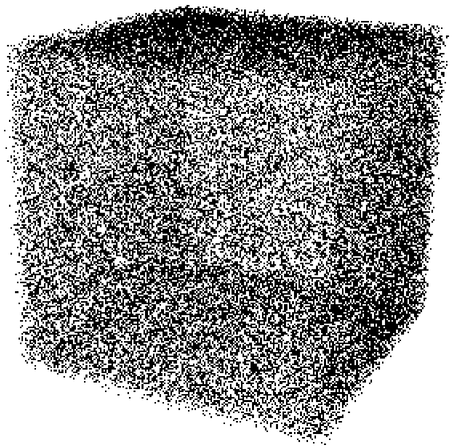
Primitive-based surface reconstruction

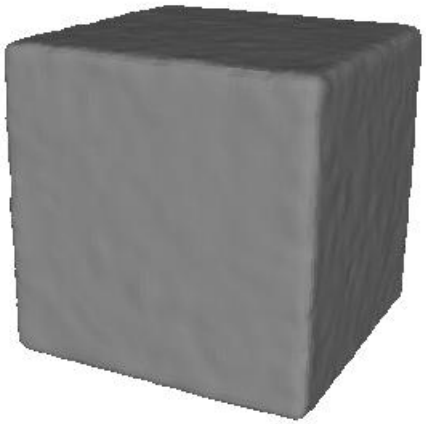
Florent Lafarge
Inria Sophia Antipolis - Mediterranee

- Geometric primitive extraction
  - Region growing
  - Ransac
  - Accumulation methods
  - Global regularities

- Surface reconstruction using geometric primitives
- Two words on template matching

Why Geometric primitives can be interesting for surface reconstruction ?



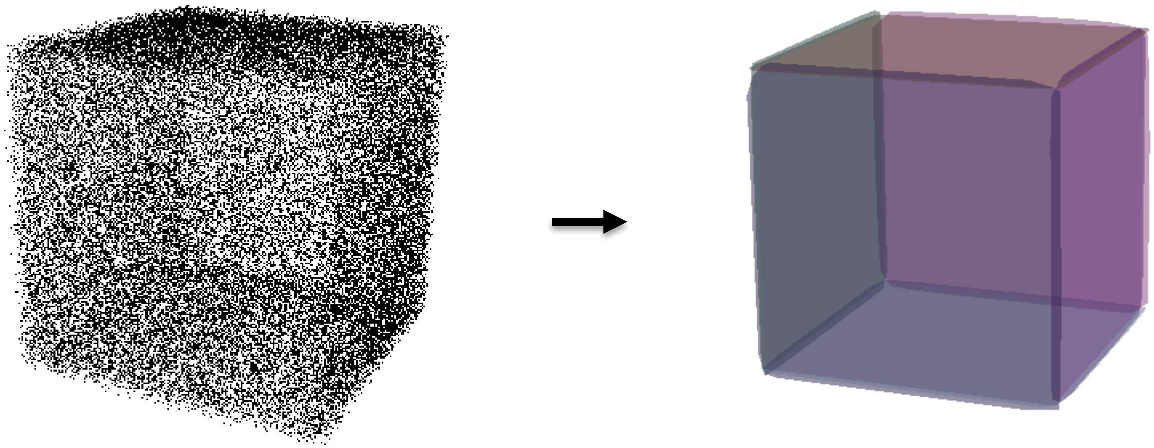


High complexity

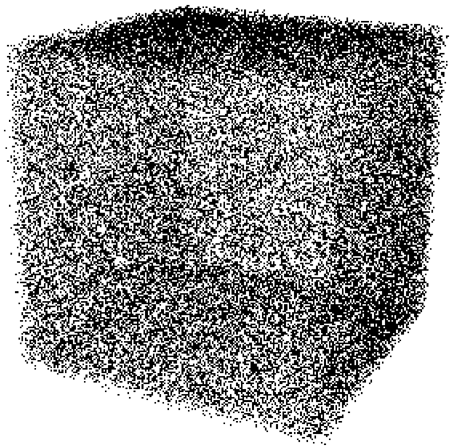
No structure

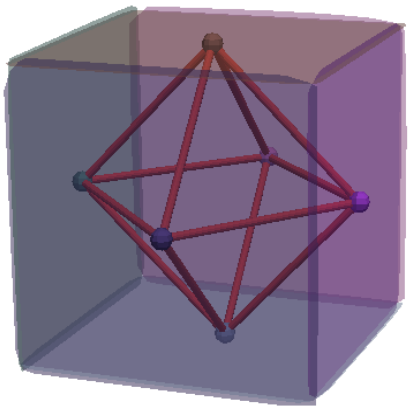
Smooth recontruction

Why Geometric primitives can be interesting for surface reconstruction ?

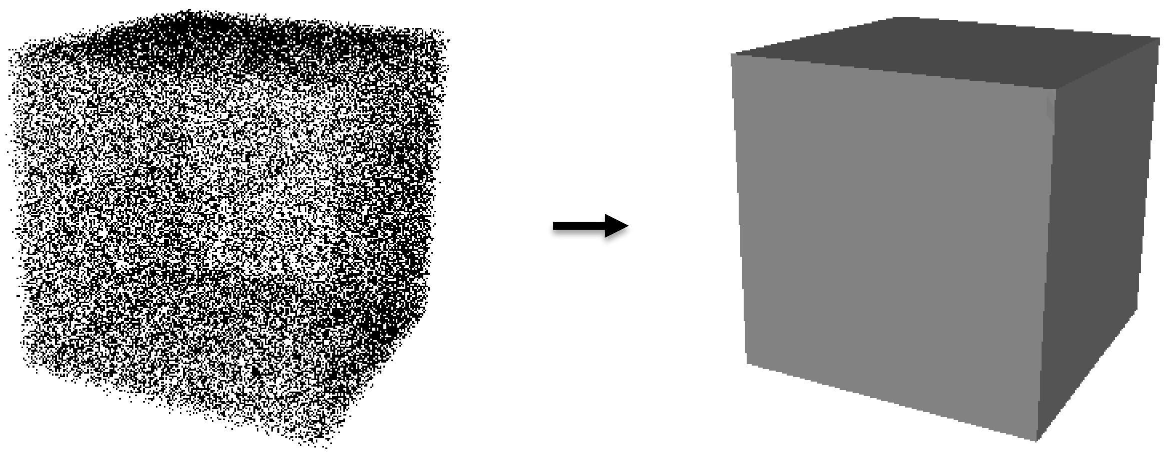


Why Geometric primitives can be interesting for surface reconstruction ?





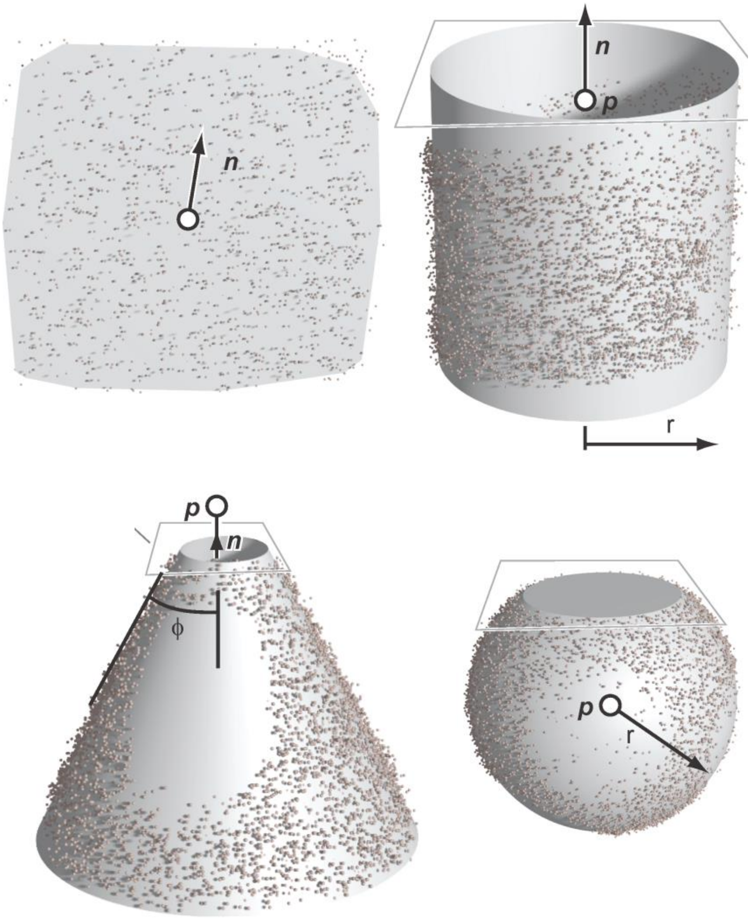
Why Geometric primitives can be interesting for surface reconstruction ?



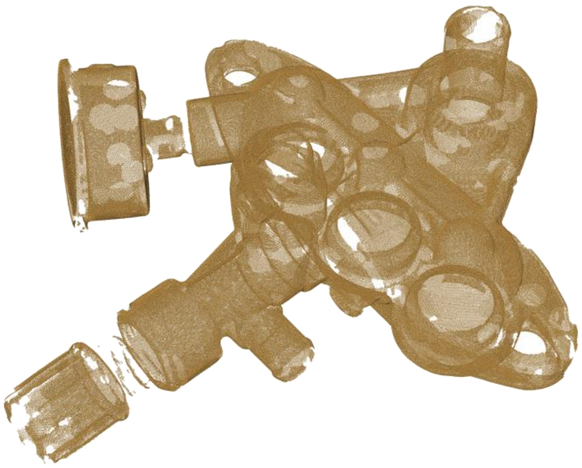
low complexity

structure

How to extract Geometric primitives from point sets ?

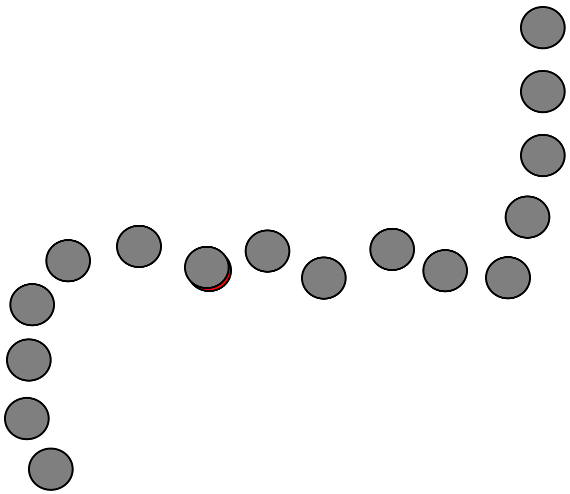


?



Region growing

- Iterative method
- Spatial propagation of a primitive

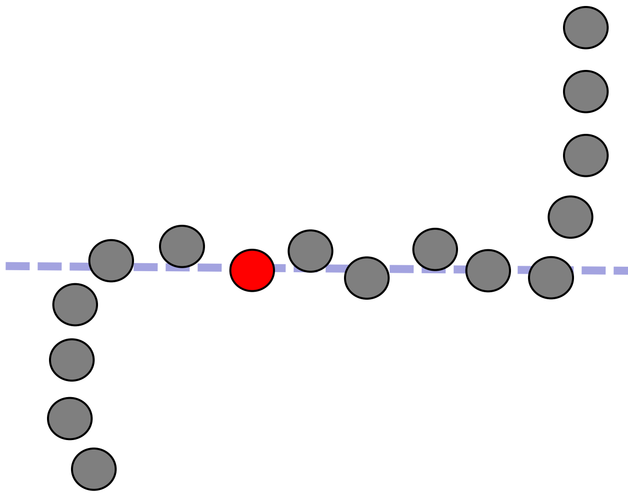


Hypothesis

- deterministic
- Efficient for relatively “clean” Data

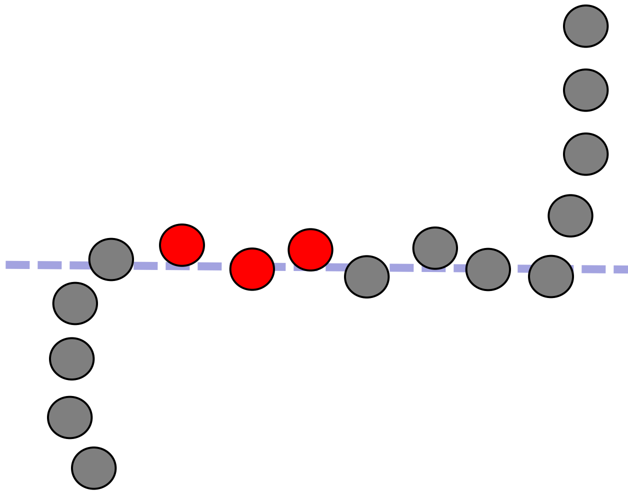
Region growing

- select a point and a primitive hypothesis

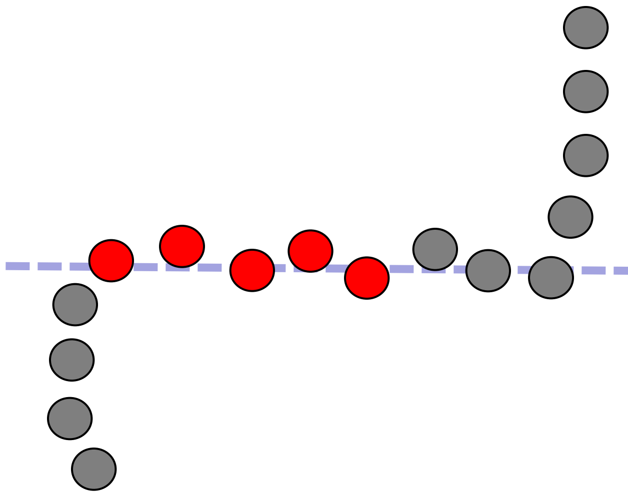


Region growing

- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis

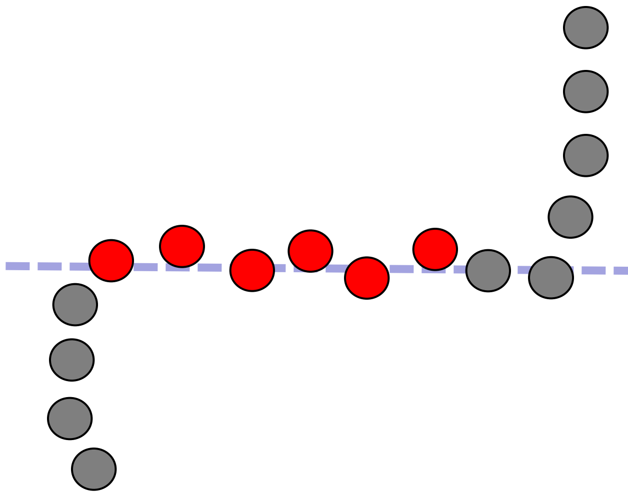


Region growing



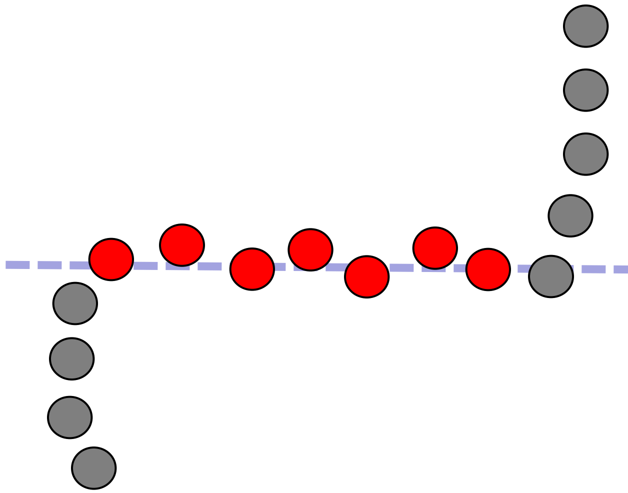
- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.

Region growing



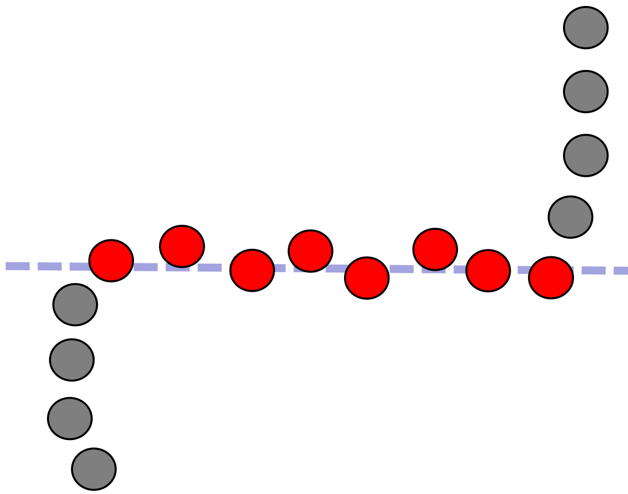
- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.

Region growing



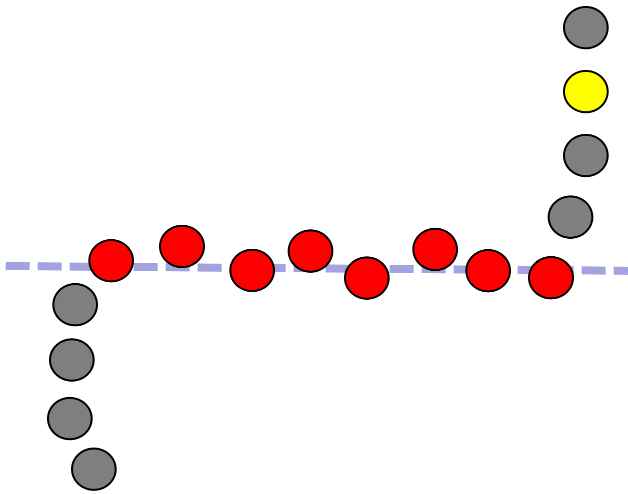
- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.

Region growing



- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.

