Lecture
Surface Reconstruction

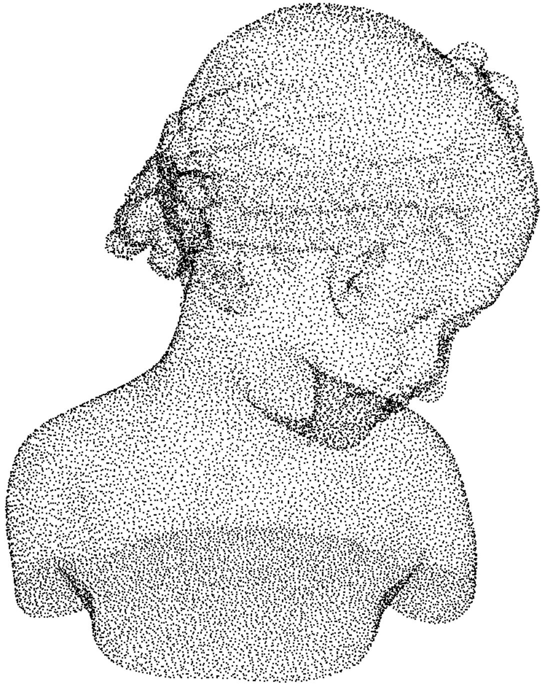
Liangliang Nan

Outline

- Introduction
- Smooth object reconstruction
  - – The pioneering work [Hoppe *et al*. 1992]
  - – Poisson reconstruction [Kazhdan *et al*. 2006]
  - – Piecewise smooth reconstruction
- Piecewise planar object reconstruction [Nan and Wonka. 2017]

Introduction

- Data sources
  - Laser scanning with a turntable

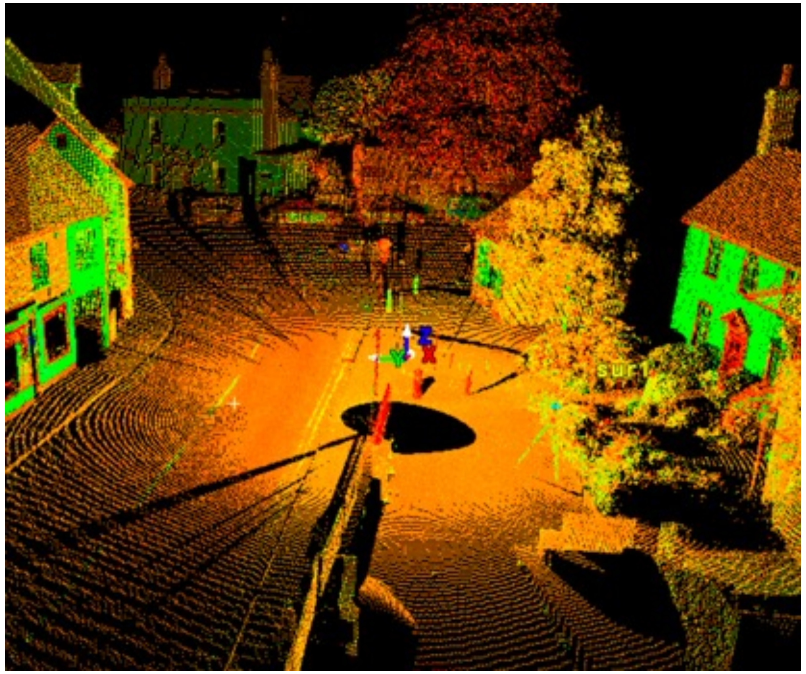


Introduction

- Data sources
  - Laser scanning with a hand-held scanner

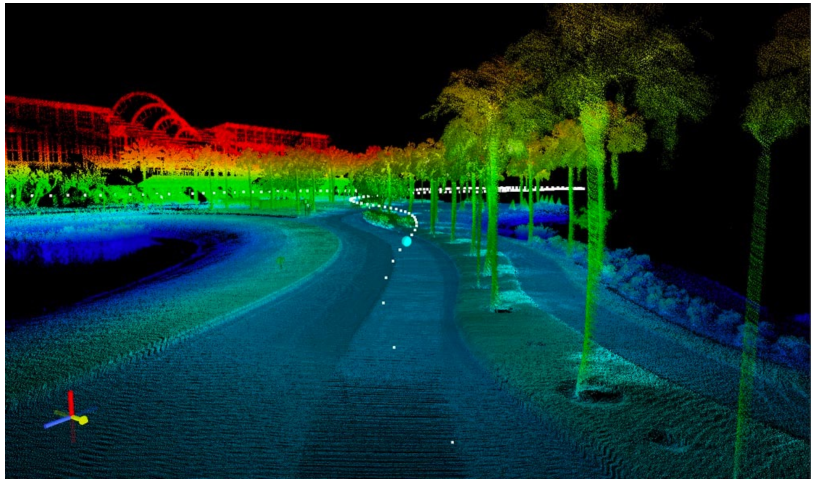
Introduction

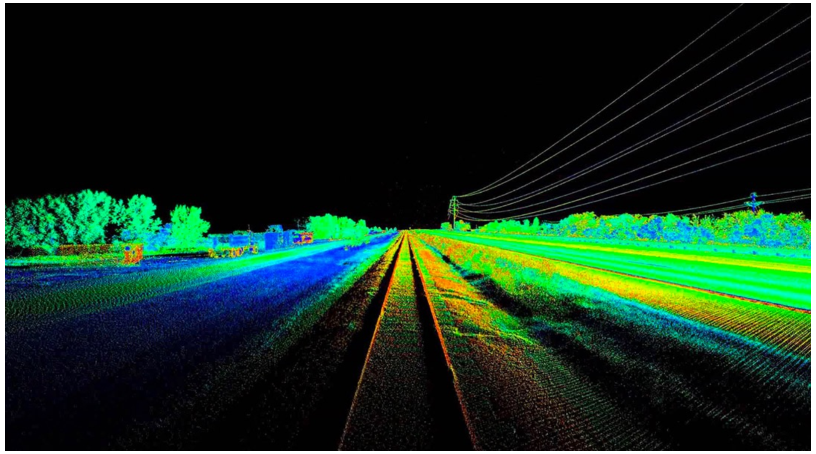
- Data sources
  - Laser scanning with static laser scanner (range of 100, 200… meters)



Introduction

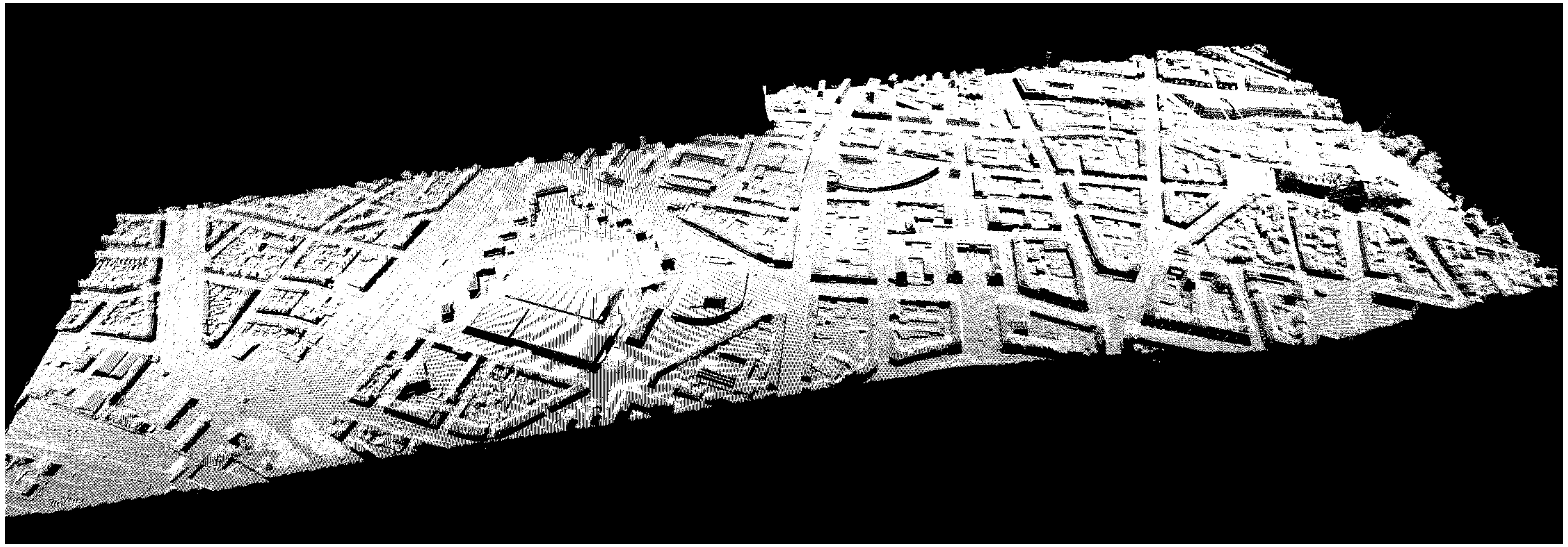
- Data sources
  - Laser scanning – mobile scanners





Introduction

- Data sources
  - Laser scanning – airborne LiDAR



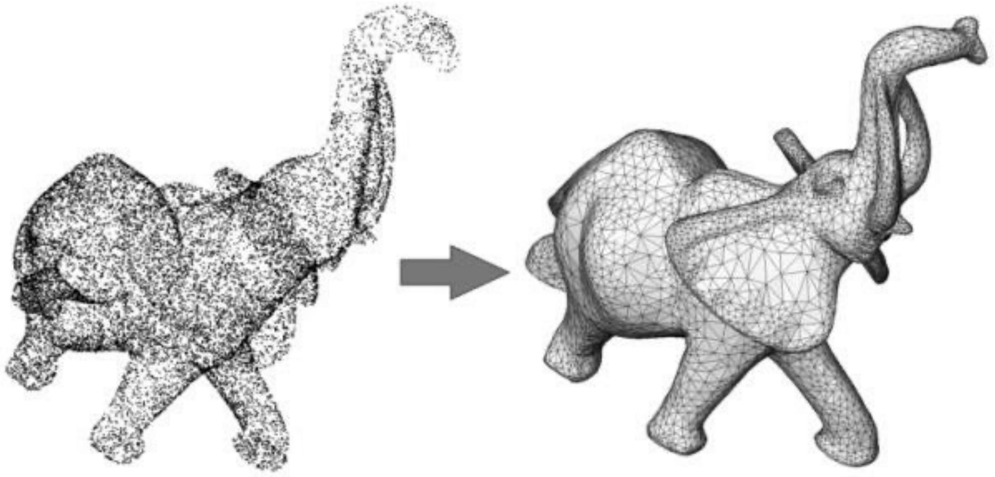
Introduction

- Data sources
  - Laser scanning
  - Structure from Motion (SfM) and Multi-view stereo (MVS)

MVS point clouds

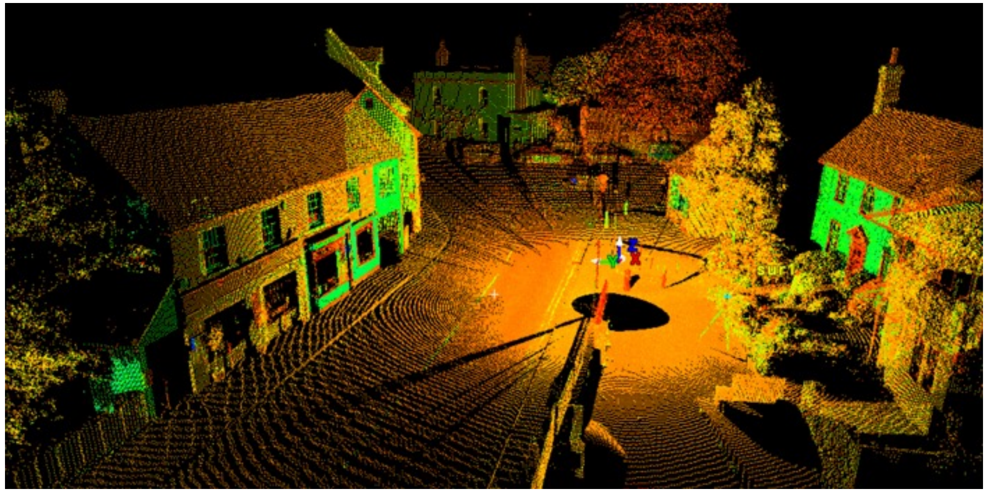
Introduction

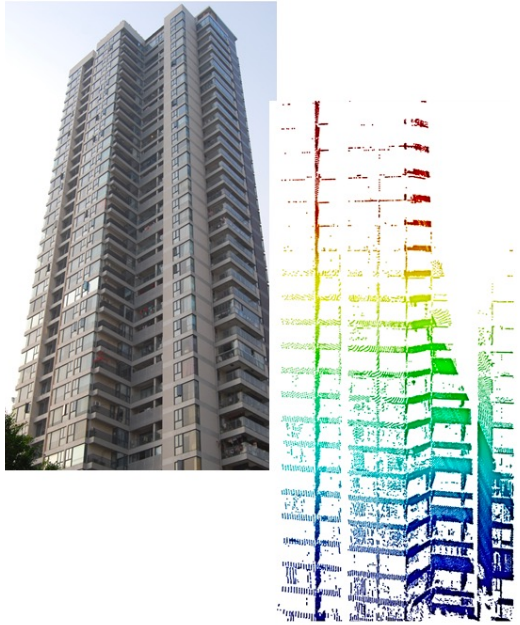
- Surface reconstruction
  - Input: point set *P* sampled over a surface *S*
    - Non-uniform sampling
    - With holes
    - With uncertainty (noise)
  - Output: surface approximating *S* in terms of topology and geometry
    - Desired
      - Watertight
      - Intersection free



Introduction

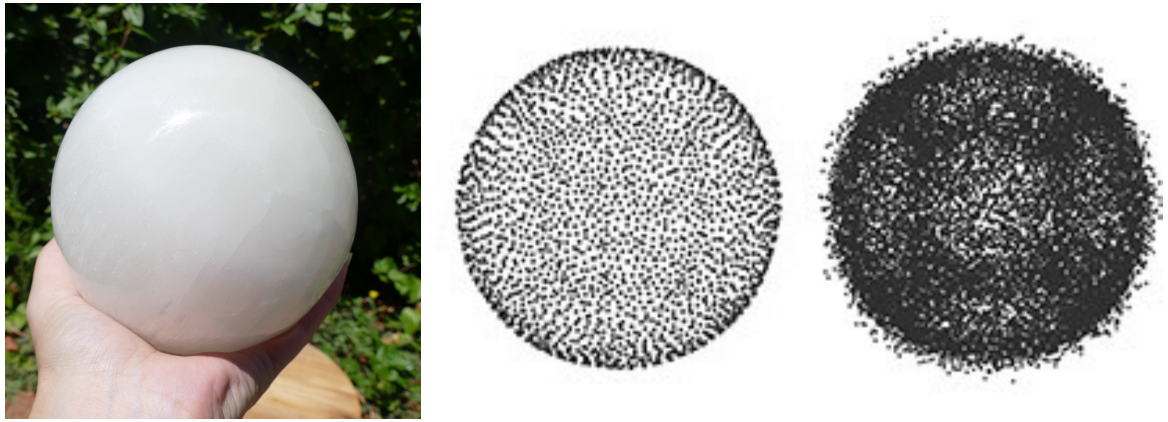
- Challenges
  - The point samples may not be uniformly distributed
    - Oblique scanning angles
    - Laser energy attenuation

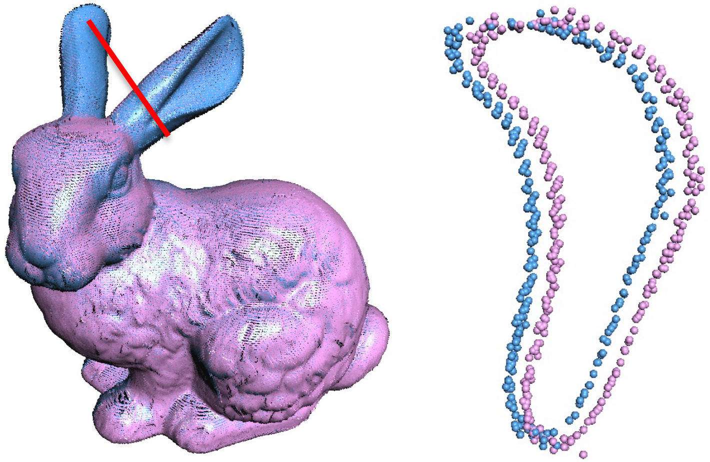




Introduction

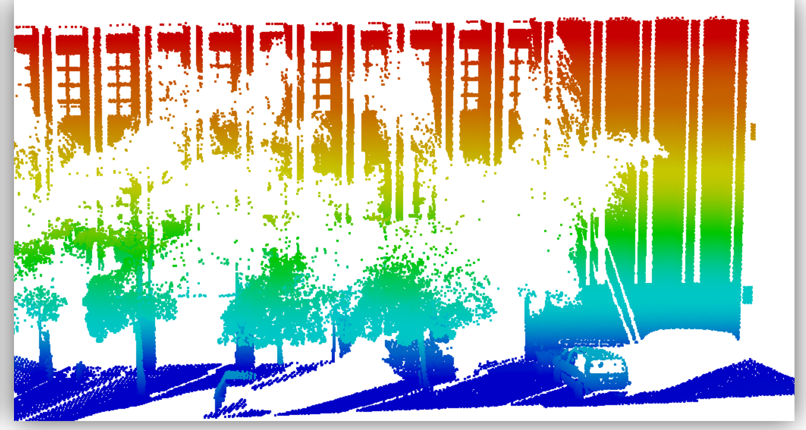
- Challenges
  - The point samples may not be uniformly distributed
  - The positions and normals are generally noisy
    - Sampling inaccuracy
    - Scan misregistration





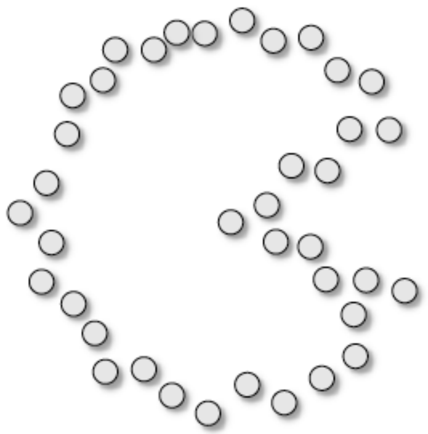
Introduction

- Challenges
  - The point samples may not be uniformly distributed
  - The positions and normals are generally noisy
  - Missing data
    - Material properties, inaccessibility, occlusion, etc.



