Nuages de Points et Modélisation 3D

3 - From local properties to surface reconstruction

Alexandre Boulch (that's me)

CV

- Senior scientist at Valeo in the team valeo.ai.
- Researcher at ONERA
- Thesis at ENPC

Research

- 3D understanding of scenes
  - From point clouds
  - From images

Björn Michele

CV

- PhD student in the valeo.ai team & IRISA OBELIX lab.
- Master in Computer Science

Research

- Domain Adaptation for 3D data
  - Mostly for point clouds

valeo.ai

“… as soon as it works, no one calls it AI anymore.”
Attributed to John McCarthy (mathematician)

“AI is a collective name for problems which we do not yet know how to solve properly by computer”
Attributed to Bertram Raphael (computer scientist)

“AI does not exists!”
Luc Julia (Scientific Director, Renault Group)

valeo.ai

Numpy

Optimization

Points

Neural networks

Segmentation

Machine learning

Pytorch

Geometry

Surfaces

Differentiable optimization

Scikit-learn

Random forests

Classification

Images

Overview

Machine learning courses

• Surface reconstruction	Today
• Descriptors and machine learning • Image based processing	ML course 1
• Geometric deep learning	ML course 2
• Convolutional and Transformer based architectures	ML course 3
• Tasks and corresponding architectures	ML course 4

Overview - Evaluation

Machine learning courses

● Surface reconstruction	Today
● Descriptors and machine learning ● Image based processing	**ML course 1**
● Geometric deep learning	ML course 2
● Convolutional and Transformer based architectures	ML course 3
● Tasks and corresponding architectures	**ML course 4**
● **QCM** + opening session	

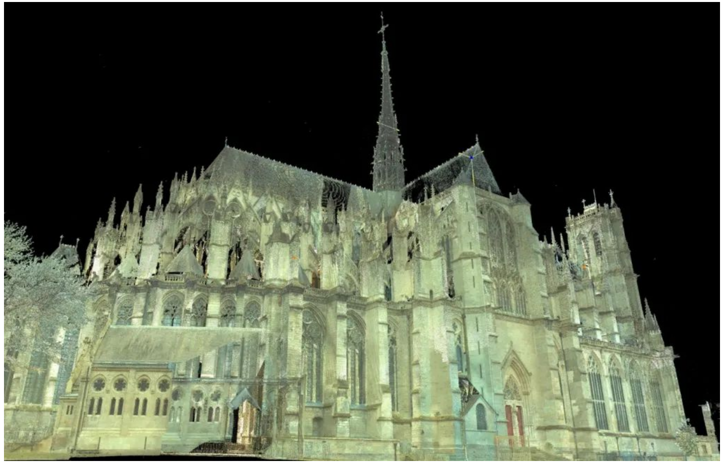
Overview

- Local features
- Surface reconstruction
- Model segmentation

I - From point clouds to surfaces

3D rendering

Patrimony saving

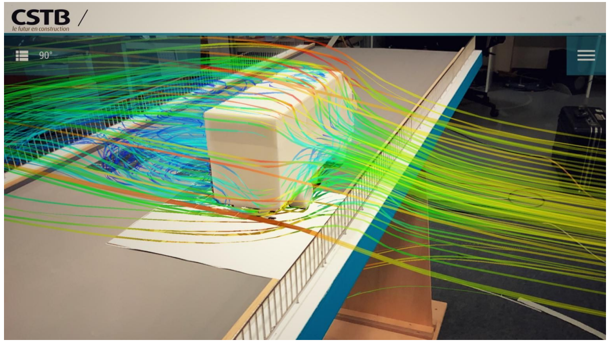


Amiens cathedral

Archeology

3D modelling in archaeology: The application of Structure from Motion methods to the study of the megalithic necropolis of Panoria

Simulations

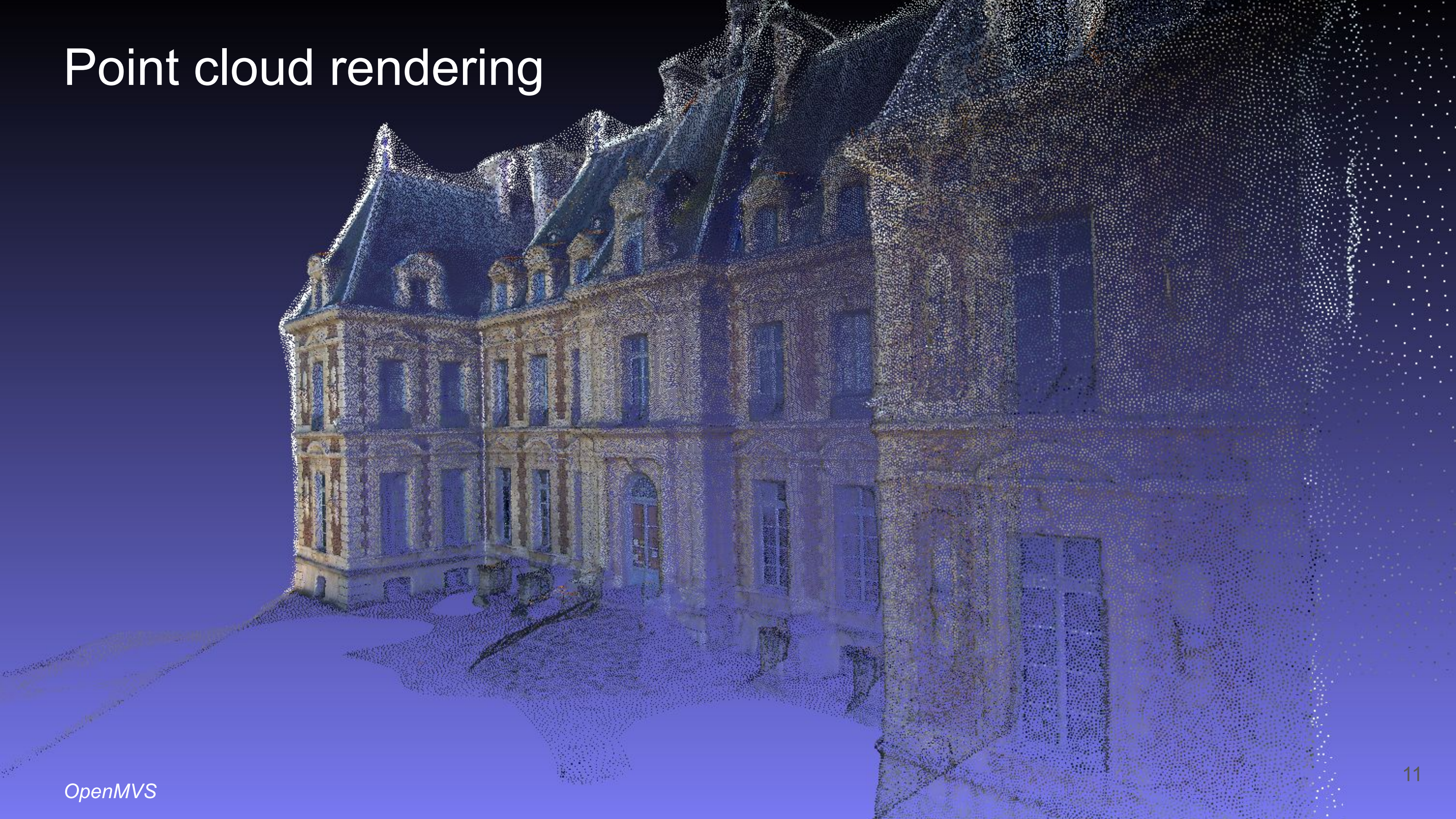


CSTB

Games

Cyberpunk 2077 - Technology preview

Point cloud rendering



OpenMVS

valeo.ai

Point clouds rendering

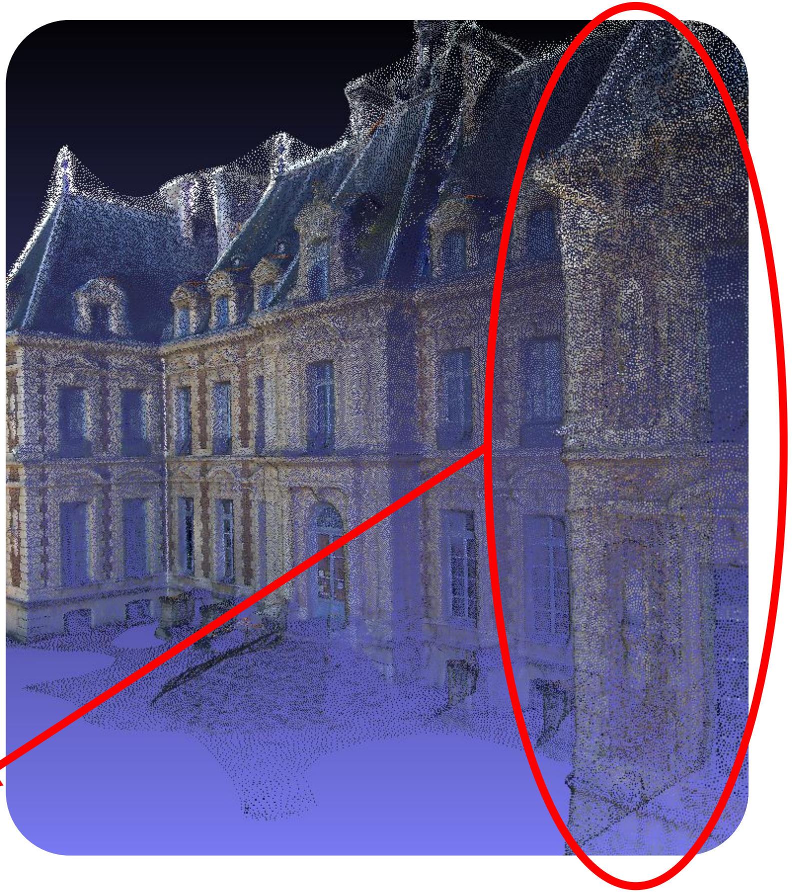
I - From point cloud to surfaces

Point clouds are simple:

$$P = \{x \in \mathbb{R}^3\}$$

However:

- Points are independent
- Not easy to render



Wrong point size
Wrong density
⇒ see through

Point clouds rendering

I - From point cloud to surfaces

Point clouds are simple:

$$P = \{x \in \mathbb{R}^3\}$$

However:

- Points are independent
- Not easy to render

Point clouds

I - From point cloud to surfaces

Point clouds are simple:

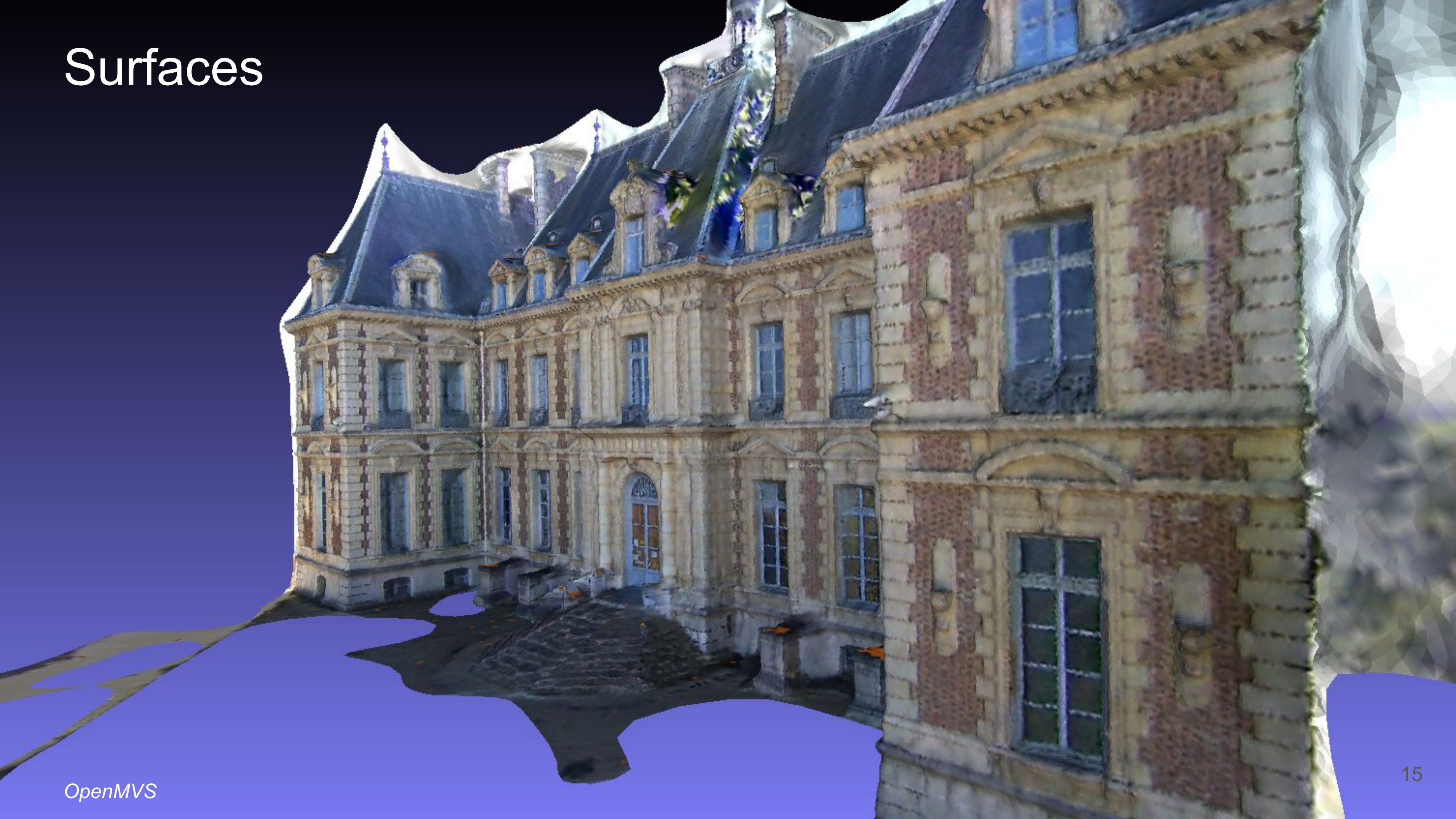
$$P = \{x \in \mathbb{R}^3\}$$

However:

- Points are independent
- Not easy to render

⇒ **Last course:** point rendering is not dead (but it requires a bit of work)

Surfaces



OpenMVS


unity

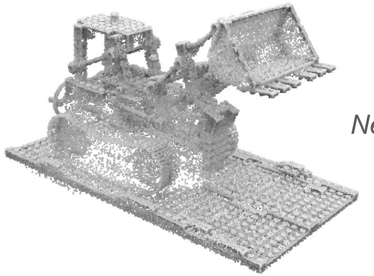
What are point clouds?

