UC San Diego

Machine Learning meets Geometry

L18: Surface Reconstruction

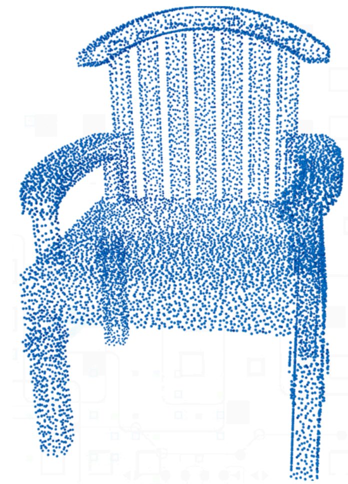
Hao Su

Ack: Minghua Liu helped to prepare slides

Surface Reconstruction

- Explicit Algorithms
- Implicit Algorithms

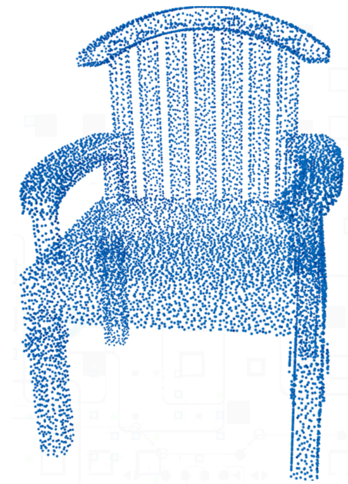
Surface Reconstruction Task



- Input: point cloud (with or without normals)
- Output: triangle mesh

Two Basic Families

- Explicit algorithms
  - Directly connect the input points with triangles, e.g.,
    - ball-pivoting algorithm
    - extrinsic-intrinsic ratio algorithm



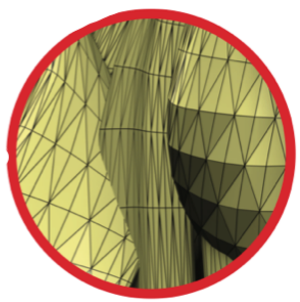
- Implicit algorithms
  - Approximate the input points by implicit field functions $S = \{x : F(x) = 0\}$
  - Then extract iso-surfaces, e.g.,
    - poisson surface reconstruction
    - reconstruction with RBF

Some Desired Properties of the Algorithm

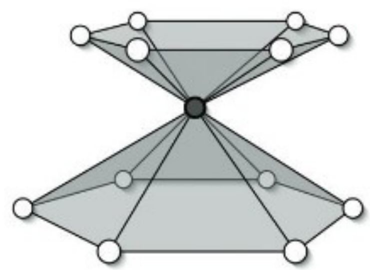
- Fast: The input point cloud may be large. We expect the computation to be fast.
- Robust: May recover the underlying surface structure even when the input point cloud is noisy
- Output mesh is desired to satisfy some geometric constraints

Geometric Constraint: Manifold

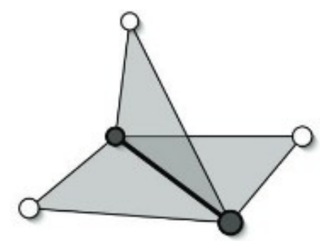
- A mesh is **manifold** if it does not contain:
  - self intersection
  - non-manifold edge (has more than 2 incident faces)
  - non-manifold vertex (one-ring neighborhood is not connected after removing the vertex)



self intersection



non-manifold vertex

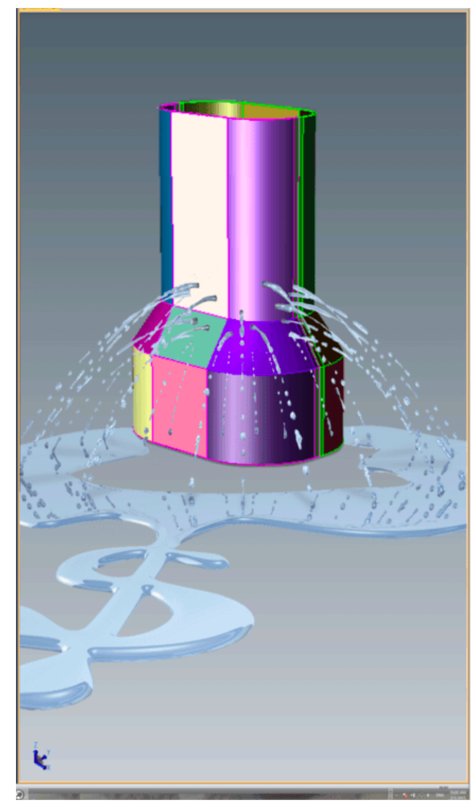


non-manifold edge

- A useful property for many subsequent geometry processing pipelines
  - e.g., to add texture maps and …

Geometric Constraint: Watertight

- A **manifold** mesh is **watertight** if each edge has **exactly** two incident faces, i.e., no boundary edges.
- Defines the interial, hence the volume of a solid object
- Required by many physical-simulation algorithms:
  - Estimate mass from density
  - Collision between objects
  - Force simulation
  - …



https://transmagic.com/wp-content/uploads/2016/05/watertight-solid-3d-cad-models-transmagic.png

Surface Reconstruction

- Explicit Algorithms
  - Ball-Pivoting Algorithm
  - Extrinsic-Intrinsic Ratio Algorithm
- Implicit Algorithms

Ball-Pivoting Algorithm

- Input: a point cloud and a hyper-parameter ρ

http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

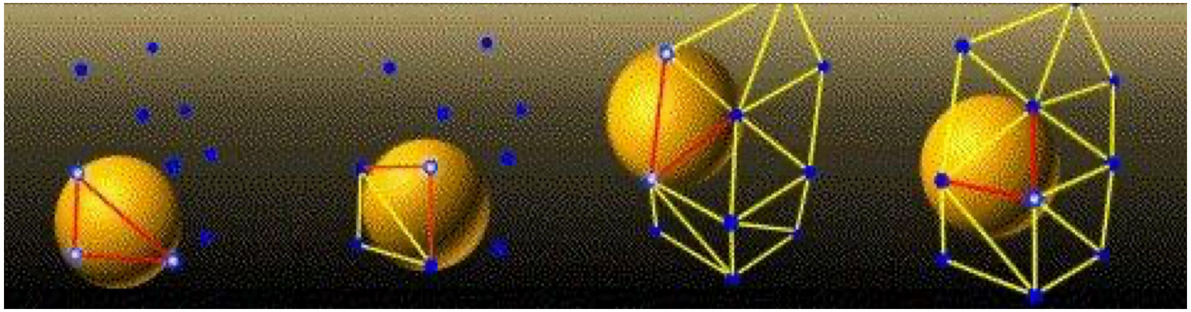
Ball-Pivoting Algorithm

- Input: a point cloud and a hyper-parameter ρ
- Assumption:
  - input points are dense enough that a ball of radius ρ cannot pass through the surface without touching the points.

http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Ball-Pivoting Algorithm

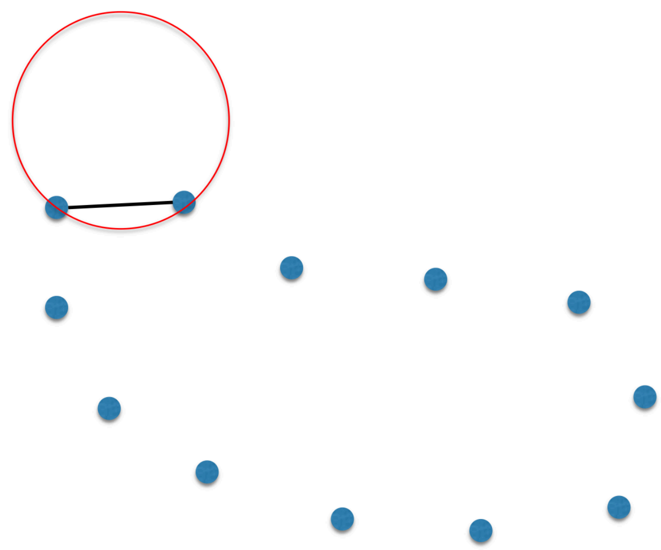
- Input: a point cloud and a hyper-parameter ρ
- Assumption:
  - input points are dense enough that a ball of radius ρ cannot pass through the surface without touching the points.
- Principle for face formation:
  - three points form a triangle if a ball of radius ρ touches them without containing any other points.



http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Ball-Pivoting Algorithm (2D)

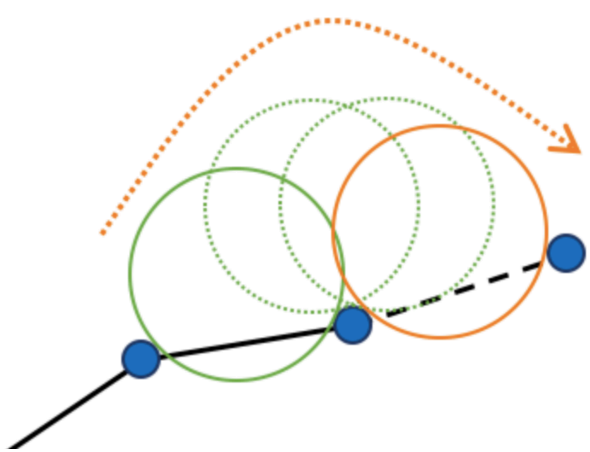
- Starting with a corner point and a ρ-ball
- Verify potential edges (triangles) in the ρ -neighborhood by the previous principle