Introduction

- Challenges
  - The point samples may not be uniformly distributed
  - The positions and normals are generally noisy
  - Missing data
  - Ill-posed problem

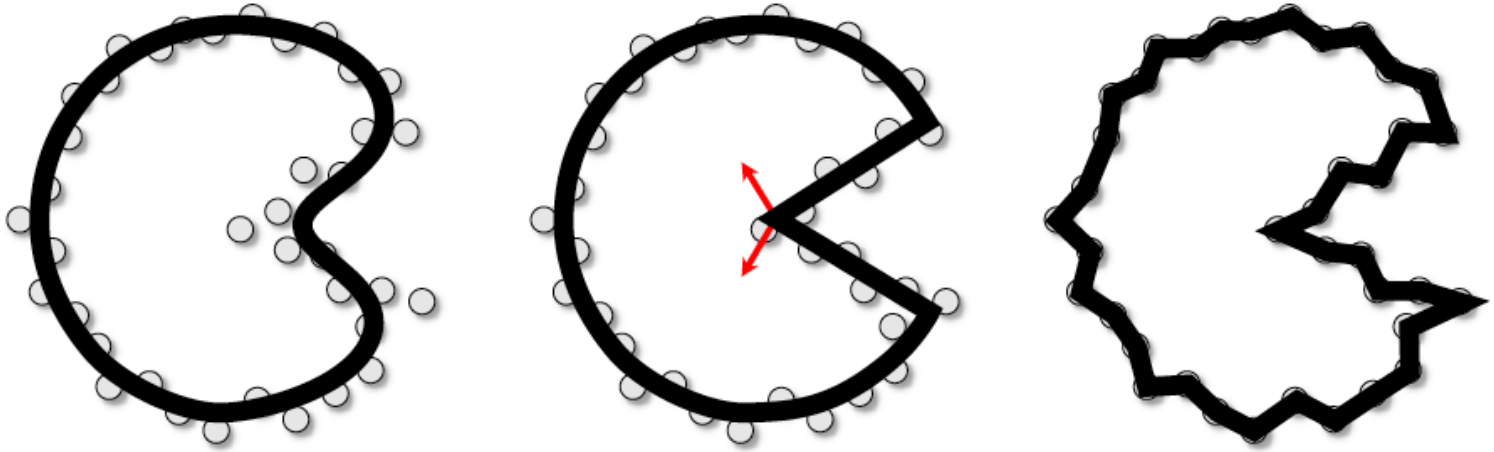


Many candidate surfaces for the reconstruction problem!

Introduction

- Challenges
  - The point samples may not be uniformly distributed
  - The positions and normals are generally noisy
  - Missing data
  - Ill-posed problem

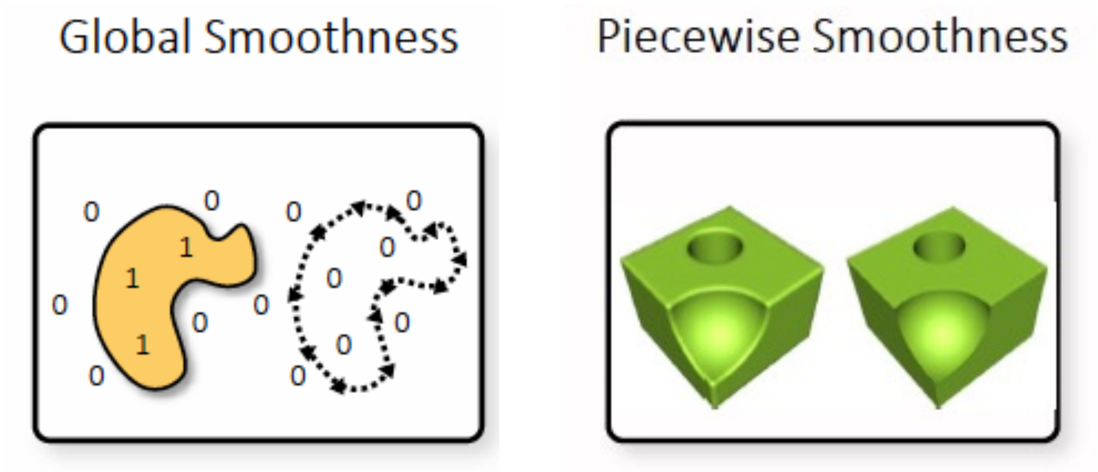
How to pick?



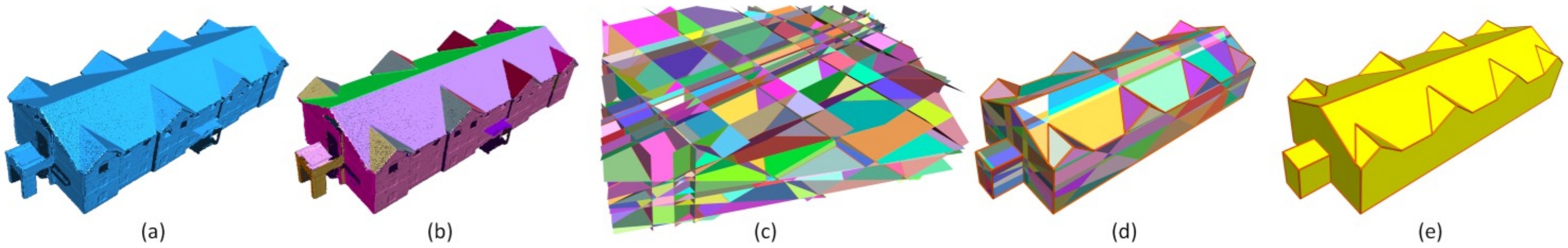
Many candidate surfaces for the reconstruction problem!

General Ideas

- Surface smoothness priors



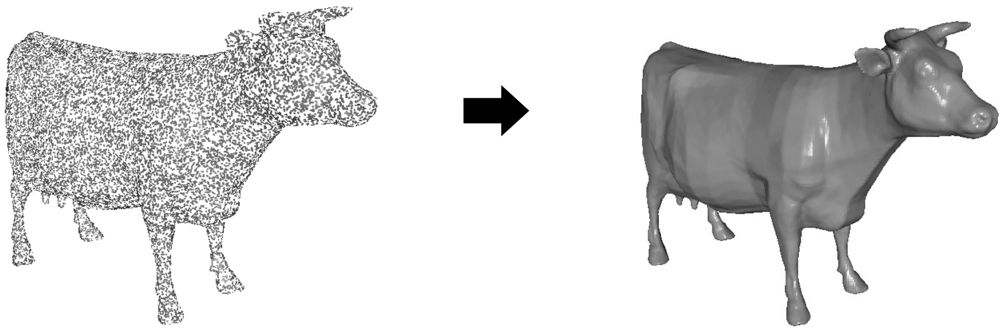
- Domain-specific priors



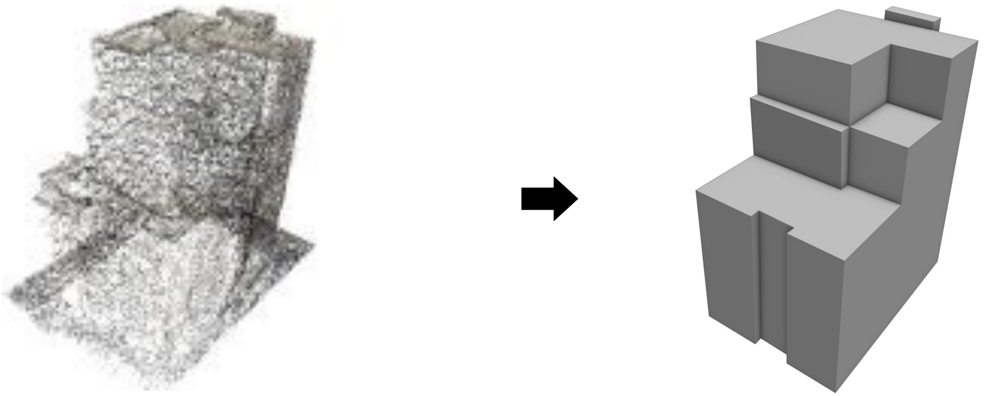
[Nan and Wonka 2017]

Introduction

- Smooth surface reconstruction



- Piecewise-planar object reconstruction

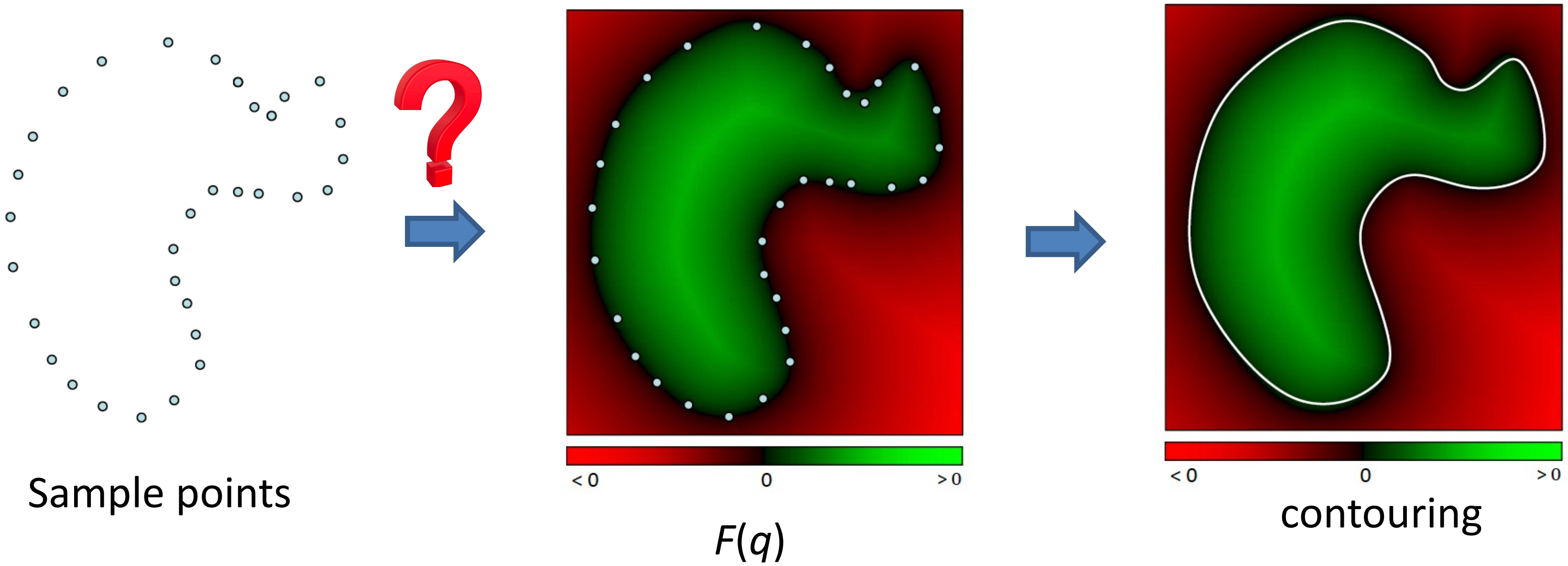


Today’s Agenda

- Introduction
- Smooth object reconstruction
  - – The pioneering work [Hoppe *et al*. 1992]
  - – Poisson reconstruction [Kazhdan *et al*. 2006]
  - – Piecewise smooth reconstruction
- Piecewise planar object reconstruction [Nan and Wonka. 2017]

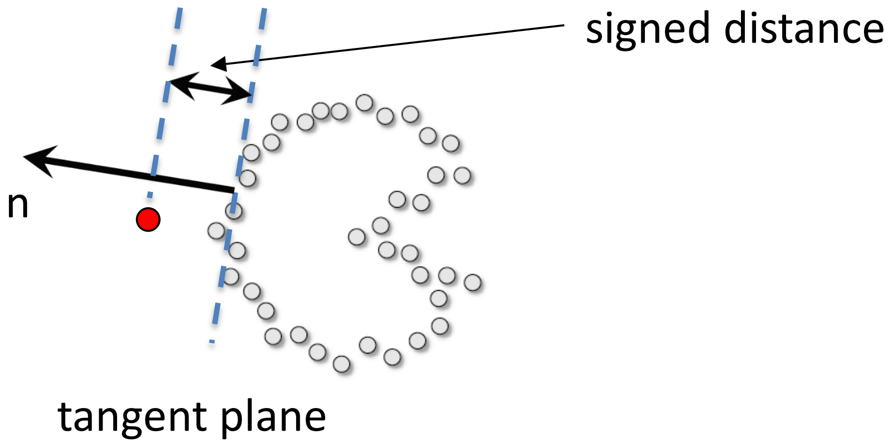
The pioneering work [Hoppe *et al.* 1992]

- Two main steps
  - Estimate signed geometric distance to the unknown surface
  - Extract the zero-set of the distance field using a contouring algorithm



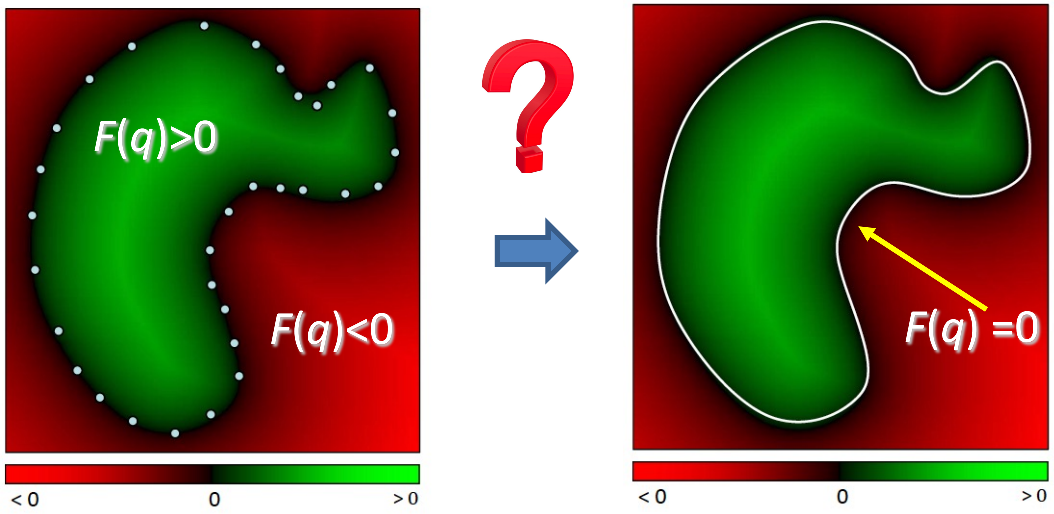
The pioneering work [Hoppe *et al.* 1992]

- Define a signed distance function (SDF)
  - Associate an oriented plane (tangent plane) with each of the data points
    - Tangent plane is a local linear approximation to the surface.
    - Used to define signed distance function to surface.



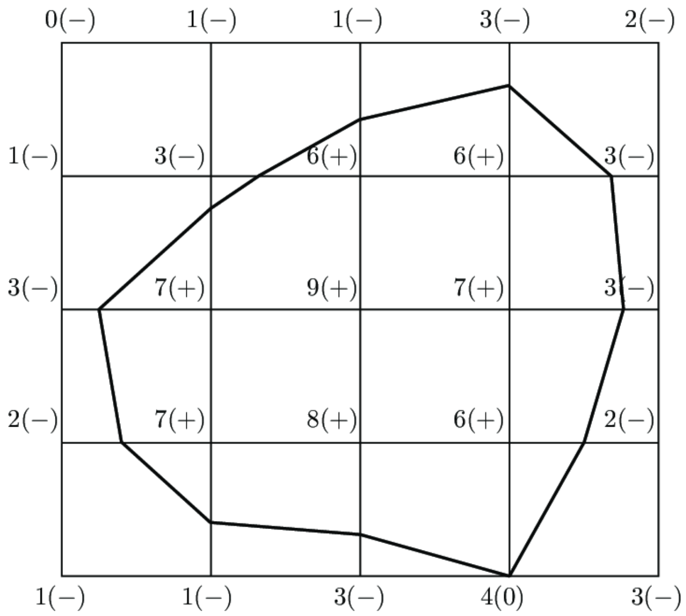
The pioneering work [Hoppe *et al.* 1992]

- Contour tracing
  - Extract 0-set iso-surface from the scalar field
    - Marching cubes

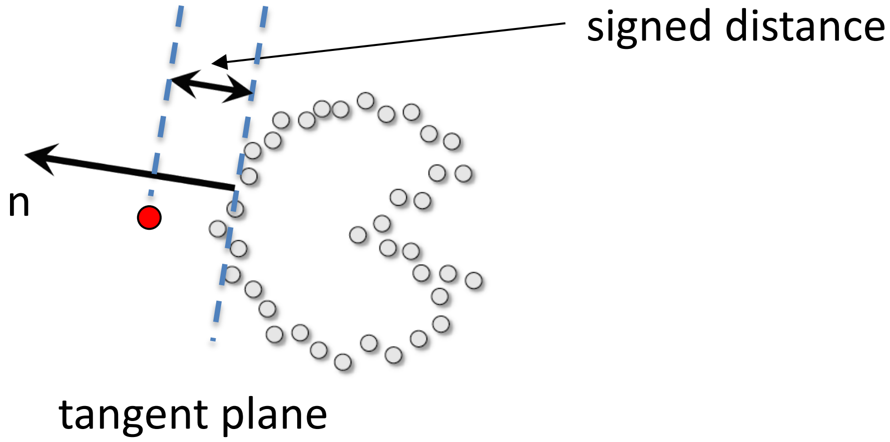


The pioneering work [Hoppe *et al.* 1992]

- Contour tracing
  - Extract 0-set iso-surface from the scalar field
    - Marching cubes

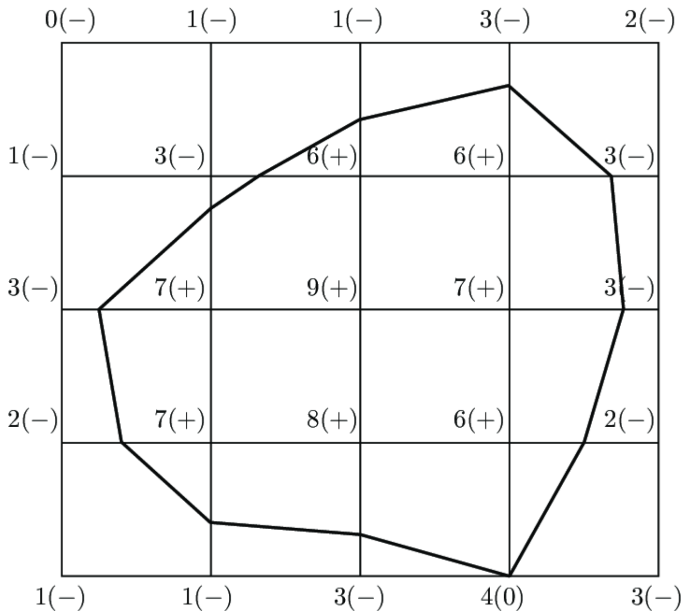


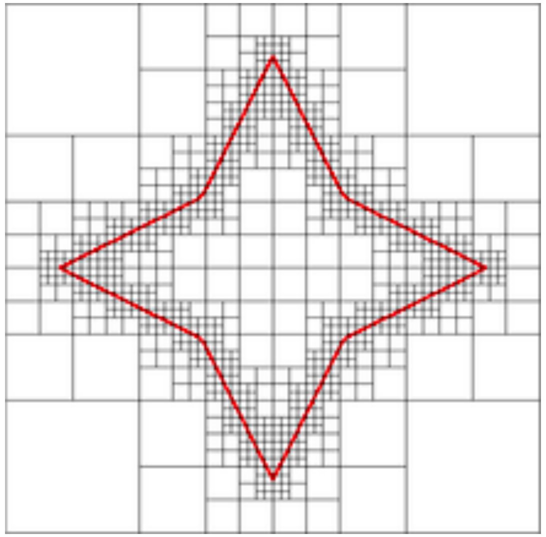
Marching squares (2D version of marching cubes)



