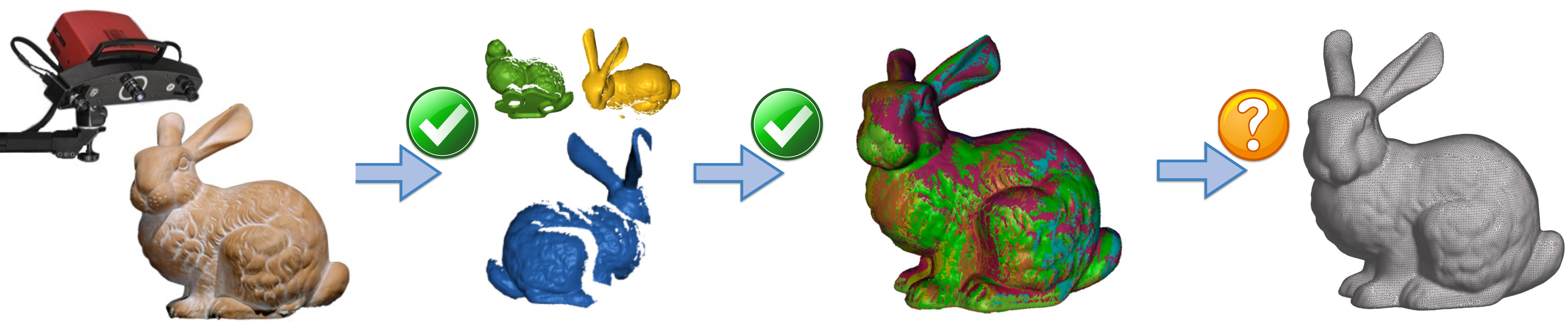
3D Surface Reconstruction

Physical Object



Triangle Mesh

3D Surface Reconstruction



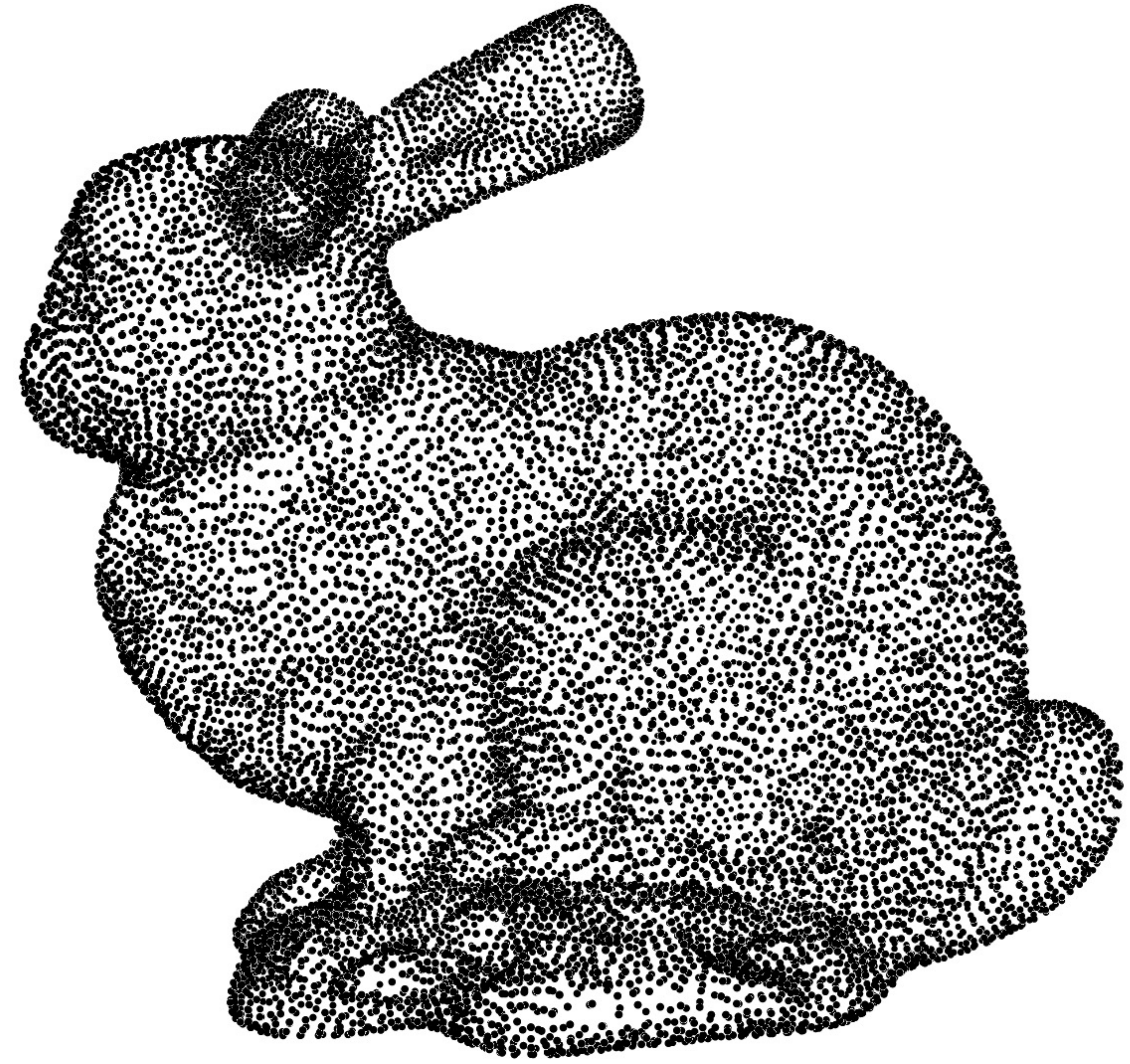
Physical Object

3D Points,
Range Images

Registered Point Set

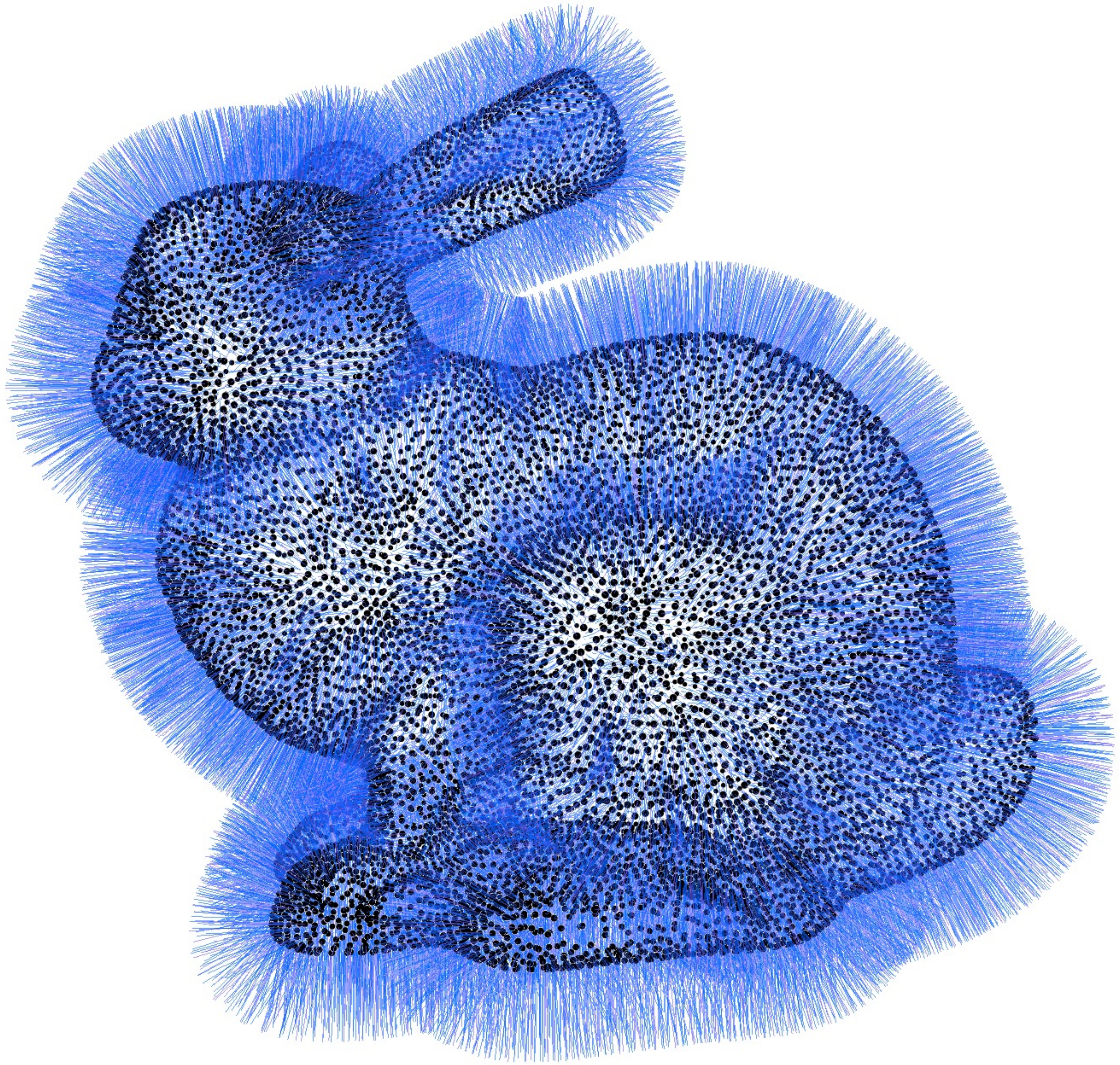
Triangle Mesh

Input



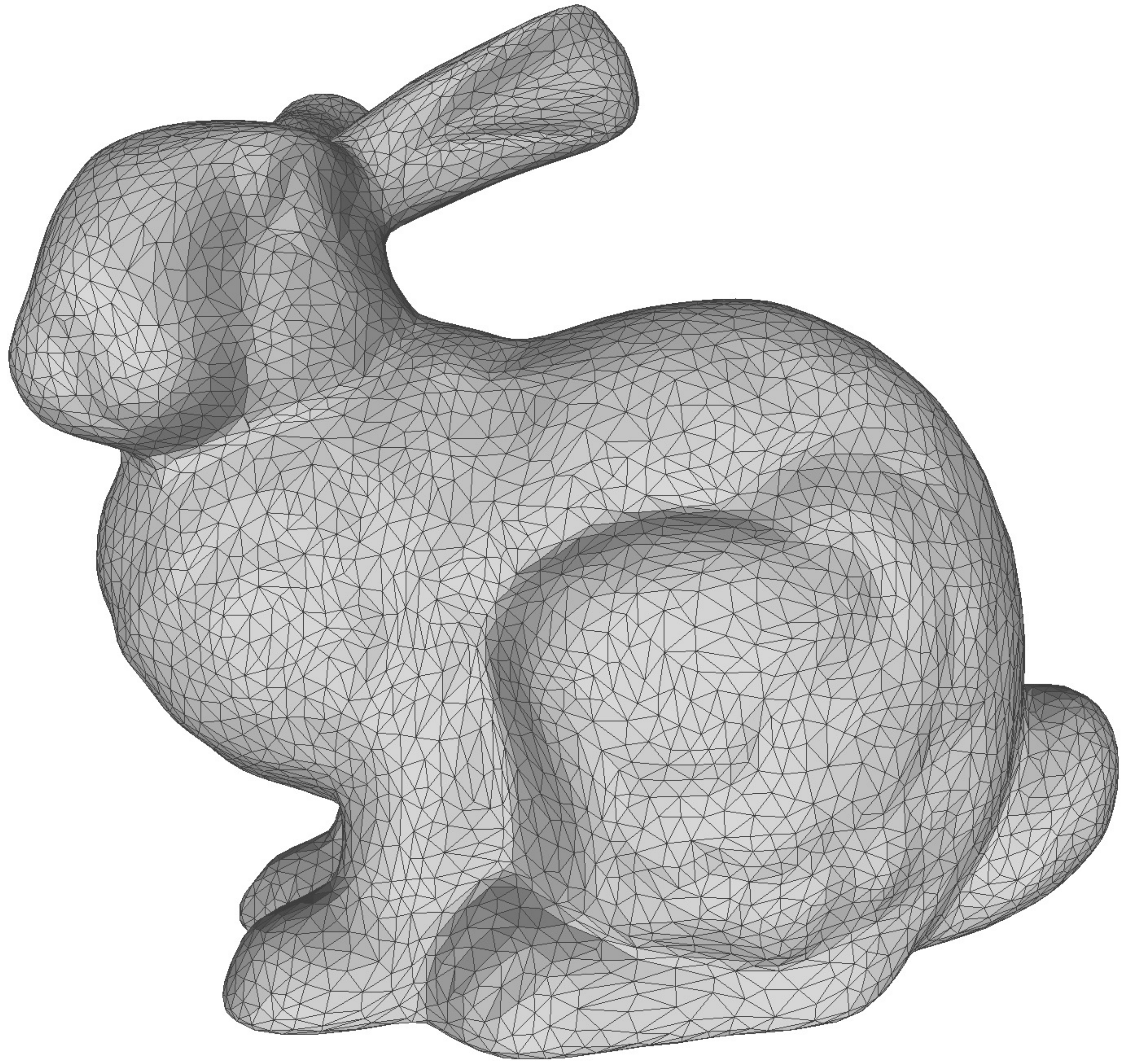
3D Point Set

Input



3D Point Set + Normal

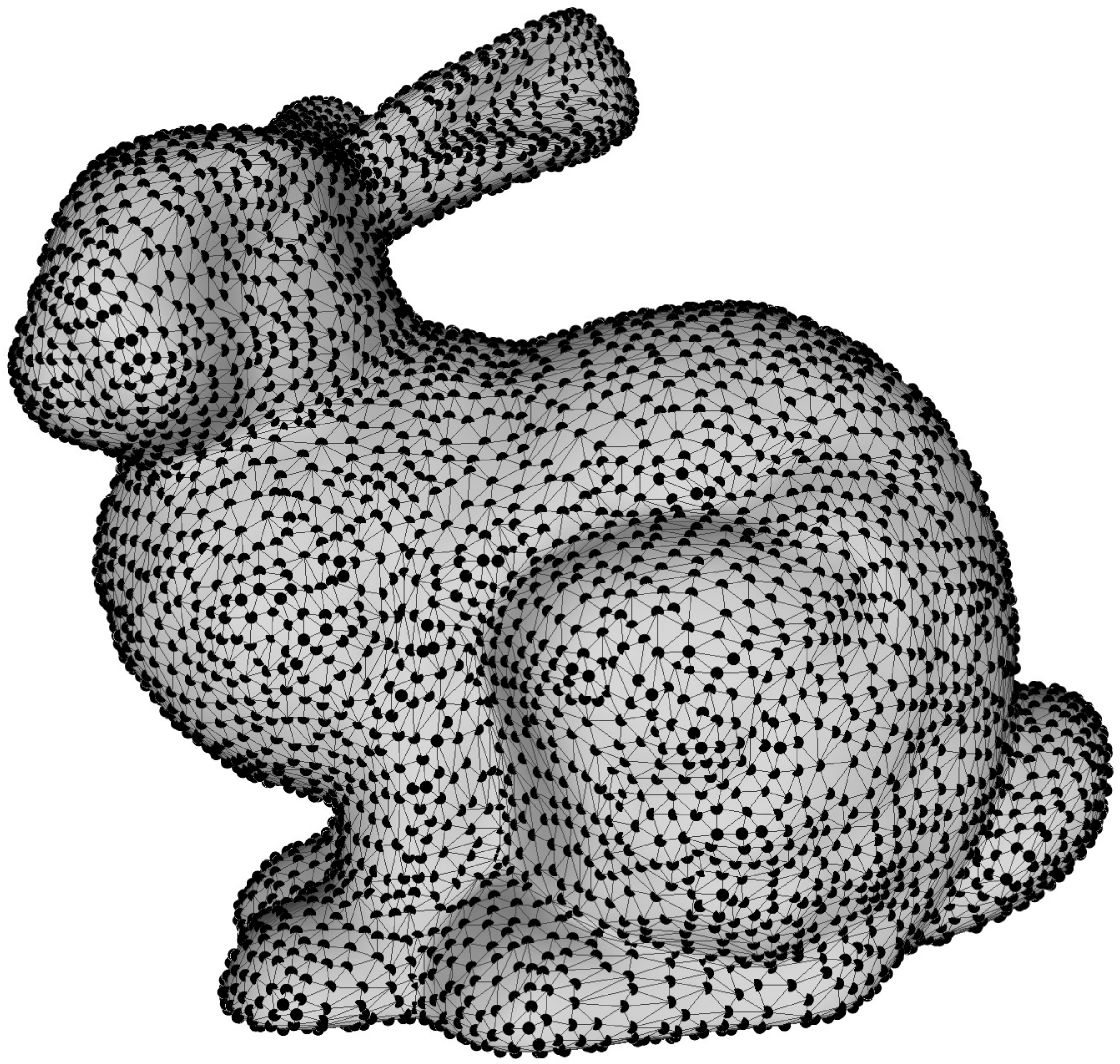
Output



3D Triangle Mesh

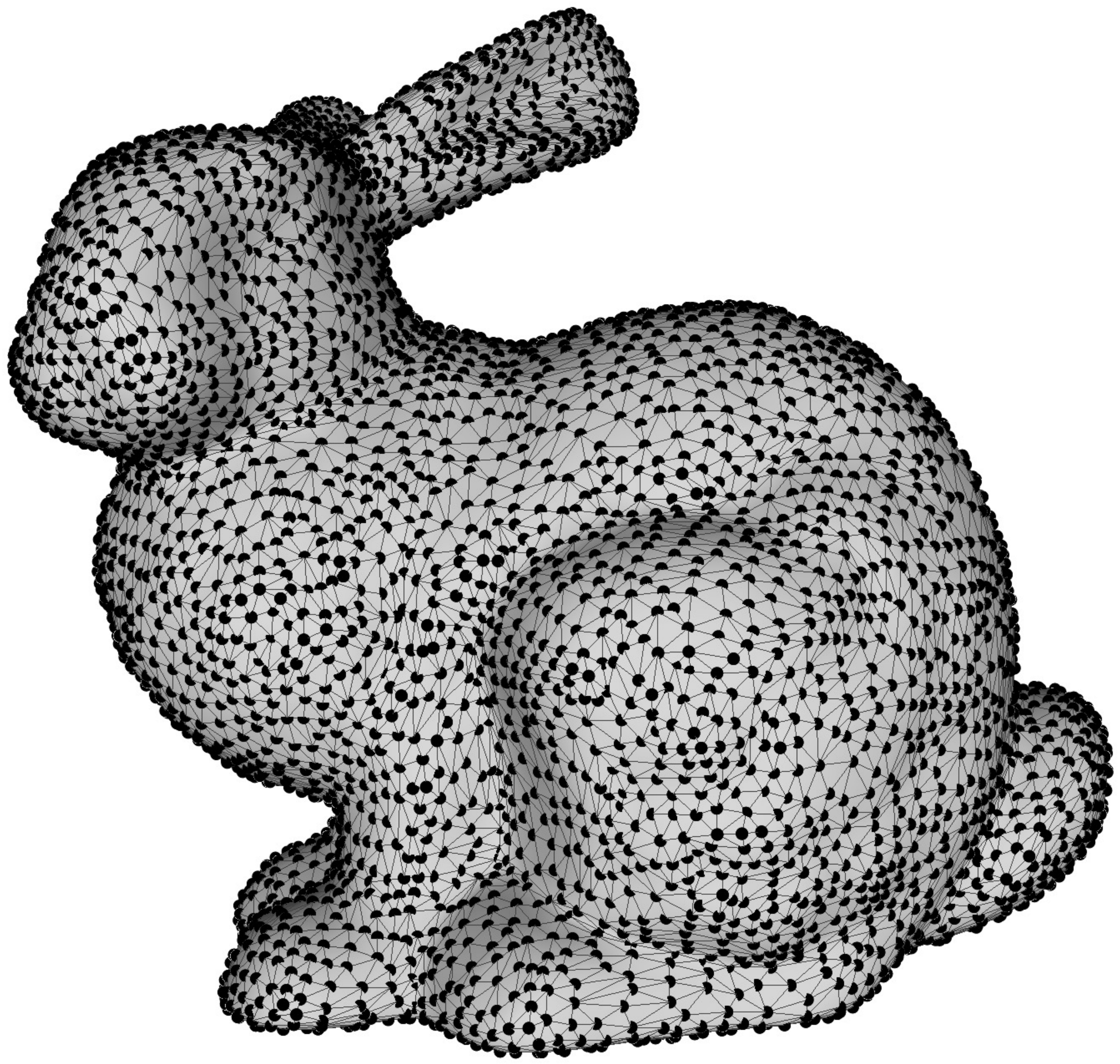
Problem Assumptions

Problem Assumptions



Interpolation

Problem Assumptions

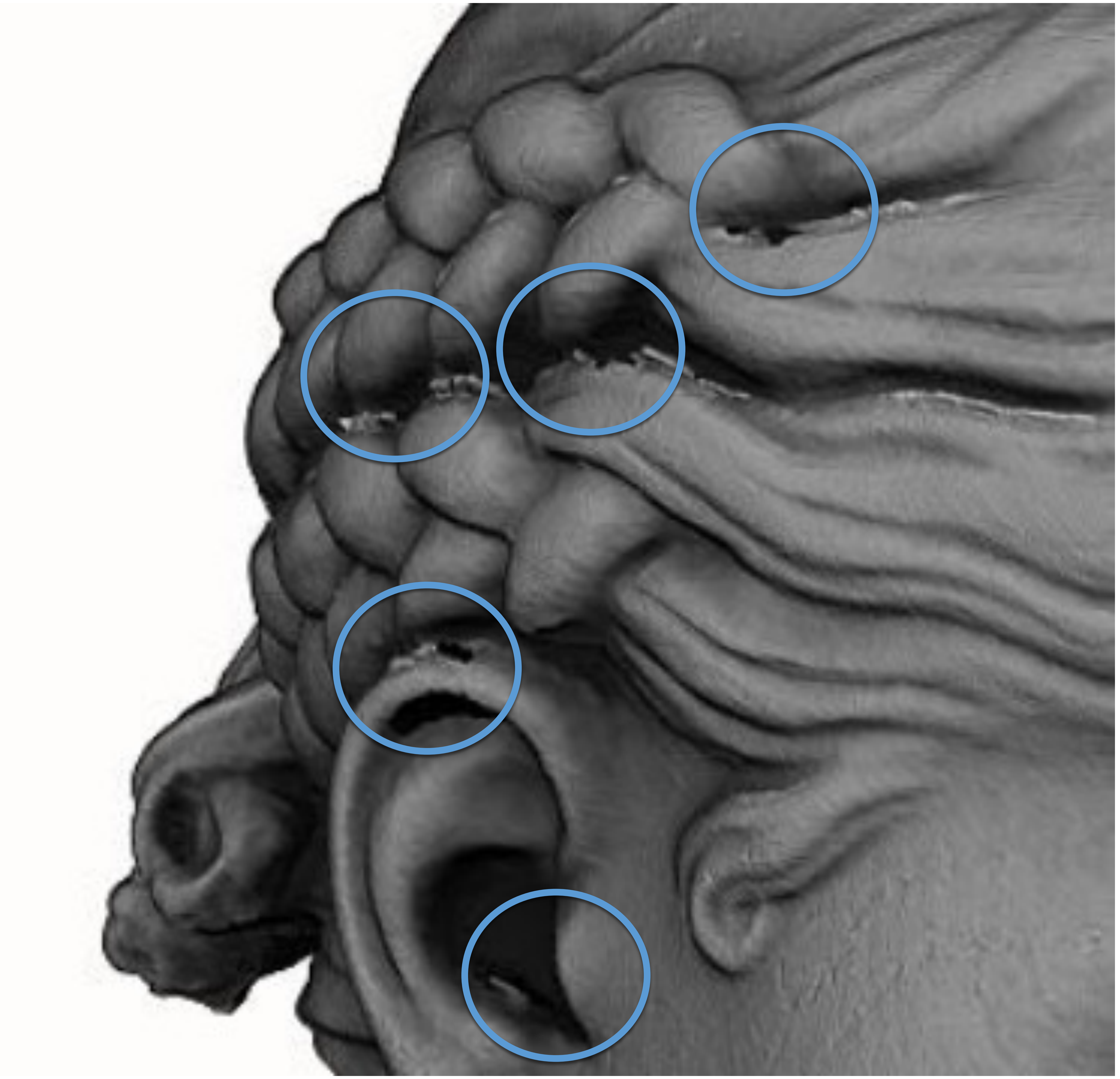


Interpolation

Approximation

Explicit Reconstruction

- Interpolate input data
- Surface is explicitly represented by piecewise parametric functions
- Often assume input data is perfect, e.g. dense sampling, noise free, etc.

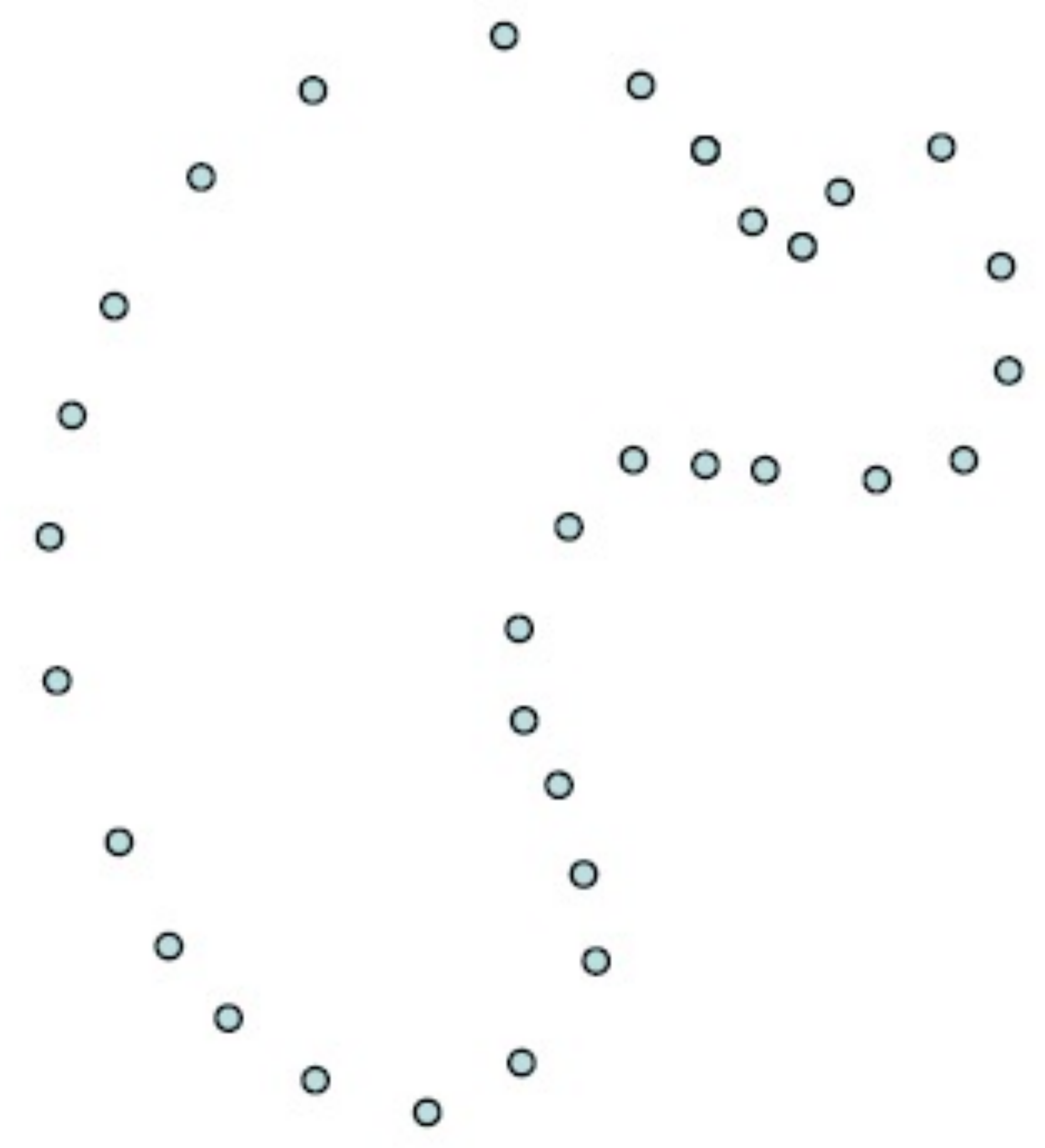


Implicit Reconstruction

- Approximate input data
- Surface is defined to be the 0-level set of a scalar-valued function $F(x)$

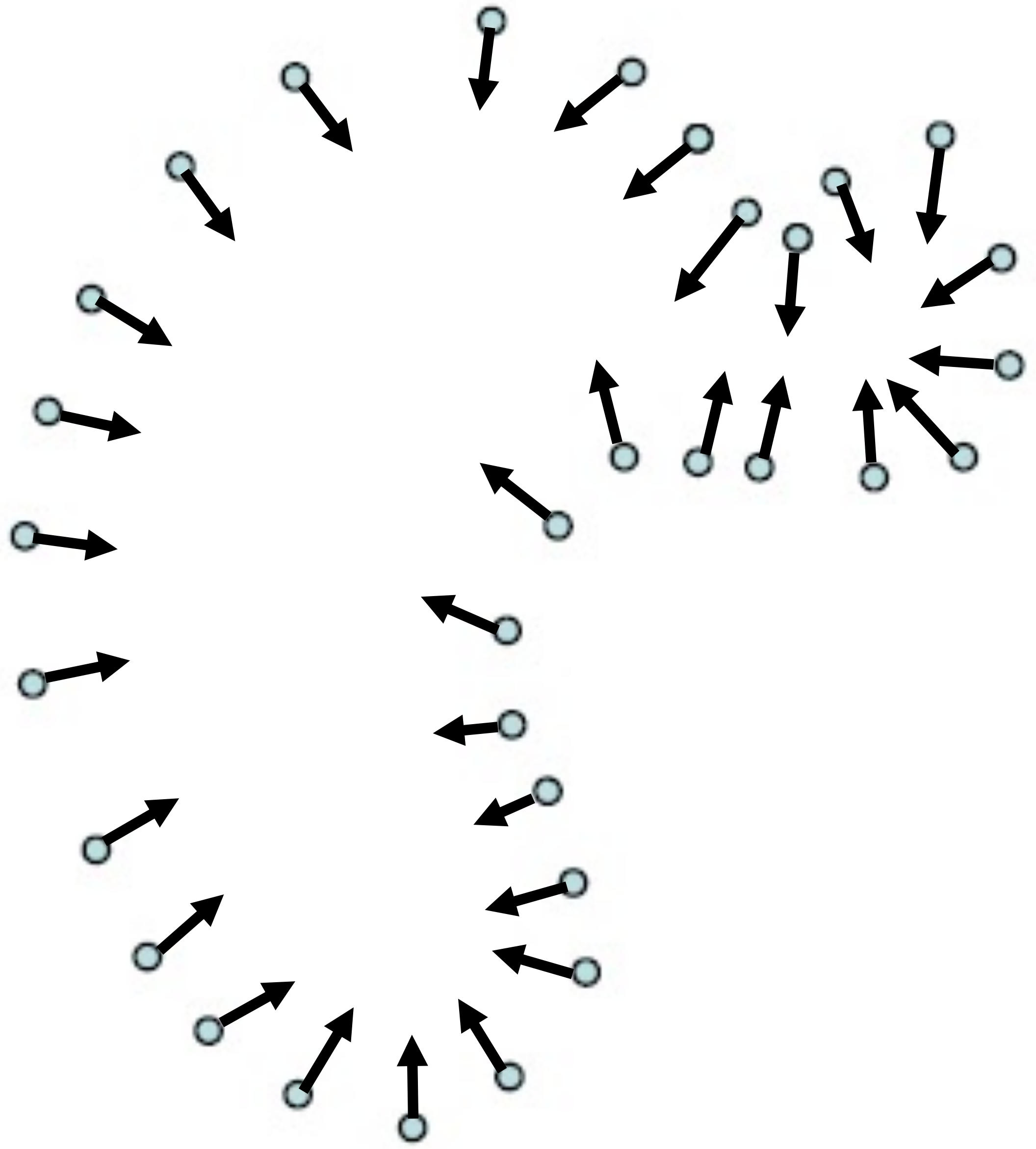
Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
3. Discretize function $F(x)$
4. Extract zero Iso-surface