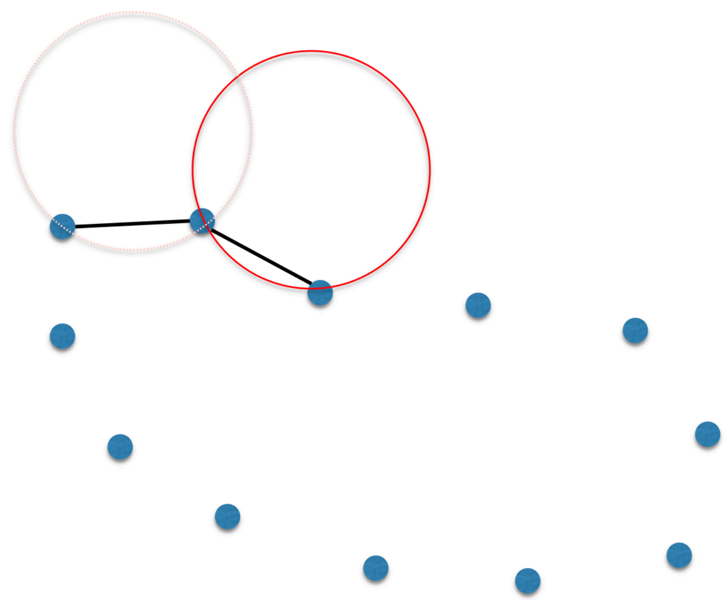


http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Ball-Pivoting Algorithm (2D)

- The ball pivots around an edge (triangles) until it touches another point, forming another triangle.

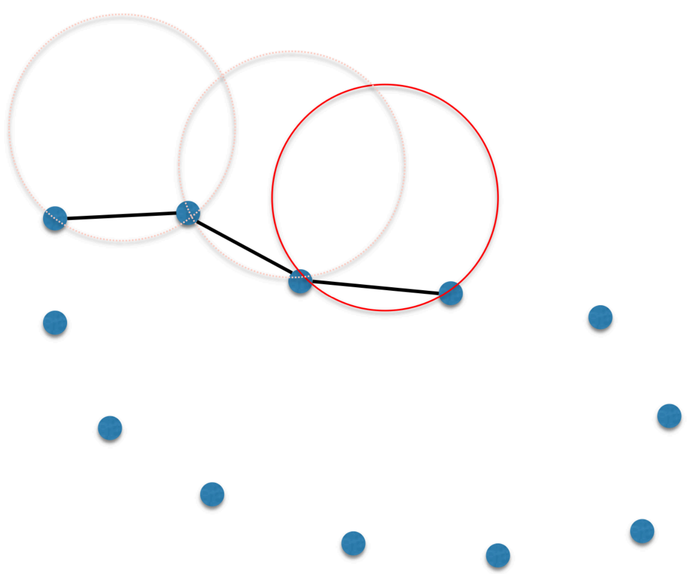


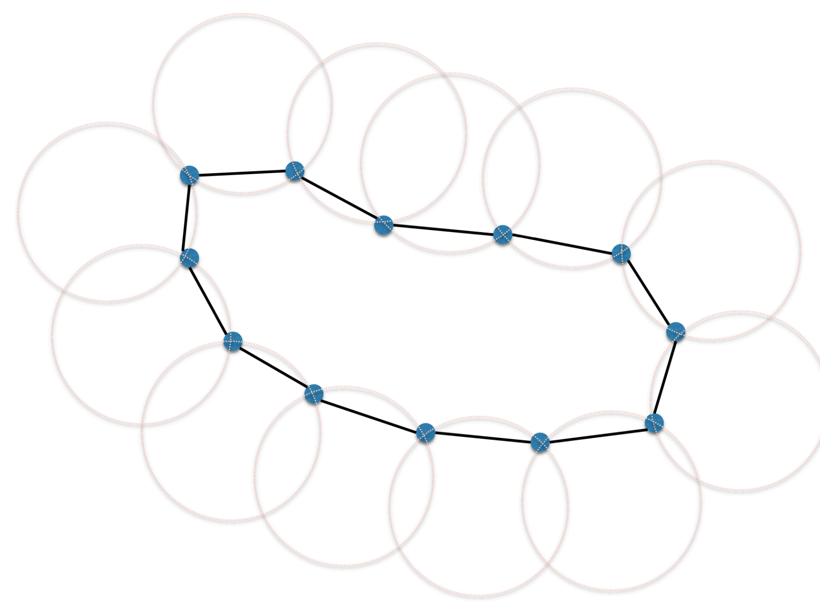


http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Ball-Pivoting Algorithm (2D)

- The process continues until all reachable edges have been tried
- Then starts from another seed triangle, until all points have been considered.

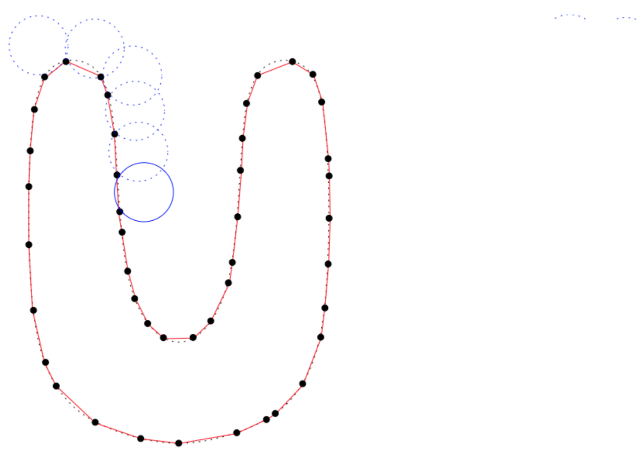




http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Radius ρ Matters

- Appropriate radius (a)

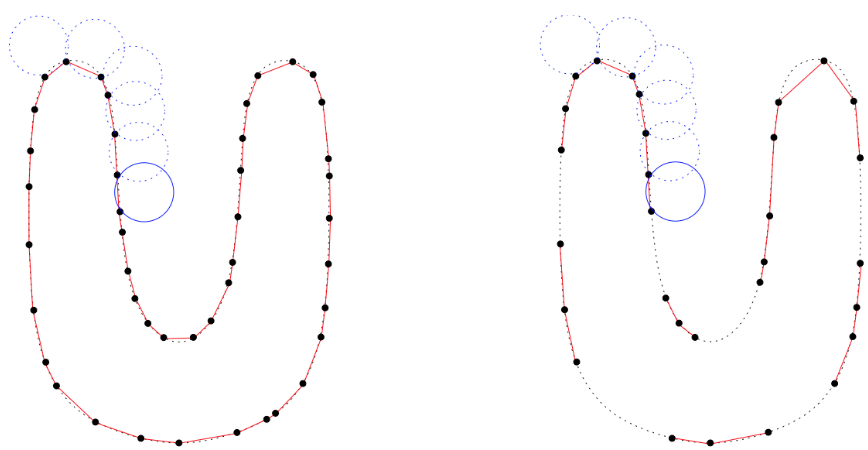


(a)

http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Radius ρ Matters

- Appropriate radius (a)
- Radius too small: some of the edges will not be created, leaving holes. (b)

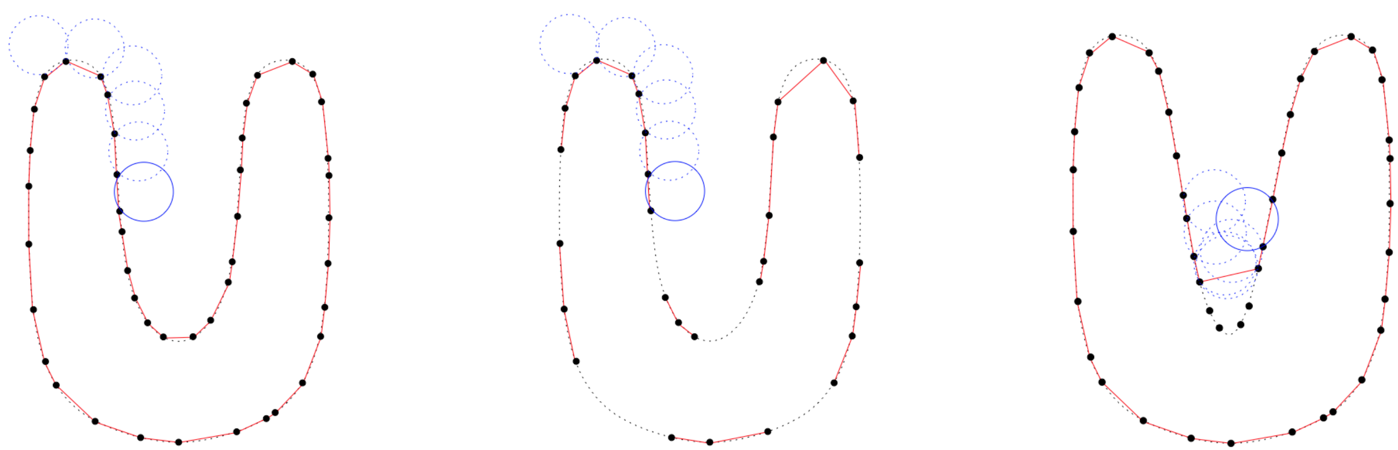


(a) (b)

http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Radius ρ Matters

- Appropriate radius (a)
- Radius too small: some of the edges will not be created, leaving holes. (b)
- Large radius: some of the points will not be reached (when the curvature of the manifold is larger than $1/\rho$) (c)