Region growing



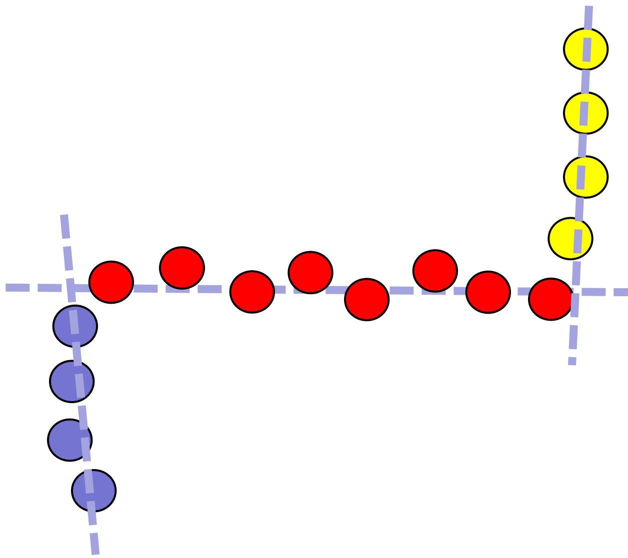
- select a point and a primitive hypothesis
- propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.
- select a remaining point and a primitive Hypothesis, and iterate

Region growing

▪ select a point and a primitive hypothesis

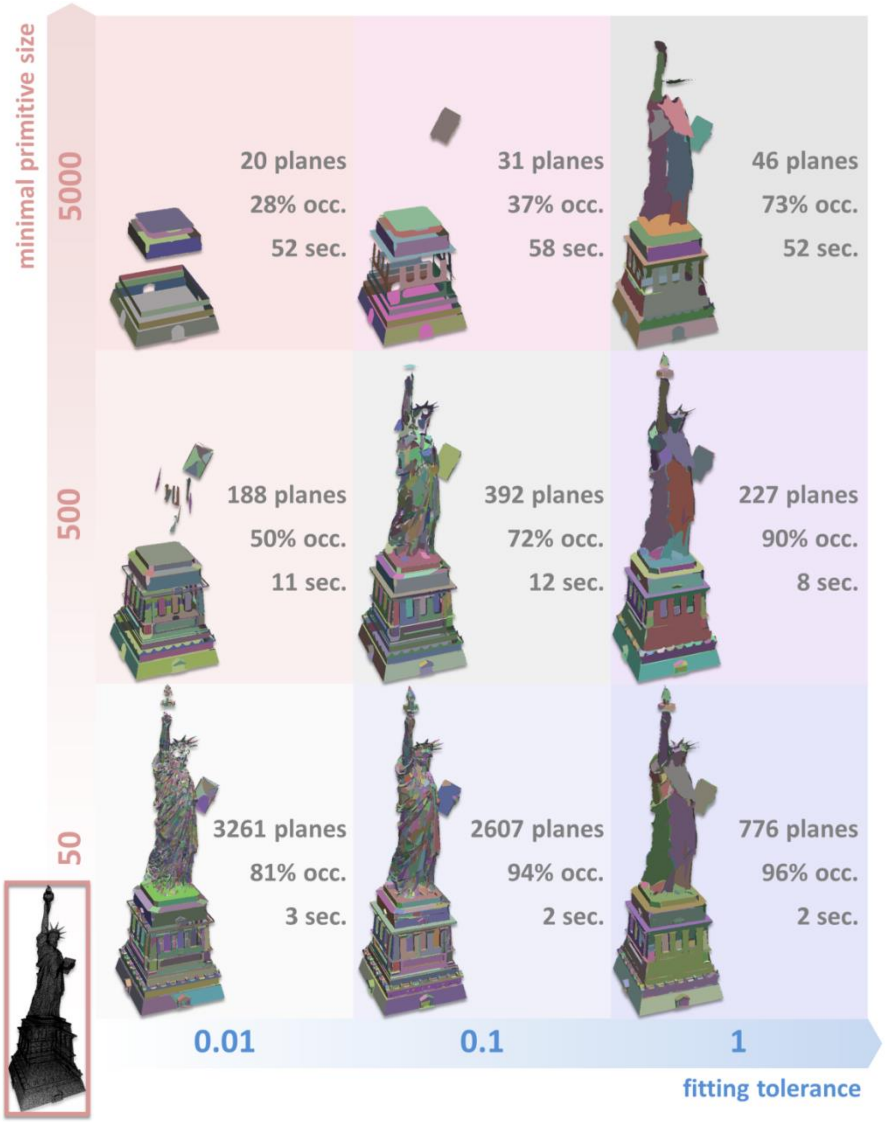
▪ propagate to the neighbors if they verify the hypothesis, and iterate until no point verifies the hypothesis anymore.

▪ select a remaining point and a primitive Hypothesis, and iterate



the parameters to specify

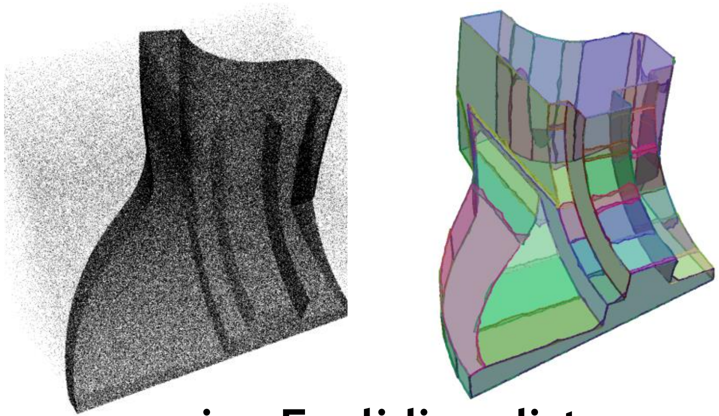
- minimum number of points needed to fit the primitive
- fitting tolerance



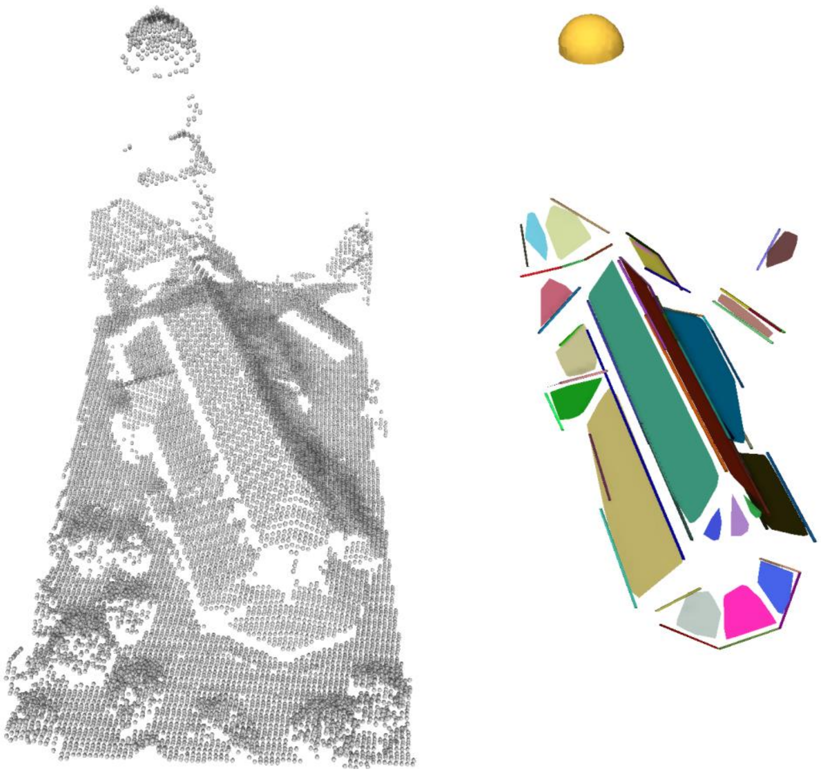
Region growing

- need to know the nearest neighbors
- the primitive hypothesis has to be relevant when starting the growing
- .. but the primitive hypothesis can also be updated during the growing
- not optimal when noisy data

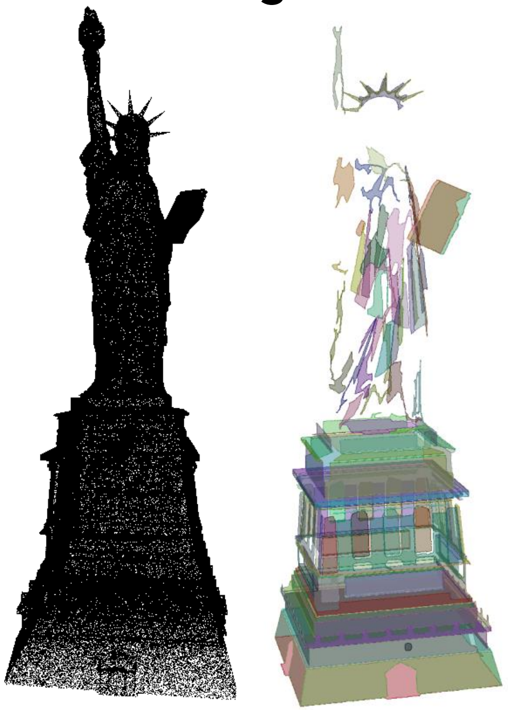
Region growing



using Euclidian distance



using normals



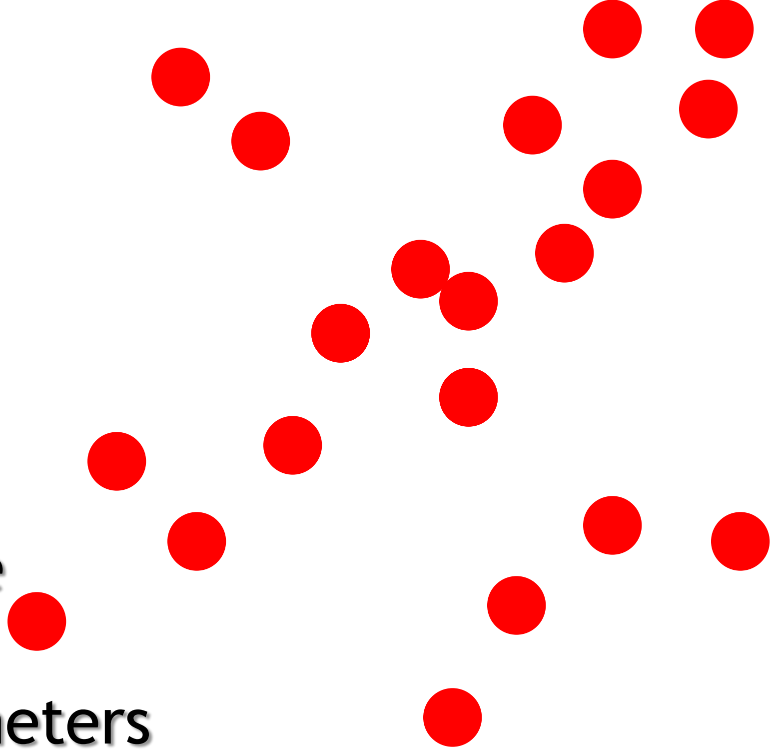
using normals and Euclidian distance

Ransac (RANdom SAmple Consensus)

- Iterative method
- Estimation of the primitive parameters by a random sampling of data
- Designed to be efficient with outlier-laden Data
- Non-deterministic

[Fischler & Bolles]

Ransac Algorithm



- Sample (randomly)
the number of points
required to fit the primitive

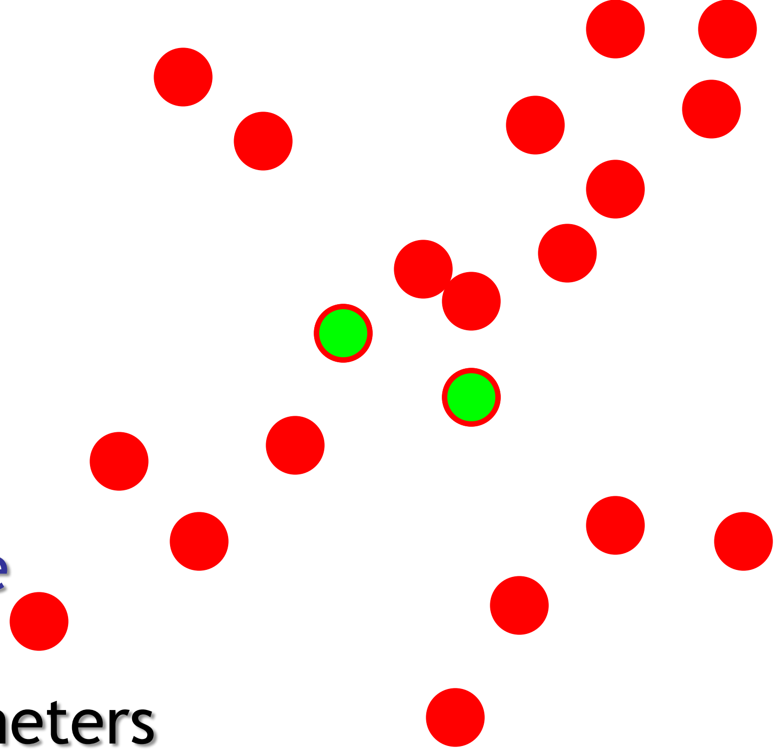
- Solve for primitive parameters
using samples

- Score by the fraction of inliers
within a preset threshold of the primitive

Repeat these 3 steps until the best primitive is found with high confidence

[Fischler & Bolles]

Ransac Algorithm



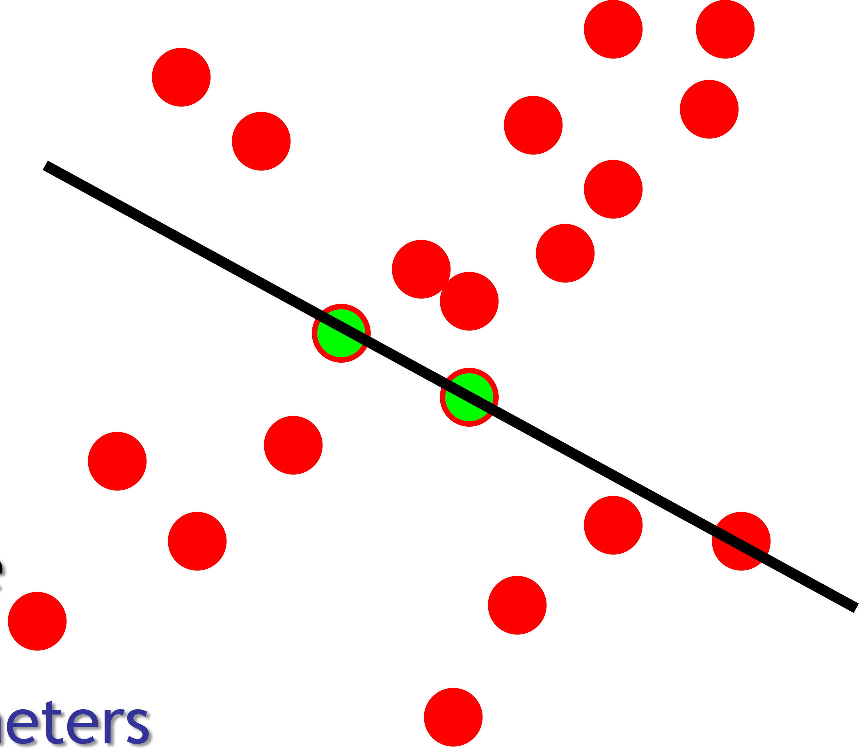
- Sample (randomly) the number of points required to fit the primitive

- Solve for primitive parameters using samples

- Score by the fraction of inliers within a preset threshold of the primitive

Repeat these 3 steps until the best primitive is found with high confidence

Ransac Algorithm



- Sample (randomly) the number of points required to fit the primitive

- Solve for primitive parameters using samples

- Score by the fraction of inliers within a preset threshold of the primitive

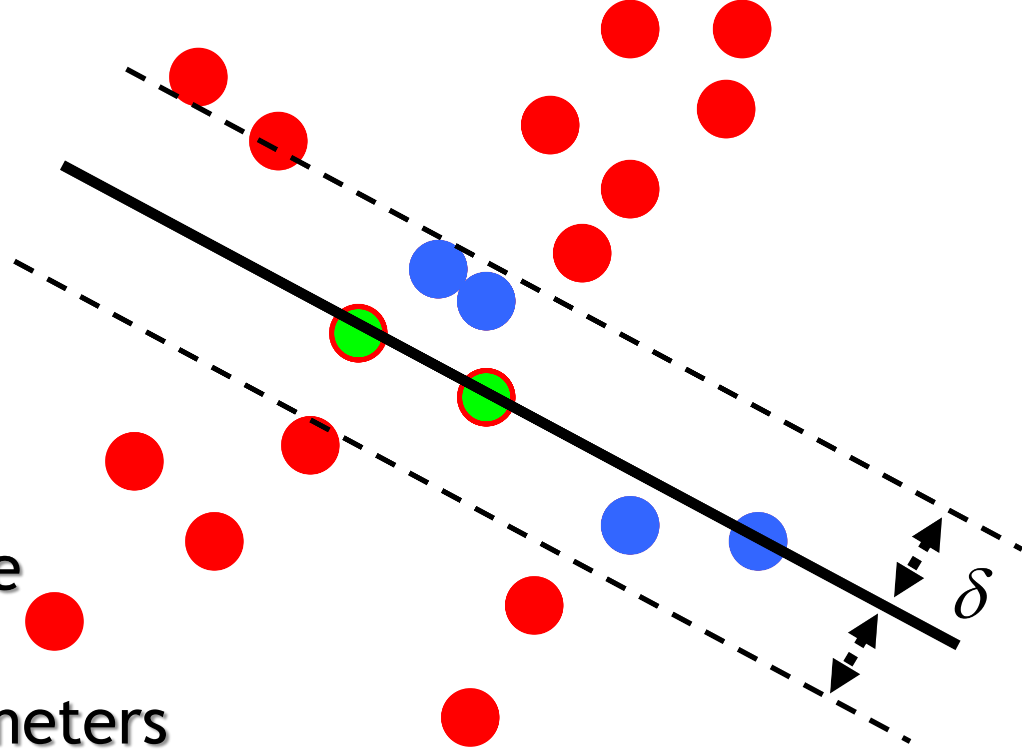
Repeat these 3 steps until the best primitive is found with high confidence

Ransac Algorithm

- Sample (randomly) the number of points required to fit the primitive

- Solve for primitive parameters using samples

- Score by the fraction of inliers within a preset threshold of the primitive

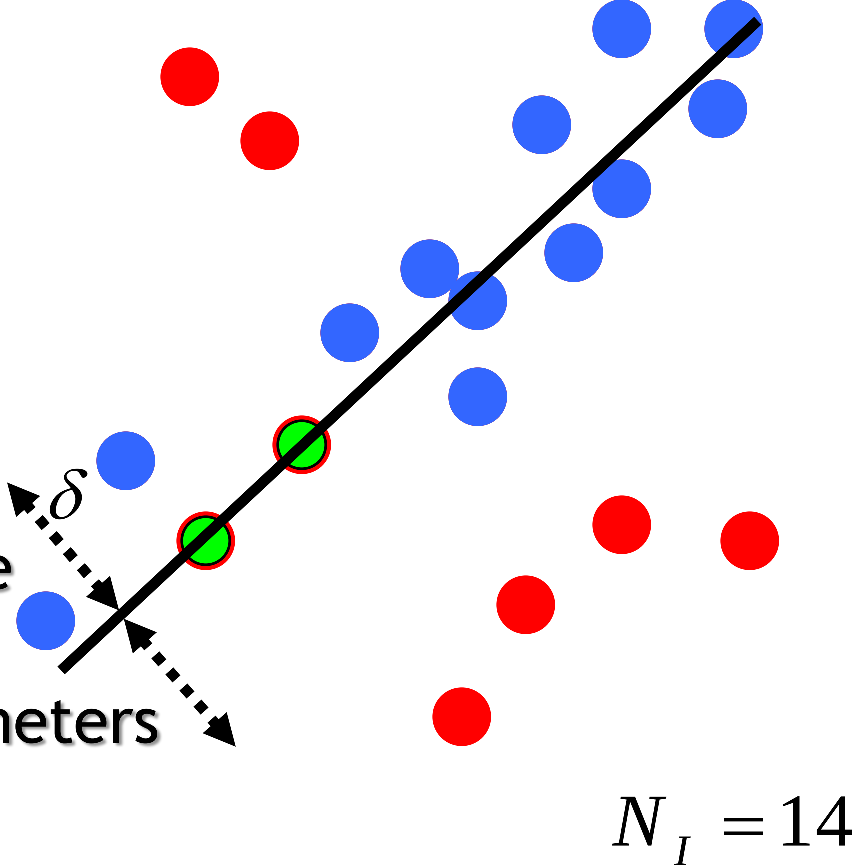


$$N_I = 6$$

Repeat these 3 steps until the best primitive is found with high confidence

Ransac Algorithm

- Sample (randomly) the number of points required to fit the primitive



- Solve for primitive parameters using samples

- Score by the fraction of inliers within a preset threshold of the primitive

Repeat these 3 steps until the best primitive is found with high confidence

the parameters to specify

- minimum number of points needed to fit the primitive
- Distance threshold δ
- Number of samples

To be chosen so that at least one random sample is free from outliers with a certain probability

Accumulation methods

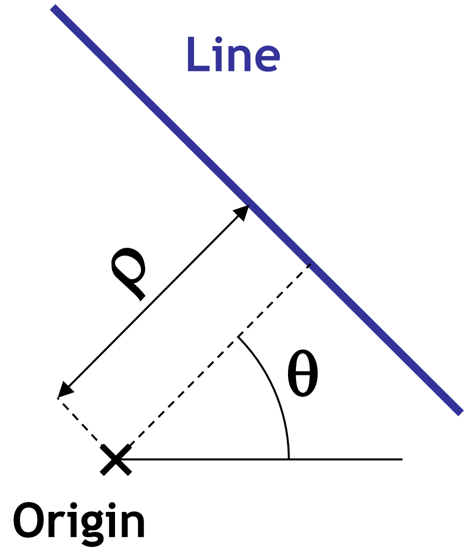
- Accumulate local primitive hypotheses in a space of primitive parameters

