

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 or images
- End-to-end driving
 - From sensor to decision

valeo.ai

A Paris-based research team

working on Valeo's AI applications, esp. perception
for ADAS/AD

Open research

- Publications in Conferences / Journals
- Code publication under open source license

<https://www.valeo.com/en/valeo-ai/>

<https://valeoai.github.io/blog/>

<https://github.com/valeoai/>

<https://twitter.com/valeoai>

valeo.ai

Collaborations with Valeo's AI network
(150+) and world-class academics.



cnam

MINES

TELECOM

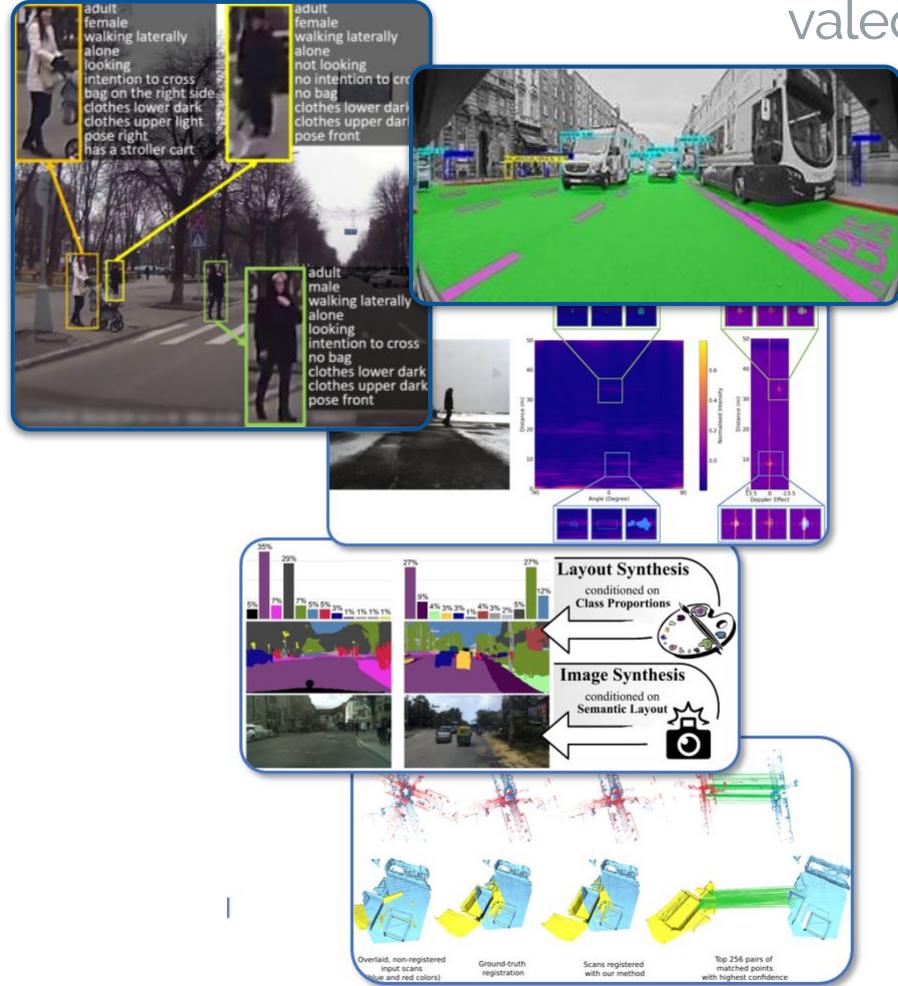
PR[AI]RIE
Paris Artificial Intelligence Research Institute

EPFL



mpii
max planck institut
informatik

Université
Bretagne Sud
UBS:





100 publications

CVPR (25), ICCV+ECCV (22)

NeurIPS+ICLR+ICML (7)

11 Journals

ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation

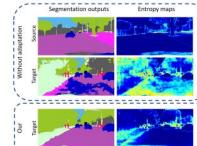
Tuan-Hung Vu¹ Himalaya Jain¹ Maxime Bucher¹ Matthieu Cord^{1,2} Patrick Pérez¹
¹valeo.ai, Paris, France
²Sorbonne University, Paris, France

Abstract

Semantic segmentation is a key problem for many computer vision services. While approaches based on domain adaptation have recently broken records on different benchmarks, generalizing well to diverse testing environments remains a major challenge. In numerous real world applications, there is indeed a large gap between data available at training time and the data encountered in severe performance loss at run-time. In this work, we address the task of unsupervised domain adaptation in semantic segmentation by focusing on the entropy of the pixel-wise predictions. To this end, we propose two complementary methods using (i) an entropy loss and (ii) an adversarial loss respectively. We demonstrate state-of-the-art performance on semantic segmentation on the challenging “synthetic-2-real” set-up, and show that the approach can also be used for detection.

1. Introduction

Semantic segmentation is the task of assigning class labels to all pixels in an image. In practice, segmentation models often suffer as the backbone of complex computer vision tasks like autonomous vehicles, which demand high accuracy in a large variety of urban environments. For example, under adverse weather, the system must be able to recognize, for instance, lane stipples or road markings, their appearances being largely different from ones in the training set. A more extreme and important example is so-called “synthetic-2-real” set up [31, 30] – training on synthetic data and testing on real-world data that are real scenes. Fully-supervised approaches [33, 47, 2] have not yet guaranteed a good generalization to arbitrary test cases. On the other hand, a model trained on one domain, named as *source*, usually suffers a dramatic drop in performance when applied on another domain, named as *target*.



¹Code available at <https://gitlab.com/valeoai/ADVENT>.

Semi-supervised learning addresses a closely related

Talks & services

Talks, tutorials, panels

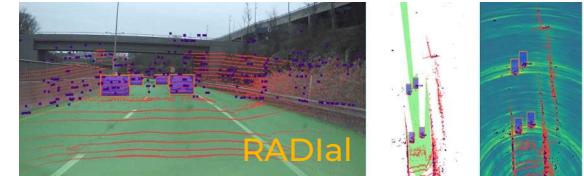
Reviewing and chairing

Juries and committees

Code & Datasets

38 research codes

3 datasets



Numpy

Optimization

Points

Segmentation

Machine learning

Neural networks

Surfaces

Geometry

Pytorch

Random forests

Scikit-learn

Differentiable
optimization

Classification

Images

Overview

Machine learning courses

- Surface reconstruction
- ML1
 - Descriptors and machine learning
 - Image based processing
- ML2
 - Geometric deep learning
- ML3
 - Convolutional and Transformer based architectures
- ML4
 - Tasks and corresponding architectures
- Opening session

Evaluation

Practical sessions

- ML2 and ML4 are evaluated

Final evaluation

- Scientific paper reading and analysis
- 10 minutes presentation with slides
- 10 questions and discussion
- Scheduled after ML4

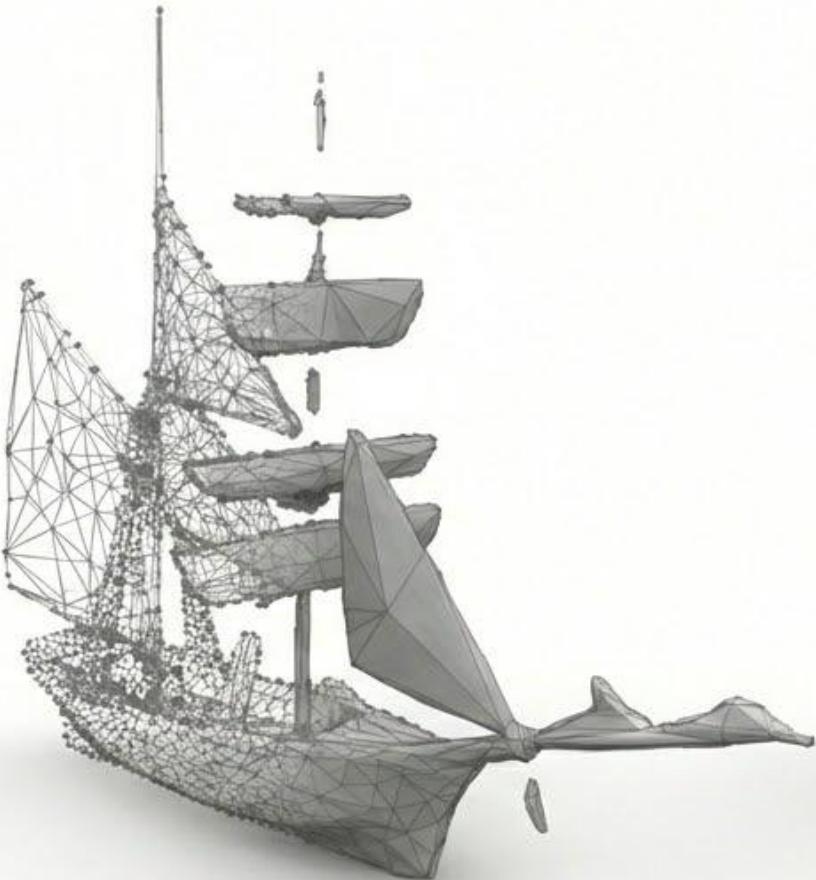
Ressources

Web page: <https://boulch.eu/teaching/>

Github: https://github.com/aboulch/MSIA_points

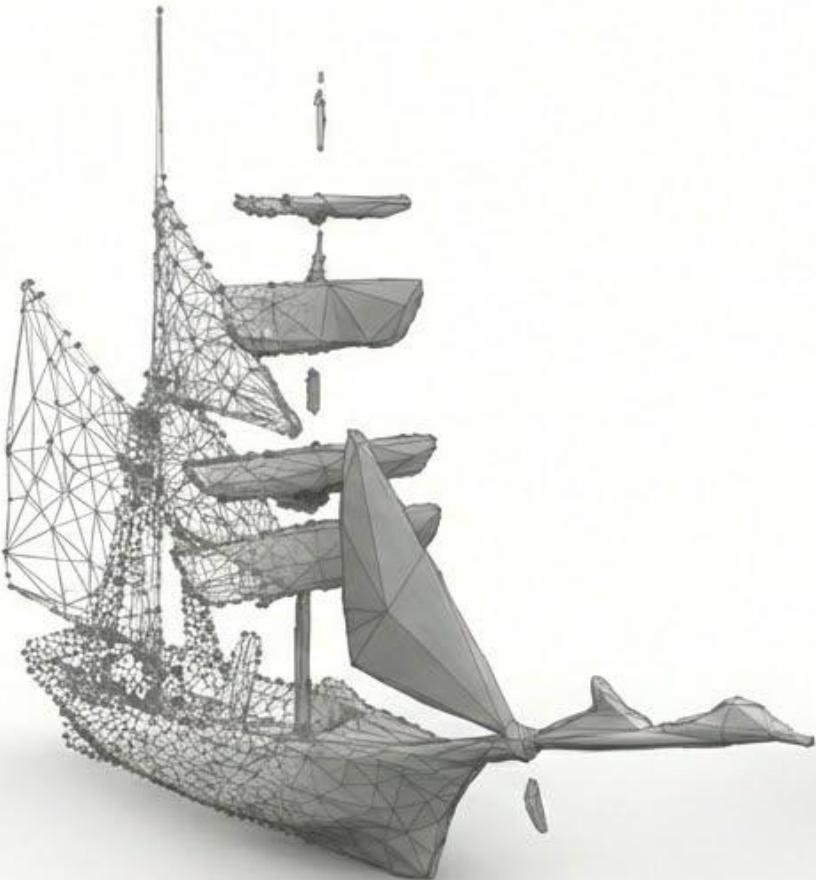
Email: alexandre.boulch@valeo.com





Overview

- I. From point clouds to surfaces
- II. Local features
 - A. Normal estimation
 - B. Normal orientation
- III. Surface reconstruction
 - A. Ball pivoting
 - B. Delaunay reconstruction
 - C. Poisson reconstruction
 - D. RANSAC



Overview

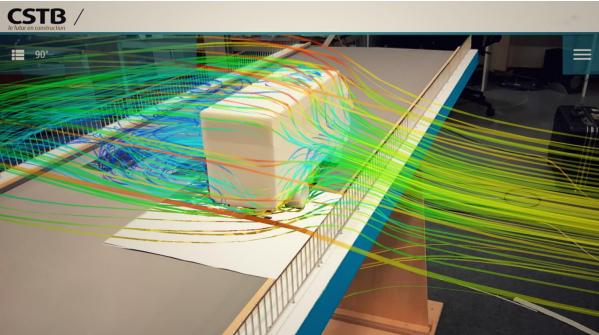
- I. From point clouds to surfaces**
- II. Local features**
 - A. Normal estimation
 - B. Normal orientation
- III. Surface reconstruction**
 - A. Ball pivoting
 - B. Delaunay reconstruction
 - C. Poisson reconstruction
 - D. RANSAC