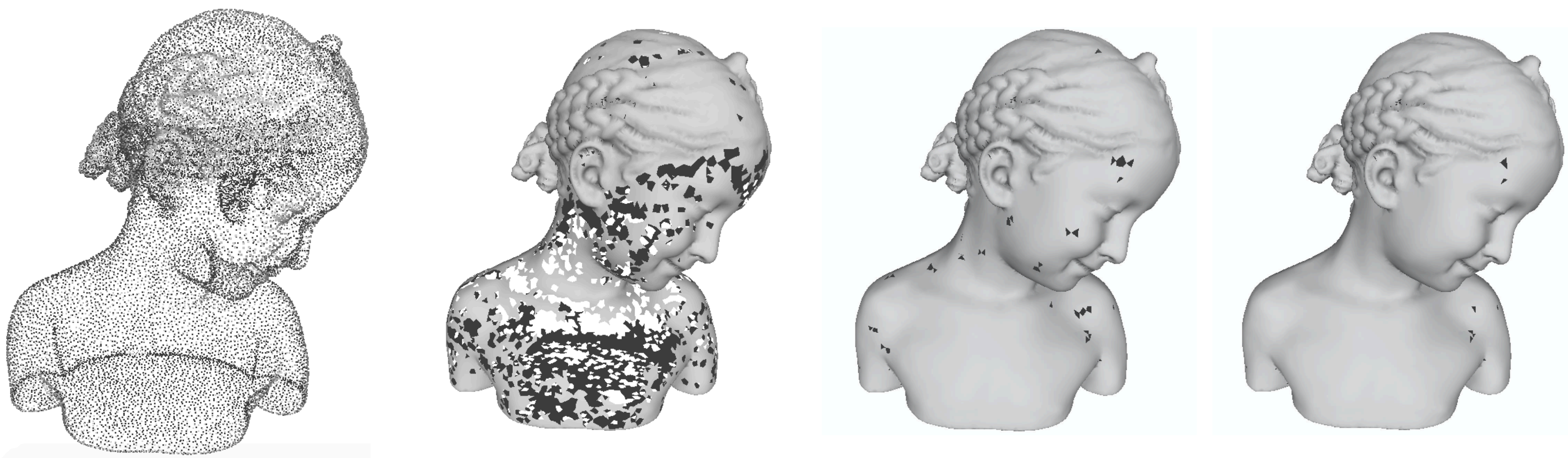


(a) (b) (c)

http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Iterative Approach

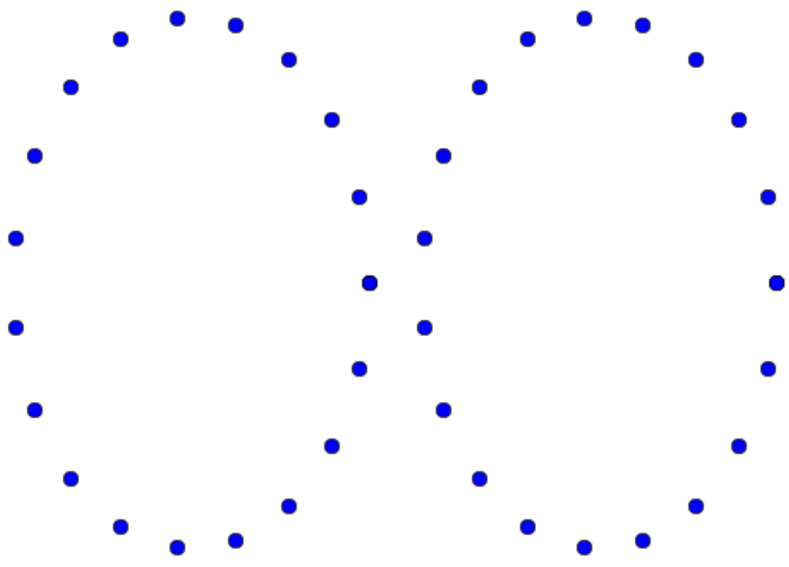
- Using multiple radius, iteratively connects the points.
- Small Radius capture high frequencies.
- Large Radius close holes.



http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Ambiguous Structures

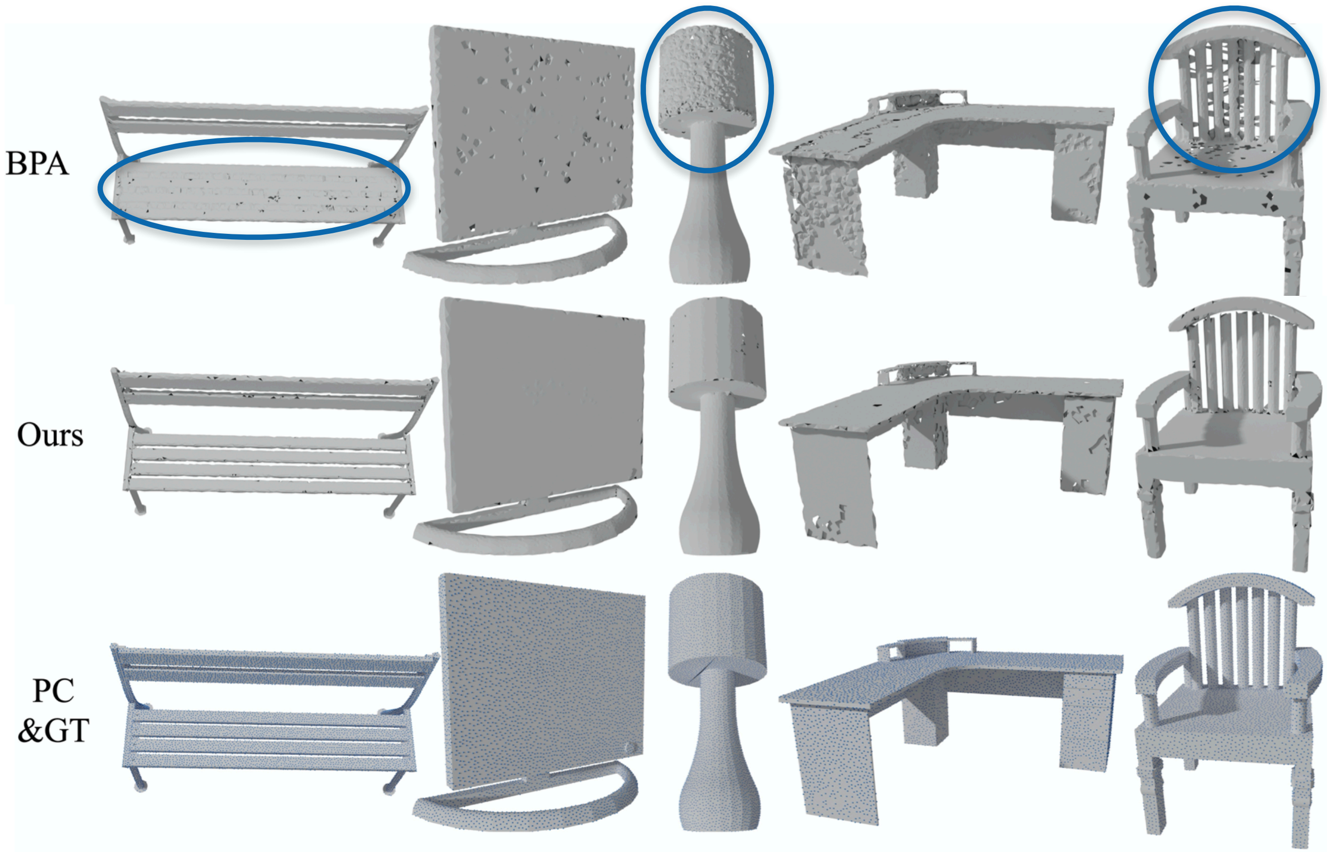
- Sometimes, defining a rule for structure estimation is hard
  - e.g., we tend to interpret the following point cloud as two disjoint ellipses; however, no ρ value allows us to separate them



Liu et al., “**Meshing Point Clouds with Predicted Intrinsic-Extrinsic Ratio Guidance.**”, *ECCV 2020*

Ambiguous Structures

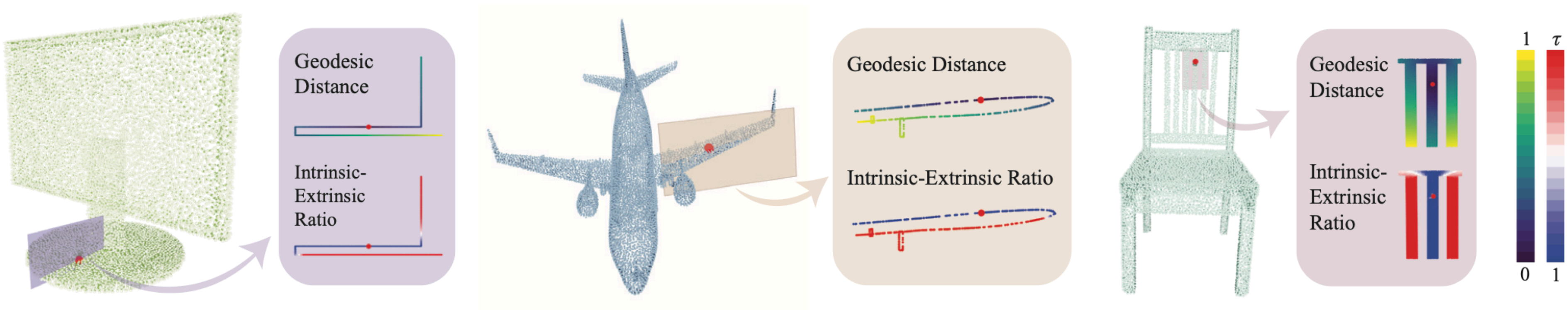
- Traditional Rule-based methods cannot handle ambiguous structures (e.g., thin structures & adjacent parts).