- extract the local maxima from the parameter space

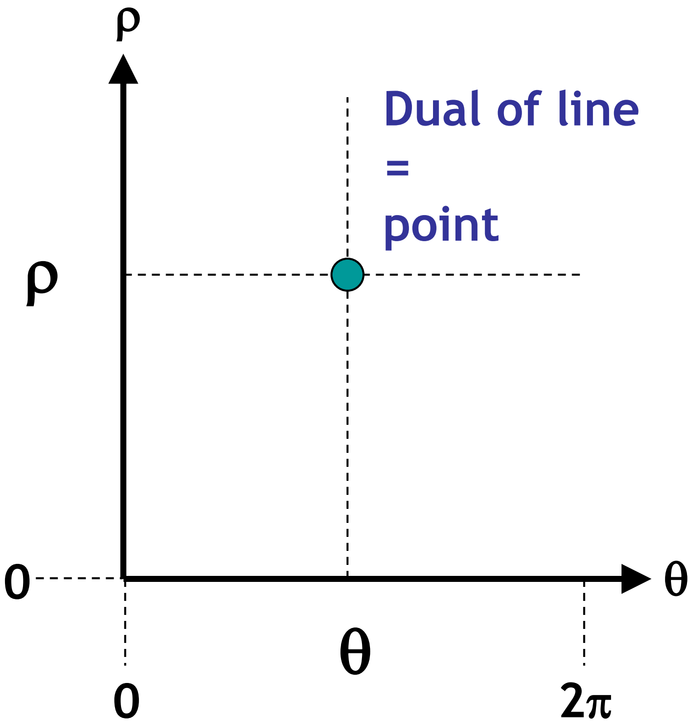
- the parameter space must be discretized

Accumulation methods: Hough transform

Case of lines in 2D

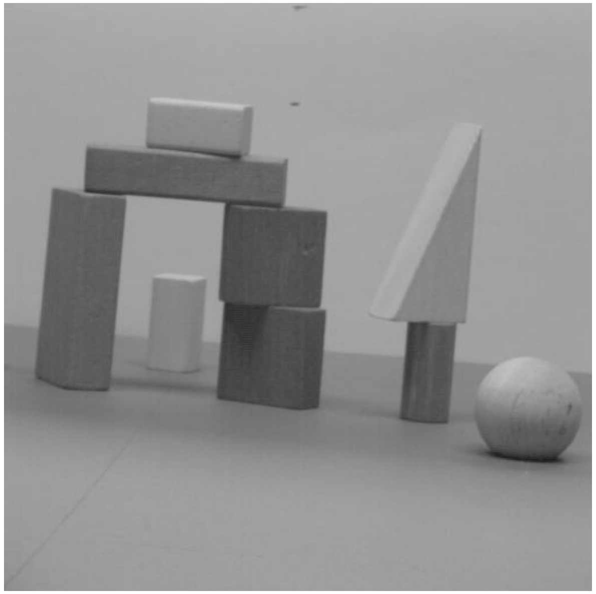


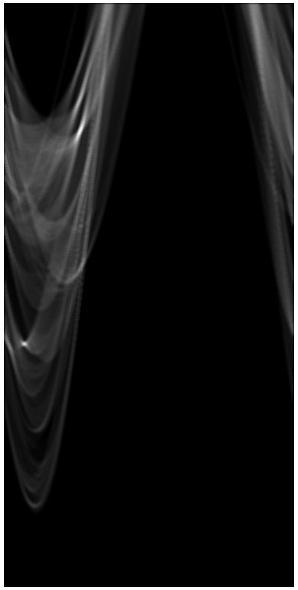
(x,y) space

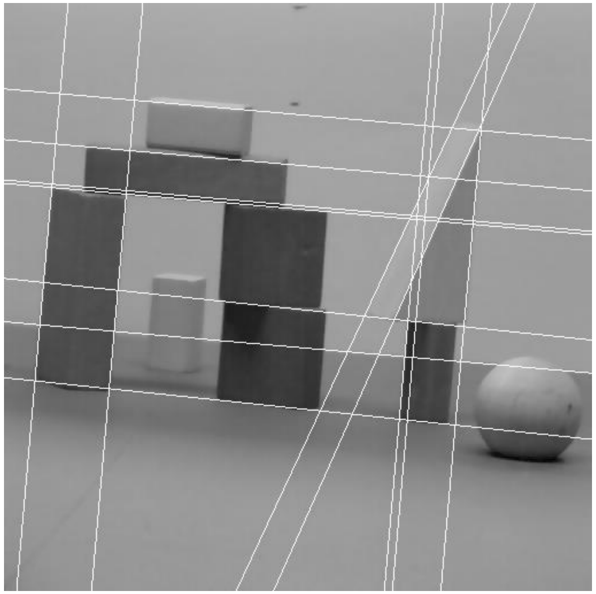


parameter space

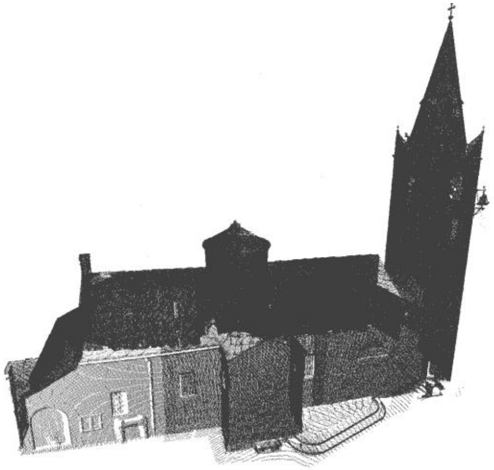
Accumulation methods: Hough transform



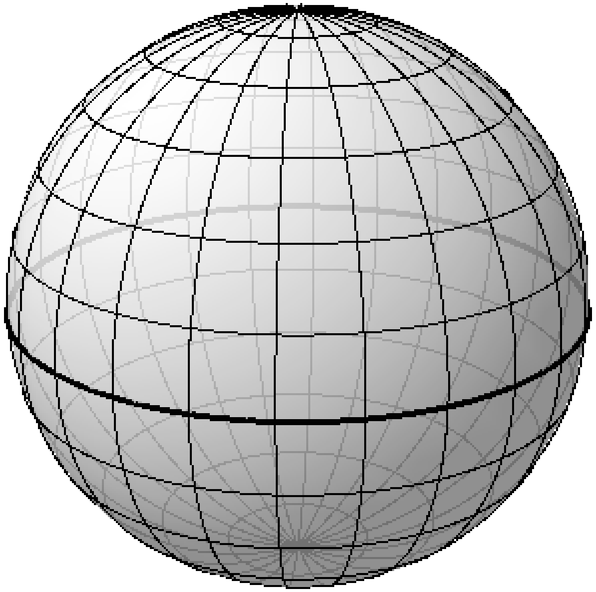




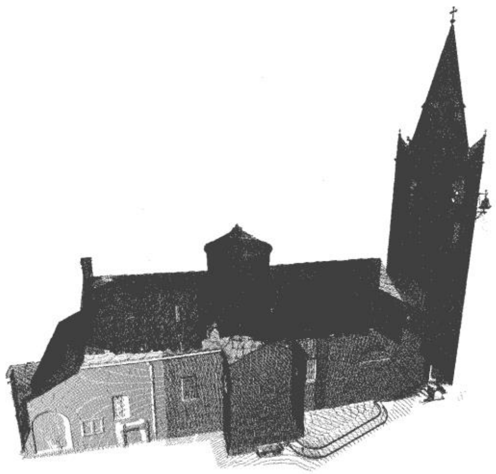
Accumulation methods: Gaussian sphere



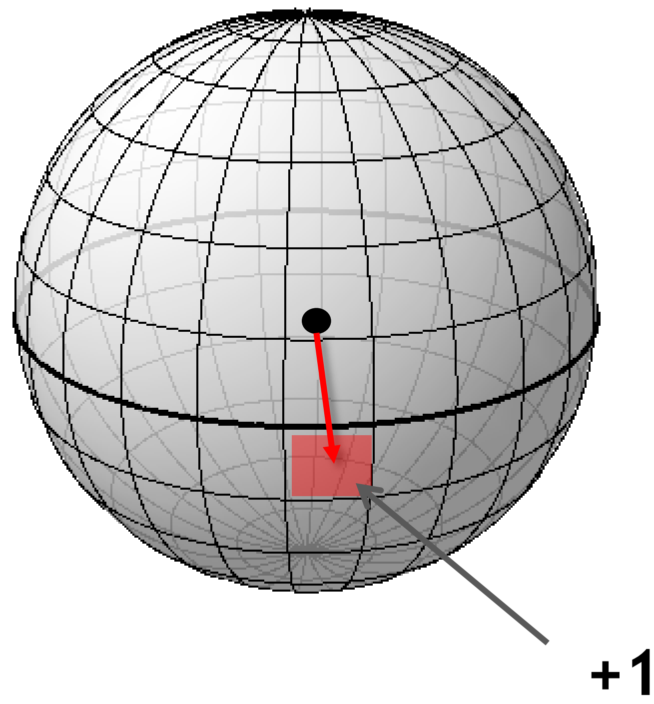
For each point of the data, we increment the sphere cell targeted by the point normal from the sphere center



Accumulation methods: Gaussian sphere



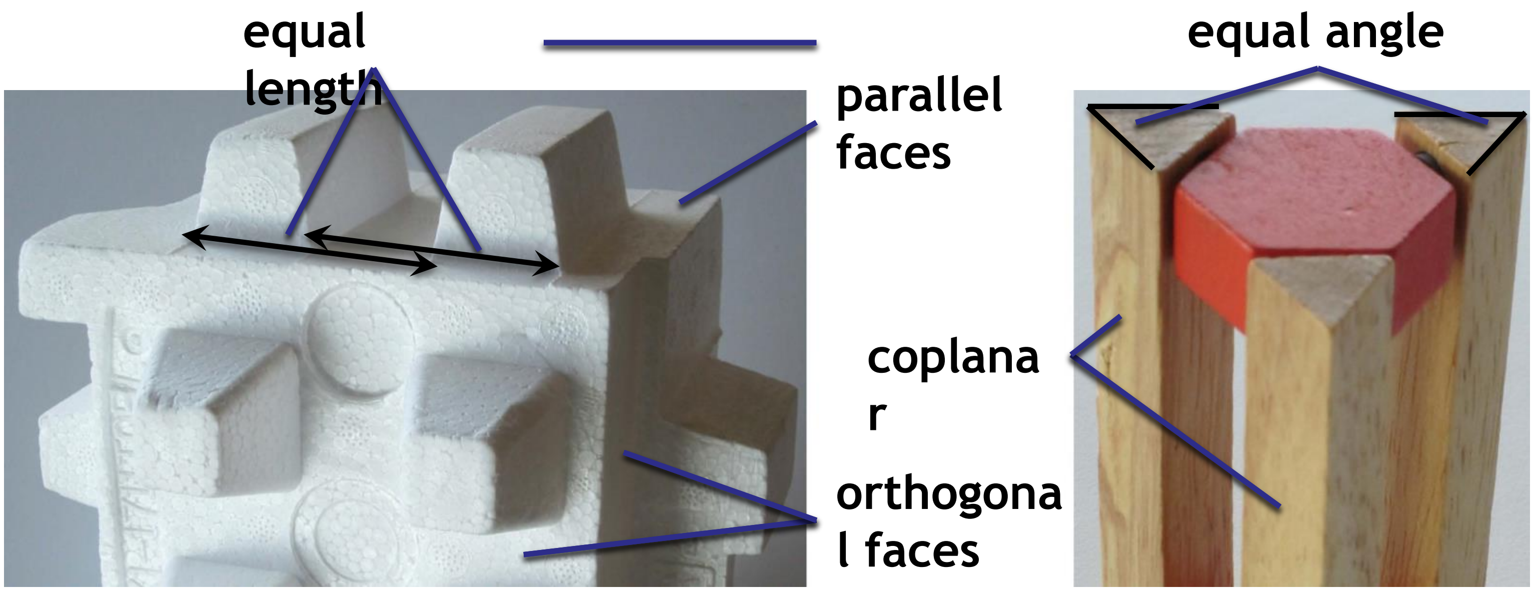
For each point of the data, we increment the sphere cell targeted by the point normal from the sphere center



Accumulation methods

- can be computationally expensive
- restricted to certain types of primitives
- can be interesting for “structuring” the primitive configuration with global regularities

Global regularity discovering

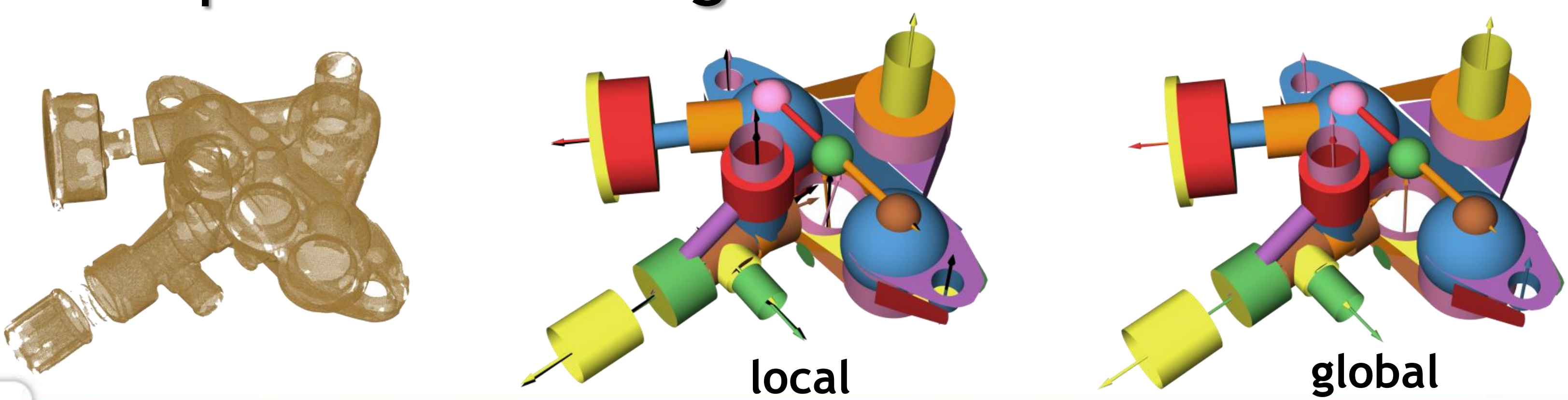


[Li et al., 2011]

Global regularity discovering

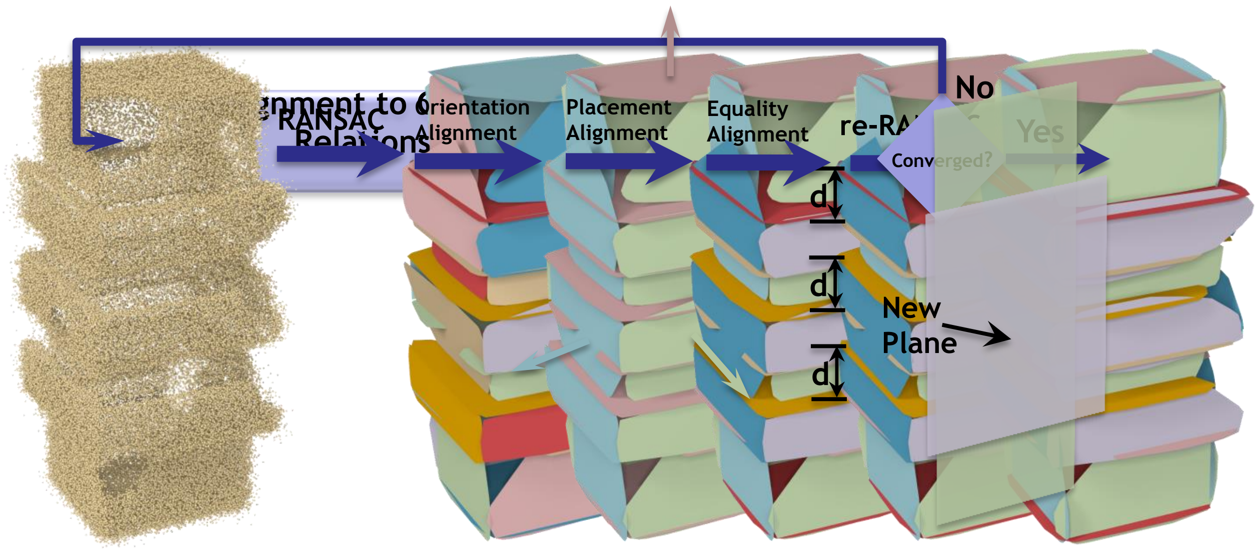
- usually primitives are detected locally, without interaction between each others

- It can be usefull to introduce interactions between primitives at a global scale



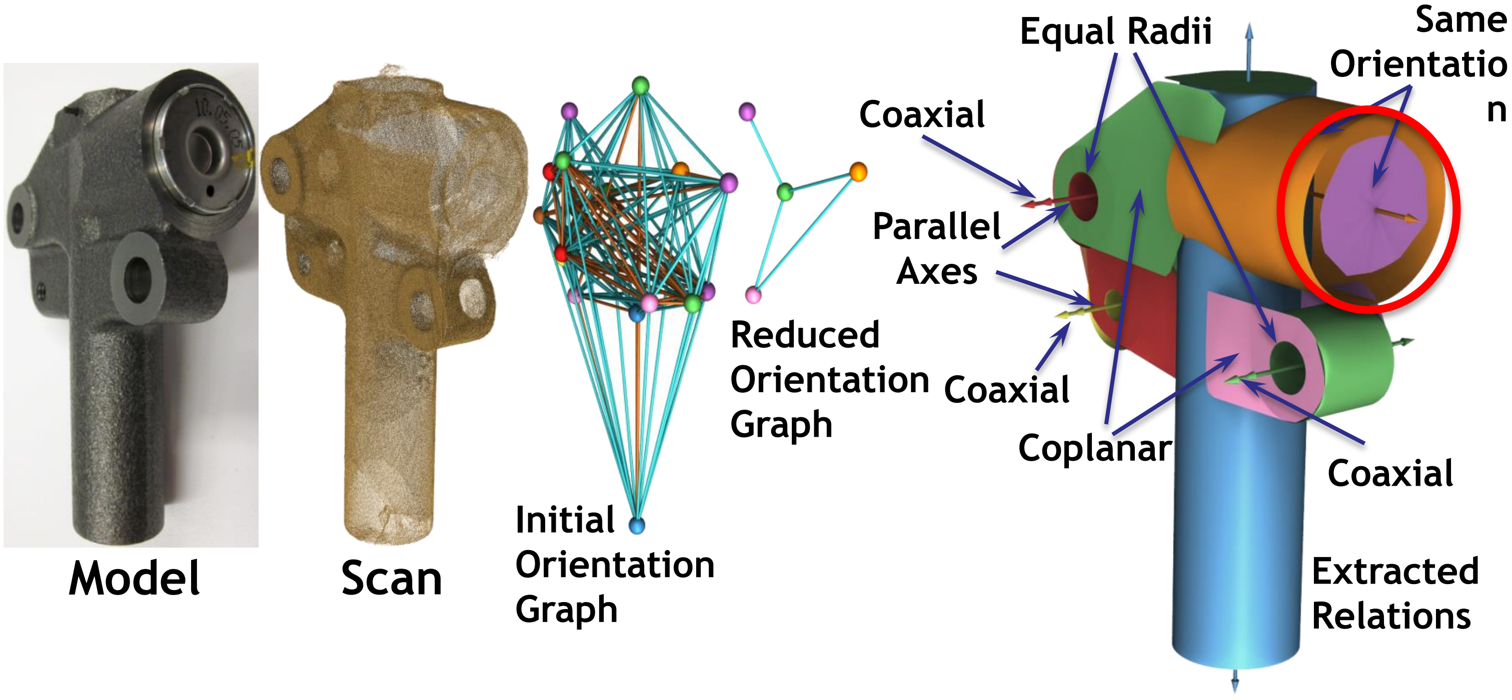
[Li et al., 2011]