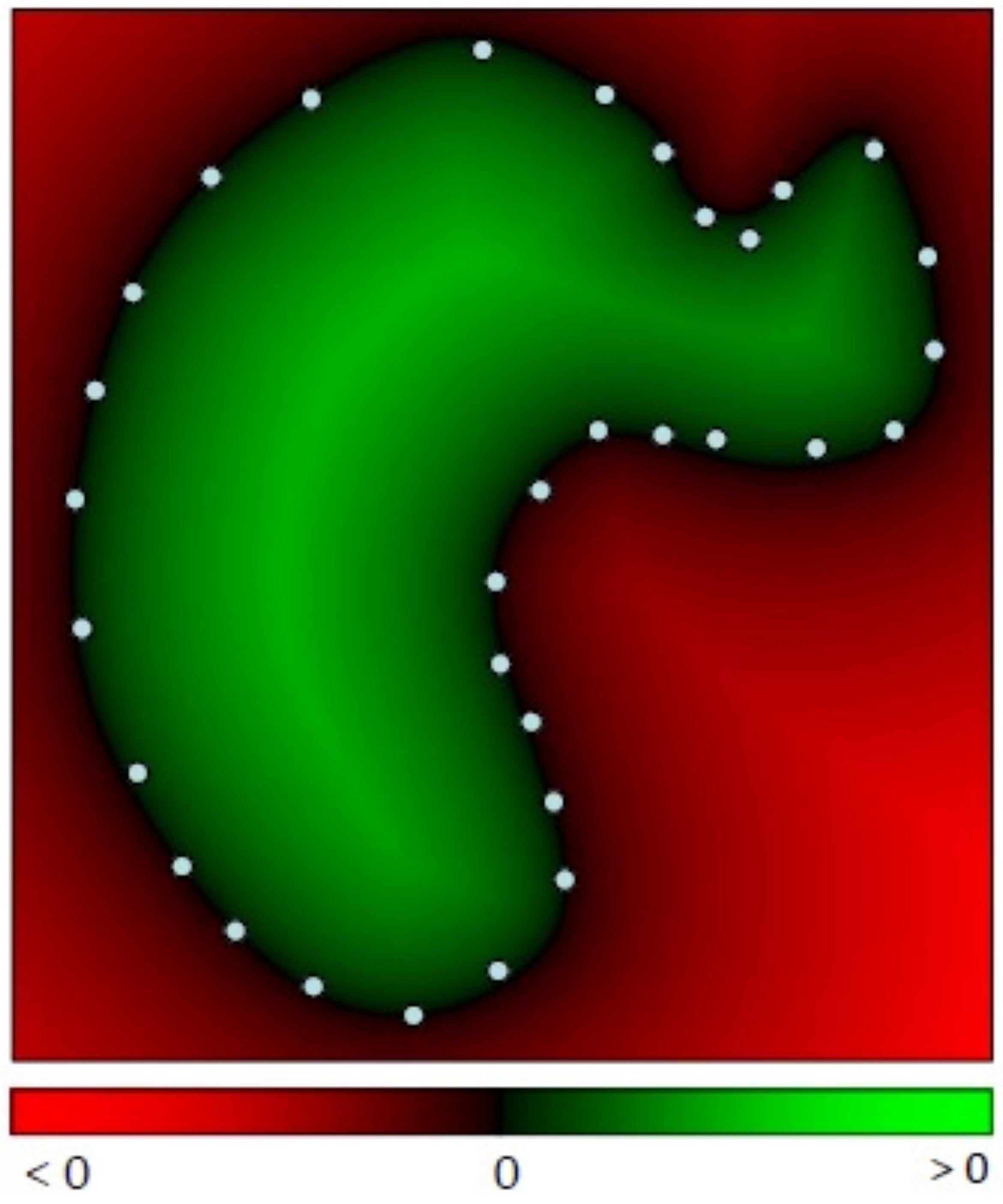
Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
3. Discretize function $F(x)$
4. Extract zero Iso-surface



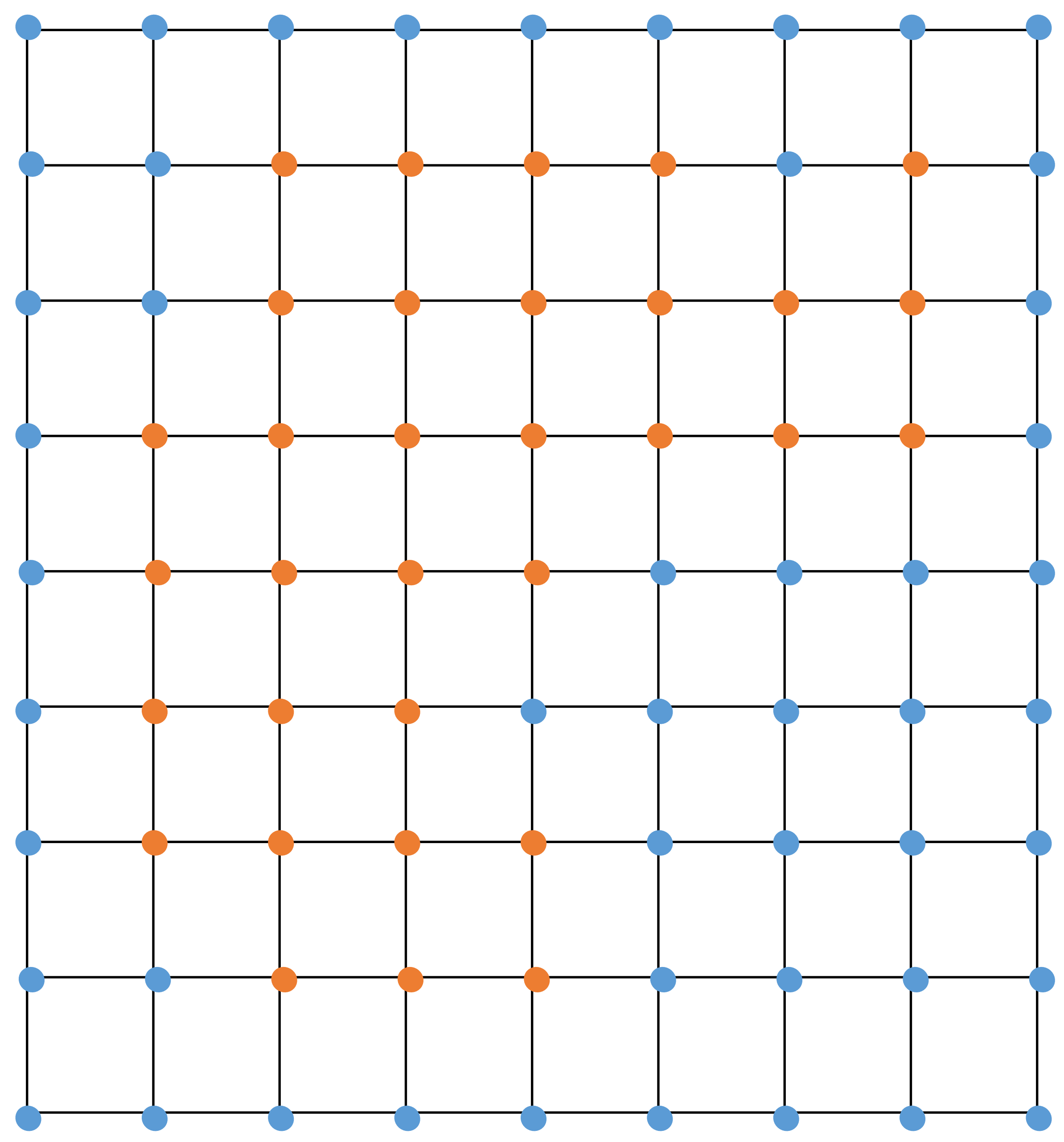
Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
3. Discretize function $F(x)$
4. Extract zero Iso-surface



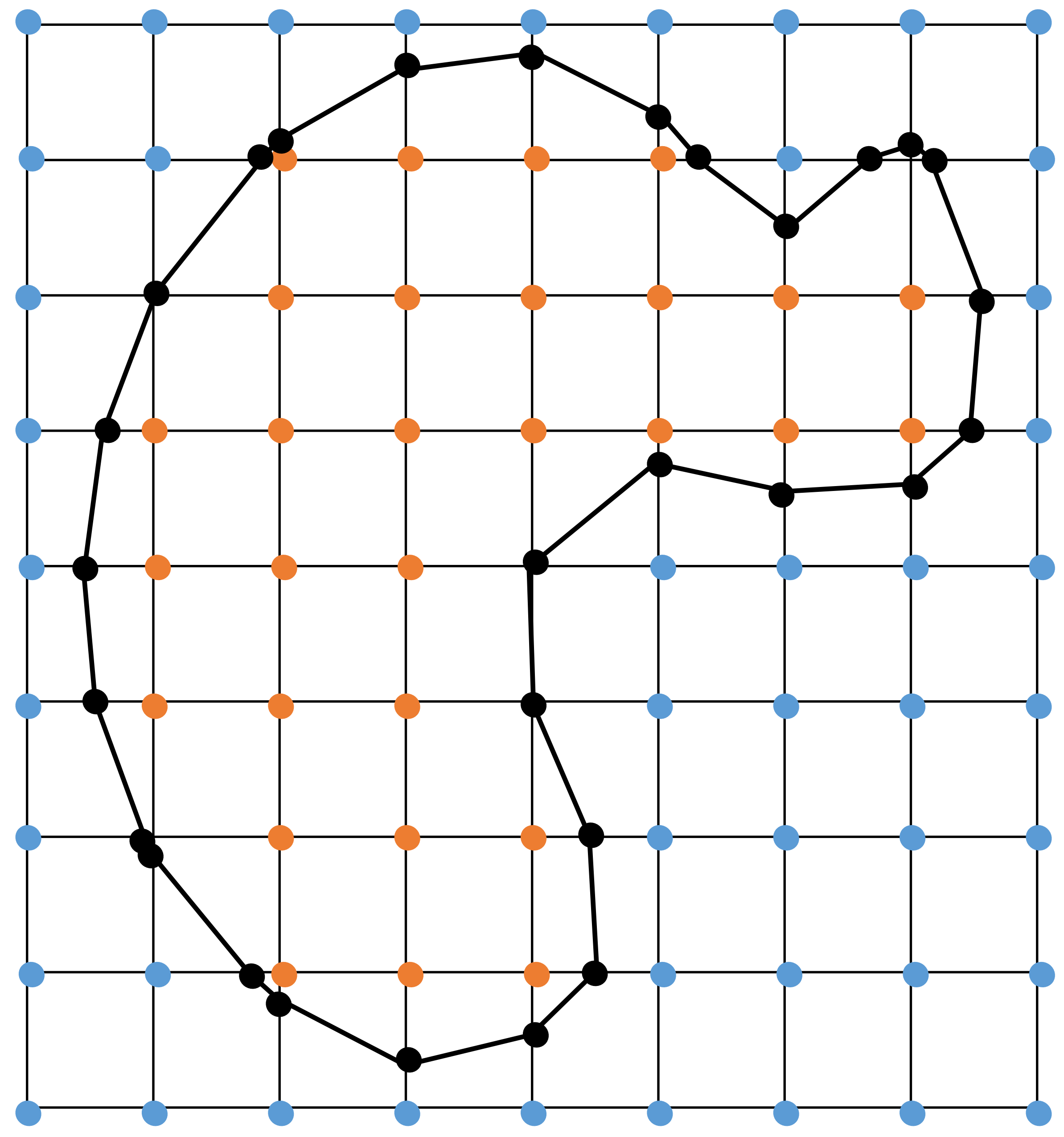
Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
3. **Discretize function $F(x)$**
4. Extract zero Iso-surface



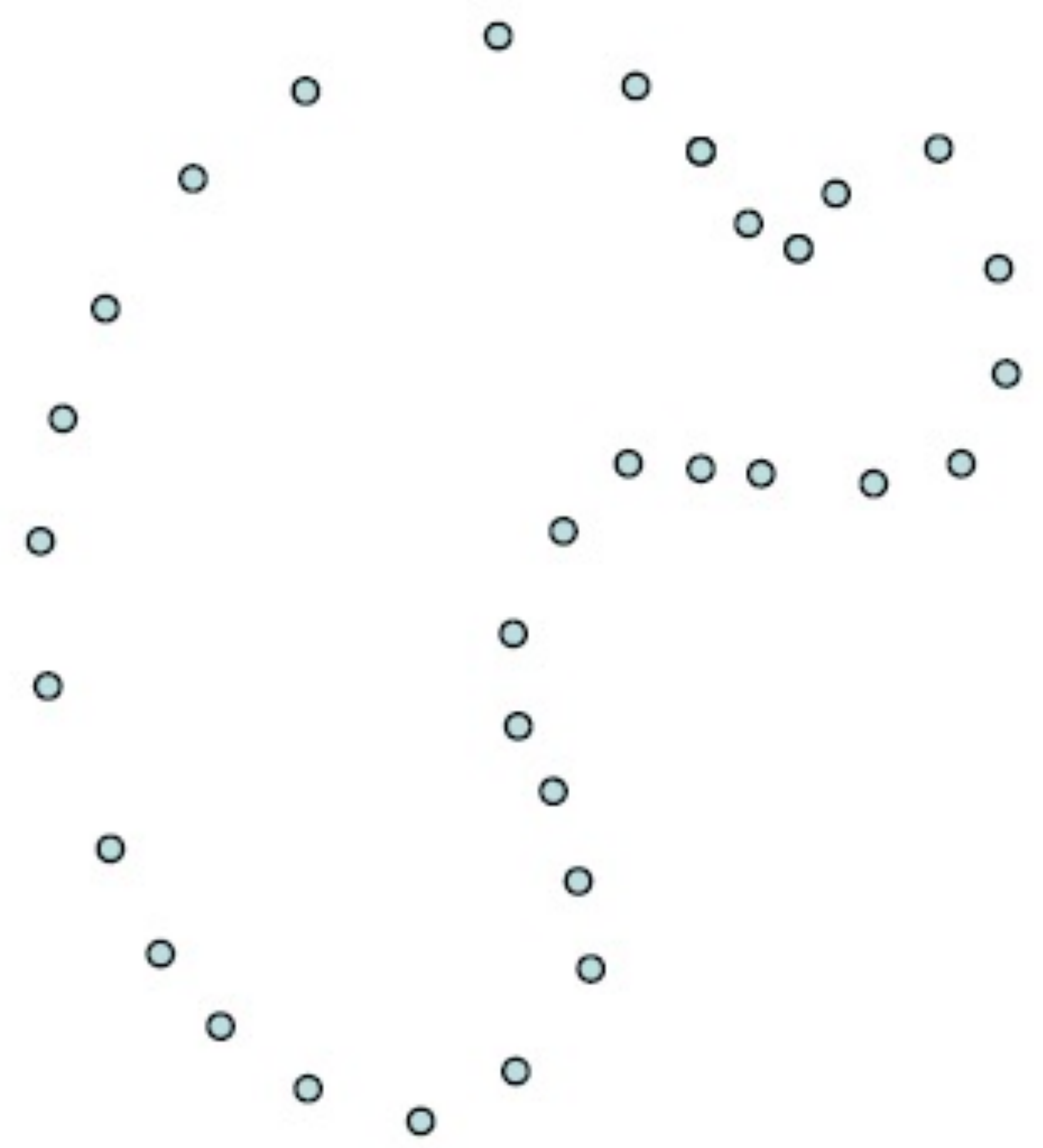
Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
3. Discretize function $F(x)$
4. Extract zero Iso-surface



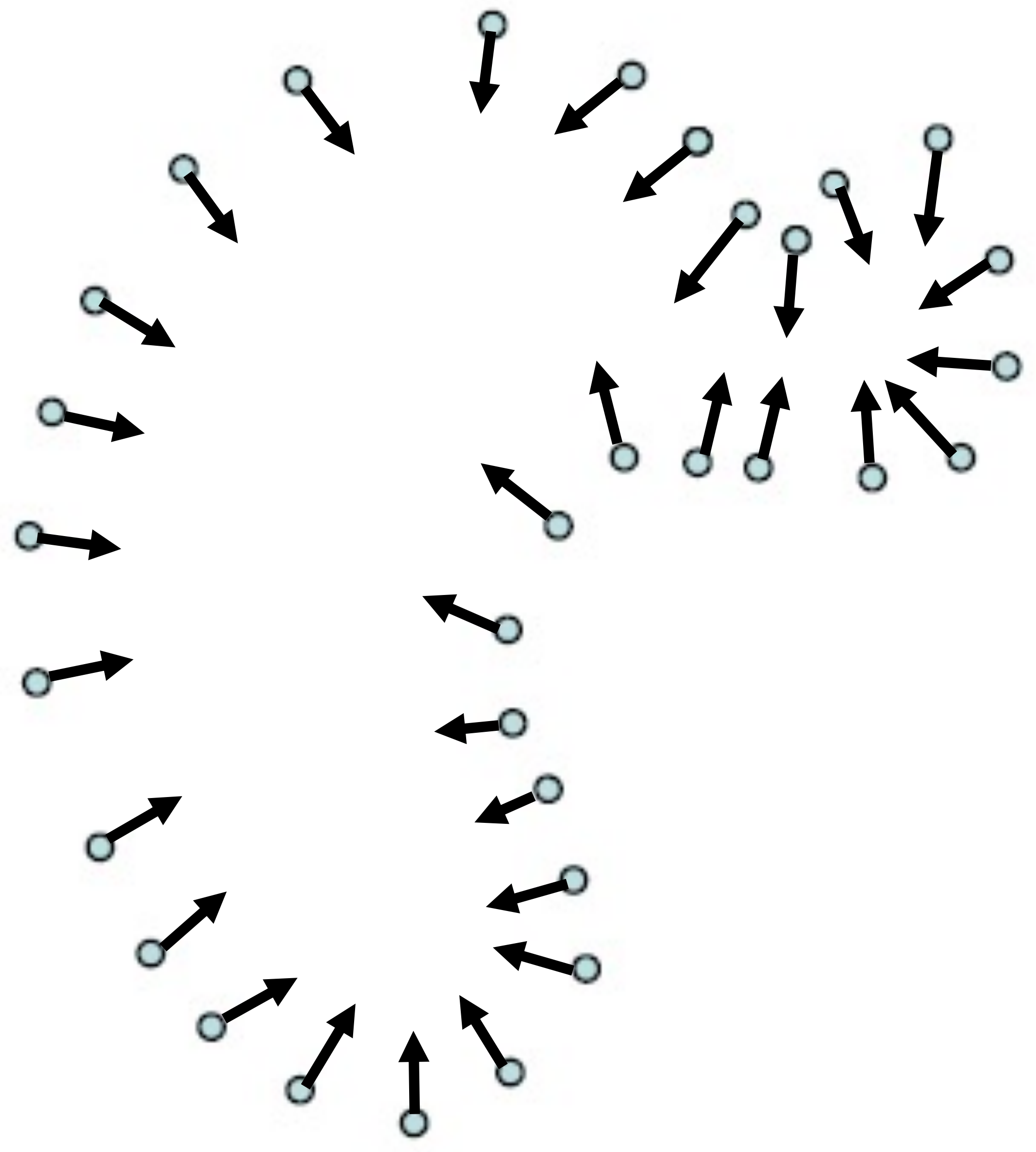
Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
3. Discretize function $F(x)$
4. Extract zero Iso-surface



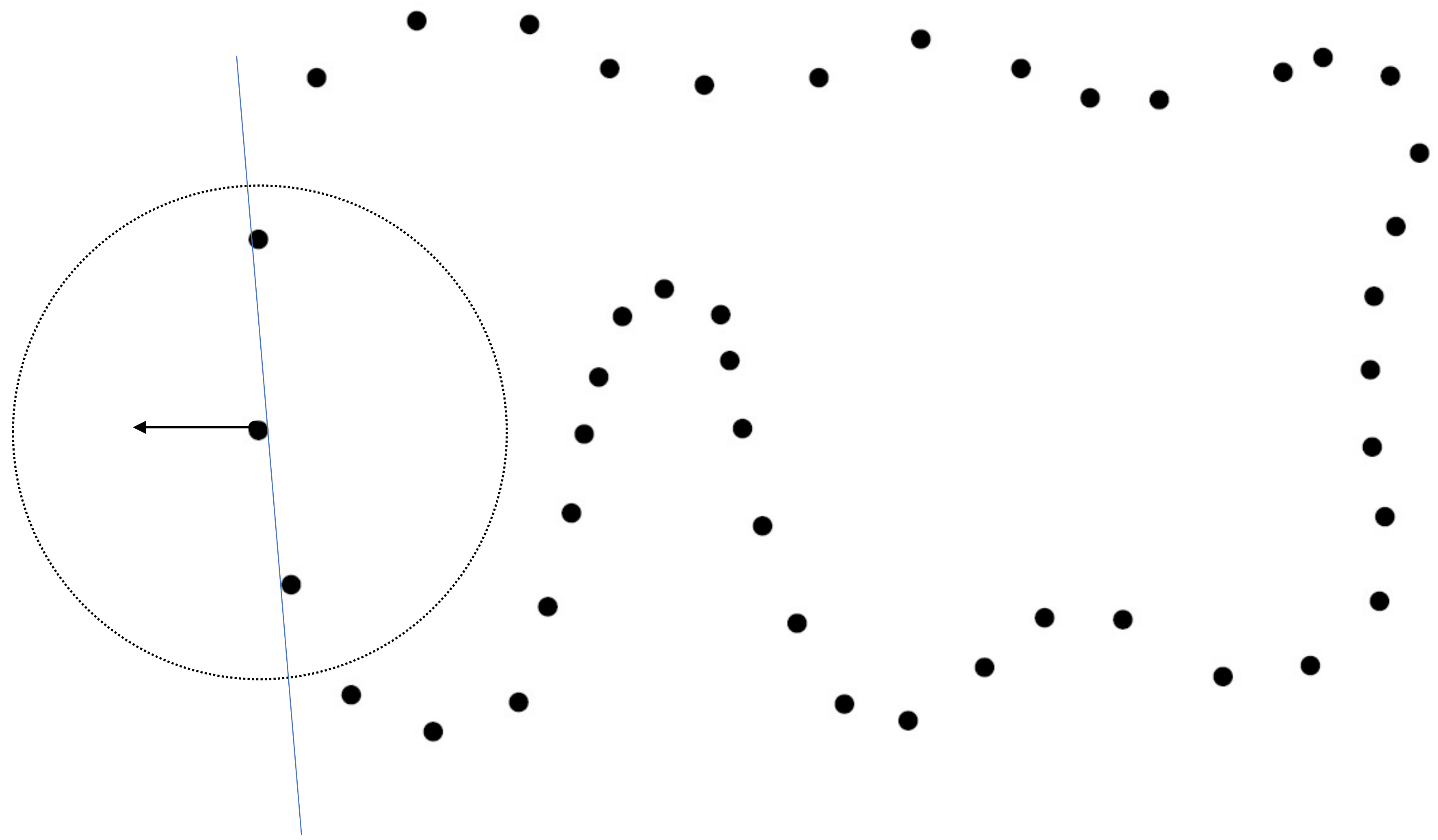
Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
3. Discretize function $F(x)$
4. Extract zero Iso-surface



Estimate normal

Step 1: Fit a line to the neighbors of each point.



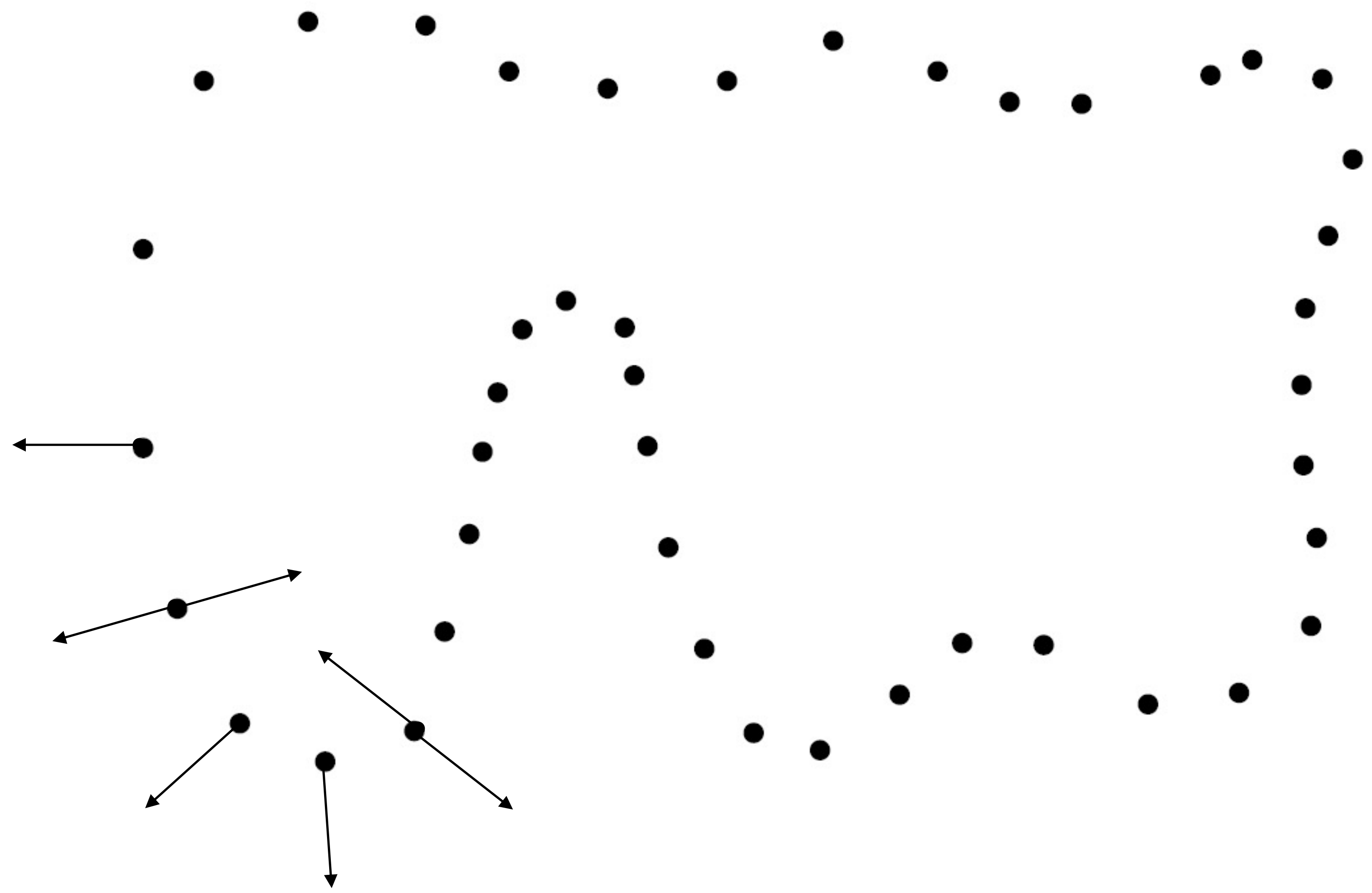
Surface reconstruction from unorganized points, Hoppe et al., ACM SIGGRAPH 1992

Estimate normal

Step 1: Fit a line to the neighbors of each point.

This doesn't guarantee a consistent orientation!

For the orientation to be consistent, neighboring points should point in similar directions.

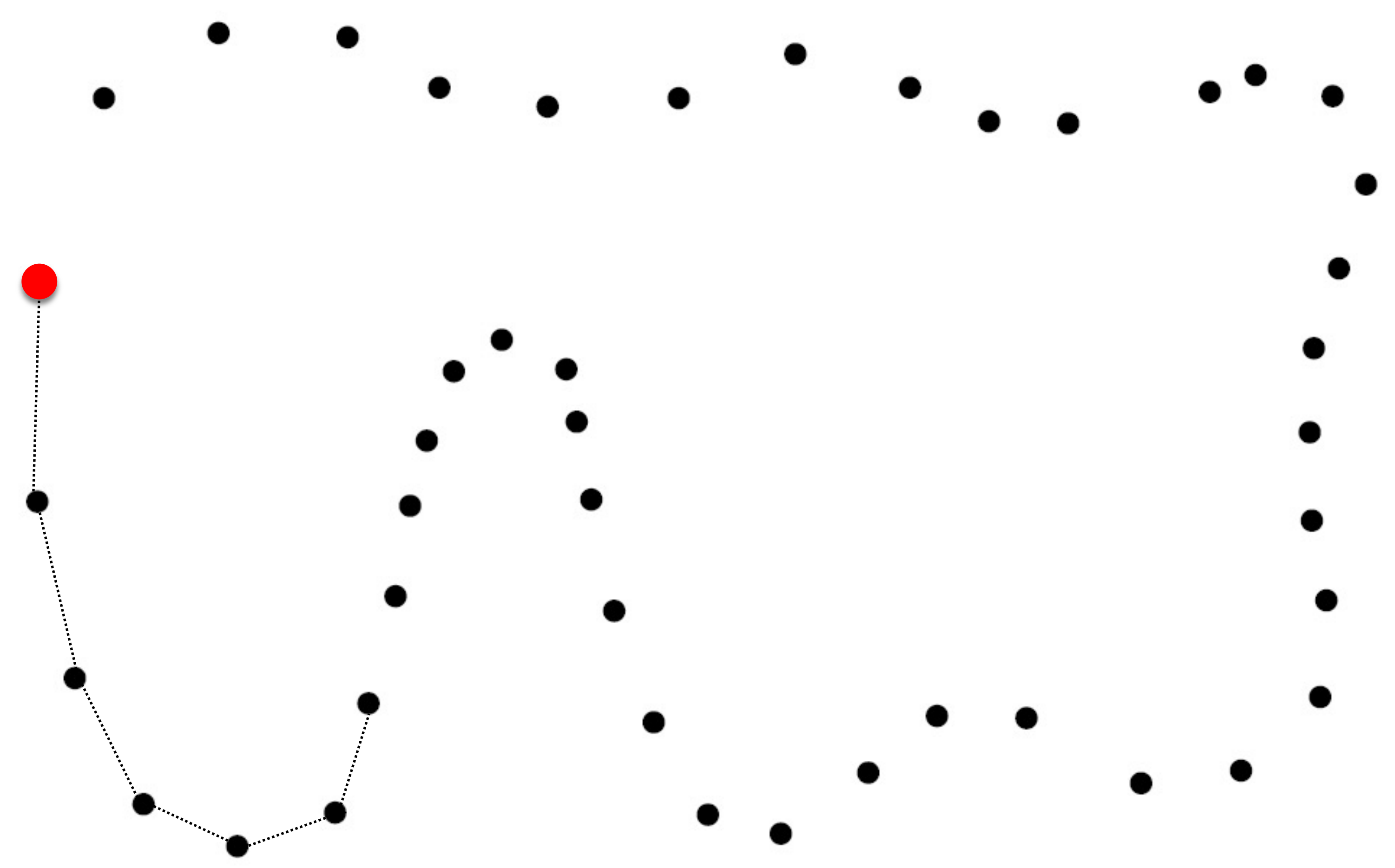


Surface reconstruction from unorganized points, Hoppe et al., ACM SIGGRAPH 1992

Estimate normal

Step 1: Fit a line to the neighbors of each point.