A point cloud:

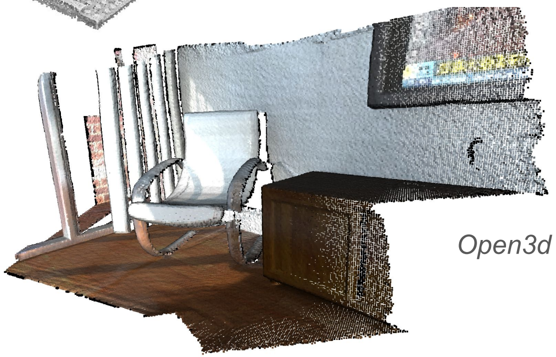
- A set of 3D coordinates

$$P = \{p \in \mathbb{R}^3\}$$

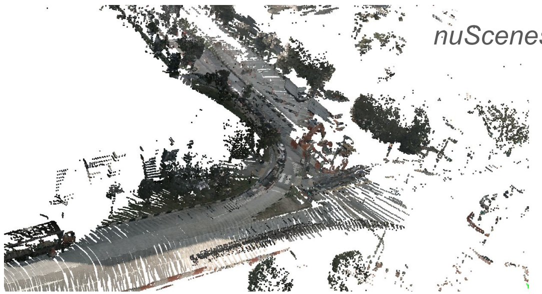
- No obvious order (at first sight)
- Sparse sample of surface
- Noisy / outliers
- Variation of size: several orders of magnitude



NeRF



Open3d



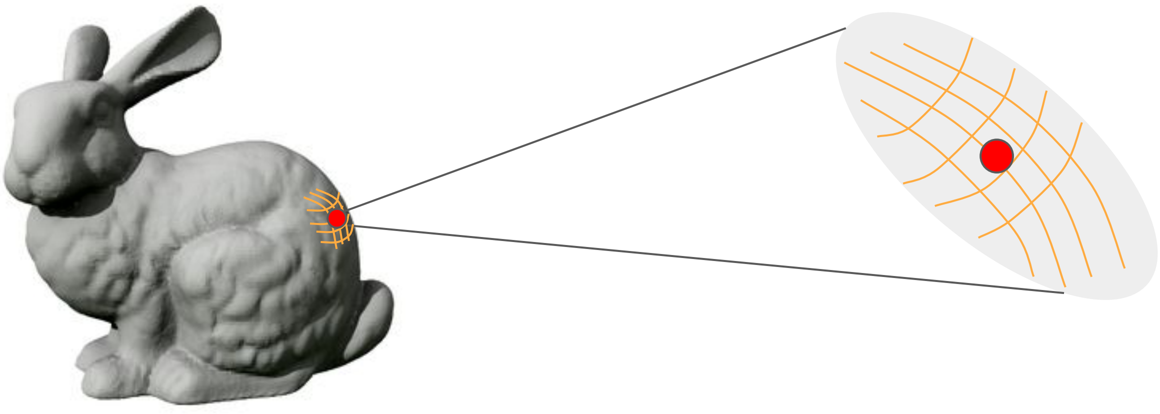
nuScenes

What is a surface

I - From point cloud to surfaces

A surface is a 2-manifold.

Locally, a manifold behave like the euclidean space, i.e, continuous.



Stanford bunny

The neighborhood is homeomorphic to a 2D-euclidean space (open 2D ball)

valeo.ai

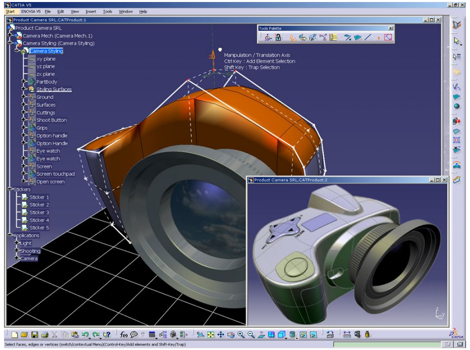
Why surfaces?

I - From point cloud to surfaces

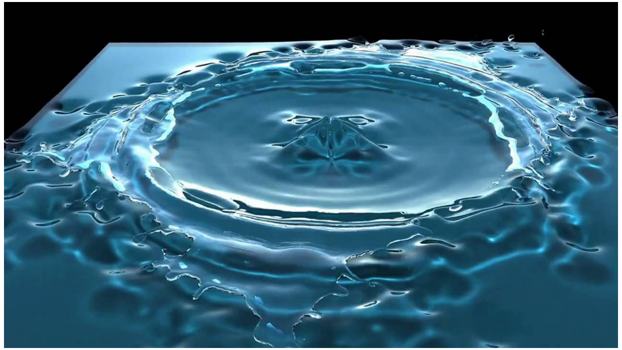
Surfaces for:

- Simulation
- Animation
- Design
- …

Blender



Catia



A Multiscale Approach to Mesh-based Surface Tension Flows

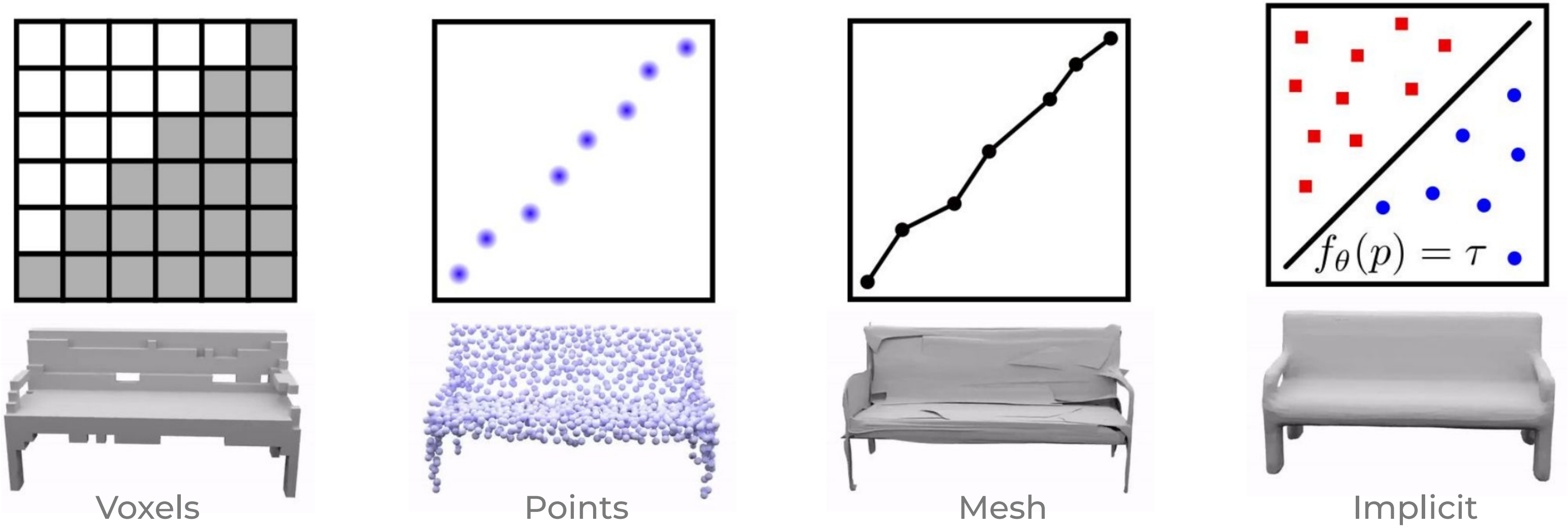
How to represent a surface

- Points
- Meshes
- Voxels
- Implicit representations
- Parametric shapes (planes, cylinder, spheres)
- Gaussians
- …

→ no unified / perfect representation

How to represent a surface?

I - From point cloud to surfaces

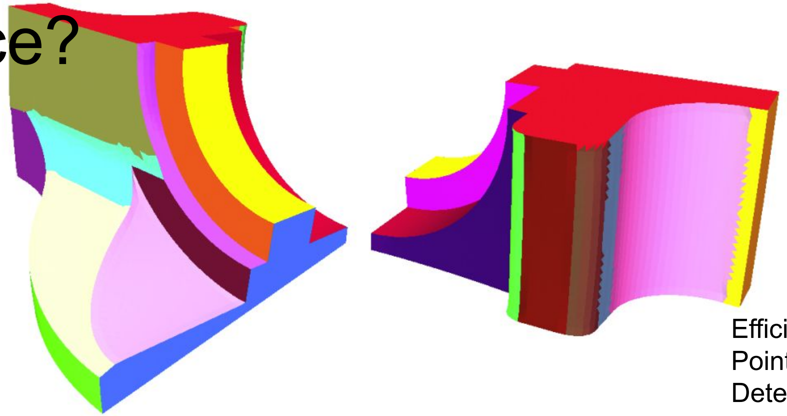


Occupancy Networks: Learning 3D Reconstruction in Function Space
Mescheder, Lars and Oechsle, Michael and Niemeyer, Michael and Nowozin, Sebastian and Geiger, Andreas
Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2019

valeo.ai

How to represent a surface?

I - From point cloud to surfaces



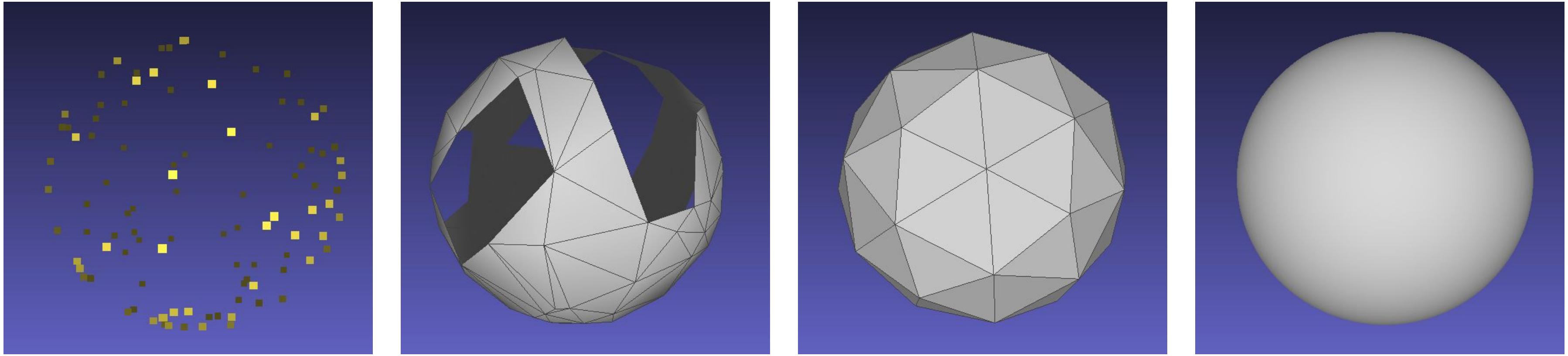
Efficient RANSAC for Point-Cloud Shape Detection

3DGS - Improving Gaussian splatting with Localized points management

What are the properties we want?

I - From point cloud to surfaces

- Sticking to the points?
- With holes?
- Abstract?
- Regular?
- Made of planes? ...



II - Local features

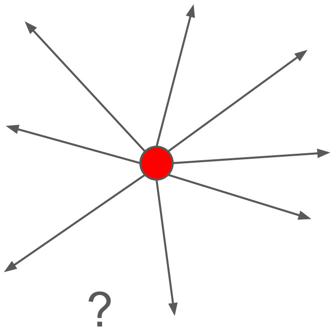
III - Local features

A - Normal estimation

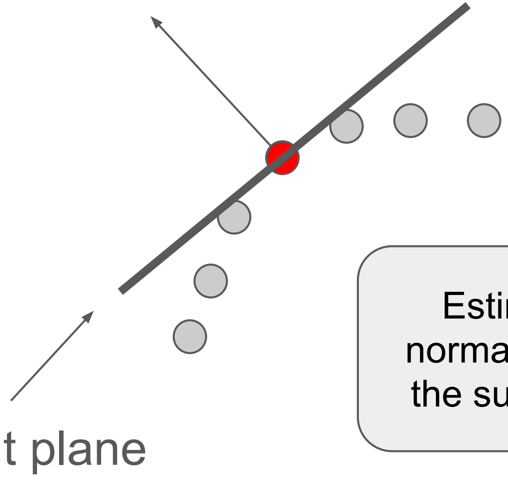
A - Normal estimation

III - Local features

A single point does not contain orientation information



Need to consider a neighborhood around a point



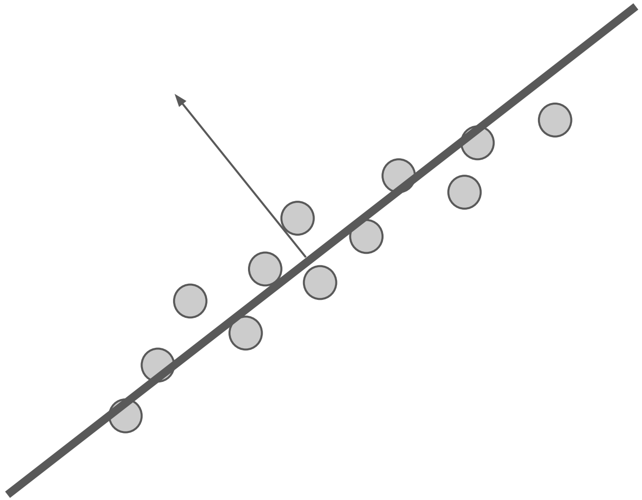
Estimating the normal ⇔ knowing the surface locally

A - Normal estimation

III - Local features

Plane fitting

- The plane is defined by the directions of the largest variance
- The normal corresponds to the direction with lowest variance
- **Principal Component Analysis**
- The normal is the eigenvector for the smallest eigenvalue.