Global regularity discovering [Globfit]



[Li et al., 2011]

Global regularity discovering [Globfit]



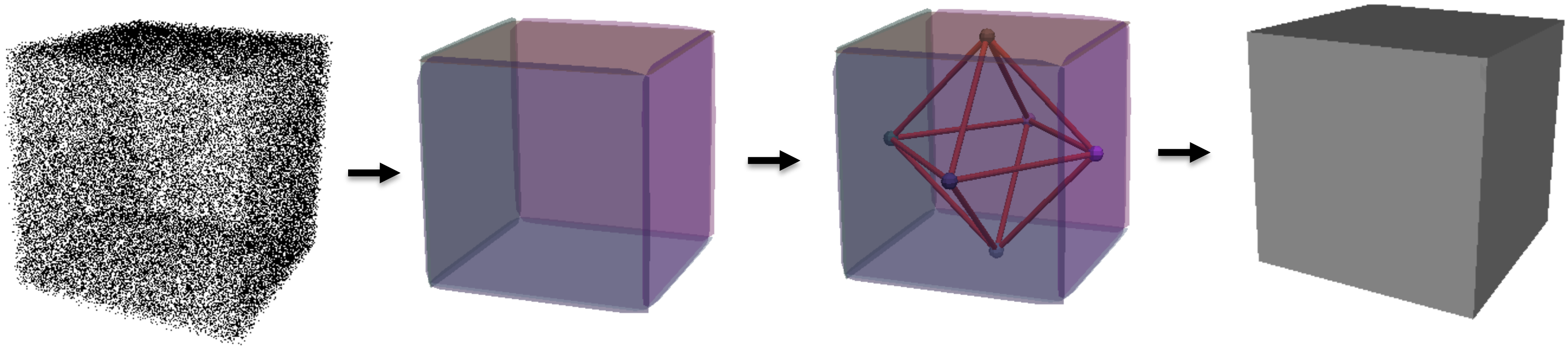
[Li et al., 2011]

- Geometric primitive extraction
- Surface reconstruction using geometric primitives
  - Graph-based
  - Space partitioning
  - Hybrid reconstruction
- Two words on template matching

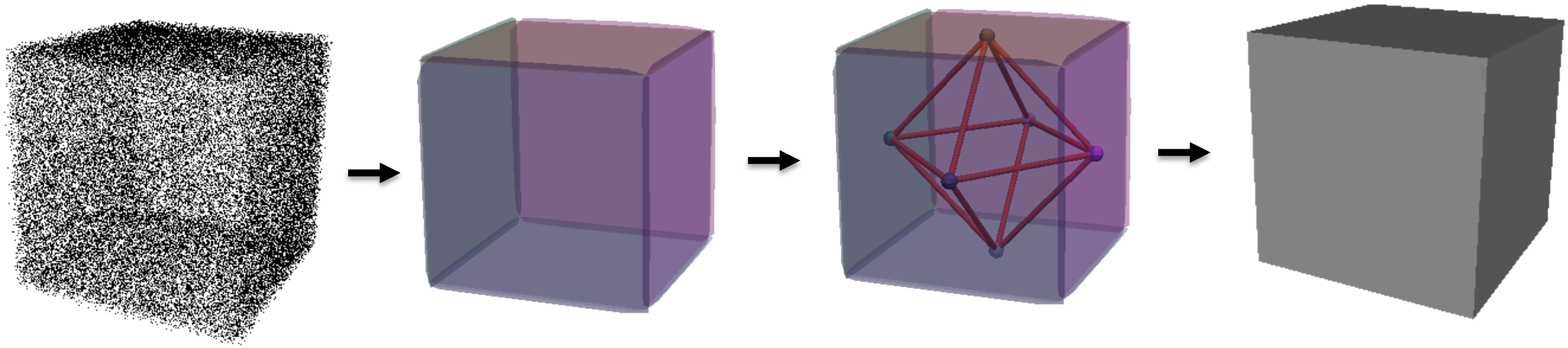
Surface reconstruction from geometric primitives

Q: What can we do once we have extracted the primitives ?

A1: compute the primitive adjacency graph, and reconstruct the surface as the dual of this graph.

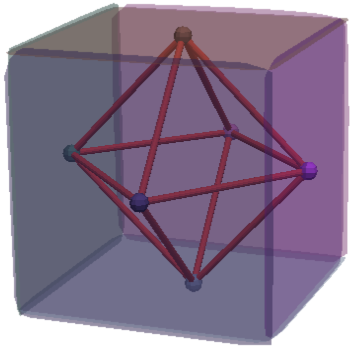


A1: compute the primitive adjacency graph, and reconstruct the surface as the dual of this graph.



If you are lucky..

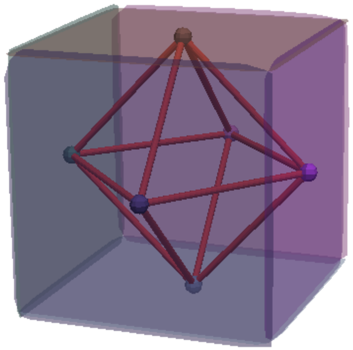
A1: compute the primitive adjacency graph, and reconstruct the surface as the dual of this graph.



Ideal case: this never happens in practice

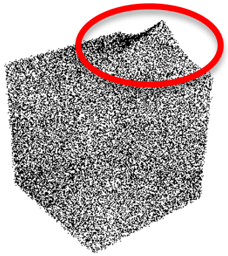
- No guarantee of finding the right primitive configuration and right adjacency graph

A1: compute the primitive adjacency graph, and reconstruct the surface as the dual of this graph.



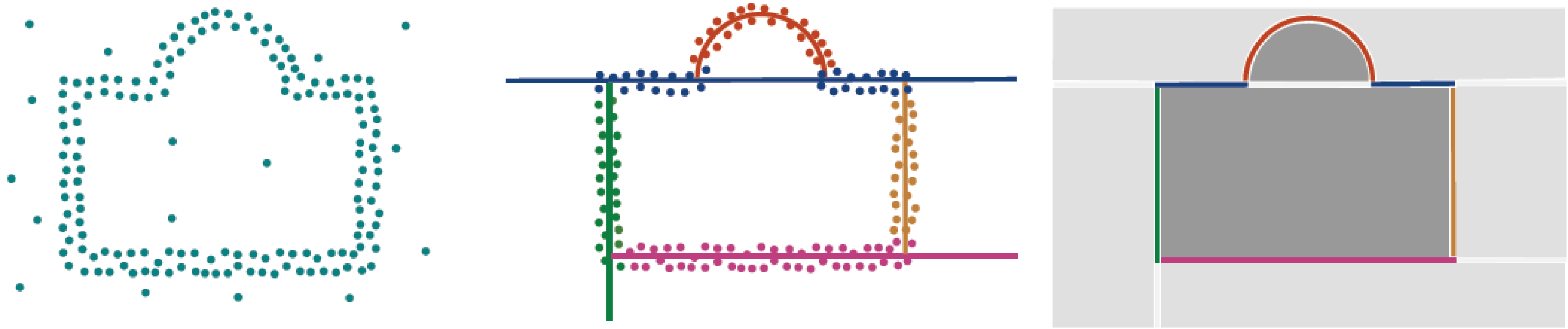
Ideal case: this never happens in practice

- No guarantee of finding the right primitive configuration and right adjacency graph



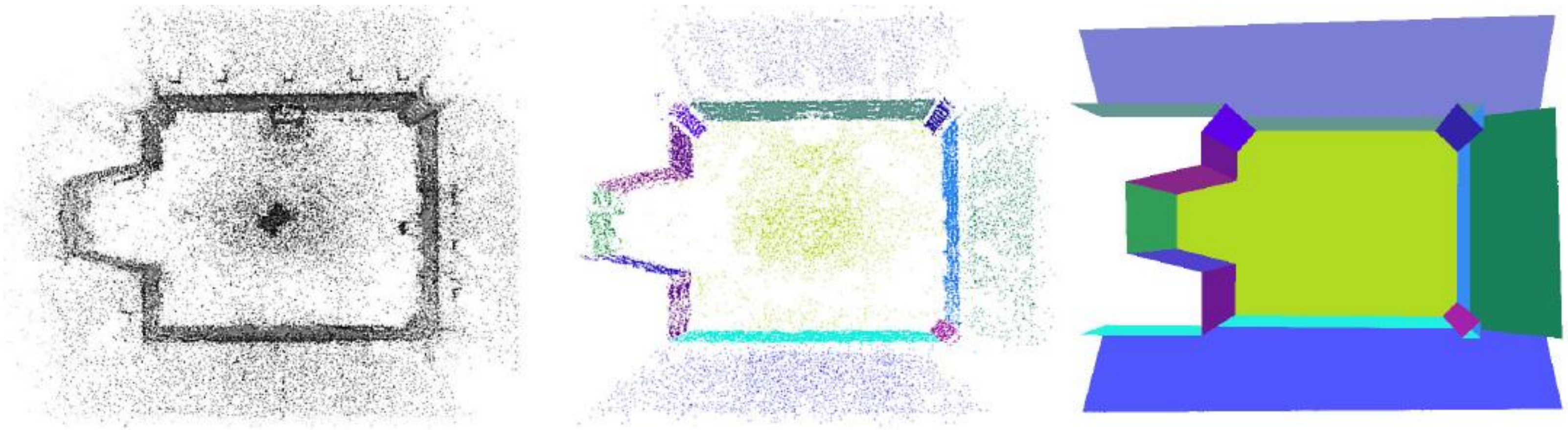
- No guarantee that the observed scene can be entirely explained by geometric primitives

- A2: Use primitives to partition the space into cells to be labeled as inside or outside



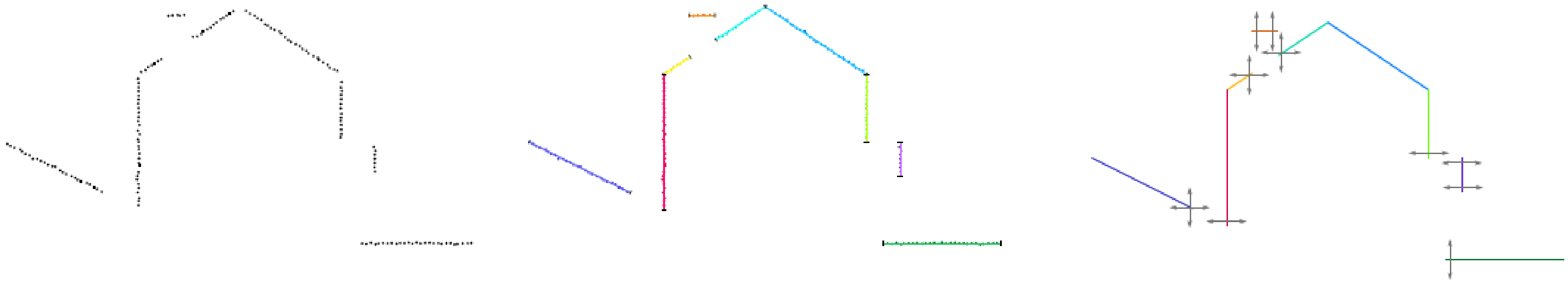
[Labatut et al., 2009]

- works well when no missing primitive

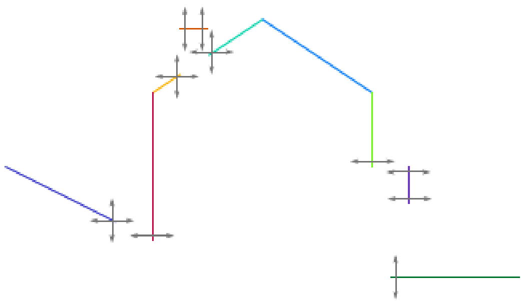


[Labatut et al., 2009]

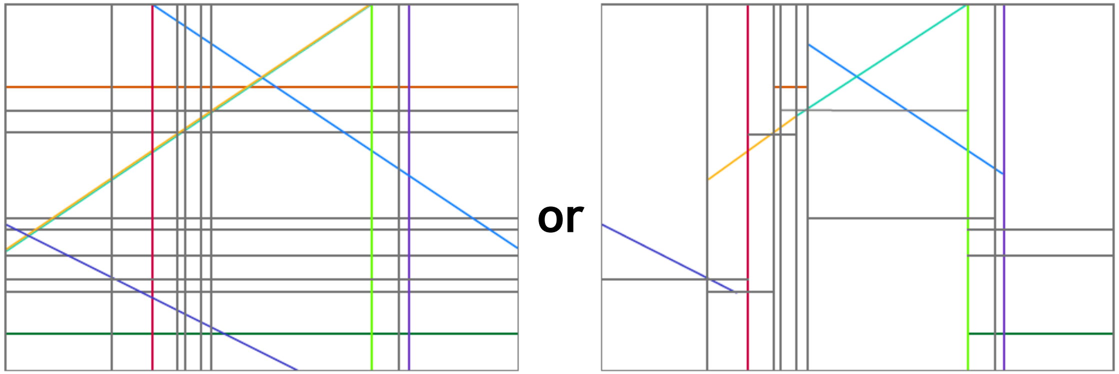
- when primitives are missed or cannot be detected, use of ghost primitives



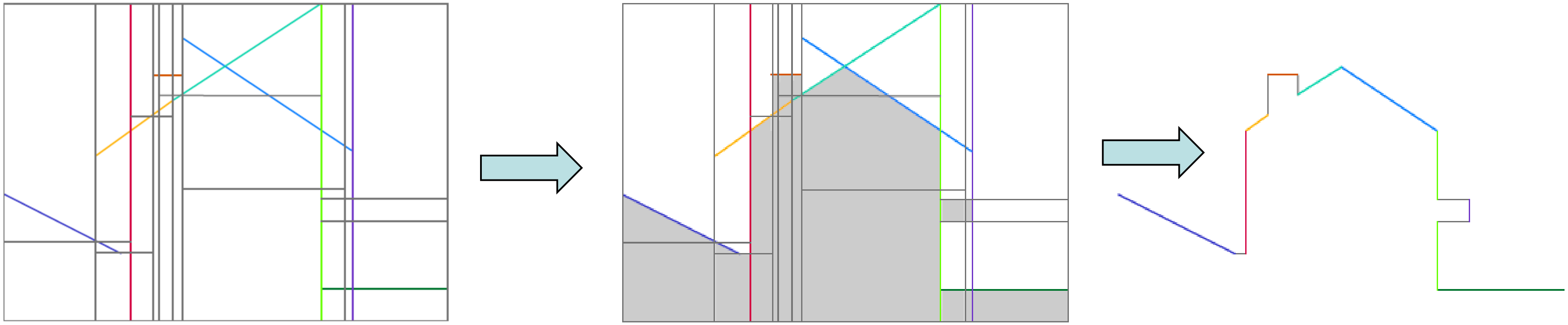
- when primitives are missed or cannot be detected, use of ghost primitives



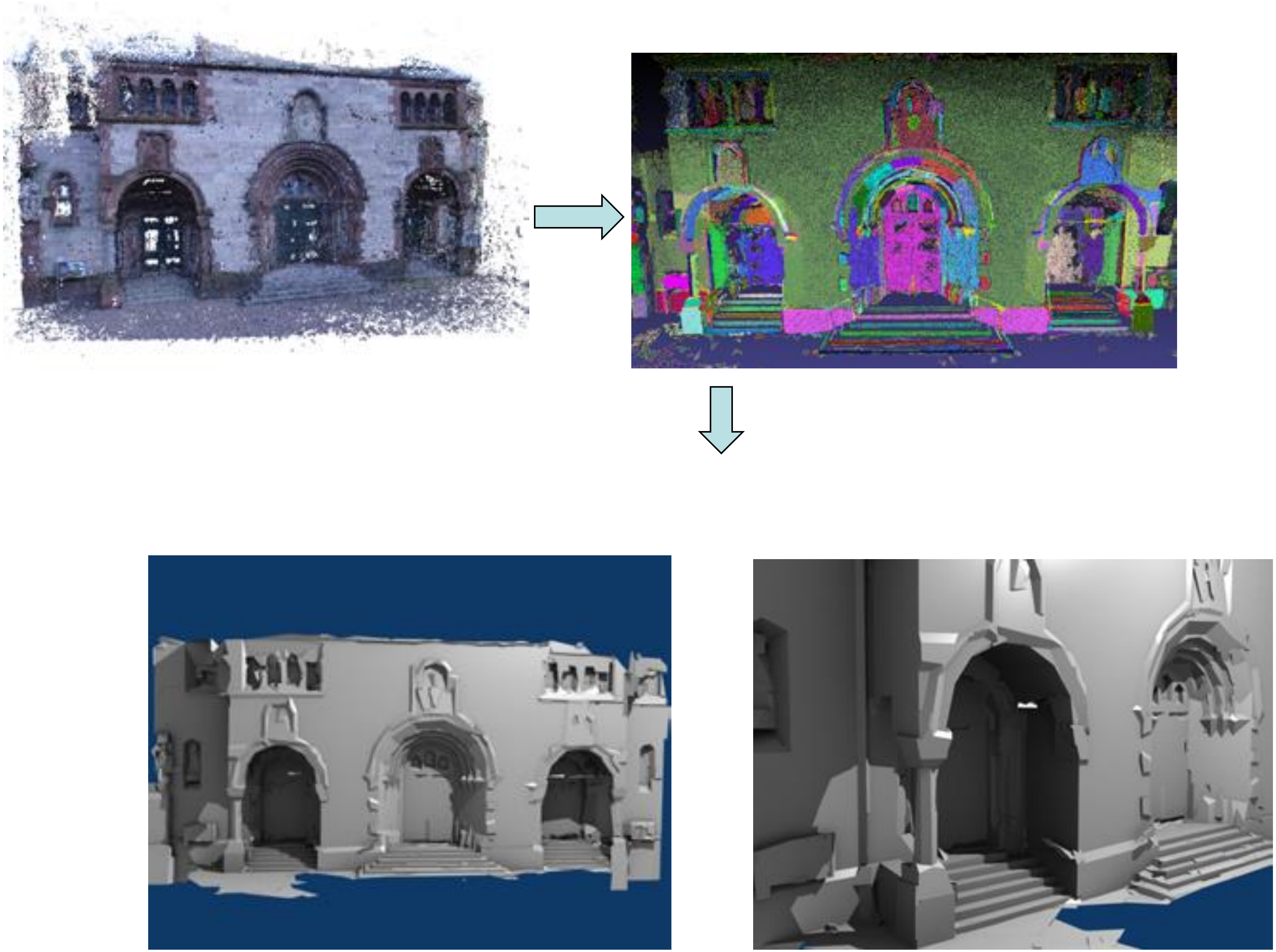
or



- when primitives are missed or cannot be detected, use of ghost primitives

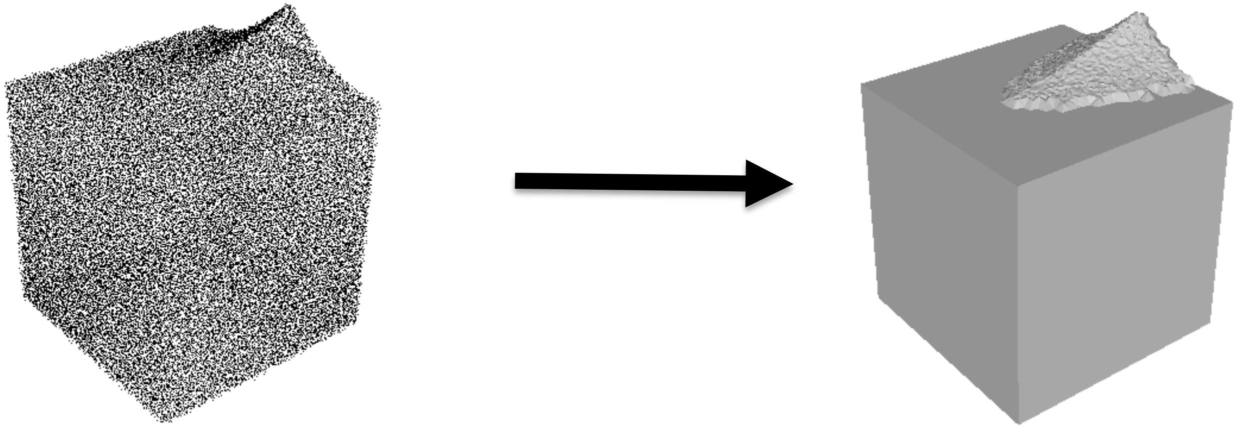


[Chauve et al., 2010]