Liu et al., "**Meshing Point Clouds with Predicted Intrinsic-Extrinsic Ratio Guidance.**", *ECCV 2020*

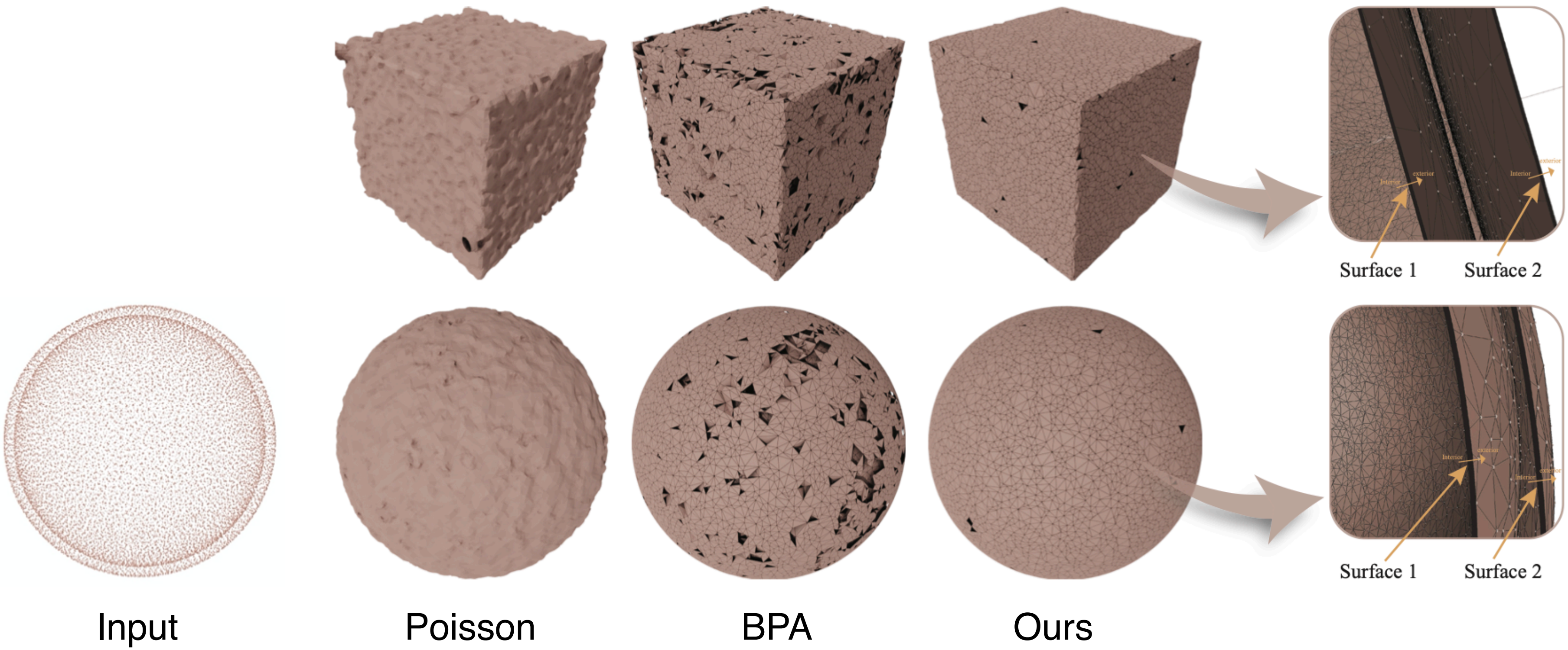
Review: Learning-Based Method

- Train a network to filter out incorrect connections.
- Utilize the Intrinsic-Extrinsic Ratio to guide the training.



Liu et al., "**Meshing Point Clouds with Predicted Intrinsic-Extrinsic Ratio Guidance.**", *ECCV 2020*

Ambiguous Structures



Liu et al., "**Meshing Point Clouds with Predicted Intrinsic-Extrinsic Ratio Guidance.**", *ECCV 2020*

Pros & Cons

- Pros:
  - Linear complexity (fast)
  - No dependence on normals

- Cons:
  - Can lead to non-manifold situations, and no water-tight guarantee

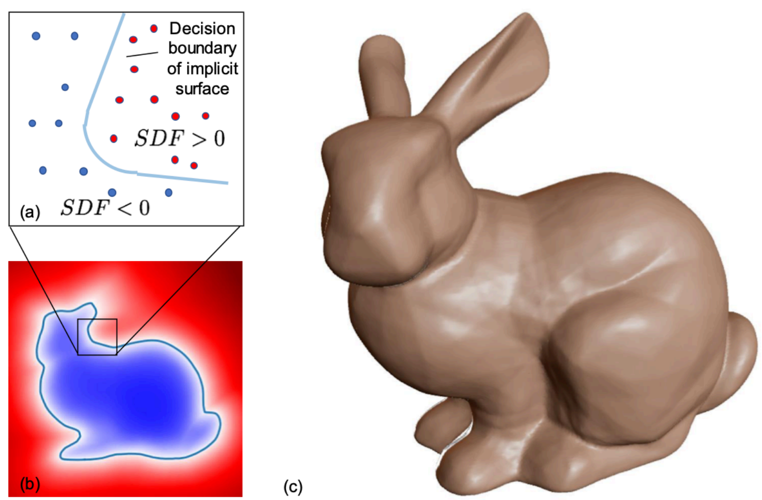
- Regarding robustness:
  - Learning can improve the robustness
  - However, current learning-based method would still not work when the sampling density is low

Surface Reconstruction

- Explicit Algorithms
- Implicit Algorithms
  - RBF implicit function estimation
  - Marching cube

Implicit Field Function

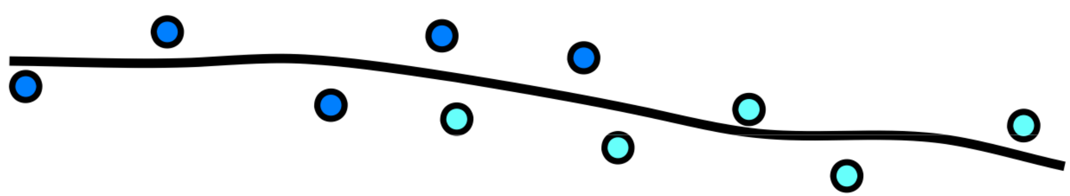
- Interior: $F(x, y, z) < 0$
- Exterior: $F(x, y, z) > 0$
- Surface: $F(x, y, z) = 0$ (zero set, zero iso-surface)
- Example implementation:
  - SDF: $F(x, y, z) =$ distance to the surface



Park et al., “**Deepsdf: Learning continuous signed distance functions for shape representation.**”, *CVPR 2019*

Implicit Meshing Algorithm

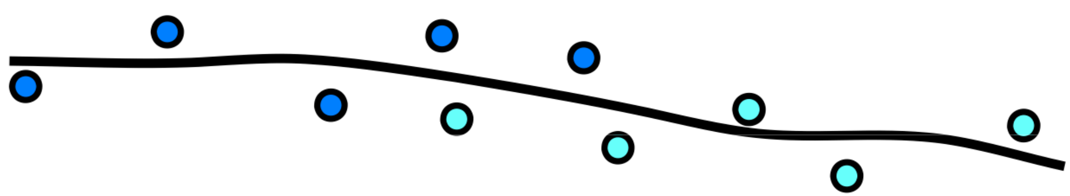
- Two basic steps:
  1. Estimate an implicit field function from data
  2. Extract the zero iso-surface



http://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/04_Surface_Reconstruction.pdf

Implicit Meshing Algorithm

- Two basic steps:
  1. **Estimate an implicit field function from data**
  2. Extract the zero iso-surface



http://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/04_Surface_Reconstruction.pdf

Radial Basis Functions

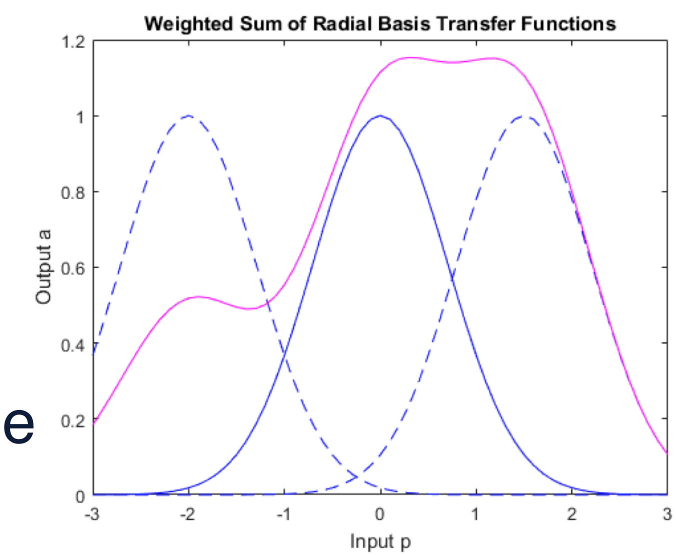
- Radial basis functions (RBF) $\phi_c(\mathbf{x})$: function value depends only on the distance from a center point c