3D rendering

Archeology



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



Simulations

CSTB

Games

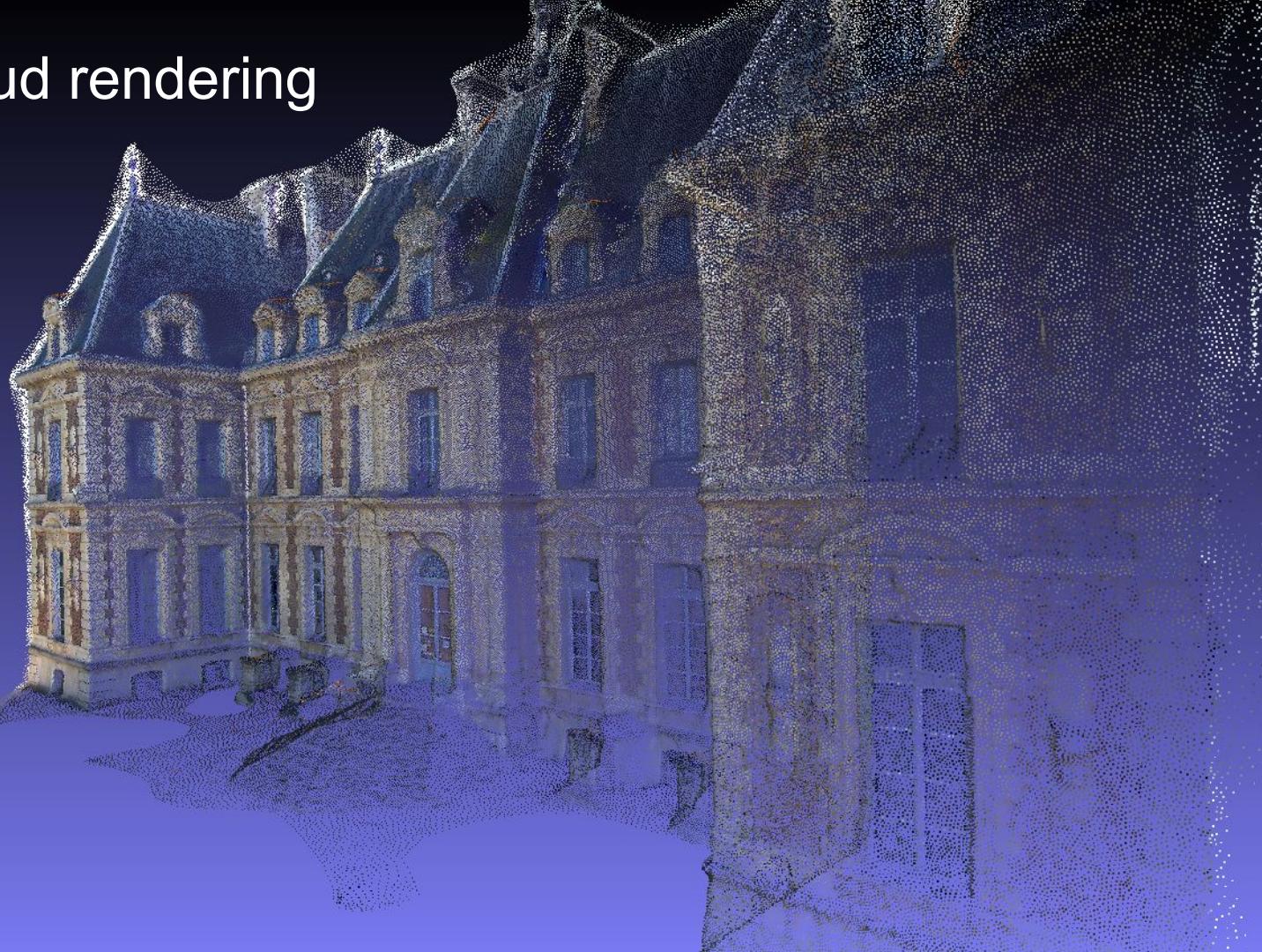


Amiens cathedral



Cyberpunk 2077 - Technology preview

Point cloud rendering



Point clouds rendering

I - From point cloud to surfaces

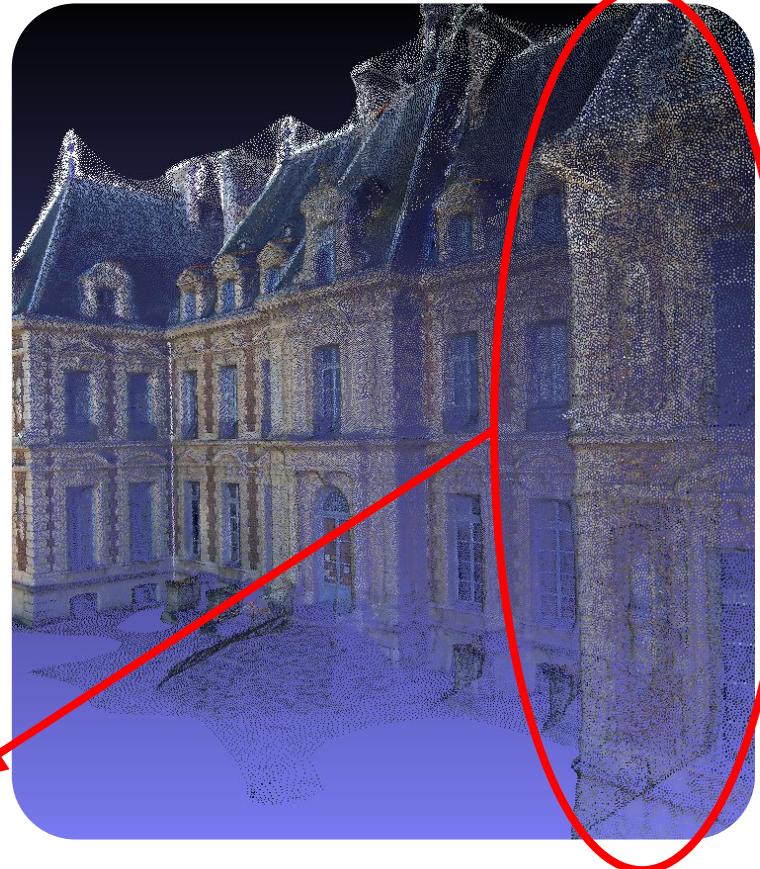
Point clouds are simple:

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

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

Higher point size
⇒ loss of details



Point clouds

I - From point cloud to surfaces

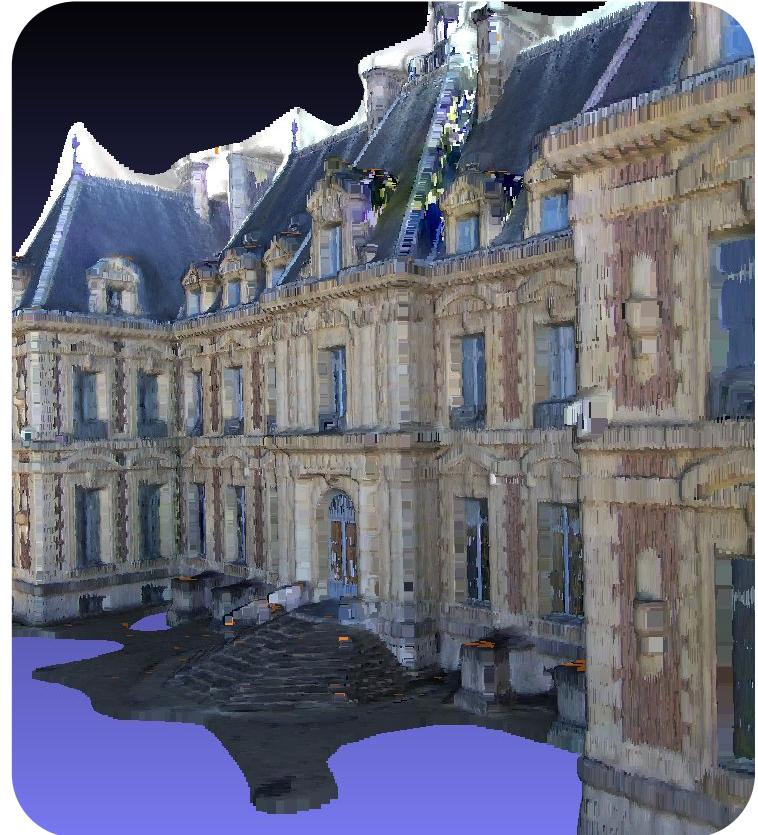
Point clouds are simple:

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

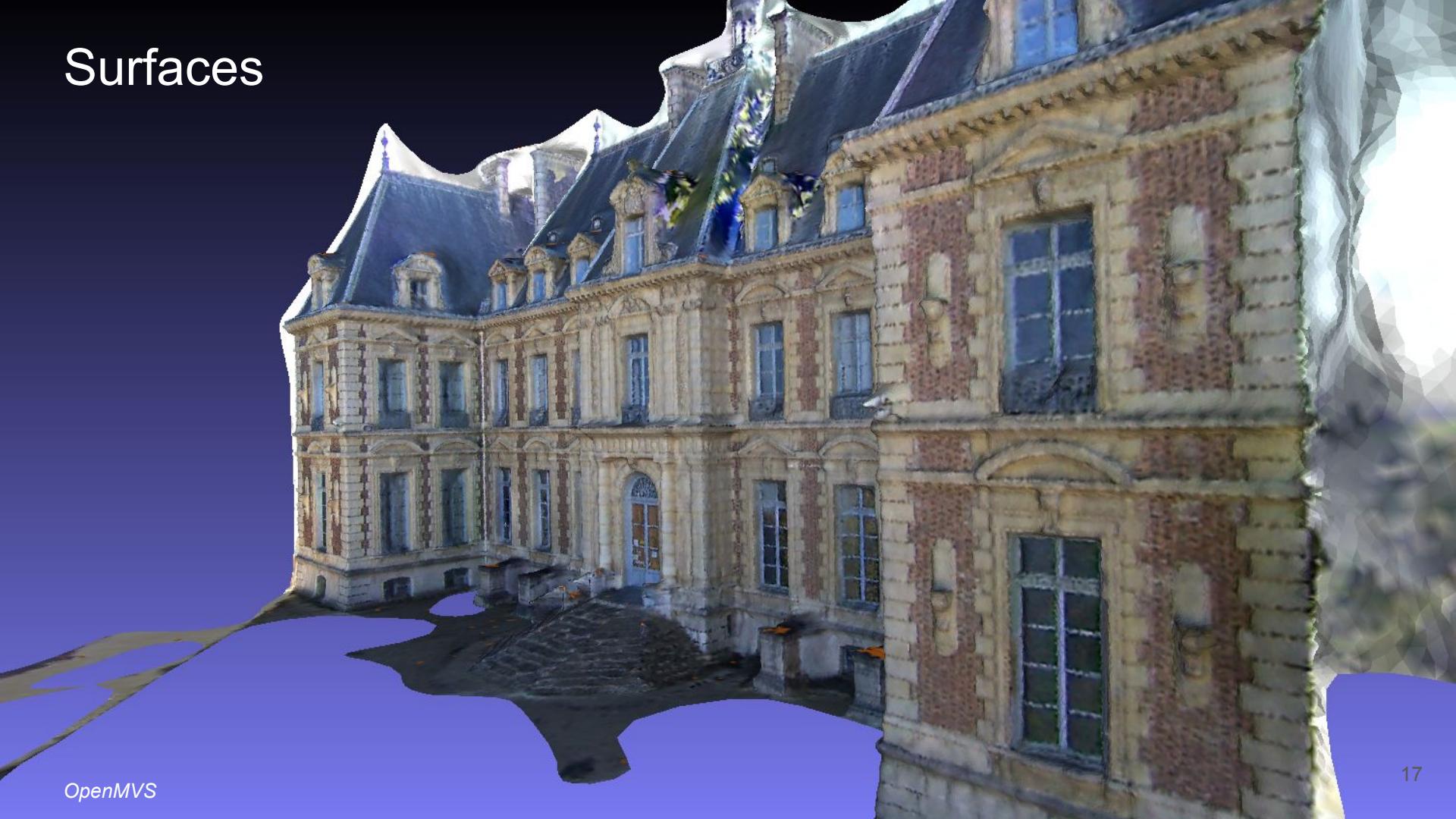
However:

- Points are independent
- Not easy to render

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



Surfaces





What are point clouds?

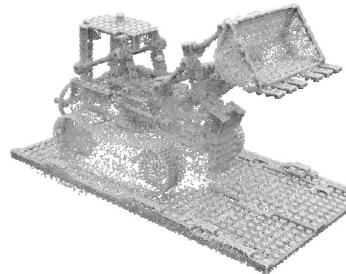
I - From point cloud to surfaces

A point cloud:

- A set of 3D coordinates

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

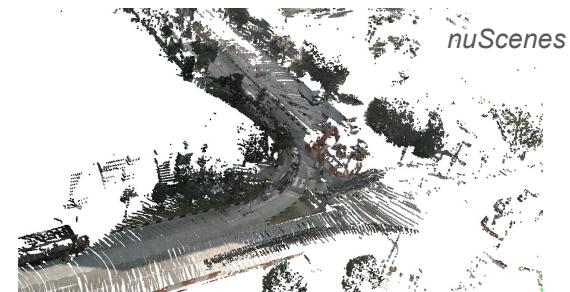
- No obvious order (at first sight)
- Sparse sample of surface
- Noisy / outliers
- Variation of size: several orders of magnitude



NeRF



Open3d



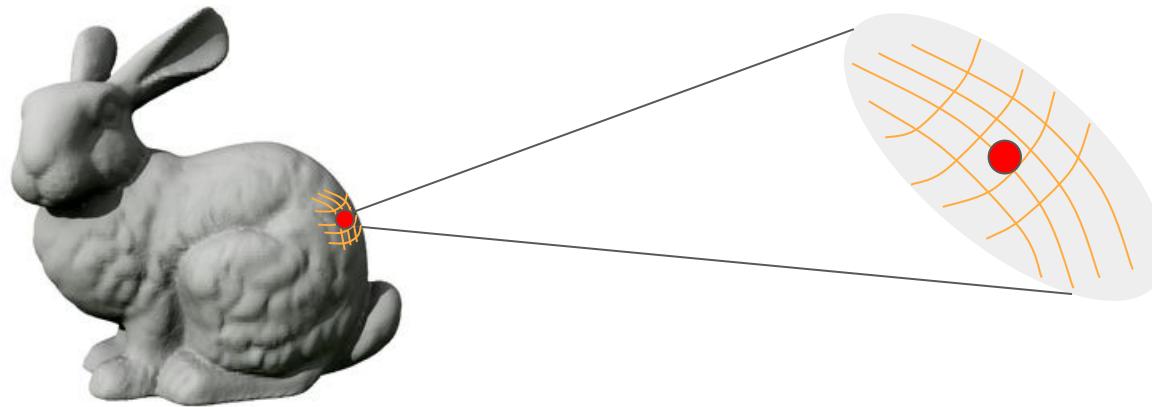
nuScenes

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)

Why surfaces?

I - From point cloud to surfaces

Surfaces for:

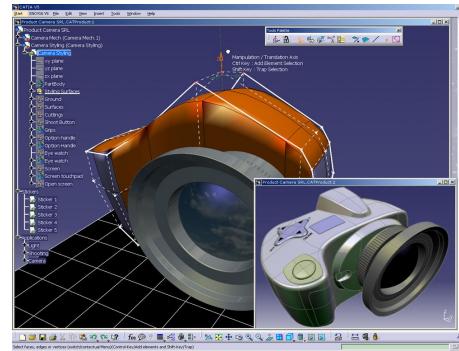
- Simulation
- Animation
- Design
- ...



A Multiscale Approach to Mesh-based Surface Tension Flows



Blender



Catia

How to represent a surface

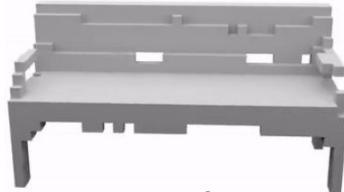
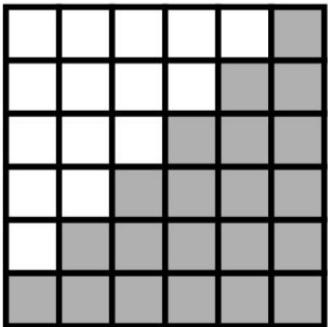
I - From point cloud to surfaces

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

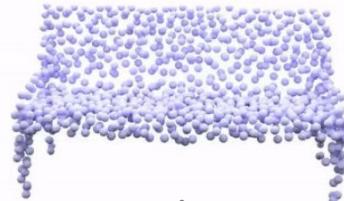
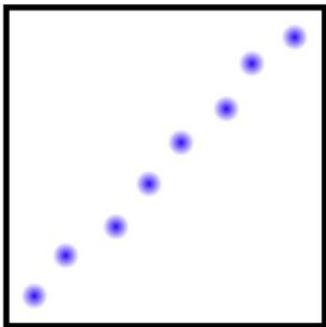
→ no unified / perfect representation

How to represent a surface?

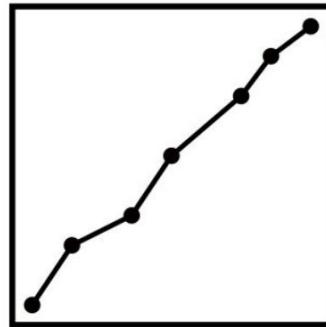
I - From point cloud to surfaces



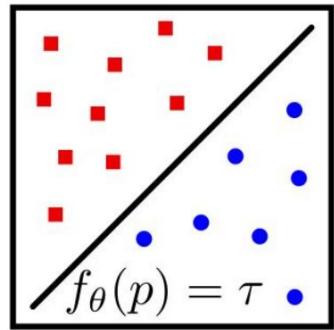
Voxels



Points



Mesh



Implicit

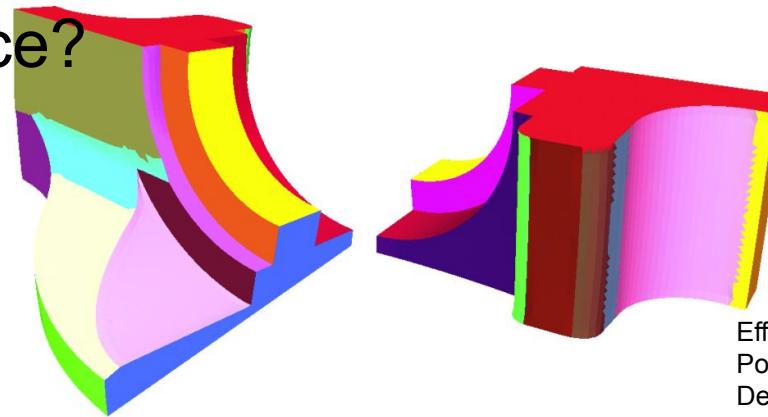
Occupancy Networks: Learning 3D Reconstruction in Function Space

Mescheder, Lars and Oechsle, Michael and Niemeyer, Michael and Nowozin, Sebastian and Geiger, Andreas

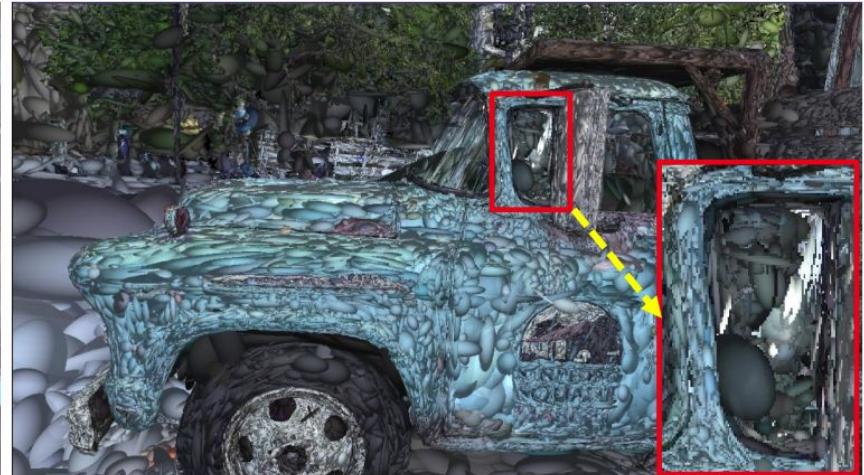
Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2019

How to represent a surface?