Step 2: Build a (Euclidian) minimal spanning tree and propagate the orientation from a root.

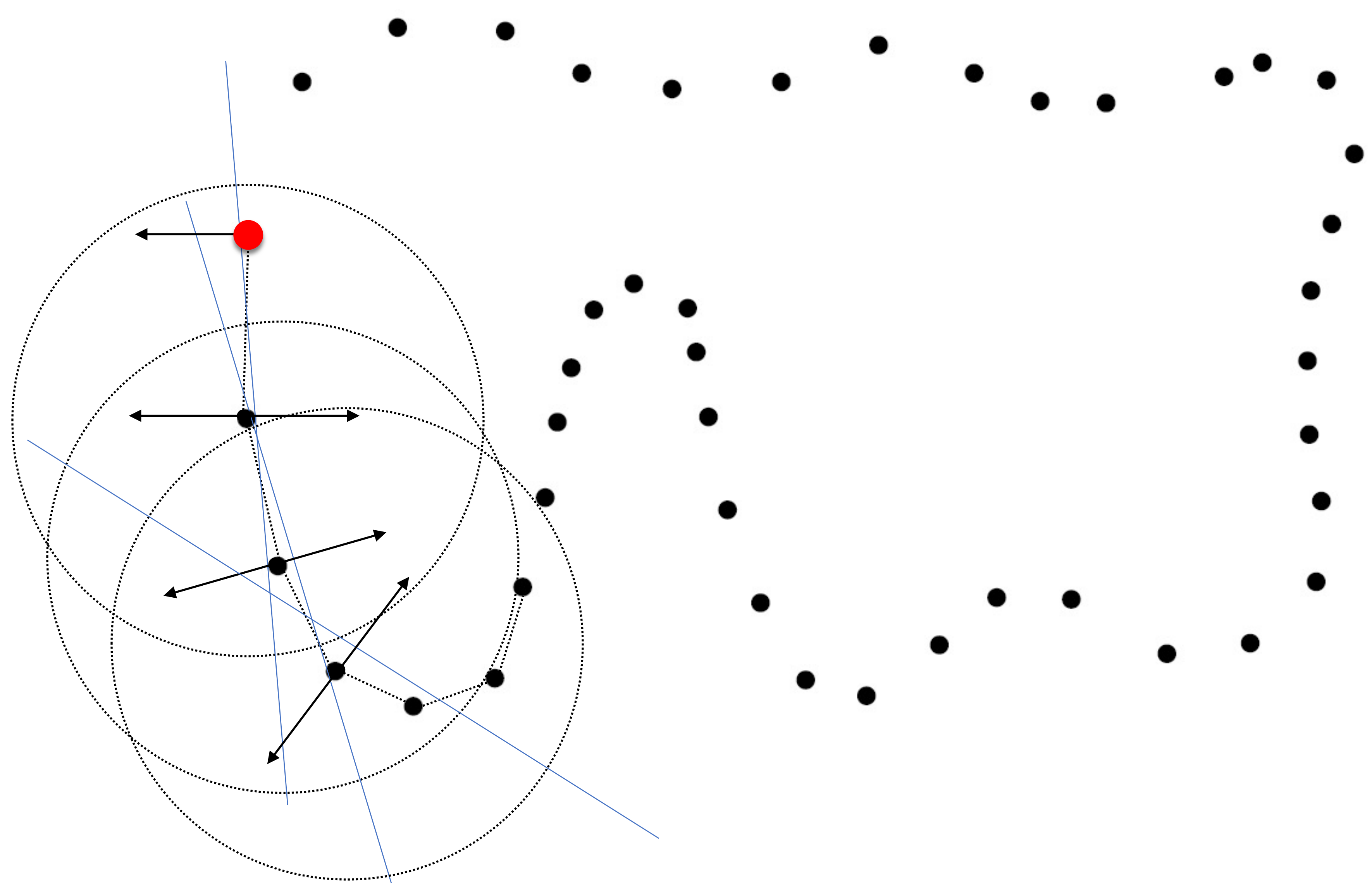


Surface reconstruction from unorganized points, Hoppe et al., ACM SIGGRAPH 1992

Estimate normal

Step 1: Fit a line to the neighbors of each point.

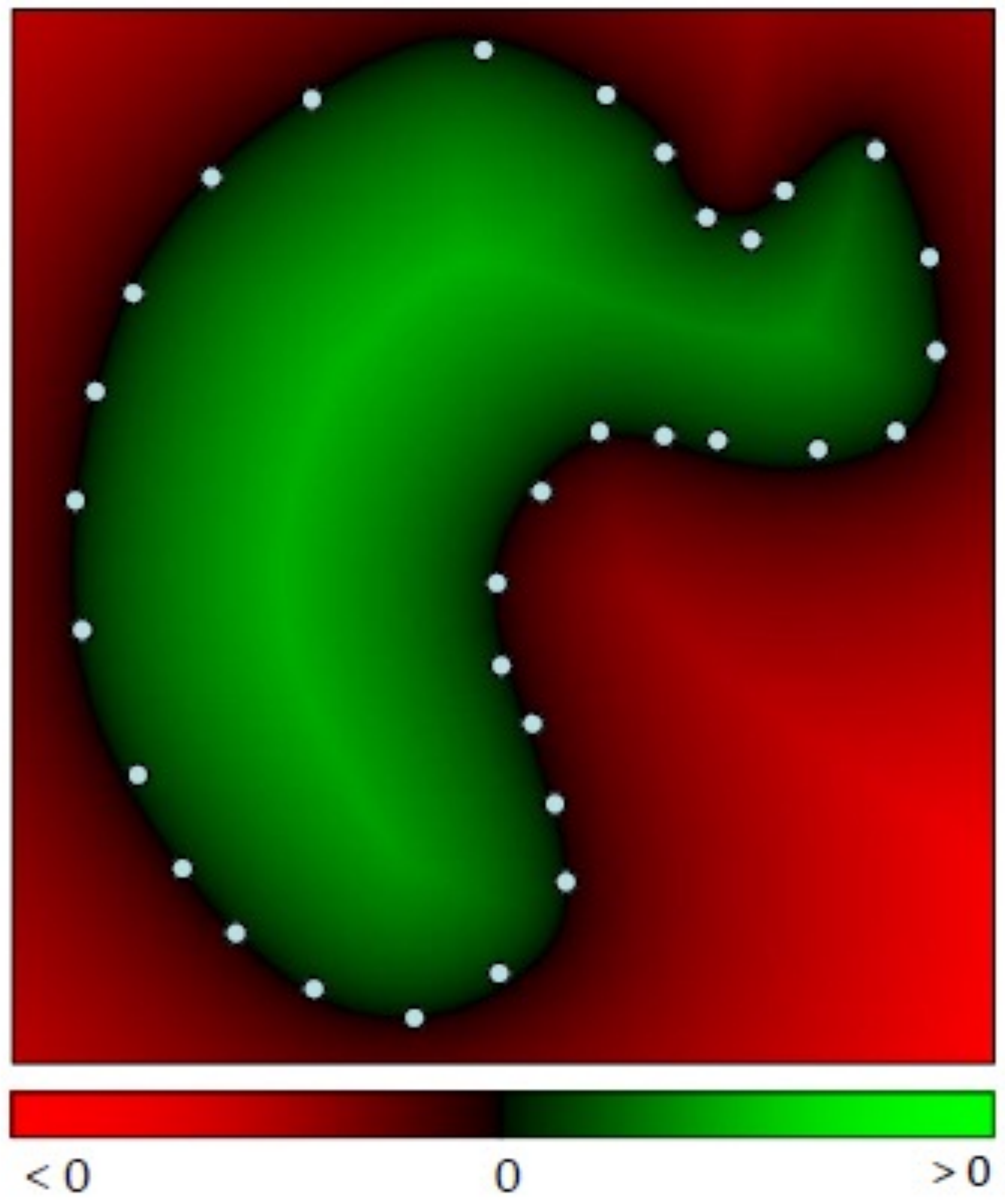
Step 2: Build a (Euclidian) minimal spanning tree and propagate the orientation from a root.



Surface reconstruction from unorganized points, Hoppe et al., ACM SIGGRAPH 1992

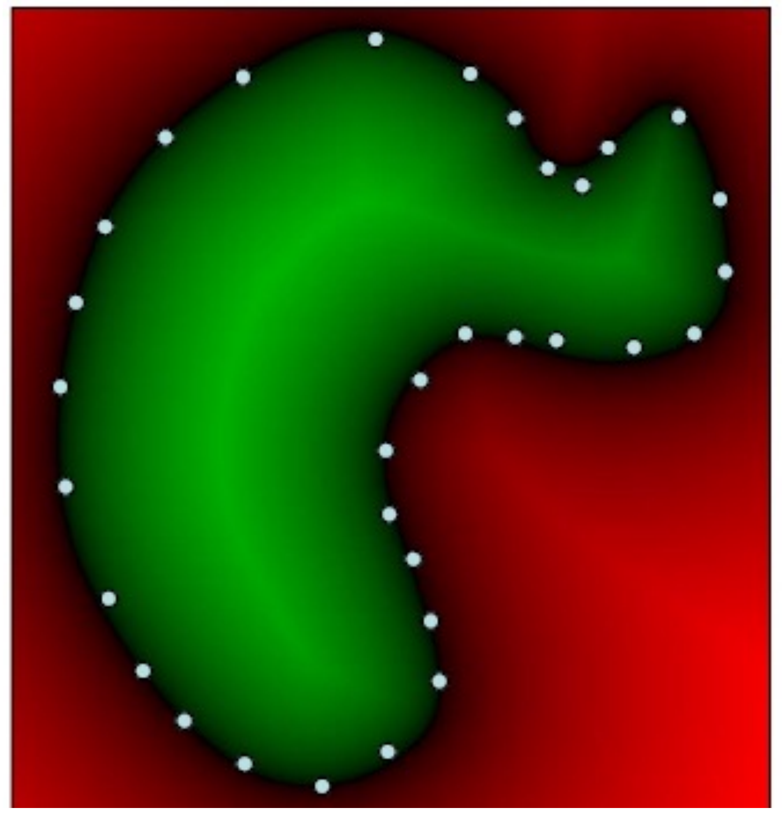
Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
3. Discretize function $F(x)$
4. Extract zero Iso-surface

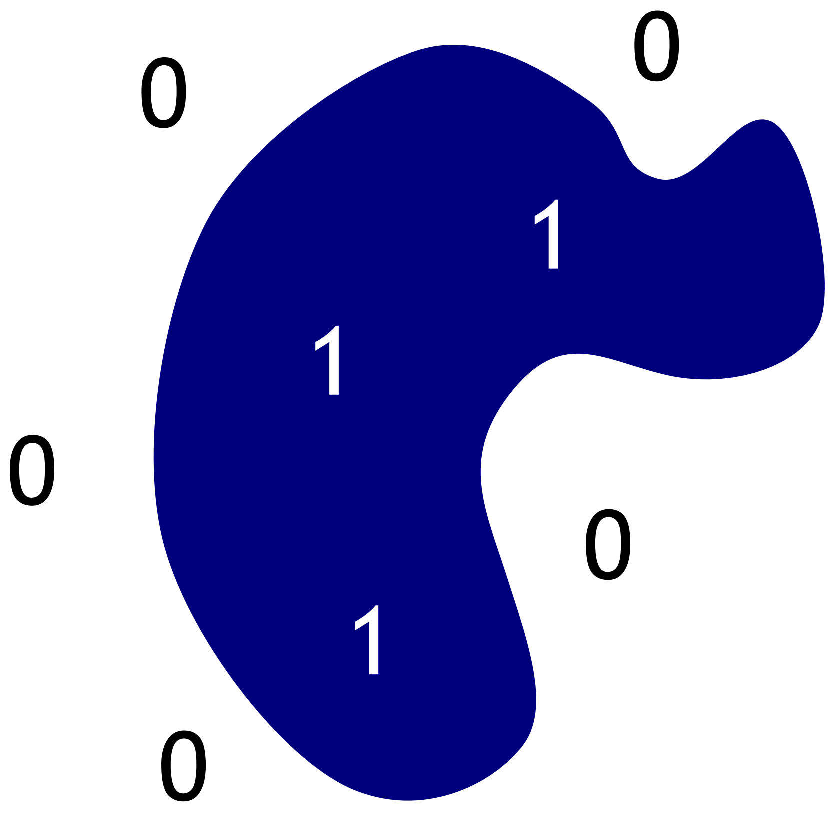


Implicit Reconstruction

1. Estimate normal (Optional)
2. **Compute function $F(x)$**

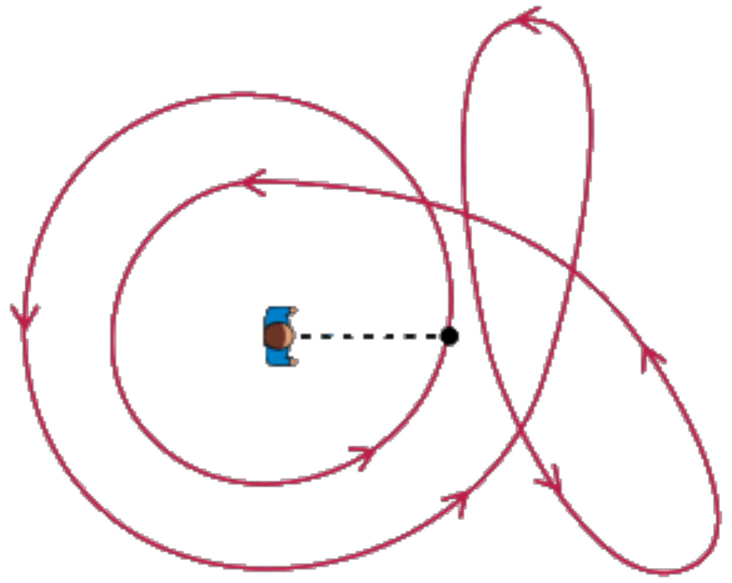


Signed distance function



Indicator function

Ray casting count



winding number

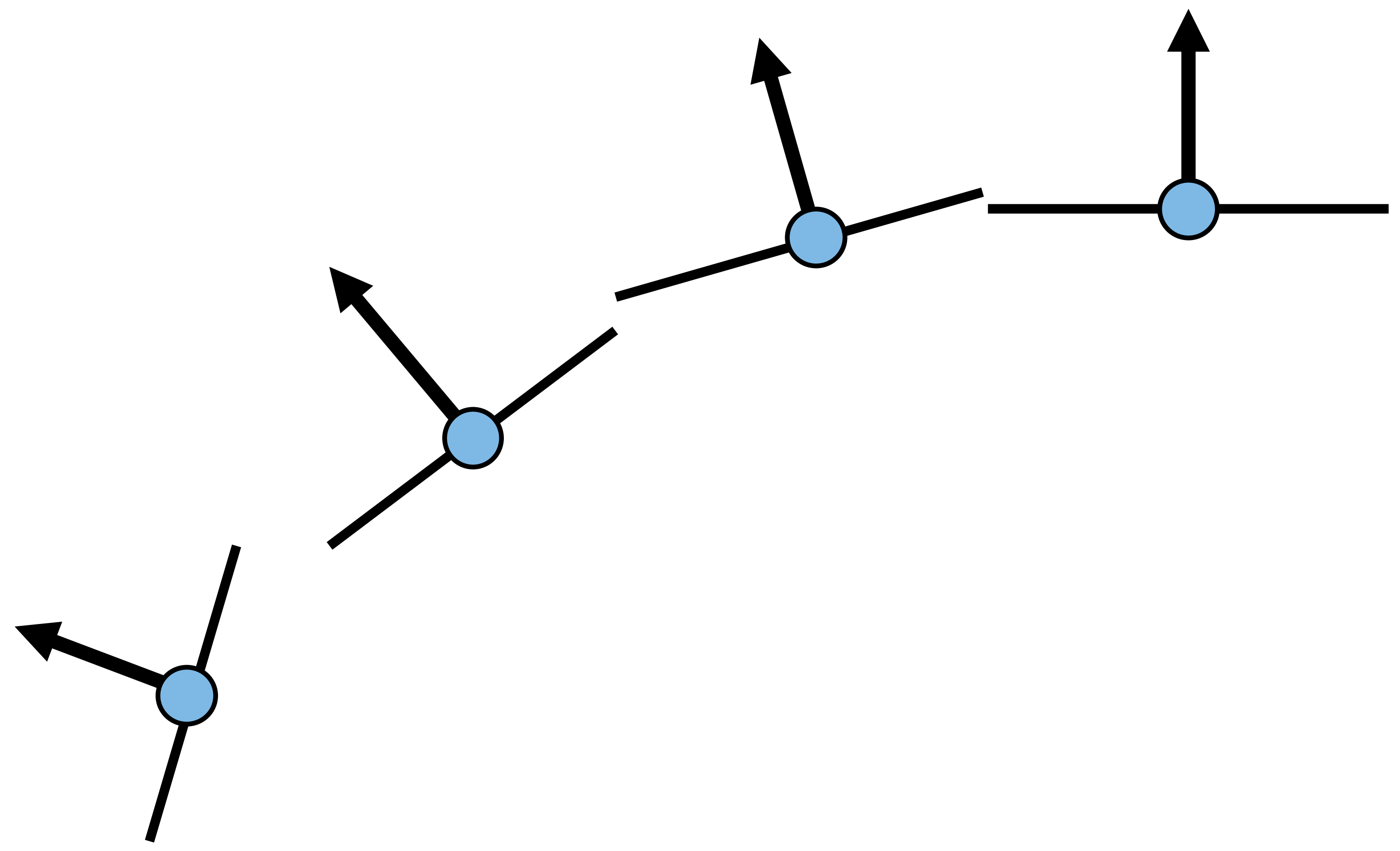
Continuous intensity function

Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
    1. **Sign Distance Function: distance from points and tangent planes, [Hoppe et al., SIGGRAPH 1992]**
    2. Sign Distance Function: radial basis function (RBF), [Carr et al., SIGGRAPH 2001]

SDF from Points and Normals

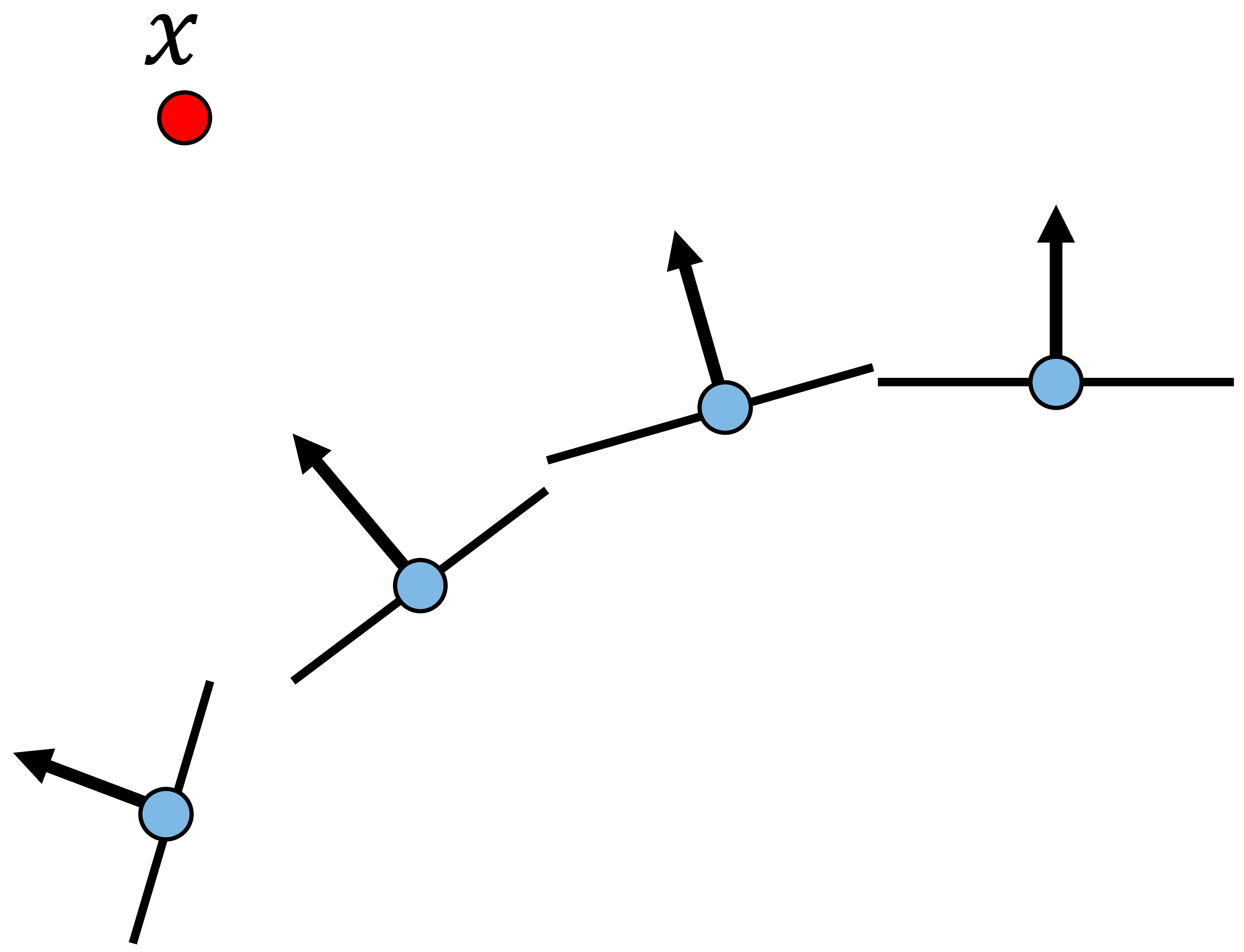
- Compute signed distance to the tangent plane of the closest point



Surface reconstruction from unorganized points, Hoppe et al., ACM SIGGRAPH 1992

SDF from Points and Normals

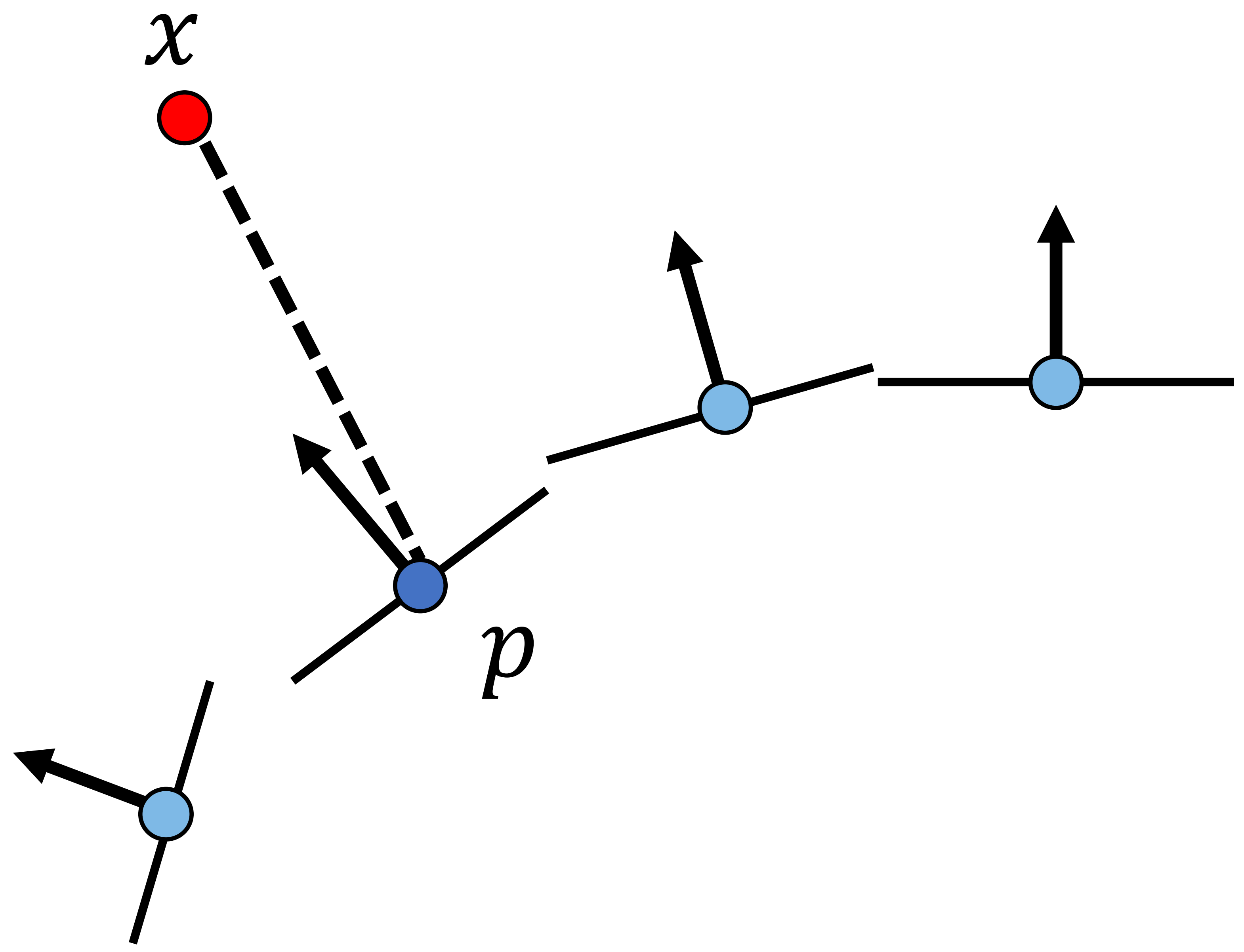
- Compute signed distance to the tangent plane of the closest point



Surface reconstruction from unorganized points, Hoppe et al., ACM SIGGRAPH 1992

SDF from Points and Normals

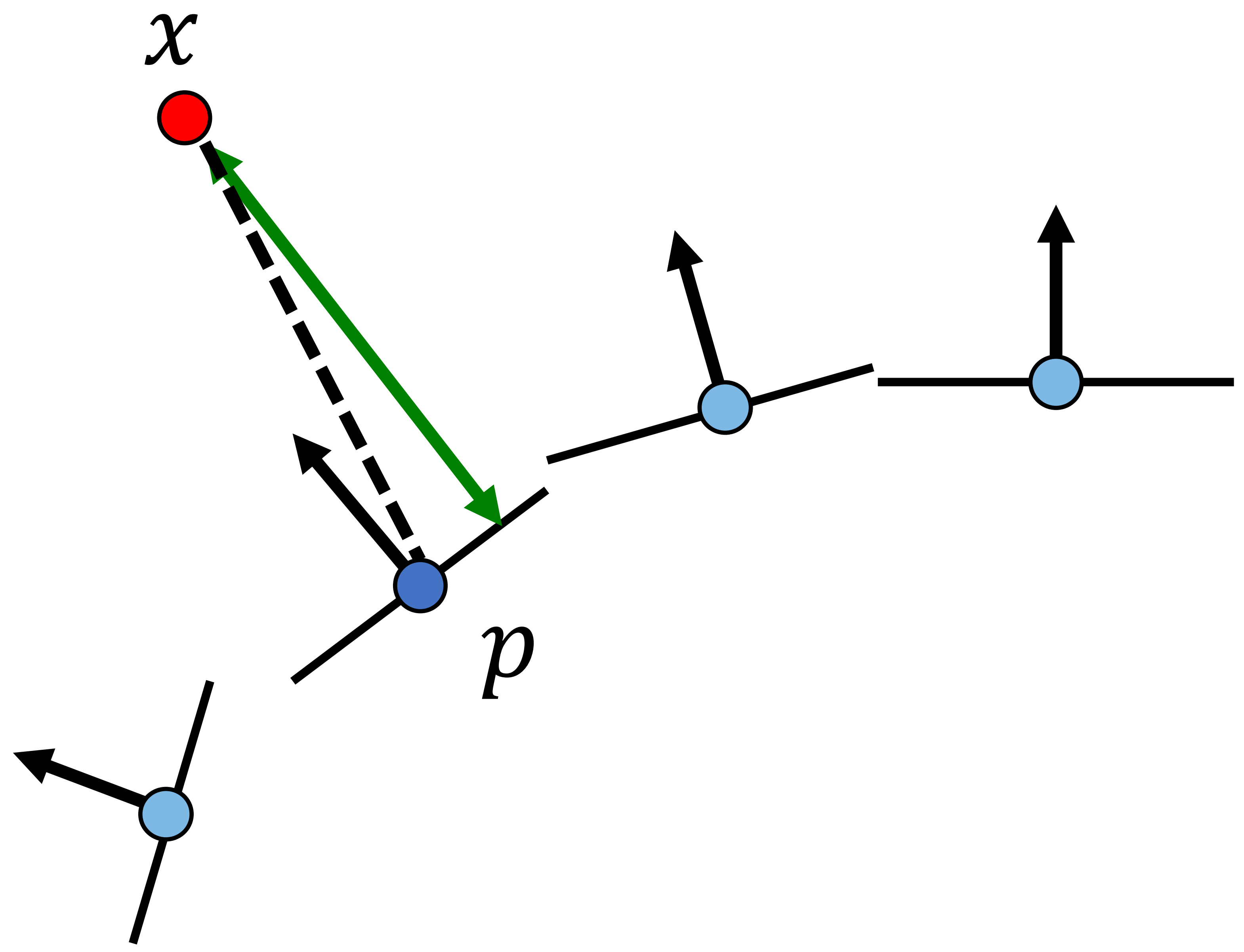
- Compute signed distance to the tangent plane of the closest point