The pioneering work [Hoppe *et al.* 1992]

- Contour tracing
  - Extract 0-set iso-surface from the scalar field
    - Marching cubes
    - Irregular grid

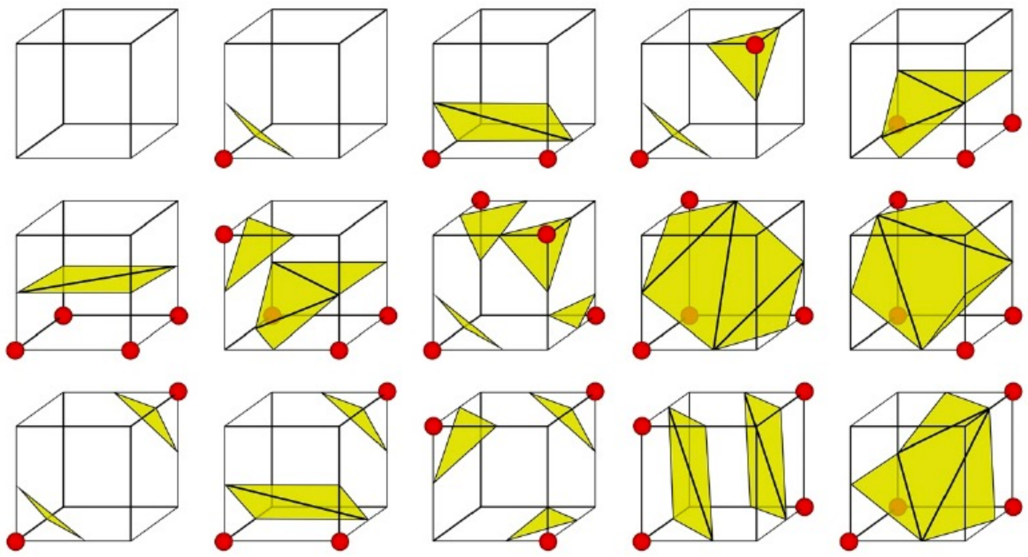


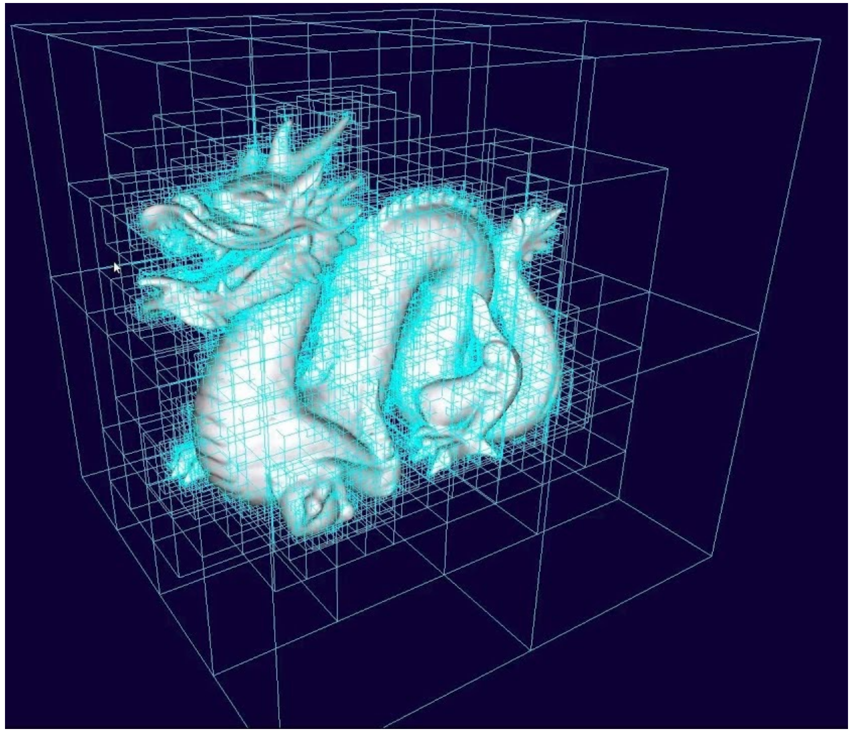


Marching squares (2D version of marching cubes)

The pioneering work [Hoppe *et al.* 1992]

- Contour tracing
  - Extract 0-set iso-surface from the scalar field
  - Marching cubes

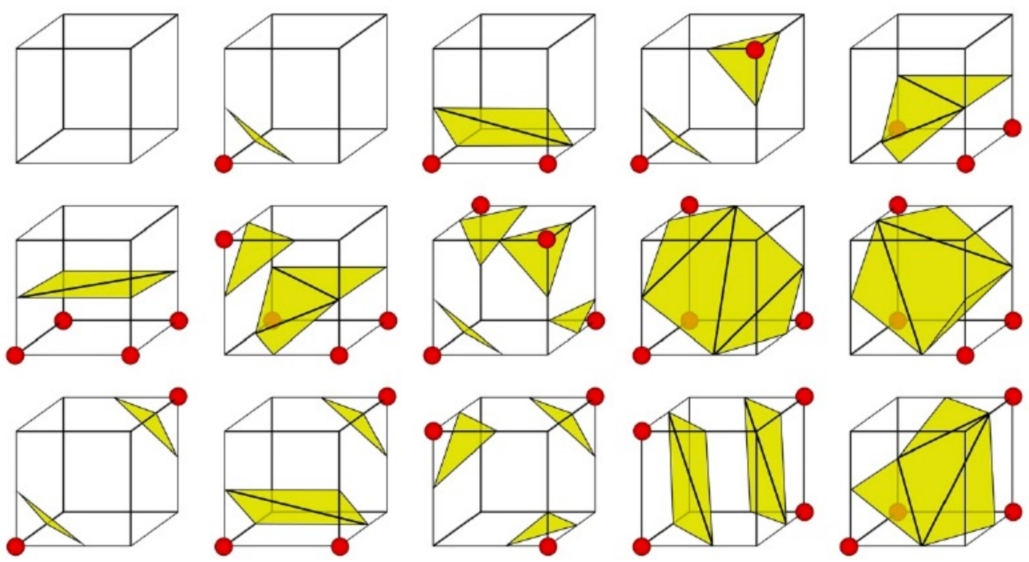


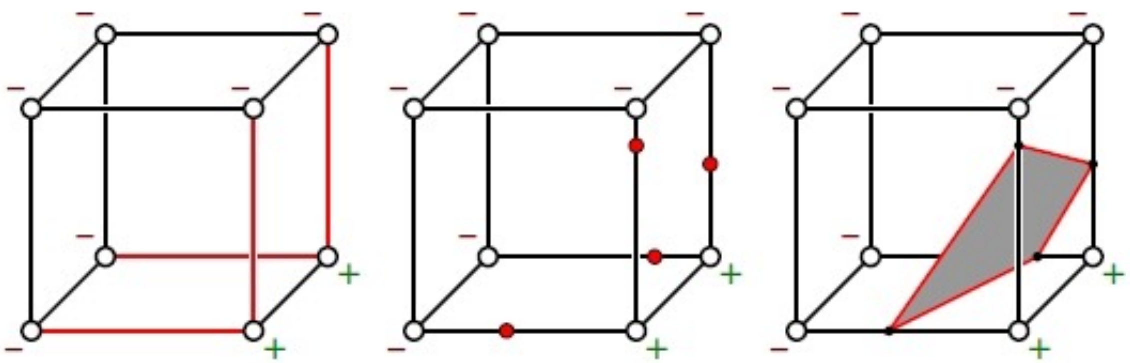


Marching cubes

The pioneering work [Hoppe *et al.* 1992]

- Contour tracing
  - Extract 0-set iso-surface from the scalar field
  - Marching cubes

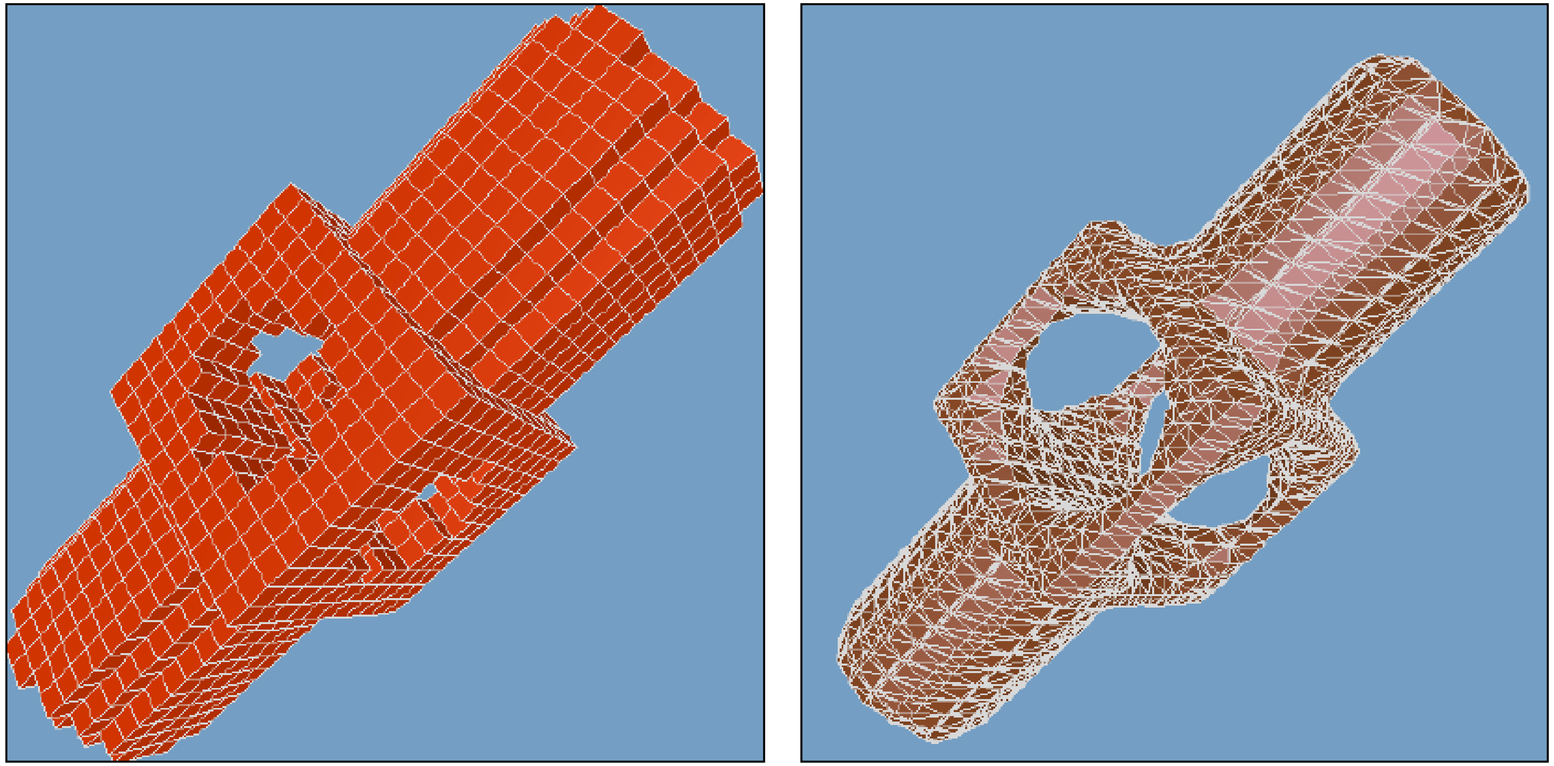




Marching cubes

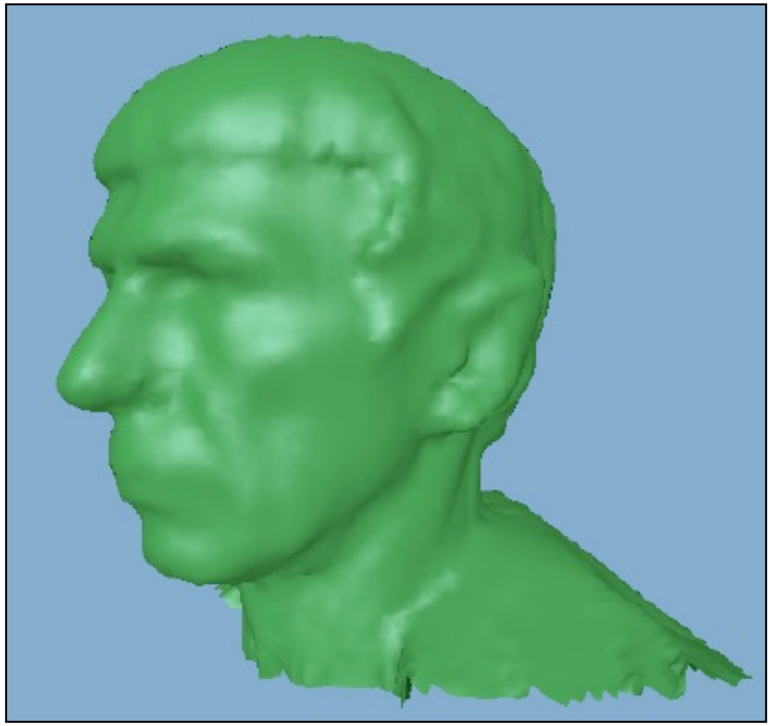
The pioneering work [Hoppe *et al.* 1992]

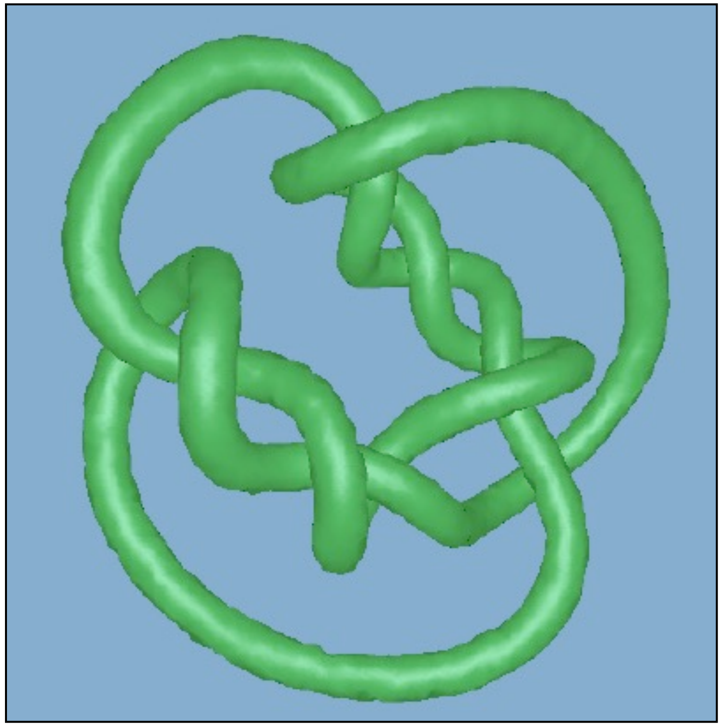
- Contour tracing



The pioneering work [Hoppe *et al.* 1992]

- Reconstruction results



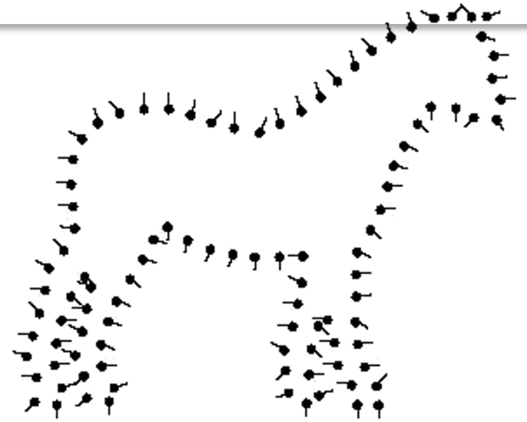


Outline

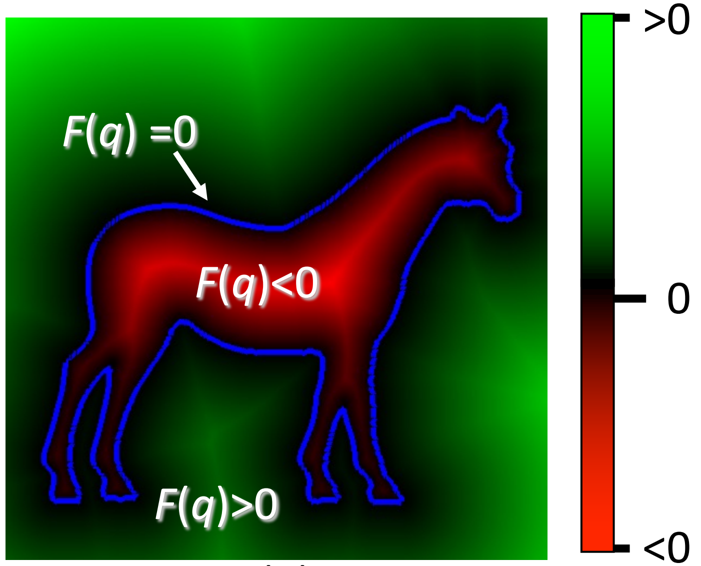
- Introduction
- Smooth object reconstruction
  - The pioneering work [Hoppe *et al*. 1992]
  - Poisson reconstruction [Kazhdan *et al*. 2006]
  - Piecewise smooth reconstruction
- Piecewise planar object reconstruction [Nan and Wonka. 2017]

Poisson Reconstruction

- Inherited idea from [Hoppe et al. 1992]
- Discrete SDF -> Implicit function fitting
  - Define a 3D scalar function
    - Zero values at the points
    - Positive values outside
    - Negative values inside
  - Extract the zero isosurface