I - From point cloud to surfaces



Efficient RANSAC for
Point-Cloud Shape
Detection

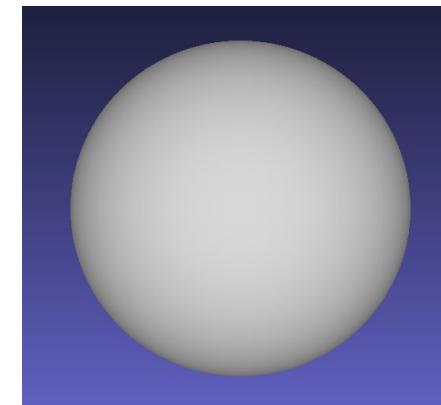
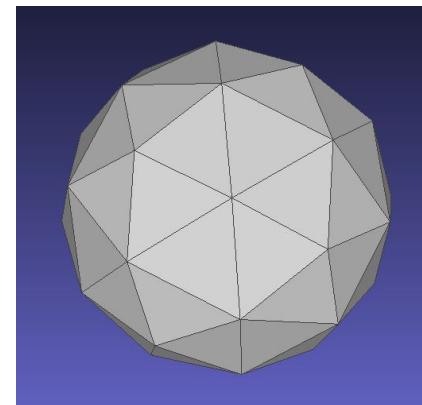
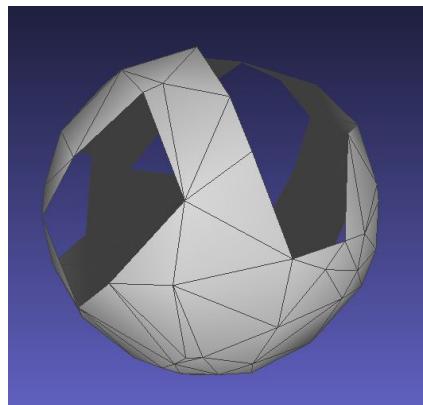
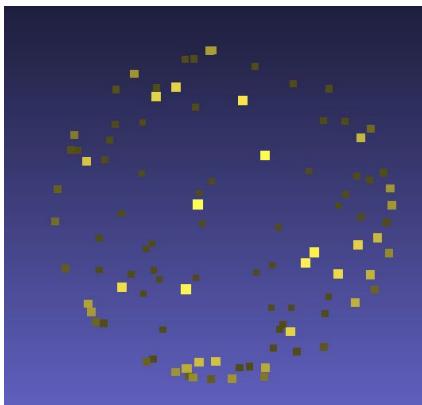


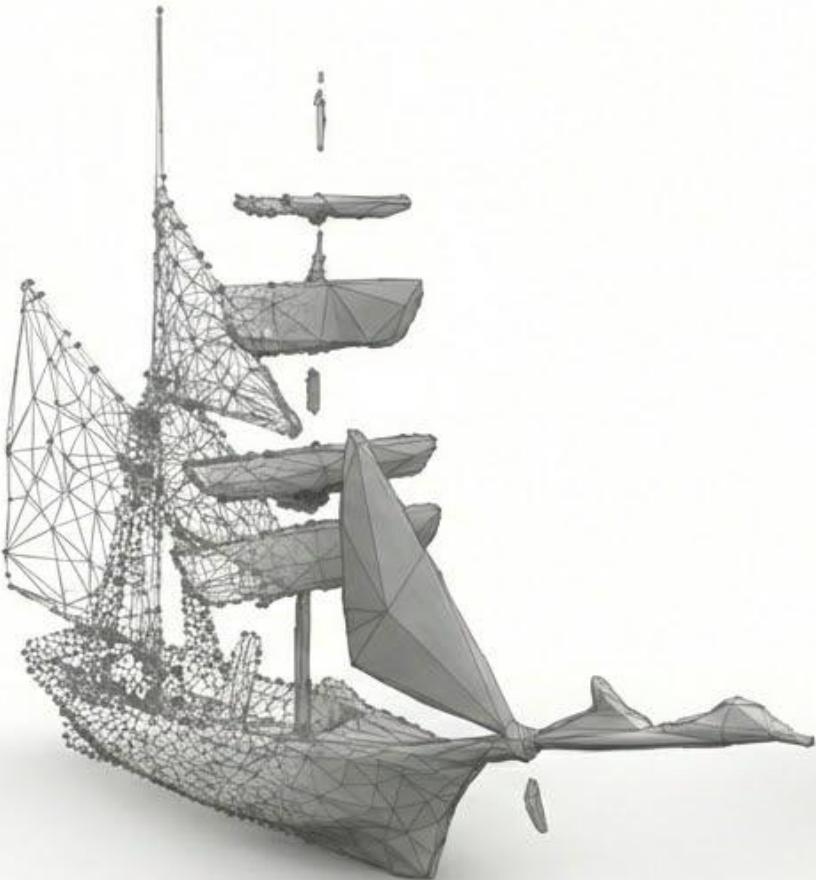
3DGS - Improving Gaussian splatting with Localized points management

What are the properties we want?

I - From point cloud to surfaces

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





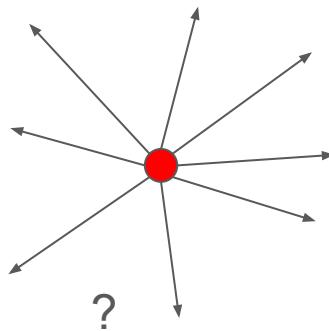
Overview

- I. From point clouds to surfaces
- II. Local features
 - A. Normal estimation
 - B. Normal orientation
- III. Surface reconstruction
 - A. Ball pivoting
 - B. Delaunay reconstruction
 - C. Poisson reconstruction
 - D. RANSAC

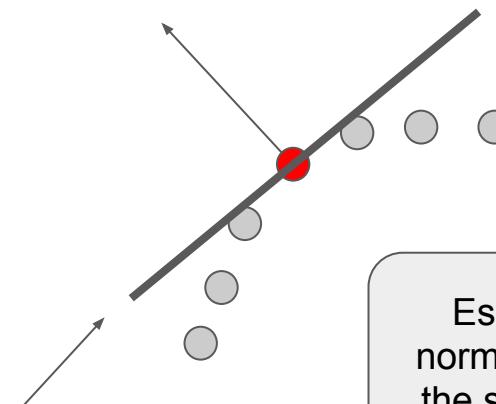
A - Normal estimation

III - Local features

A single point does not contain orientation information



Need to consider a neighborhood around a point



Tangent plane

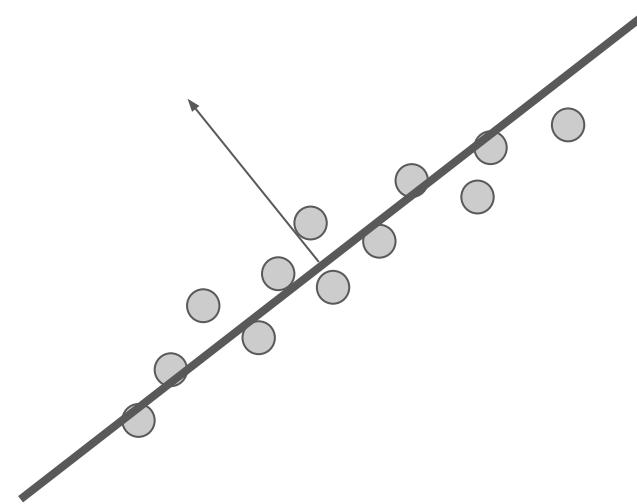
Estimating the normal \Leftrightarrow knowing the surface locally

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

III - Local features

Plane fitting with PCA

- Compute covariance matrix

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

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

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

A - Normal estimation

III - Local features

Plane fitting with PCA

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

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

$$D = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \quad P = [e_1 \quad e_2 \quad e_3]$$

A - Normal estimation

III - Local features

Plane fitting with PCA

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

$$D = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \boxed{\lambda_3} \end{pmatrix} \quad P = [e_1 \quad e_2 \quad \boxed{e_3}]$$

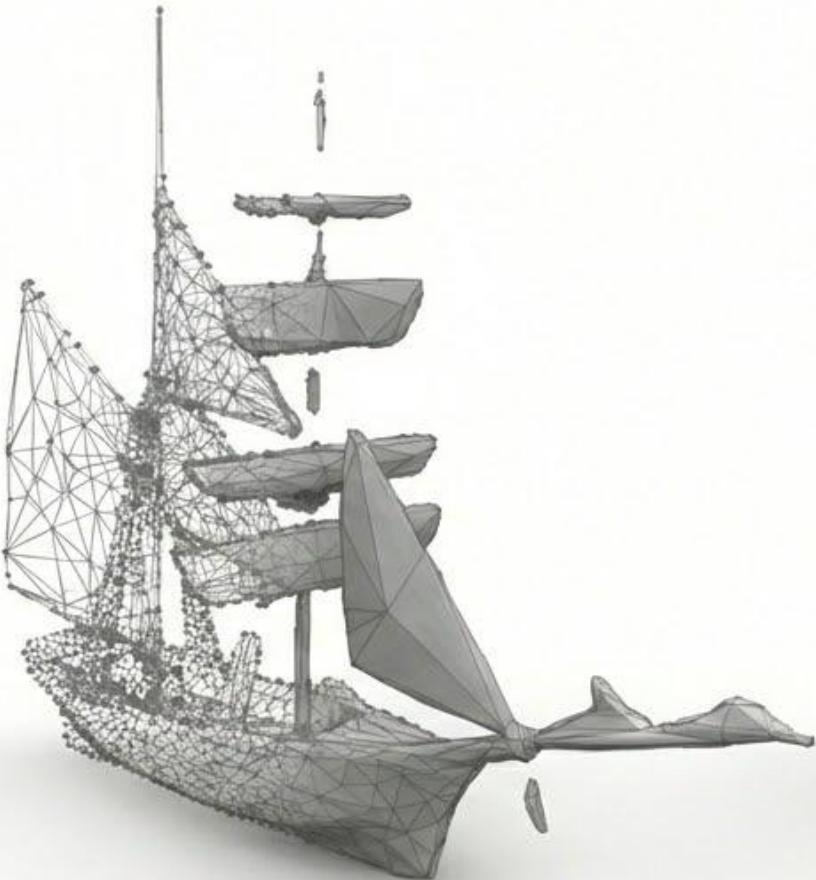
A - Normal estimation

III - Local features

Plane fitting with PCA

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

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



Overview

- I. From point clouds to surfaces
- II. Local features
 - A. Normal estimation
 - B. Normal orientation
- III. Surface reconstruction
 - A. Ball pivoting
 - B. Delaunay reconstruction
 - C. Poisson reconstruction
 - D. RANSAC

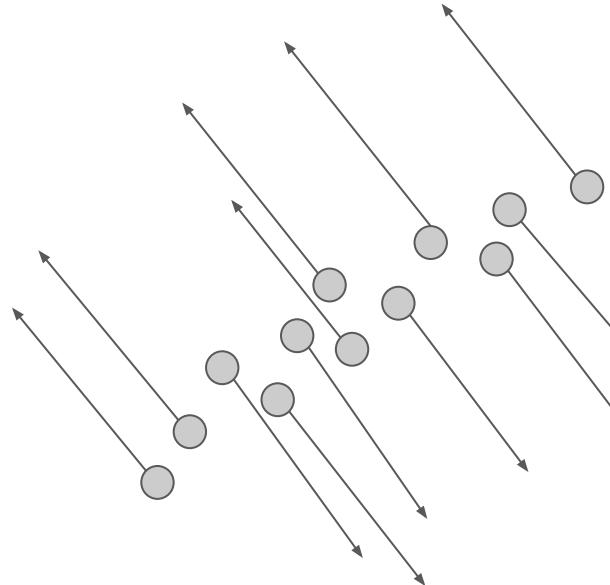
B - Normal orientation

III - Local features

The plane fitting algorithm:

✓ direction

✗ orientation

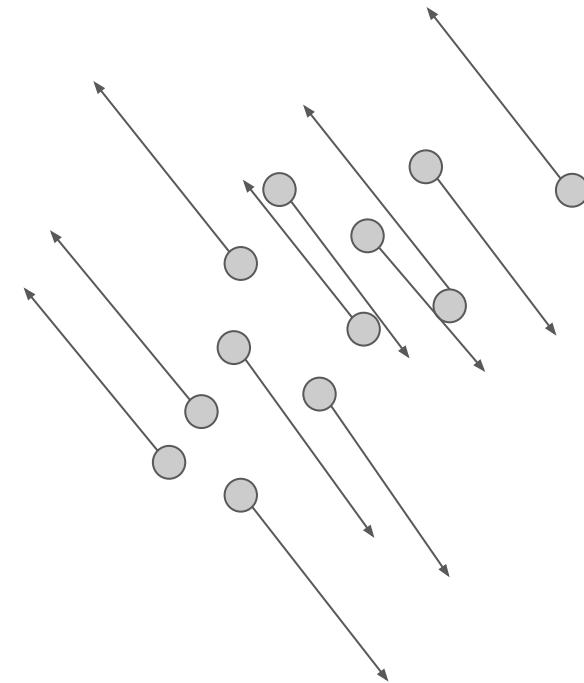


B - Normal orientation

III - Local features

Normal orientation with minimal spanning tree

Idea: propagate the orientation from one seed



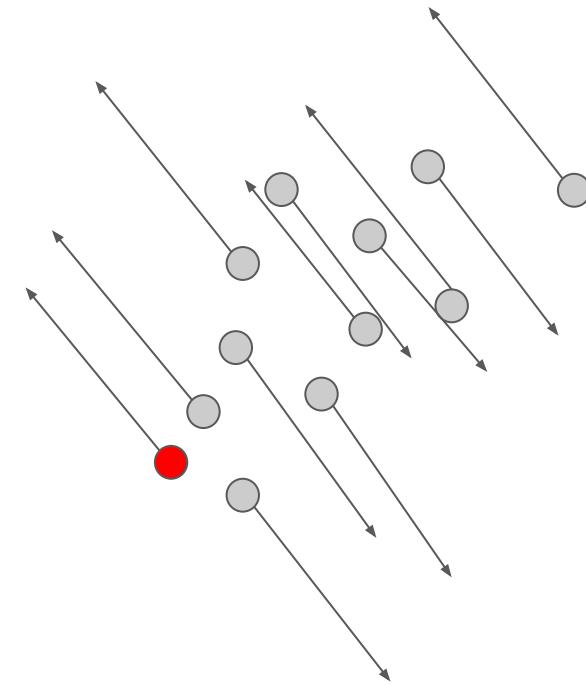
B - Normal orientation

III - Local features

Normal orientation with minimal spanning tree

Idea: propagate the orientation from one seed

1. Select a seed



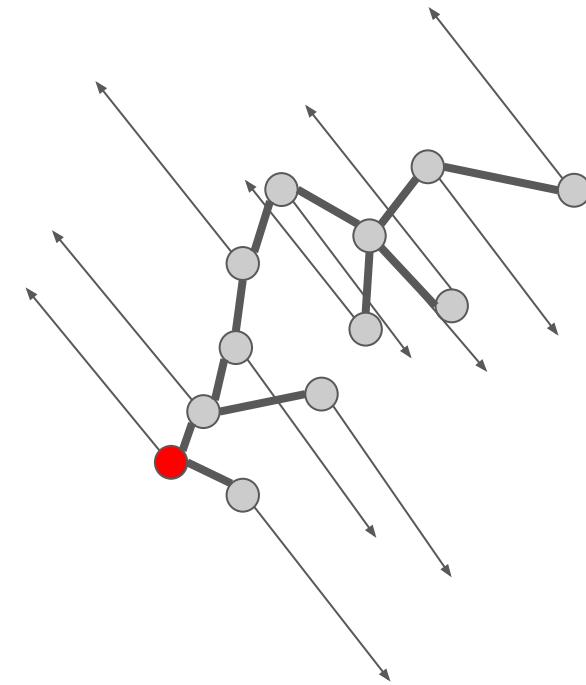
B - Normal orientation

III - Local features

Normal orientation with minimal spanning tree

Idea: propagate the orientation from one seed

1. Select a seed
2. Build a spanning tree



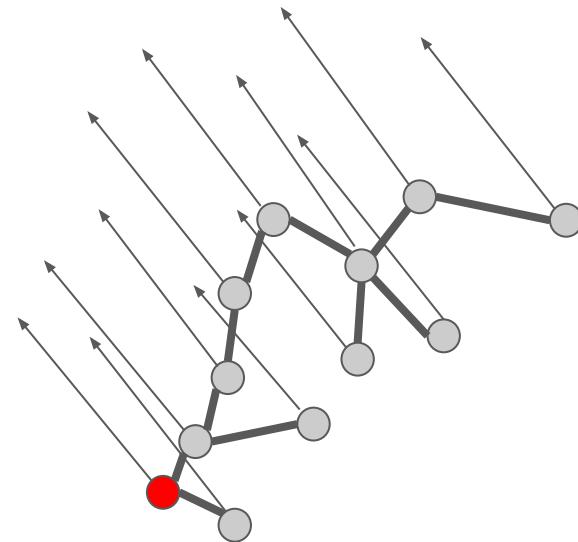
B - Normal orientation

III - Local features

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

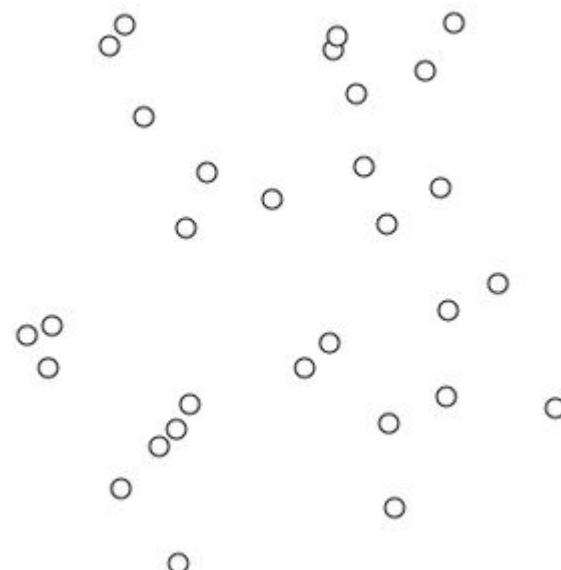


B - Normal orientation

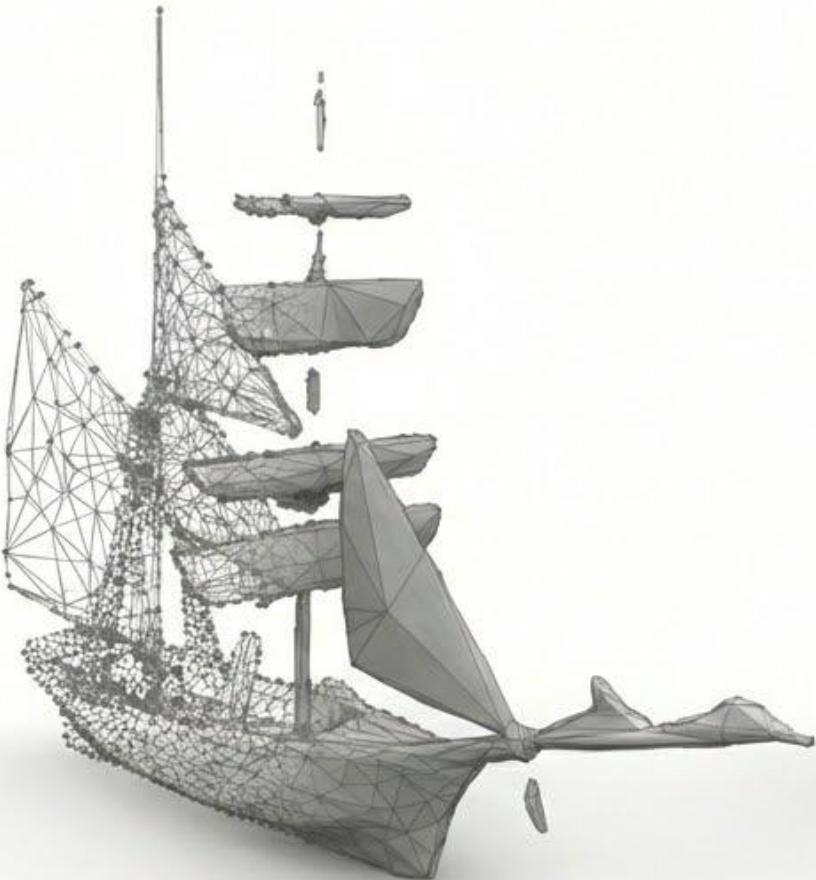
III - Local features

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



Overview

- I. From point clouds to surfaces
- II. Local features
 - A. Normal estimation
 - B. Normal orientation
- III. Surface reconstruction**
 - A. Ball pivoting
 - B. Delaunay reconstruction
 - C. Poisson reconstruction
 - D. RANSAC

Ball pivoting

III - Surface reconstruction

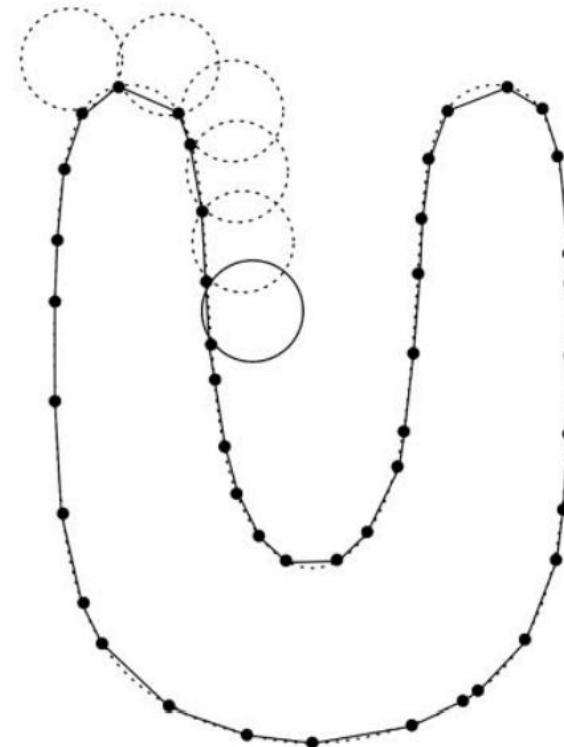
Principles:

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

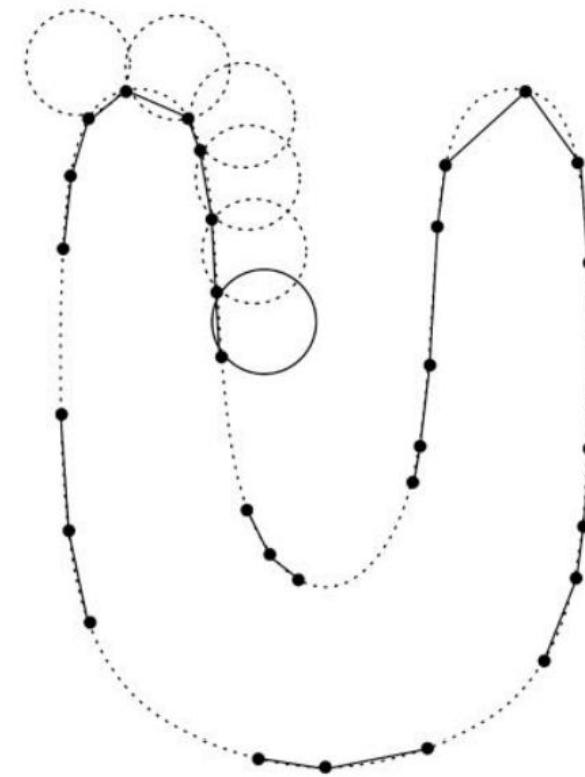
Iterate over the unexplored edges



Ball pivoting

III - Surface reconstruction

Space larger than sphere diameter creates a hole.



Ball pivoting

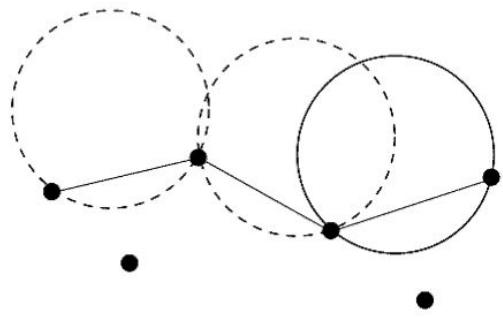
III - Surface reconstruction

Sphere with to big radius leads to a loss of details

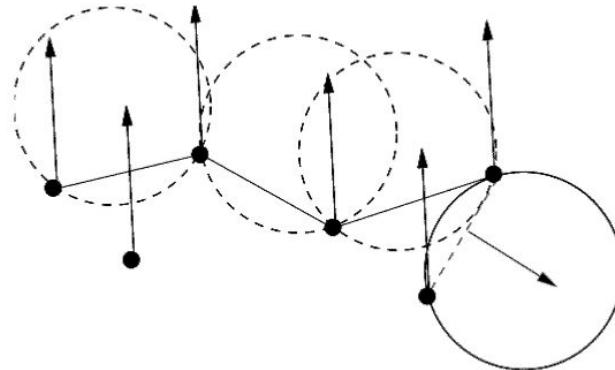


Ball pivoting

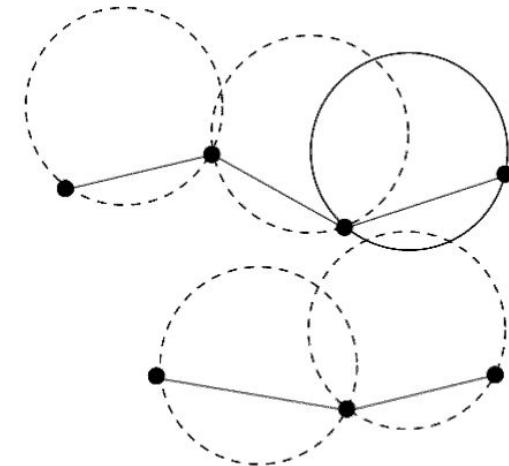
III - Surface reconstruction



Presence of noise



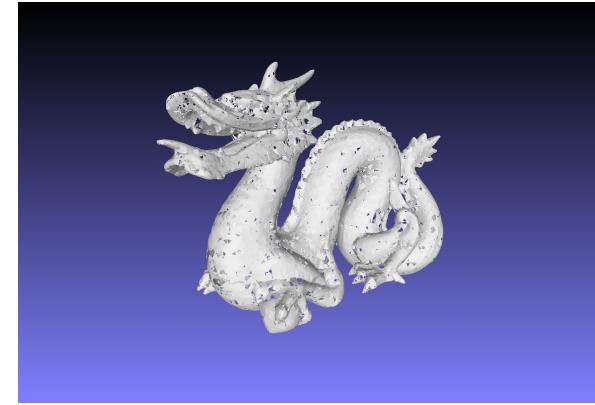
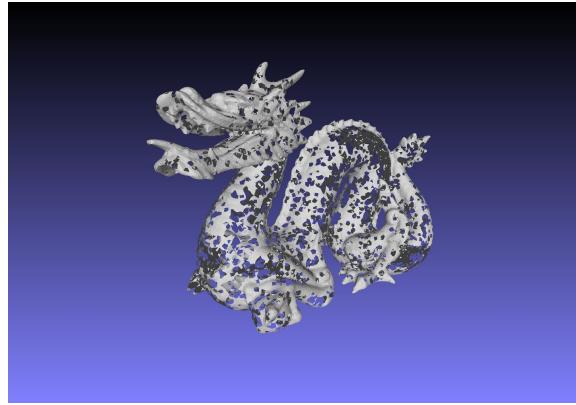
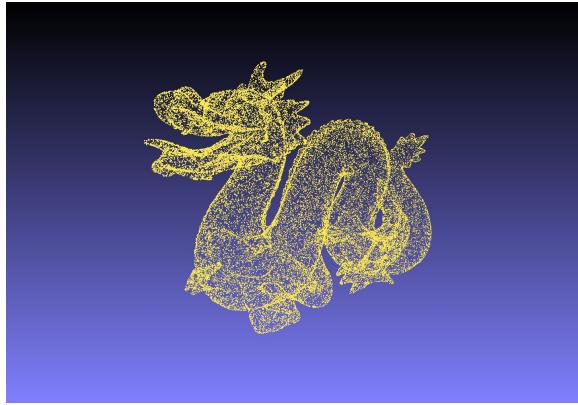
Filter according to
normals



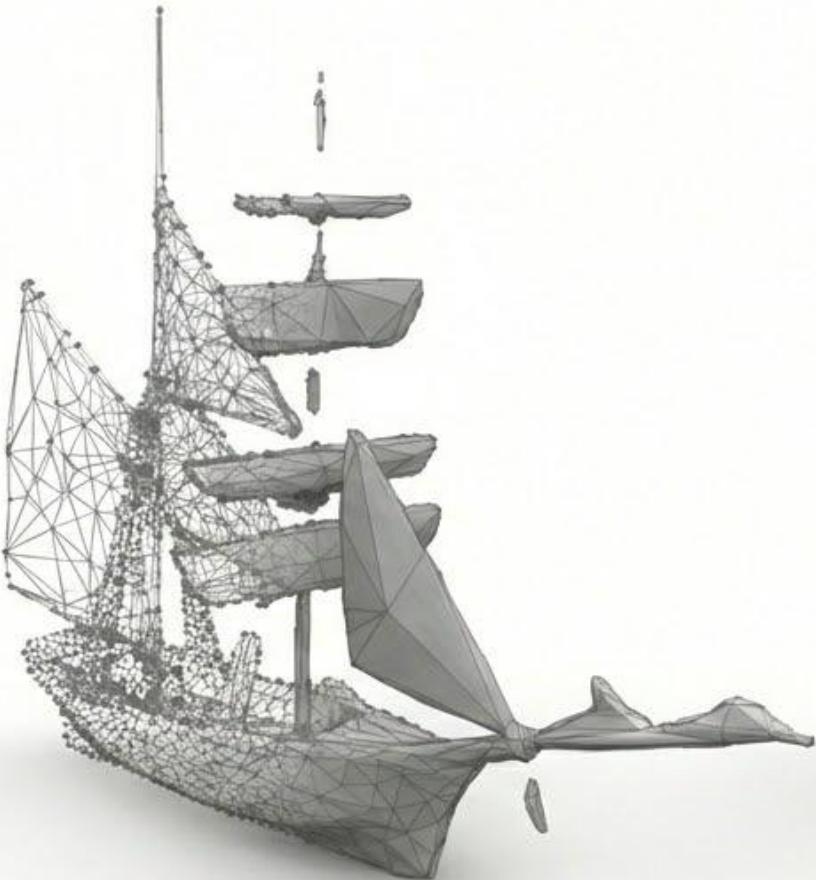
Double side
surface

Ball pivoting

III - Surface reconstruction



Increasing sphere radius



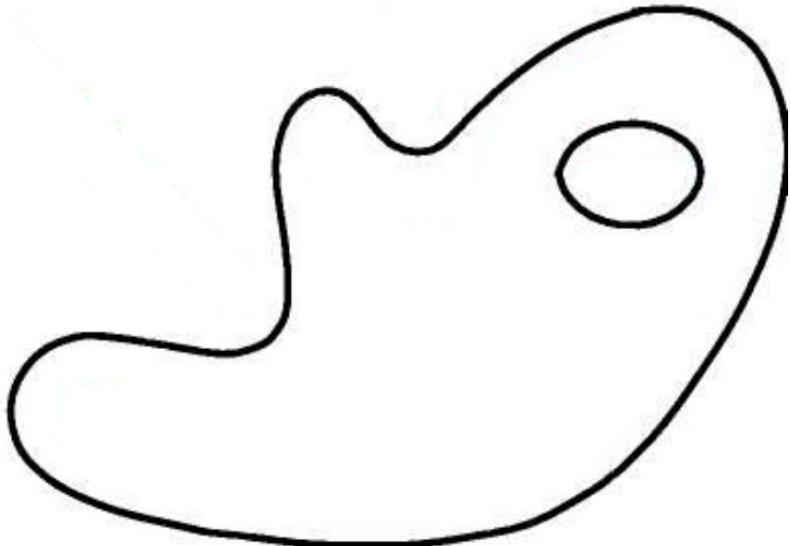
Overview

- I. From point clouds to surfaces
- II. Local features
 - A. Normal estimation
 - B. Normal orientation
- III. Surface reconstruction**
 - A. Ball pivoting
 - B. Delaunay reconstruction
 - C. Poisson reconstruction
 - D. RANSAC