Surface reconstruction from unorganized points, Hoppe et al., ACM SIGGRAPH 1992

SDF from Points and Normals

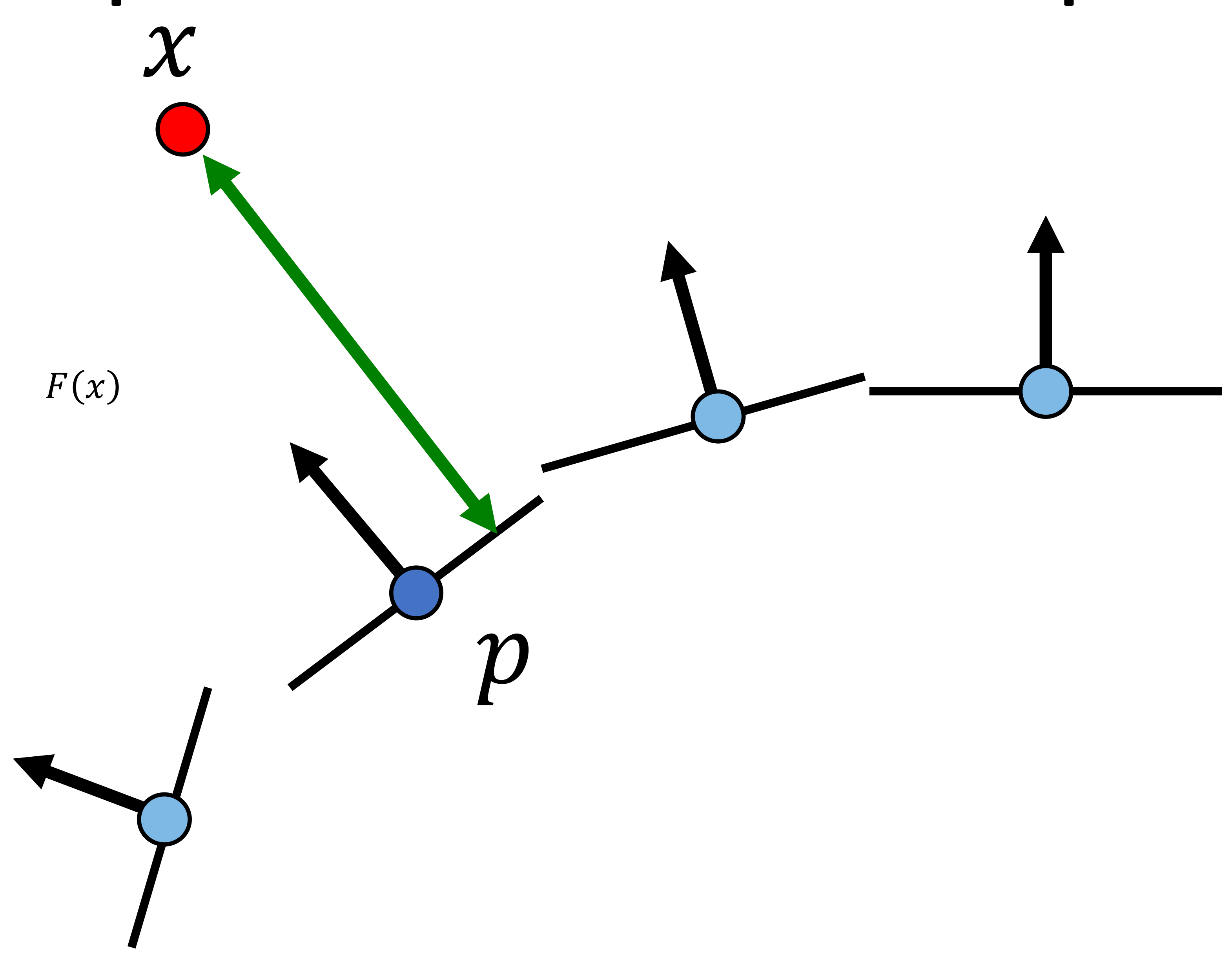
- Compute signed distance to the tangent plane of the closest point



Surface reconstruction from unorganized points, Hoppe et al., ACM SIGGRAPH 1992

SDF from Points and Normals

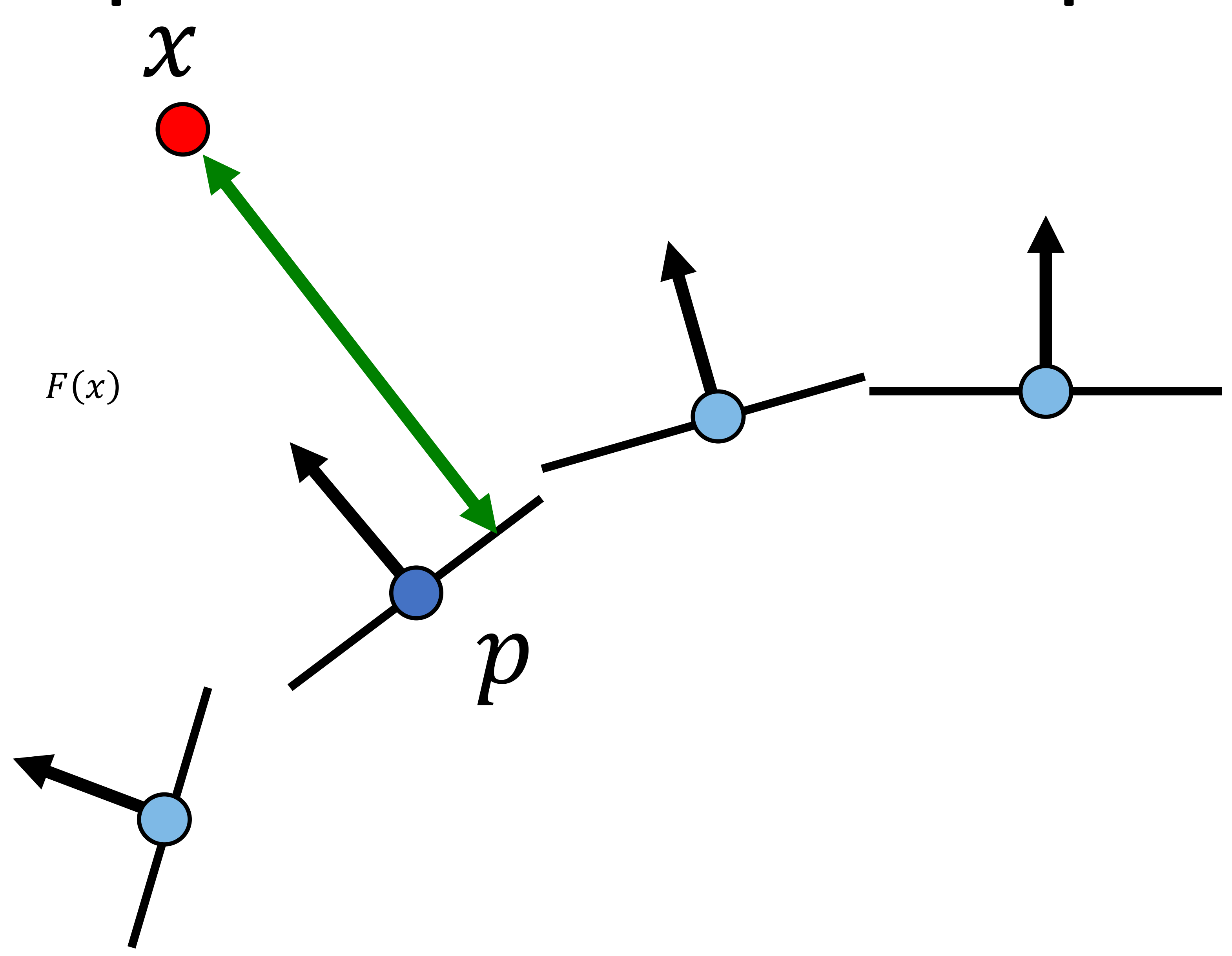
- Compute signed distance to the tangent plane of the closest point



Surface reconstruction from unorganized points, Hoppe et al., ACM SIGGRAPH 1992

SDF from Points and Normals

- Compute signed distance to the tangent plane of the closest point
- Is the function continuous?

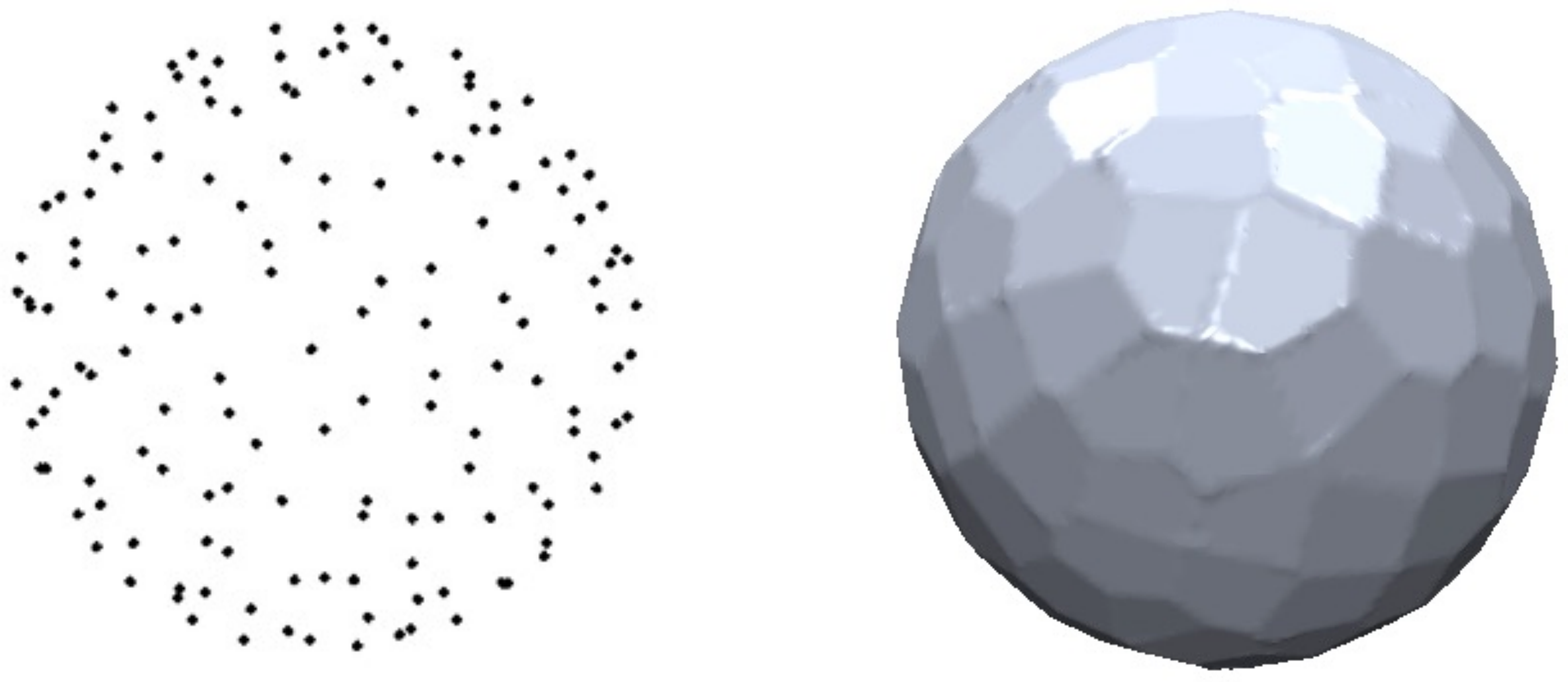


Surface reconstruction from unorganized points, Hoppe et al., ACM SIGGRAPH 1992

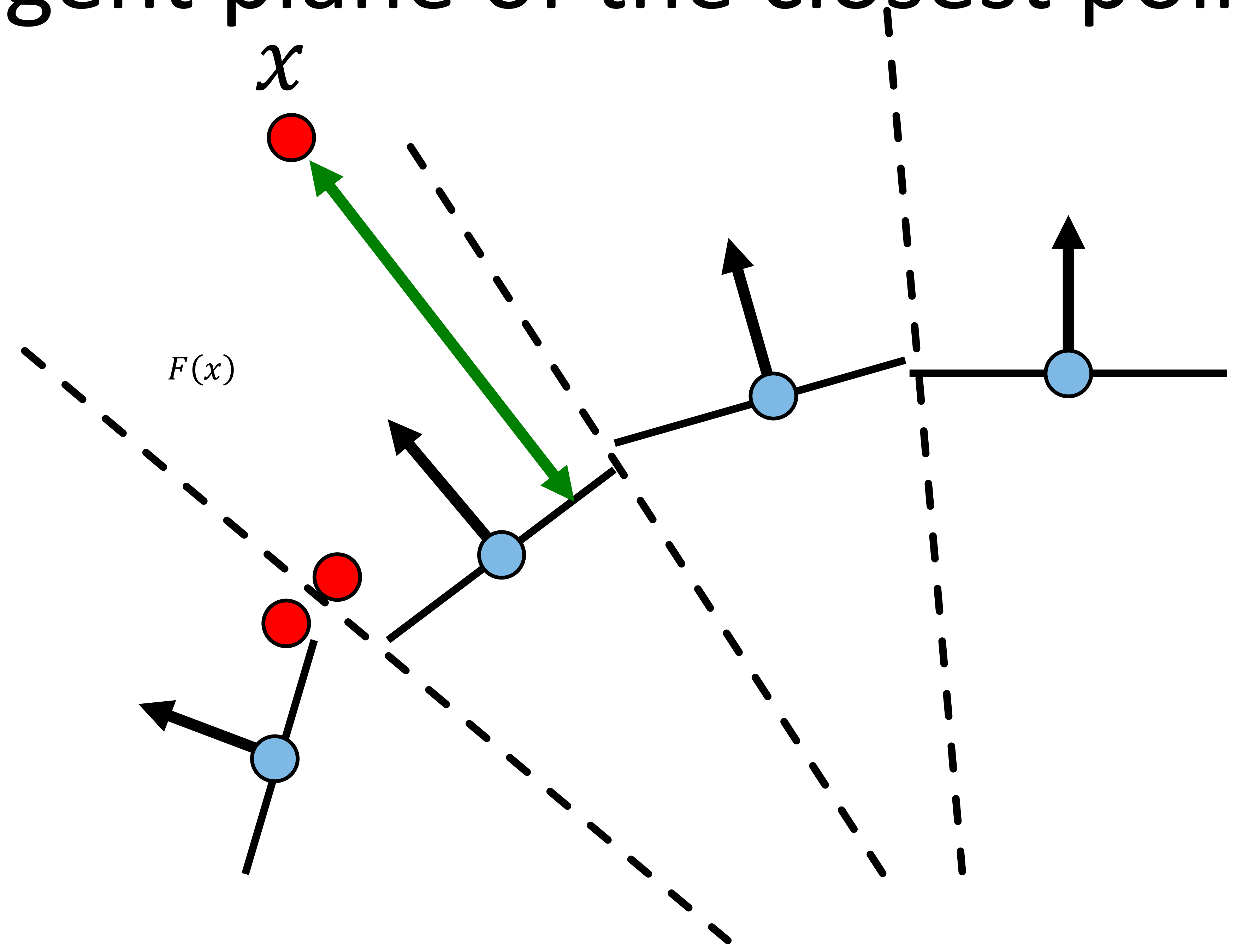
SDF from Points and Normals

- Compute signed distance to the tangent plane of the closest point

- The function is discontinuous



Input Output



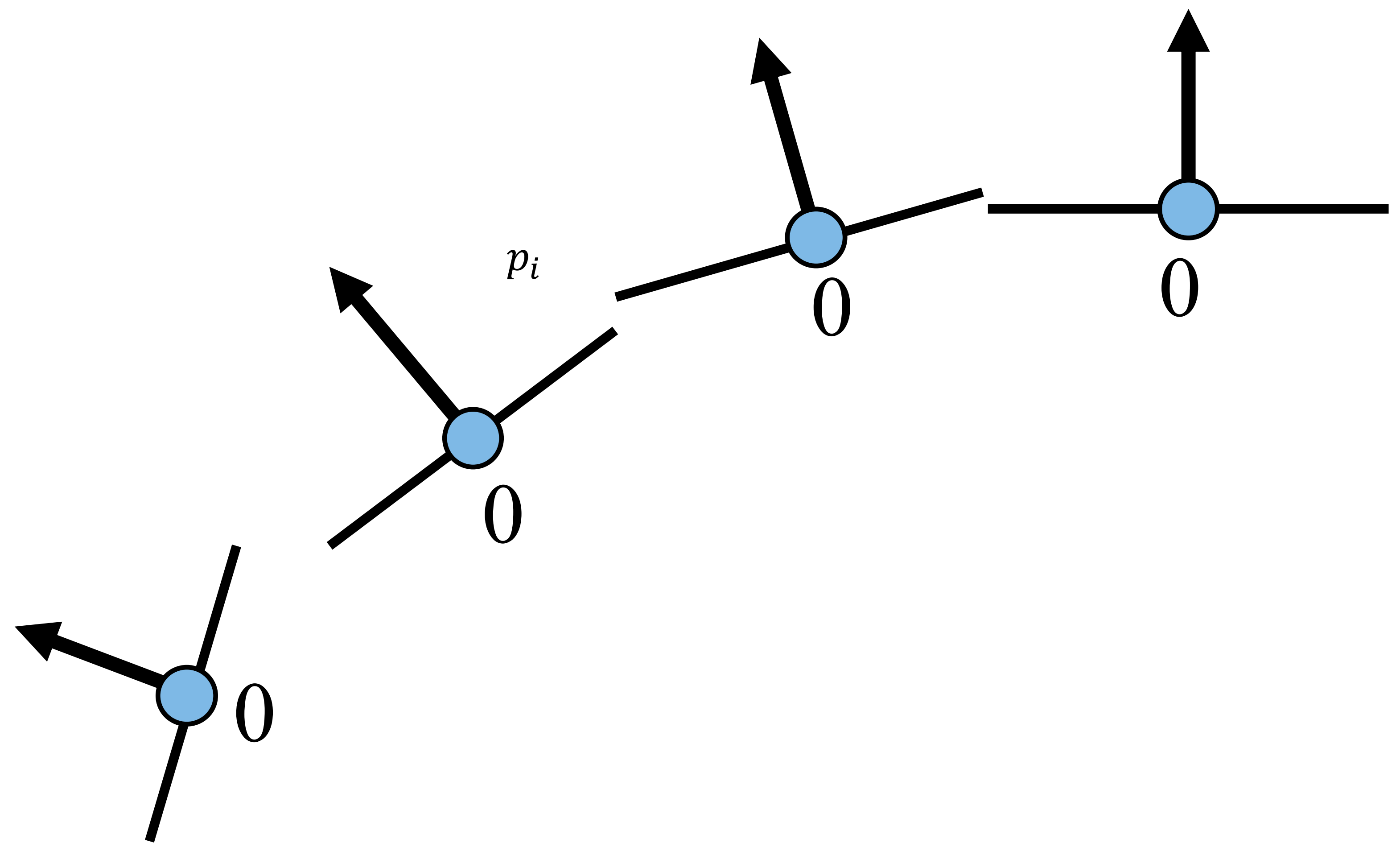
Surface reconstruction from unorganized points, Hoppe et al., ACM SIGGRAPH 1992

Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
    1. Sign Distance Function: distance from points and tangent planes, [Hoppe et al., SIGGRAPH 1992]
    2. **Sign Distance Function: radial basis function (RBF), [Carr et al., SIGGRAPH 2001]**

Smooth SDF

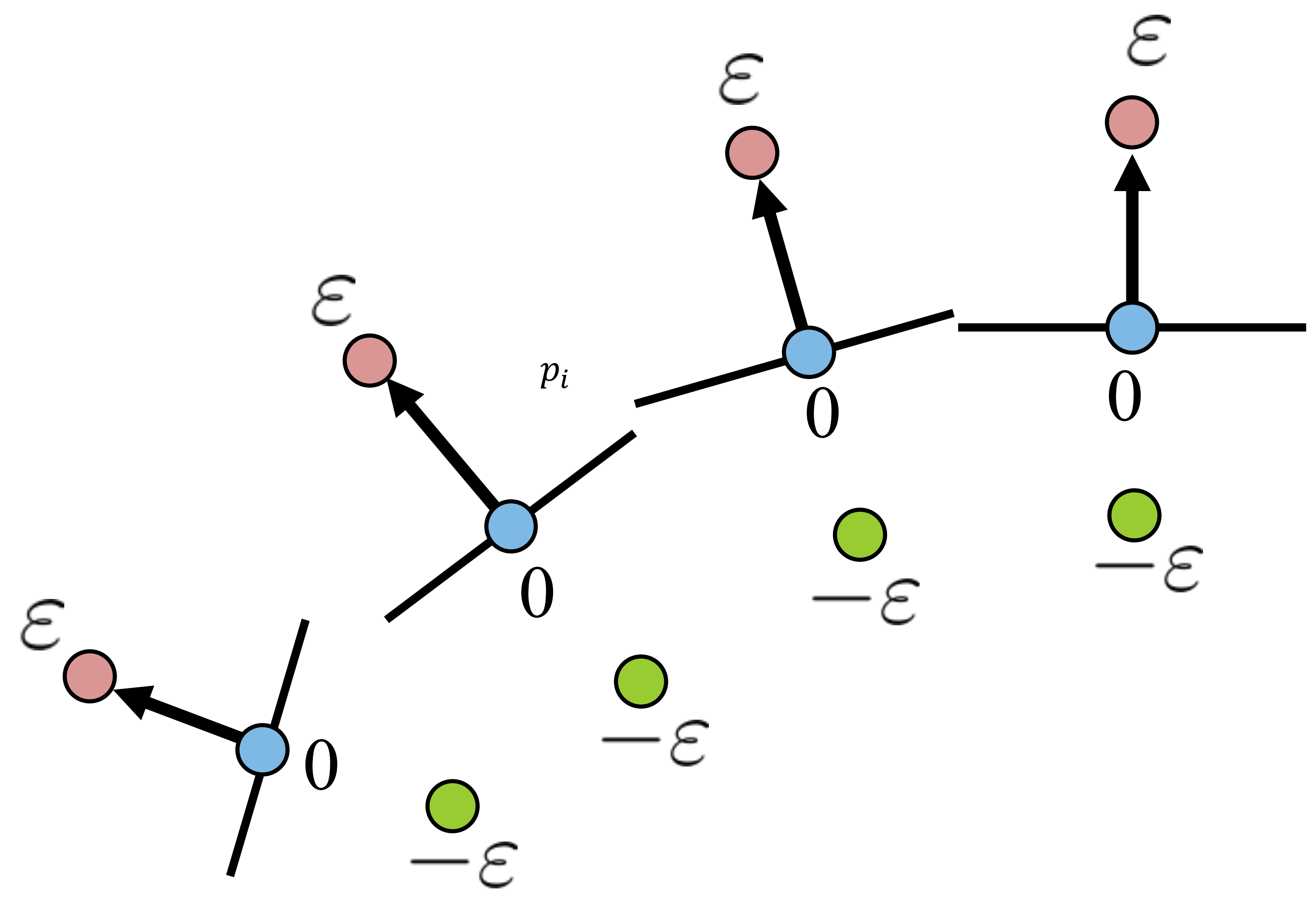
- Desired Properties:
  - F is smooth
  - $F(p_i) = 0$
  - Avoid trivial $F \equiv 0$



Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Smooth SDF

- Desired Properties:
  - F is smooth
  - $F(p_i) = 0$
  - Avoid trivial $F \equiv 0$
    - Add off-surface constraints

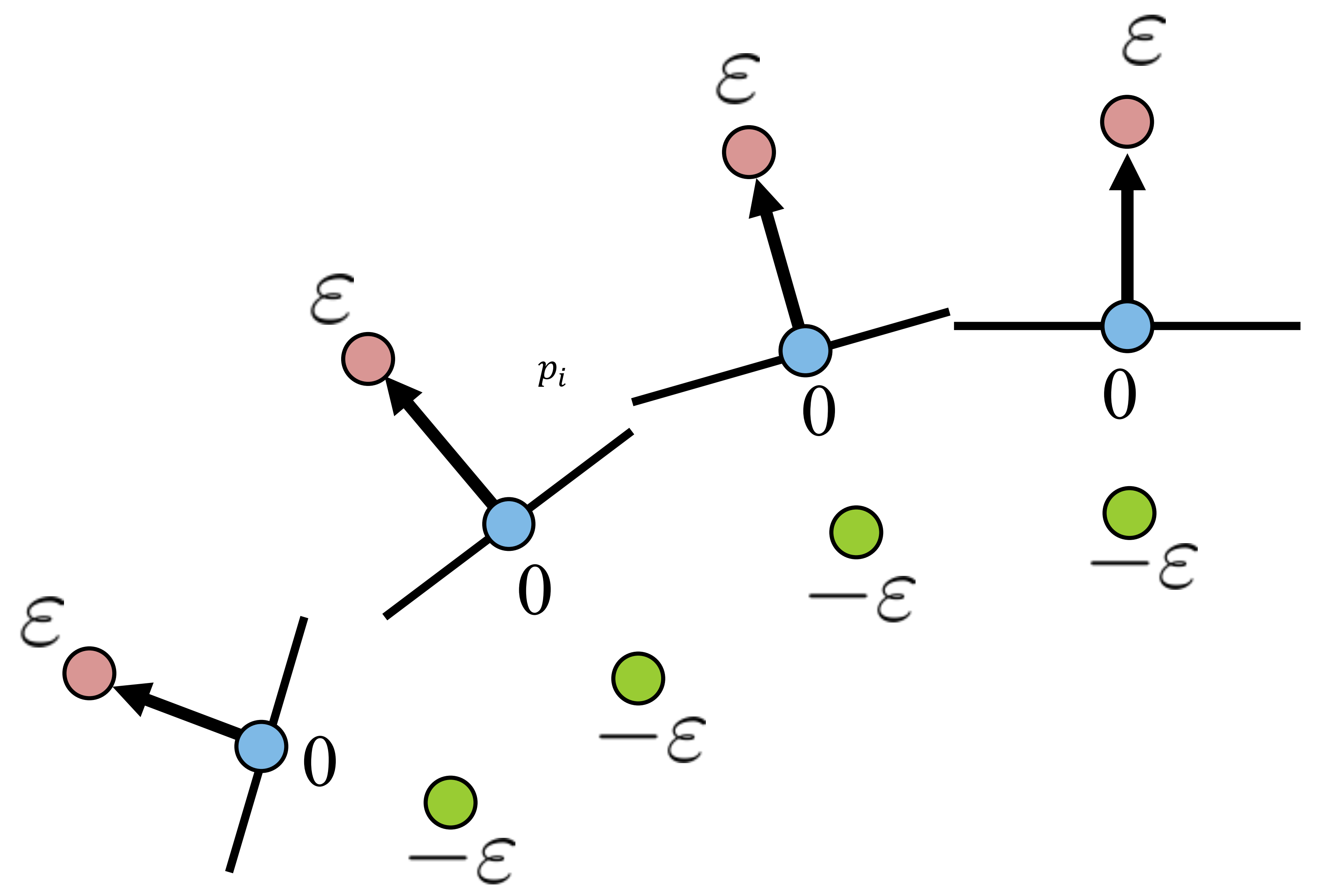


Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Smooth SDF

- Desired Properties:
  - F is smooth
  - $F(p_i) = 0$
  - Avoid trivial $F \equiv 0$
    - Add off-surface constraints

$$\begin{cases} F(p_i) = 0 \\ F(p_i + \varepsilon n_i) = \varepsilon \\ F(p_i - \varepsilon n_i) = -\varepsilon \end{cases}$$

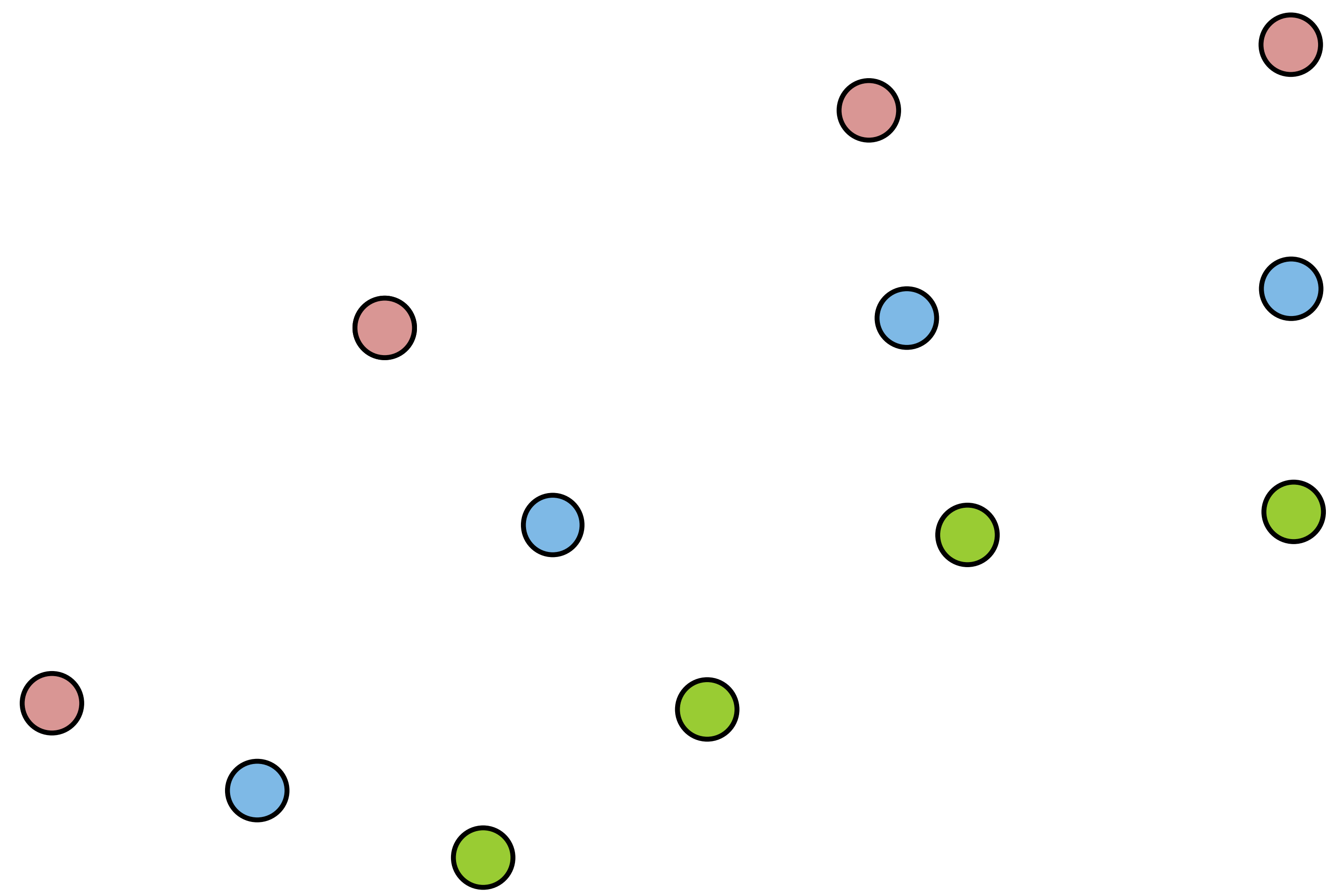


Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Radial Basis Function

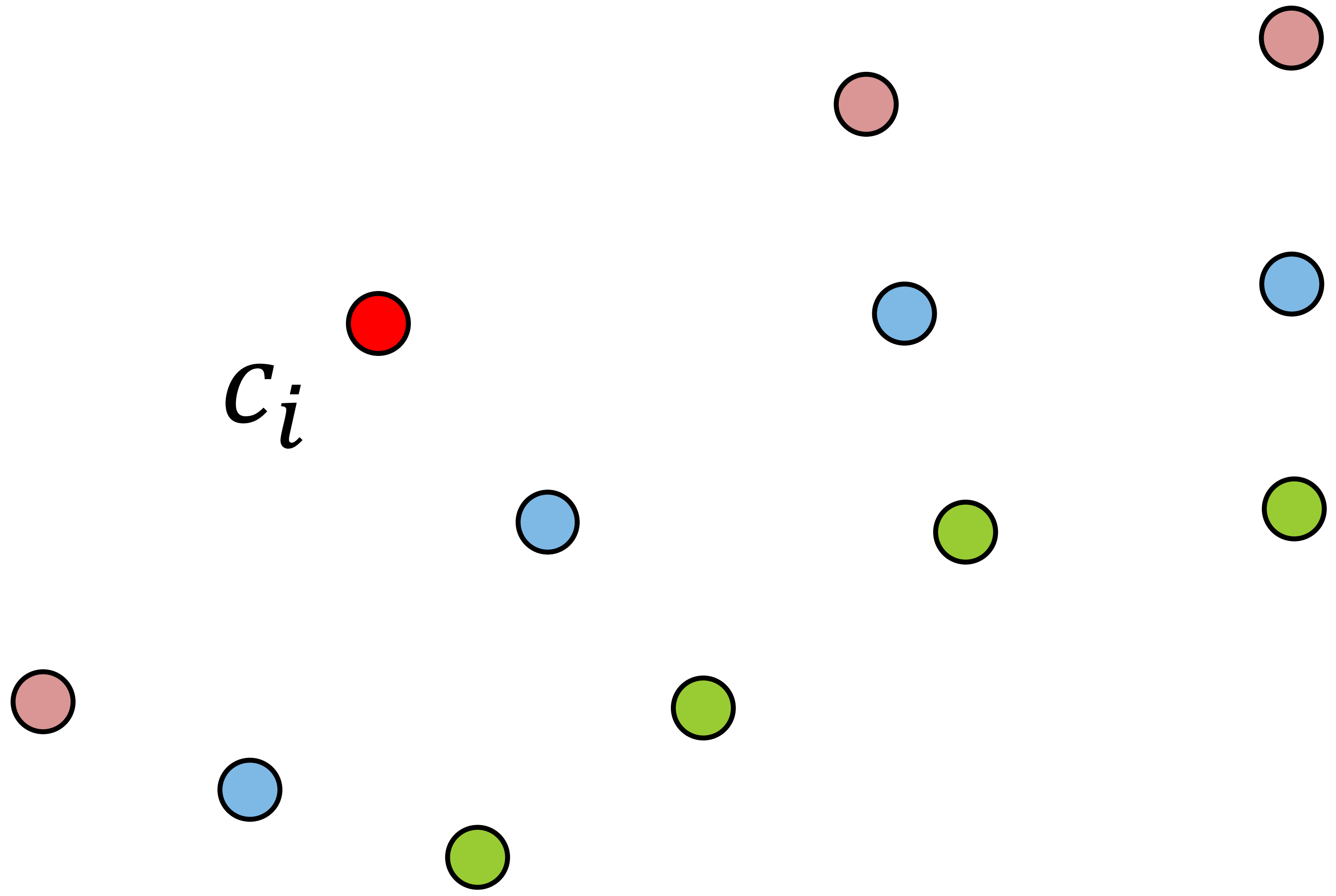
Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Radial Basis Function



Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

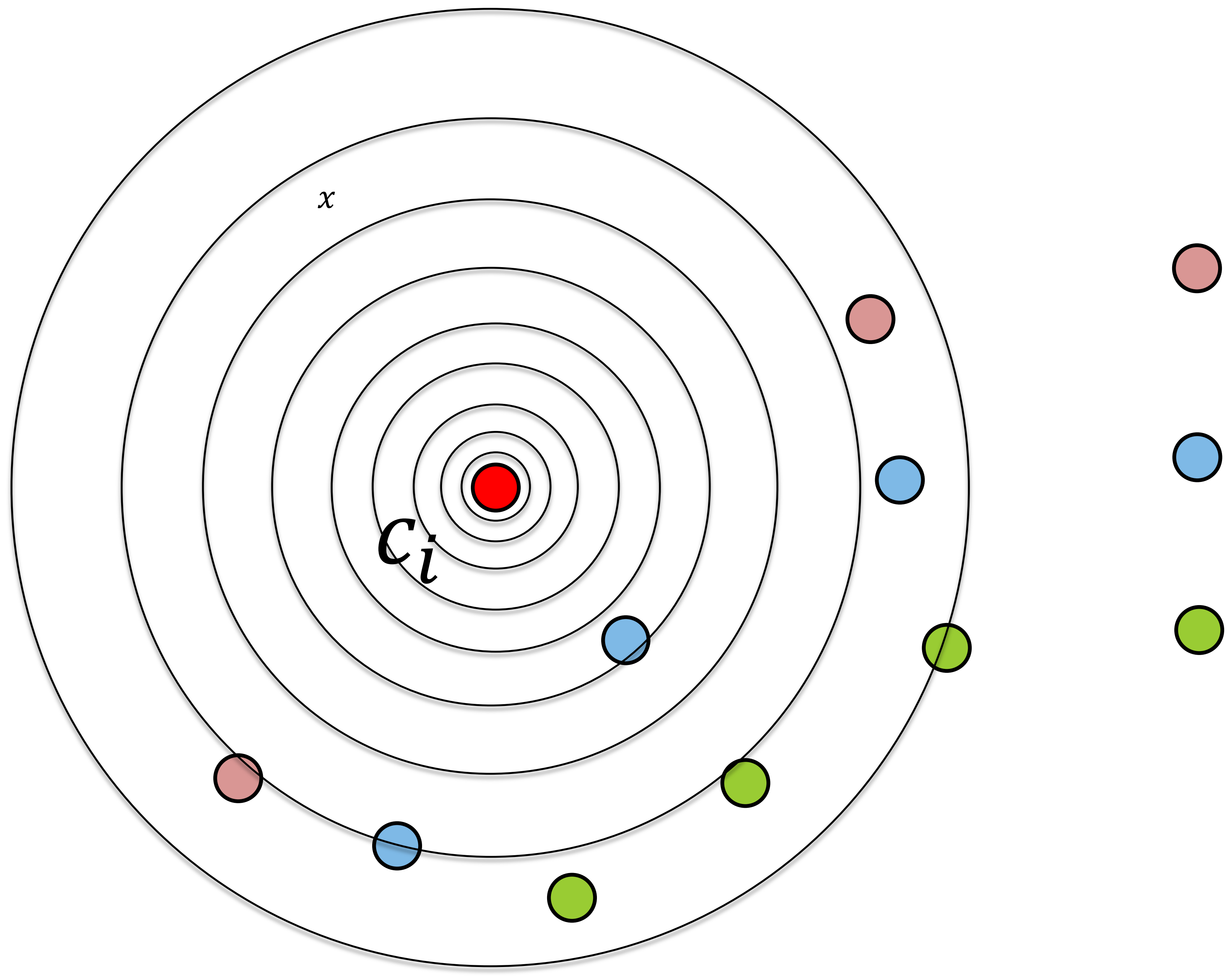
Radial Basis Function



Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

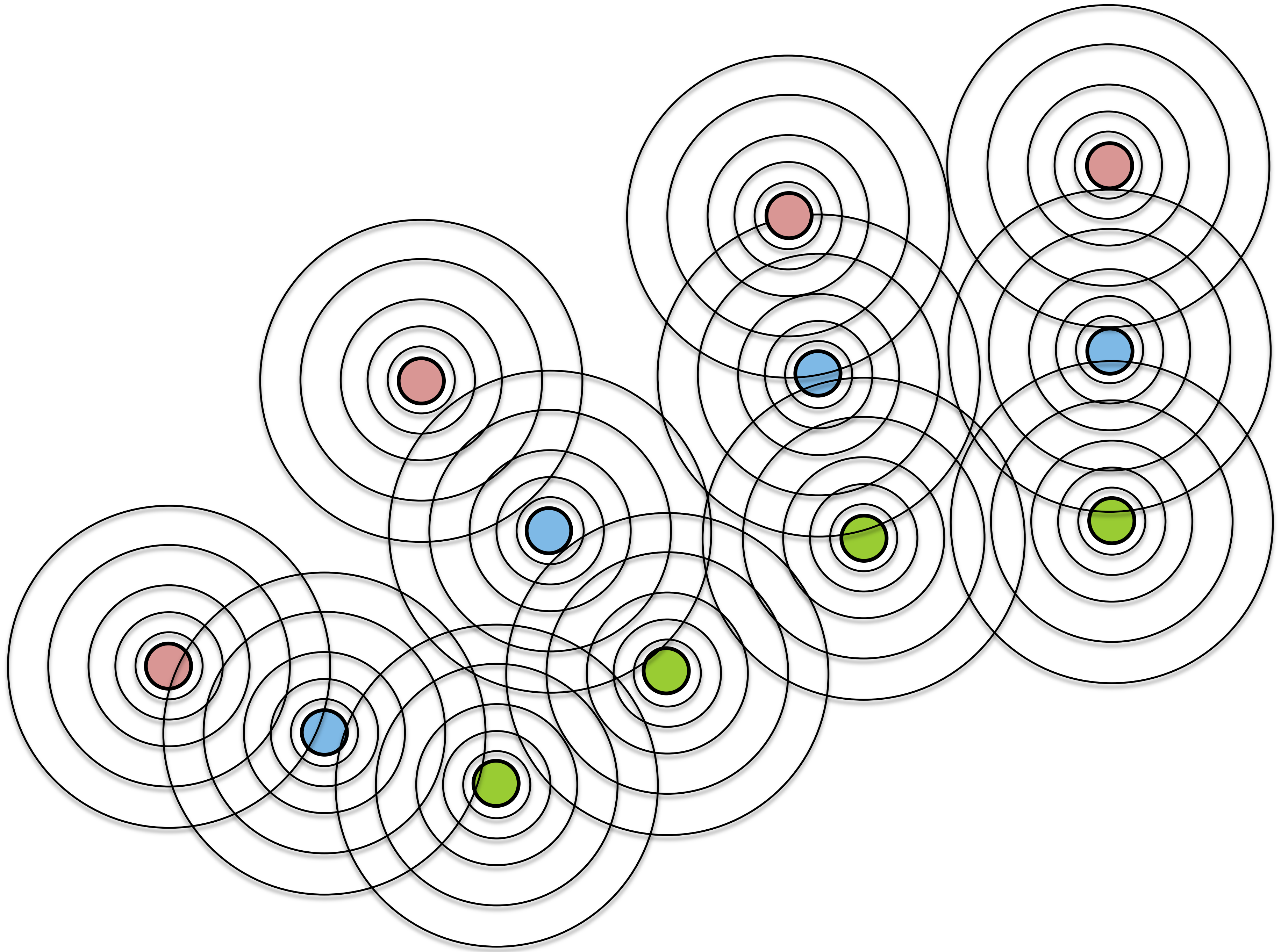
Radial Basis Function

$$\varphi(\|x - c_i\|) = (\|x - c_i\|)^3$$



Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

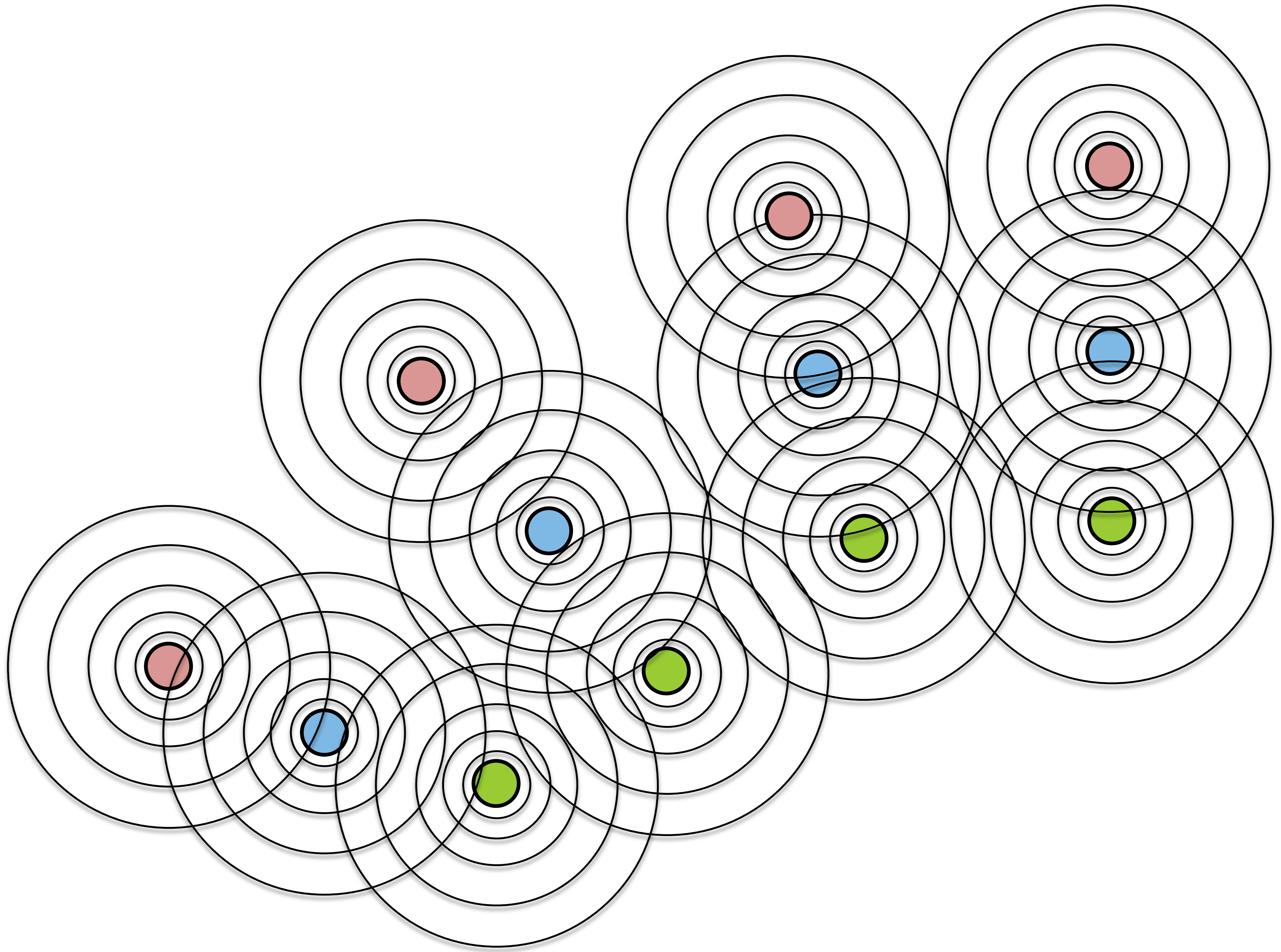
Radial Basis Function



Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Radial Basis Function

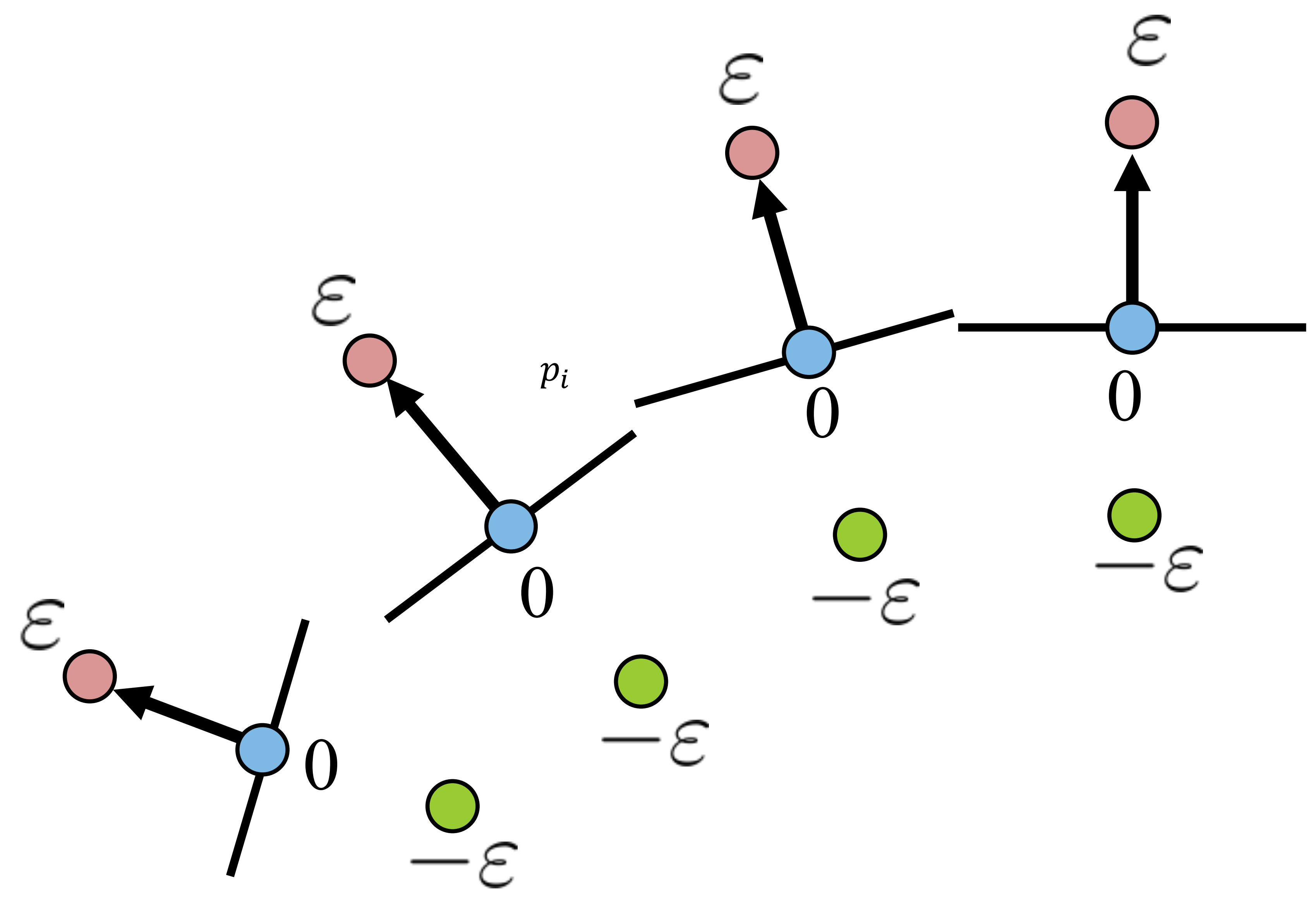
$$F(x) = \sum_{i=0}^{N-1} w_i \varphi(\|x - c_i\|)$$



Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Radial Basis Function Interpolation

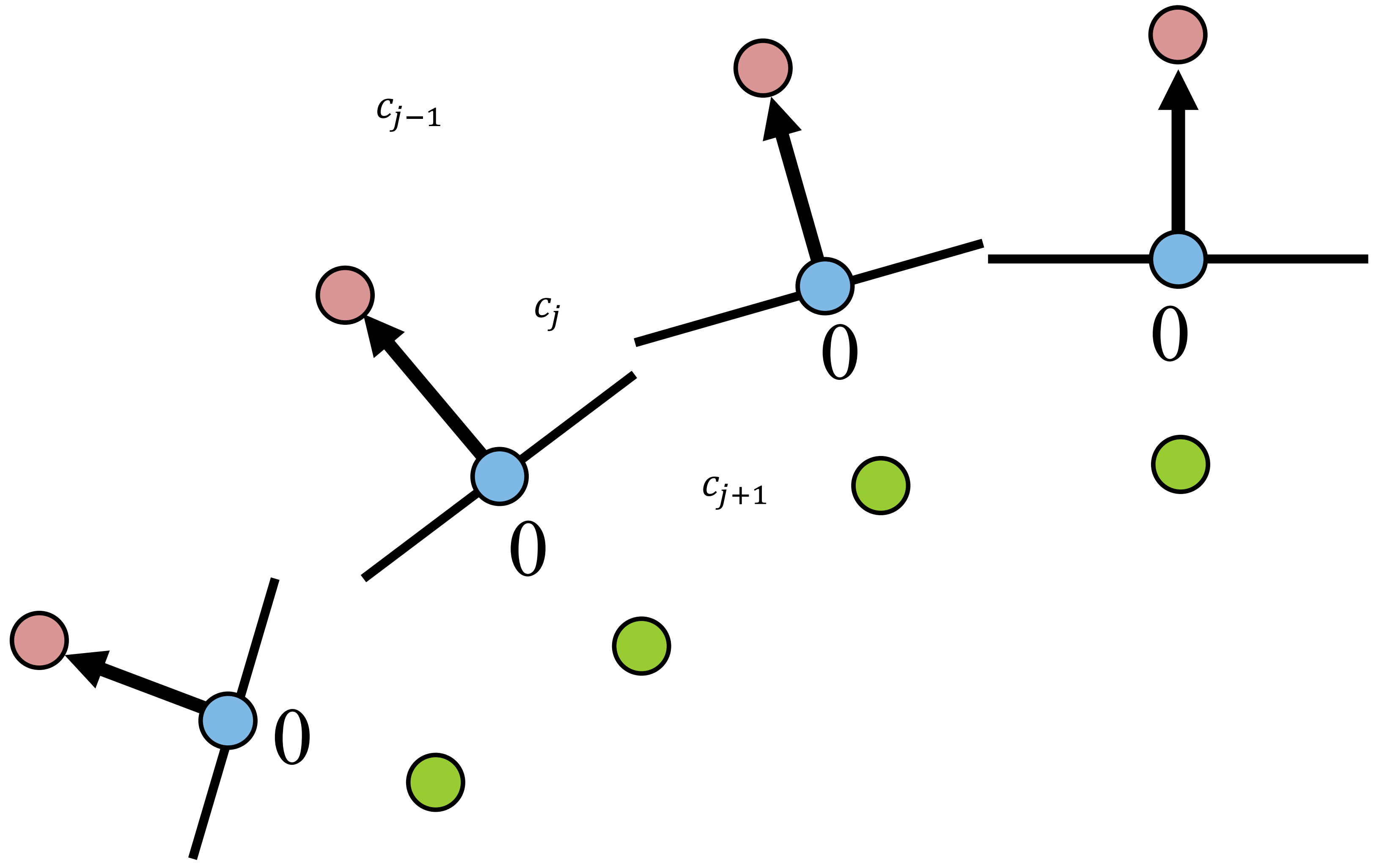
$$\begin{cases} F(p_i) = 0 \\ F(p_i + \varepsilon n_i) = \varepsilon \\ F(p_i - \varepsilon n_i) = -\varepsilon \end{cases}$$



Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Radial Basis Function Interpolation

$$\begin{cases} F(p_i) = F(c_j) = 0 \\ F(p_i + \varepsilon n_i) = F(c_{j-1}) = \varepsilon \\ F(p_i - \varepsilon n_i) = F(c_{j+1}) = -\varepsilon \end{cases}$$

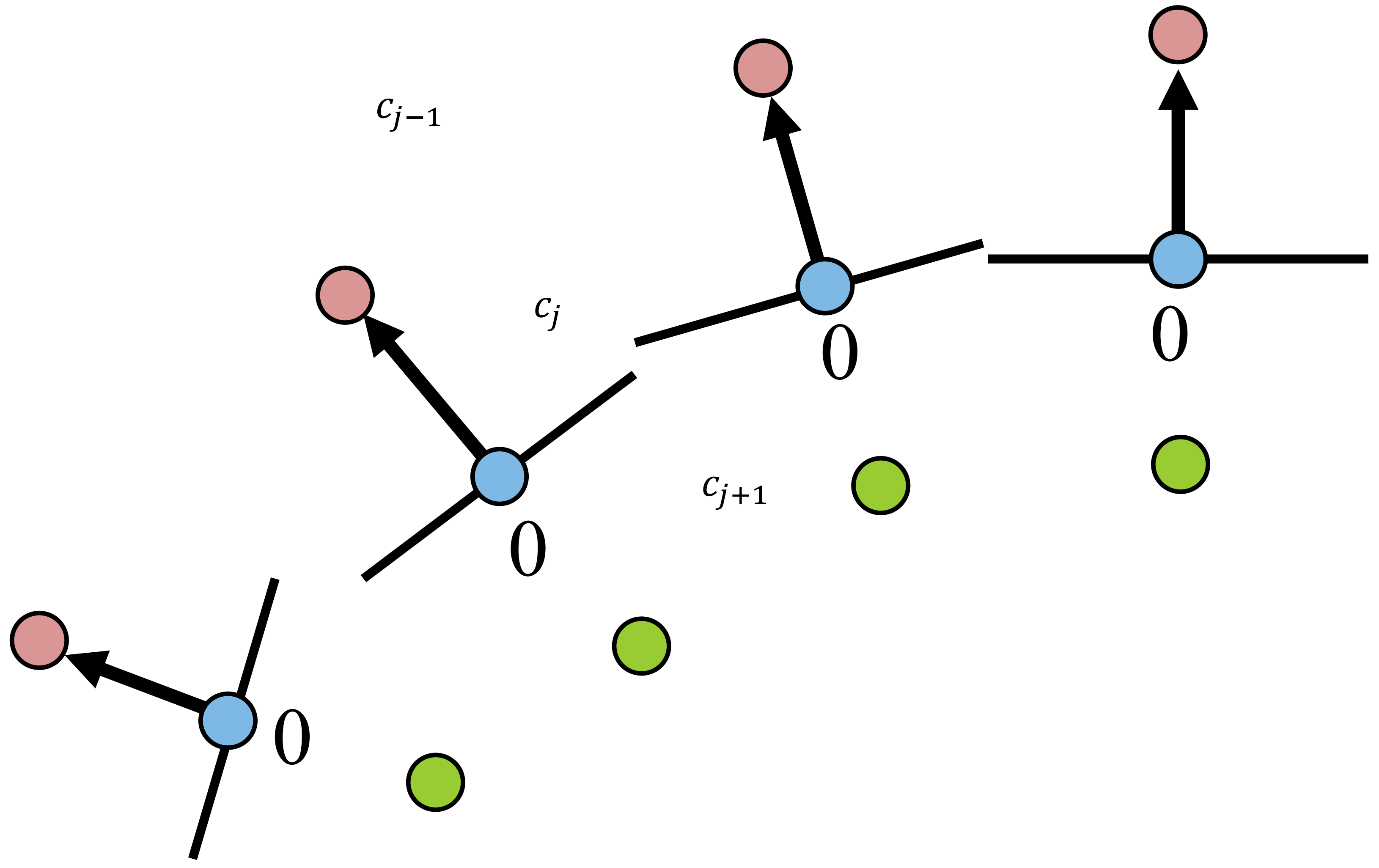


Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Radial Basis Function Interpolation

$$F(x) = \sum_{i=0}^{N-1} w_i \varphi(\|x - c_i\|)$$

$$\begin{cases} F(c_0) = \varepsilon \\ F(c_1) = 0 \\ F(c_2) = -\varepsilon \\ \vdots \\ F(c_{N-3}) = \varepsilon \\ F(c_{N-2}) = 0 \\ F(c_{N-1}) = -\varepsilon \end{cases}$$



Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Radial Basis Function Interpolation

$$F(x) = \sum_{i=0}^{N-1} w_i \varphi(\|x - c_i\|)$$

$$\left\{\begin{array}{l} F(c_0) = \varepsilon \\ F(c_1) = 0 \\ F(c_2) = -\varepsilon \\ \vdots \\ F(c_{N-3}) = \varepsilon \\ F(c_{N-2}) = 0 \\ F(c_{N-1}) = -\varepsilon \end{array}\right.$$

$$\rightarrow \boxed{\sum_{i=0}^{N-1} w_i \varphi(\|c_0 - c_i\|) = \varepsilon}$$

Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Radial Basis Function Interpolation

$$F(x) = \sum_{i=0}^{N-1} w_i \varphi(\|x - c_i\|)$$

$$\left\{\begin{array}{l} F(c_0) = \varepsilon \\ F(c_1) = 0 \\ F(c_2) = -\varepsilon \\ \vdots \\ F(c_{N-3}) = \varepsilon \\ F(c_{N-2}) = 0 \\ F(c_{N-1}) = -\varepsilon \end{array}\right. \rightarrow \boxed{\begin{array}{c} \displaystyle\sum_{i=0}^{N-1} w_i \varphi(\|c_0 - c_i\|) = \varepsilon \\ \downarrow \\ \begin{pmatrix} \varphi(\|c_j - c_0\|) & \cdots & \varphi(\|c_0 - c_{N-1}\|) \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_{N-1} \end{pmatrix} = \varepsilon \end{array}}$$

Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Radial Basis Function Interpolation

- Symmetric linear system to get the weights:

$$\begin{pmatrix} \varphi(\|c_0 - c_0\|) & \cdots & \varphi(\|c_0 - c_{N-1}\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|c_{N-1} - c_0\|) & \cdots & \varphi(\|c_{N-1} - c_{N-1}\|) \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_{N-1} \end{pmatrix} = \begin{pmatrix} \varepsilon \\ \vdots \\ -\varepsilon \end{pmatrix}$$

$$\downarrow$$

$$A\,x = b$$

Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Radial Basis Function Interpolation