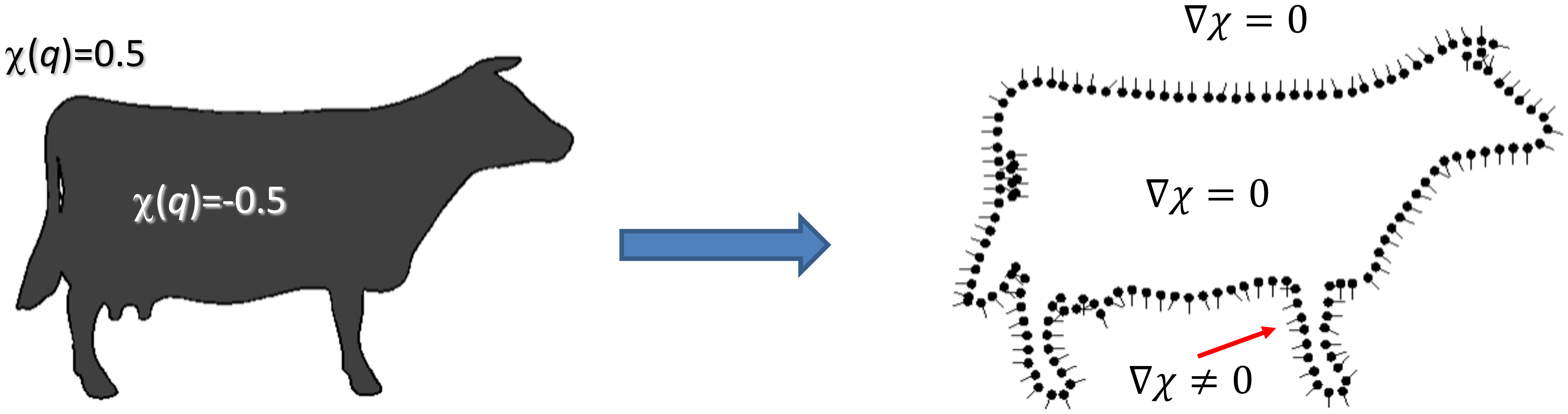
Sample points



$F(q)$

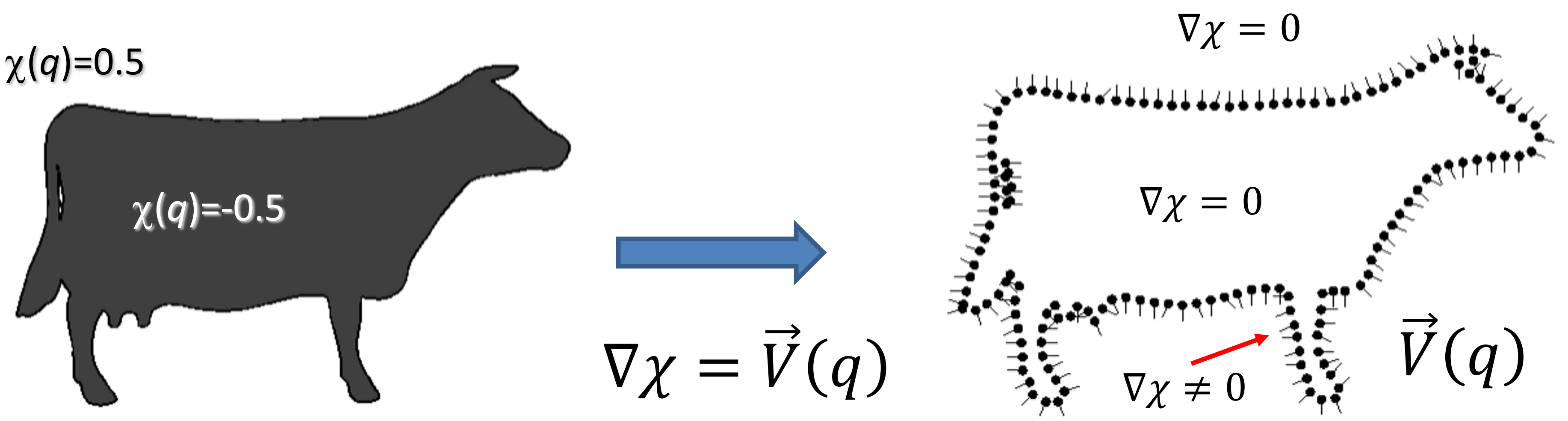
Poisson Reconstruction

- The idea
  - The indicator function (χ)
    - Interior: a constant negative value
    - Exterior: a constant positive value
  - The gradient of the indicator function ($\nabla\chi$)
    - Zero everywhere except close to the boundary



Poisson Reconstruction

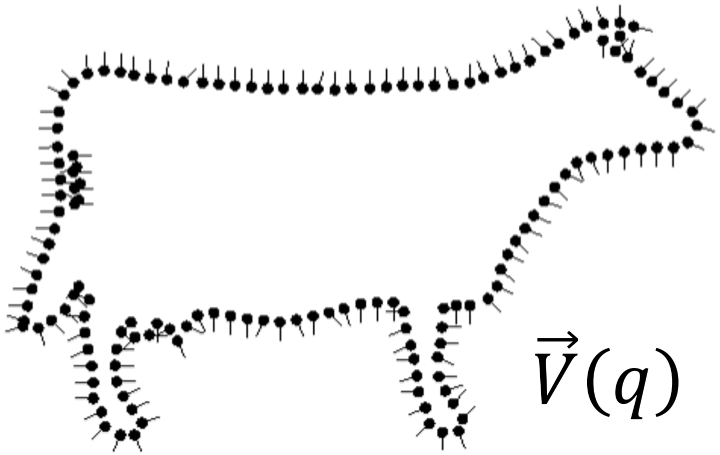
- The idea
  - The indicator function (χ)
  - The gradient of the indicator function ($\nabla\chi$)
  - Oriented points $\vec{V}(q)$ $\approx$ discretization of gradient of indicator function



Poisson Reconstruction

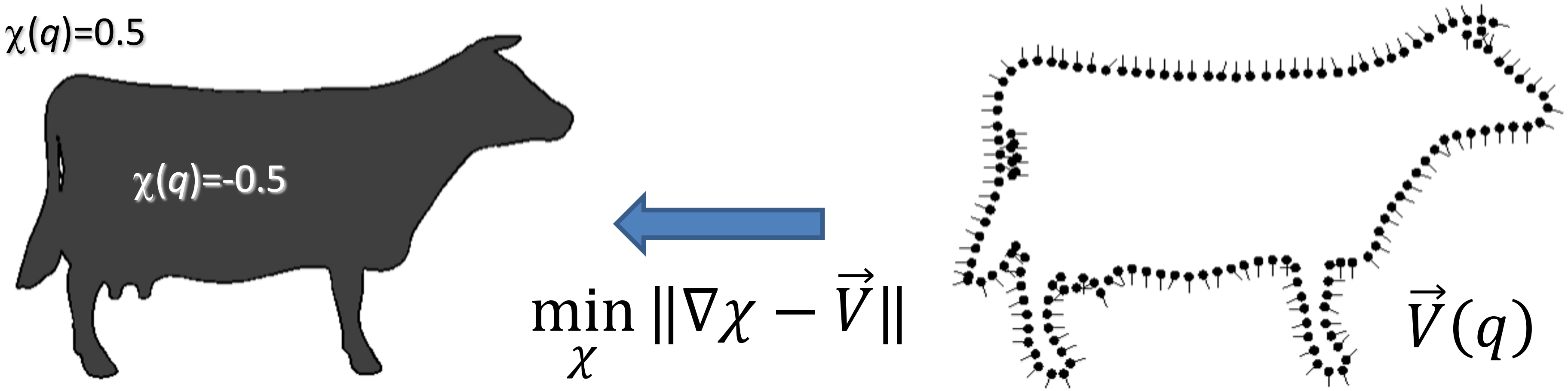
- The idea
  - The indicator function (χ)
  - The gradient of the indicator function ($\nabla\chi$)
  - Oriented points $\vec{V}(q)$ $\approx$ discretization of gradient of indicator function
  - Reconstruction

$$\nabla\chi = \vec{V}(q)$$

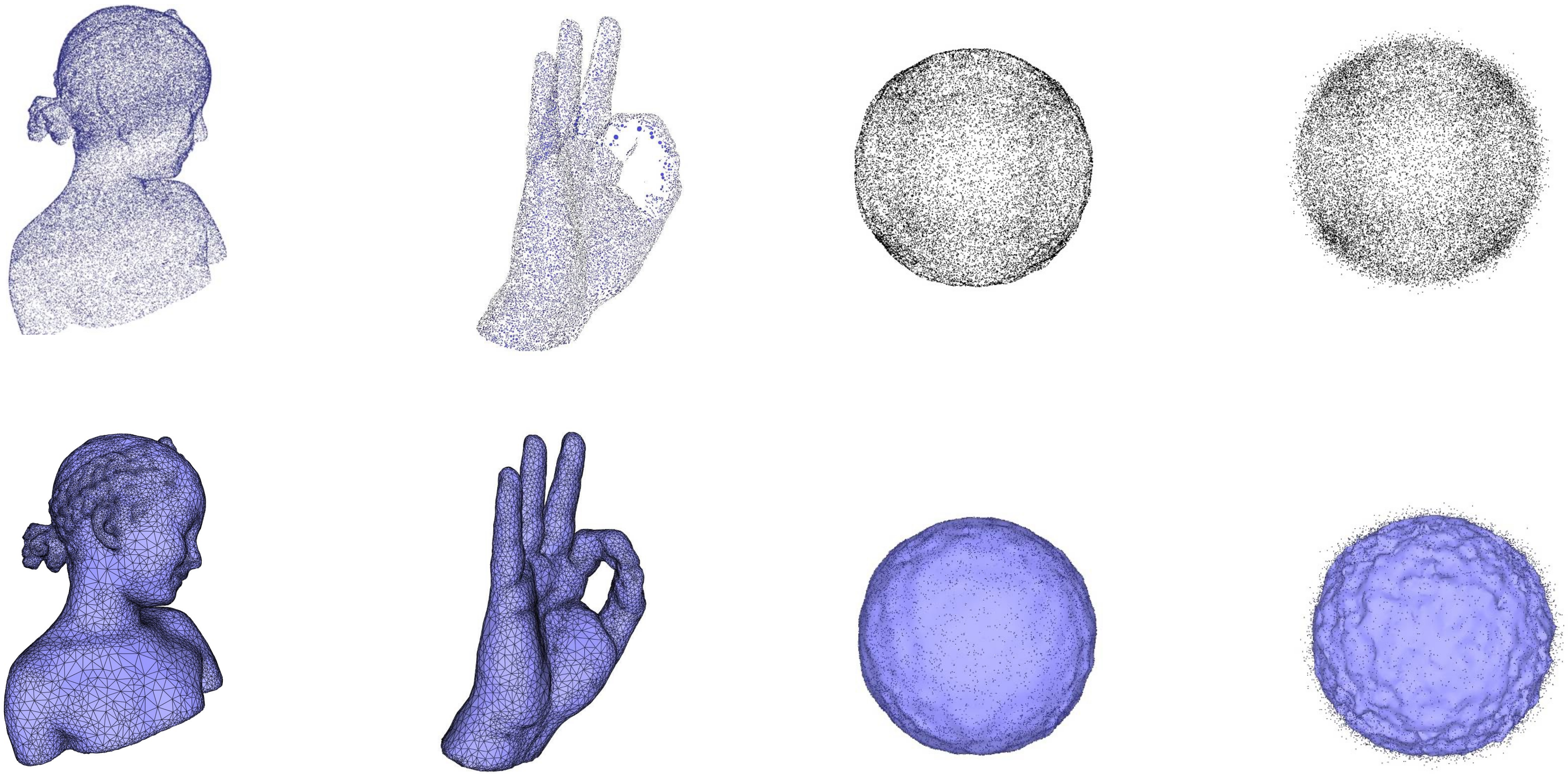


Poisson Reconstruction

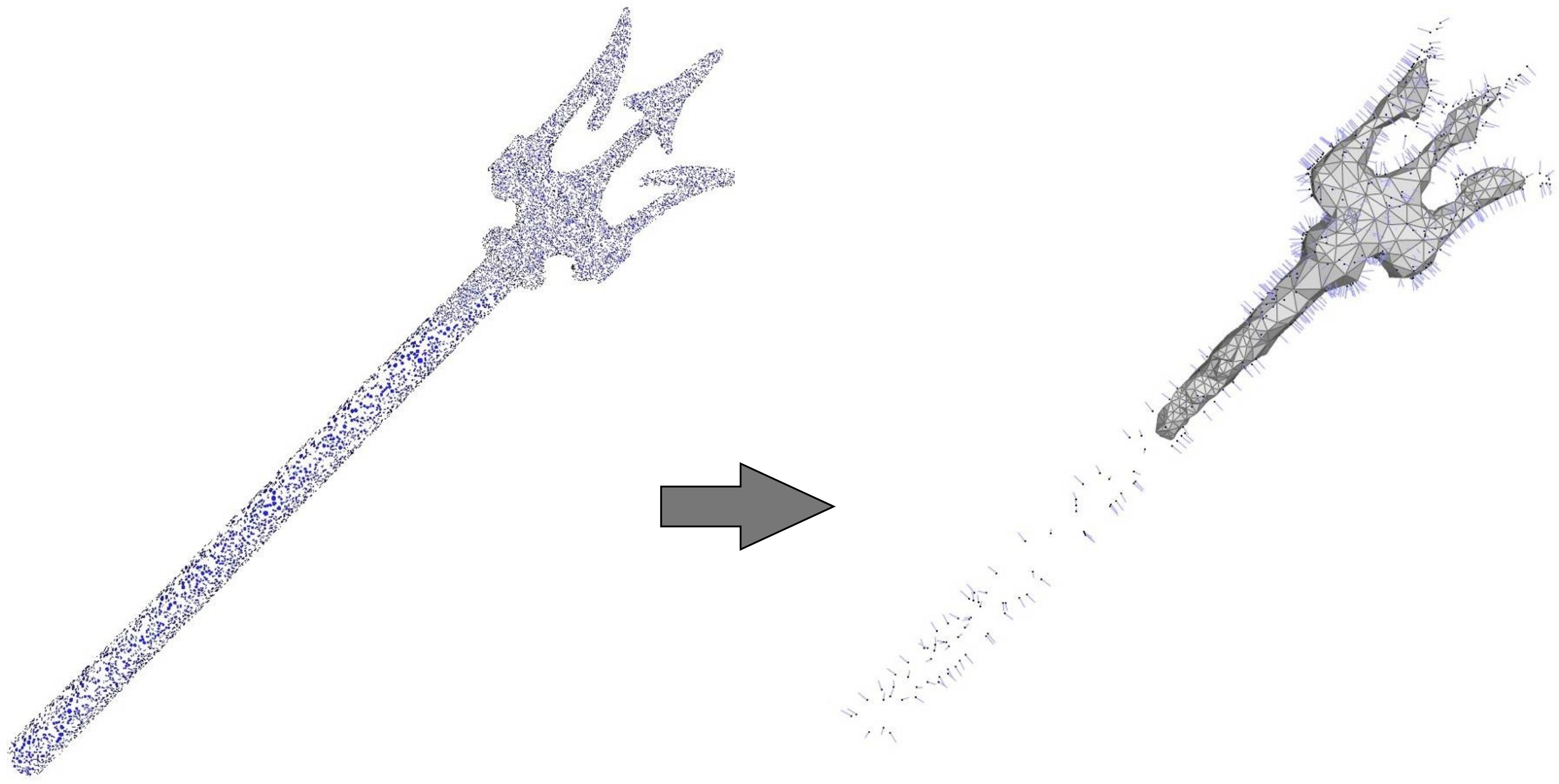
- The idea
  - The indicator function (χ)
  - The gradient of the indicator function ($\nabla\chi$)
  - Oriented points $\vec{V}(q)$ $\approx$ discretization of gradient of indicator function
  - Reconstruction: finding the indicator function + iso-surface extraction



Poisson Reconstruction



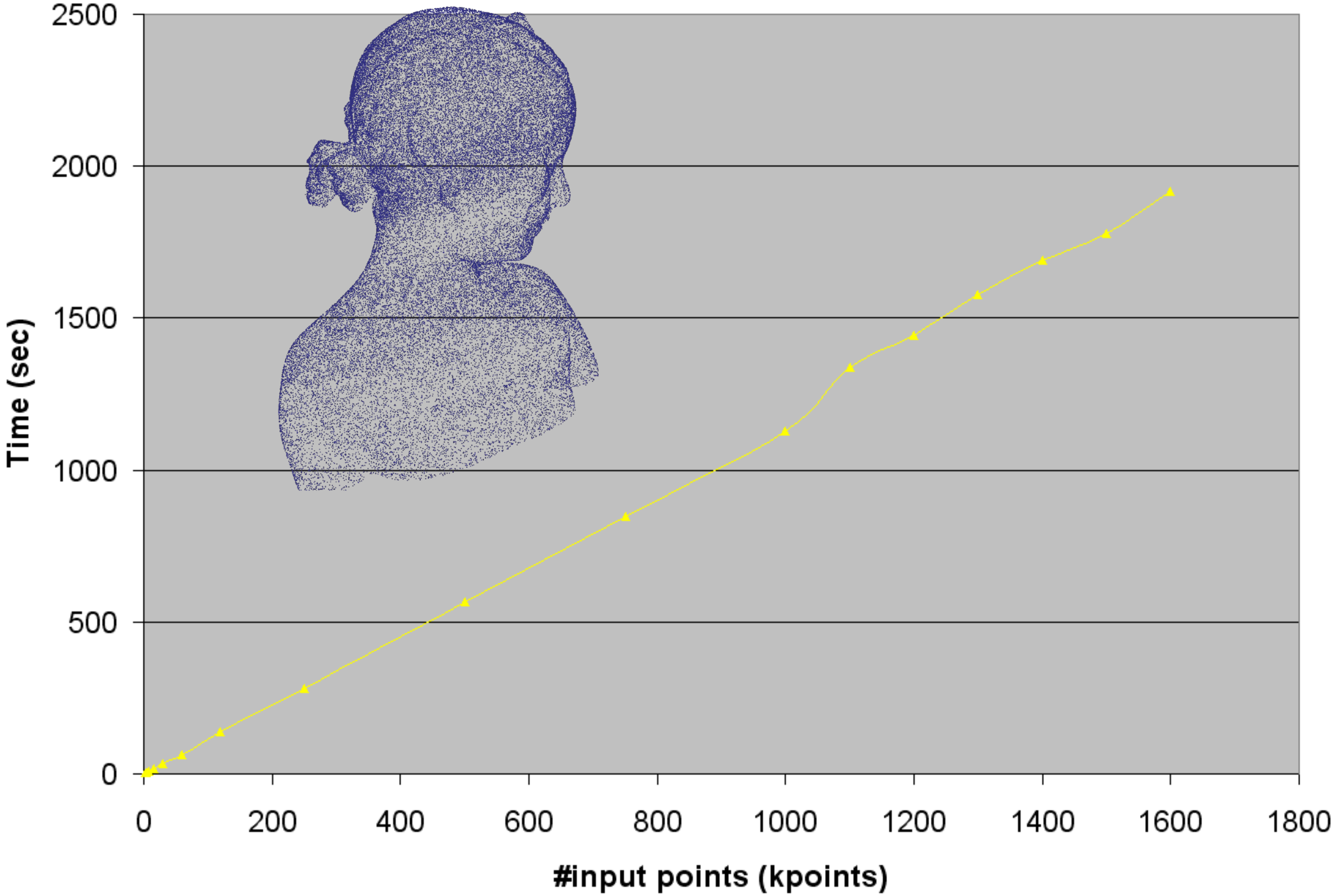
Poisson Reconstruction



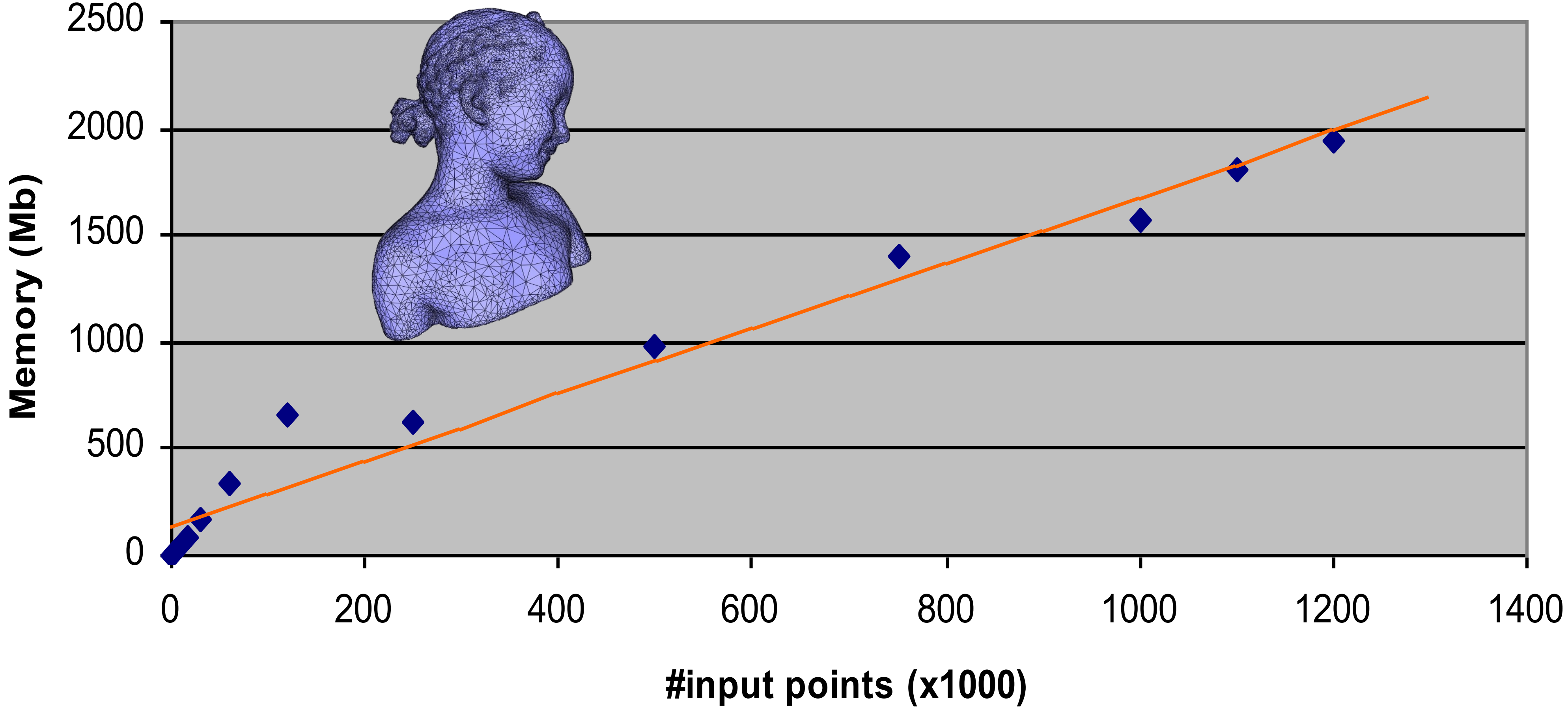
Left: 50K points sampled on
Neptune trident