- Use a weighted sum of radial basis functions to approximate the shape:

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

$$f(\mathbf{x}) = \sum_{i=1}^{n} \omega_i \phi\left(\|\mathbf{x} - \mathbf{x}_i\|\right) + p(x)$$

where p is a polynomial of low degree



http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Constraints: Avoiding Non-Trivial Solutions

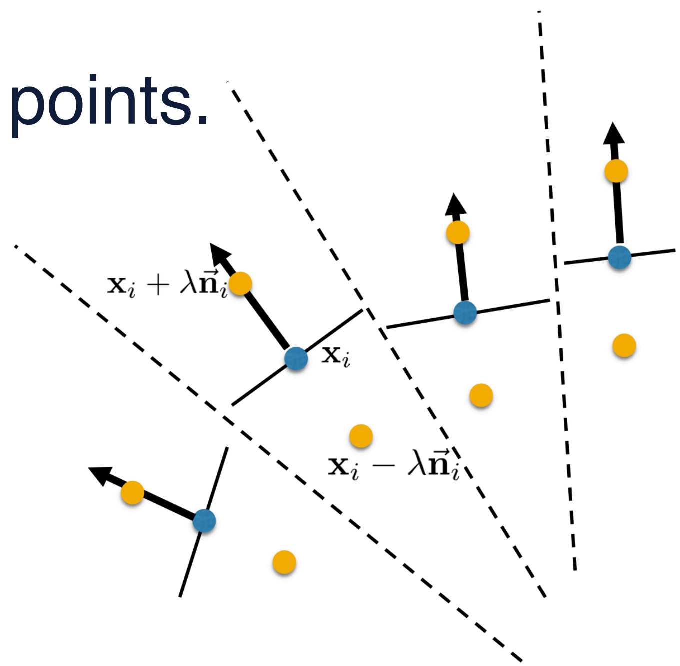
- Only force input points $\mathbf{x}_i$ to have zero value $f\left(\mathbf{x}_i\right) = 0$ is not enough, it may get the trivial solution $f(\mathbf{x}) \equiv 0$

- Use normal to add off-surface points.

$$f\left(\mathbf{x}_i\right) = 0$$

$$f\left(\mathbf{x}_i + \lambda \overrightarrow{\mathbf{n}}_i\right) = \lambda$$

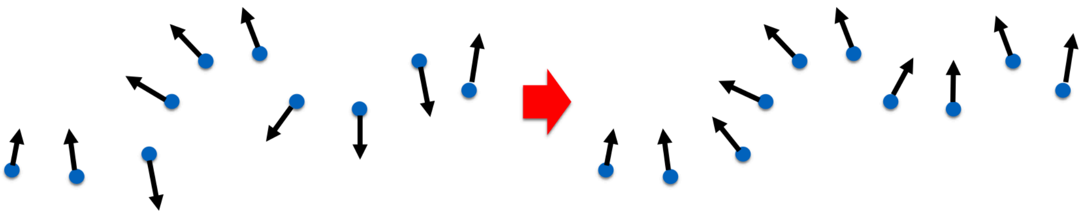
$$f\left(\mathbf{x}_i - \lambda \overrightarrow{\mathbf{n}}_i\right) = -\lambda$$

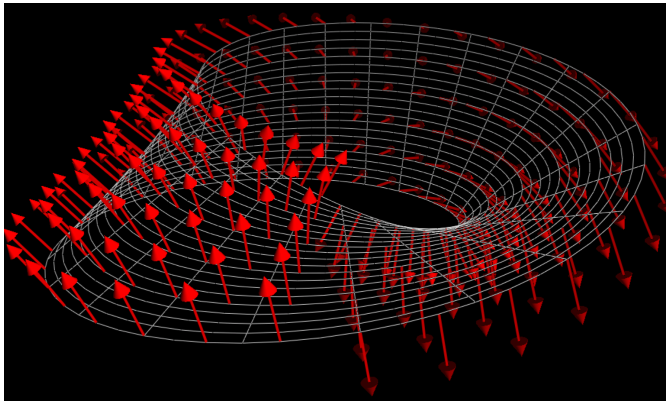


http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Consistent Normals are Required

- We just assumed consistent normals
- Normal is typically required to build watertight meshes
- However, obtaining consistent normal orientation is non-trivial (discussions deferred to later).





http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf
https://trecs.se/MöbiusStrip.php

Estimate Parameters

- Variables:
  - $n + l$ variables on ω_i (RBF coef.) and c_i (polynomial coef.)
- Solve a linear system of $3n + l$ equations
  - $3n$: from the point, inside, and outside
  - l: additional constraints to guarantee the smoothness and integrability of f

$$\begin{pmatrix} A & P \\ P^{\mathsf{T}} & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = B \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

$$A_{i,j} = \phi(|x_i - x_j|), \qquad i, j = 1, \ldots, N,$$
$$P_{i,j} = p_j(x_i), \qquad i = 1, \ldots, N, \quad j = 1, \ldots, \ell.$$

Implementation Details

- Triharmonic basis functions: $\phi(r) = r^3$
  - Need its extrapolation ability
  - Should **not** use RBF with compact or local support (e.g., Gaussian density)
- Polynomial: third-order is practically good

http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Implementation Details

- Do not need to use all the input data points as RBF centers
  - Use a greedy algorithm to select a subset of points
- Noisy data
  - Exact interpolation?
  - Treat the linear equation as solving a linear square problem and add a smoothness term

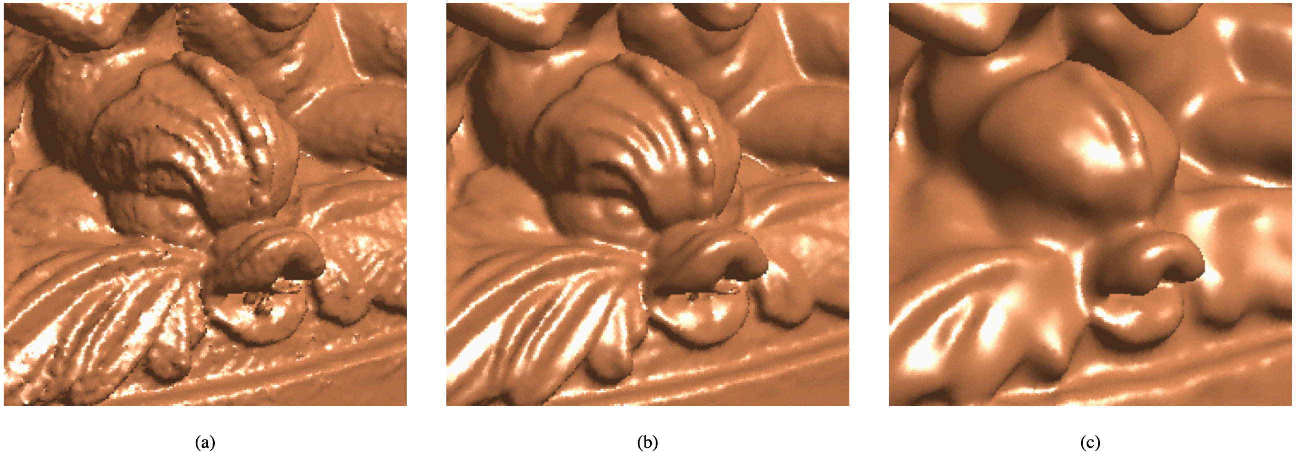


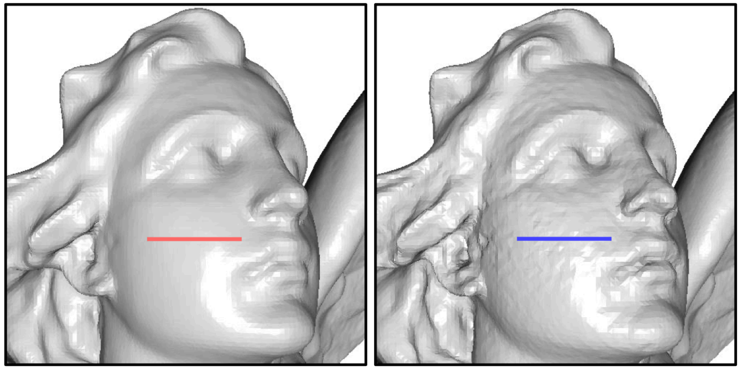
Figure 9: (a) Exact fit, (b) medium amount of smoothing applied (the RBF approximates at data points), (c) increased smoothing.