- Dense linear system solver

https://eigen.tuxfamily.org/dox/group__DenseDecompositionBenchmark.html

- The size of A grows quadratically with the size of $|x|$

$$\begin{pmatrix} \varphi(\|c_0 - c_0\|) & \cdots & \varphi(\|c_0 - c_{N-1}\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|c_{N-1} - c_0\|) & \cdots & \varphi(\|c_{N-1} - c_{N-1}\|) \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_{N-1} \end{pmatrix} = \begin{pmatrix} \varepsilon \\ \vdots \\ -\varepsilon \end{pmatrix}$$

$$\downarrow$$

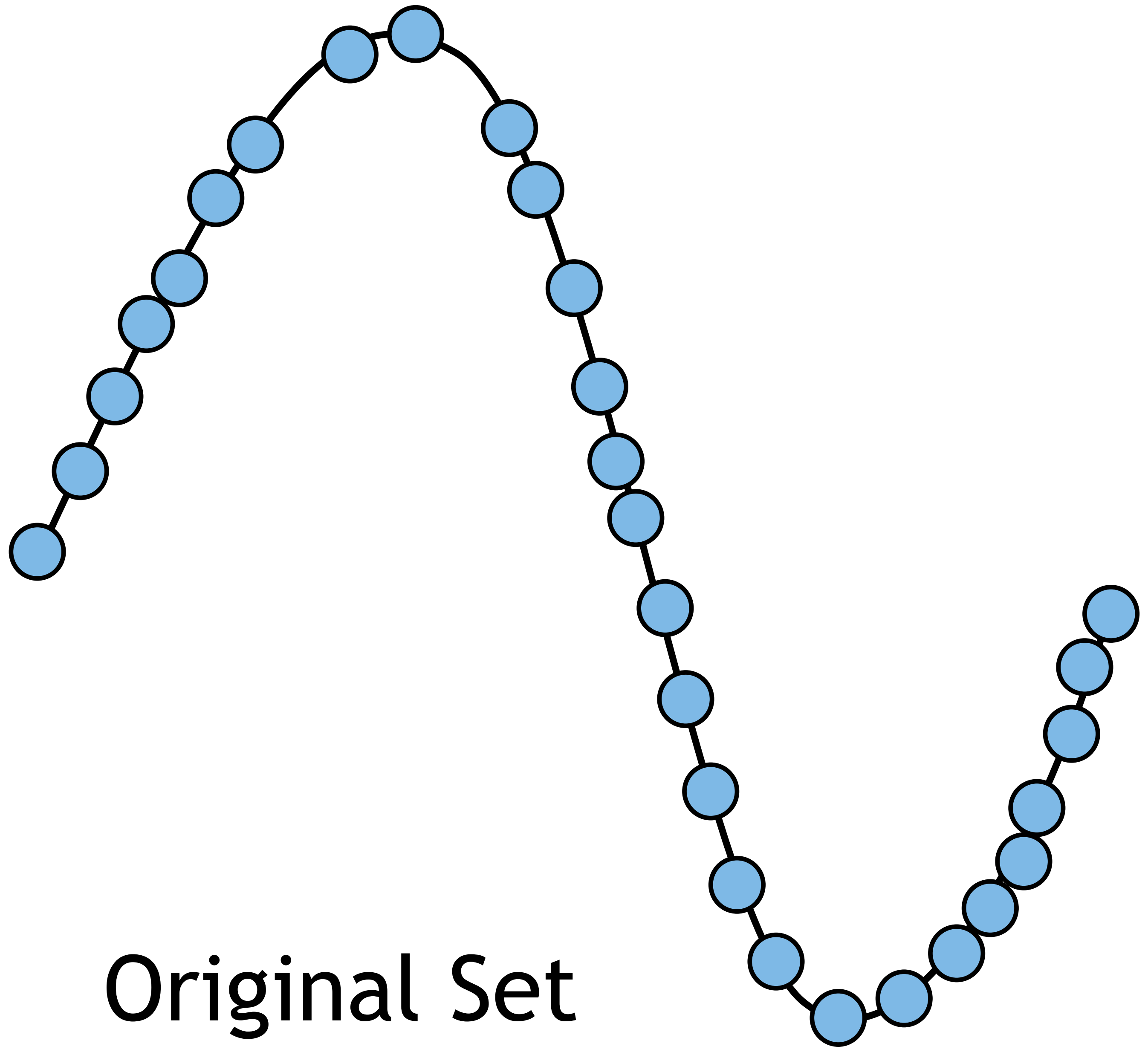
$$A\,x = b$$

Radial Basis Function Interpolation

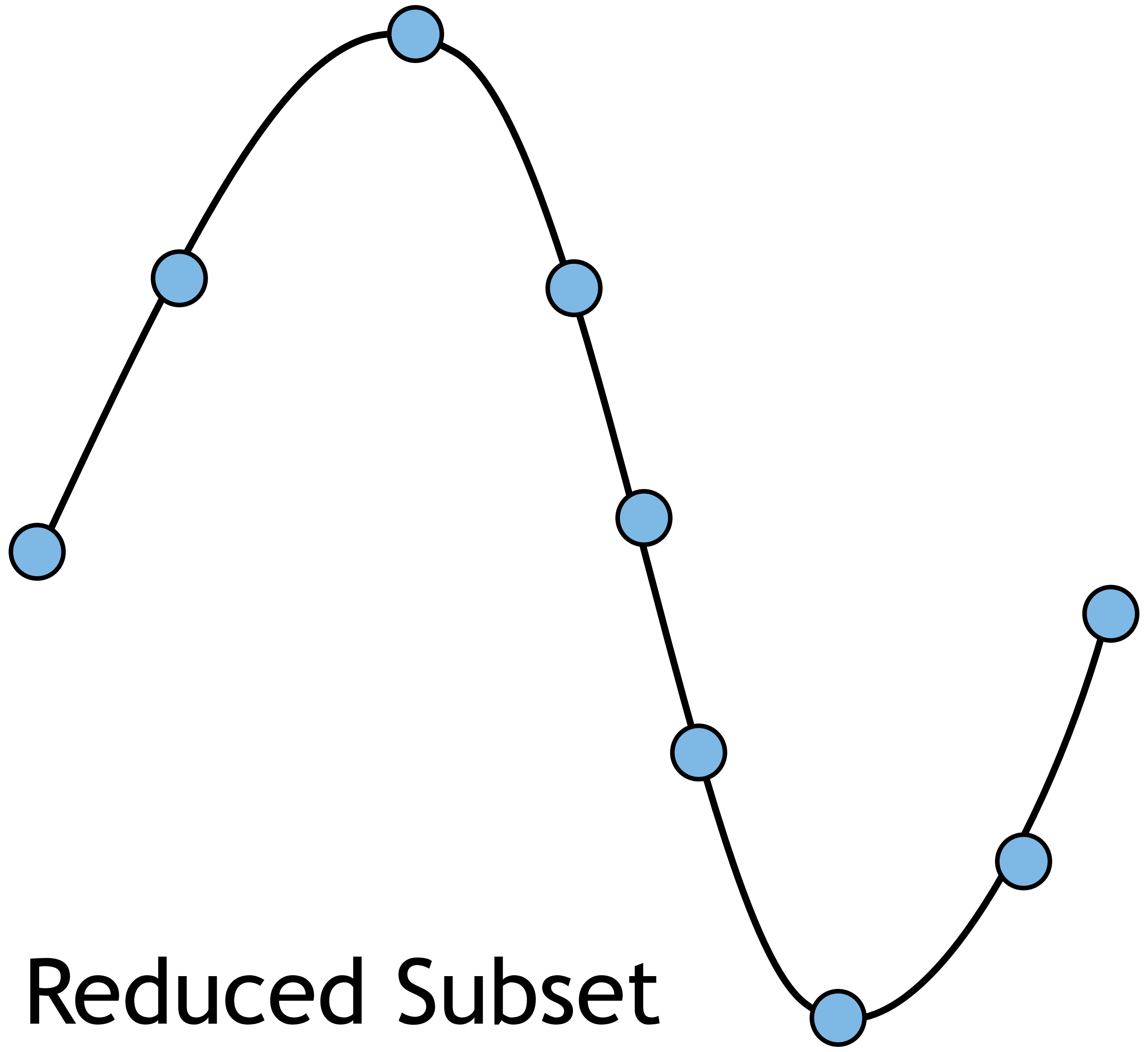
- Dense linear system solver

https://eigen.tuxfamily.org/dox/group__DenseDecompositionBenchmark.html

- Choose a subset of the input



Original Set



Reduced Subset

Radial Basis Function Interpolation

- Dense matrix –> sparse matrix

https://eigen.tuxfamily.org/dox/group__TopicSparseSystems.html

$$\begin{pmatrix} \varphi(\|c_0 - c_0\|) & \cdots & \varphi(\|c_0 - c_{N-1}\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|c_{N-1} - c_0\|) & \cdots & \varphi(\|c_{N-1} - c_{N-1}\|) \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_{N-1} \end{pmatrix} = \begin{pmatrix} \varepsilon \\ \vdots \\ -\varepsilon \end{pmatrix}$$

$$\Big\downarrow$$

$$A\,x = b$$

Radial Basis Function Interpolation

- Dense matrix –> sparse matrix

https://eigen.tuxfamily.org/dox/group__TopicSparseSystems.html

- Thresholding $\varphi(r) = \begin{cases} r^3 & 0 \leq r \leq \sigma \\ 0 & r > \sigma \end{cases}$, σ is a tunable parameter

Shengxin Zhu. Compactly supported radial basis functions: how and why? SIAM Review, 2012

Radial Basis Function Interpolation

- Dense matrix –> sparse matrix

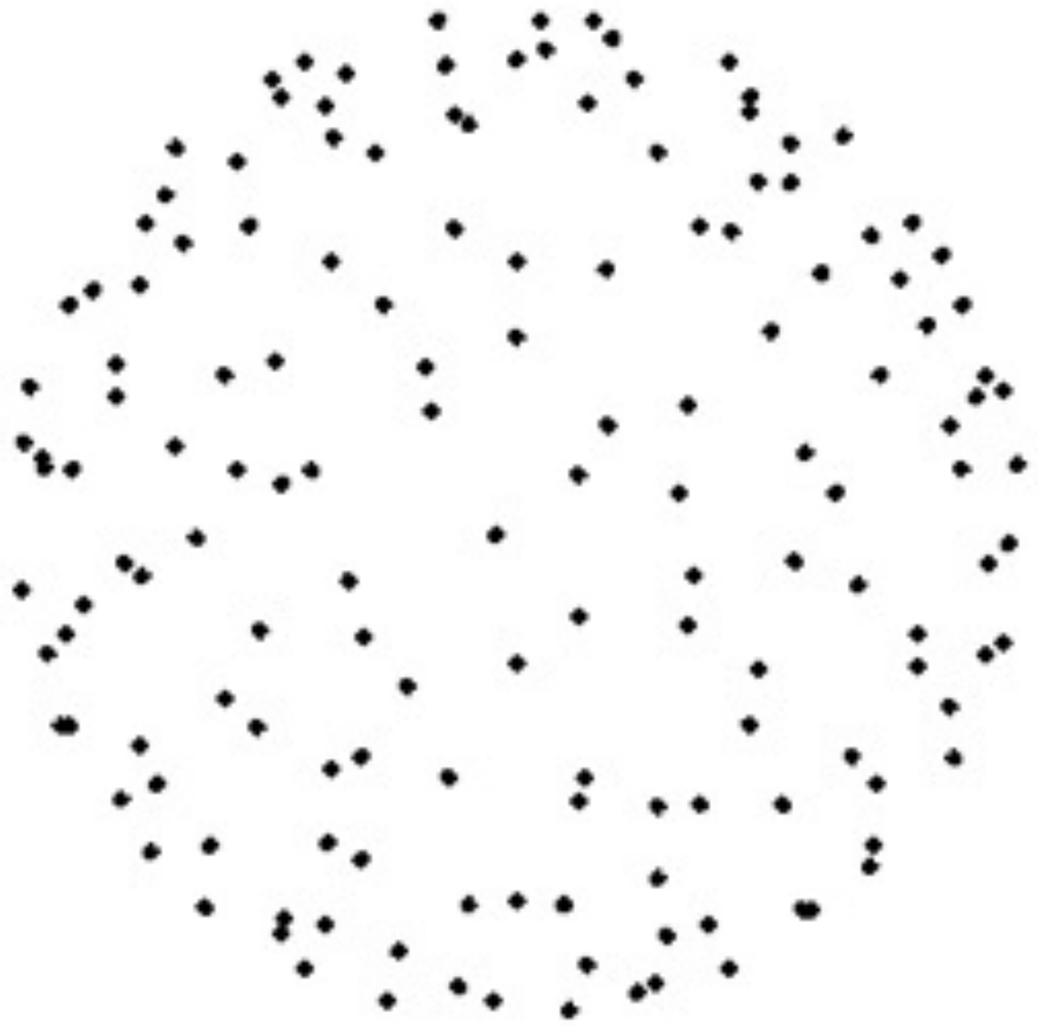
https://eigen.tuxfamily.org/dox/group__TopicSparseSystems.html

- Thresholding $\varphi(r) = \begin{cases} r^3 & 0 \le r \le \sigma \\ 0 & r > \sigma \end{cases}$, σ is a tunable parameter

- Wendland basis function: $\varphi(r) = \begin{cases} \left(1 - \frac{r}{\sigma}\right)_+^4 \left(4\frac{r}{\sigma} + 1\right) & 0 \le r \le \sigma \\ 0 & r > \sigma \end{cases}$, σ is a tunable parameter

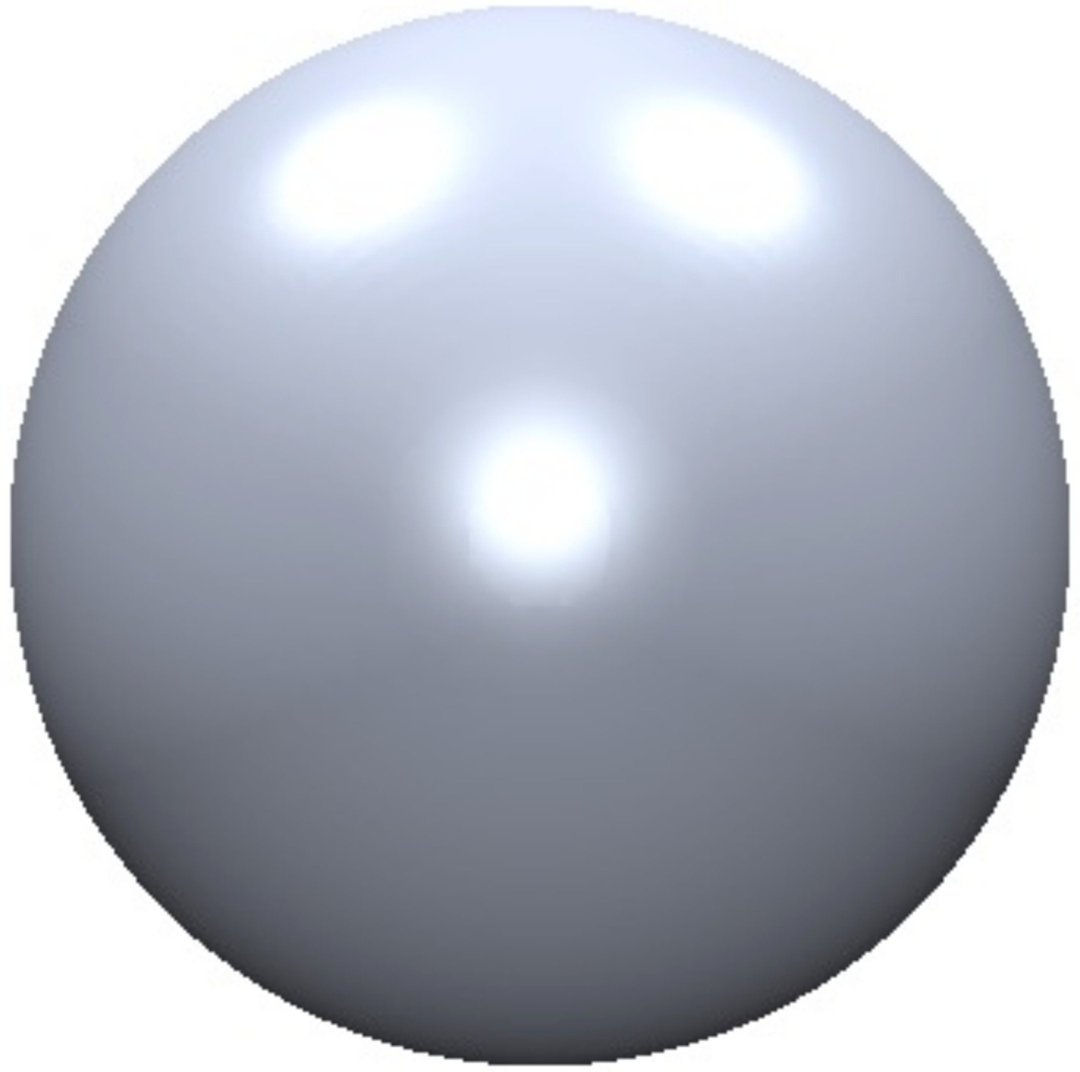
Shengxin Zhu. Compactly supported radial basis functions: how and why? SIAM Review, 2012

Comparison of the Two SDFs



Input

Distance to plane

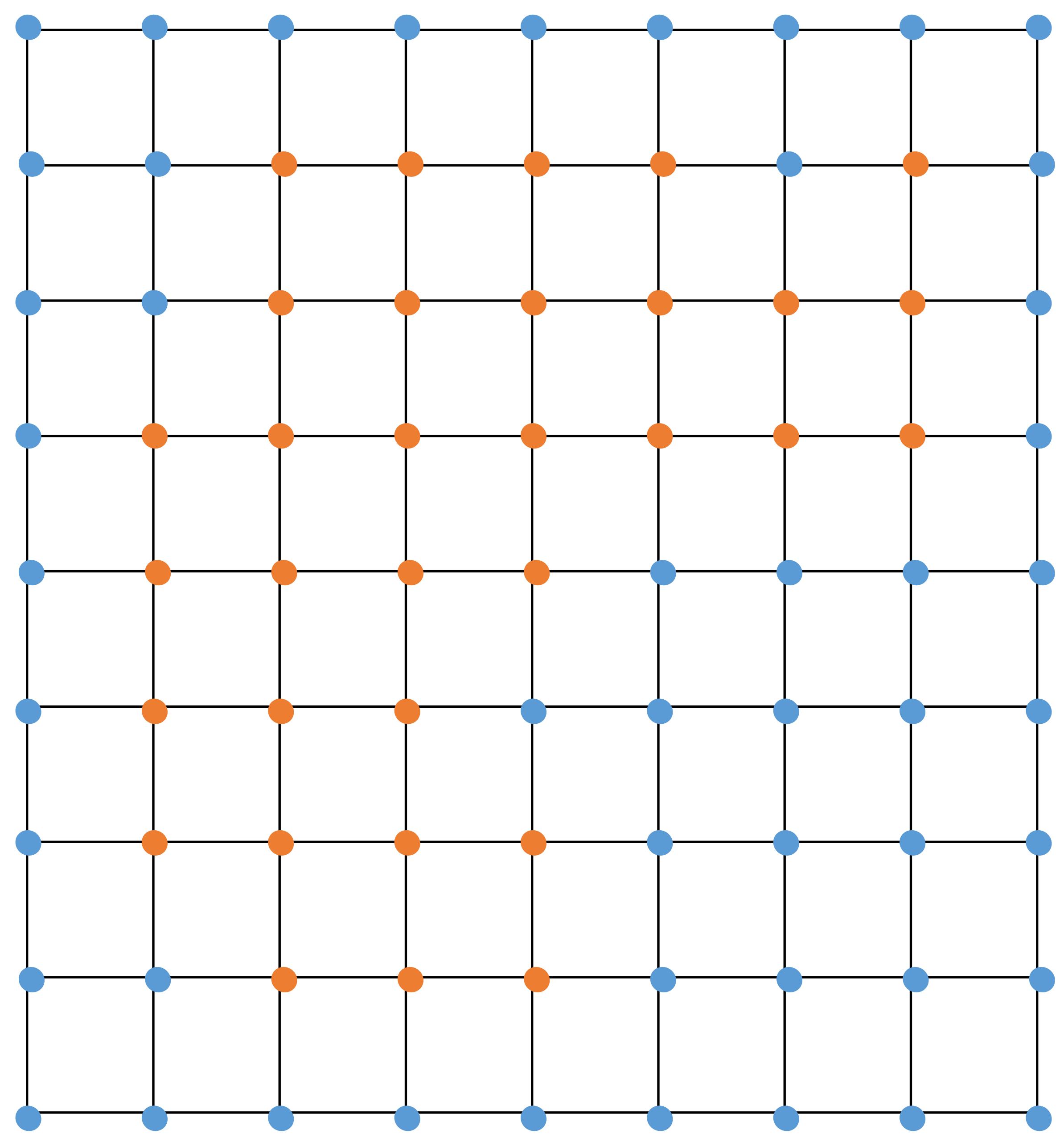


Triharmonic

Reconstruction and representation of 3D objects with radial basis functions, Carr et al., ACM SIGGRAPH 2001

Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
3. **Discretize function $F(x)$**
4. Extract zero Iso-surface

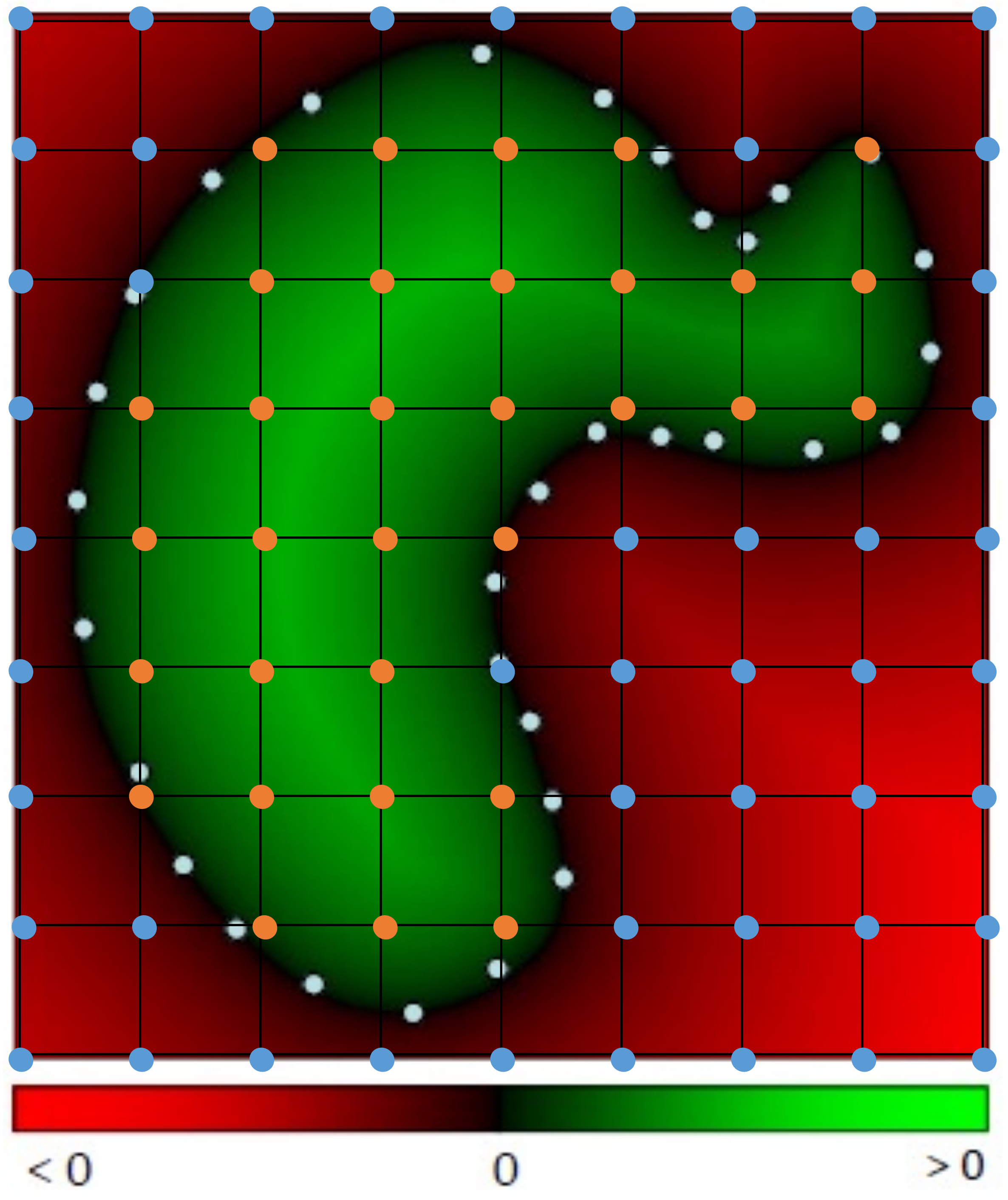


Discretization

- Can‘t explicitly compute all the roots
  - Sampling the level set is difficult (root finding)

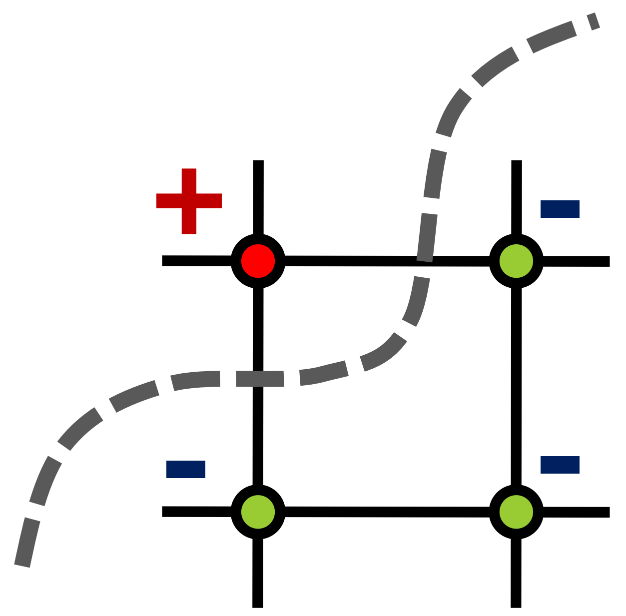
Discretization

- Can't explicitly compute all the roots
  - Sampling the level set is difficult (root finding)
- Solution: find approximate roots by trapping the implicit surface in a grid (lattice)



Discretization

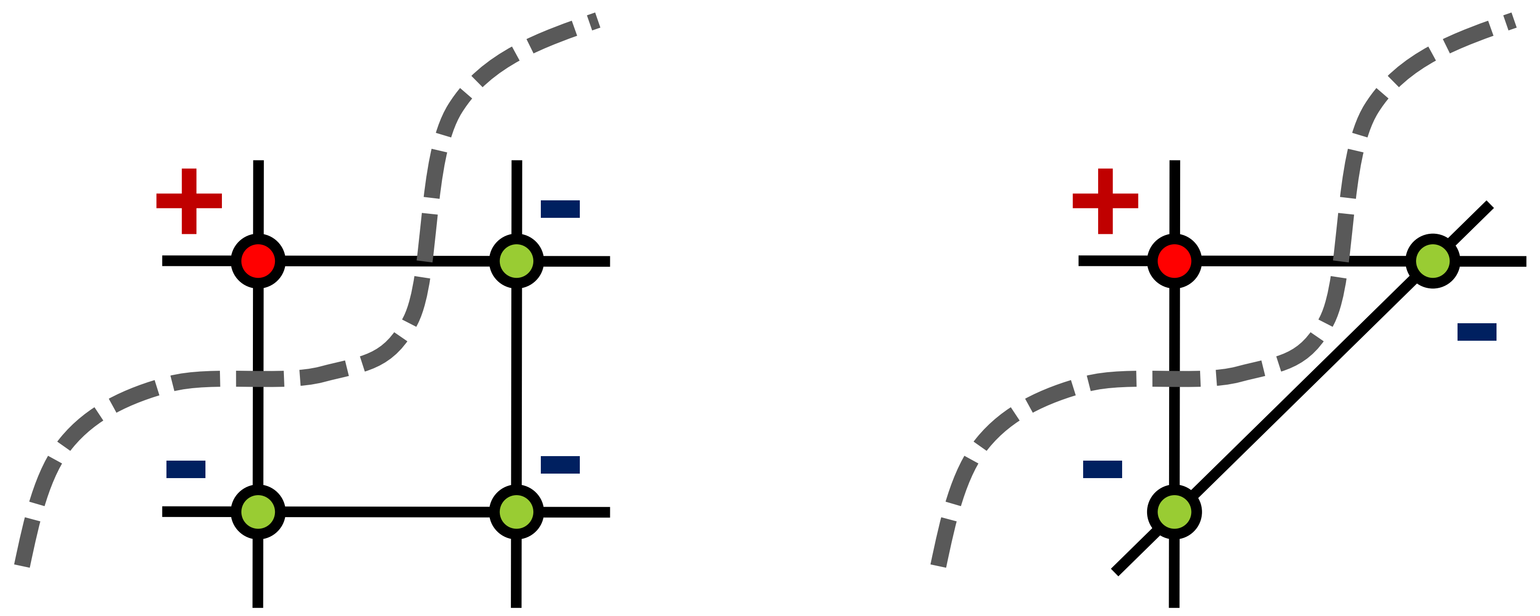
- Can‘t explicitly compute all the roots
  - Sampling the level set is difficult (root finding)
- Solution: find approximate roots by trapping the implicit surface in a grid (lattice)



$F(\mathbf{x}) < 0$

Discretization

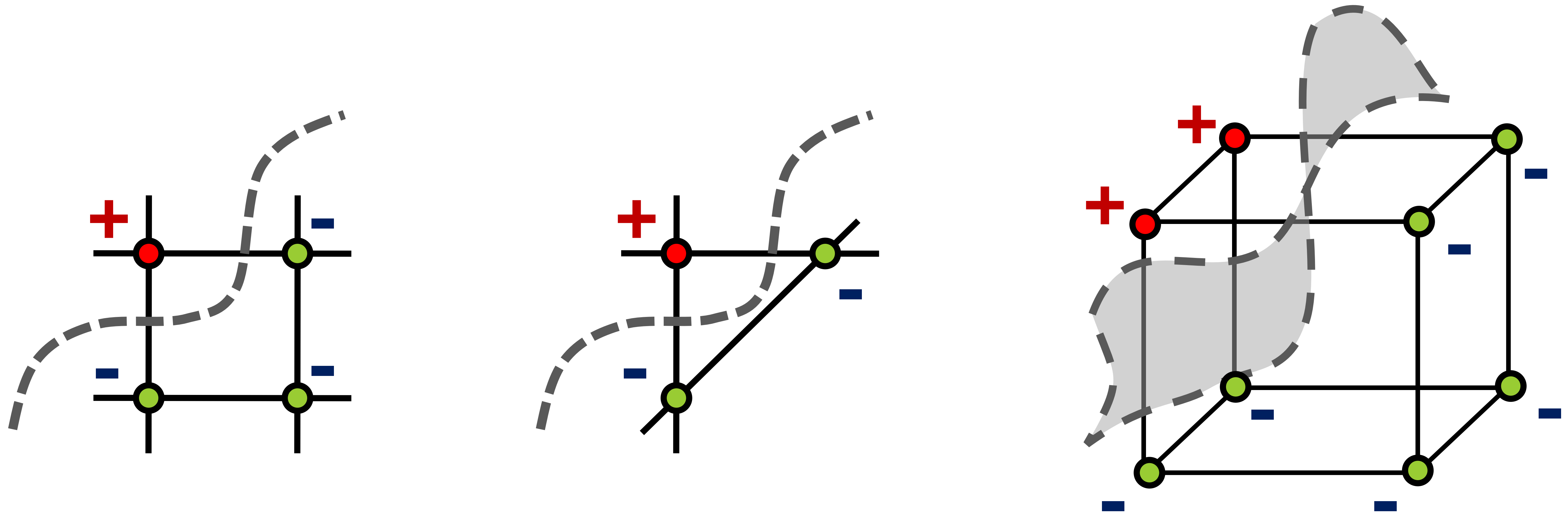
- Can‘t explicitly compute all the roots
  - Sampling the level set is difficult (root finding)
- Solution: find approximate roots by trapping the implicit surface in a grid (lattice)