Surface

B - Delauney reconstruction

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



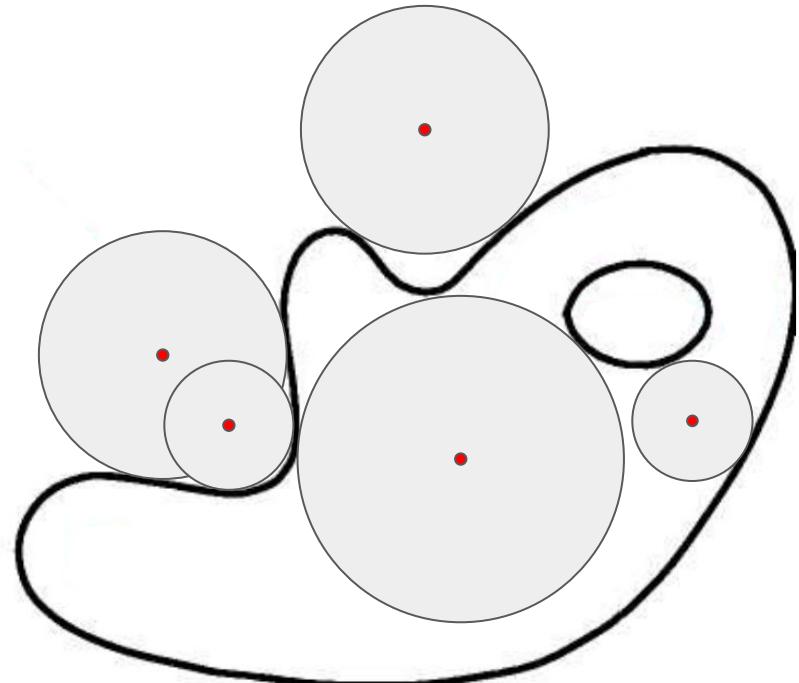
Medial axis

B - Delauney reconstruction

Let's put a sphere such that:

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

Let's consider all the possible spheres



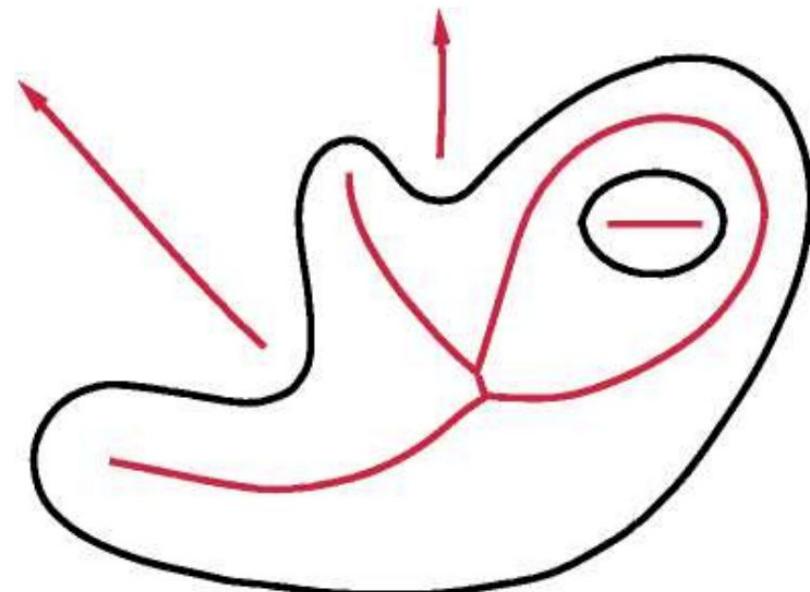
Medial axis

B - Delauney reconstruction

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

The medial axis is the dual of the
surface.

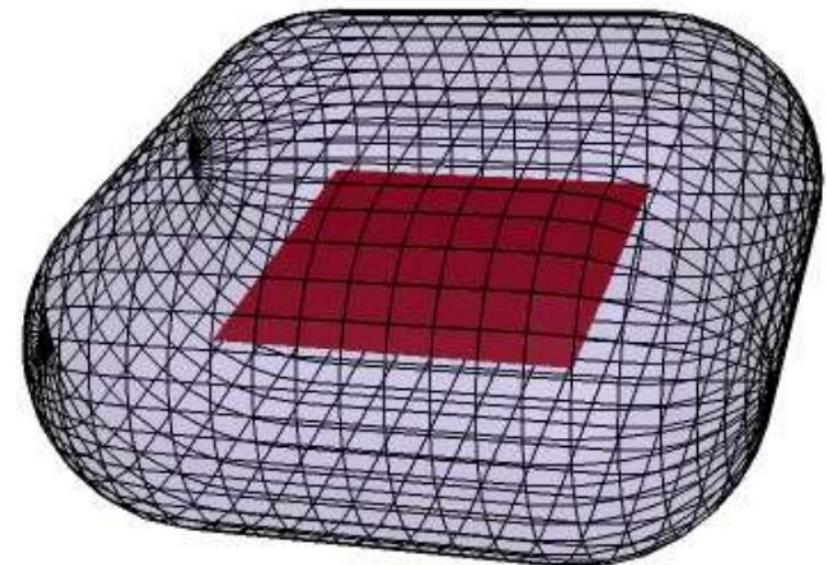
⇒ from a medial axis, it is possible to
find the surface



Medial axis

B - Delauney reconstruction

The medial axis generalizes to 3D



Voronoi diagram

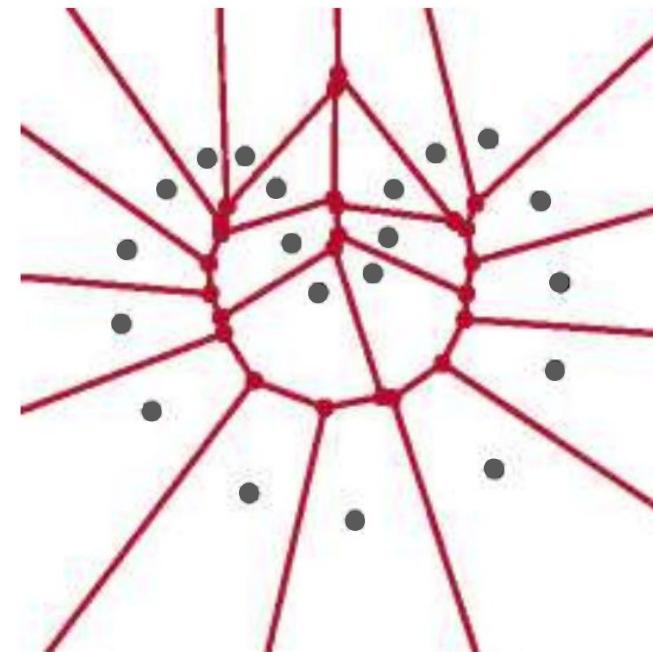
B - Delauney reconstruction

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

⇒ yes this the Voronoi diagram

On an edge → equal distance to 2 points

On a node → equal distance to 3 points

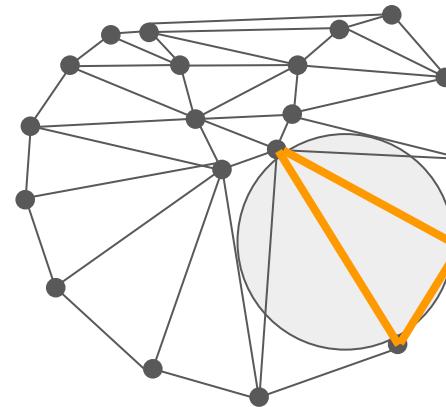


Voronoi diagram

B - Delauney reconstruction

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

Iterate over the triplets.

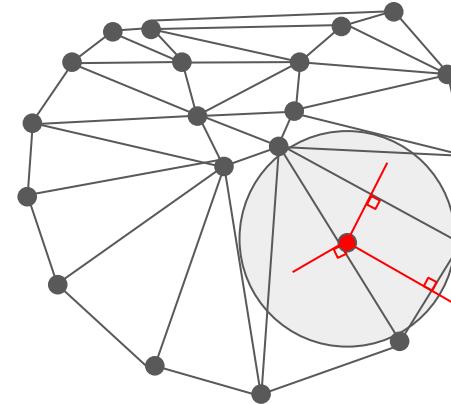


Voronoi diagram

B - Delauney reconstruction

Create the graph.

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

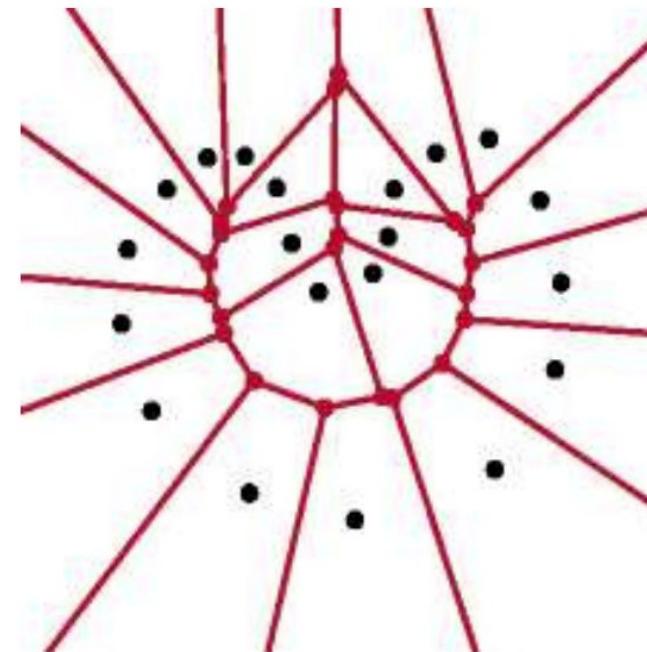


Voronoi diagram

B - Delauney reconstruction

Create the graph.

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



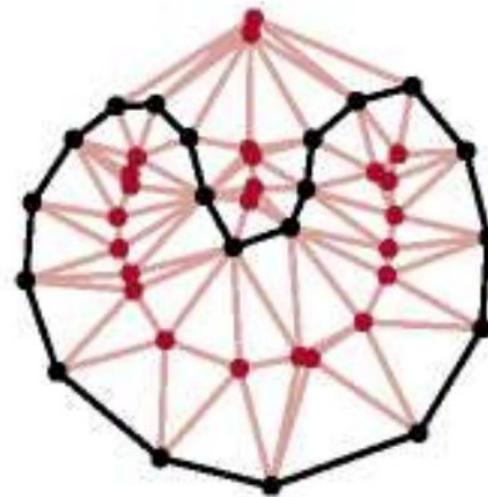
Augmented Delauney triangulation

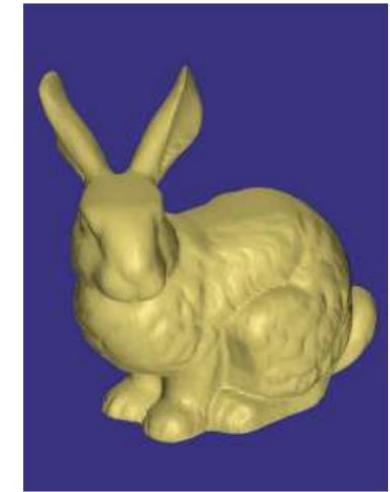
B - Delauney reconstruction

Recompute the Delaunay triangulation
with the Voronoï nodes.

Remove edges linked to Voronoï nodes

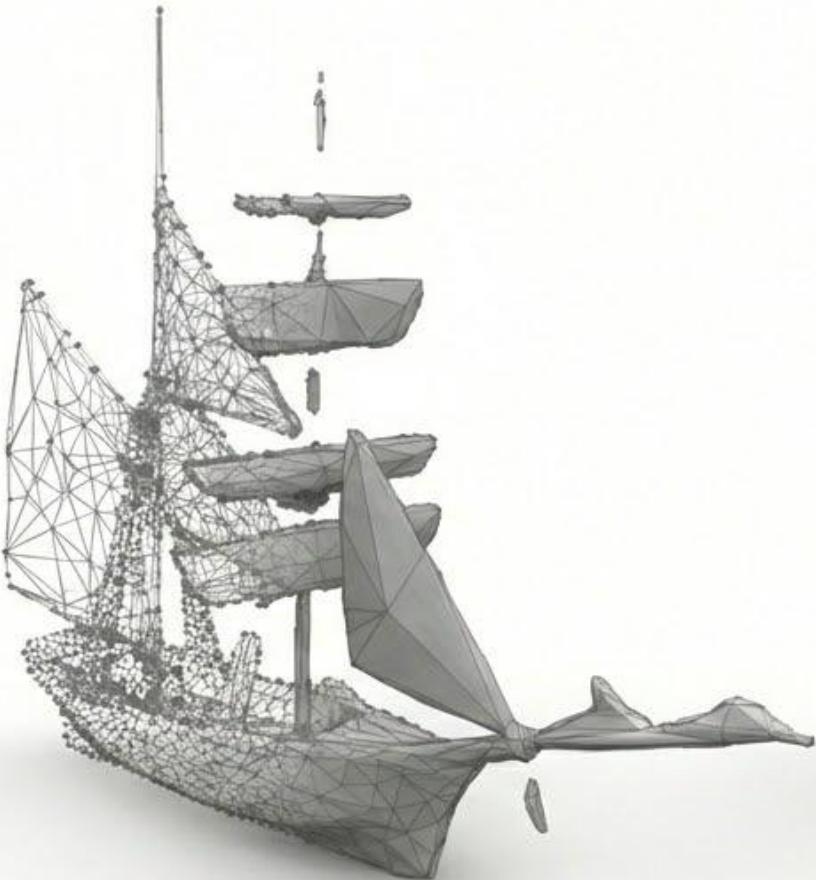
The remaining is the **crust**.