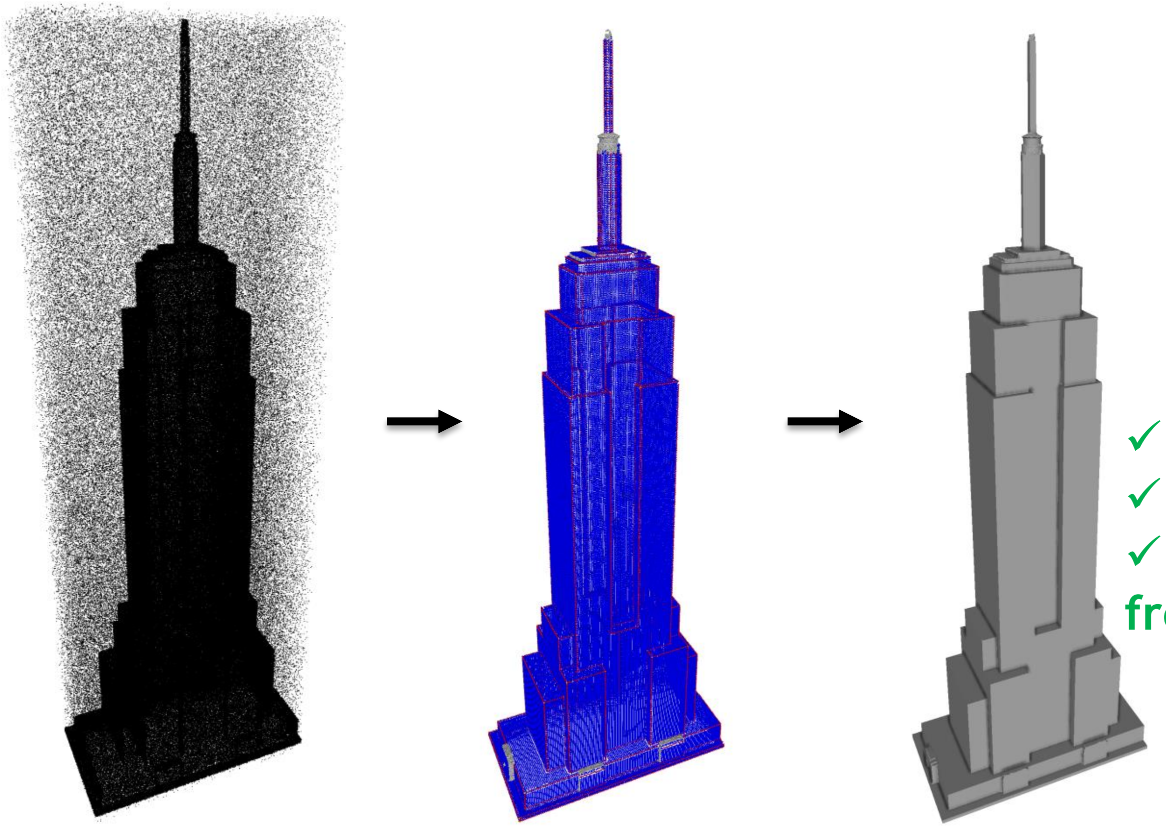
[Chauve et al., 2010]

- A3: reconstruct an hybrid surface as a combination of canonical parts idealizing the primitives and free-form parts representing the smooth or undetected canonical elements



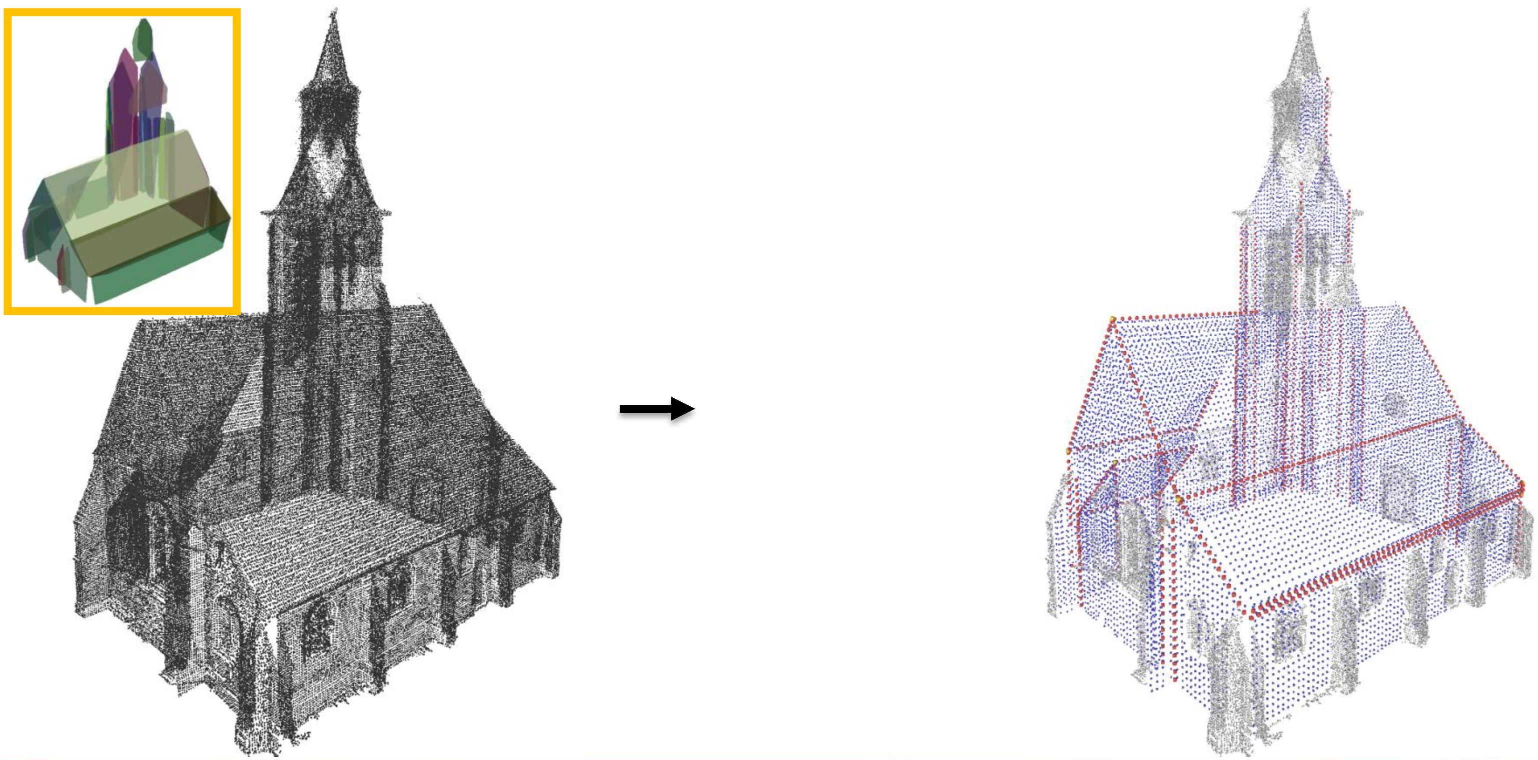
Hybrid reconstruction by structuring

Starting from a point set and a configuration of planar primitives extracted under a tolerance ε

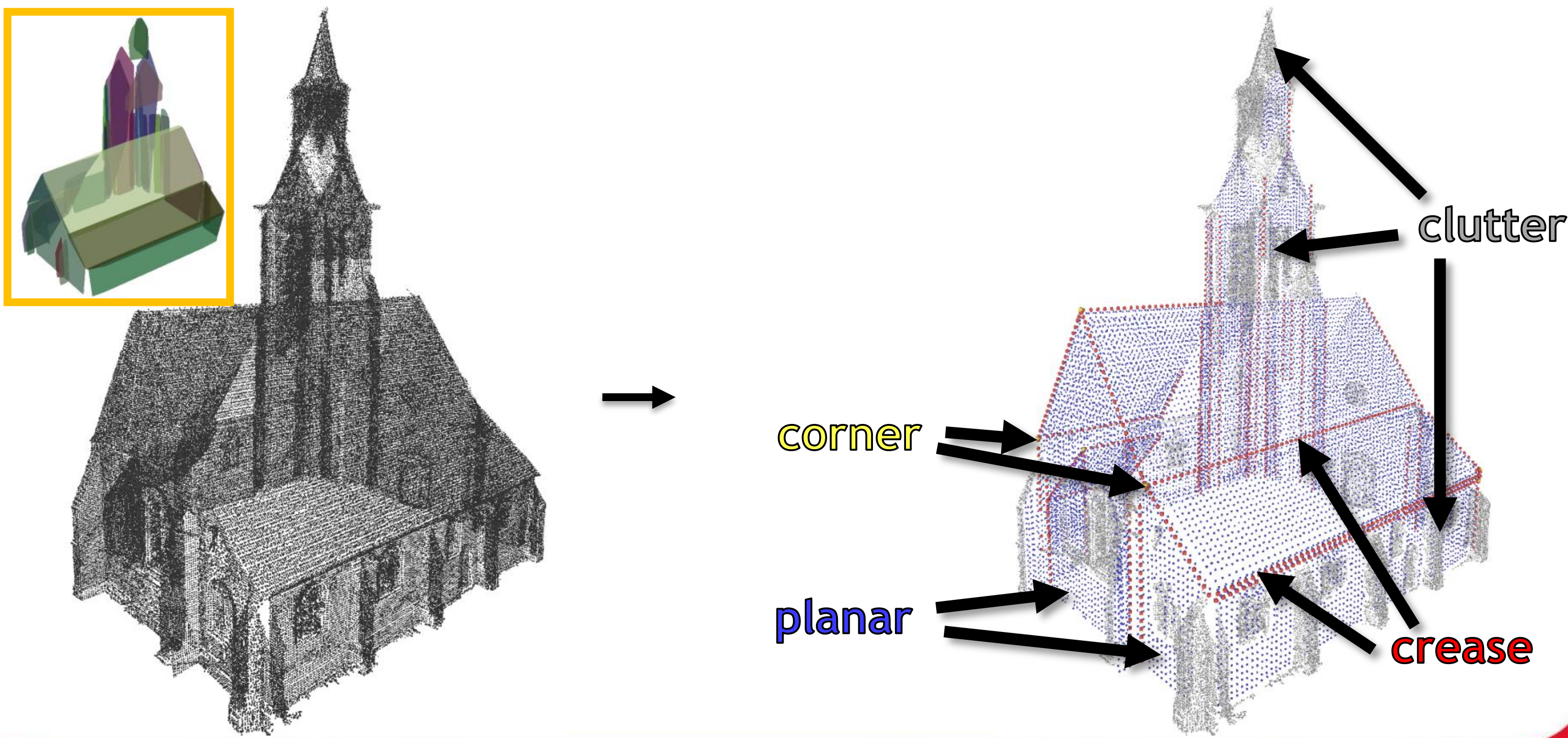


- ✓ 2-manifold
- ✓ watertight
- ✓ intersection-free

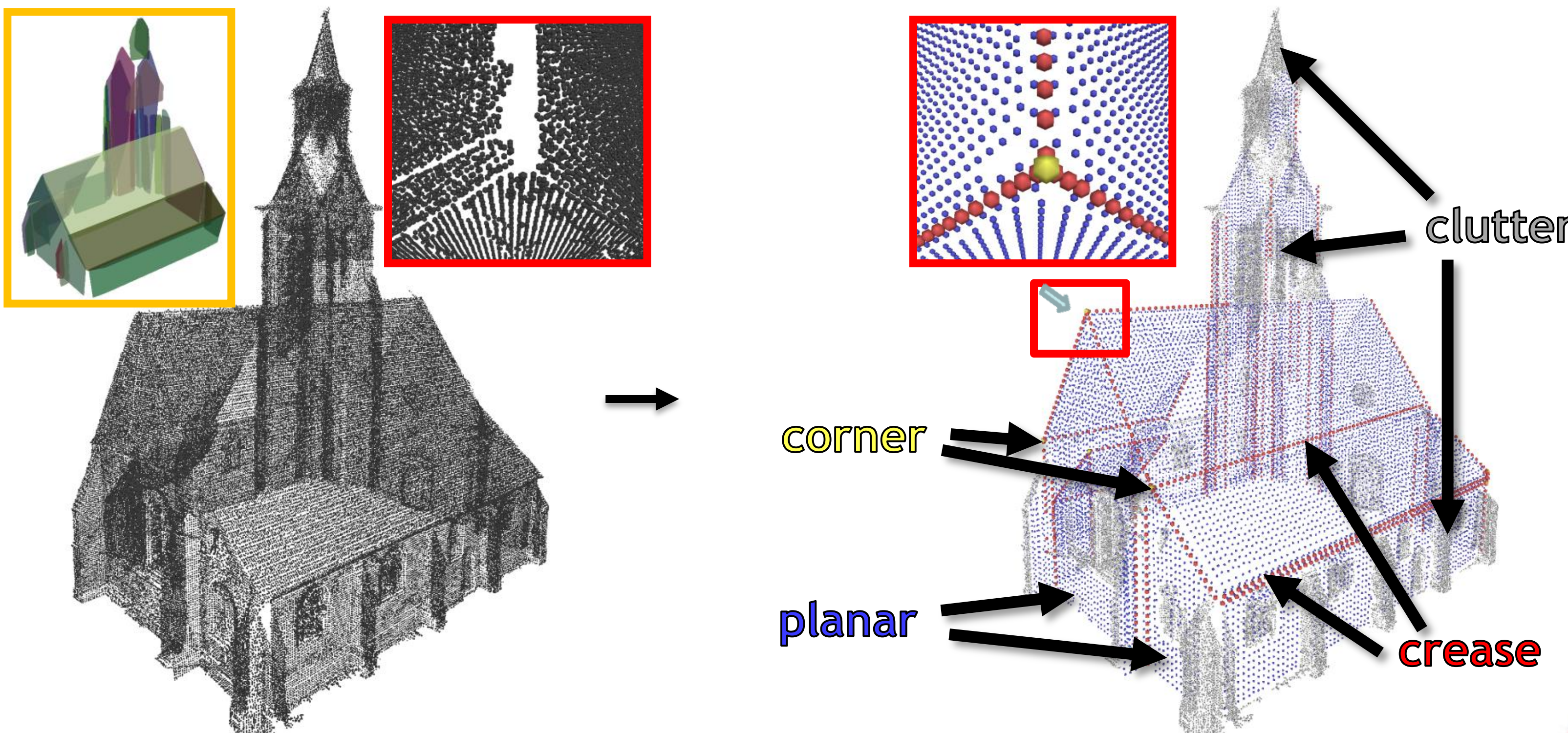
- **3 ideas**



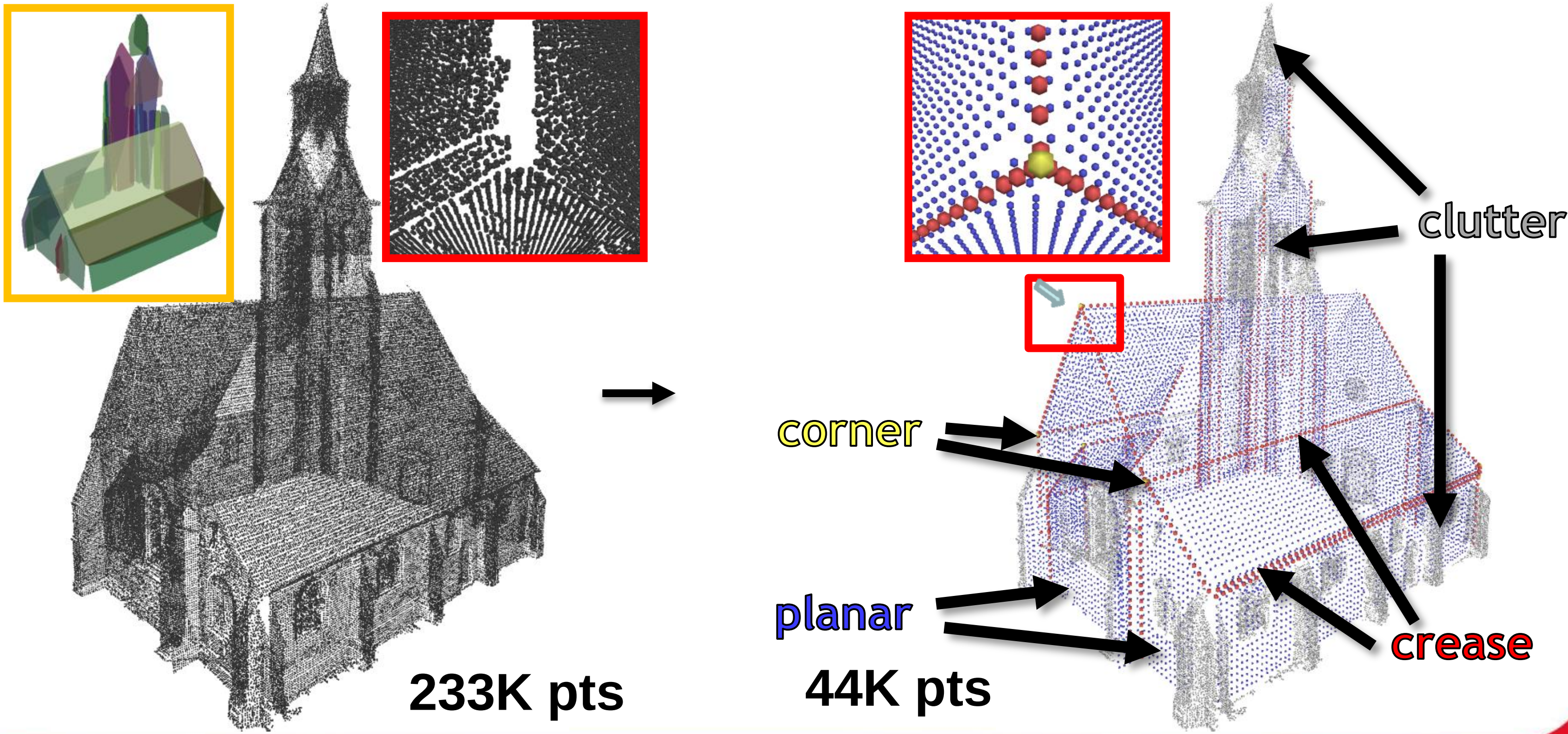
- **3 ideas**
  - Meaning insertion



- **3 ideas**
  - Meaning insertion
  - Structure idealization under Delaunay triangulation

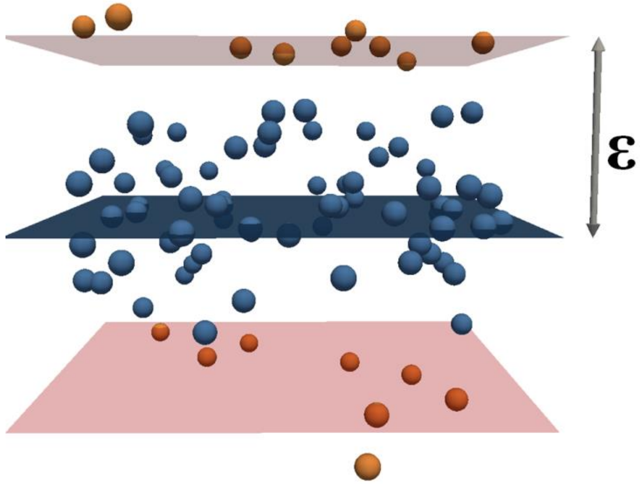


- **3 ideas**
  - Meaning insertion
  - Structure idealization under Delaunay triangulation
  - Complexity reduction