Right: point set simplified to 1K
then reconstructed

Poisson duration wrt #input points

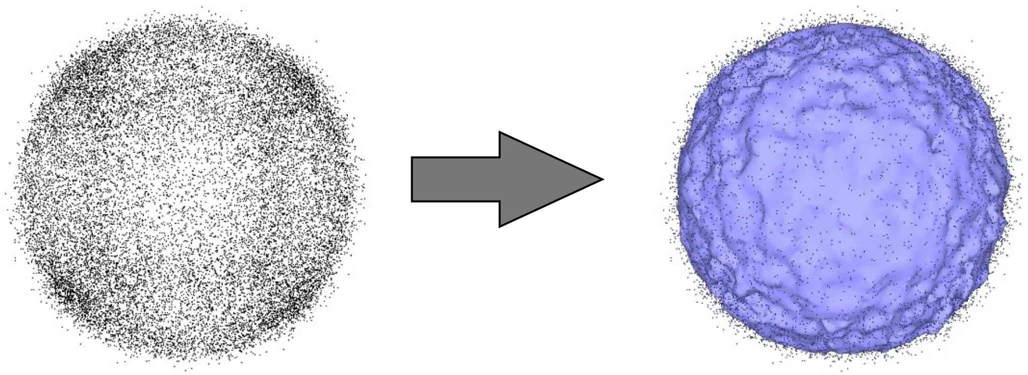


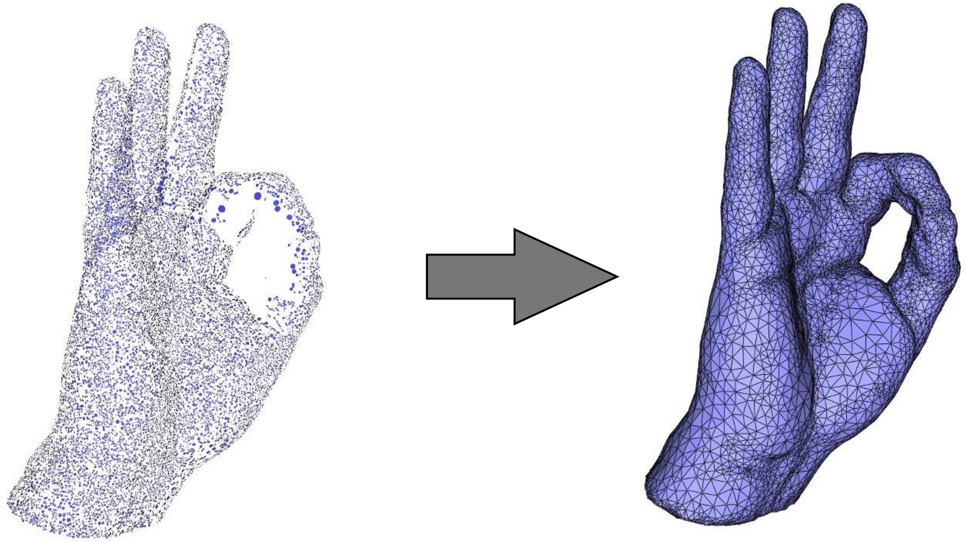
Memory wrt #input points



Poisson Reconstruction

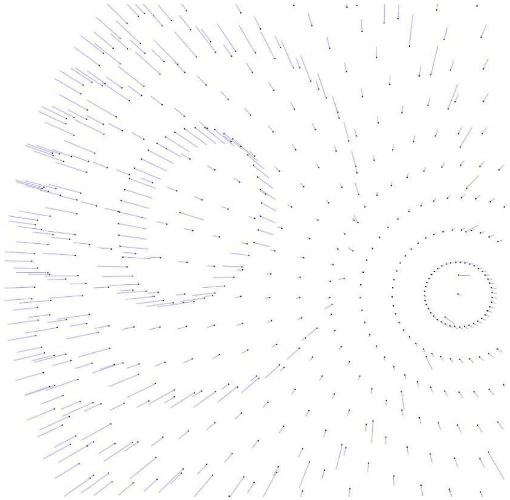
- Properties
  - ✓ Supports noisy, non-uniform data
  - ✓ Can fill reasonably large holes

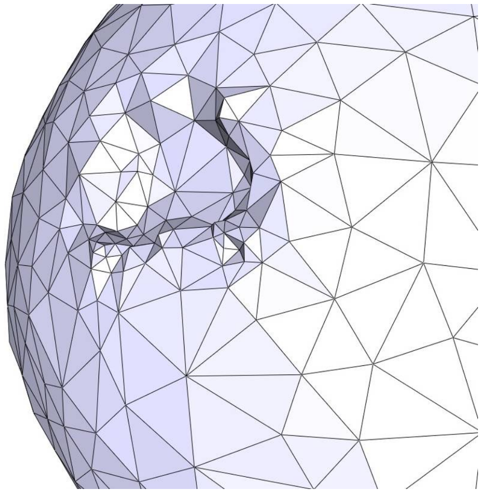




Poisson Reconstruction

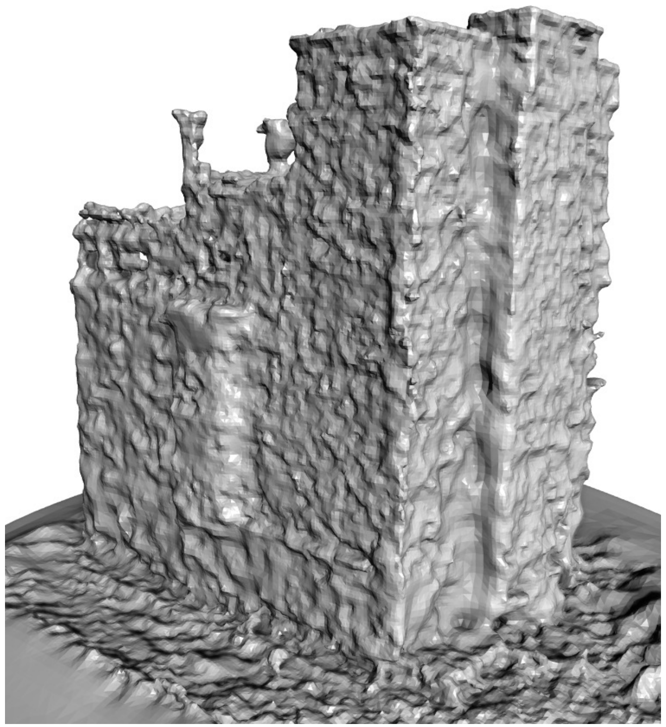
- Properties
  - ✓ Supports noisy, non-uniform data
  - ✓ Can fill reasonably large holes
- Limitations
  - It requires good normal information





Poisson Reconstruction

- Properties
  - ✓ Supports noisy, non-uniform data
  - ✓ Can fill reasonably large holes
- Limitations
  - It requires good normal information
  - Sharp features are oversmoothed
    - Not good for piecewise planar objects

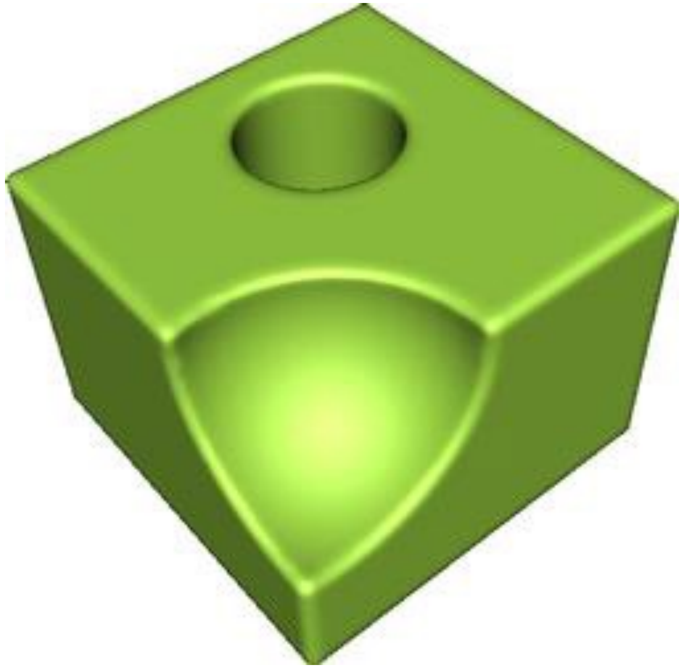


Outline

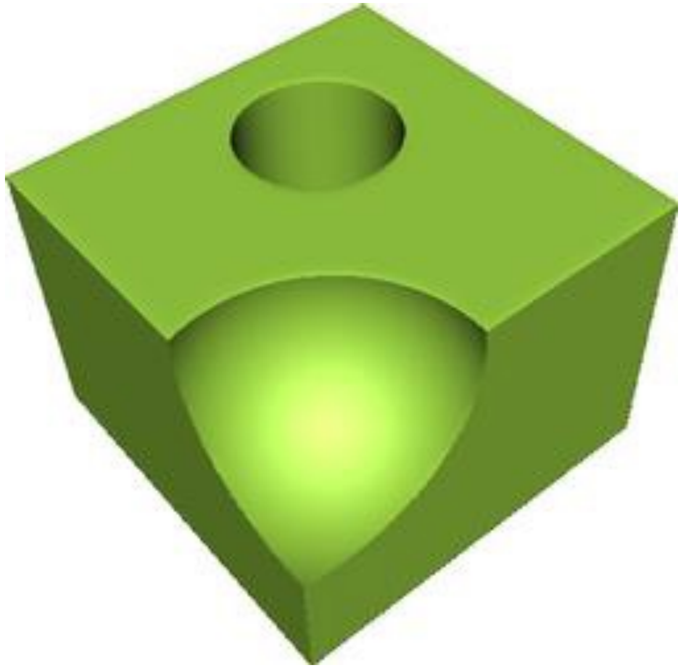
- Introduction
- Smooth object reconstruction
  - The pioneering work [Hoppe *et al*. 1992]
  - Poisson reconstruction [Kazhdan *et al*. 2006]
  - Piecewise smooth reconstruction
- Piecewise planar object reconstruction [Nan and Wonka. 2017]

Piecewise Smooth Reconstruction

- Piecewise-smooth



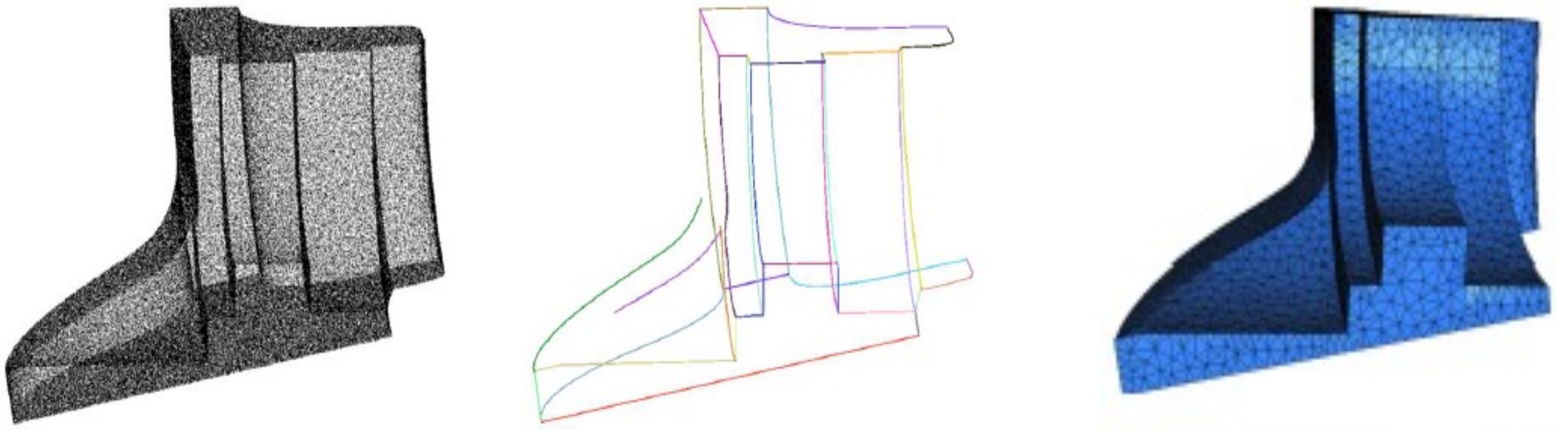
smooth



piecewise-smooth

Piecewise Smooth Reconstruction

- Feature detection
  - Extract a set of sharp features
  - Decompose the point cloud into smooth patches
- Smooth reconstruction patch by patch
- Stitch the patches

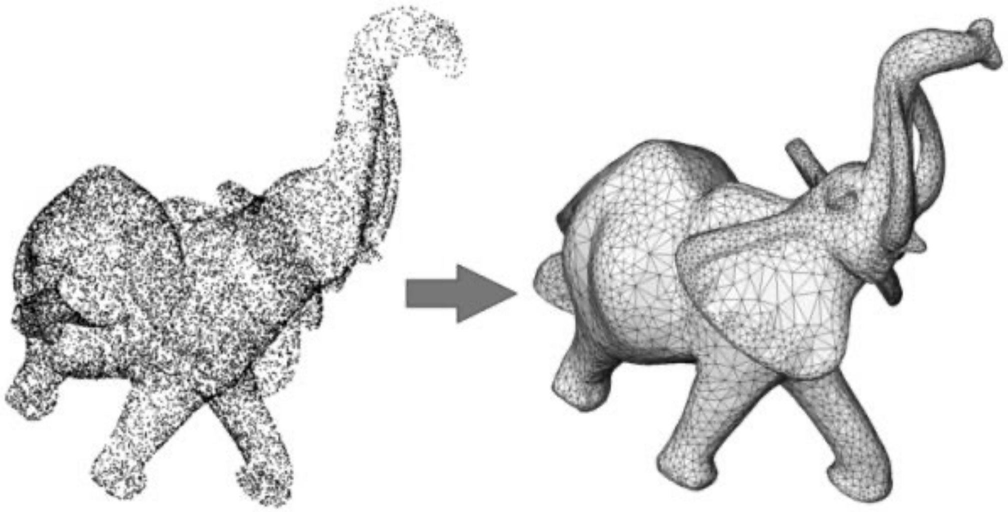


Outline

- Introduction
- Smooth object reconstruction
  - The pioneering work [Hoppe *et al*. 1992]
  - Poisson reconstruction [Kazhdan *et al*. 2006]
  - Piecewise smooth reconstruction
- Piecewise planar object reconstruction

Polygonal Surface Reconstruction

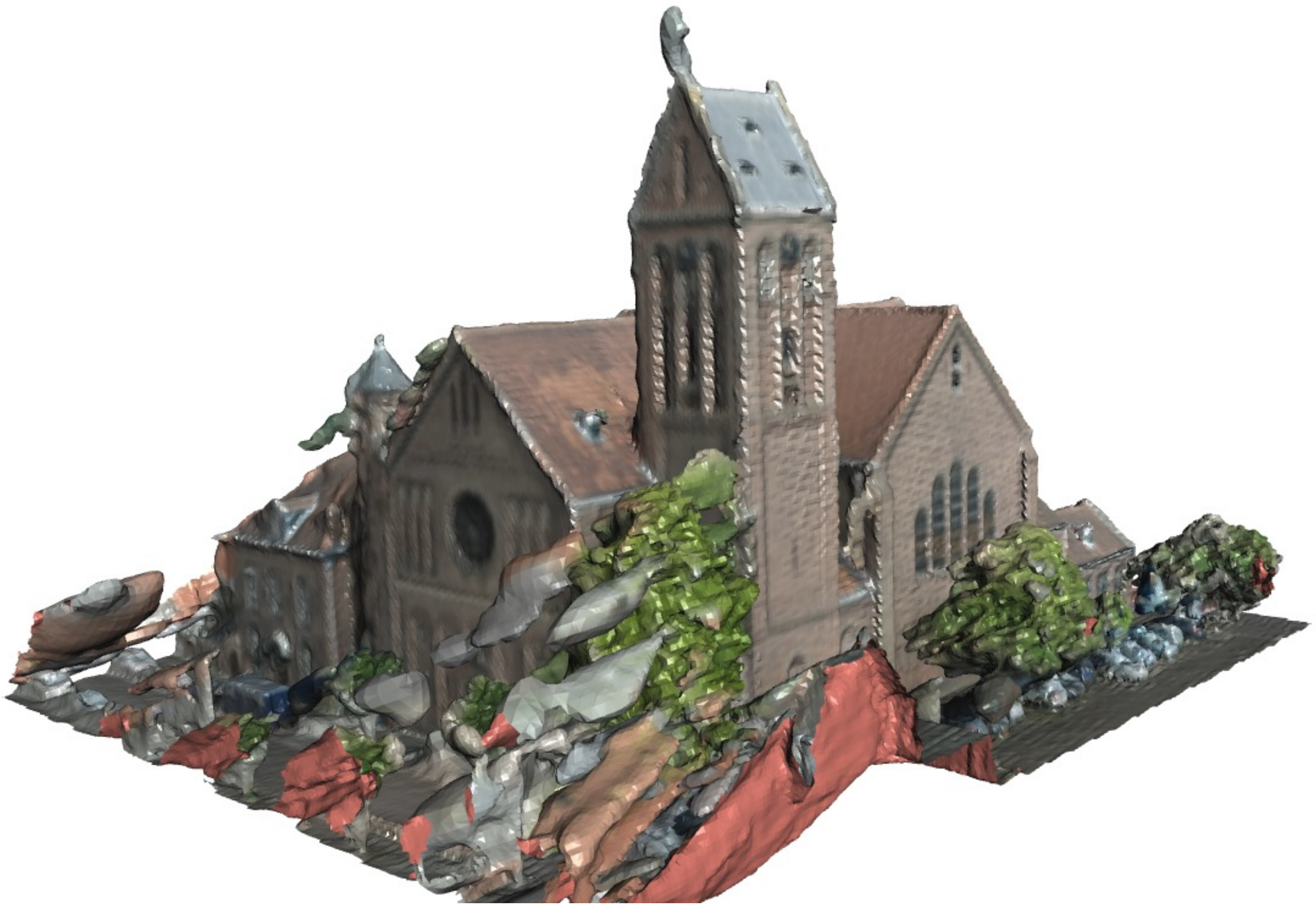
- Surface Reconstruction Methods
  - Smooth surfaces
    - Fit noisy data; robust to non-uniform distribution; fill (small) holes