A - Normal estimation

Plane fitting with PCA

- Compute covariance matrix

Average: $\bar{x} = \frac{1}{n} \sum_{x \in P} x$

Covariance: $Cov \in \mathbb{R}^3 \times \mathbb{R}^3$

$$Cov(i, j) = \frac{1}{n} \sum_{x \in P} (x_i - \bar{x}_i)(x_j - \bar{x}_j) = \frac{1}{n}{}^{\top}XX$$

A - Normal estimation

Plane fitting with PCA

- Compute covariance matrix

- Diagonalize the Matrix, with P orthonormal (Cov is positive, real, symmetric)

$$Cov = PDP^{-1}$$

$$D = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \qquad P = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}$$

A - Normal estimation

Plane fitting with PCA

- Compute covariance matrix
- Diagonalize the Matrix
- Find the lowest eigenvalue and eigenvector $\lambda_1 \geq \lambda_2 \geq \lambda_3$

$$D = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \boxed{\lambda_3} \end{pmatrix} \qquad P = \begin{bmatrix} e_1 & e_2 & \boxed{e_3} \end{bmatrix}$$

A - Normal estimation

Plane fitting with PCA

- Compute covariance matrix
- Diagonalize the Matrix
- Find the lowest eigenvalue and eigenvector

$$\mathbf{n} = \overrightarrow{n} = \overrightarrow{e}_3$$

III - Local features

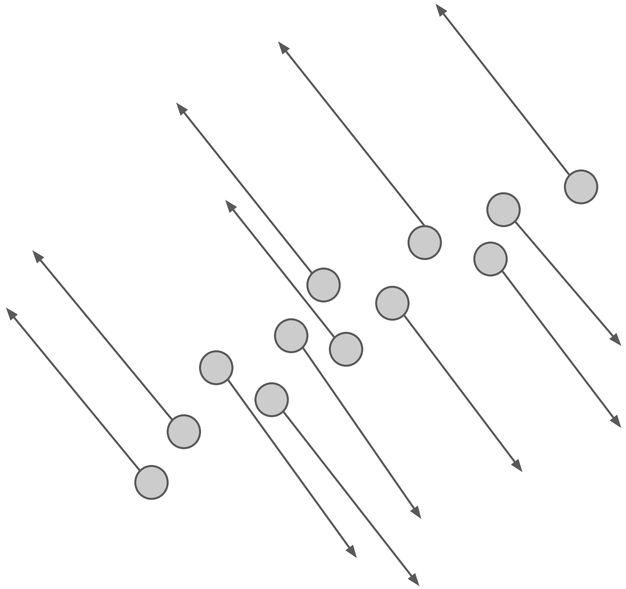
B - Normal orientation

B - Normal orientation

The plane fitting algorithm:

✓ direction

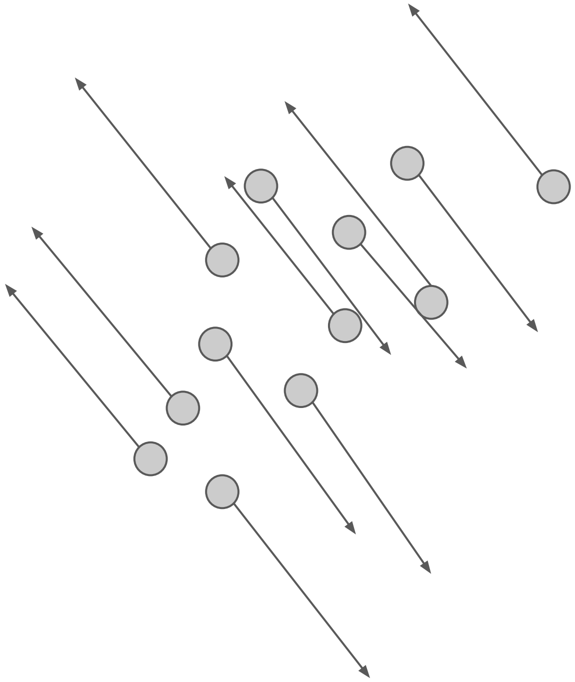
✗ orientation



B - Normal orientation

Normal orientation with minimal spanning tree

Idea: propagate the orientation from one seed

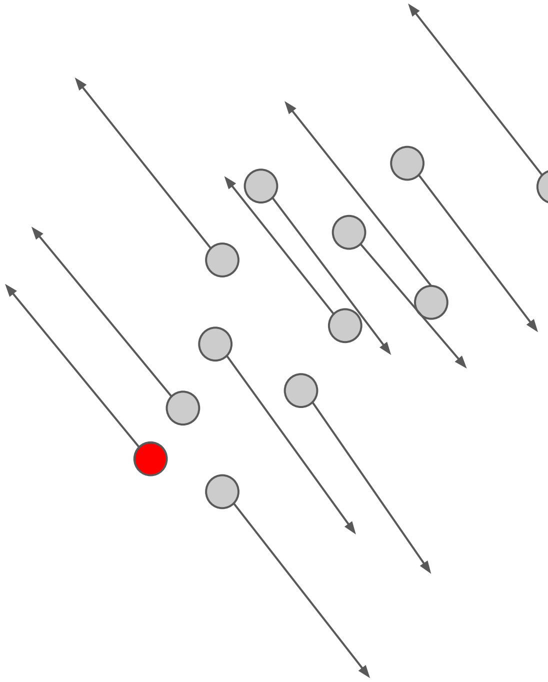


B - Normal orientation

Normal orientation with minimal spanning tree

Idea: propagate the orientation from one seed

1. Select a seed



B - Normal orientation

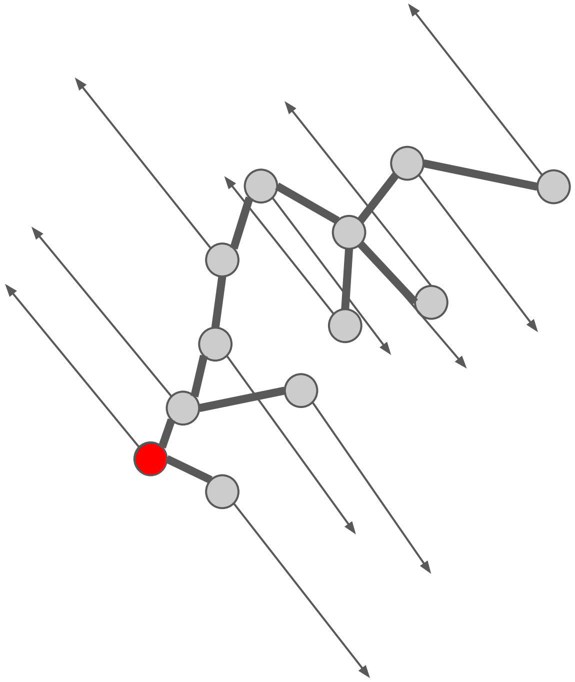
III - Local features

valeo.ai

Normal orientation with minimal spanning tree

Idea: propagate the orientation from one seed

1. Select a seed
2. Build a spanning tree



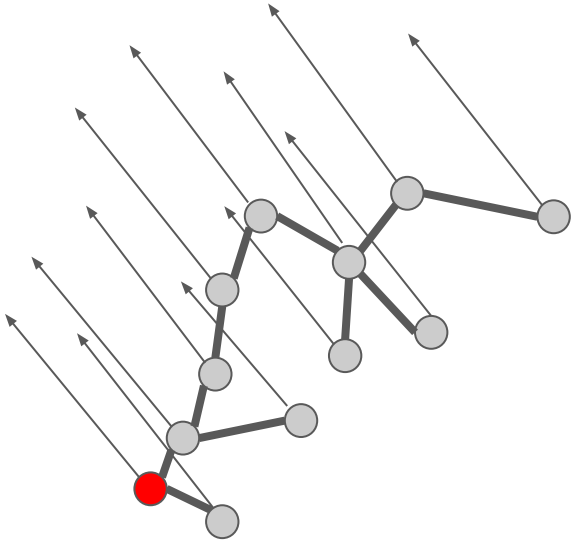
B - Normal orientation

valeo.ai

Normal orientation with minimal spanning tree

Idea: propagate the orientation from one seed

1. Select a seed
2. Build a spanning tree
3. Consistently orient the normals (inner product > 0) from parent to child

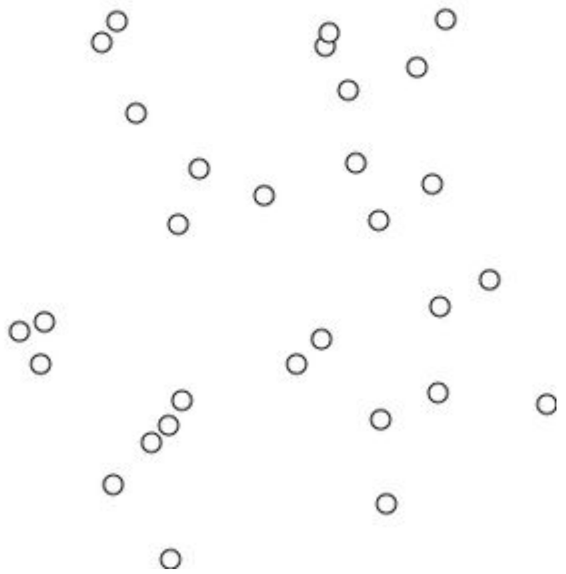


B - Normal orientation

III - Local features

The Kruskal algorithm

- Compute the KNN graph (KDTree)
- Sort the edges of the graph (increasing distances)
- Loop on all edges
    - Add the edge if it does not create a loop



Kruskal algorithm, Wikipedia

III - Surface reconstruction

III - Surface reconstruction

A - Ball pivoting

Ball pivoting

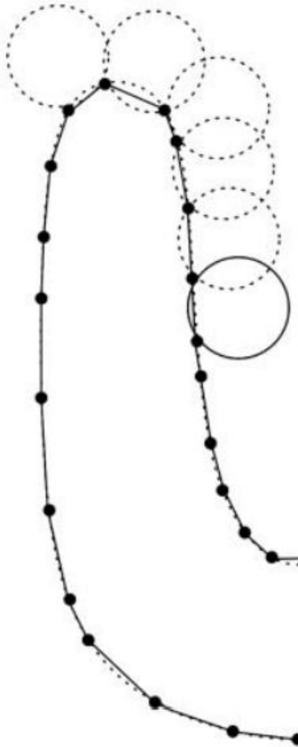
Principles:

Make a sphere “roll” on the points

2D: When the sphere touch 2 points, create an edge

3D: When the sphere touch 3 points, create a facet

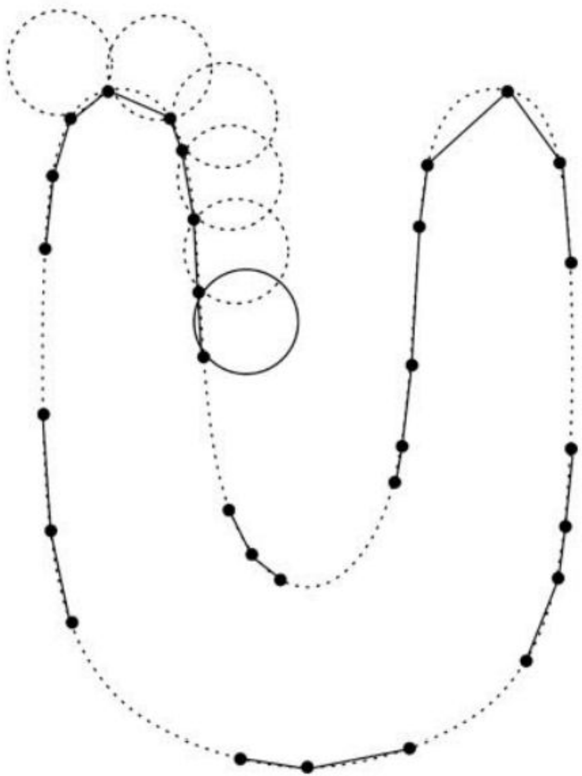
Iterate over the unexplored edges



Ball pivoting

III - Surface reconstruction

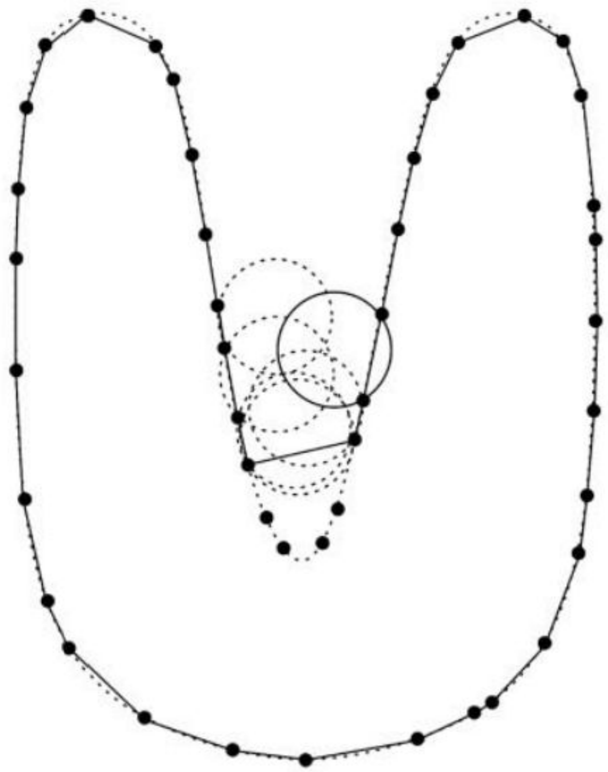
Space larger than sphere diameter creates a hole.



Ball pivoting

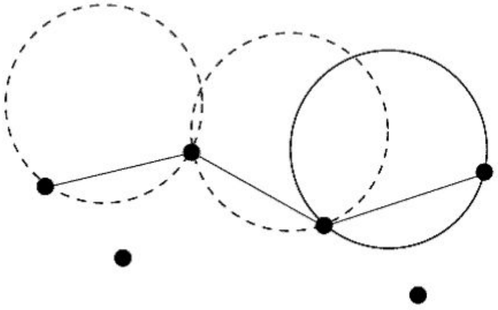
III - Surface reconstruction

Sphere with to big radius leads to a loss of details

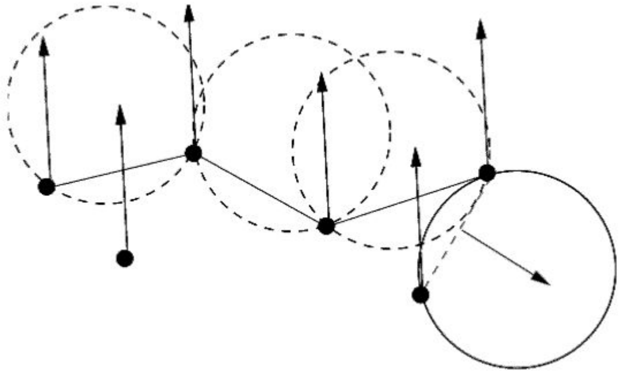


Ball pivoting

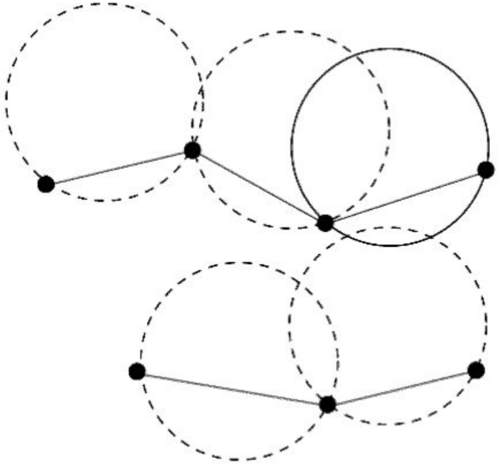
III - Surface reconstruction



Presence of noise



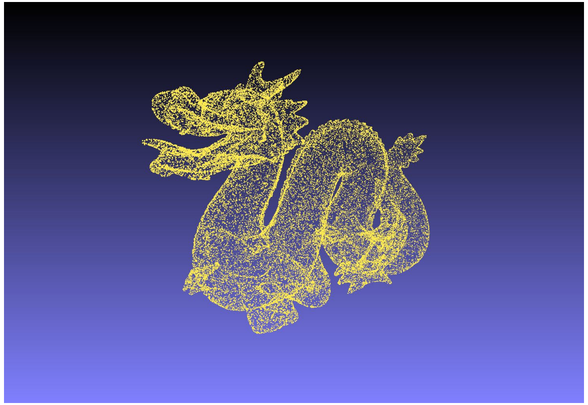
Filter according to normals

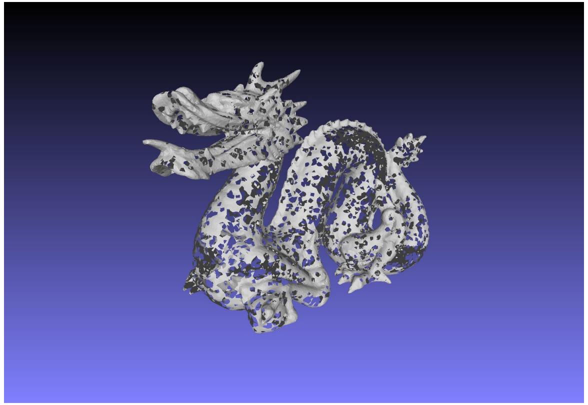


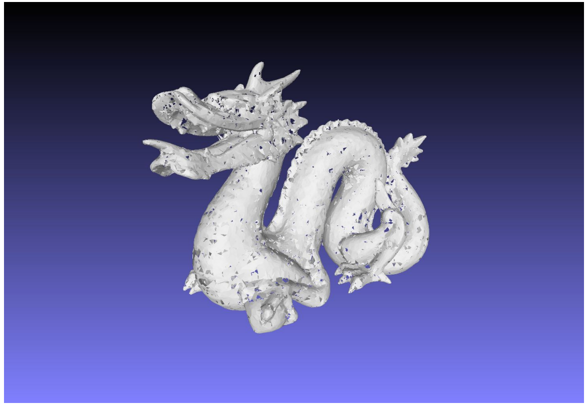
Double side surface

Ball pivoting

III - Surface reconstruction







Increasing sphere radius

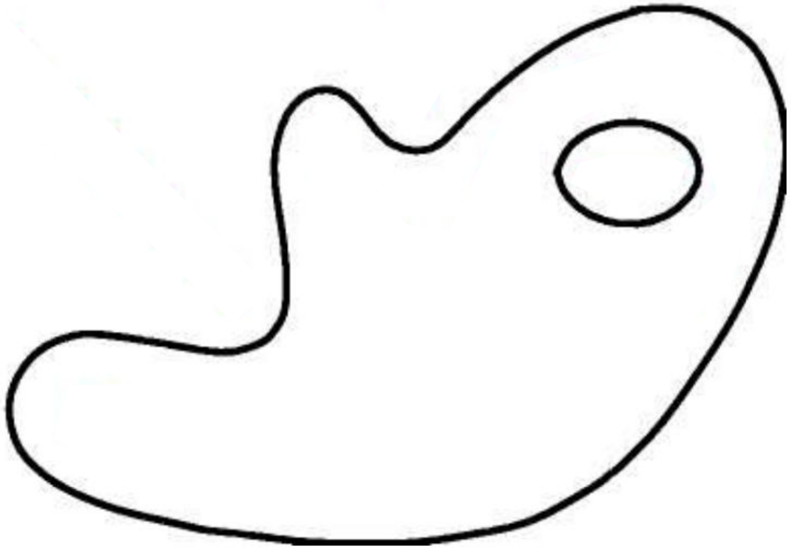
III - Surface reconstruction

B - Delauney reconstruction

Surface

B - Delauney reconstruction

Consider an abstract surface 2D surface (no hole).