Carr, Jonathan C., et al. "**Reconstruction and representation of 3D objects with radial basis functions**." 2001

More than RBF

- Kazhdan M, Bolitho M, Hoppe H. “**Poisson surface reconstruction**.” ESGP, 2006.
    - Robust to noise, adapt to the sampling density
    - Over-smoothing

- Kazhdan M, Hoppe H. “**Screened poisson surface reconstruction**.” ToG, 2013.
    - Sharper reconstruction, faster
    - But it assumes clean data



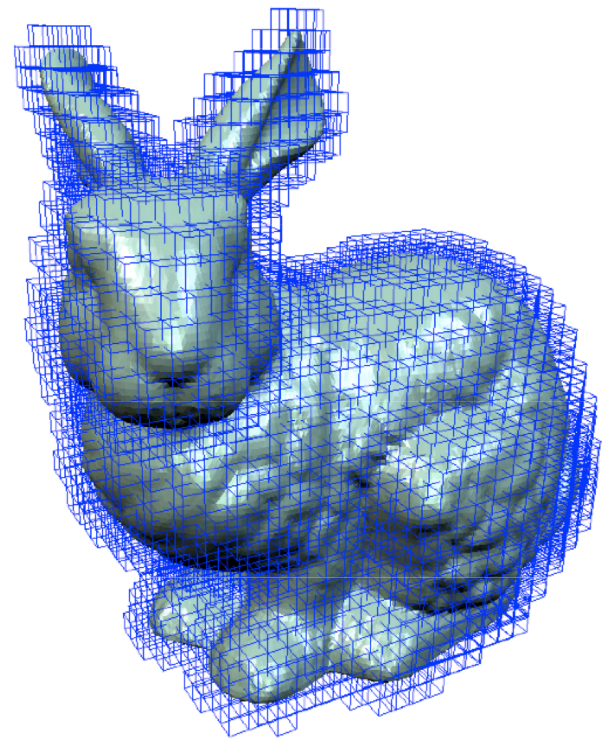
POISSON SCREENED POISSON

http://www.banterle.com/francesco/courses/2017/be_3drec/slides/lecture16.pdf

Implicit Meshing Algorithm

- Two basic steps:
  1. Estimate an implicit field function from data
  2. **Extract the zero iso-surface**

Input: a signed distance field (Implicitly assumed knowing the inside/outside of the shape, often needs to be estimated with **normal** information)



http://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/04_Surface_Reconstruction.pdf

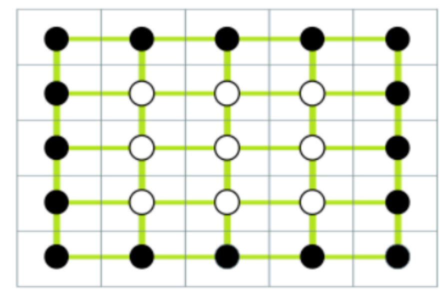
Classical Solution: 2D Marching Square

1	1	1	1	1
1	2	3	2	1
1	3	3	3	1
1	2	3	2	1
1	1	1	1	1

Threshold with iso-value ⇒

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

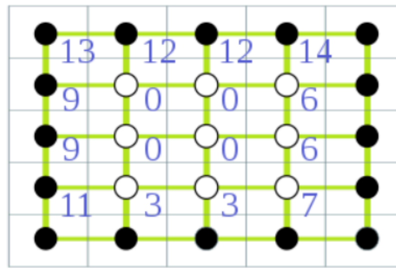
Binary image to cells ⇒



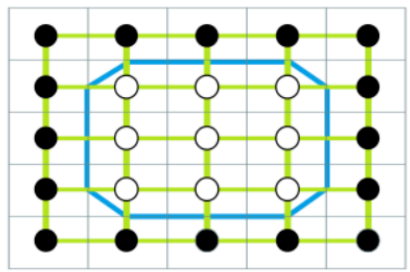
https://en.wikipedia.org/wiki/Marching_squares

Classical Solution:
2D Marching Square

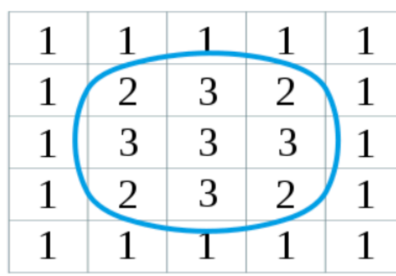
Give every cell a number based on which corners are true/false



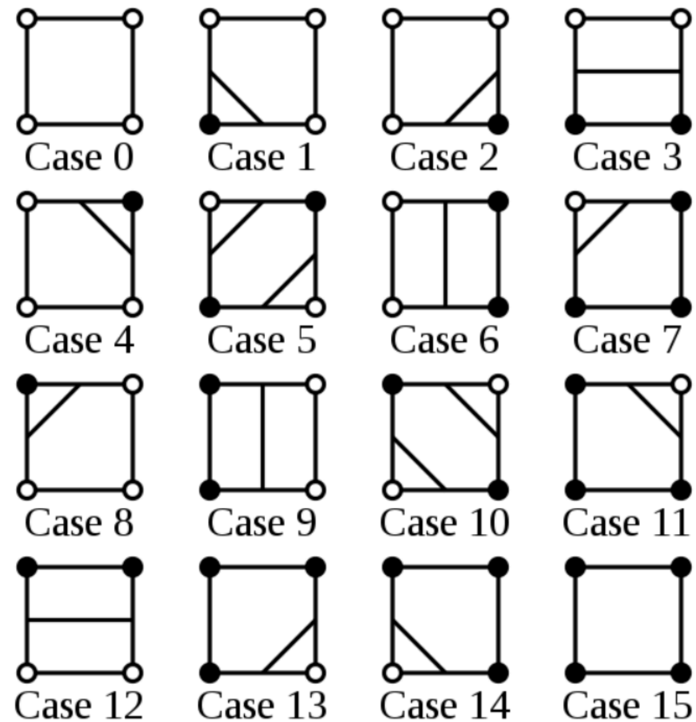
Look up the contour lines in the database and put them in the cells



Look at the original values and use linear interpolation to determine a more accurate position of all the line end-points



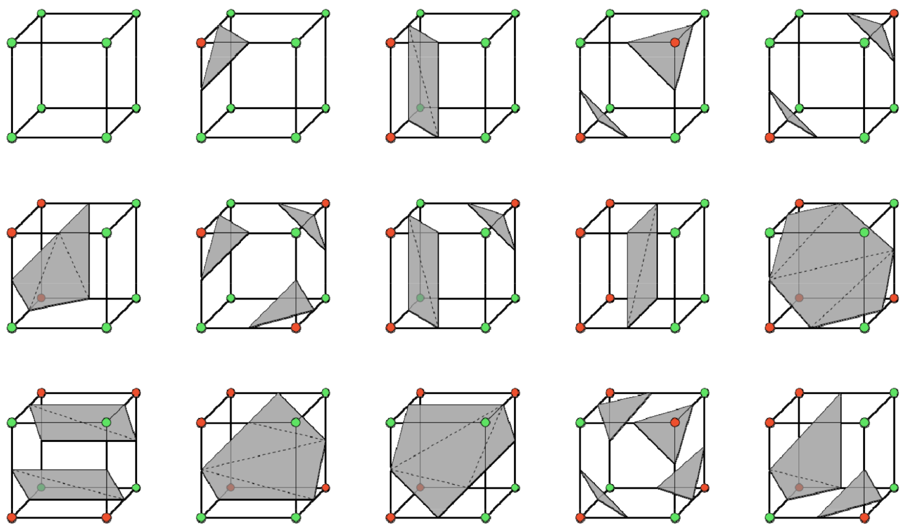
Look-up table contour lines

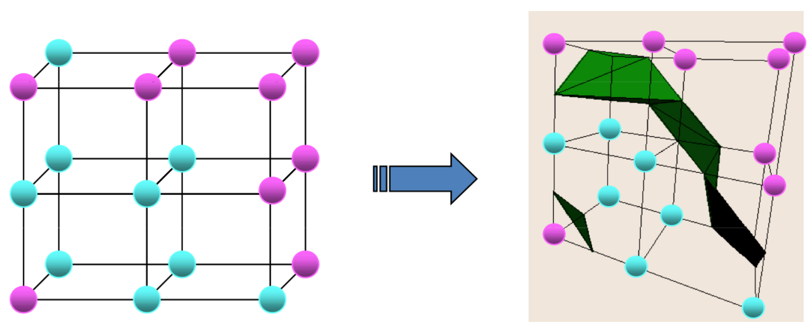


https://en.wikipedia.org/wiki/Marching_squares

Classical Solution: 3D Marching Square

- $2^8 = 256$ cases
- The first published version exploits rotation and inversion, and only considers 15 unique cases:

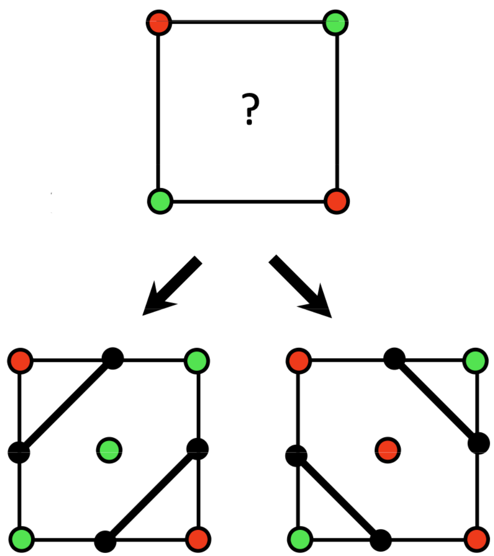


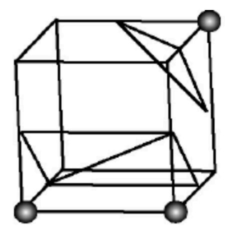


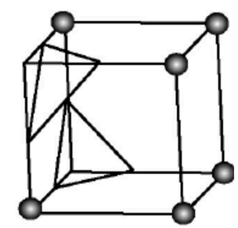
http://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/04_Surface_Reconstruction.pdf