$F(\mathbf{x}) < 0$

Discretization

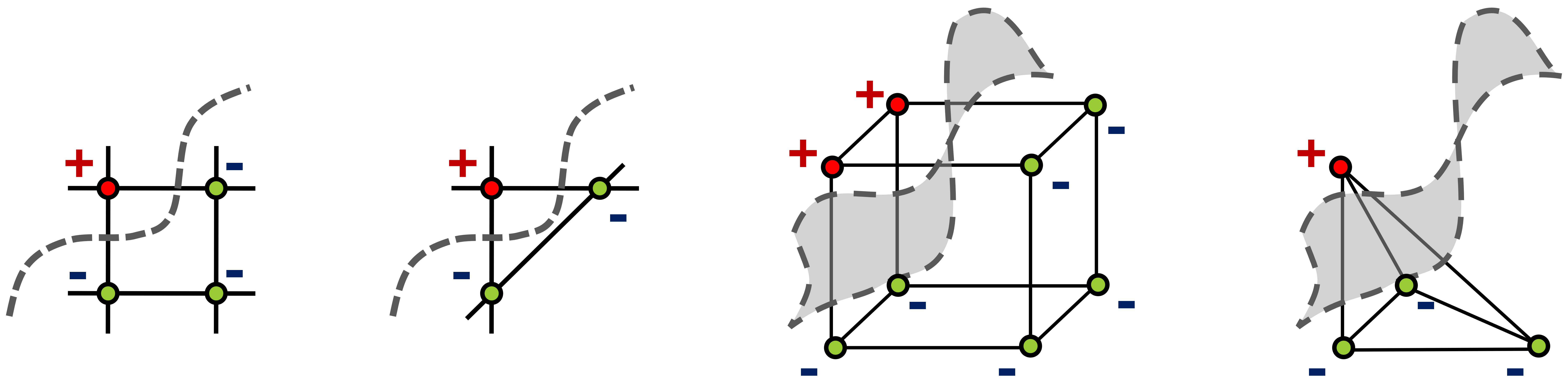
- Can‘t explicitly compute all the roots
  - Sampling the level set is difficult (root finding)
- Solution: find approximate roots by trapping the implicit surface in a grid (lattice)



$$F(\mathbf{x}) < 0$$

Discretization

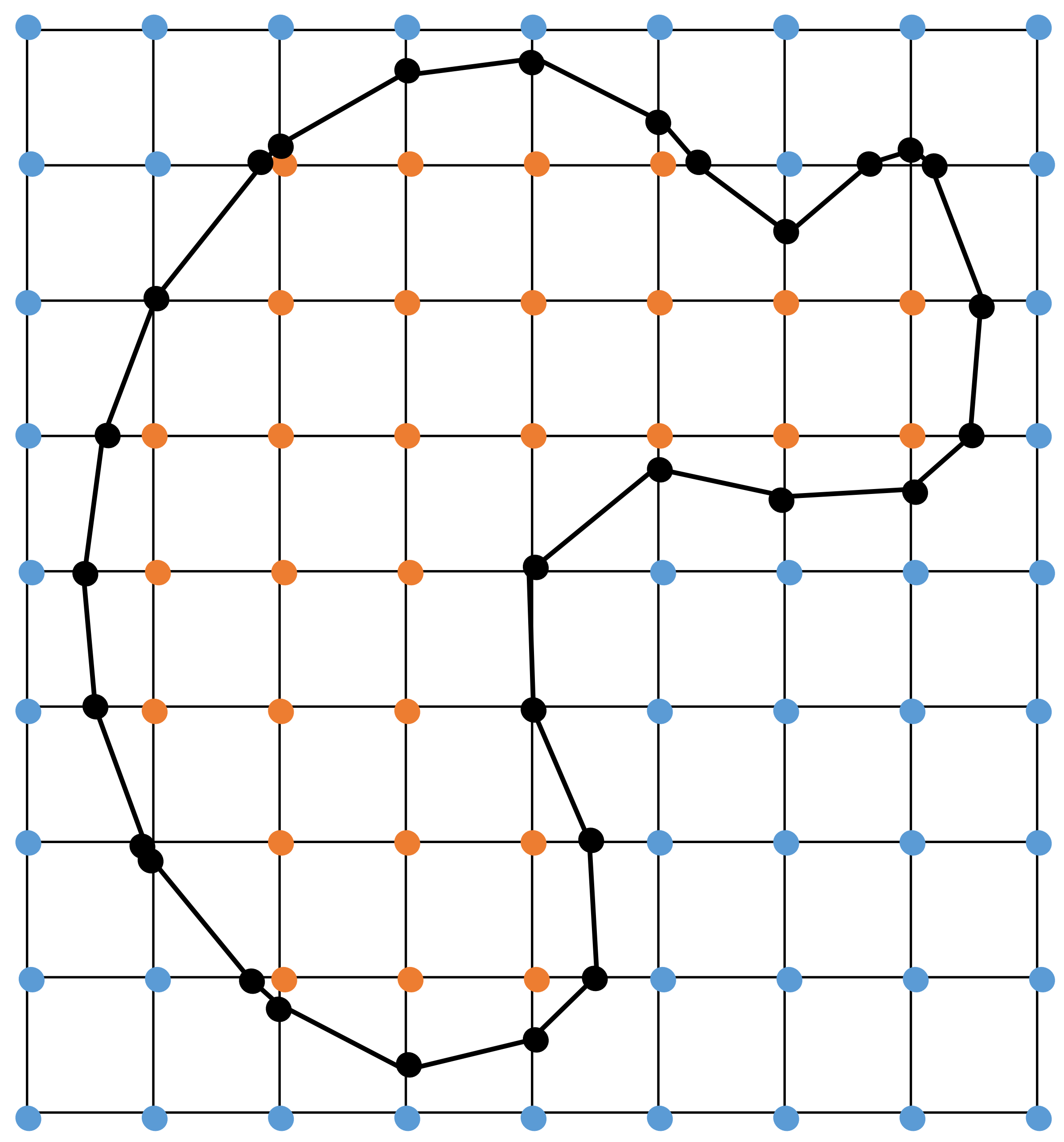
- Can‘t explicitly compute all the roots
  - Sampling the level set is difficult (root finding)
- Solution: find approximate roots by trapping the implicit surface in a grid (lattice)



$F(\mathbf{x}) < 0$

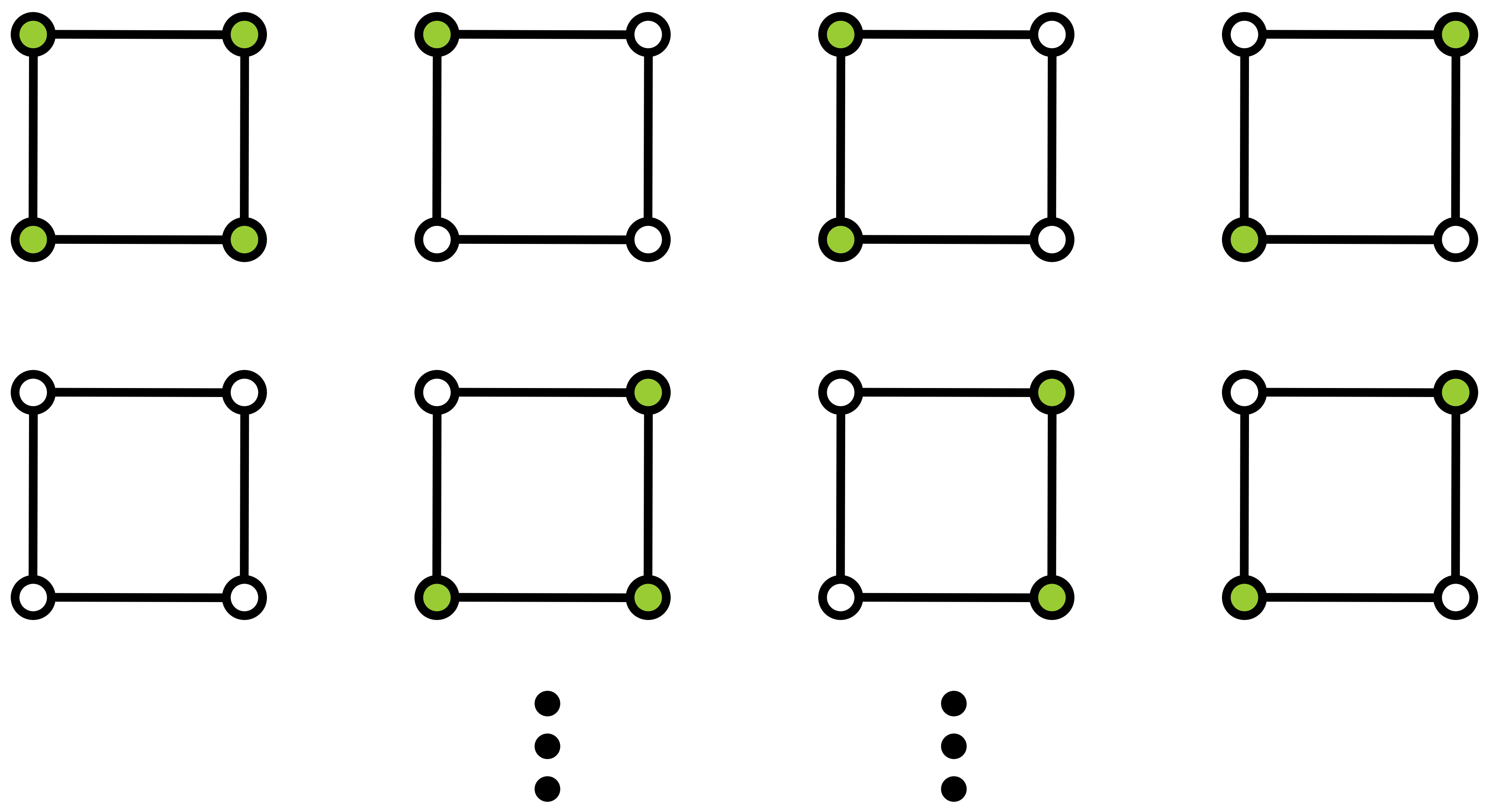
Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
3. Discretize function $F(x)$
4. **Extract zero Iso-surface**



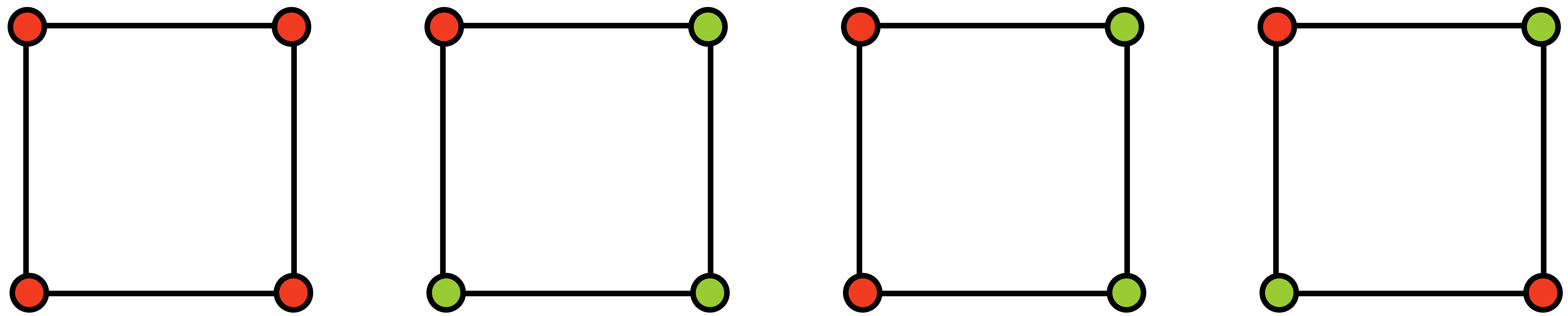
Marching Squares

- 16 different configurations in 2D



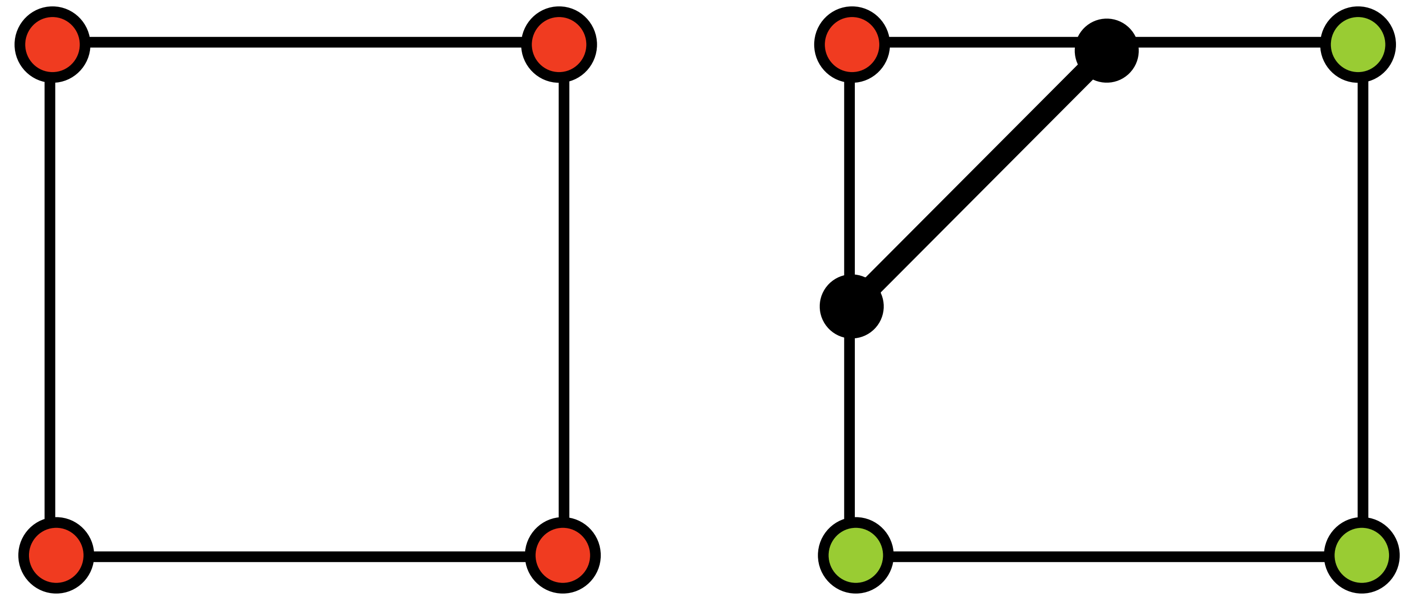
Tessellation in 2D

- 4 equivalence classes (up to rotational and reflection symmetry + complement)



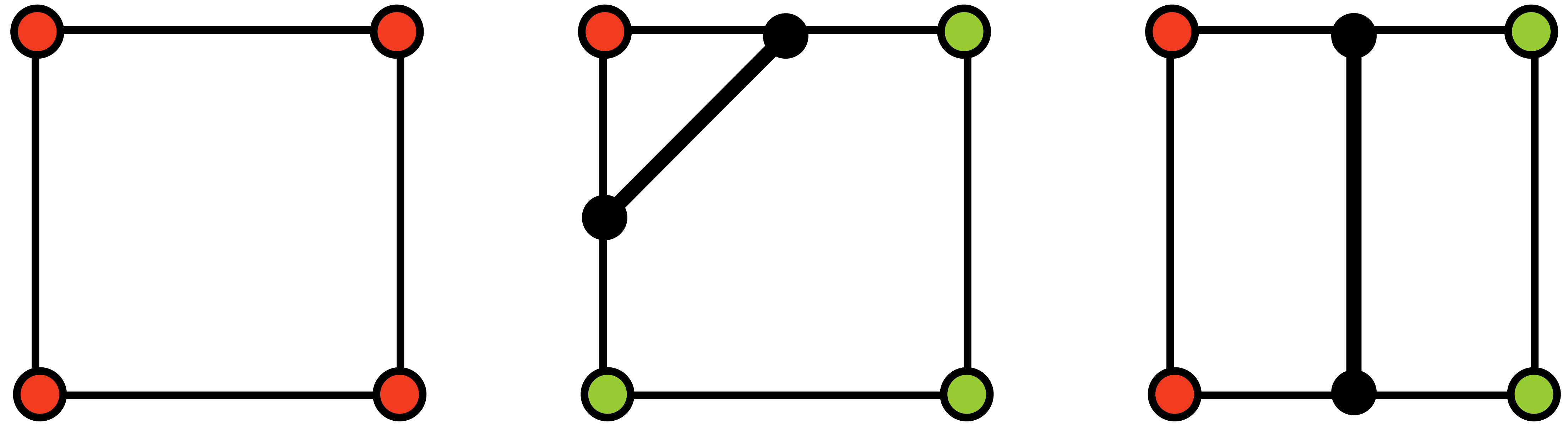
Tessellation in 2D

- 4 equivalence classes (up to rotational and reflection symmetry + complement)



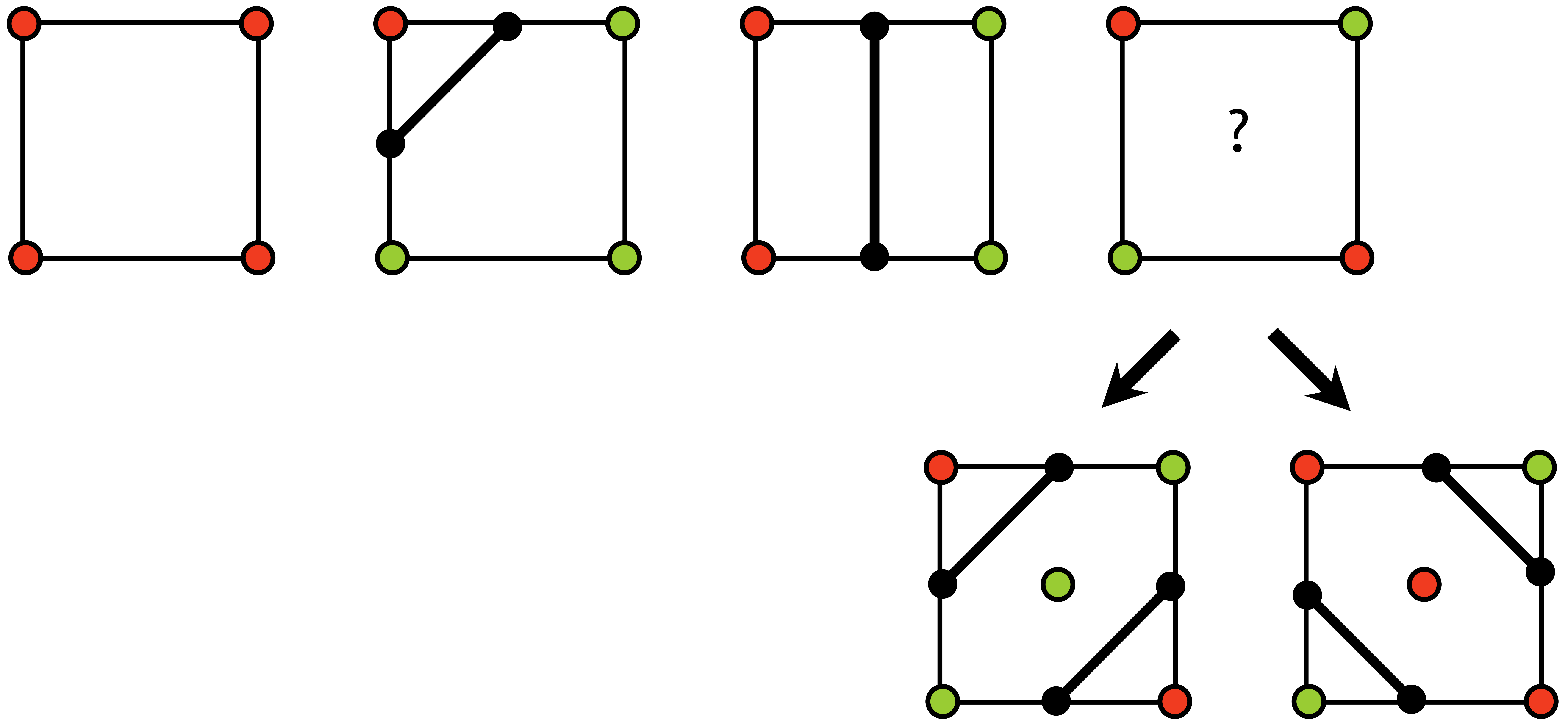
Tessellation in 2D

- 4 equivalence classes (up to rotational and reflection symmetry + complement)



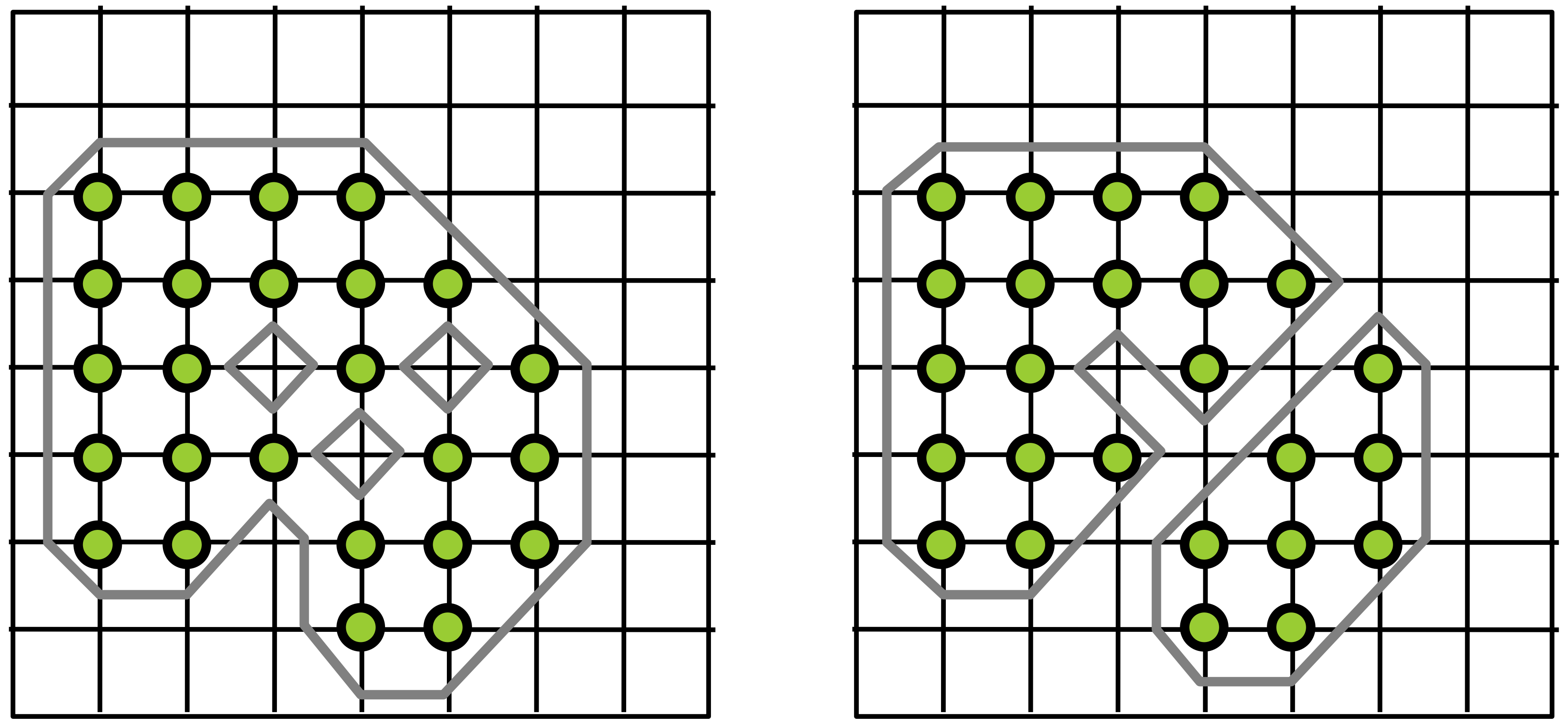
Tessellation in 2D

- 4 equivalence classes (up to rotational and reflection symmetry + complement)

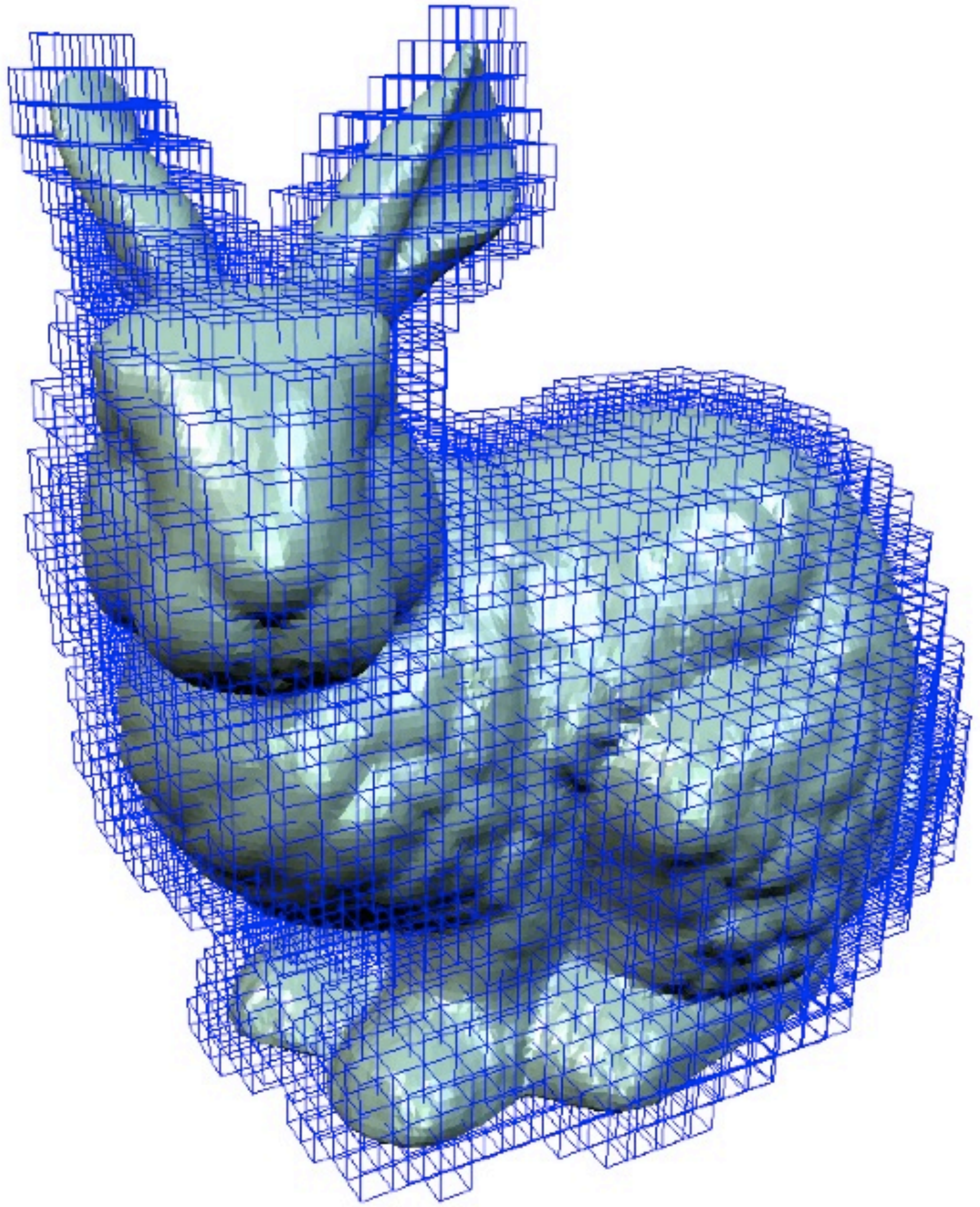


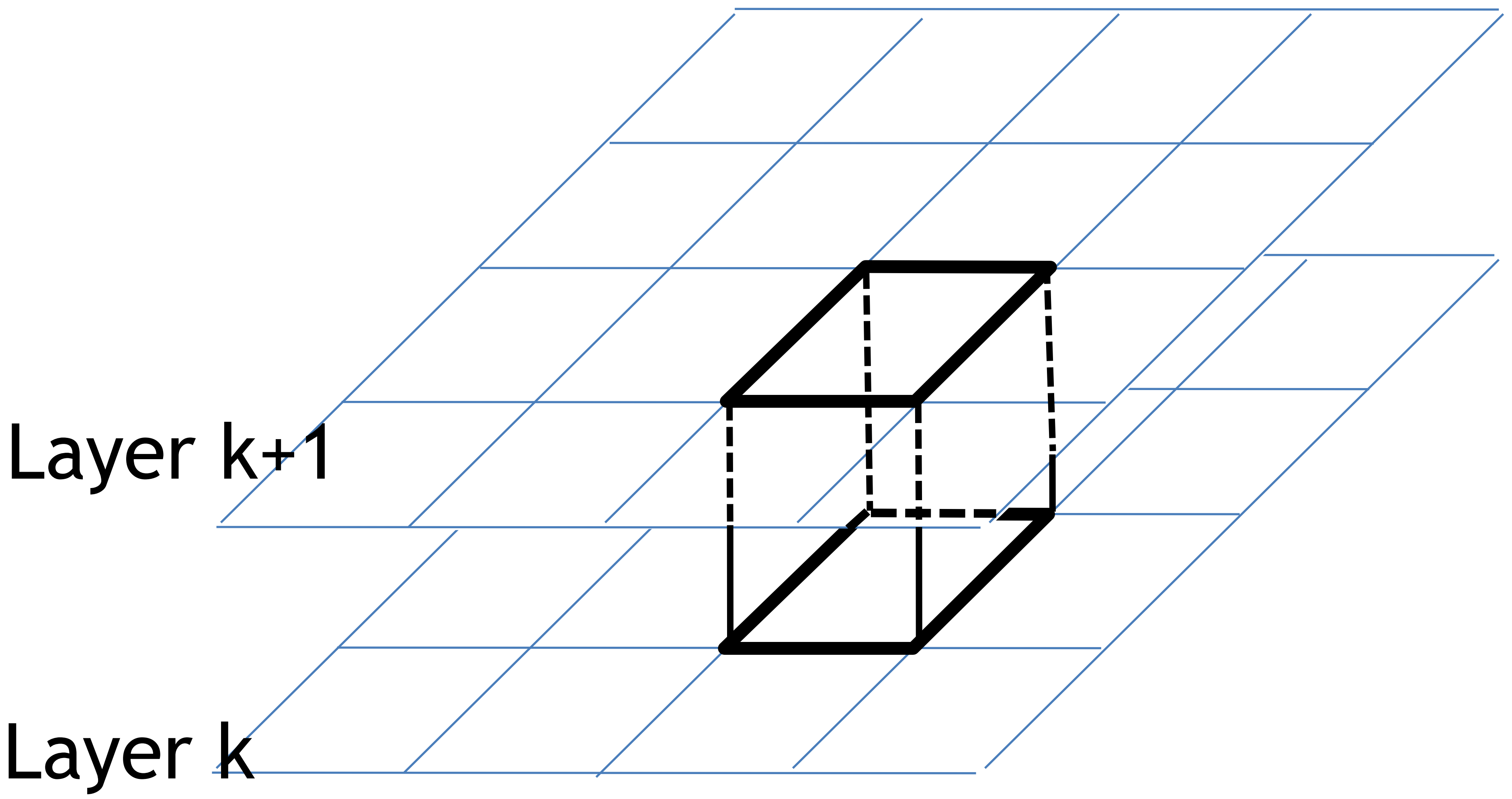
Tessellation in 2D

- Always pick consistently to avoid problems with the resulting mesh



3D: Marching Cubes

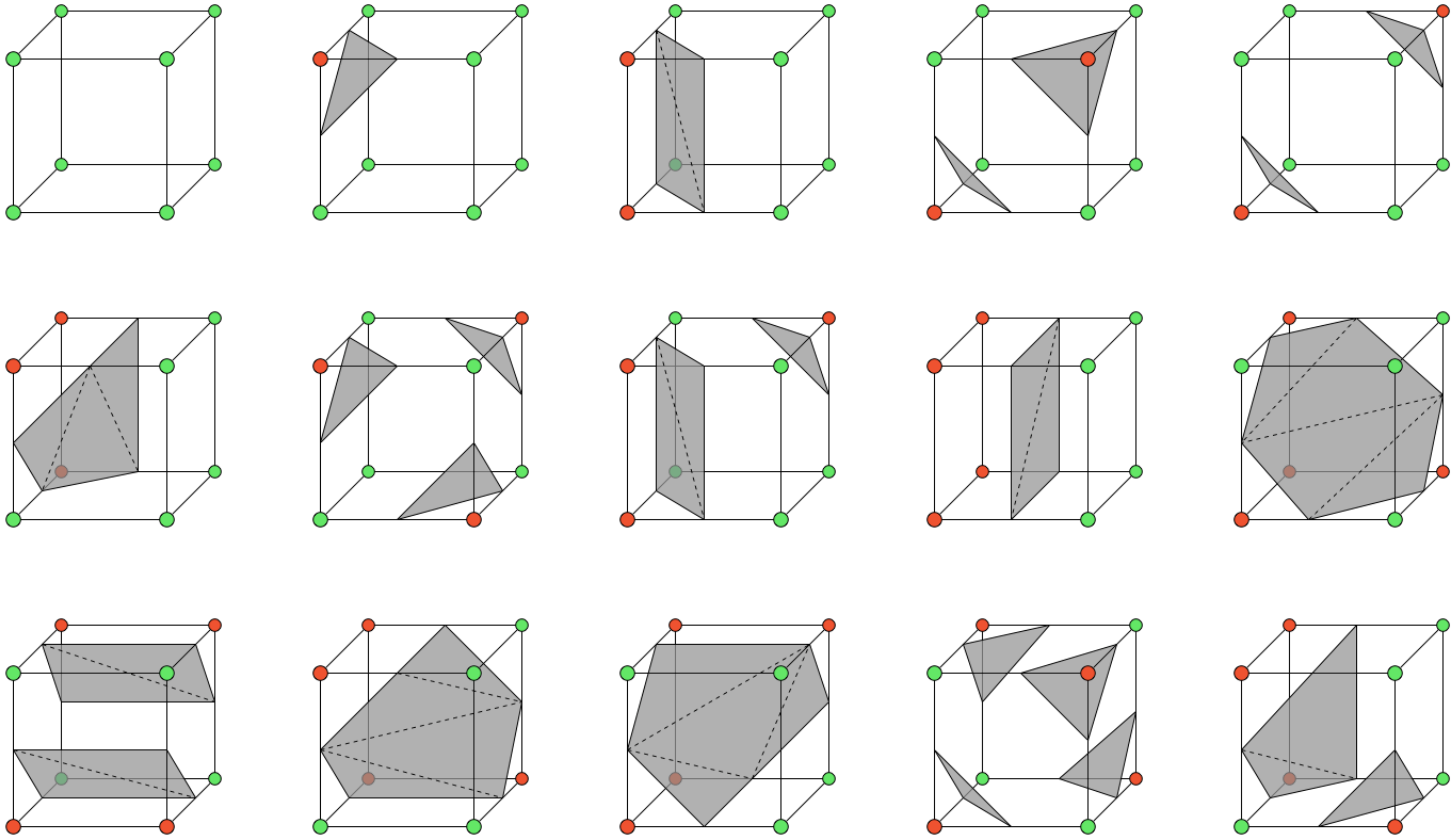




Marching cubes: A high resolution 3D surface construction algorithm, *Lorensen* et al., ACM SIGGRAPH 1987