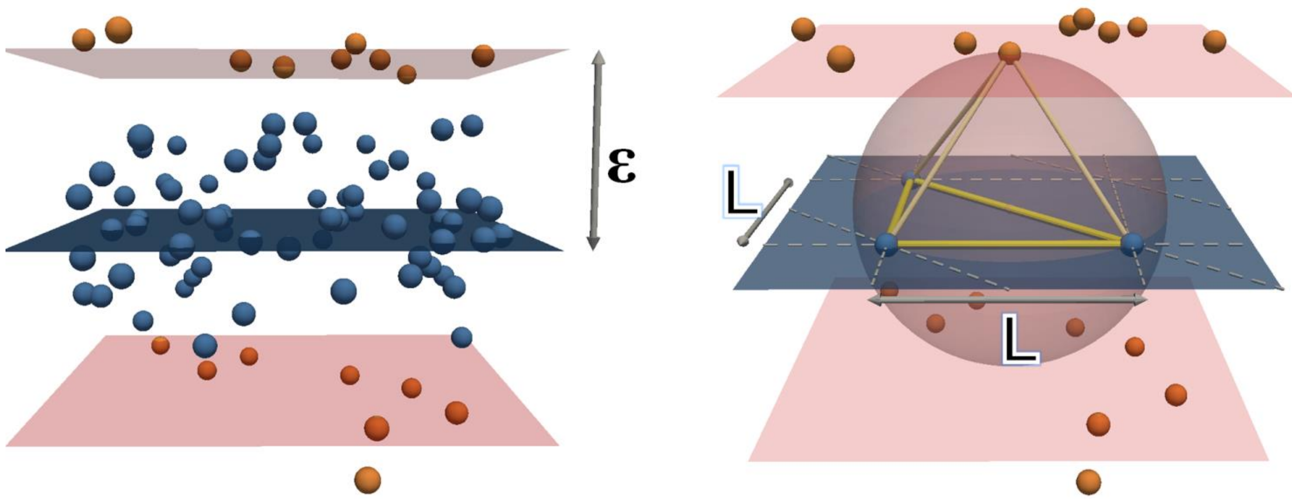
Replacement of the inliers by an *ideal* layout of planar points

- Occupancy 2D-grid projected on the planar primitive



Replacement of the inliers by an *ideal* layout of planar points

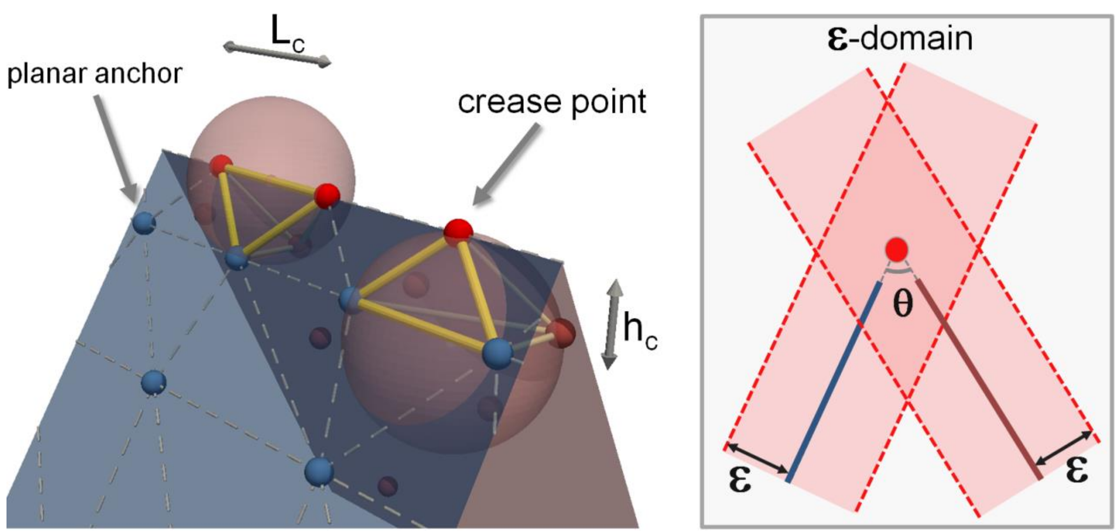
- Occupancy 2D-grid projected on the planar primitive
- Facet existence condition in Delaunay: $\boxed{L_p < \sqrt{2}\,\varepsilon}$



Preservation of edges between adjacent primitives

- Occupancy 1D-grid projected on the intersection line
- Facet existence condition in Delaunay:

$$\begin{cases} L_c = 2\varepsilon \\ h_c = \varepsilon \times \cos\frac{\theta}{2} \end{cases}$$

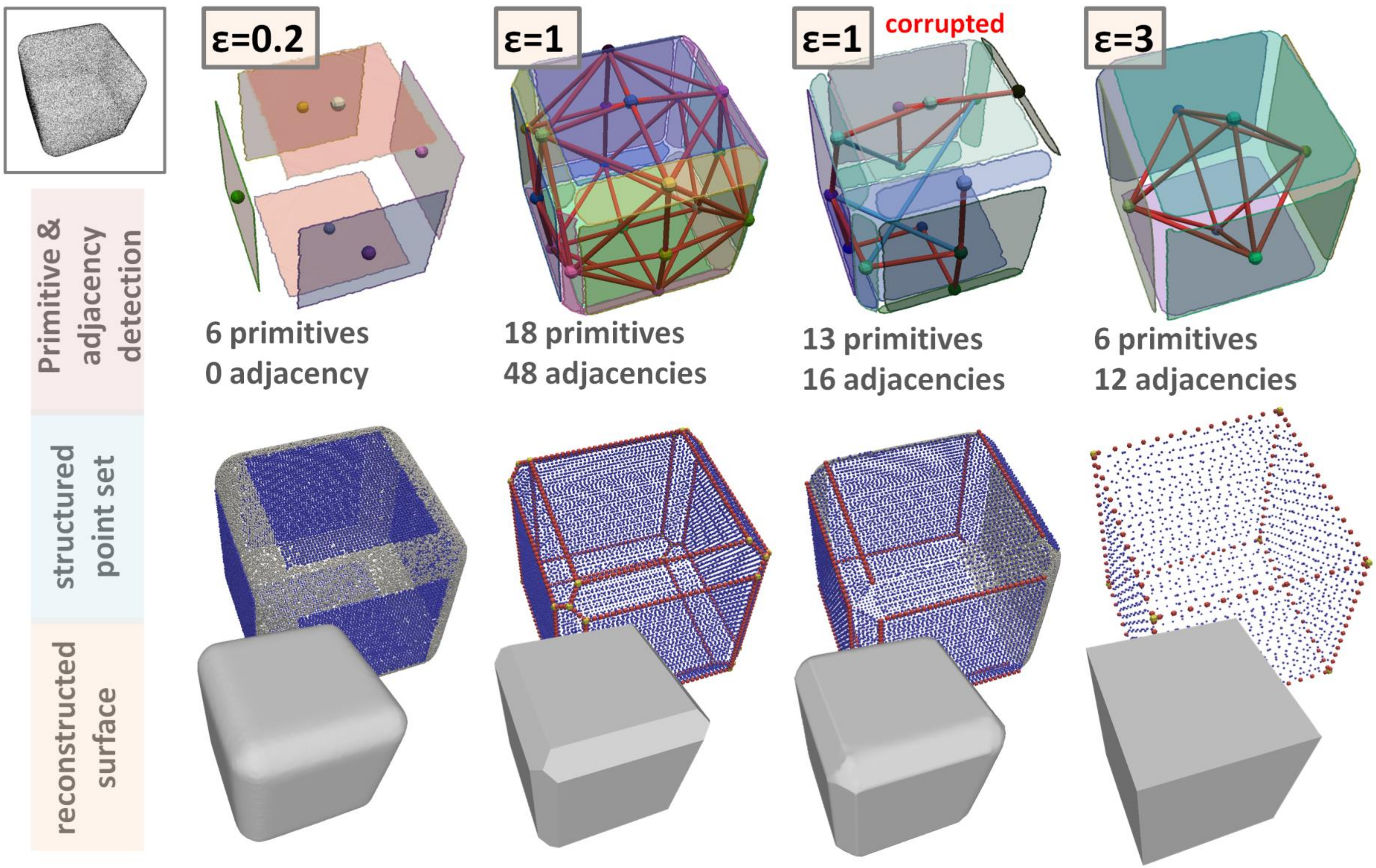


- ***Corner* points**

added by detecting the potential n-cycles extracted from the detected 3-cycles from the primitive

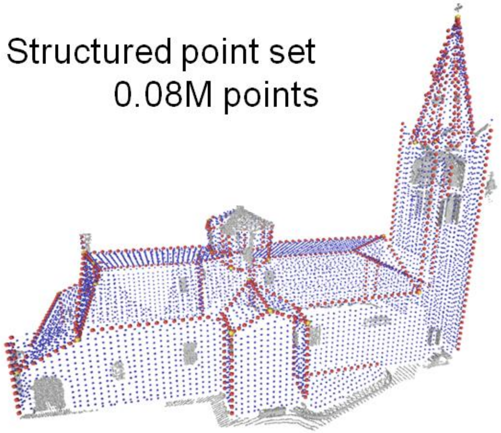
- ***Clutter* points**

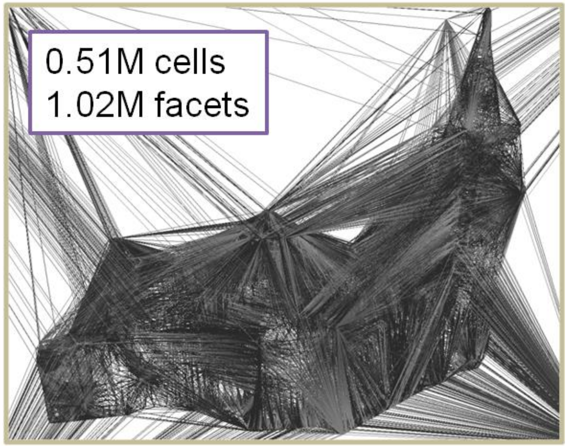
correspond to the input points which have not been detected as belonging to planar primitives


ε=0.2
ε=1
ε=1
corrupted
ε=3
Primitive & adjacency detection
structured point set
reconstructed surface
6 primitives
0 adjacency
18 primitives
48 adjacencies
13 primitives
16 adjacencies
6 primitives
12 adjacencies

Space partitioning: 3D delaunay triangulation from the structured point set

- tetrahedra do not intersect the primitive-induced surfaces
- each vertex of the triangulation inherits from a structural type





Labeling the Delaunay cells

- a graph $(\mathcal{C},\mathcal{F})$

  $\mathcal{C} = \{c_1,\ldots,c_n\}$ the set of Delaunay cells

  $\mathcal{F} = \{f_1,\ldots,f_m\}$ the set of triangular facets separating two cells

- a cut ($\mathcal{C}_{in}$,$\mathcal{C}_{out}$) in the graph

  The set of facets separating $\mathcal{C}_{in}$ from $\mathcal{C}_{out}$ forms a surface $\mathcal{S}$

- a cost function C measuring the quality of a cut

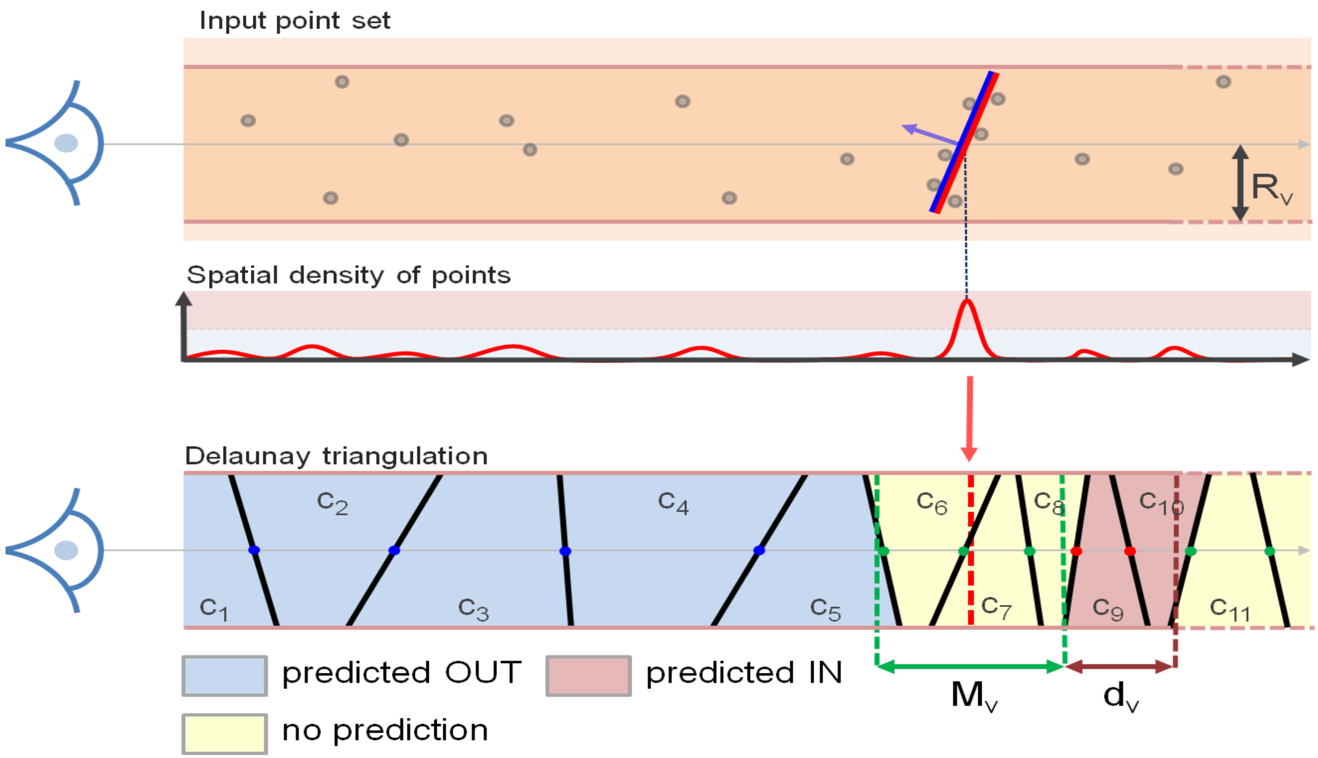
$$C(\mathcal{S}) = \underbrace{\sum_{f_i \in \mathcal{S}} a(f_i)\, Q(f_i)}_{\text{Geometric quality}} + \underbrace{\sum_{c_k \in \mathcal{C}_{in}} P_{out}(c_k) + \sum_{c_k \in \mathcal{C}_{out}} P_{in}(c_k)}_{\text{Visibility prediction}}$$

- an optimization algorithm for finding the optimal cut [Boykov2004]

Visibility prediction

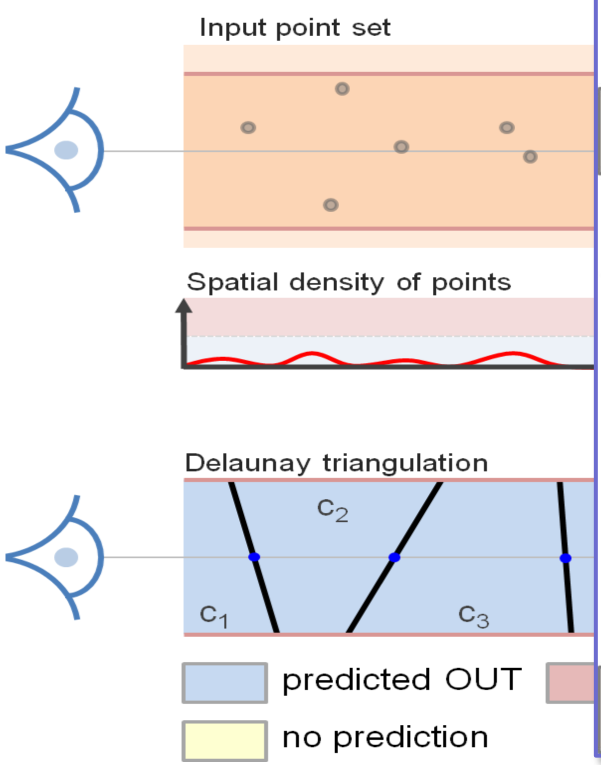
- detection of visibility patches by ray shooting
- *inside/outside* prediction of Delaunay cells crossed by a ray

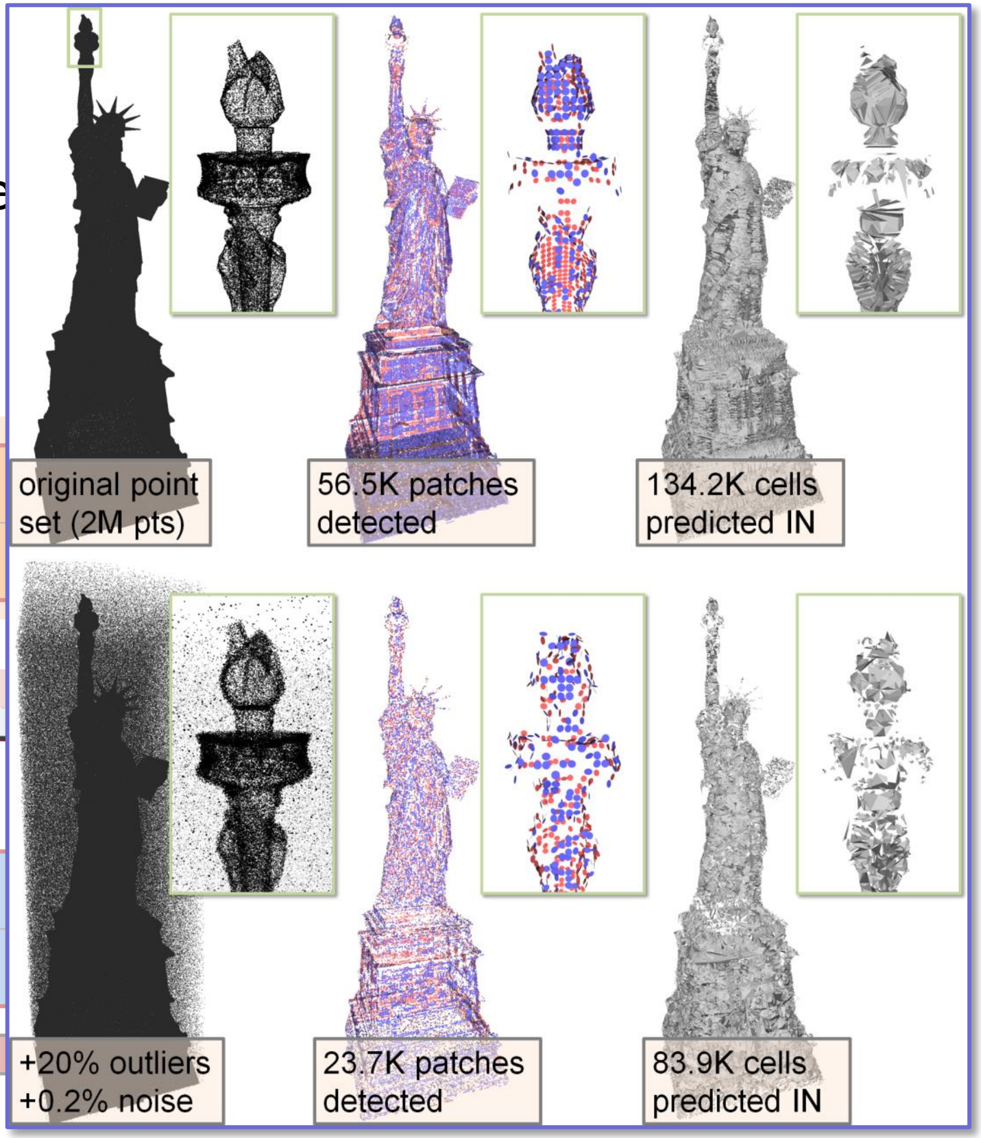
$$\begin{cases} P_{out}(c_k) = \beta \cdot 1_{\{c_k \in \mathcal{P}_{out}\}} \\ P_{in}(c_k) = \beta \cdot 1_{\{c_k \in \mathcal{P}_{in}\}} \end{cases}$$