A New Voronoi-Based Surface Reconstruction Algorithm, Amenta et al.

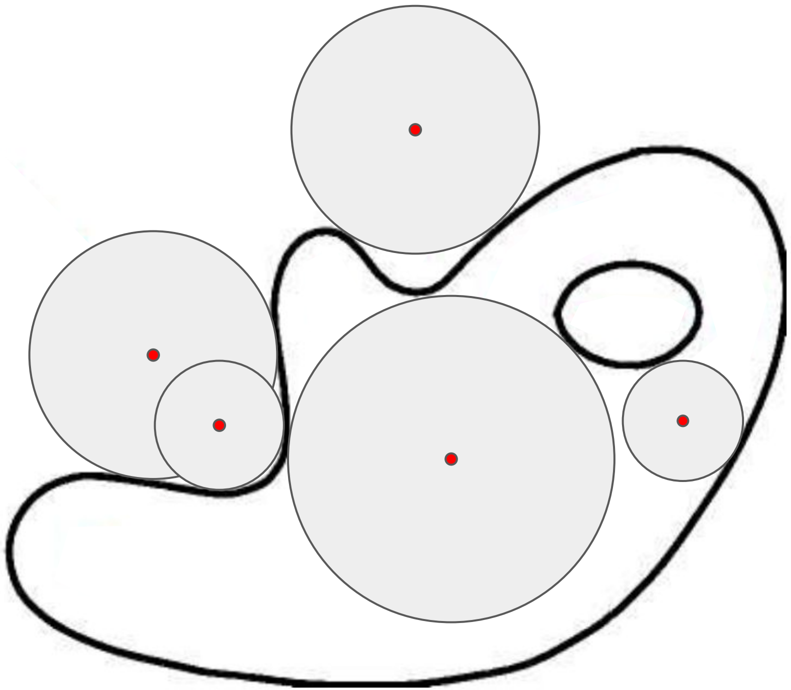
Medial axis

B - Delauney reconstruction

Let's put a sphere such that:

- It touches at least to points on the surface
- It does not include part of the surface

Let's consider all the possible spheres



A New Voronoi-Based Surface Reconstruction Algorithm, Amenta et al.

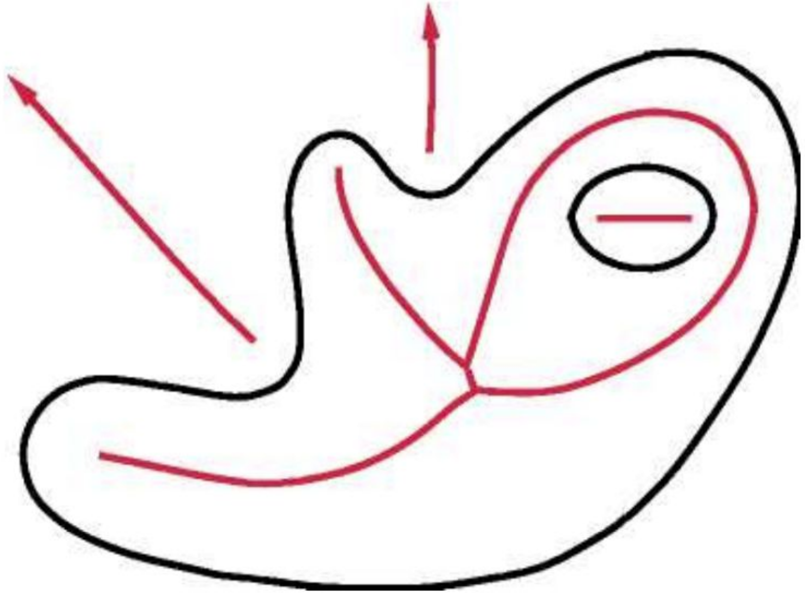
Medial axis

B - Delauney reconstruction

The set of centers of these spheres is the **medial axis**

The medial axis is the dual of the surface.

⇒ from a medial axis, it is possible to find the surface



A New Voronoi-Based Surface Reconstruction Algorithm, Amenta et al.

Medial axis

B - Delauney reconstruction

The medial axis generalizes to 3D

A New Voronoi-Based Surface Reconstruction Algorithm, Amenta et al.

Voronoï diagram

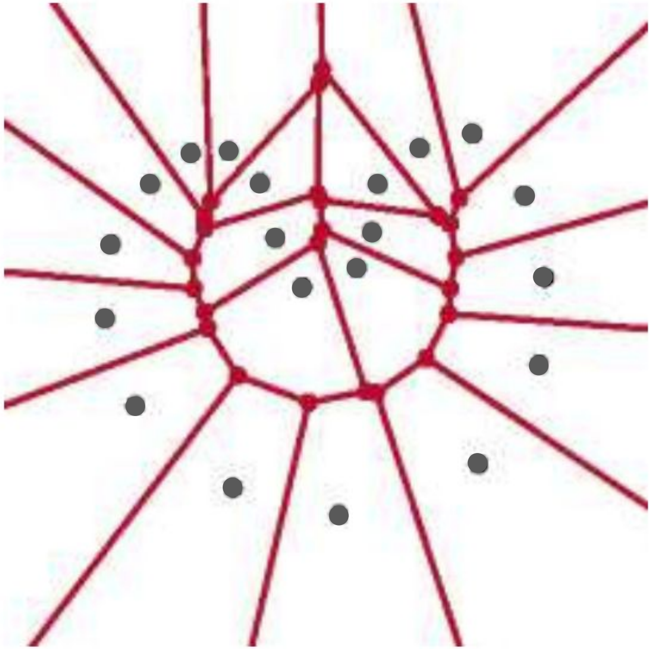
B - Delauney reconstruction

Is there an equivalent of the medial axis for point clouds?

⇒ yes this the Voronoï diagram

On an edge → equal distance to 2 points

On a node → equal distance to 3 points



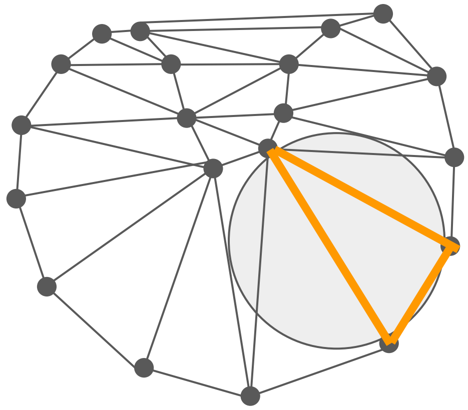
A New Voronoi-Based Surface Reconstruction Algorithm, Amenta et al.

Voronoï diagram

B - Delauney reconstruction

Given 3 points, create a triangle if the sphere encompassing the 3 points is empty.

Iterate over the triplets.



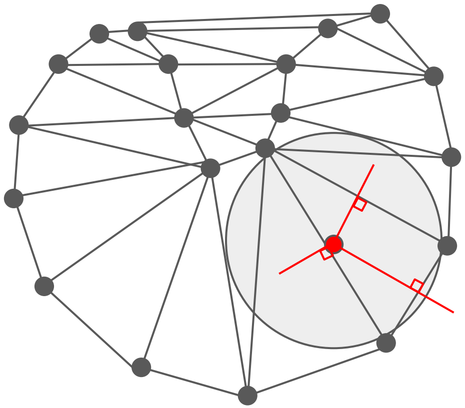
A New Voronoi-Based Surface Reconstruction Algorithm, Amenta et al.

Voronoï diagram

B - Delauney reconstruction

Create the graph.

- nodes → the center of the spheres
- Edges → mediator of the edges of the triangles



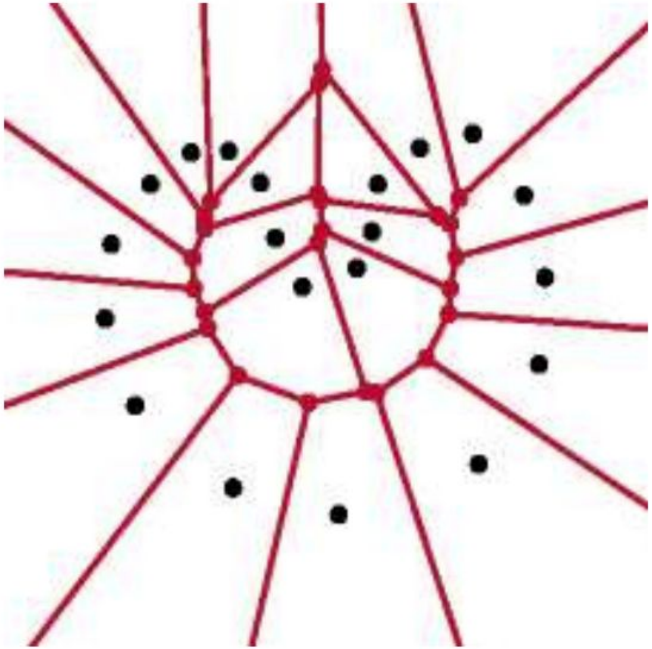
A New Voronoi-Based Surface Reconstruction Algorithm, Amenta et al.

Voronoï diagram

B - Delauney reconstruction

Create the graph.

- nodes → the center of the spheres
- Edges → mediator of the edges of the triangles



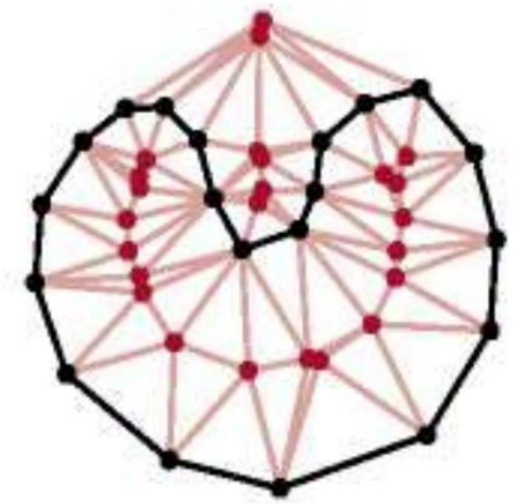
Augmented Delauney triangulation

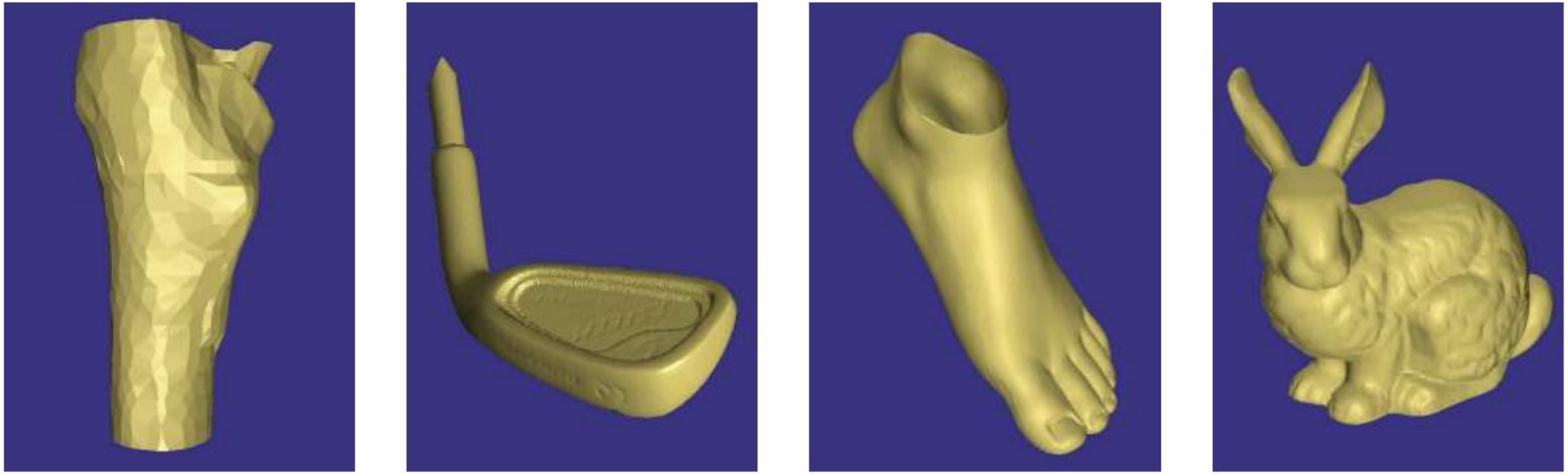
B - Delauney reconstruction

Recompute the Delaunay triangulation with the Voronoï nodes.

Remove edges linked to Voronoï nodes

The remaining is the **crust**.