Poisson Surface Reconstruction [Kazhdan *et al.* 06]

Polygonal Surface Reconstruction

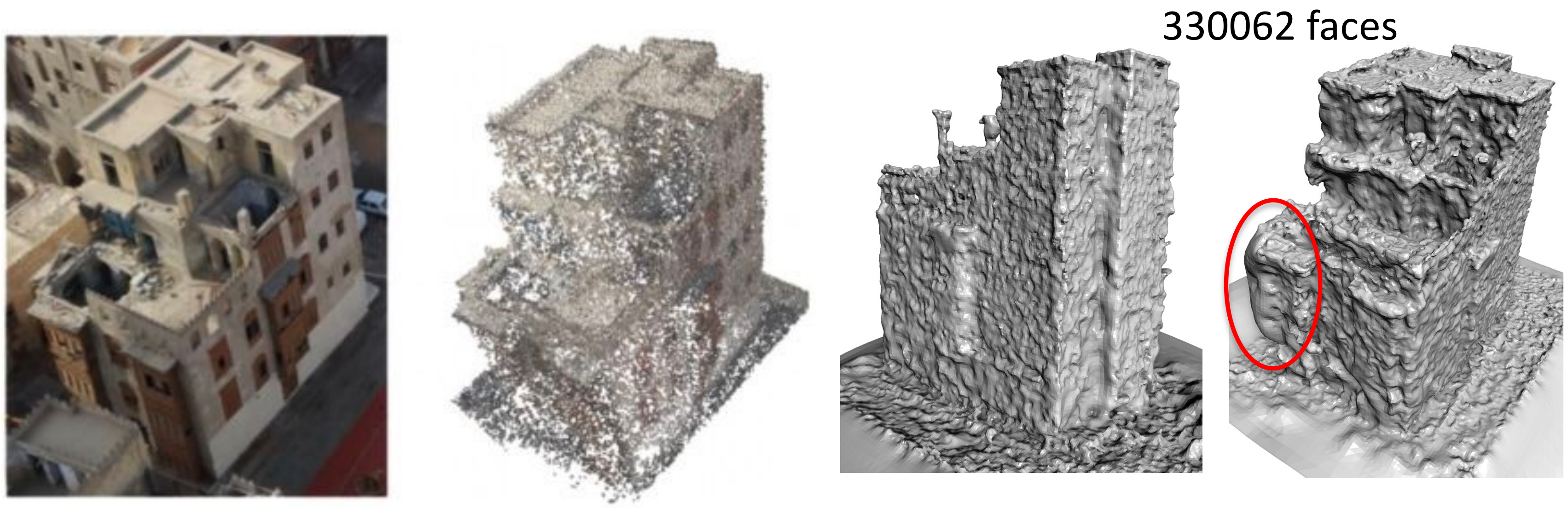
- Surface Reconstruction Methods
  - Smooth surfaces
  - Piecewise planar objects



Poisson surface reconstruction

Polygonal Surface Reconstruction

- Surface Reconstruction Methods
  - Smooth surfaces
  - Piecewise planar objects



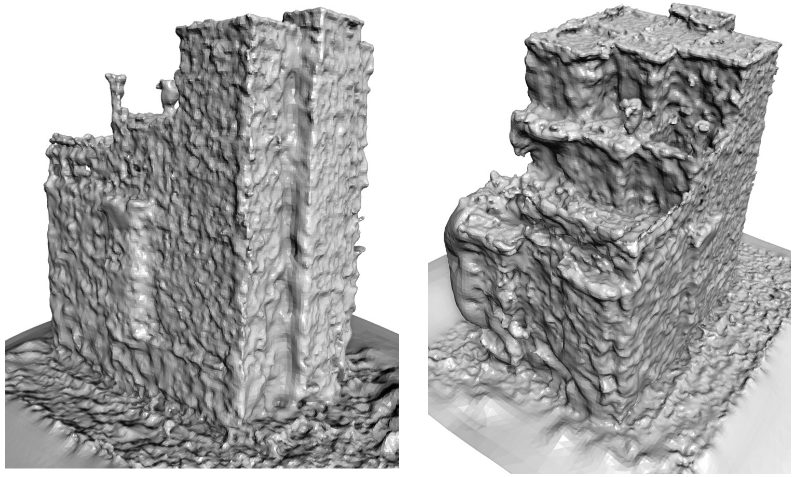
Result of [Kazhdan *et al.* 06]

- Unsatisfied results
  - Bumpy
  - Large number of faces
  - Unacceptable hole filling
- Rare direct applications
  - Post-processing required
    - Topologically correct
    - Simplified

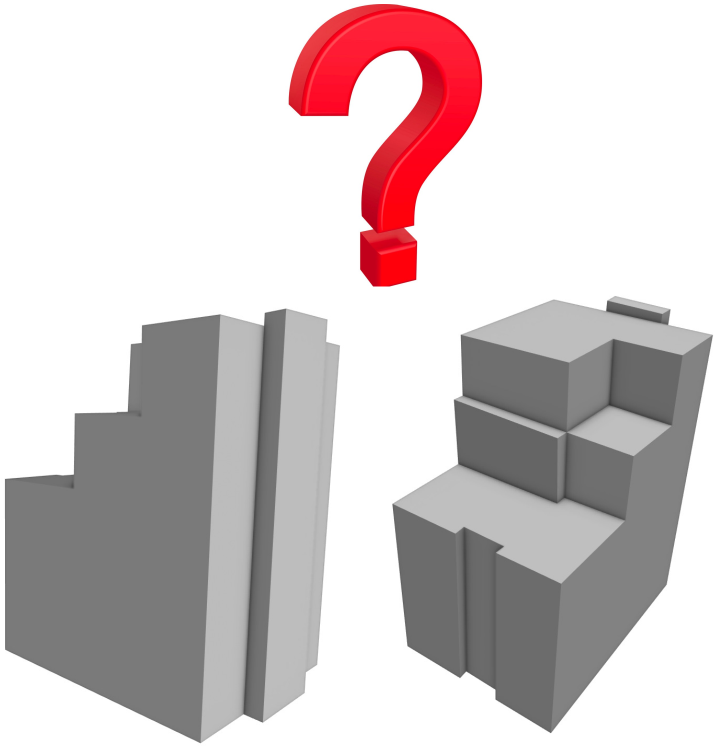
Polygonal Surface Reconstruction

- Surface Reconstruction Methods
  - Smooth surfaces
  - Piecewise planar objects

330062 faces



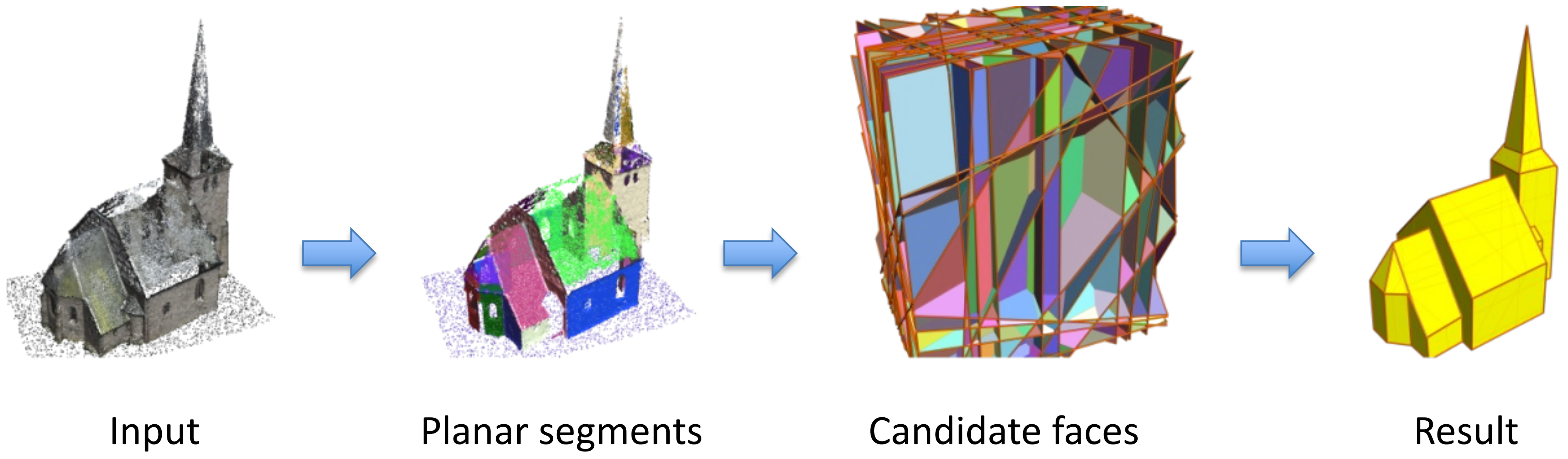
Result of [Kazhdan *et al.* 06]



[Nan and Wonka 17]

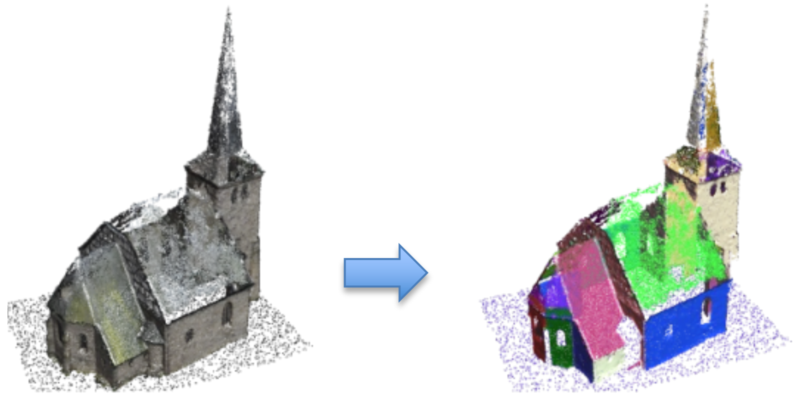
Polygonal Surface Reconstruction

- Overview



Polygonal Surface Reconstruction

- Plane extraction

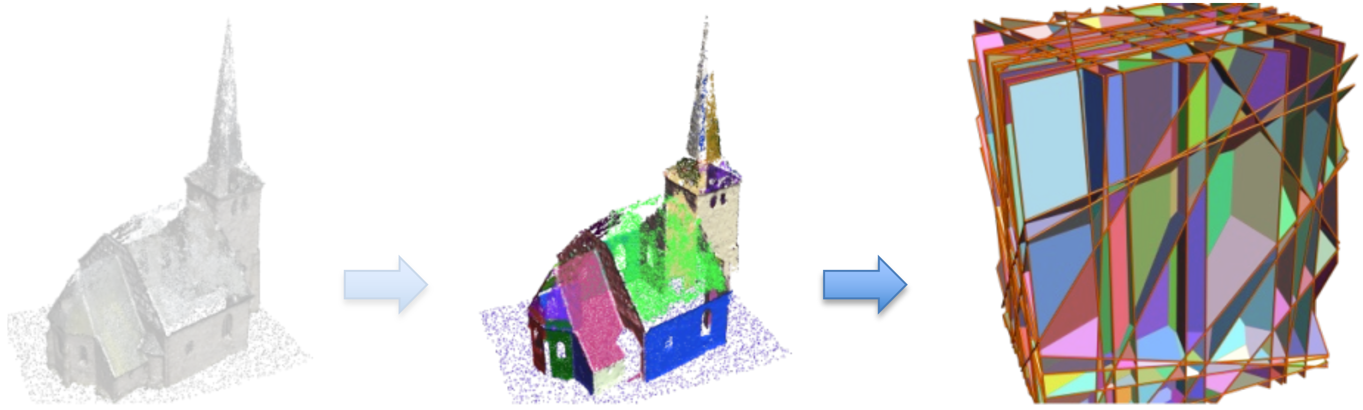


Input   Planar segments

RANSAC algorithm
- Random 3 points -> plane
- Scoring, accept or reject
- Repeat
  - Plane from the remaining points
  - Stop if no plane can be extracted

Polygonal Surface Reconstruction

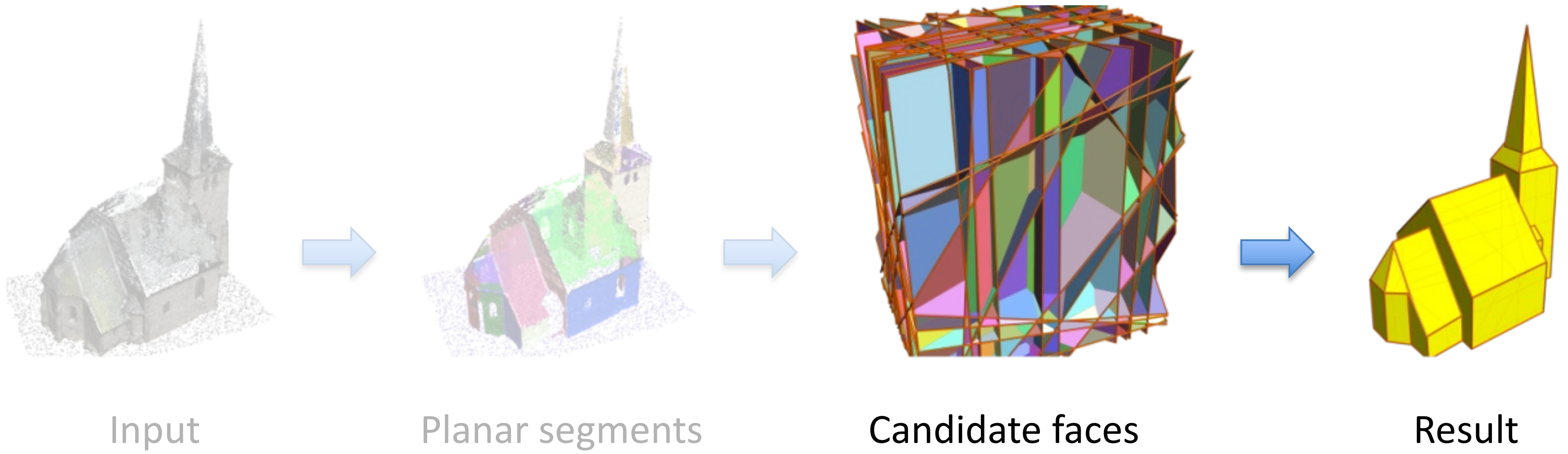
- Candidate generation
  - Supporting plane clipping
  - Pairwise intersection



Input　　Planar segments　　Candidate faces

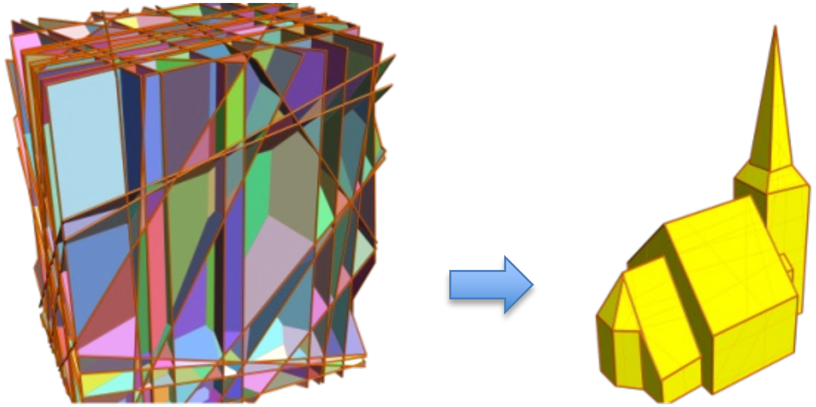
Polygonal Surface Reconstruction

- Face selection



Polygonal Surface Reconstruction

- Face selection
  - Labeling problem
  - Linear integer program



Candidate faces Result

N candidate faces $F = \{f_i | 1 \leq i \leq N\}$

$$\text{Variables: } x_i = \begin{cases} 1, & \text{face } f_i \text{ will be chosen} \\ 0, & \text{face } f_i \text{ will } \textbf{not} \text{ be chosen} \end{cases}$$

Polygonal Surface Reconstruction

- Objective function
  - Data fitting
    - Favors selecting faces with more support
    - Percentage of unused points

$$E_f = 1 - \frac{1}{|P|} \sum_{i=1}^{N} x_i \cdot support(f_i)$$

Polygonal Surface Reconstruction

- Objective function
  - Data fitting
    - Favors selecting faces with more support
    - Percentage of unused points

$$E_f = 1 - \frac{1}{|P|}\sum_{i=1}^{N} x_i \cdot support(f_i)$$

Confidence weighted number of supporting point

$$support(f) = \sum_{p,f|dist(p,f)<\varepsilon} (1 - \frac{dist(p,f)}{\varepsilon}) \cdot conf(p)$$

Polygonal Surface Reconstruction

- Objective function
  - Data fitting
    - Favors selecting faces with more support
    - Percentage of unused points

$$E_f = 1 - \frac{1}{|P|} \sum_{i=1}^{N} x_i \cdot support(f_i)$$

Confidence weighted number of supporting point

$$support(f) = \sum_{p,f|dist(p,f)<\varepsilon} \left(1 - \frac{dist(p,f)}{\varepsilon}\right) \cdot conf(p)$$