Visibility prediction

- detection of visibi
- *inside/outside* pre ray





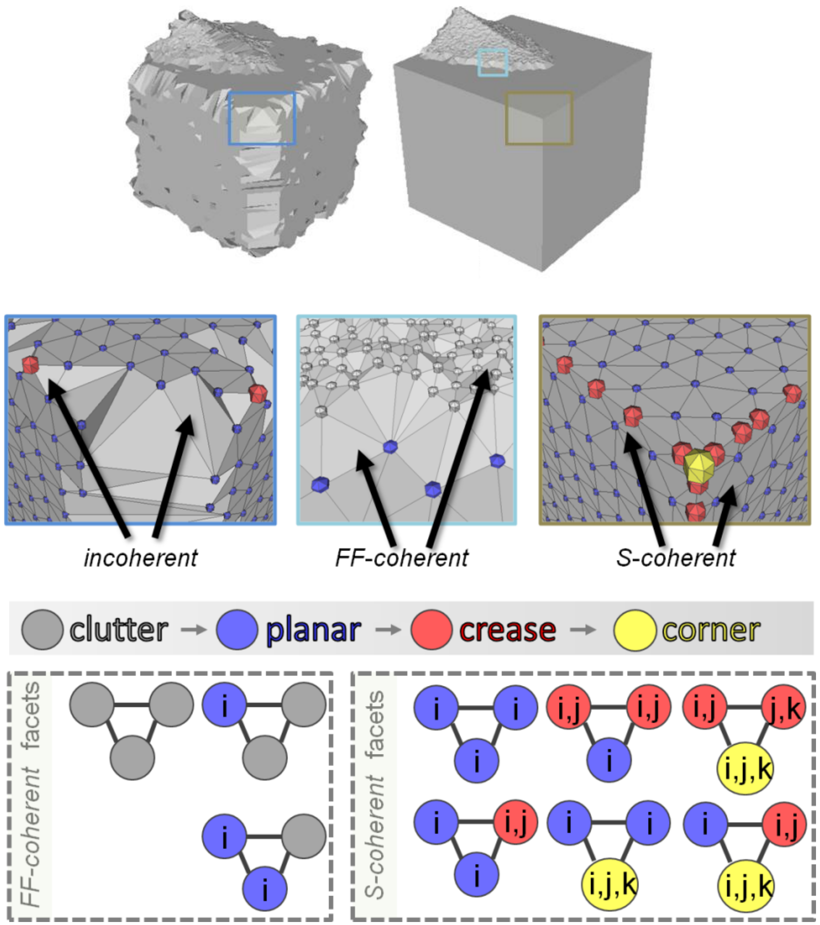
Geometric quality

▪ S-coherent facets
Plausible facets as a portion of a canonical part

▪ FF-coherent facets
Plausible facets as a portion of a freeform shape.

▪ Incoherent facets
all the remaining cases

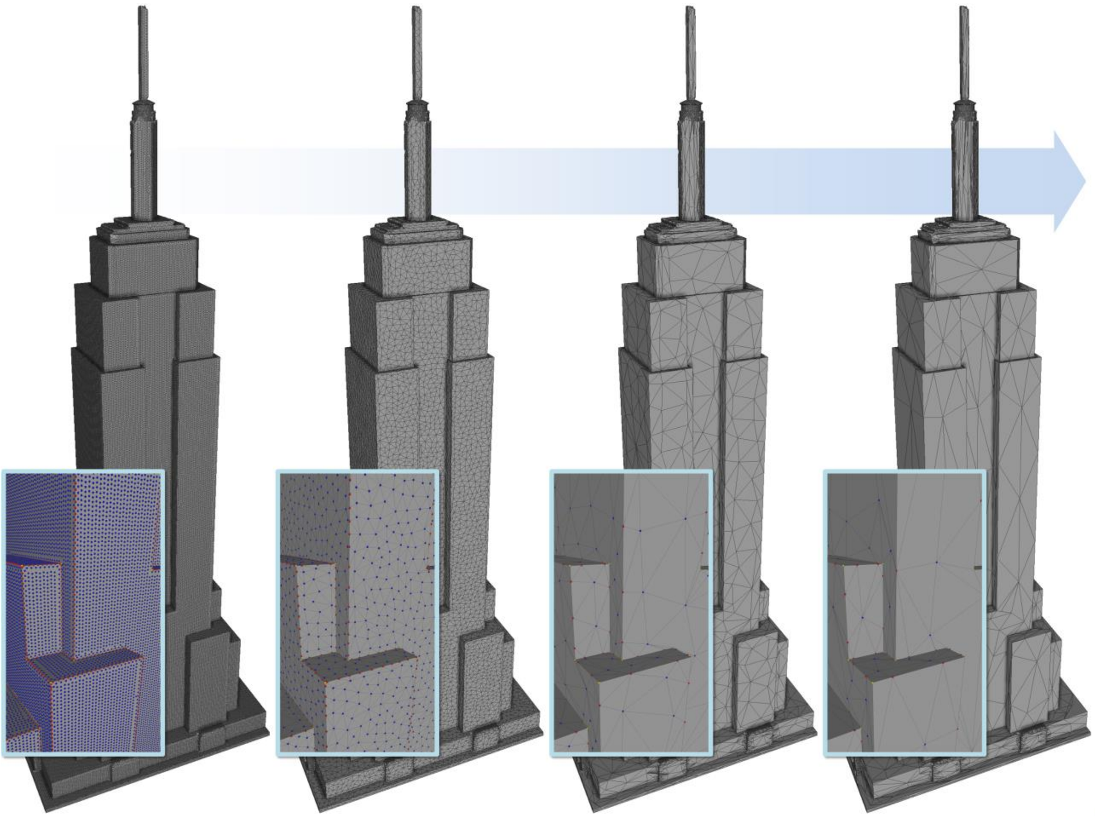
$$Q(f_i) = \begin{cases} 0 & \text{if } f_i \textit{ S-coherent} \\ g(f_i) & \text{if } f_i \textit{ FF-coherent} \\ \gamma & \text{if } f_i \textit{ incoherent} \end{cases}$$



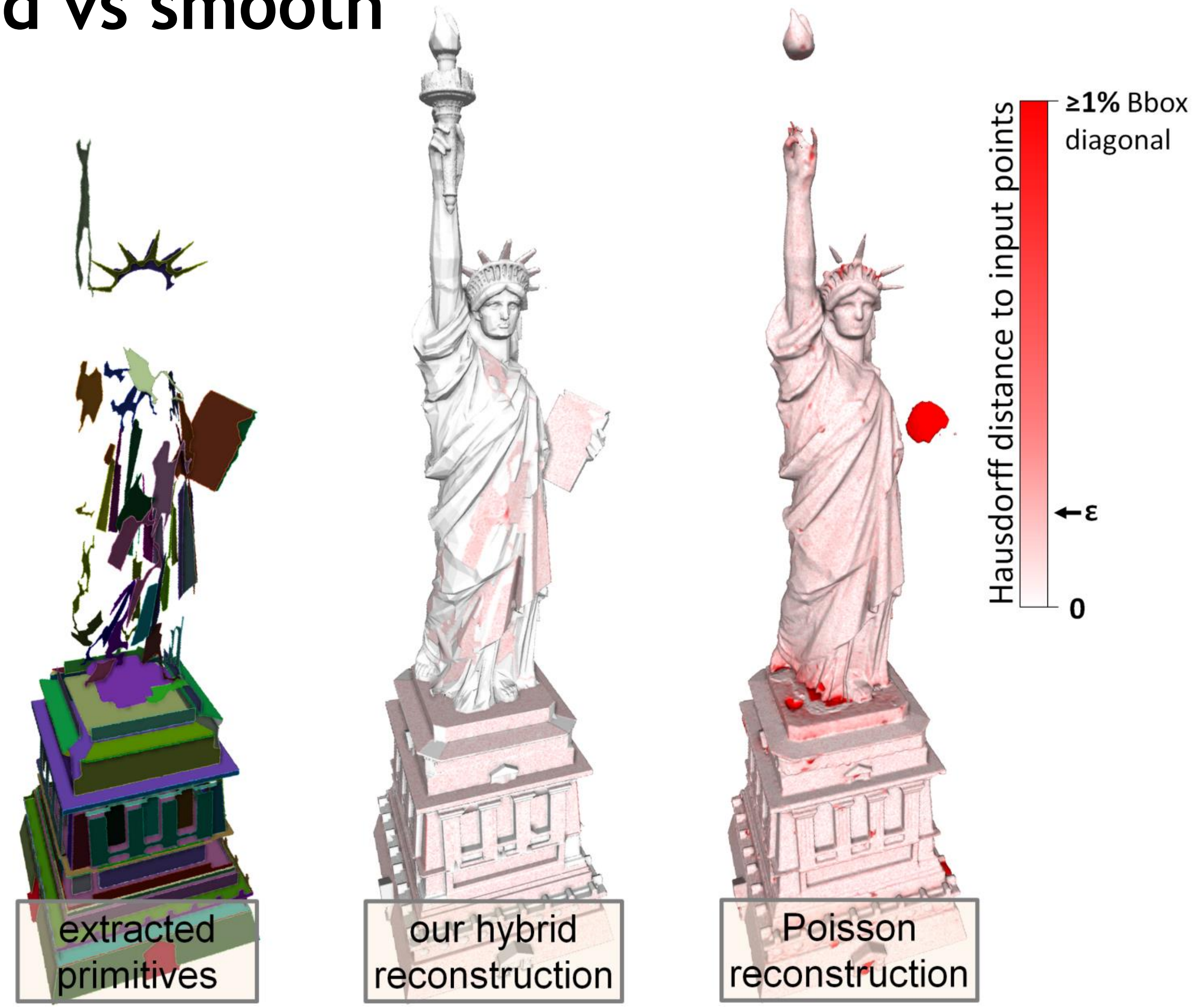
Surface simplification: edge-collapse exploiting the structural meaning of vertices

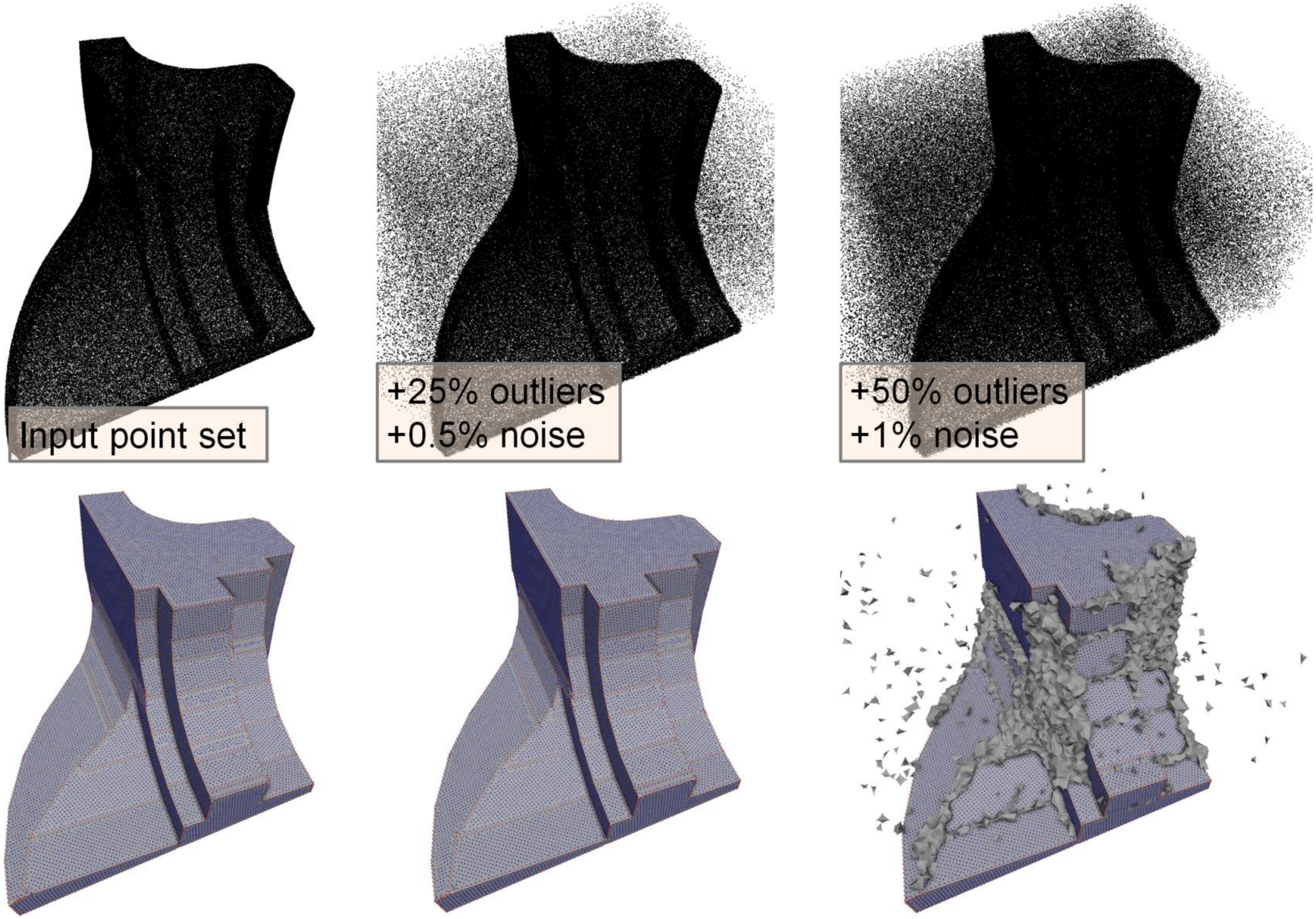
- canonical parts edge length cost to edges linking identical *planar* or *crease* vertices