Marching Cubes

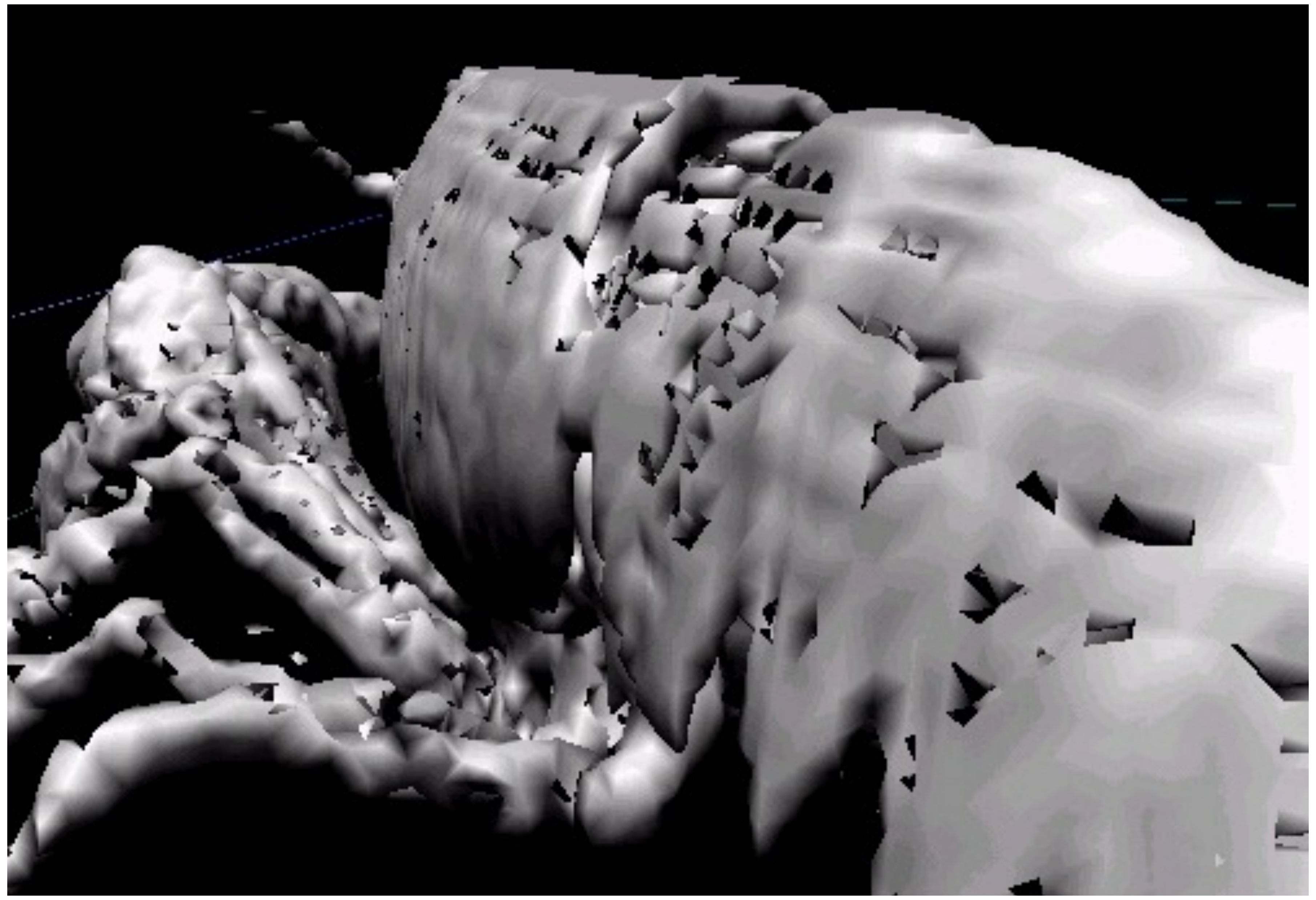
- 15 Unique cases (by rotation, reflection and complement)

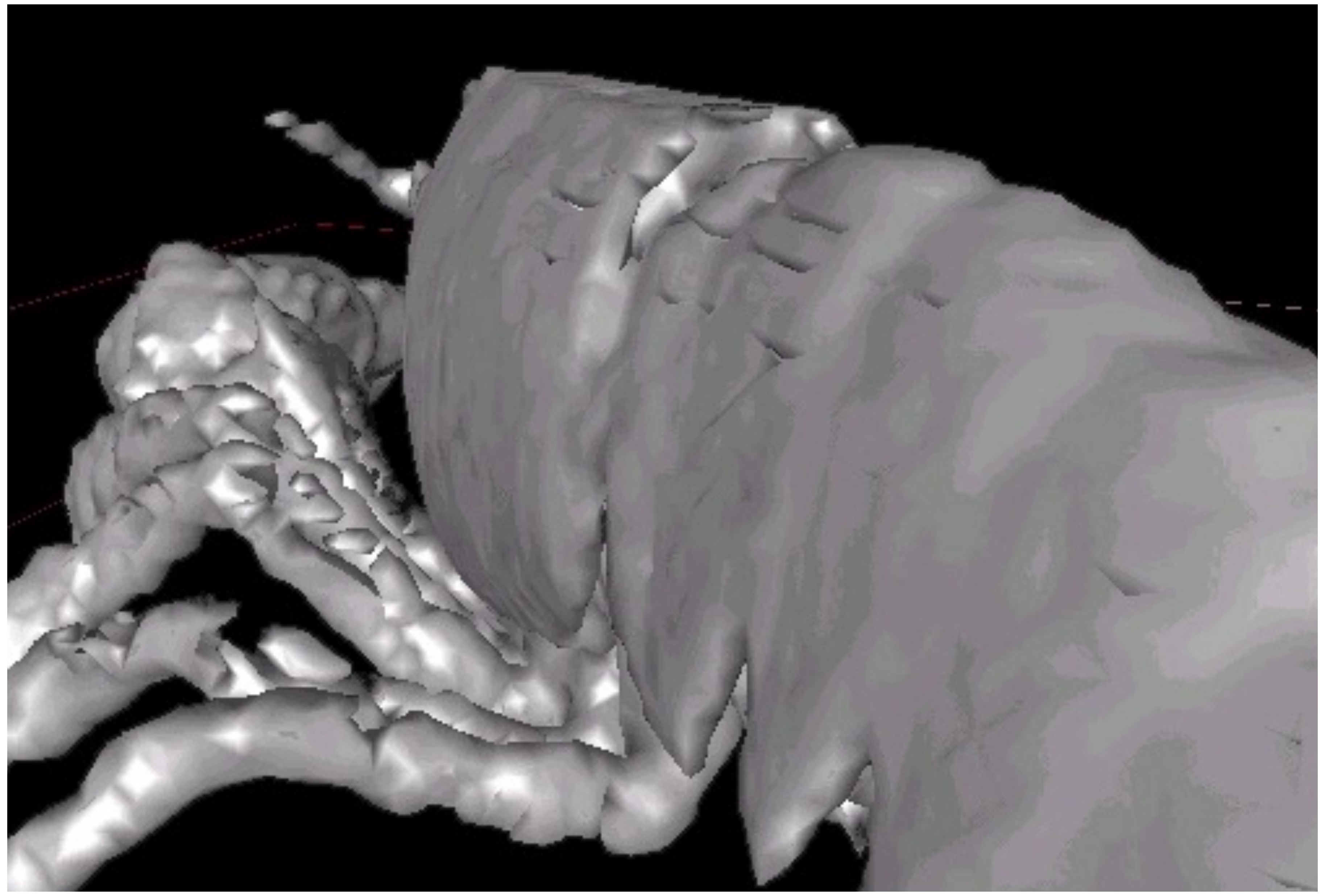


Marching cubes: A high resolution 3D surface construction algorithm, *Lorensen* et al., ACM SIGGRAPH 1987

Marching Cubes – Problems

- Have to make consistent choices for neighboring cubes – otherwise get holes

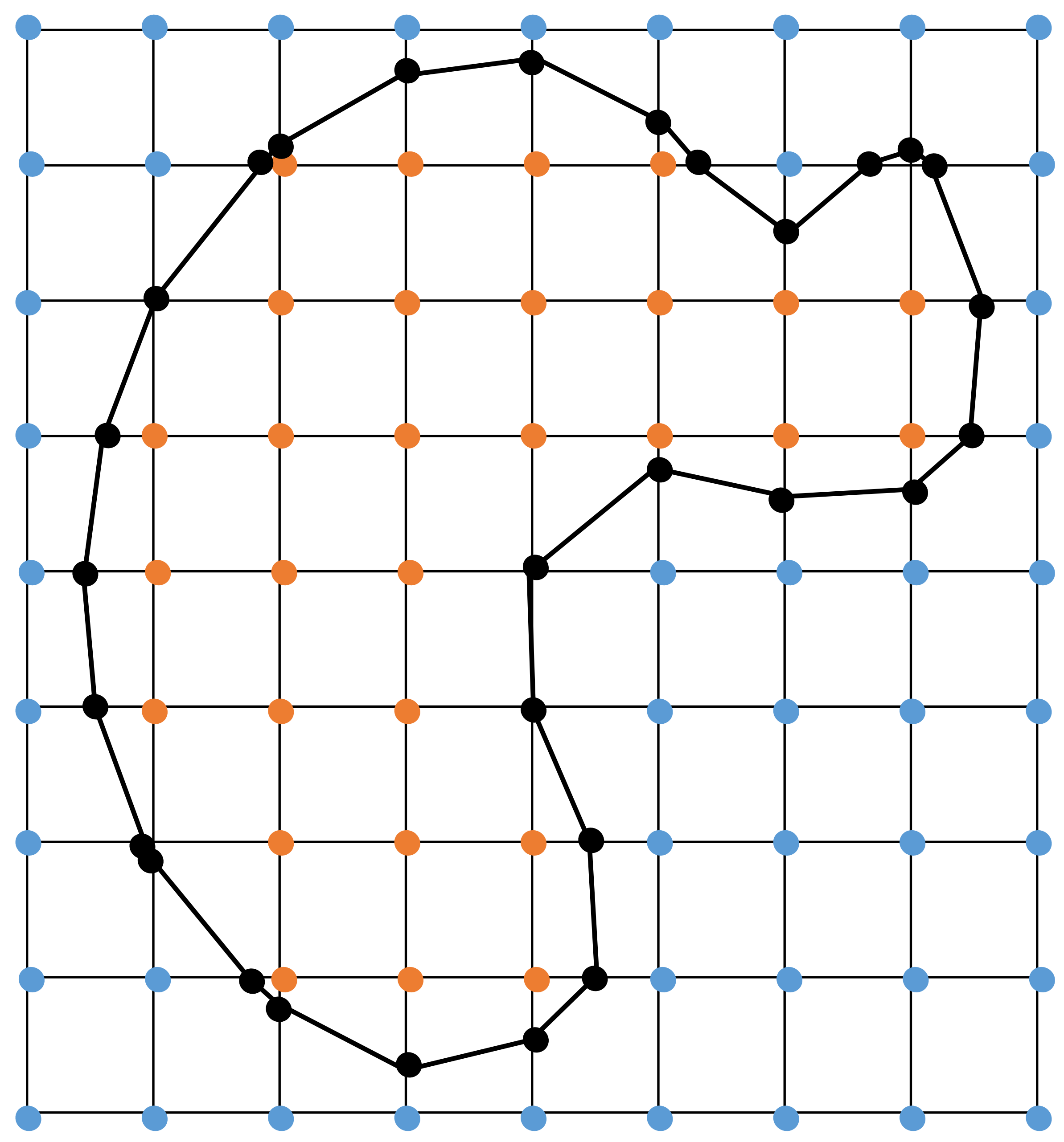




Marching cubes: A high resolution 3D surface construction algorithm, *Lorensen* et al., ACM SIGGRAPH 1987

Implicit Reconstruction

1. Estimate normal (Optional)
2. Compute function $F(x)$
3. Discretize function $F(x)$
4. Extract zero Iso-surface



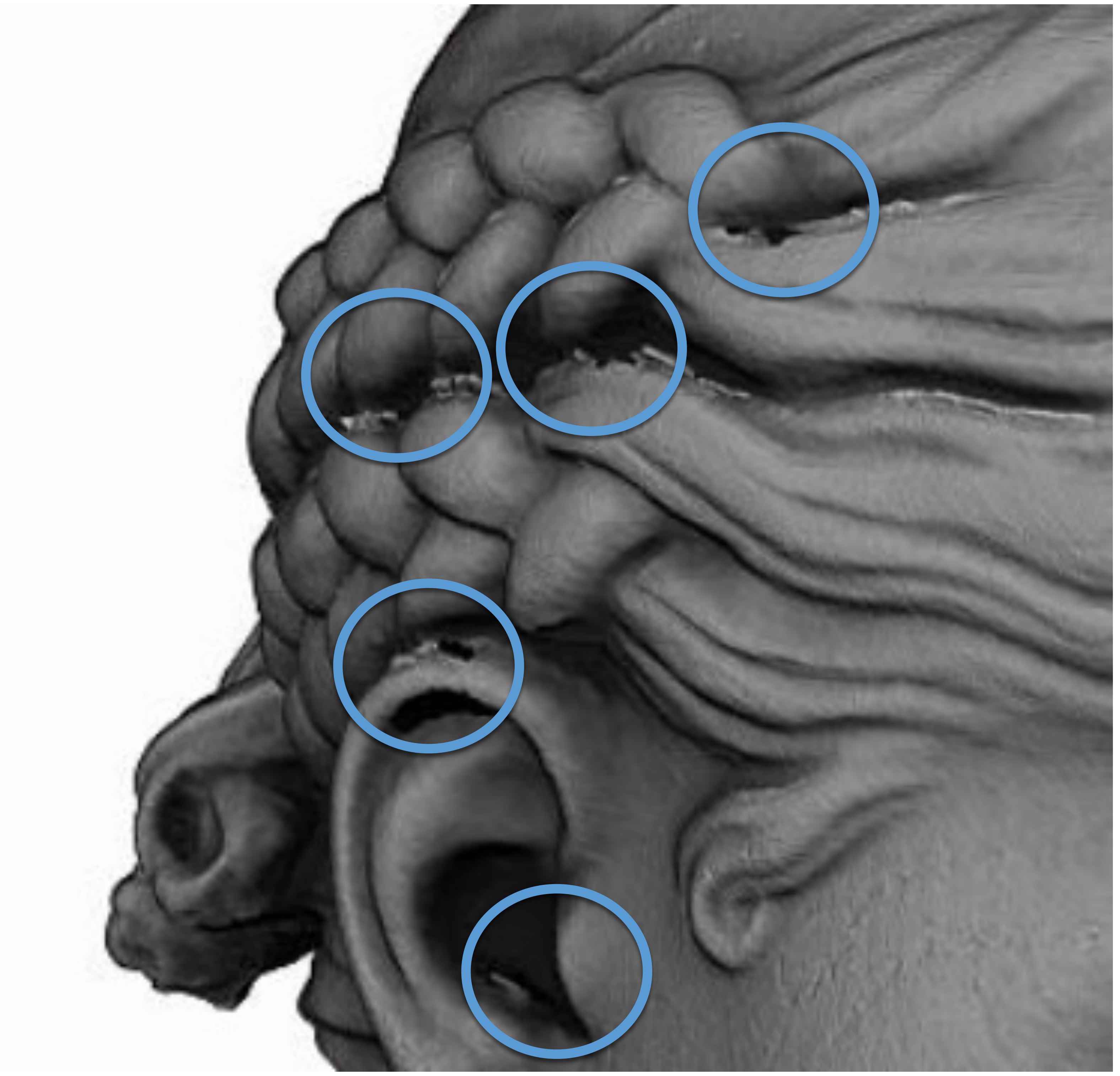
Homework

- Tasks:
  - Compute a function represented as a signed distance field of a 3D point cloud with given (but possibly unnormalized) normal.
  - Apply the marching cubes algorithm to extract a triangle mesh from the zero-level set of the function.
  - Experiment with various reconstruction parameters

Summary

- Approximate input data
- Surface is defined to be the 0-level set of a scalar-valued function $F(x)$

Implicit Reconstruction



Summary

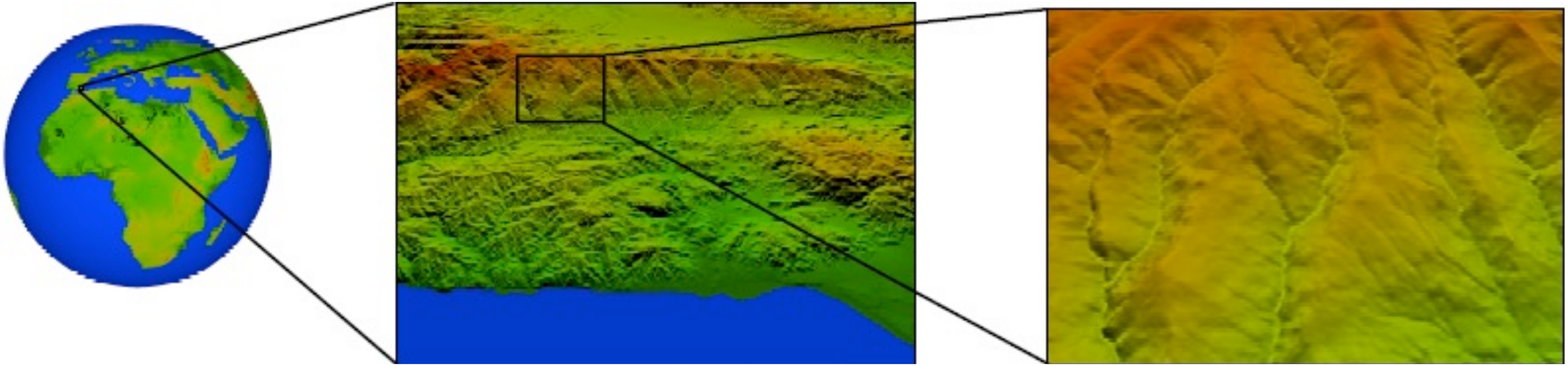
1. Estimate normal (Optional)
2. Compute function $F(x)$
3. Discretize function $F(x)$
4. Extract zero Iso-surface

More Assumptions

More Assumptions: Envelope

M. Kazhdan, Chuang, Rusinkiewicz, and H. Hoppe. Poisson Surface Reconstruction with Boundary Constraints. *Symposium on Geometry Processing* (**2020**)

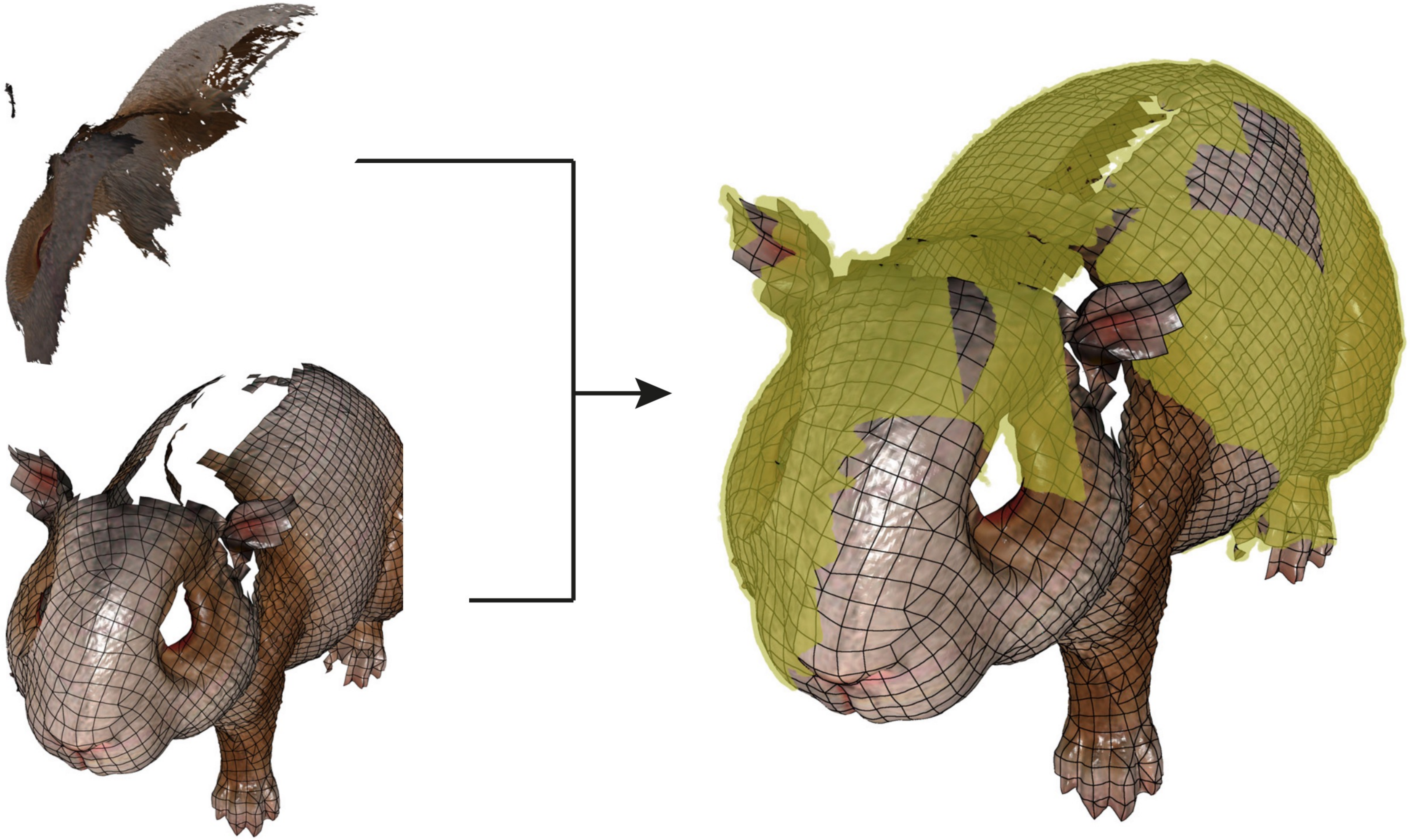
More Assumptions: Out of Core



USGS *Earth*: $2.2\text{x}10^{10}$ Points

M. Bolitho, M. Kazhdan, R. Burns, and H. Hoppe. Multilevel Streaming for Out-of-Core Surface Reconstruction. *Symposium on Geometry Processing* (**2007**)

More Assumptions: Real-time

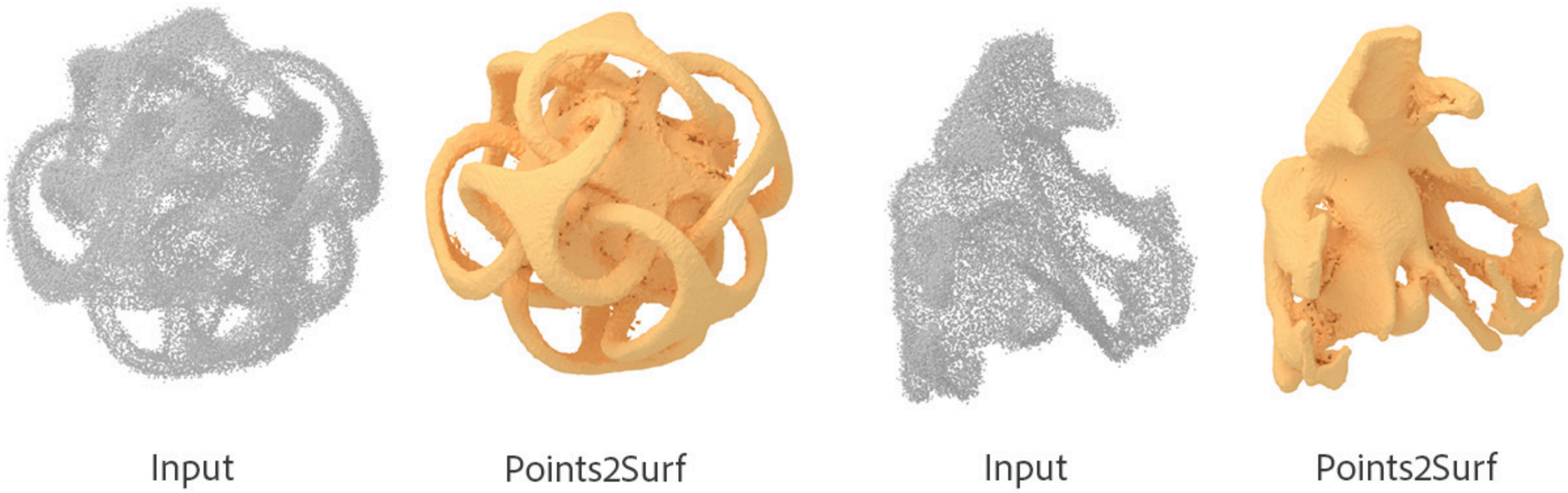


Nico Schertler, Marco Tarini, Wenzel Jakob, Misha Kazhdan, Stefan Gumhold, Daniele Panozzo.
Field-Aligned Online Surface Reconstruction.
ACM Transactions on Graphics (SIGGRAPH), 2017

More Assumptions: Topology

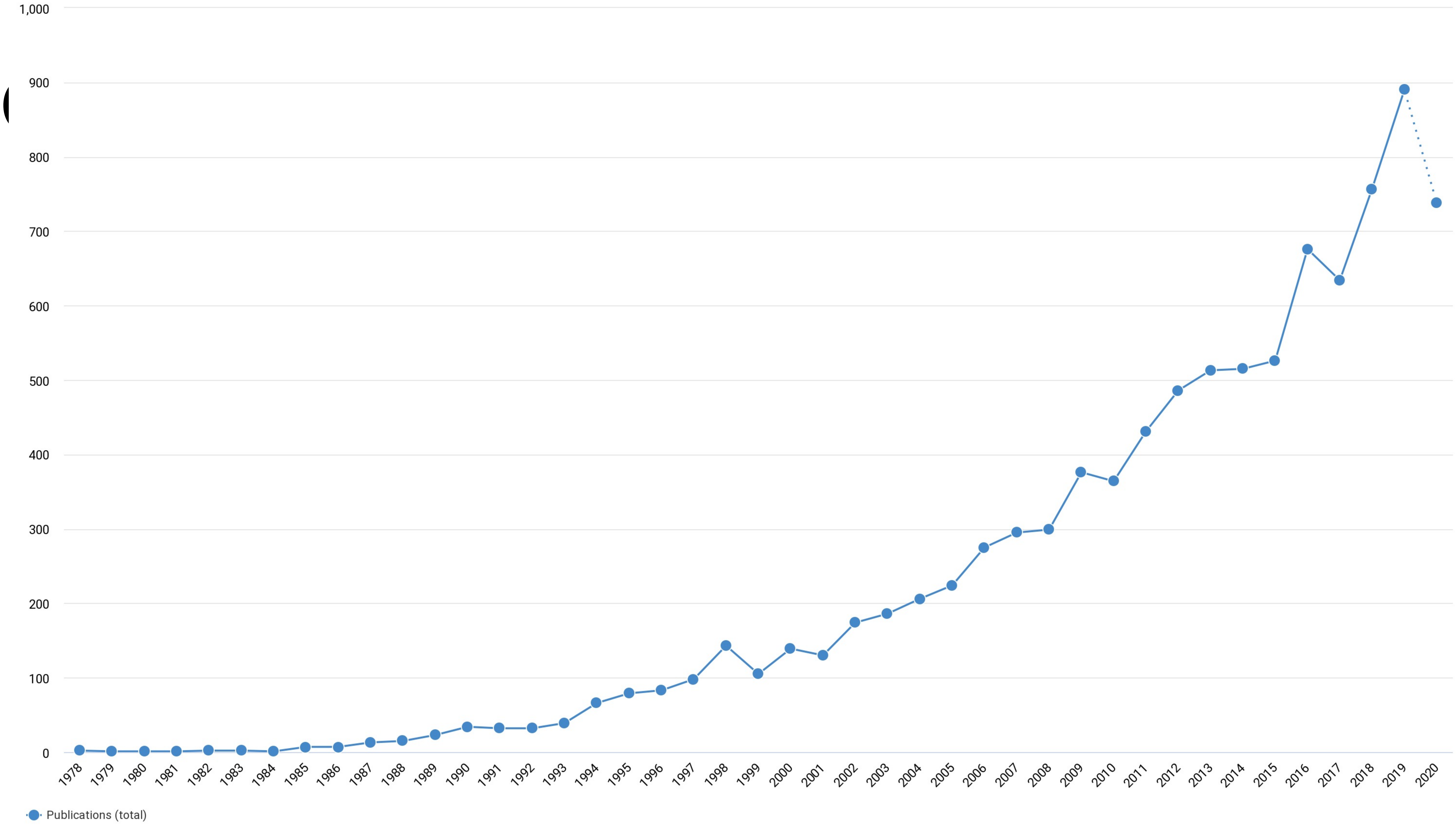
Roee Lazar, Nadav Dym, Yam Kushinksy, Zhiyang Huang, Tao Ju, Yaron Lipman.
Robust Optimization for Topological Surface Reconstruction.
ACM Transactions on Graphics (Proc. ACM Siggraph 2018)

More Assumptions: Generalization



Philipp Erler, Paul Guerrero Stefan Ohrhallinger, Niloy J. Mitra, Michael Wimmer.
Points2Surf: Learning Implicit Surfaces from Point Cloud Patches. ECCV 2020.

Publications in each year. (Criteria: see below)



Source: https://app.dimensions.ai
Exported: October 07, 2020
Criteria: Text - '3D surface reconstruction' in title and abstract.

Thank you!