Pros: theoretically grounded, very good results without noise

Cons: not smooth, cannot handle noise



Overview

- I. From point clouds to surfaces
- II. Local features
 - A. Normal estimation
 - B. Normal orientation
- III. Surface reconstruction**
 - A. Ball pivoting
 - B. Delaunay reconstruction
 - C. Poisson reconstruction
 - D. RANSAC

Overview

C - The Poisson reconstruction pipeline



Oriented points

$$\vec{V}$$

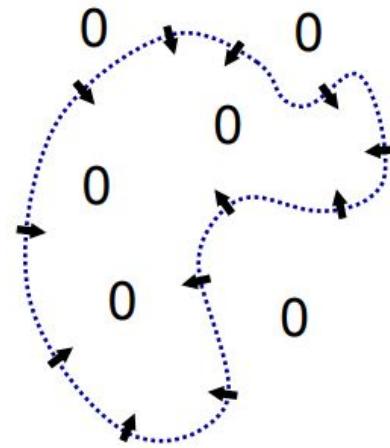
Overview

C - The Poisson reconstruction pipeline



Oriented points

$$\vec{V}$$



Indicator gradient

$$\nabla \chi_M$$

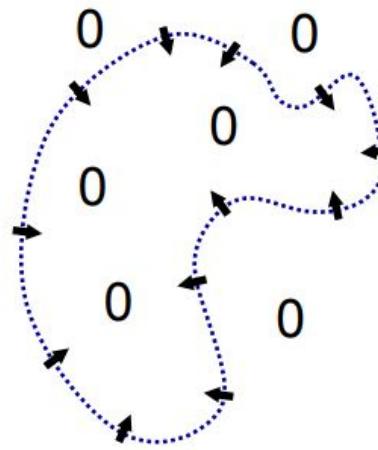
Overview

C - The Poisson reconstruction pipeline



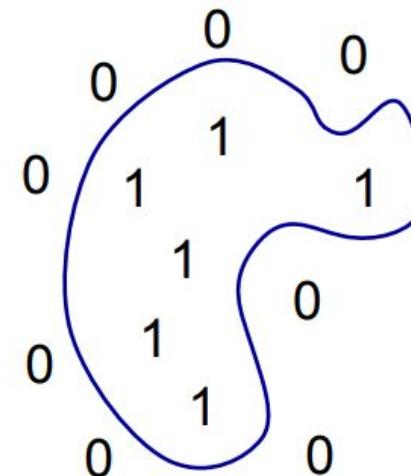
Oriented points

$$\vec{V}$$



Indicator gradient

$$\nabla \chi_M$$



Indicator function

$$\chi_M$$

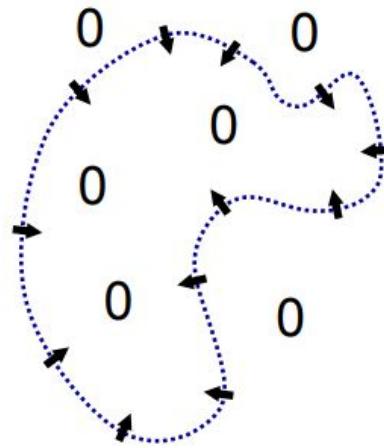
Overview

C - The Poisson reconstruction pipeline



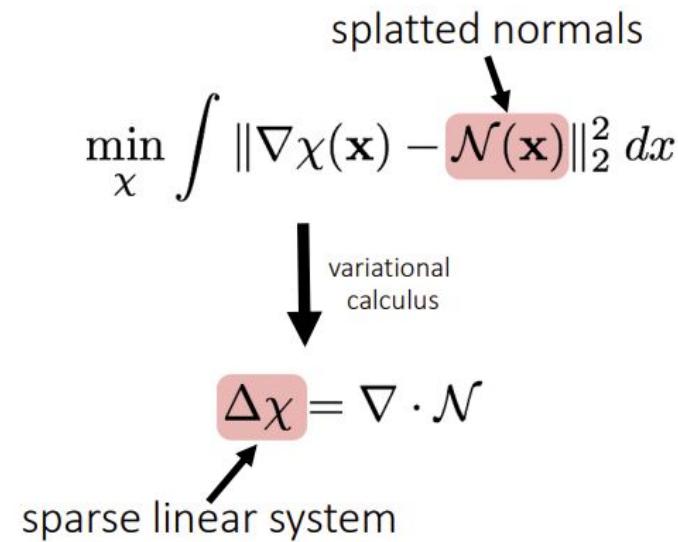
Oriented points

$$\vec{V}$$



Indicator gradient

$$\nabla \chi_M$$



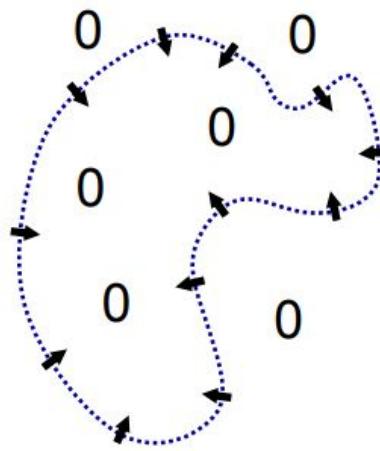
Overview

C - The Poisson reconstruction pipeline



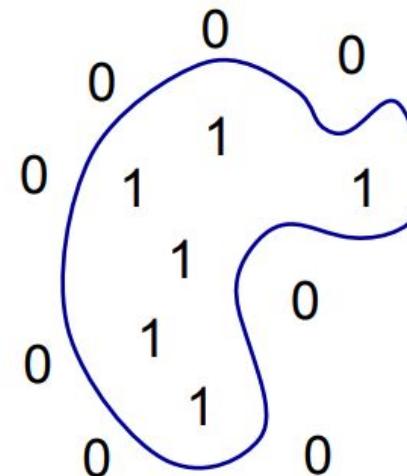
Oriented points

$$\vec{V}$$



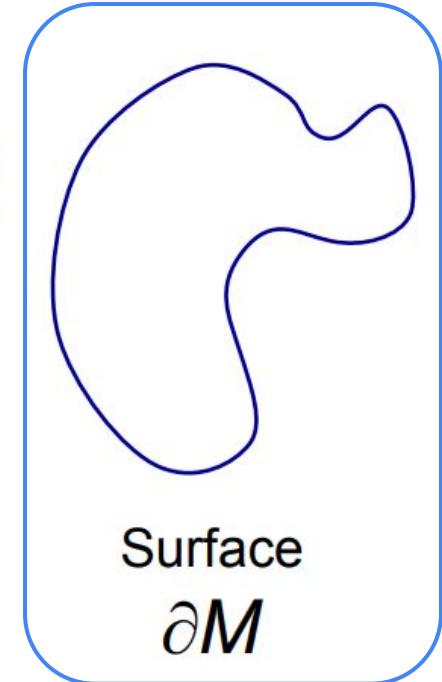
Indicator gradient

$$\nabla \chi_M$$



Indicator function

$$\chi_M$$

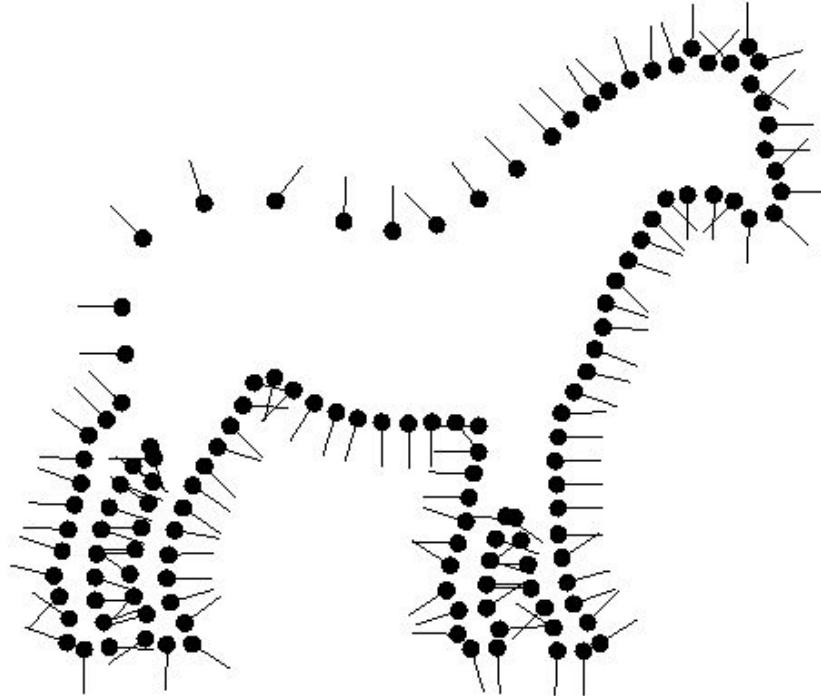


In practice

C - The Poisson reconstruction pipeline

Given the Points:

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

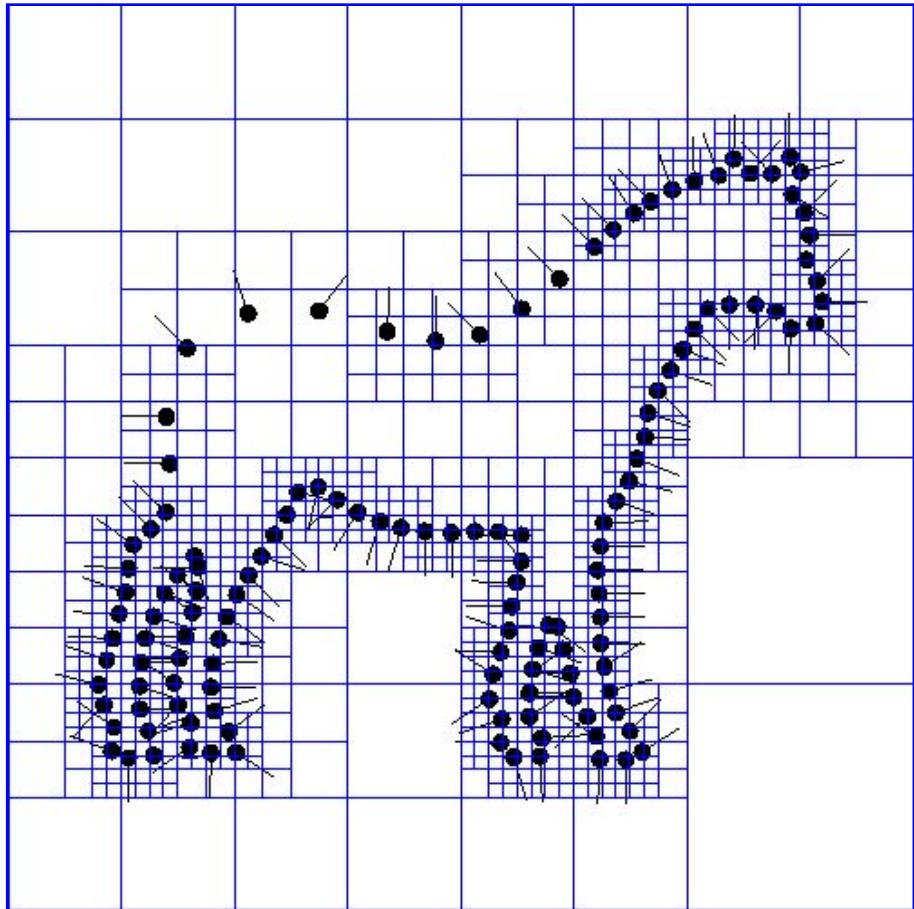


In practice

C - The Poisson reconstruction pipeline

Given the Points:

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

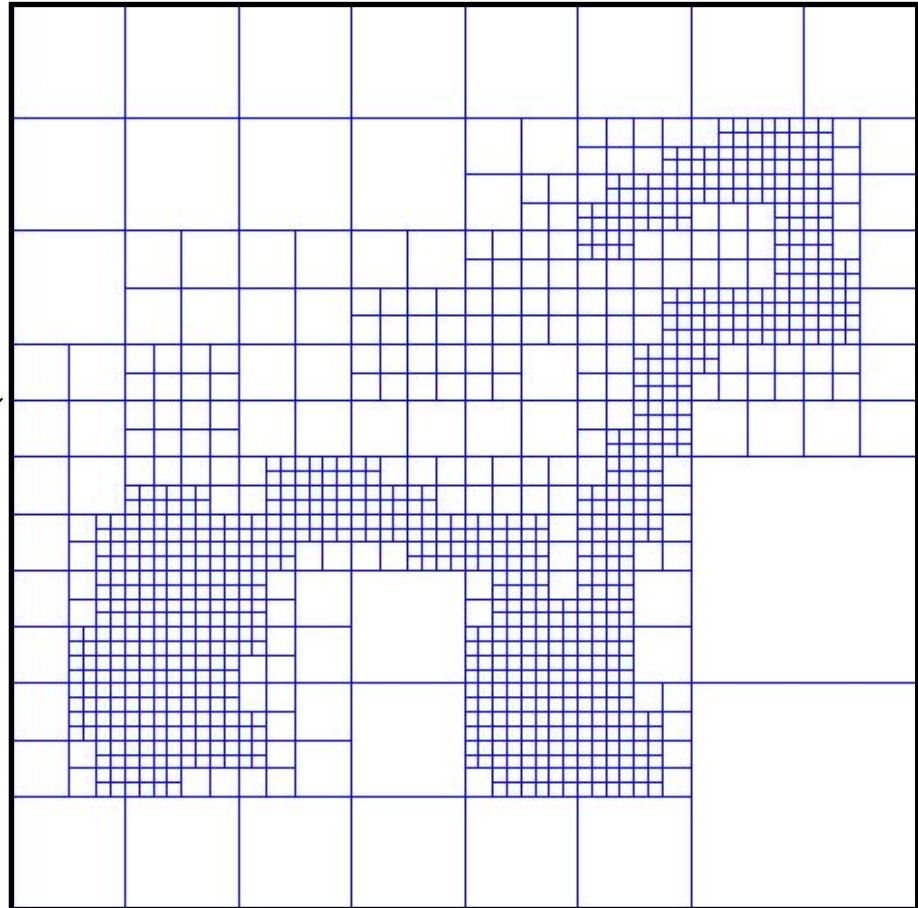
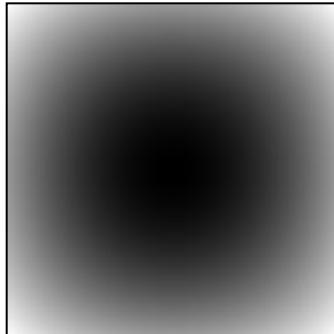


In practice

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

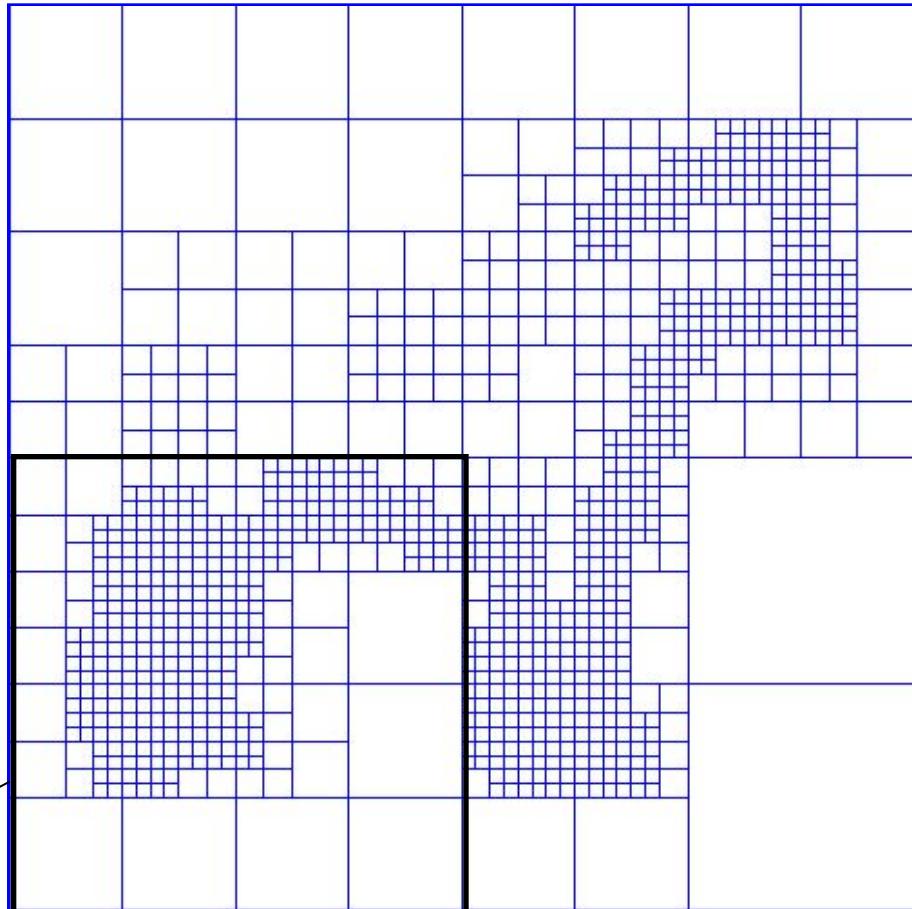


In practice

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

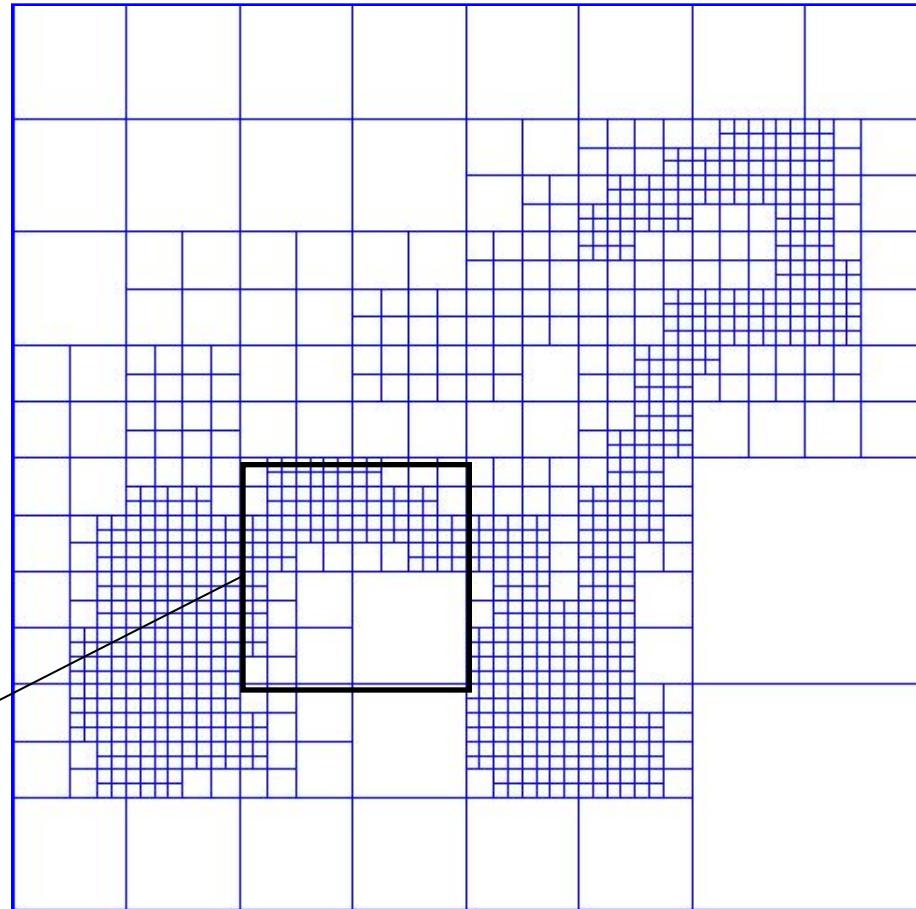
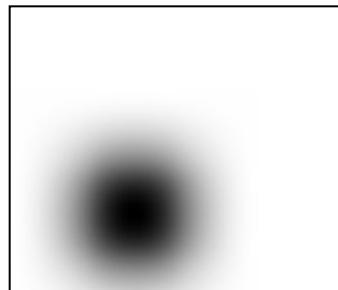