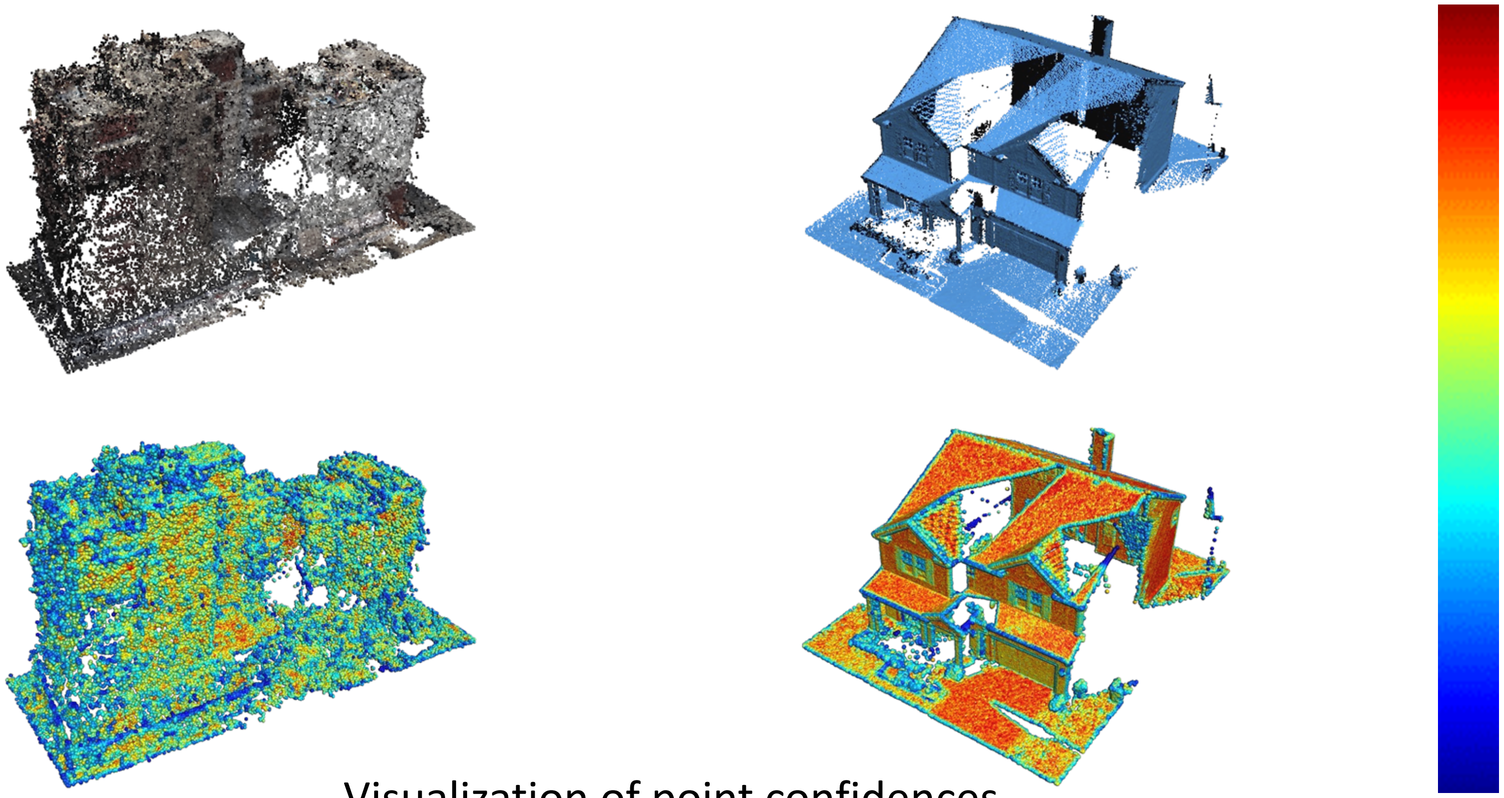
Point confidence

$$conf(p) = \frac{1}{3} \sum_{i=1}^{3} \left(1 - \frac{3\lambda_i^1}{\lambda_i^1 + \lambda_i^2 + \lambda_i^3}\right) \cdot \frac{\lambda_i^2}{\lambda_i^3} \qquad \lambda_i^1 \leq \lambda_i^2 \leq \lambda_i^3$$

planarity — local sampling uniformity

Polygonal Surface Reconstruction

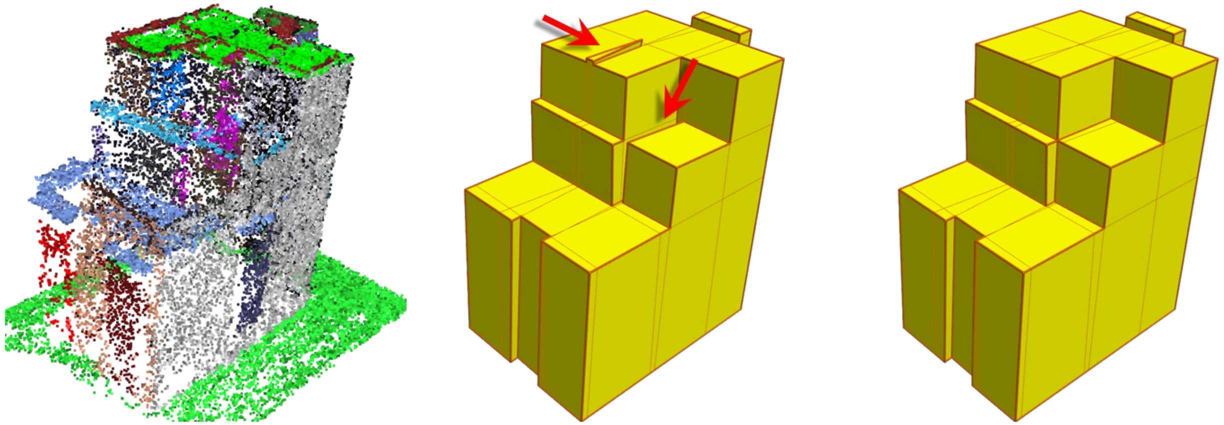
- Objective function
  - Data fitting



Visualization of point confidences

Polygonal Surface Reconstruction

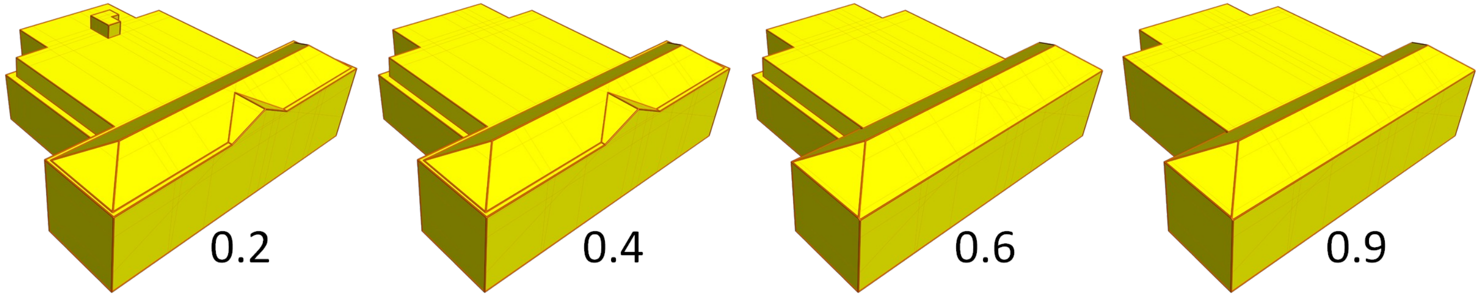
- Objective function
  - Data fitting
  - Model complexity
    - Penalize sharp corners



(a) (b) (c)

Polygonal Surface Reconstruction

- Objective function
  - Data fitting
  - Model complexity
    - Penalize sharp corners

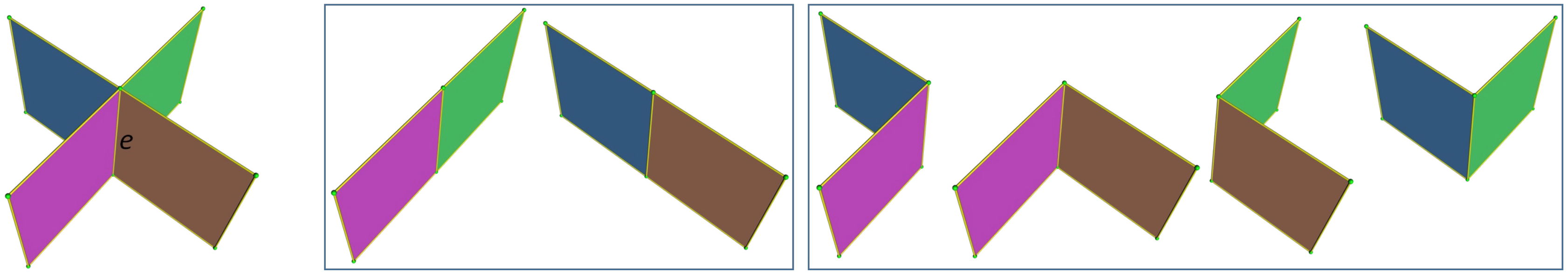


(a) (b) (c) (d)

Polygonal Surface Reconstruction

- Objective function
  - Data fitting
  - Model complexity
    - Penalize sharp corners

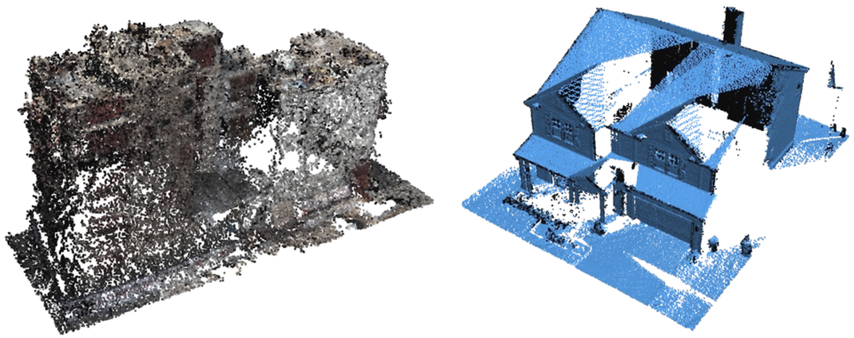
$$E_m = \frac{1}{|E|} \sum_{i=1}^{|E|} corner(e_i)$$

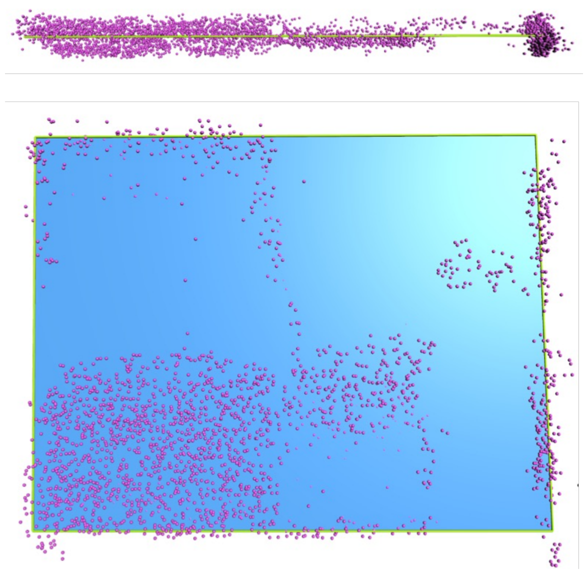


Intersecting two faces

Polygonal Surface Reconstruction

- Objective function
  - Data fitting
  - Model complexity
  - Point coverage

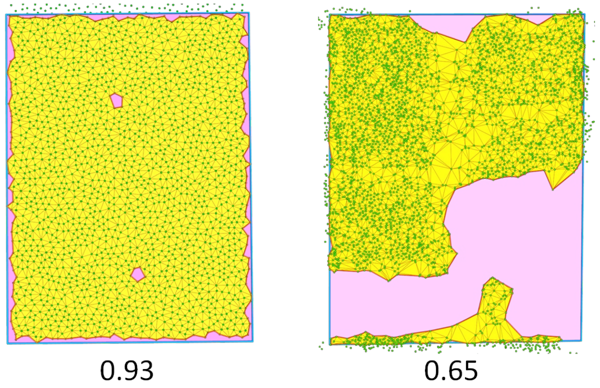




Polygonal Surface Reconstruction

- Objective function
  - Data fitting
  - Model complexity
  - Point coverage

$$E_c = \frac{1}{area(M)} \sum_{i=1}^{N} x_i \cdot (area(f_i) - area(M_i^{\alpha})),$$



Polygonal Surface Reconstruction

- Face selection
  - Linear integer program

$$\min_{\mathbf{x}} \quad \lambda_f \cdot \underline{E_f} + \lambda_m \cdot \underline{E_m} + \lambda_c \cdot \underline{E_c}$$

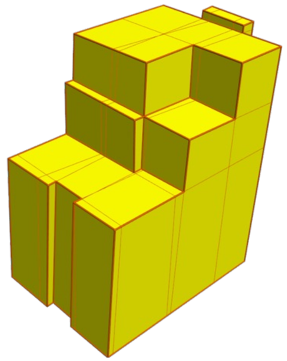
Data fitting

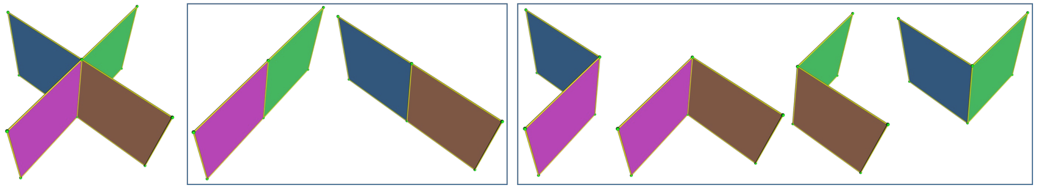
Model complexity

Point coverage

Polygonal Surface Reconstruction

- Face selection
  - Linear integer program
    - Constraints
      - Watertight
      - Manifold





$$\min_{\mathbf{x}} \quad \lambda_f \cdot E_f + \lambda_m \cdot E_m + \lambda_c \cdot E_c$$

$$\text{s.t.} \quad \begin{cases} \sum_{j \in \mathcal{N}(e_i)} x_j = 2 \quad \text{or} \quad 0, & 1 \leq i \leq |E| \\ x_i \in \{0, 1\}, & 1 \leq i \leq N \end{cases}$$

Polygonal Surface Reconstruction

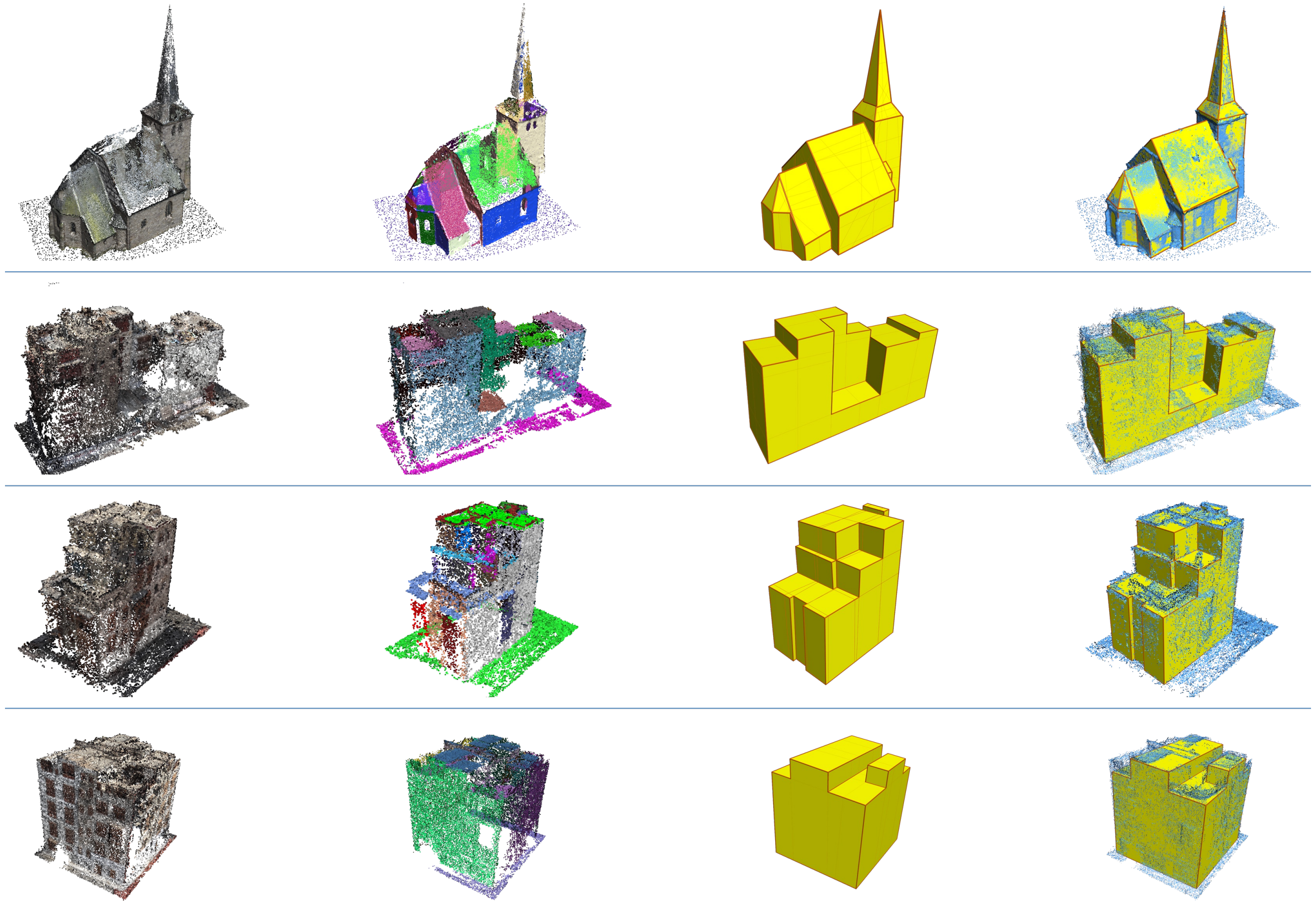
- Face selection
  - Linear integer program
    - Constraints
    - Solvers (SCIP, CBC, GLPK, Gurobi…)

$$\min_{\mathbf{x}} \quad \lambda_f \cdot E_f + \lambda_m \cdot E_m + \lambda_c \cdot E_c$$

$$\text{s.t.} \quad \begin{cases} \sum_{j \in \mathcal{N}(e_i)} x_j = 2 \quad \text{or} \quad 0, & 1 \leq i \leq |E| \\ x_i \in \{0, 1\}, & 1 \leq i \leq N \end{cases}$$