Ambiguity

- Ambiguity leads to holes:



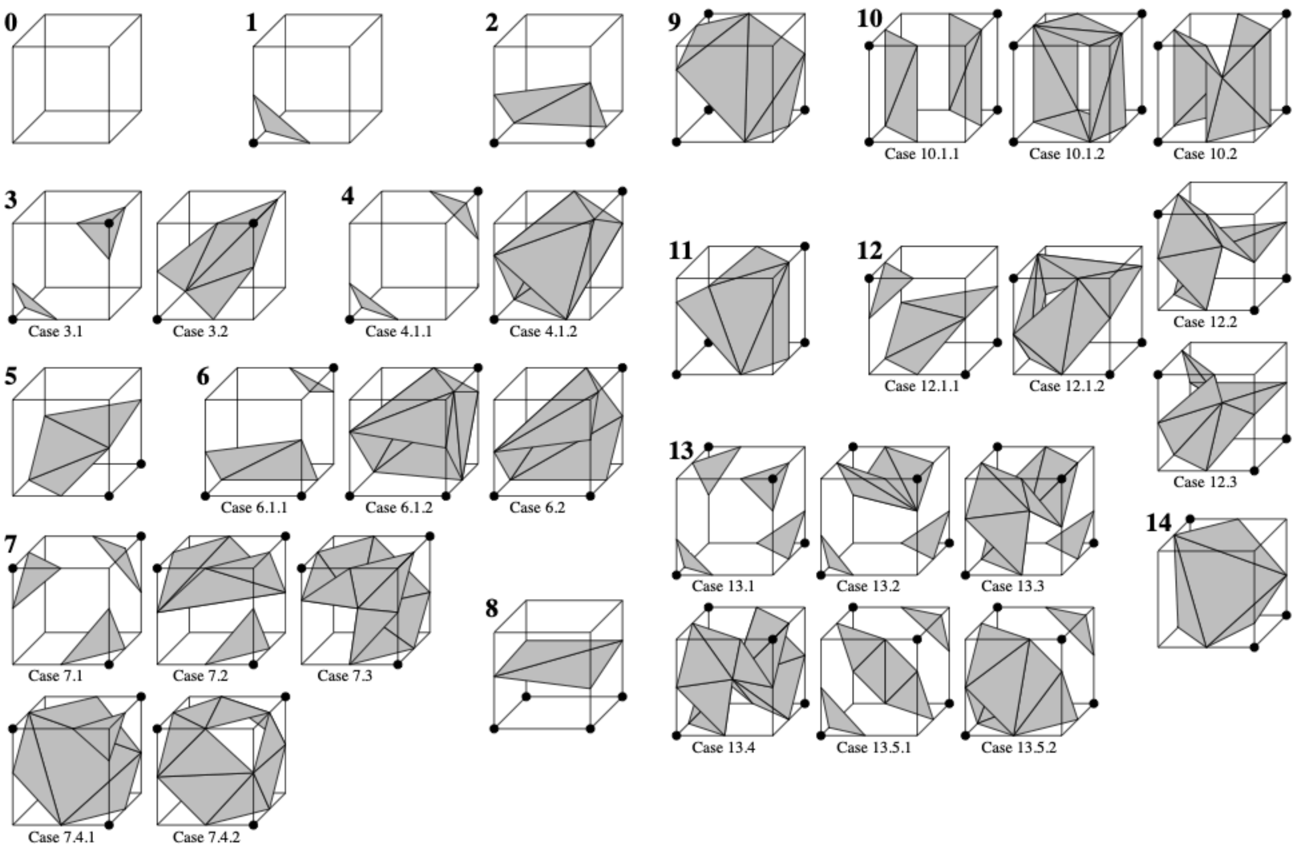


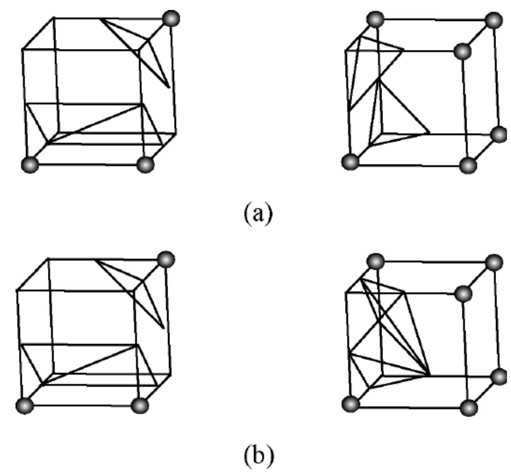


http://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/04_Surface_Reconstruction.pdf

Solution to Ambiguity

- Considering more cases in the look-up table by watching larger context:



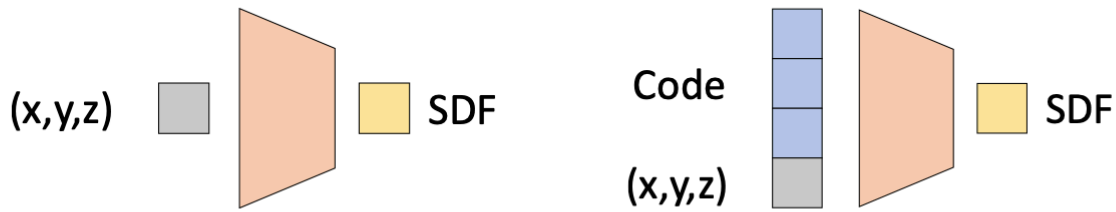


Chernyaev et al., "M**arching cubes 33: Construction of topologically correct isosurfaces.**", 1995

Comparison

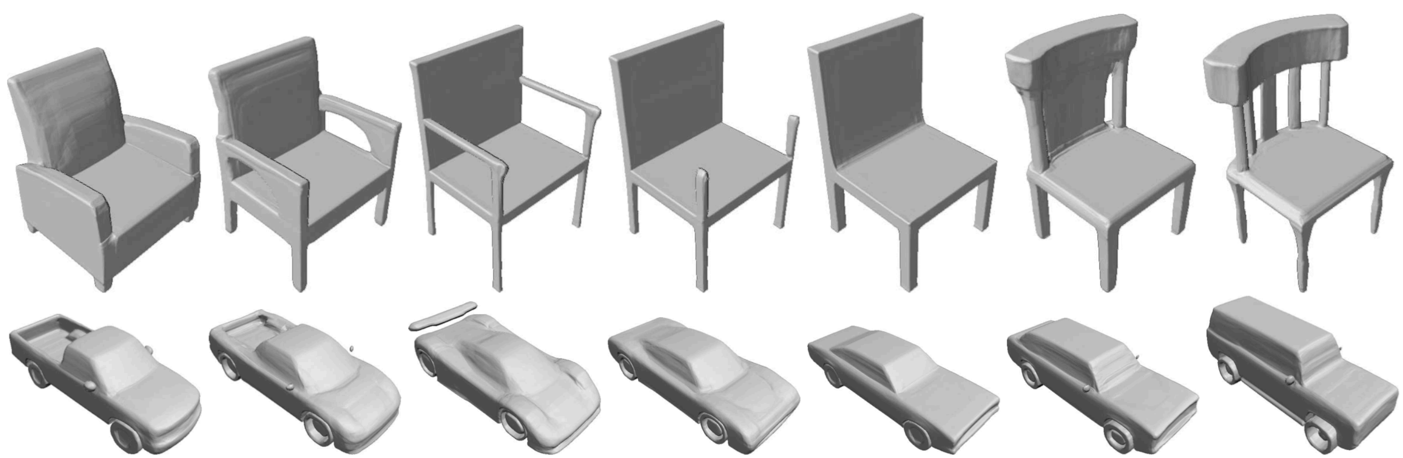
	Explicit meshing (e.g., ball-pivoting)	Implicit meshing (e.g., RBF, Poisson)
Sensitive to normals	No	Yes
Watertight manifold	No	Yes, in most cases
Complexity	Linear	• Large-scale equations to estimate implicit function • Marching cubes • Dense voxelization

Use Neural Network to Approximate Implicit Field Function



(a) Single Shape DeepSDF **(b)** Coded Shape DeepSDF

- (a) use the network to overfit a single shape
- (b) use a latent code to represent a shape, so that the network can be used for multiple shapes



Park et al., "**DeepSDF: Learning continuous signed distance functions for shape representation.**", *CVPR 2019*

Consistent Orientation?