In practice

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

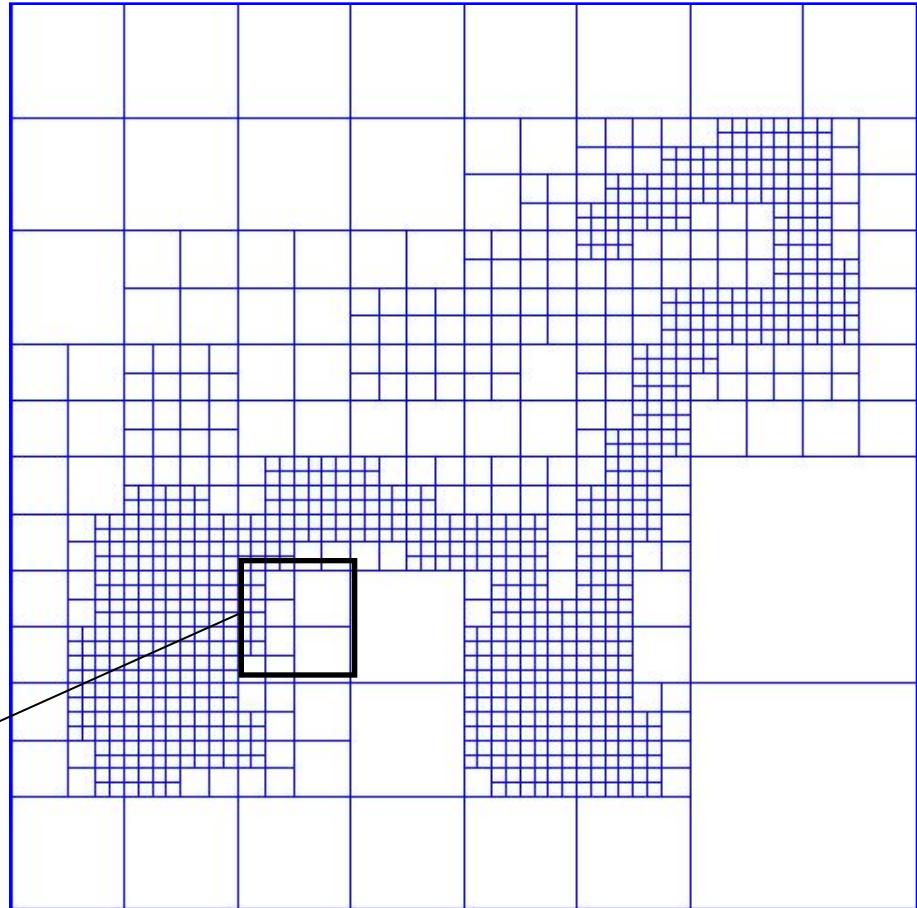
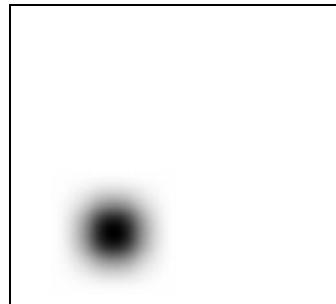


In practice

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

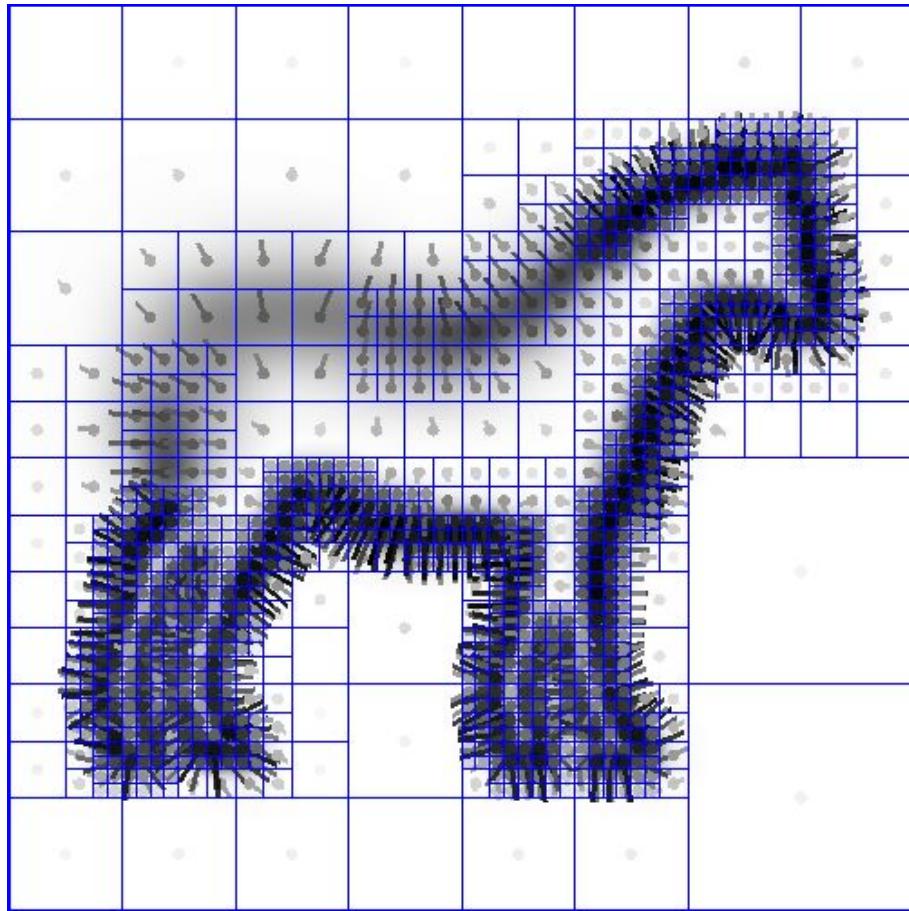


In practice

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

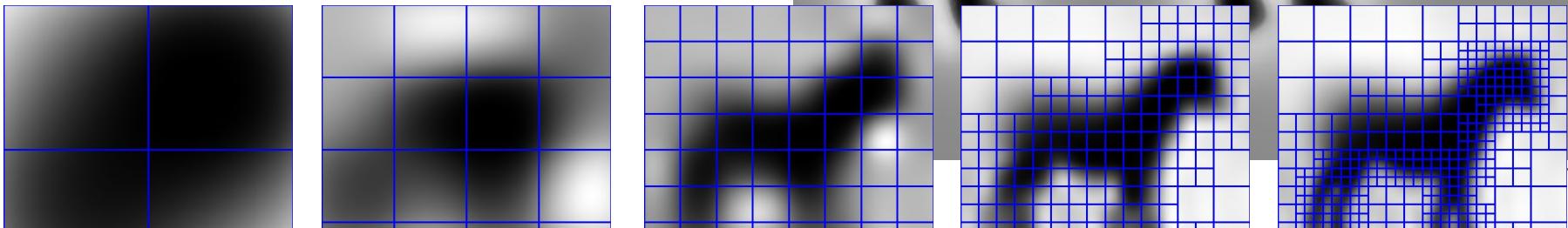


In practice

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

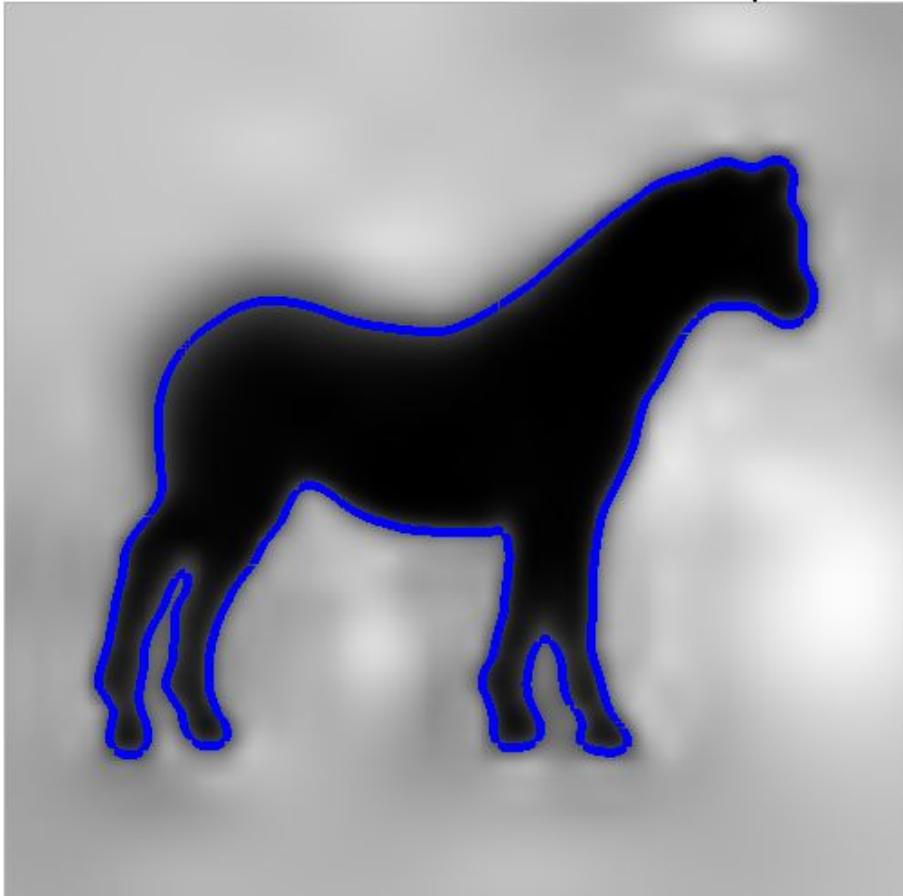


In practice

C - The Poisson reconstruction pipeline

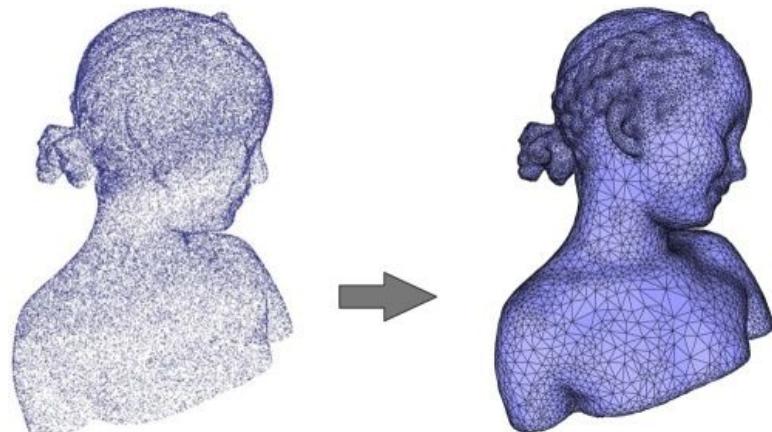
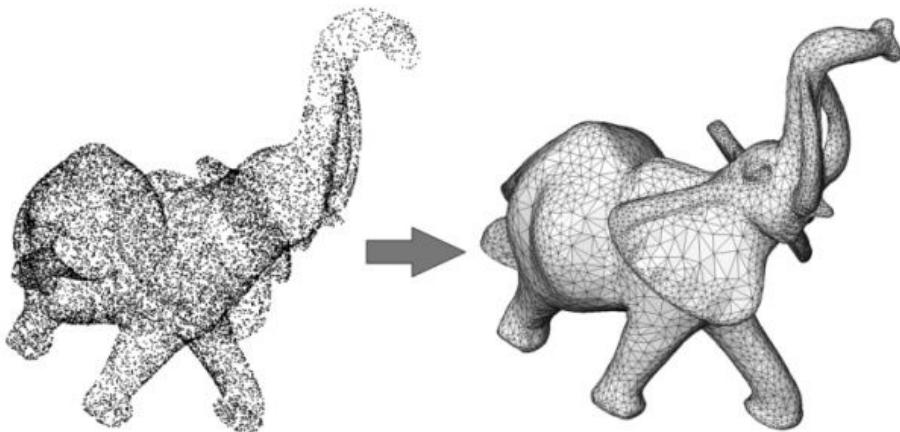
Given the Points:

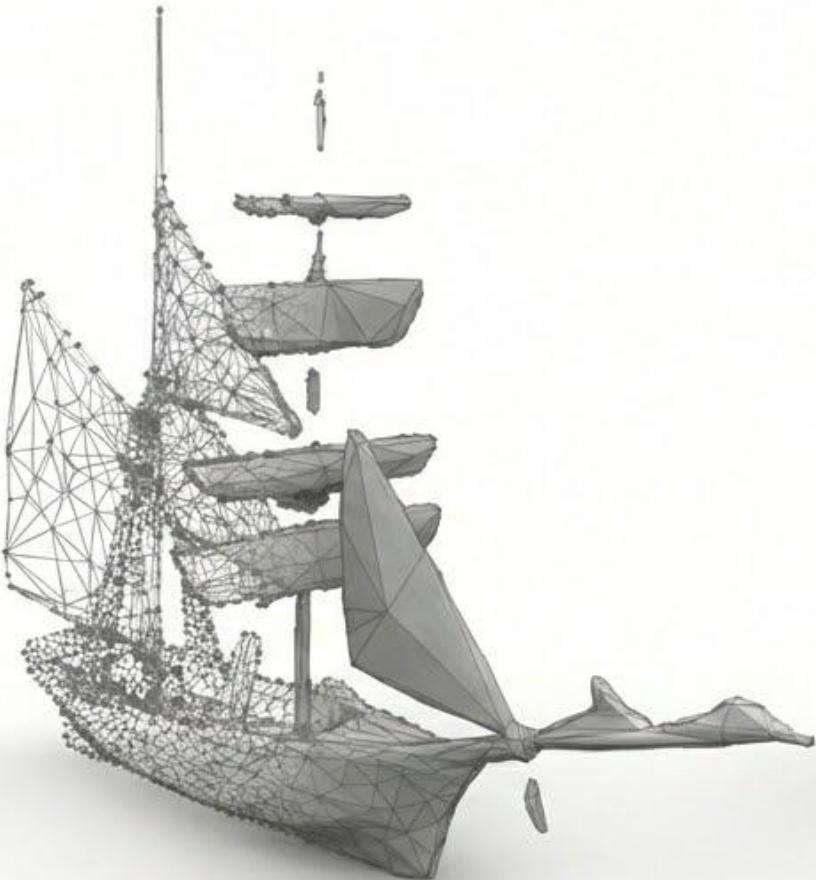
- Set octree
- Compute vector field
- Compute indicator function
- **Extract iso-surface**