Pros: theoretically grounded, very good results without noise

Cons: not smooth, cannot handle noise

III - Surface reconstruction

C - Poisson reconstruction

Overview

C - The Poisson reconstruction pipeline

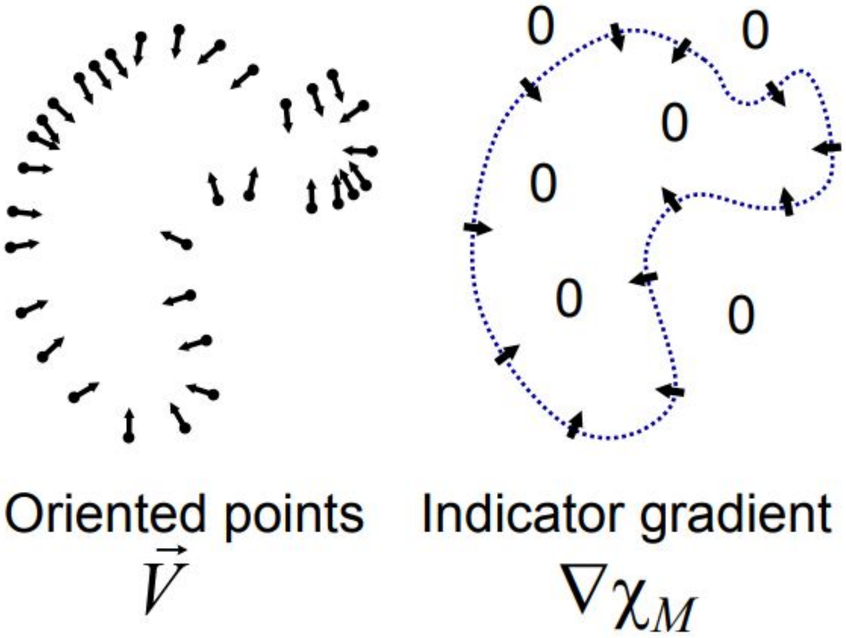
Oriented points

$\vec{V}$

Kazhdan et al. "Poisson surface reconstruction." SGP, 2006.

Overview

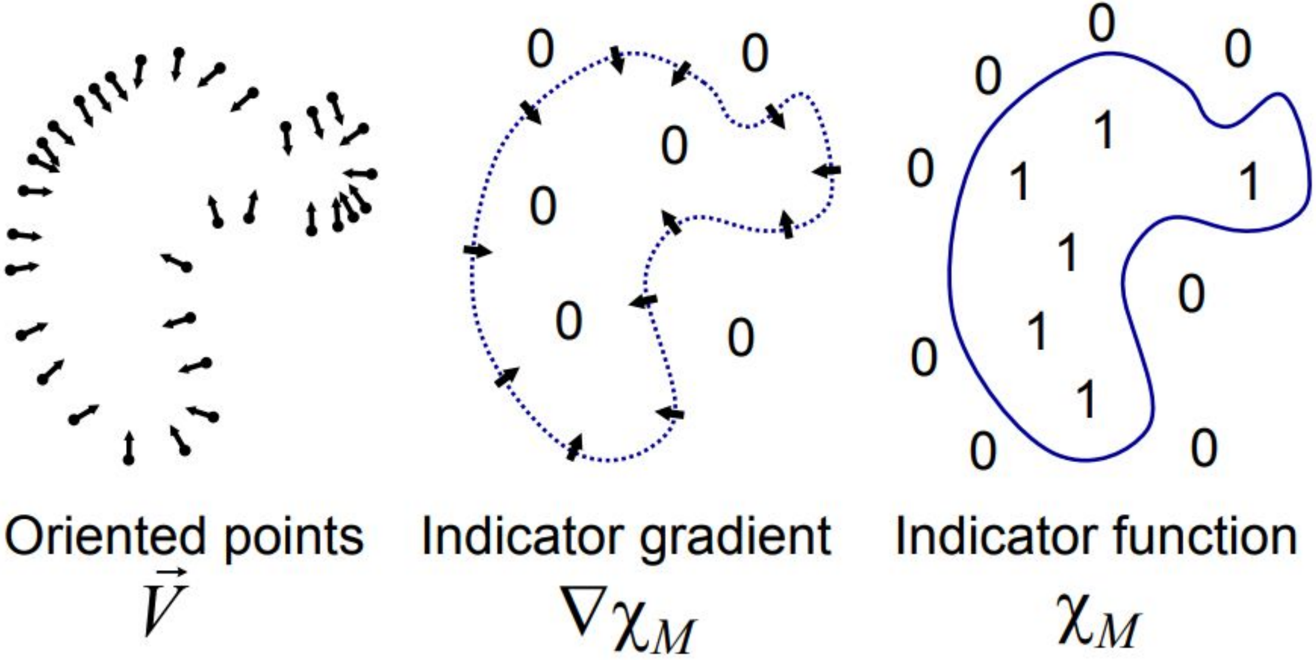
C - The Poisson reconstruction pipeline



Kazhdan et al. "Poisson surface reconstruction." SGP, 2006.

Overview

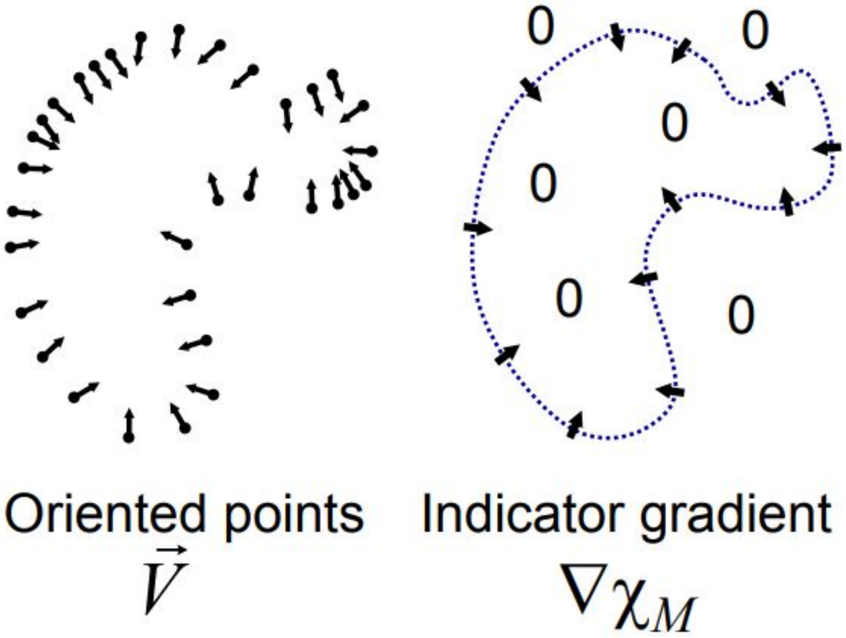
C - The Poisson reconstruction pipeline



Kazhdan et al. "Poisson surface reconstruction." SGP, 2006.

Overview

C - The Poisson reconstruction pipeline



splatted normals

$$\min_{\chi} \int \|\nabla\chi(\mathbf{x}) - \mathcal{N}(\mathbf{x})\|_2^2 \, dx$$

variational calculus

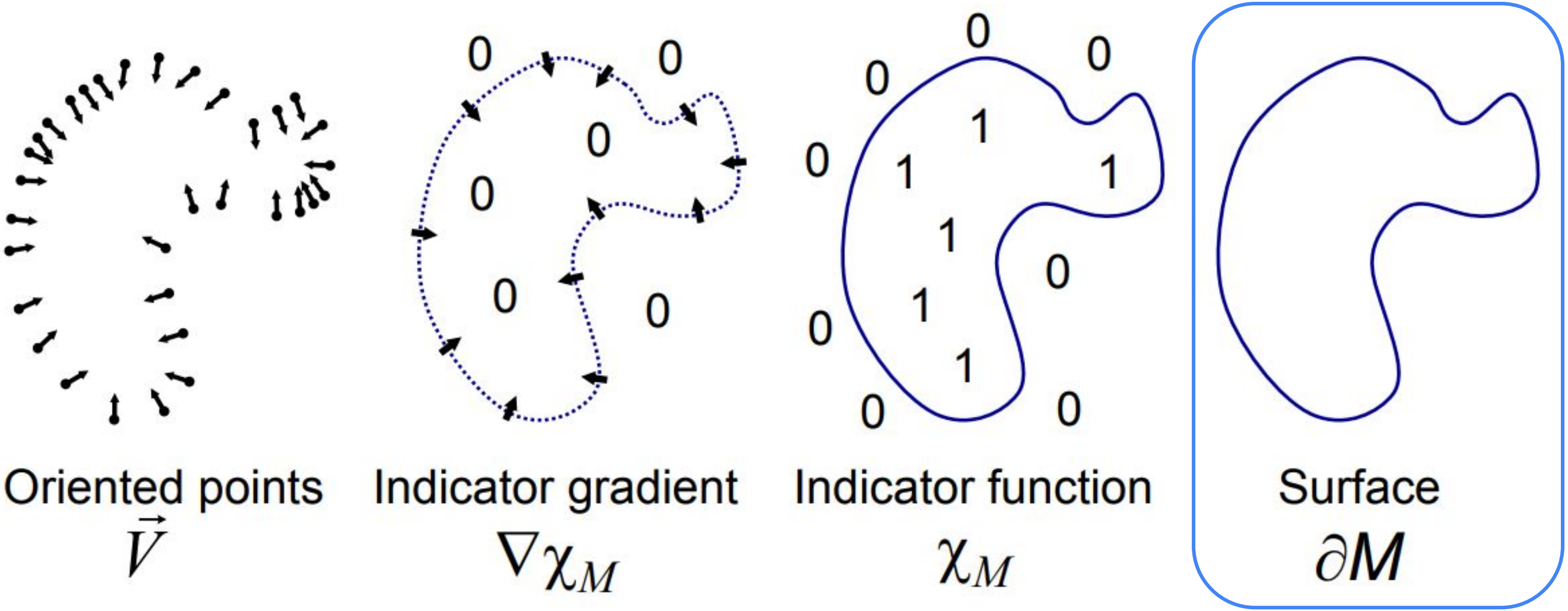
$$\Delta\chi = \nabla \cdot \mathcal{N}$$

sparse linear system

Kazhdan et al. "Poisson surface reconstruction." SGP, 2006.

Overview

C - The Poisson reconstruction pipeline



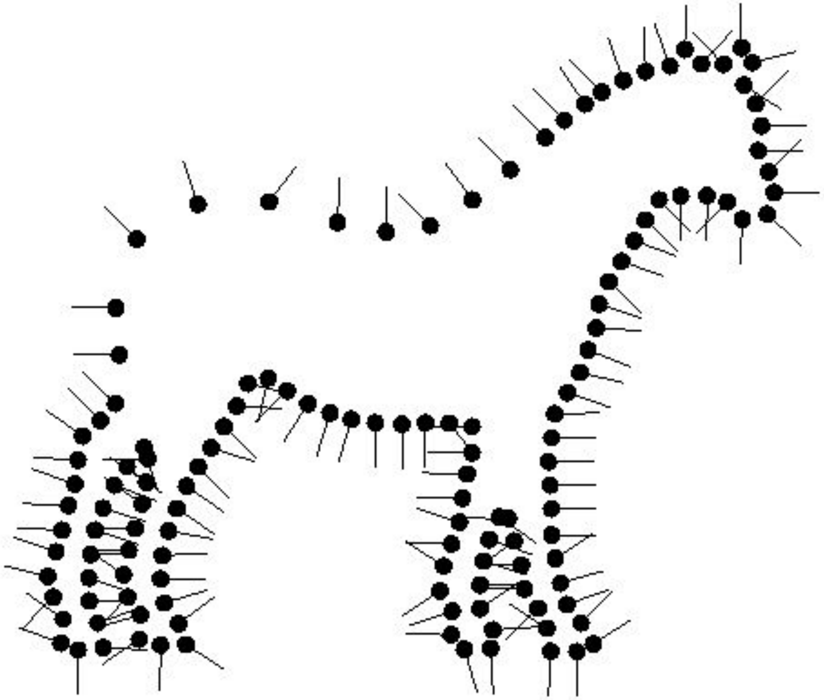
Kazhdan et al. "Poisson surface reconstruction." SGP, 2006.

In practice

C - The Poisson reconstruction pipeline

Given the Points:

- Set octree
- Compute vector field
- Compute indicator function
- Extract iso-surface

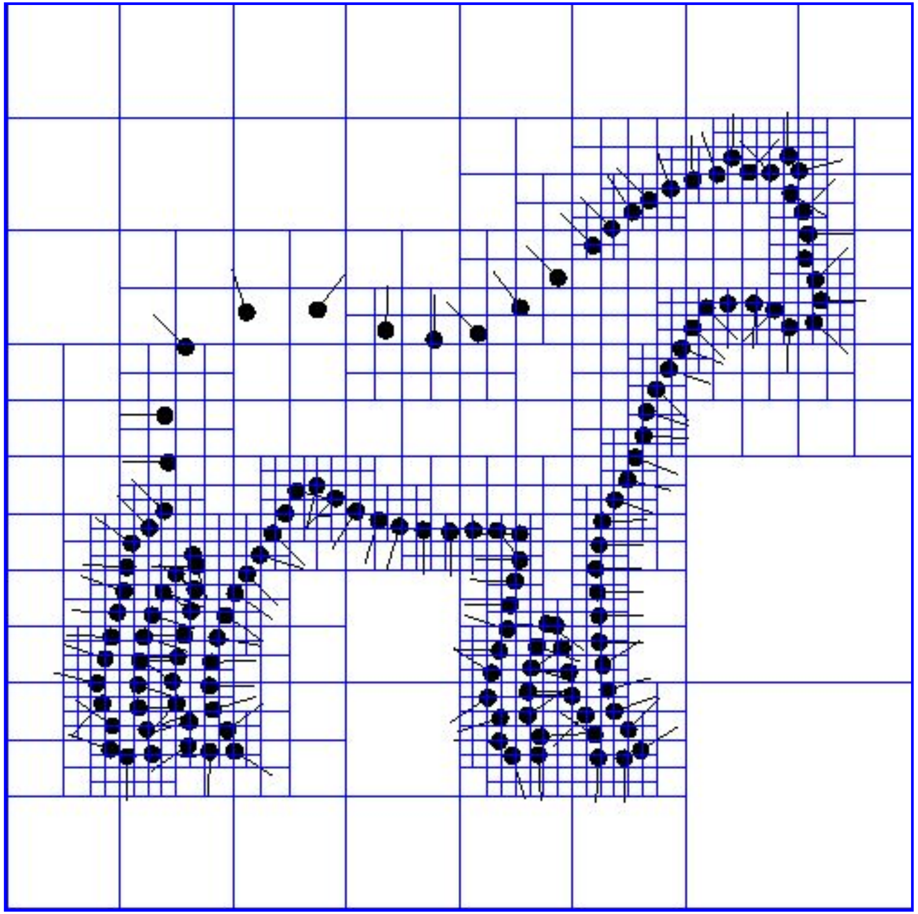


In practice

C - The Poisson reconstruction pipeline

Given the Points:

- **Set octree**
- Compute vector field
- Compute indicator function
- Extract iso-surface