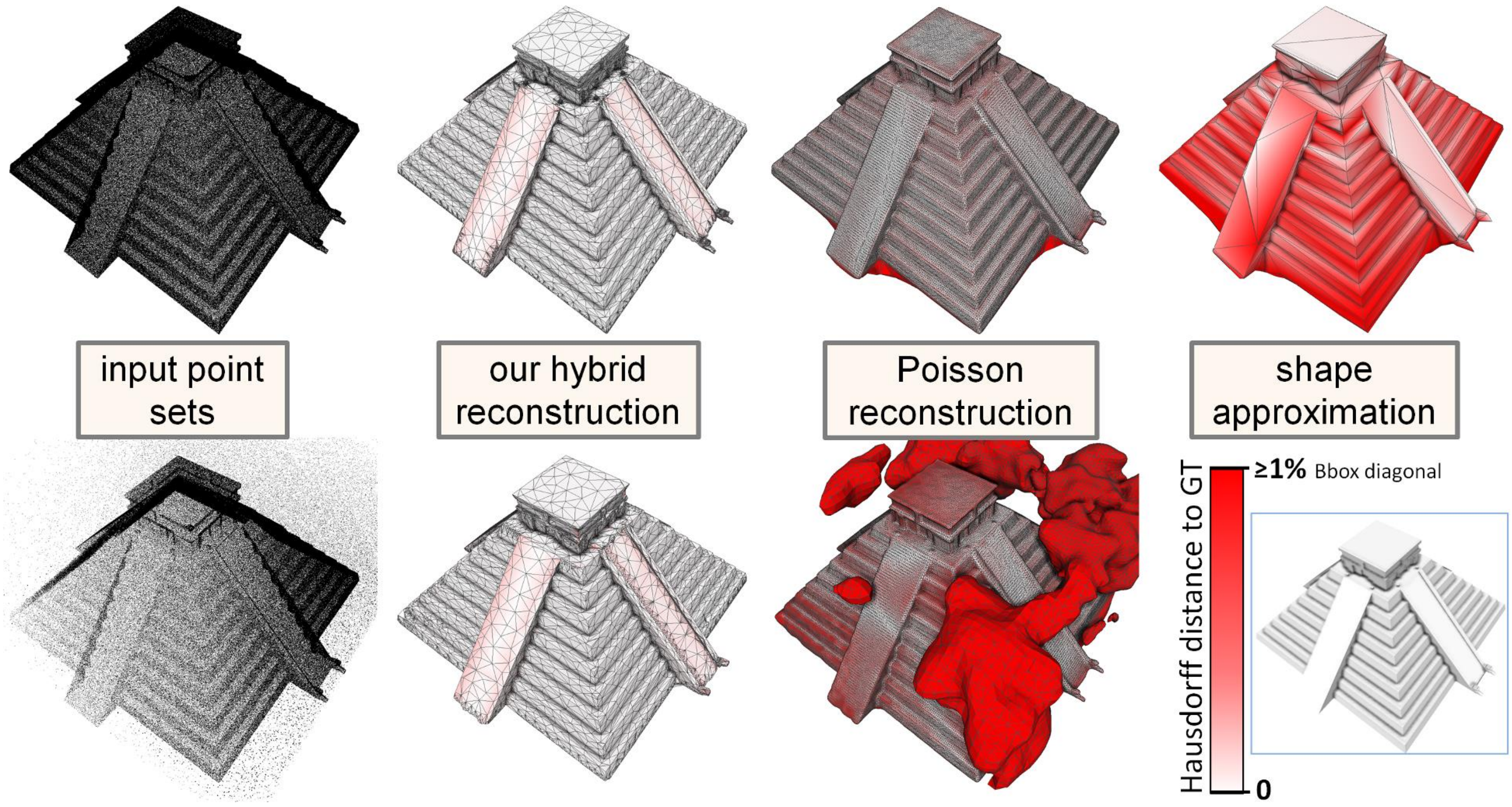
- free-form parts
Keep unchanged

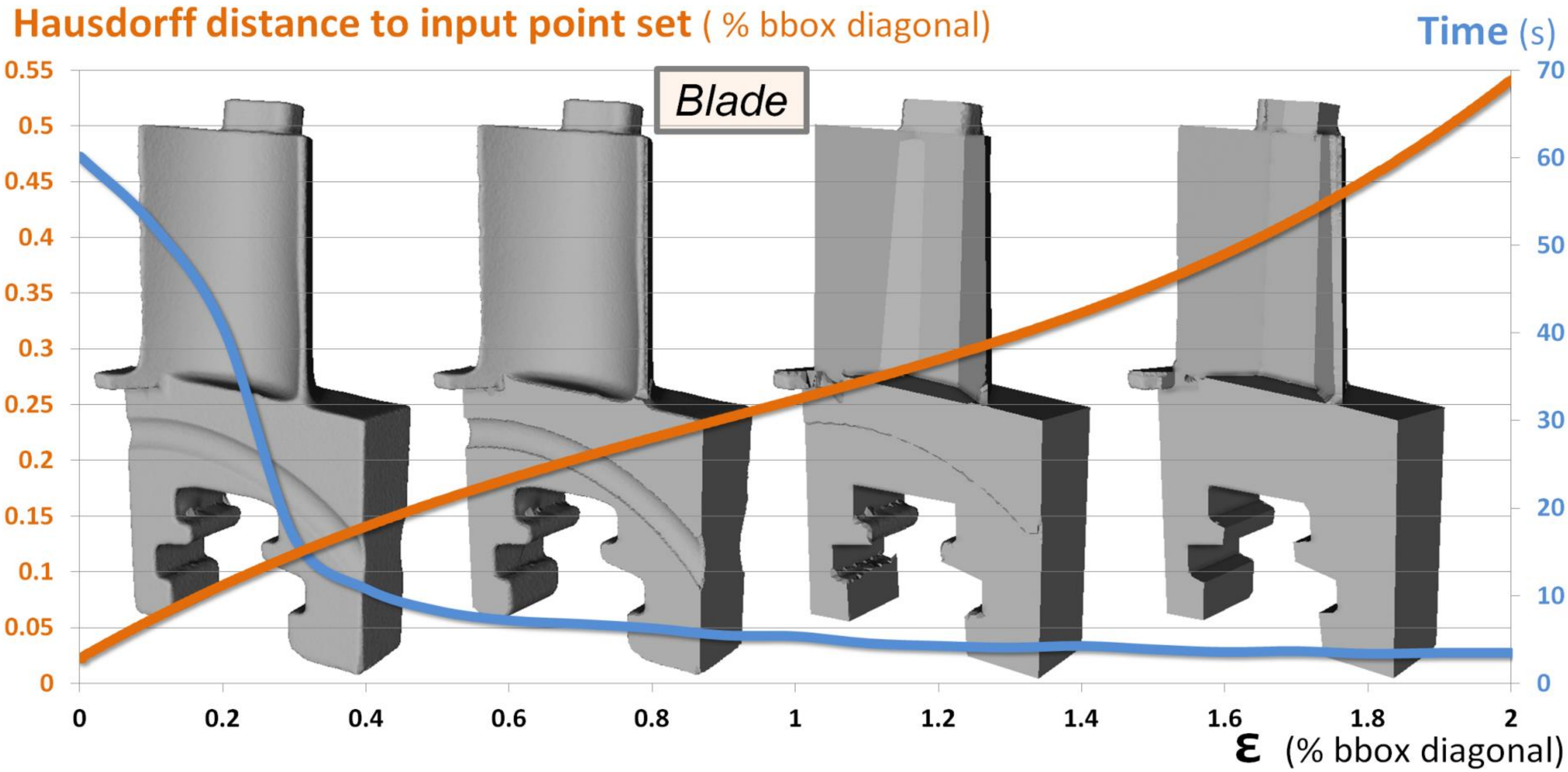


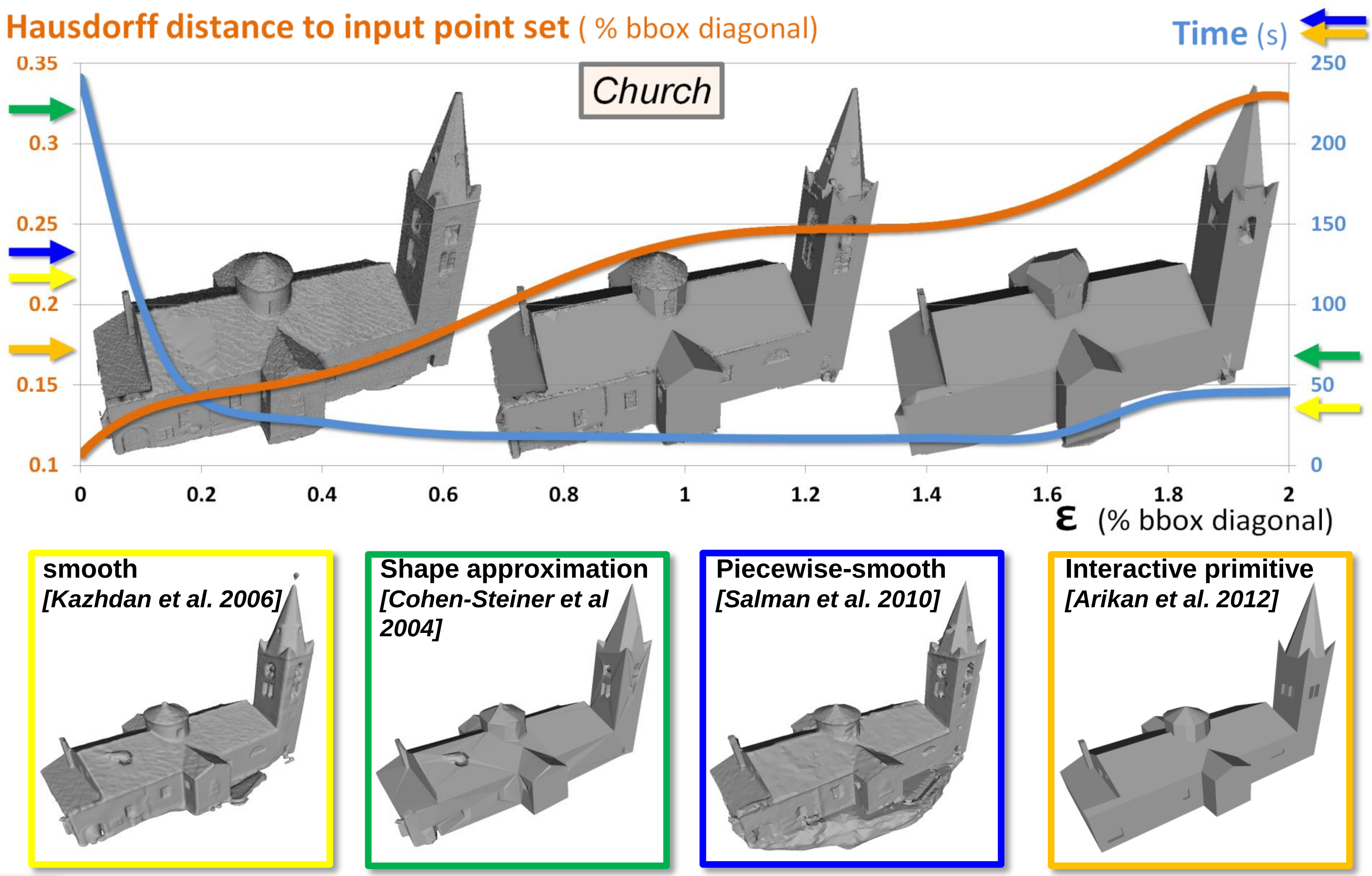
Hybrid vs smooth

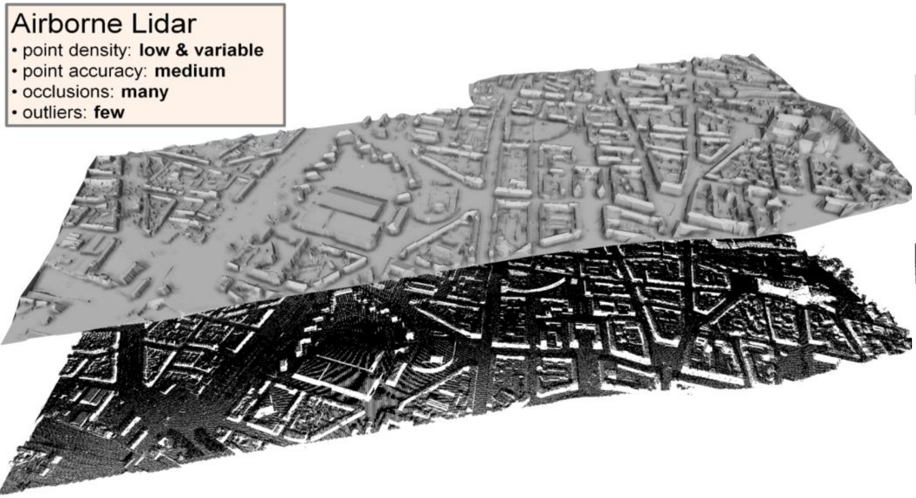


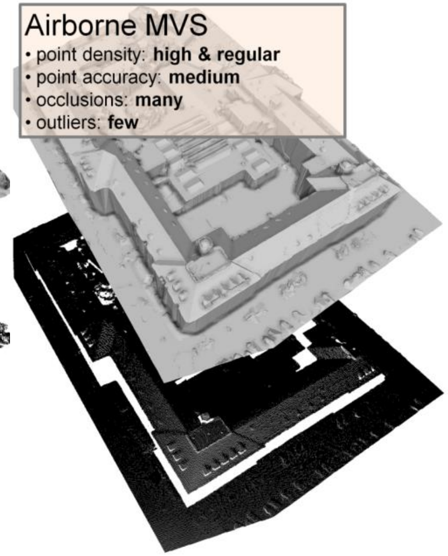

Input point set
+25% outliers
+0.5% noise
+50% outliers
+1% noise

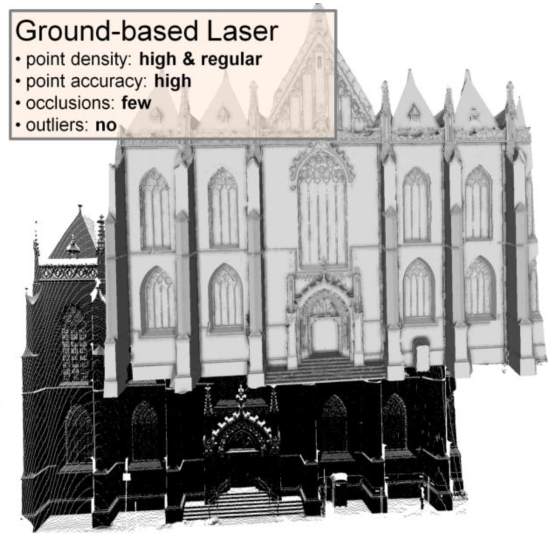

input point sets
our hybrid reconstruction
Poisson reconstruction
shape approximation
≥1% Bbox diagonal
Hausdorff distance to GT
0

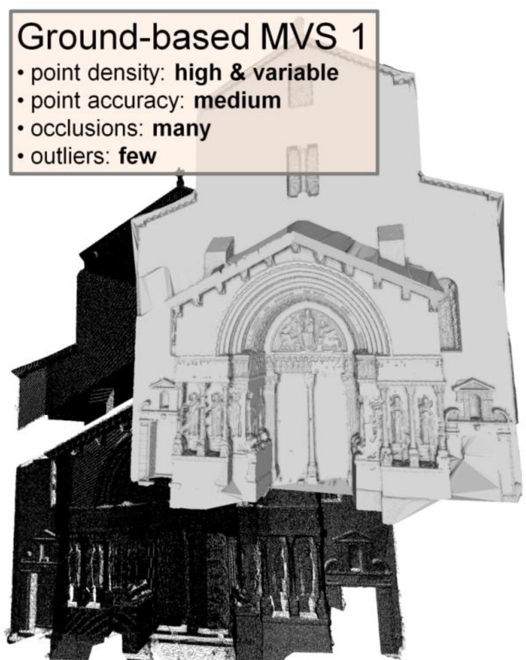

Hausdorff distance to input point set (% bbox diagonal)
Time (s)
Blade
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
70
60
50
40
30
20
10
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
ε (% bbox diagonal)

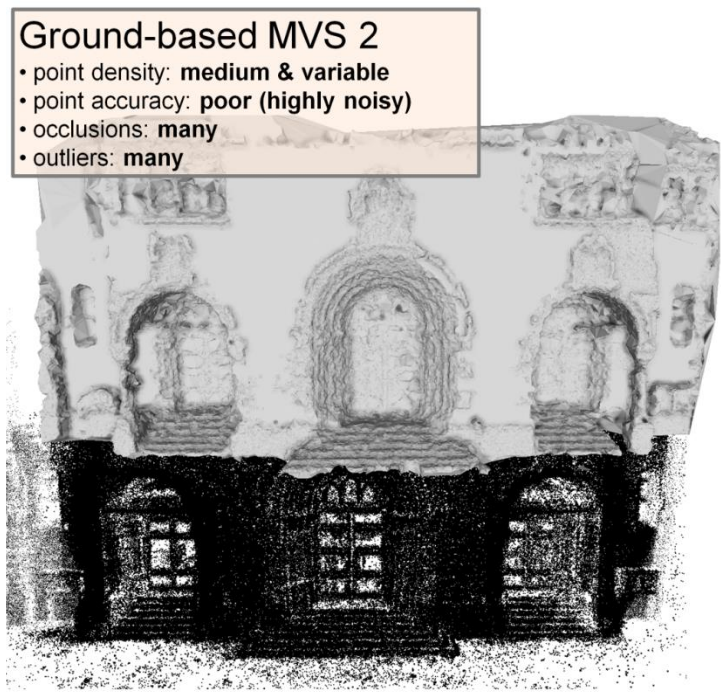

Hausdorff distance to input point set (% bbox diagonal)
Time (s)
Church
0.35
0.3
0.25
0.2
0.15
0.1
250
200
150
100
50
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
ε (% bbox diagonal)
smooth
[Kazhdan et al. 2006]
Shape approximation
[Cohen-Steiner et al 2004]
Piecewise-smooth
[Salman et al. 2010]
Interactive primitive
[Arikan et al. 2012]

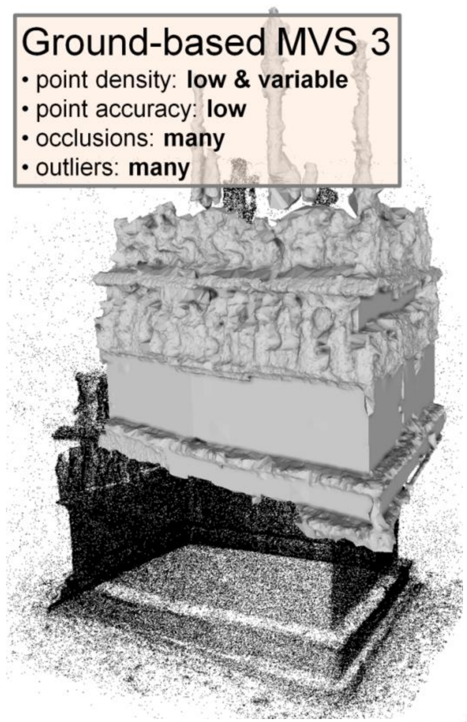

Airborne Lidar
• point density: **low & variable**
• point accuracy: **medium**
• occlusions: **many**
• outliers: **few**


Airborne MVS
• point density: **high & regular**
• point accuracy: **medium**
• occlusions: **many**
• outliers: **few**


Ground-based Laser
• point density: **high & regular**
• point accuracy: **high**
• occlusions: **few**
• outliers: **no**


Ground-based MVS 1
• point density: **high & variable**
• point accuracy: **medium**
• occlusions: **many**
• outliers: **few**

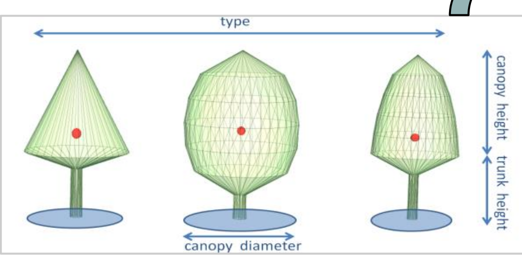

Ground-based MVS 2
• point density: **medium & variable**
• point accuracy: **poor (highly noisy)**
• occlusions: **many**
• outliers: **many**


Ground-based MVS 3
• point density: **low & variable**
• point accuracy: **low**
• occlusions: **many**
• outliers: **many**

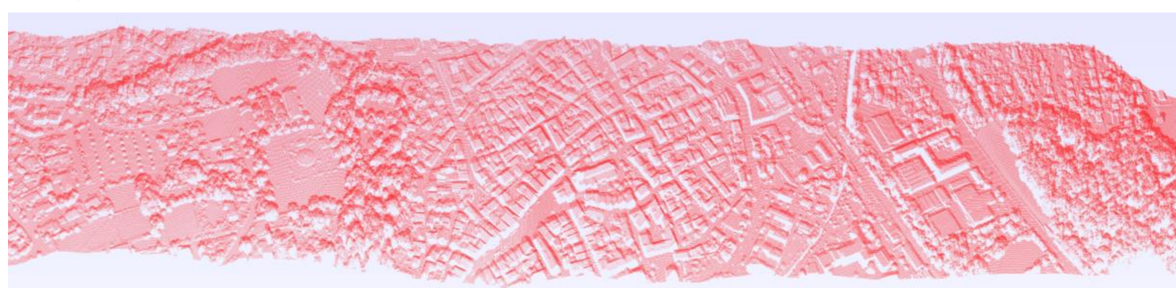
- Geometric primitive extraction
- Surface reconstruction using geometric primitives
- Two words on template matching

Template matching

- Geometric primitives are usually simple, eg planes or cylinders
- But sometimes, we need to fit more complex primitives to the data..

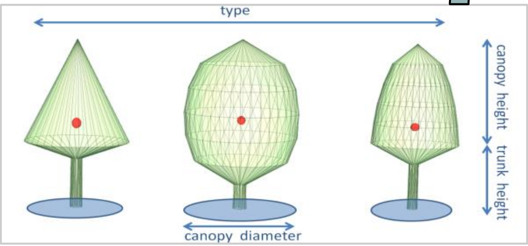


?

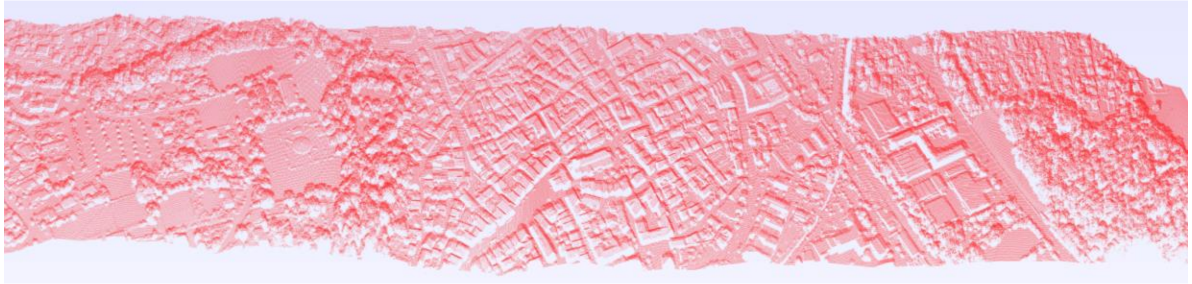


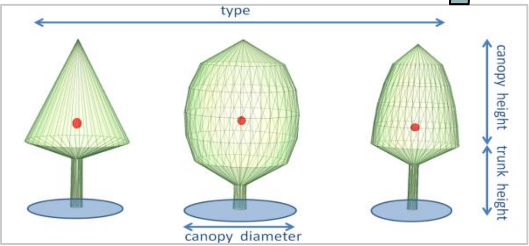
Problems

- Do we search for one or several objects in the data ?
- Do we know the number of objects?
- Can objects interact between each others ?

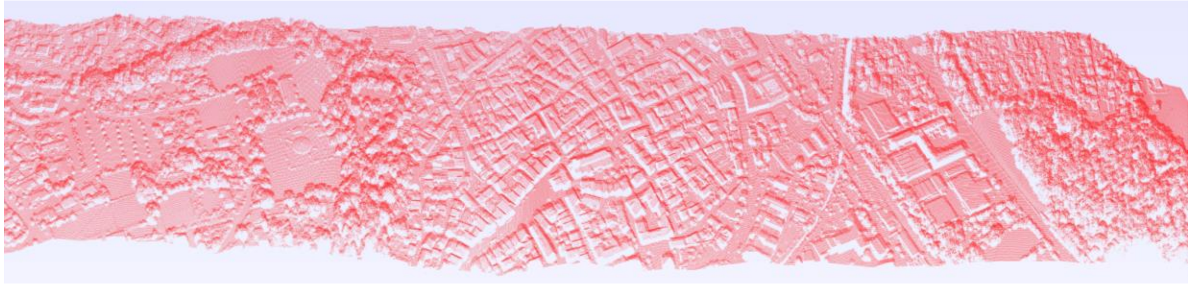


?





?



- Here, we don't know the number of objects and interactions must be inserted (spatial overlapping, tree competition..)
- .. this is not surface reconstruction anymore