Examples

C - The Poisson reconstruction pipeline





Overview

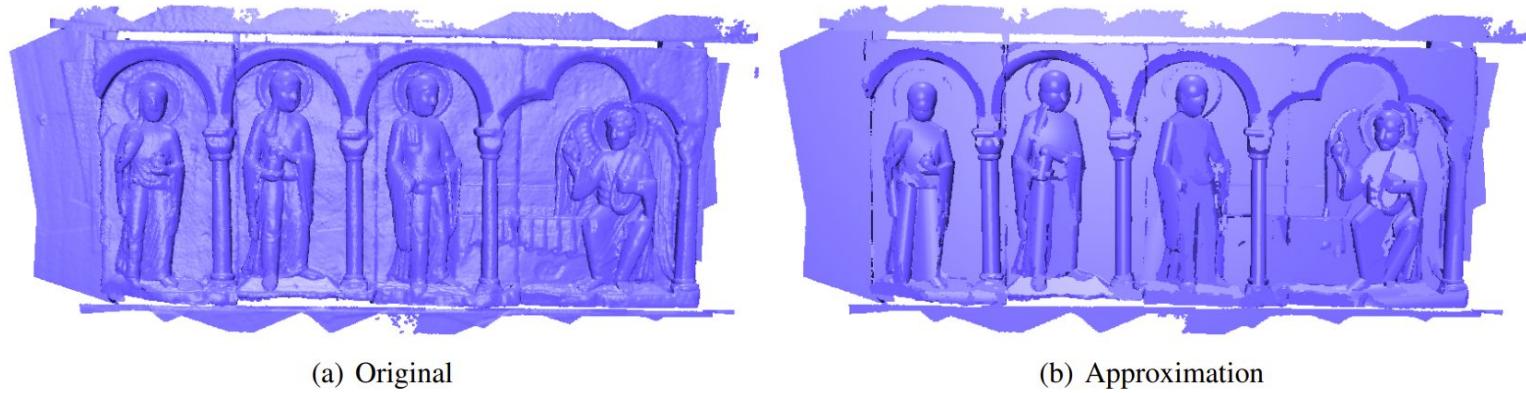
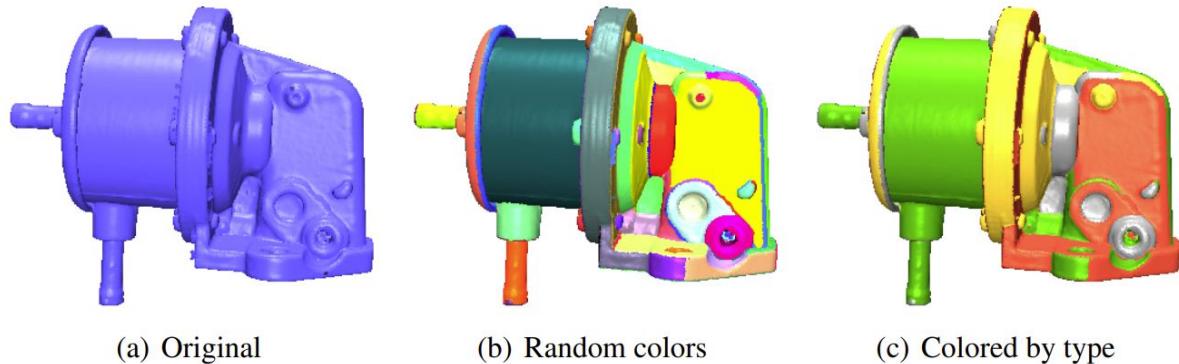
- I. From point clouds to surfaces
- II. Local features
 - A. Normal estimation
 - B. Normal orientation
- III. Surface reconstruction**
 - A. Ball pivoting
 - B. Delaunay reconstruction
 - C. Poisson reconstruction
 - D. RANSAC

Model-based approaches

Fit a model

- Abstraction
- Simplification
- CAD

...

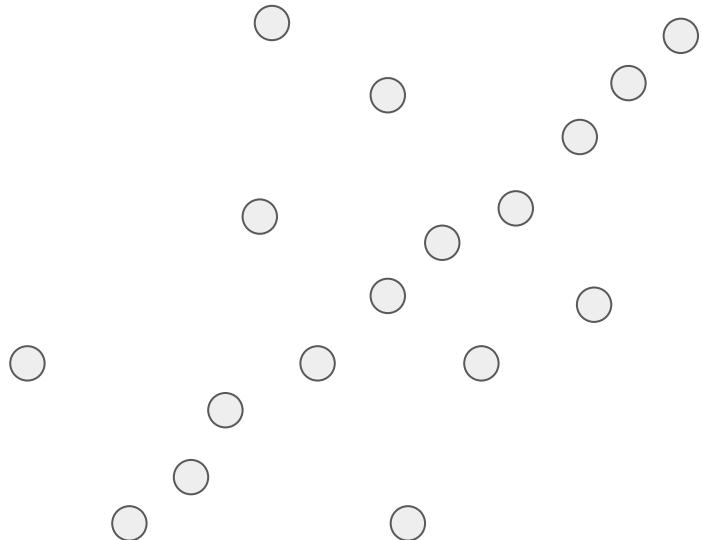


Random Sample Consensus

D - RANSAC

Random Sample Concensus

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

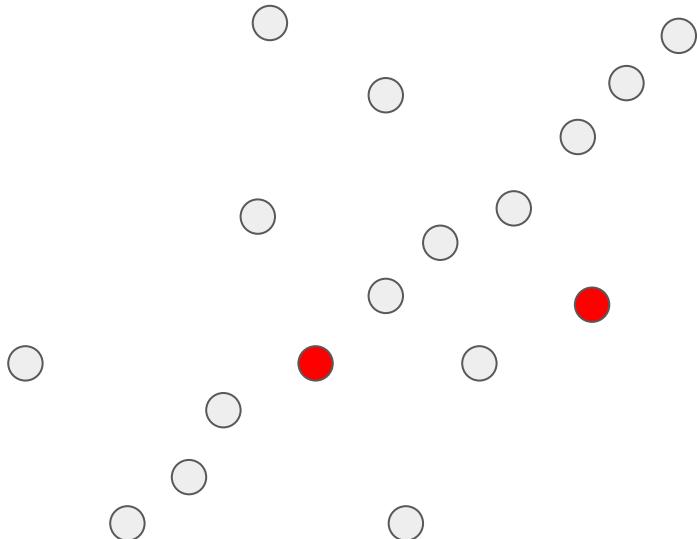


Random Sample Consensus

D - RANSAC

Random Sample Concensus

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

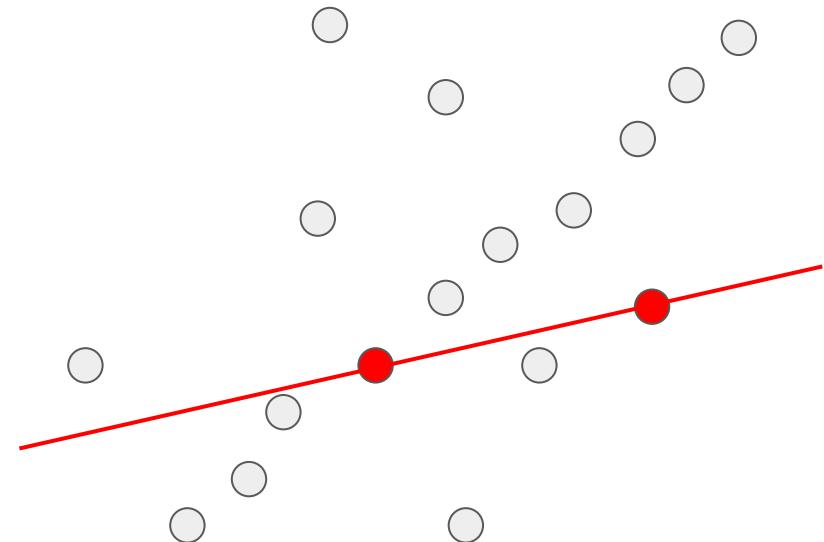


Random Sample Consensus

D - RANSAC

Random Sample Concensus

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

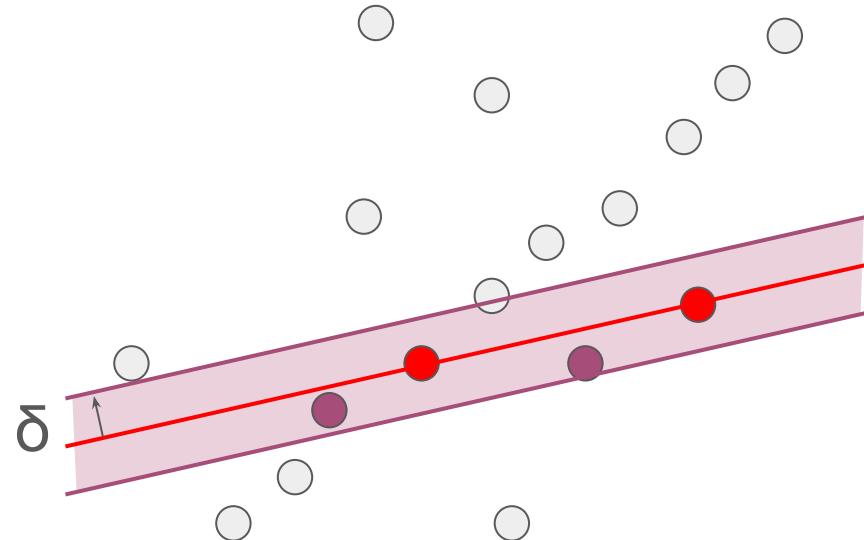


Random Sample Consensus

D - RANSAC

Random Sample Concensus

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

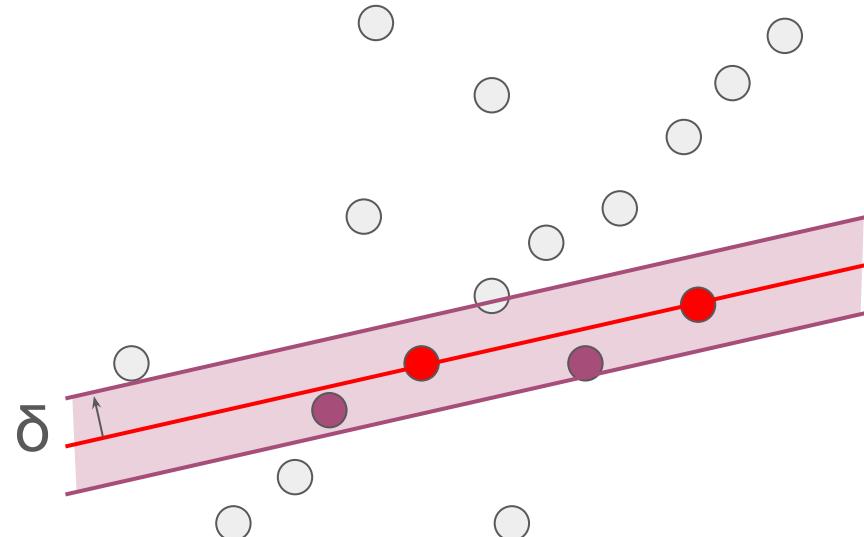


Random Sample Consensus

D - RANSAC

Random Sample Concensus

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

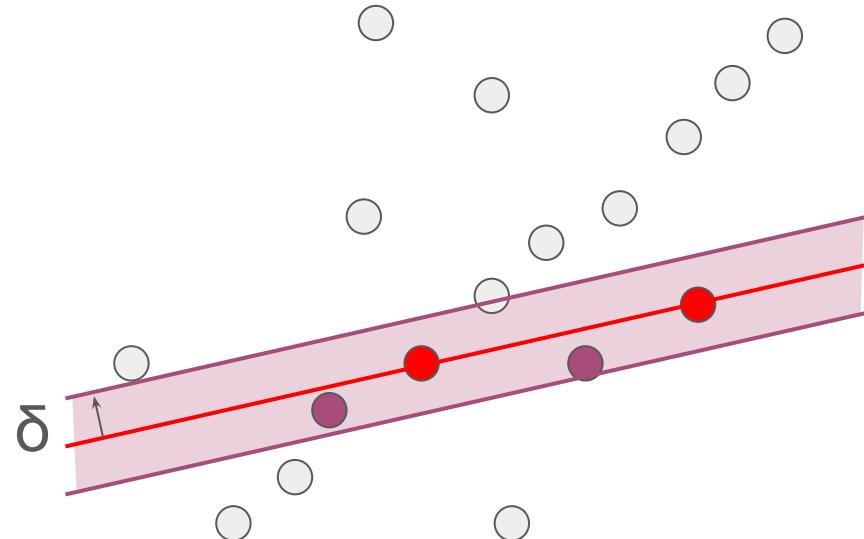


Random Sample Consensus

D - RANSAC

Random Sample Concensus

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

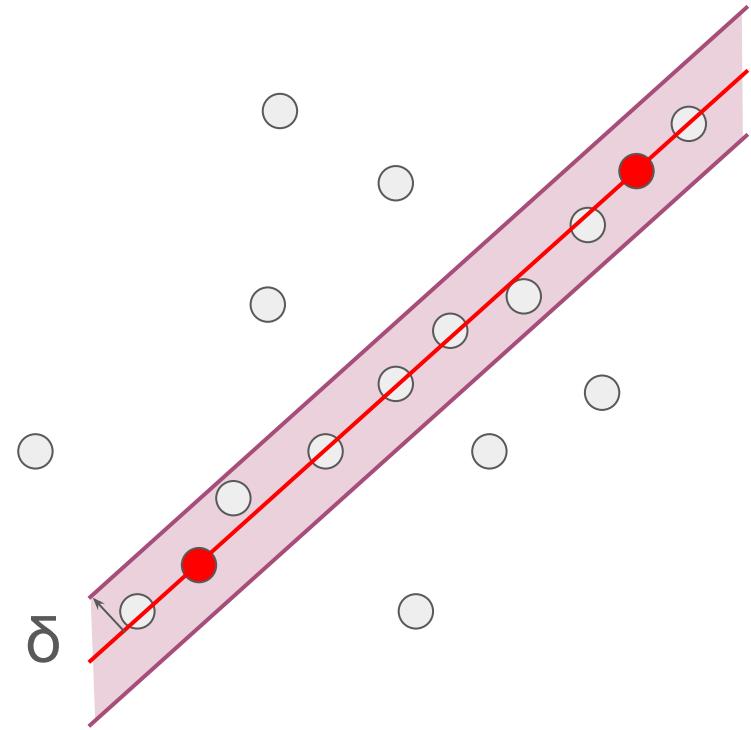


Random Sample Consensus

D - RANSAC

Random Sample Concensus

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

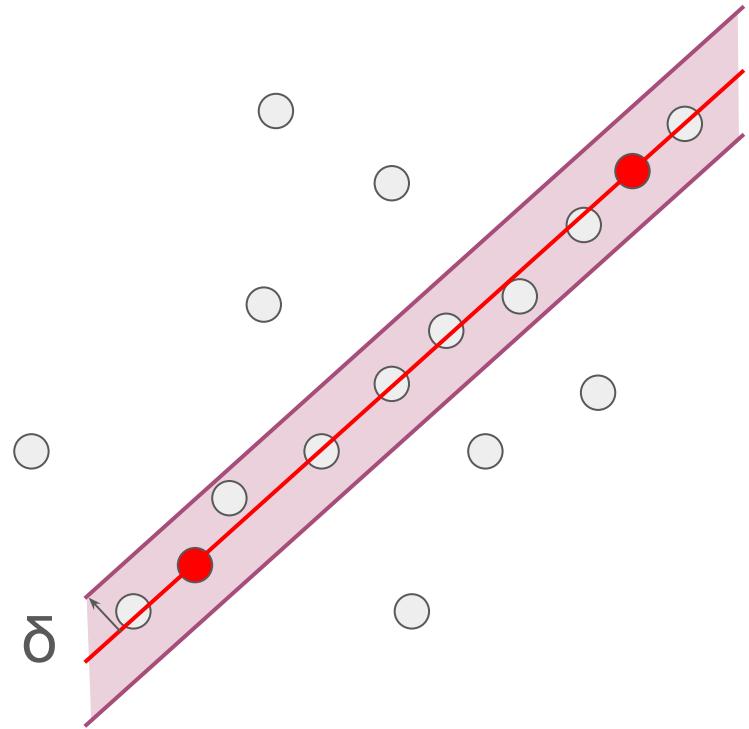


Random Sample Consensus

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

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 is 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

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

Then:

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

n and p_t are parameters to be set.

Conclusion and practical session

Surface reconstruction

Conclusion and practical session

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/UPL2159668480191960308_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>