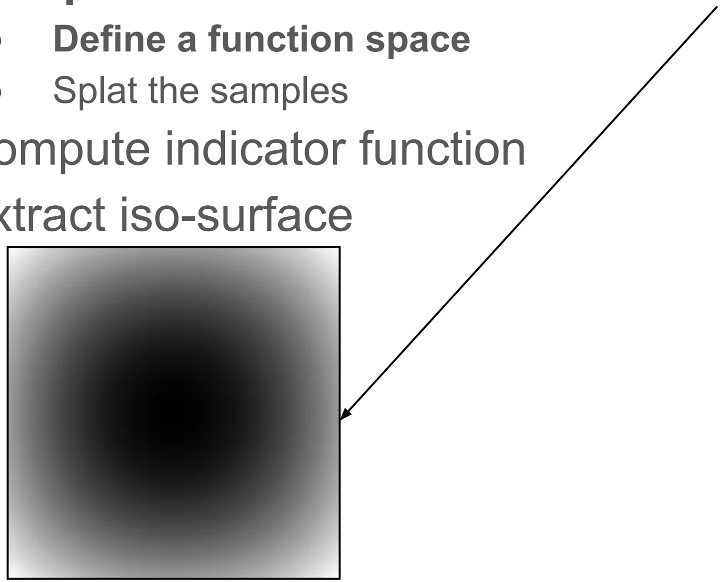


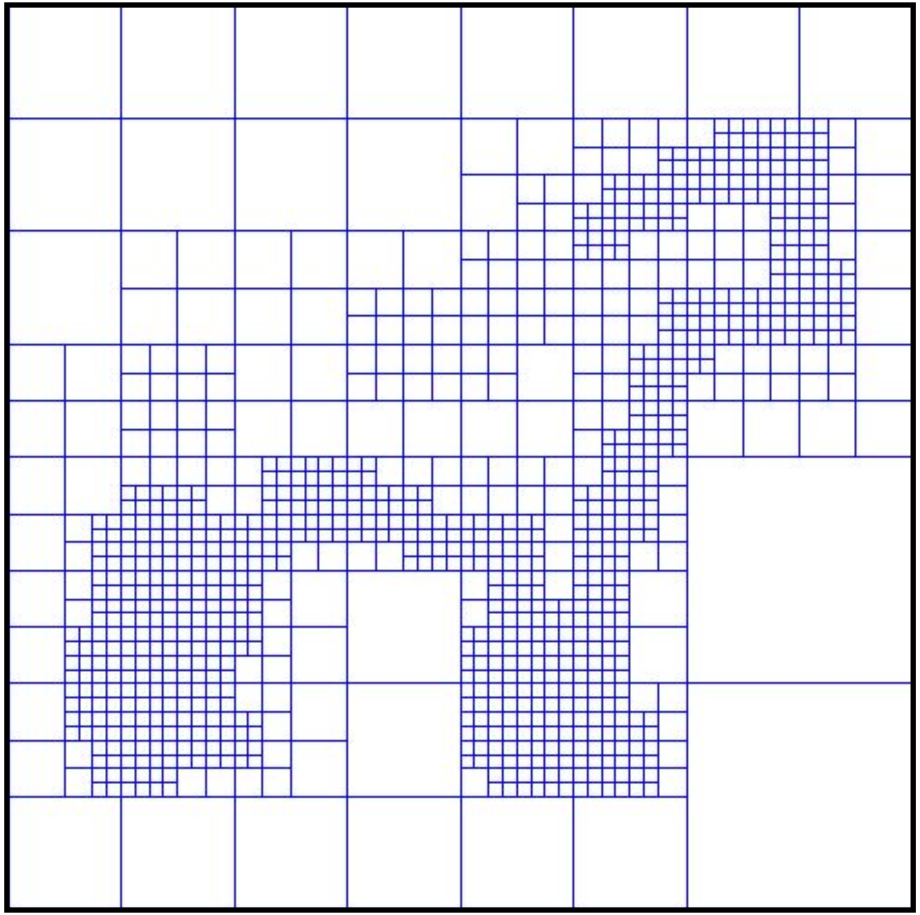
In practice

C - The Poisson reconstruction pipeline

Given the Points:

- Set octree
- **Compute vector field**
  - **Define a function space**
  - Splat the samples
- Compute indicator function
- Extract iso-surface



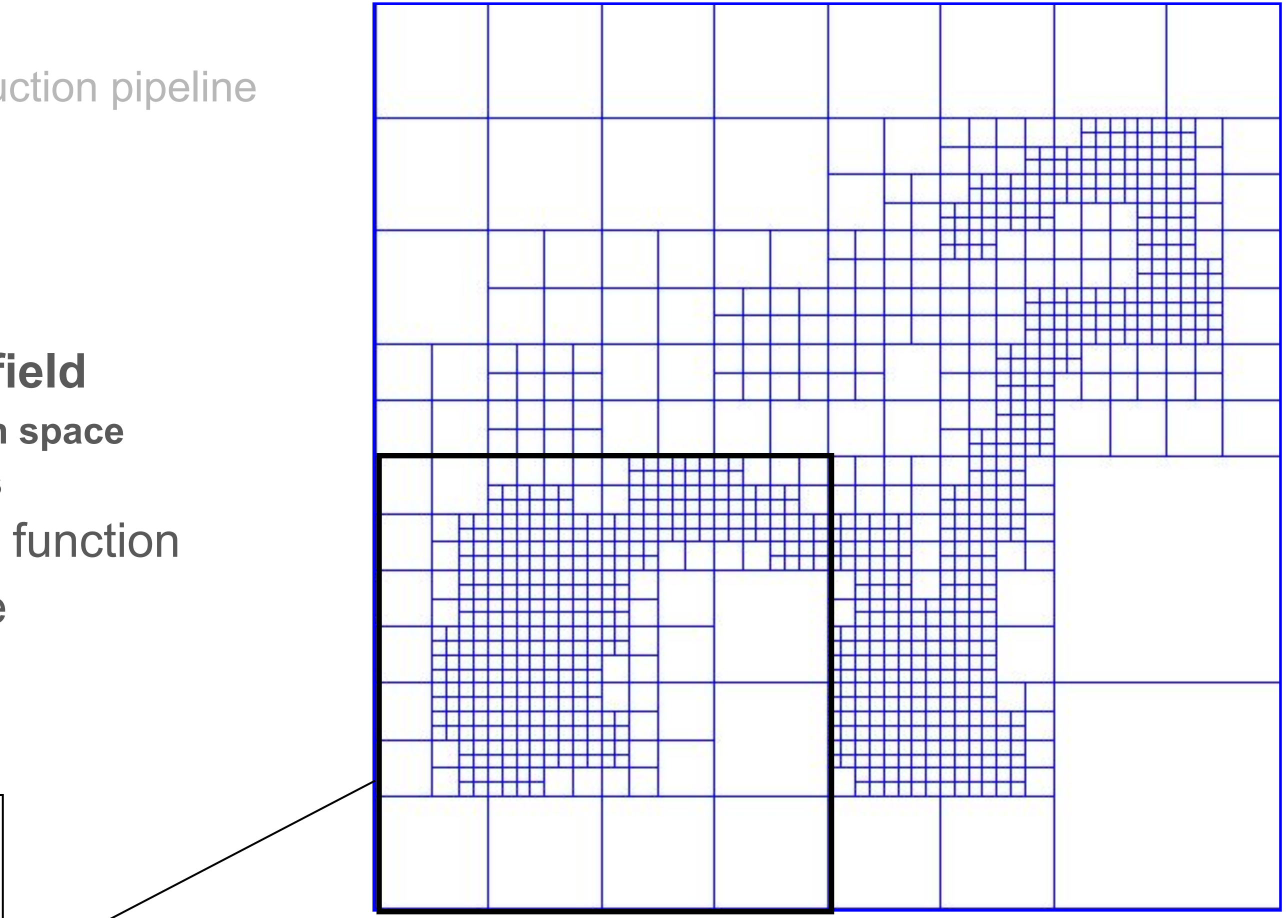


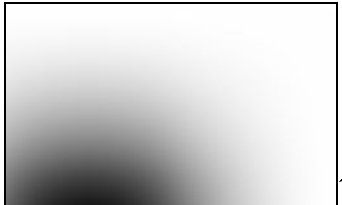
In practice

C - The Poisson reconstruction pipeline

Given the Points:

- Set octree
- **Compute vector field**
    - **Define a function space**
    - Splat the samples
- Compute indicator function
- Extract iso-surface



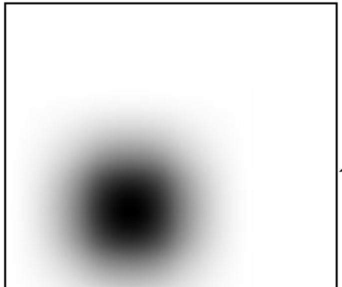


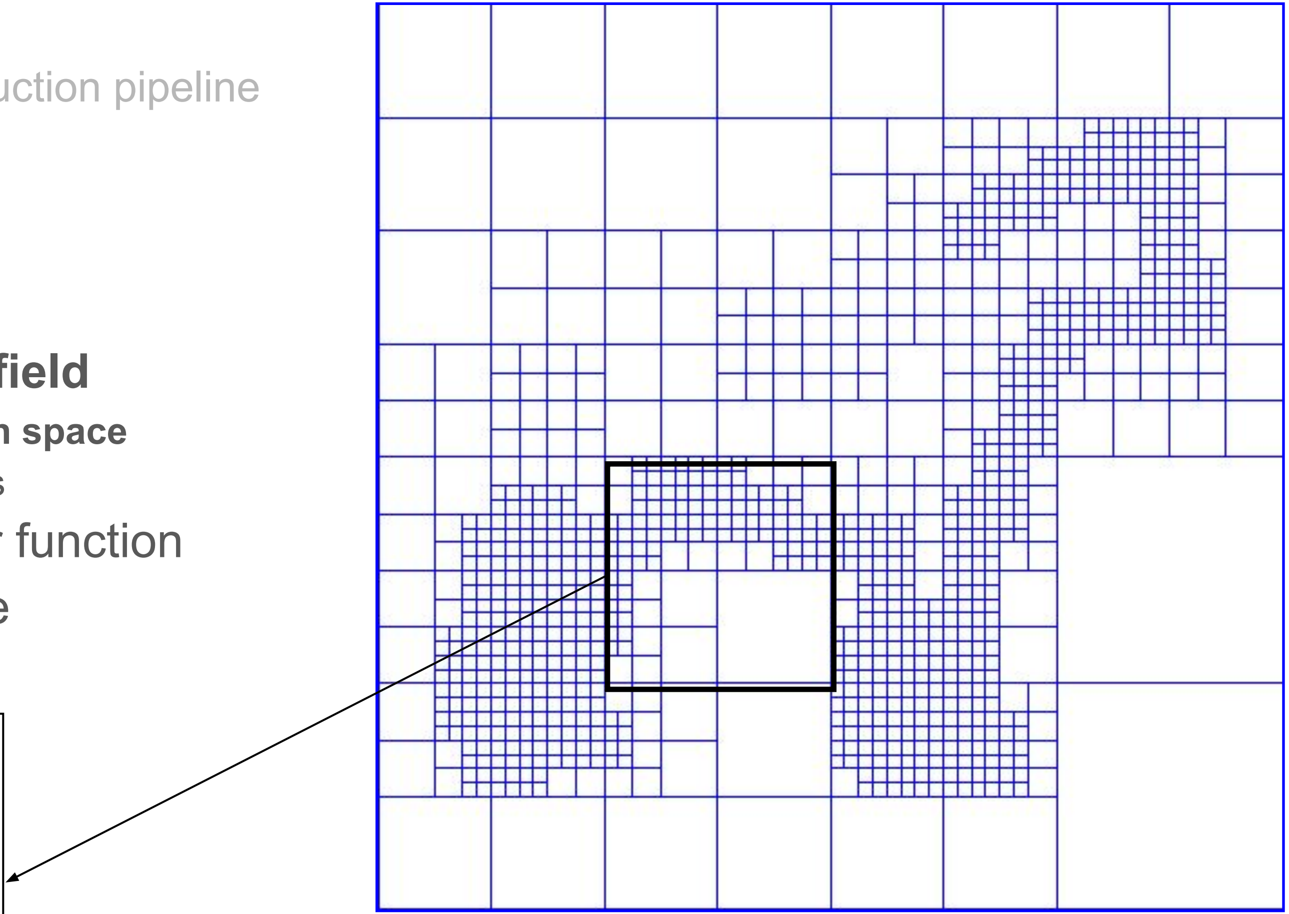
In practice

C - The Poisson reconstruction pipeline

Given the Points:

- Set octree
- **Compute vector field**
  - **Define a function space**
  - Splat the samples
- Compute indicator function
- Extract iso-surface



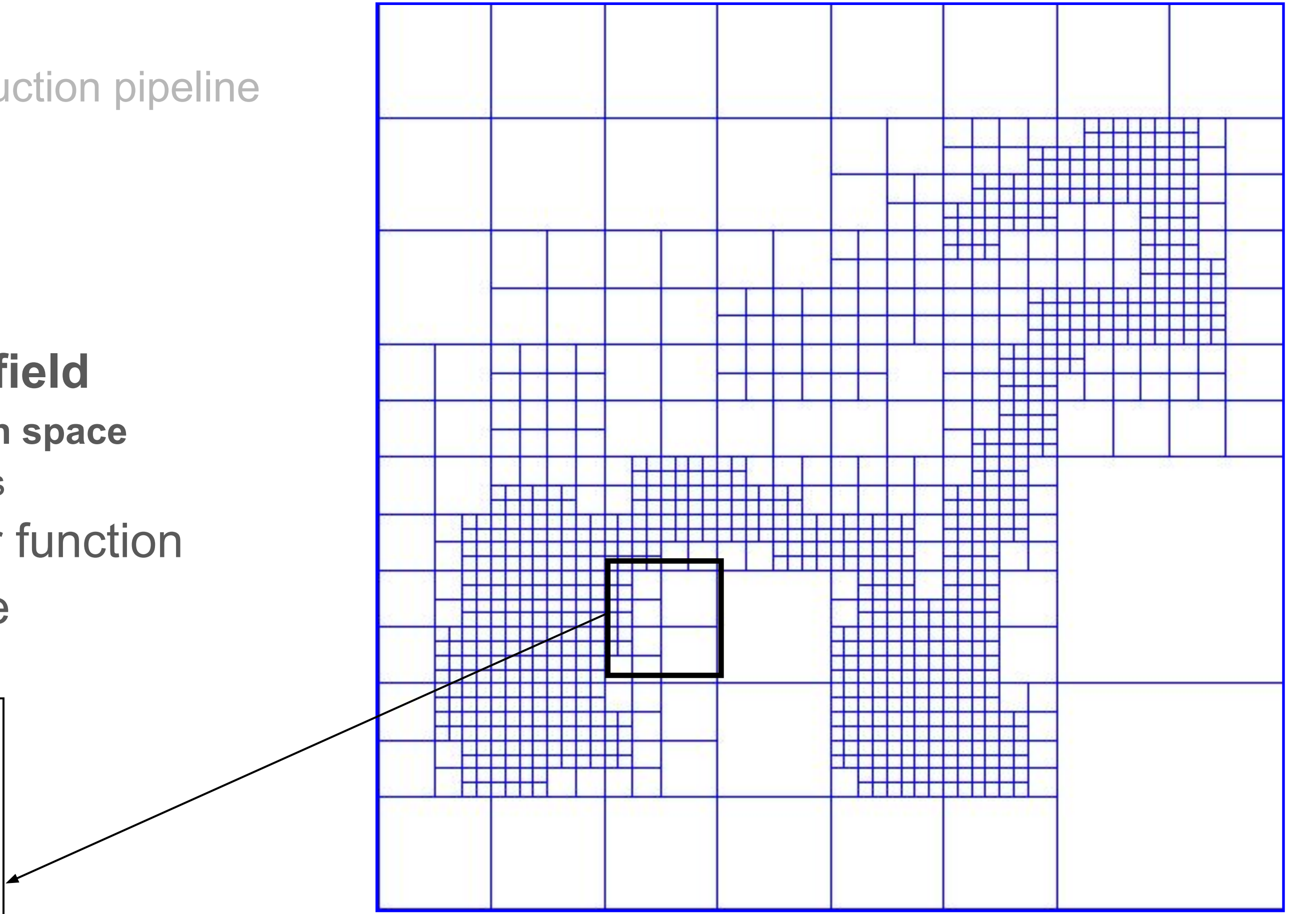


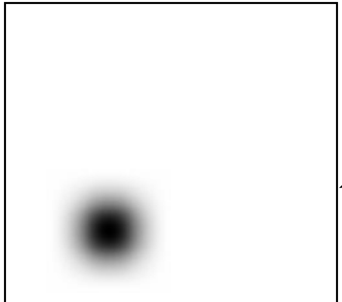
In practice

C - The Poisson reconstruction pipeline

Given the Points:

- Set octree
- **Compute vector field**
    - **Define a function space**
    - Splat the samples
- Compute indicator function
- Extract iso-surface



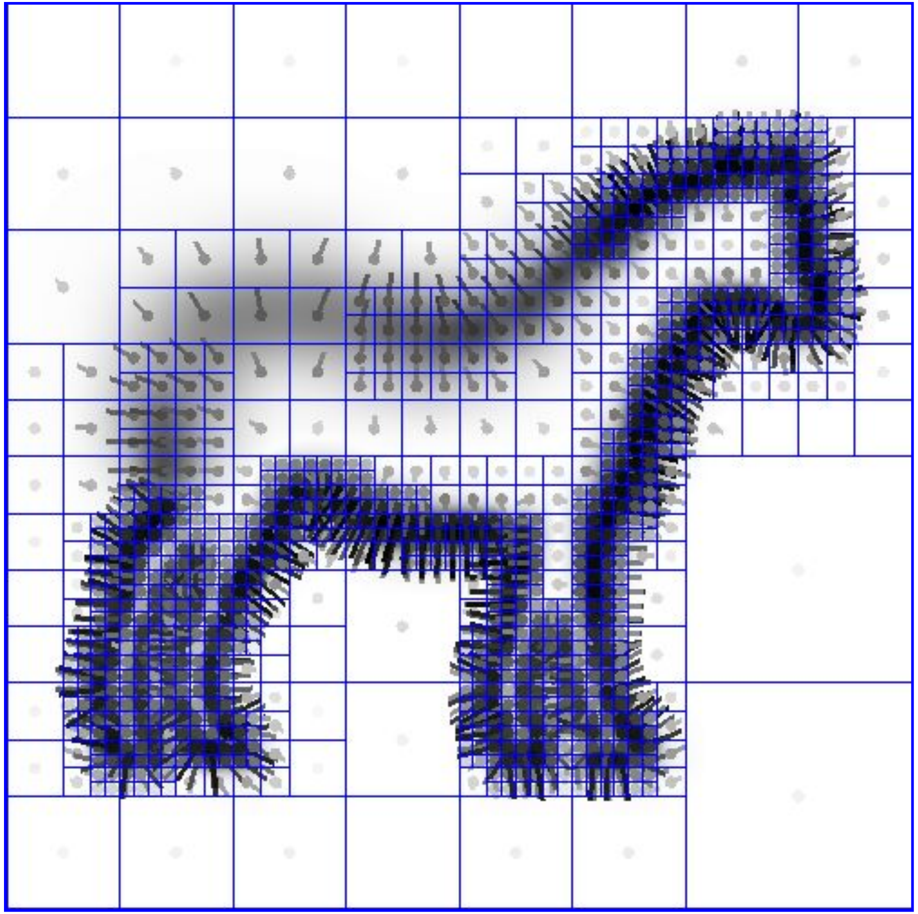


In practice

C - The Poisson reconstruction pipeline

Given the Points:

- Set octree
- **Compute vector field**
  - Define a function space
  - **Splat the samples**
- Compute indicator function
- Extract iso-surface



In practice

C - The Poisson reconstruction pipeline

Given the Points:

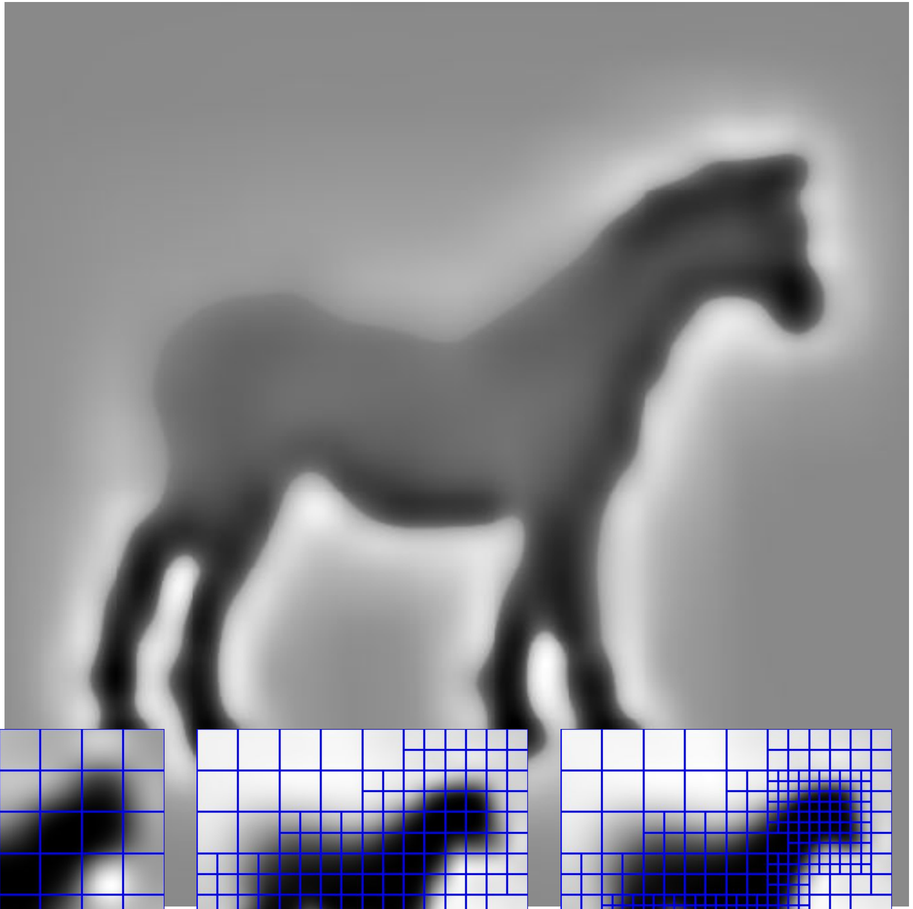
- Set octree
- Compute vector field
- **Compute indicator function**
  - **Compute divergence**
  - Solve Poisson equation
- Extract iso-surface

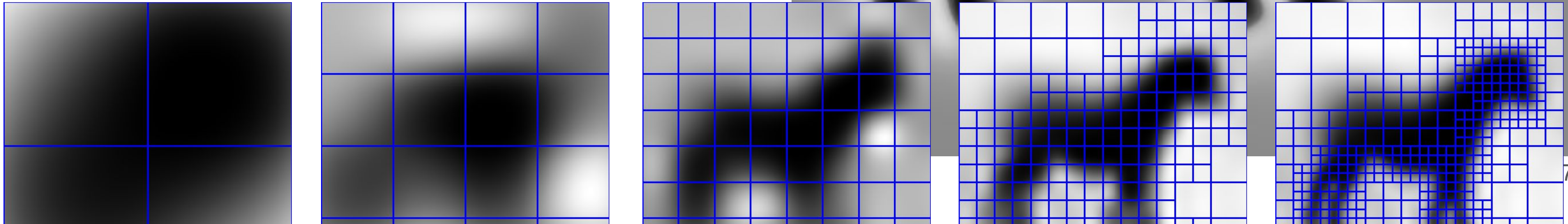
In practice

C - The Poisson reconstruction pipeline

Given the Points:

- Set octree
- Compute vector field
- **Compute indicator function**
    - Compute divergence
    - **Solve Poisson equation**
- Extract iso-surface





In practice

C - The Poisson reconstruction pipeline

Given the Points:

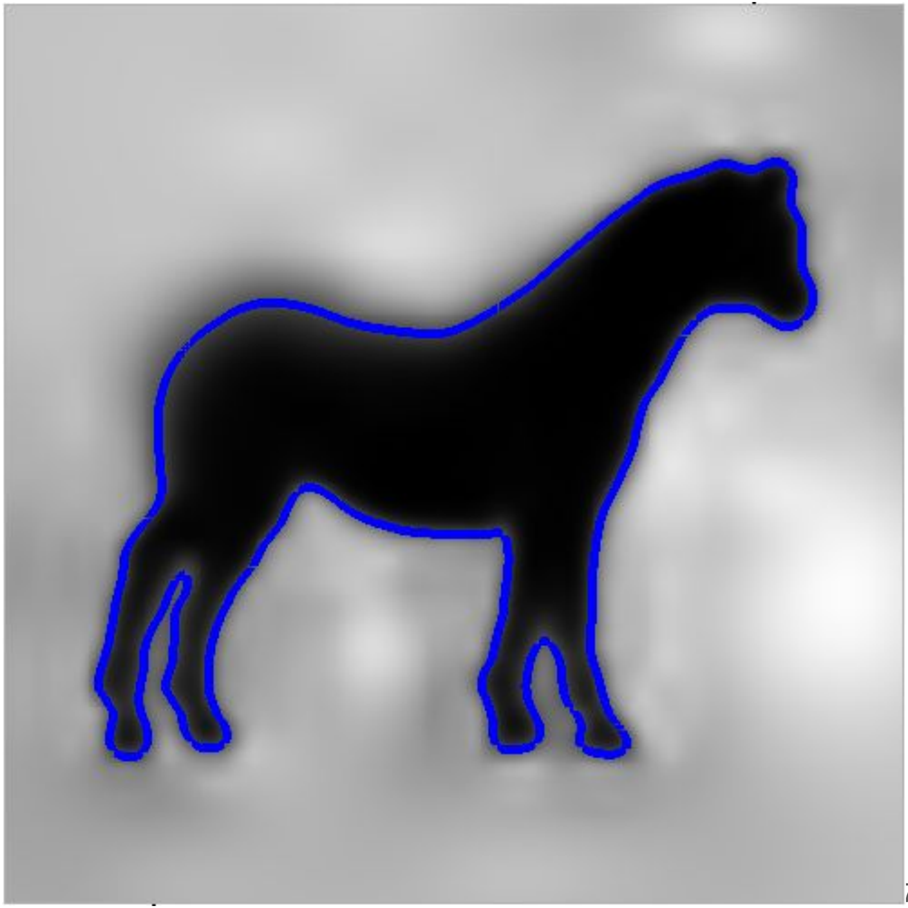
- Set octree
- Compute vector field
- **Compute indicator function**
  - Compute divergence
  - **Solve Poisson equation**
- Extract iso-surface

In practice

C - The Poisson reconstruction pipeline

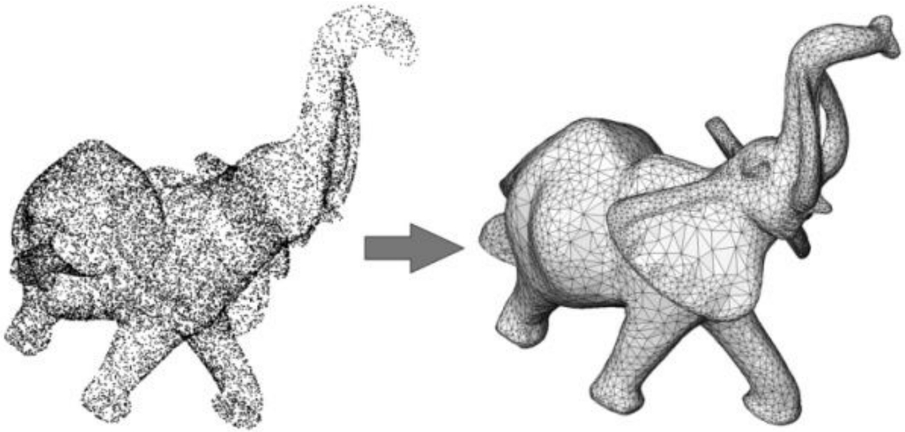
Given the Points:

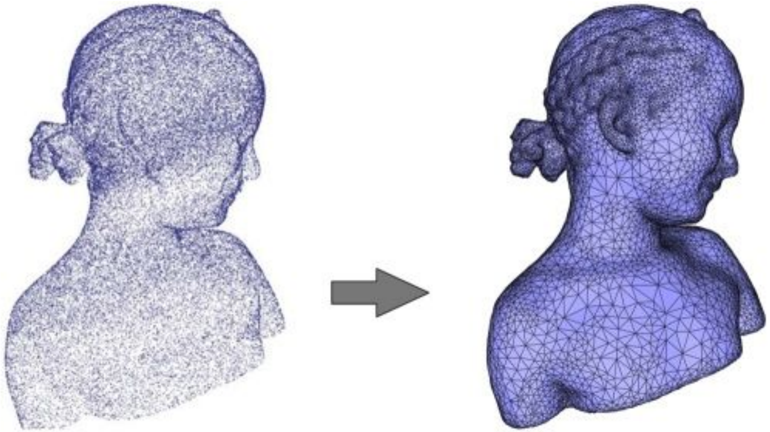
- Set octree
- Compute vector field
- Compute indicator function
- **Extract iso-surface**



Examples

C - The Poisson reconstruction pipeline





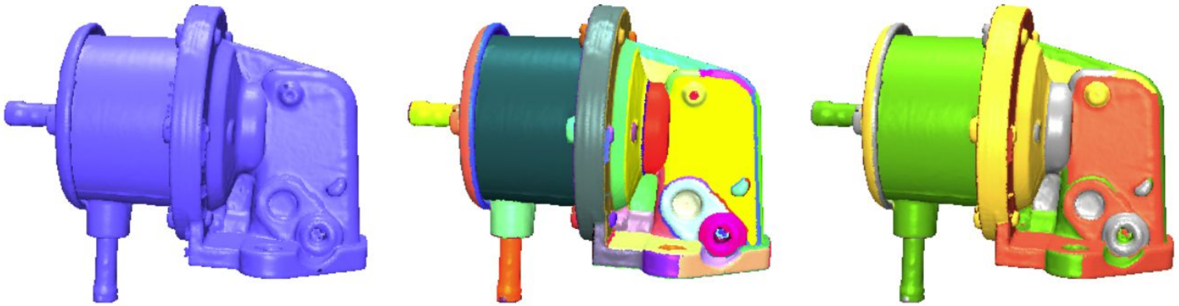
III - Surface reconstruction

D - RANSAC

Model-based approaches

Fit a model

- Abstraction
- Simplification
- CAD
- …



(a) Original (b) Random colors (c) Colored by type

(a) Original (b) Approximation

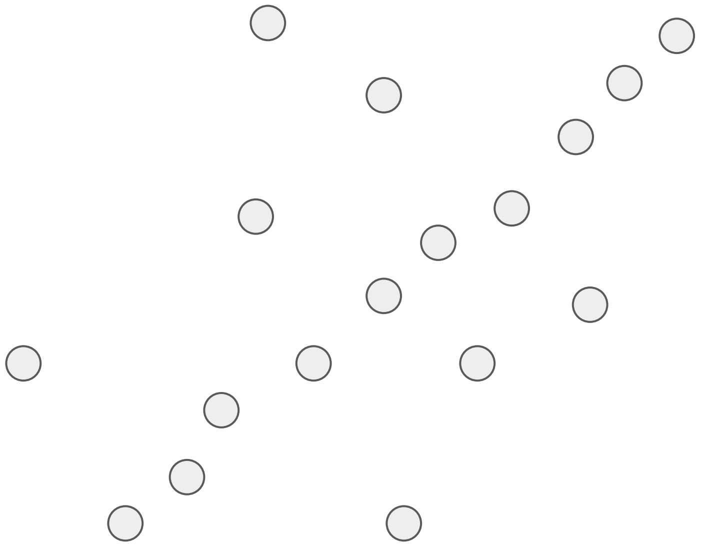
Efficient RANSAC for Point-Cloud Shape Detection, Schabel et al.