- Require the ground-truth ***signed*** implicit functions during training (e.g., signed distance, occupancy)

- However, 3D raw data, such as point cloud or triangle soup, are not necessarily consistently oriented
  - e.g., getting normal orientation for ShapeNet data is not easy

Recent Work: Sign Agnostic Learning of Shapes from Raw Data

- Unsigned distance is easy to obtain
  - Distance to the point cloud & triangle soup
- Learn ***signed*** distance from ***unsigned*** **distance** groundtruth
  - Require a special loss function

Atzmon et al., "**Sal: Sign agnostic learning of shapes from raw data.**", *CVPR 2020*

Sign Agnostic Learning

$$\text{loss}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\sim D_\chi} \tau\left(f(\boldsymbol{x};\boldsymbol{\theta}), h_\chi(\boldsymbol{x})\right)$$

- $\chi \subset \mathbb{R}^3$: input raw data (*e.g.*, a point cloud or a triangle soup)
- $f(x;\theta) : \mathbb{R}^3 \times \mathbb{R}^m \rightarrow \mathbb{R}$: learned signed function
- D_χ: distribution of the training samples defined by χ
- $h_\chi(\boldsymbol{x})$: some *unsigned* distance measure to χ
- $\tau : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}$: a similarity function

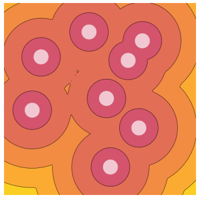
Atzmon et al., "**Sal: Sign agnostic learning of shapes from raw data.**", *CVPR 2020*

Sign Agnostic Learning

$$\text{loss}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\sim D_\chi} \tau\left(f(\boldsymbol{x};\boldsymbol{\theta}), h_\chi(\boldsymbol{x})\right)$$

- $h_\chi(\boldsymbol{x})$: some *unsigned* distance measure to χ

$$h_2(\boldsymbol{z}) = \min_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{z}-\boldsymbol{x}\|_2$$



$$h_0(z) = \begin{cases} 0 & z \in \mathcal{X} \\ 1 & z \notin \mathcal{X} \end{cases}$$

- $\tau : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}$: a similarity function

$$\tau_\ell(a, b) = \big||a| - b\big|^\ell$$

Atzmon et al., "**Sal: Sign agnostic learning of shapes from raw data.**", *CVPR 2020*

Two Local Minima

$$\text{loss}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\sim D_\chi}\tau\left(f(\boldsymbol{x};\boldsymbol{\theta}), h_\chi(\boldsymbol{x})\right)$$

- f is an ***unsigned*** function that resembles $h_\chi(\boldsymbol{x})$
- f is a ***signed*** function and $|f|$ resembles $h_\chi(\boldsymbol{x})$
- We prefer the second case to use marching cube

Atzmon et al., "**Sal: Sign agnostic learning of shapes from raw data.**", *CVPR 2020*

Two Local Minima

- Pick a special weight initialization θ^0 so that $f\left(\boldsymbol{x};\boldsymbol{\theta}^0\right) \approx \varphi(\|\boldsymbol{x}\| - r)$, where $\varphi(\|\boldsymbol{x}\| - r)$ is the signed distance function to an r-radius sphere
  - $f > 0$ if $\|x\| > r$
  - $f < 0$ if $\|x\| < r$

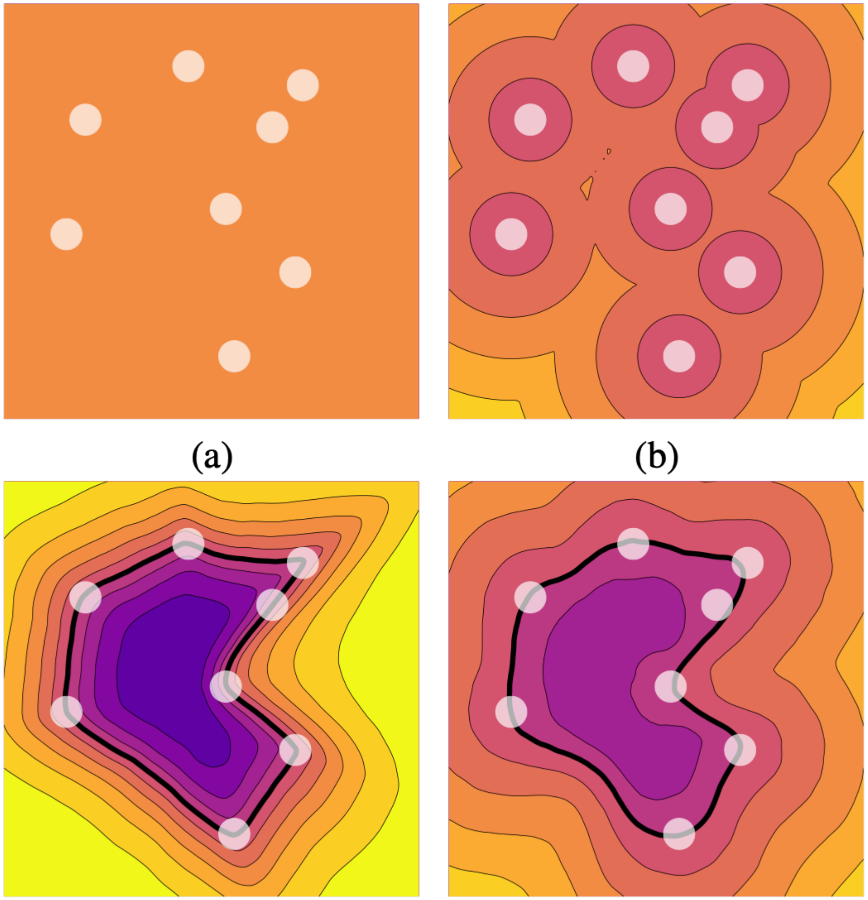
- Under such an initialization, f is not easy to converge to the unsigned local minima.

Atzmon et al., “**Sal: Sign agnostic learning of shapes from raw data.**”, *CVPR 2020*

2D Results

Unsigned function as supervision

Learned signed function



(a) (b) (c) (d)

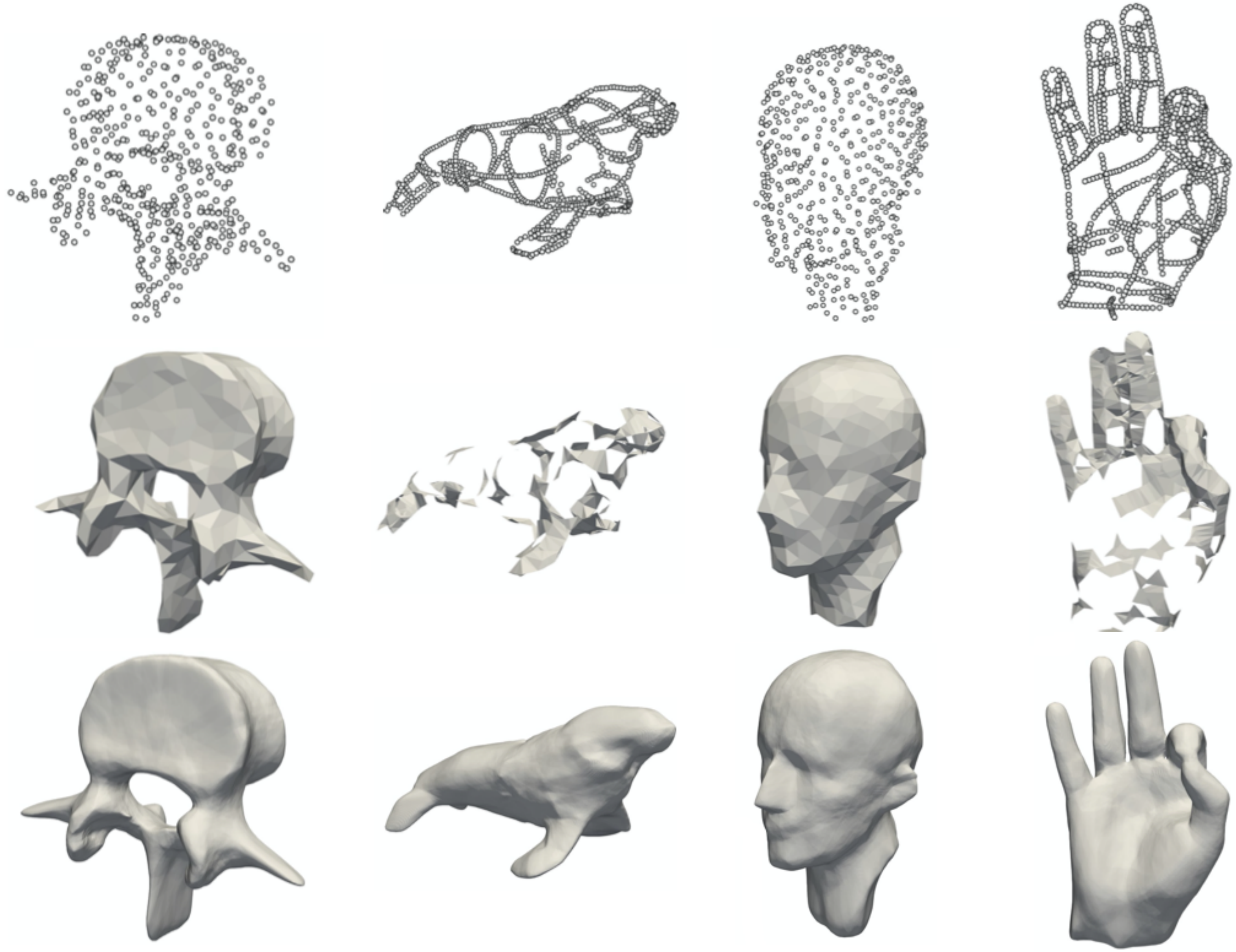
Atzmon et al., "**Sal: Sign agnostic learning of shapes from raw data.**", *CVPR 2020*

Surface Reconstruction Results from Raw Data

Raw
Point Cloud

Ball-Pivoting
Algorithm

SAL



Atzmon et al., "**Sal: Sign agnostic learning of shapes from raw data.**", *CVPR 2020*