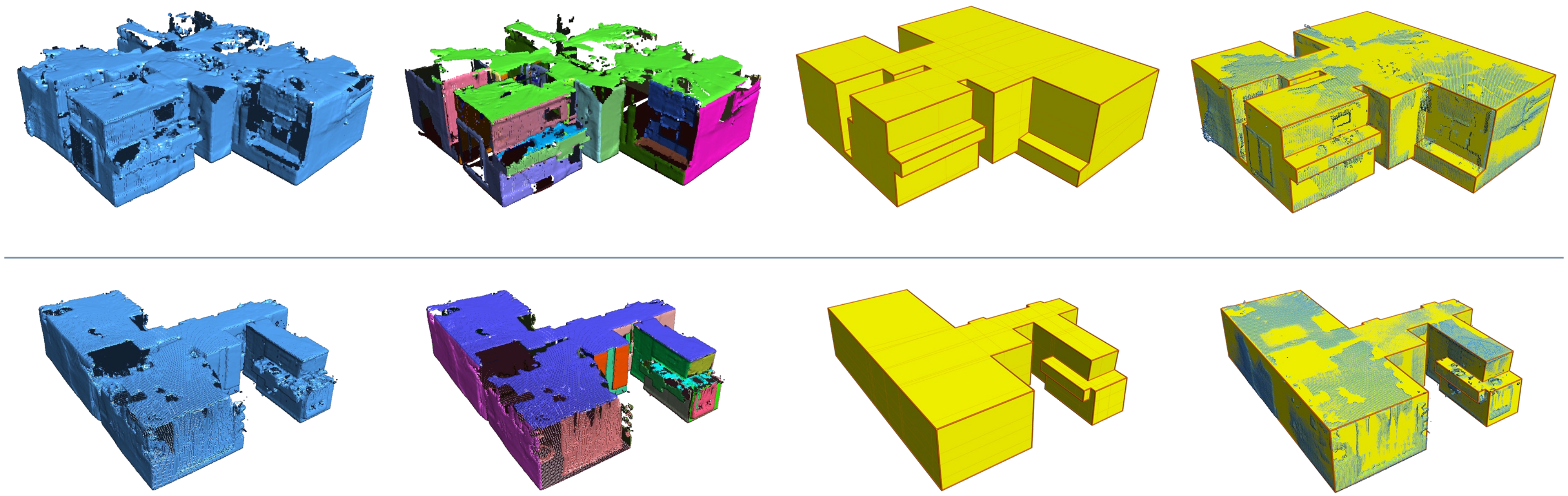
Polygonal Surface Reconstruction

- Reconstruction results



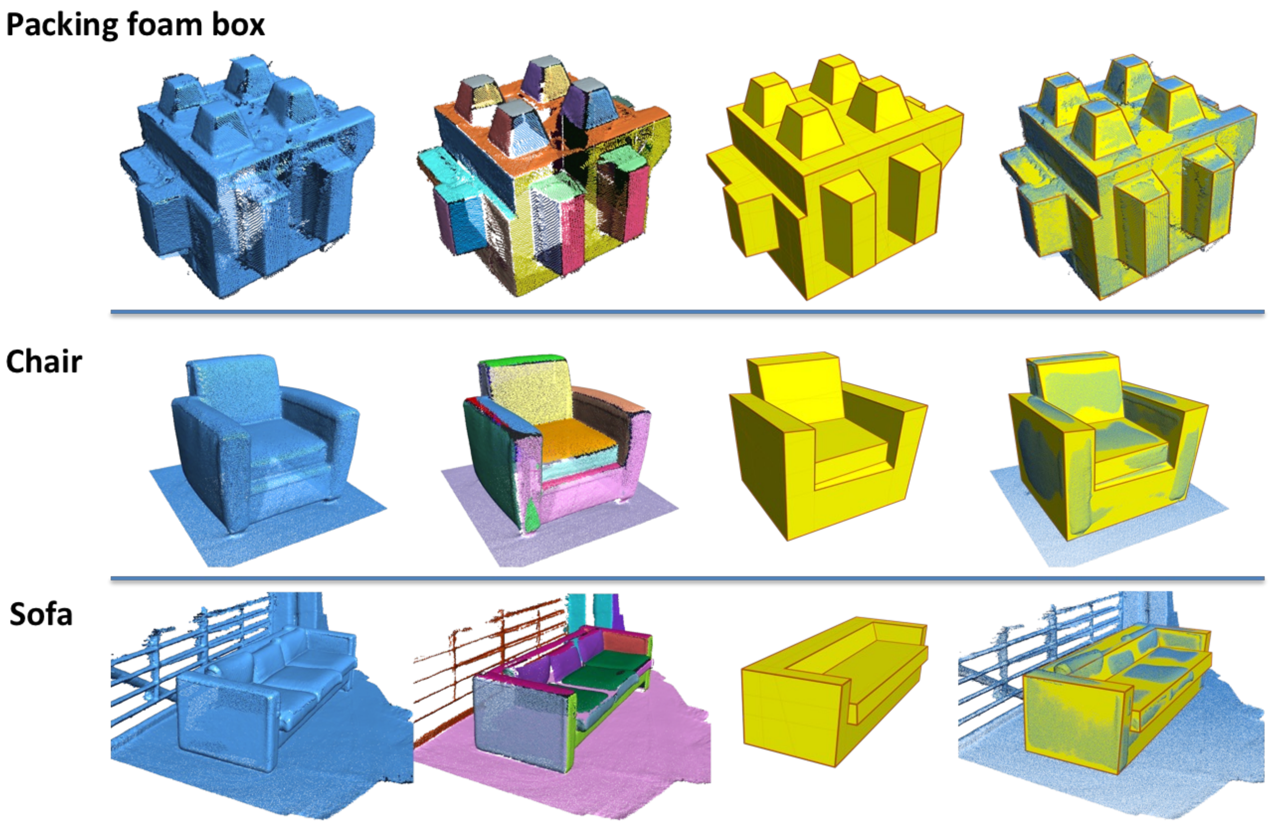
Polygonal Surface Reconstruction

- Reconstruction results



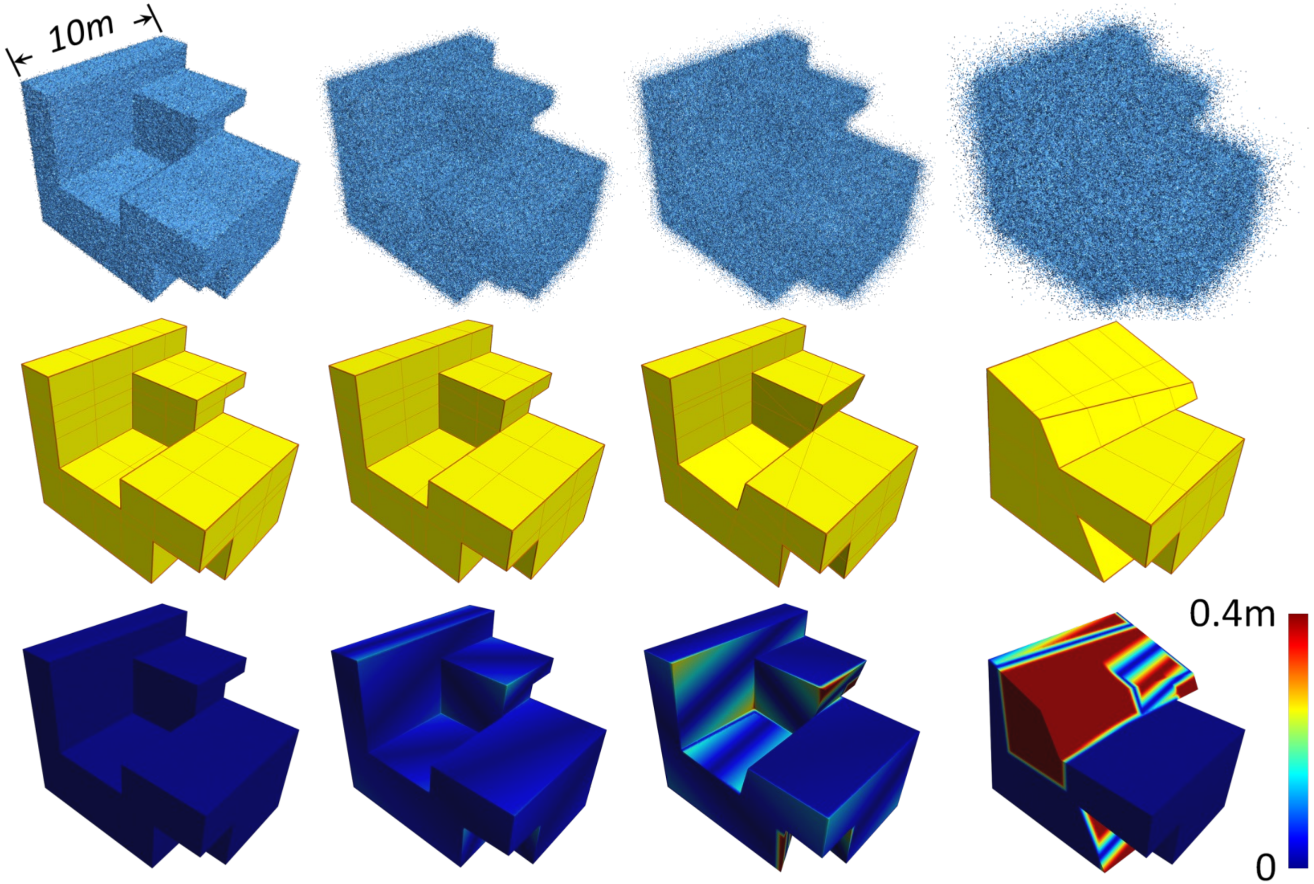
Piecewise Planar Reconstruction

- Reconstruction results



Piecewise Planar Reconstruction

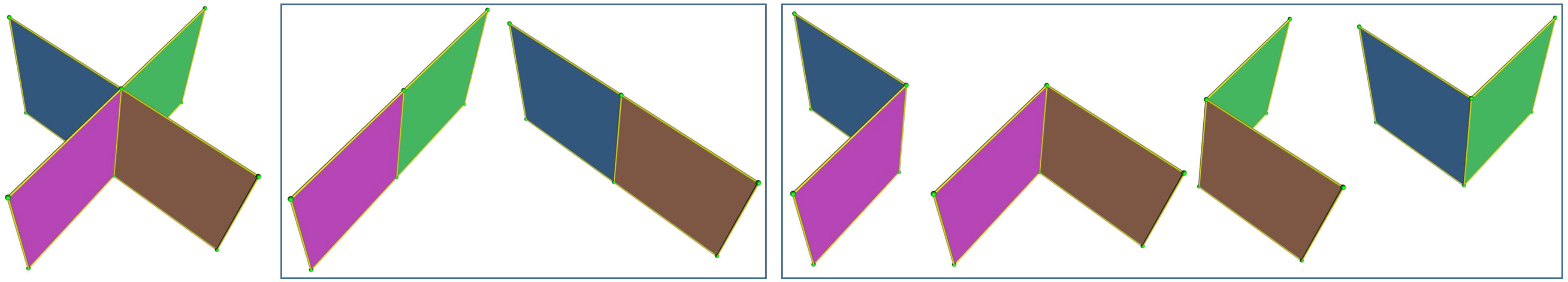
- Robustness to noise



(a) $\sigma = 0.1m$ (b) $\sigma = 0.4m$ (c) $\sigma = 0.6m$ (d) $\sigma = 0.8m$

Limitations

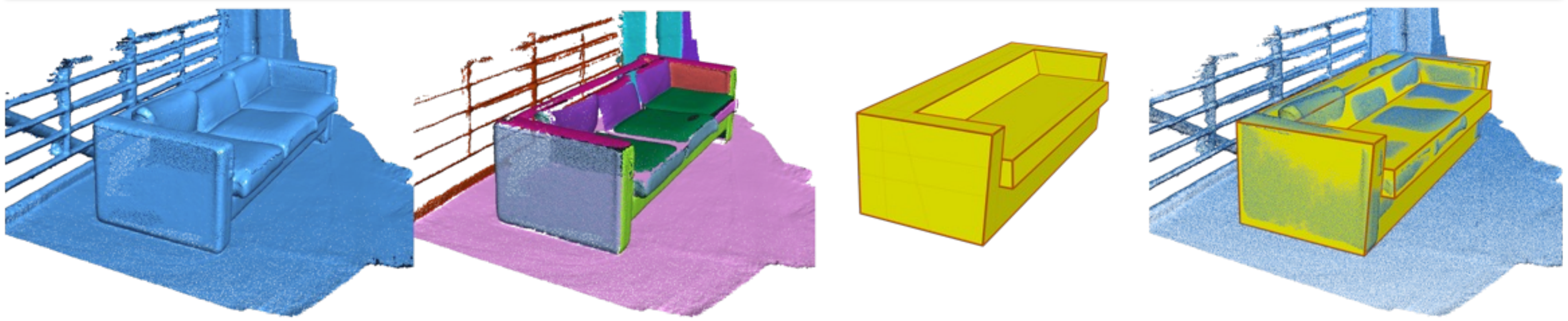
- Open surfaces



$$\text{s.t.} \quad \begin{cases} \sum_{j \in \mathcal{N}(e_i)} x_j = 2 \quad \text{or} \quad 0, & 1 \leq i \leq |E| \\ x_i \in \{0, 1\}, & 1 \leq i \leq N \end{cases}$$

Limitations

- Open surfaces



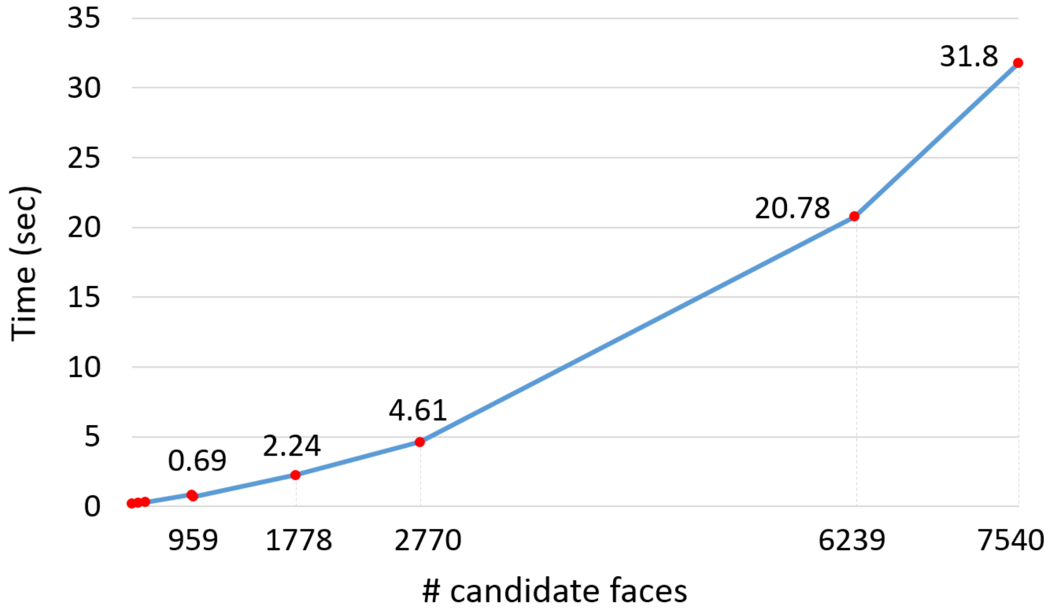
$$\text{s.t.} \quad \begin{cases} \sum_{j \in \mathcal{N}(e_i)} x_j = 2 \quad \text{or} \quad 0, & 1 \leq i \leq |E| \\ x_i \in \{0, 1\}, & 1 \leq i \leq N \end{cases}$$

Limitations

- Open surfaces
- Finer surface details
  - Fence
  - Façade decorations
  - Door handle
  - …

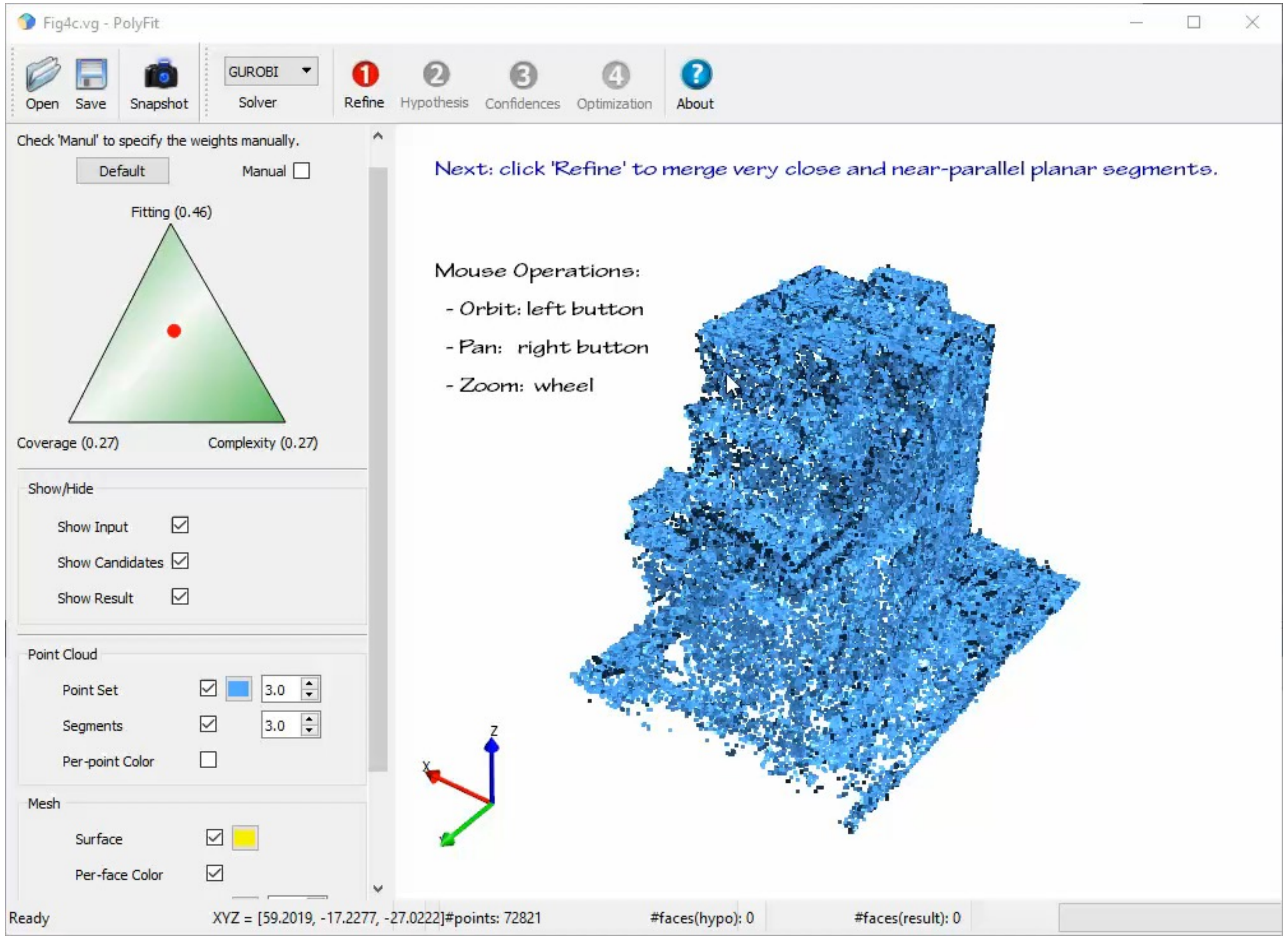
Limitations

- Open surfaces
- Finer surface details
- Complexity of the algorithm



Polygonal Surface Reconstruction

- Demo



Source Code (in C++) https://github.com/LiangliangNan/PolyFit

Next

- Feedback on the assignments
- Example questions (for the final exam)