Random Sample Consensus

D - RANSAC

Random **Sa**mple **C**oncensus

- Defining a model
  - Line
  - Defined by to (different) points

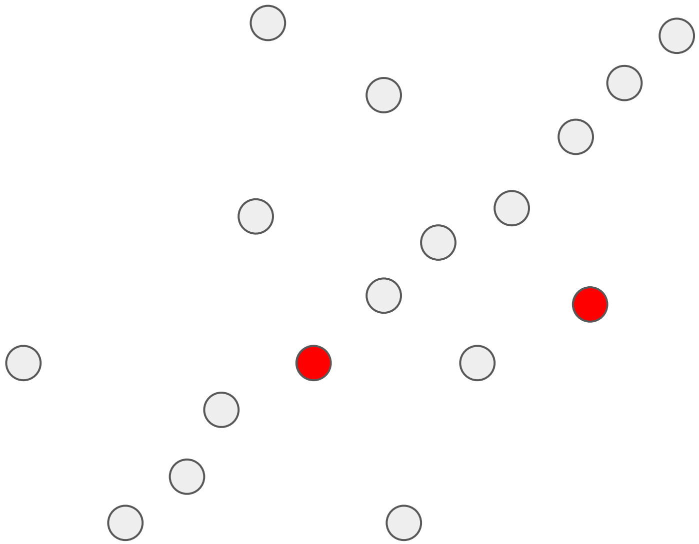


Random Sample Consensus

D - RANSAC

Random **Sa**mple **C**oncensus

- Defining a model
- Hypothesis generation
  - Pick subset of the data

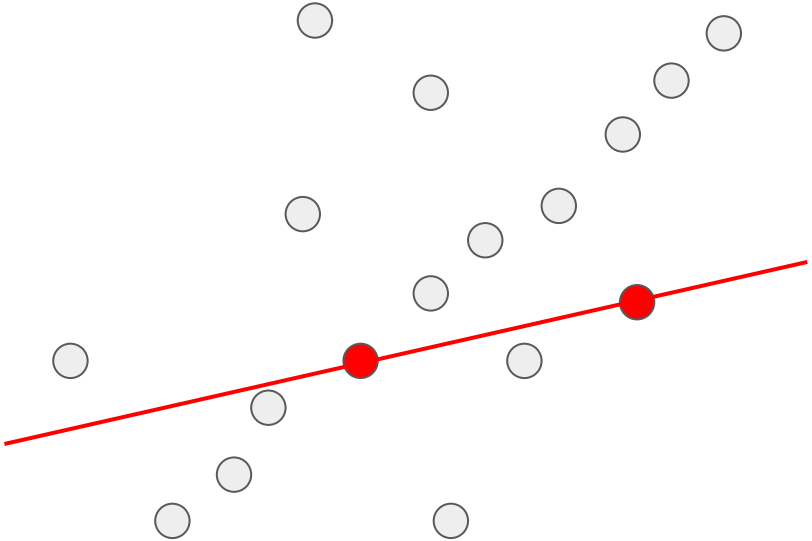


Random Sample Consensus

D - RANSAC

Random **Sa**mple **C**oncensus

- Defining a model
- Hypothesis generation
  - Pick subset of the data
  - Estimate the corresponding model

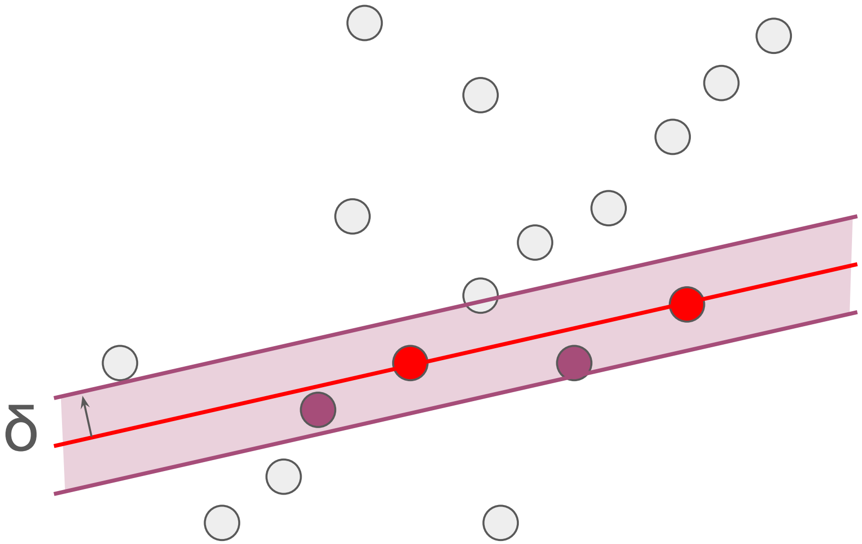


Random Sample Consensus

D - RANSAC

Random **Sa**mple **C**oncensus

- Defining a model
- Hypothesis generation
    - Pick subset of the data
    - Estimate the corresponding model
    - Compute inliers

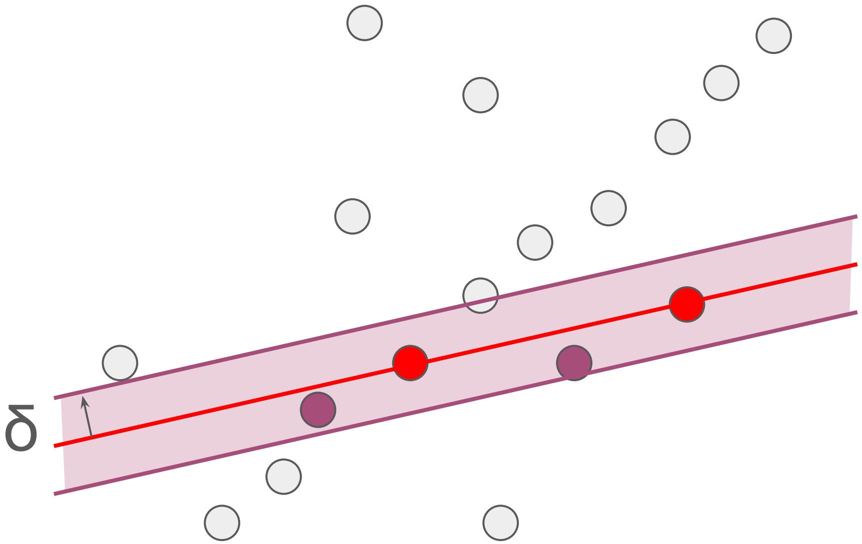


Random Sample Consensus

D - RANSAC

Random **Sa**mple **C**oncensus

- Defining a model
- Hypothesis generation
  - Pick subset of the data
  - Estimate the corresponding model
  - Compute inliers
- Compare to best candidate so far

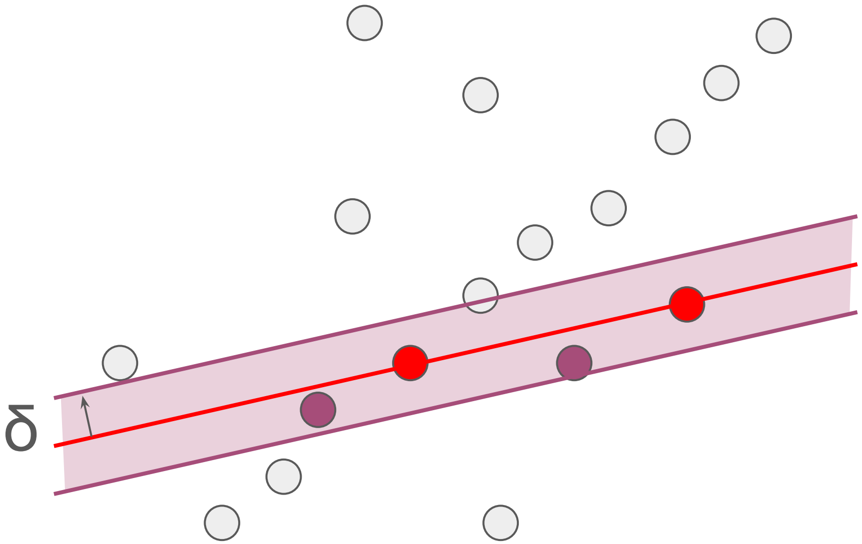


Random Sample Consensus

D - RANSAC

Random **Sa**mple **C**oncensus

- Defining a model
- Hypothesis generation
    - Pick subset of the data
    - Estimate the corresponding model
    - Compute inliers
- Compare to best candidate so far
- Iterate

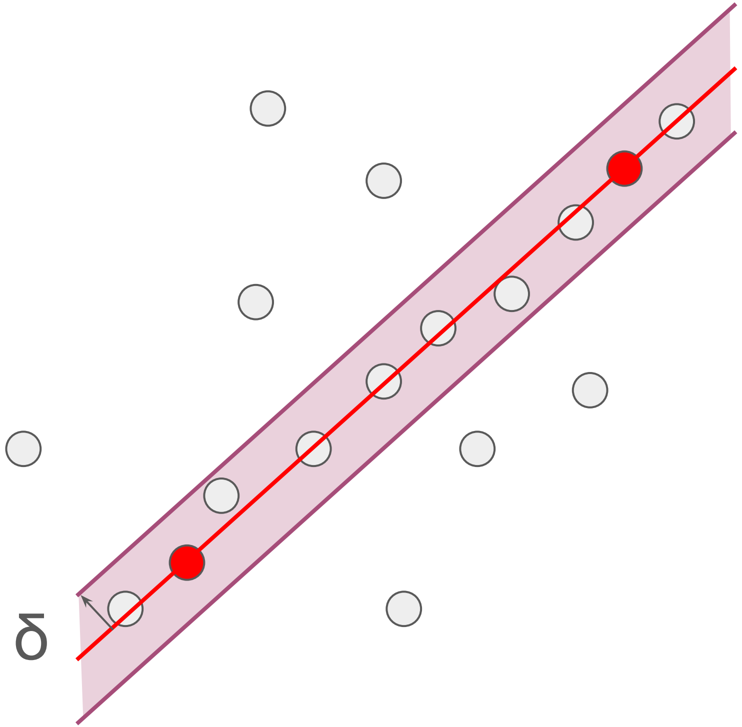


Random Sample Consensus

D - RANSAC

Random **Sa**mple **C**oncensus

- Defining a model
- Hypothesis generation
  - Pick subset of the data
  - Estimate the corresponding model
  - Compute inliers
- Compare to best candidate so far
- Iterate

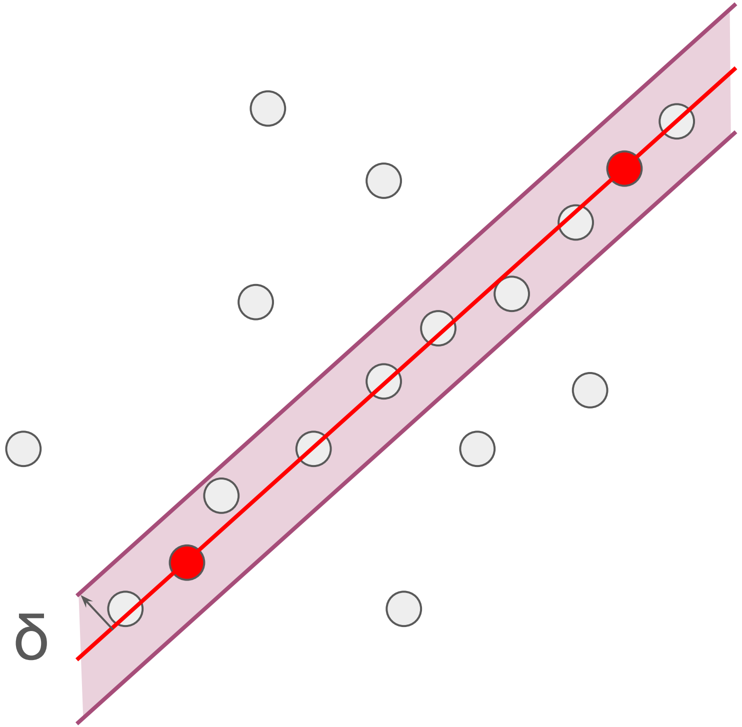


Random Sample Consensus

D - RANSAC

Parameters

- N: size of the point cloud
- k: number of points to generate an hypothesis
- δ: inlier tolerance
- T: number of models to pick in the loop



Random Sample Consensus

D - RANSAC

Estimating T

Probability of picking one inlier point :

$$\frac{n = \text{num. inliers}}{N = \text{p.c. size}}$$

Probability of picking one inlier hypothesis (k=3 inlier points):

$$P(n) = \binom{n}{k} / \binom{N}{k} \sim \left(\frac{n}{N}\right)^k$$

Probability that the hypothesis is an outlier:

$$1 - P(n)$$

Random Sample Consensus

valeo.ai

D - RANSAC

Estimating T

Probability that after none of s hypothesis is an inlier

$$(1 - \left(\frac{n}{N}\right)^k)^s$$

Probability that at least one hypothesis in an inlier

$$P(n, s) = 1 - (1 - \left(\frac{n}{N}\right)^k)^s$$

We want T such that the probability of picking an inlier to be more than p_t

$$P(n, T) = 1 - (1 - \left(\frac{n}{N}\right)^k)^T > p_t$$

Random Sample Consensus

D - RANSAC

Estimating T

We want T such that the probability of picking an inlier to be more than p_t

$$P(n,T) = 1 - (1 - \left(\frac{n}{N}\right)^k)^T > p_t$$

Then:

$$T = \frac{\ln(1-p_t)}{\ln(1-\left(\frac{n}{N}\right)^k)}$$

n and p_t are parameters to be set.

Conclusion and practical session

Surface reconstruction

Many methods have been developed with various characteristics:

Simple, Smooth, Model-based, Optimal (for given criteria), Robust to noise…

Non-learning methods are usable off-the-shelf.

Learning-based methods… see course 7

Practical session

Conclusion and practical session

Implement a RANSAC plane extractor

https://github.com/aboulch/MSIA_points/blob/main/03_surfaces/MSIA_Points_3_surfaces.ipynb

https://www.college-de-france.fr/media/jean-daniel-boissonnat/UPL215966848019
1960308_alliez_reconstruction.pdf

https://www.cs.jhu.edu/~misha/MyPapers/SGP06.ppt

https://courses.grainger.illinois.edu/cs598dwh/fa2021/lectures/Lecture%2011%20-
%203D%20Registration%20and%20Shape%20Fitting%20-%203DVision.pdf