Computer Vision

Raphaël Viards – Magellium – Imagerie et Applications

raphael.viards@magellium.fr

Partie II: Analyse de l’environnement 3D

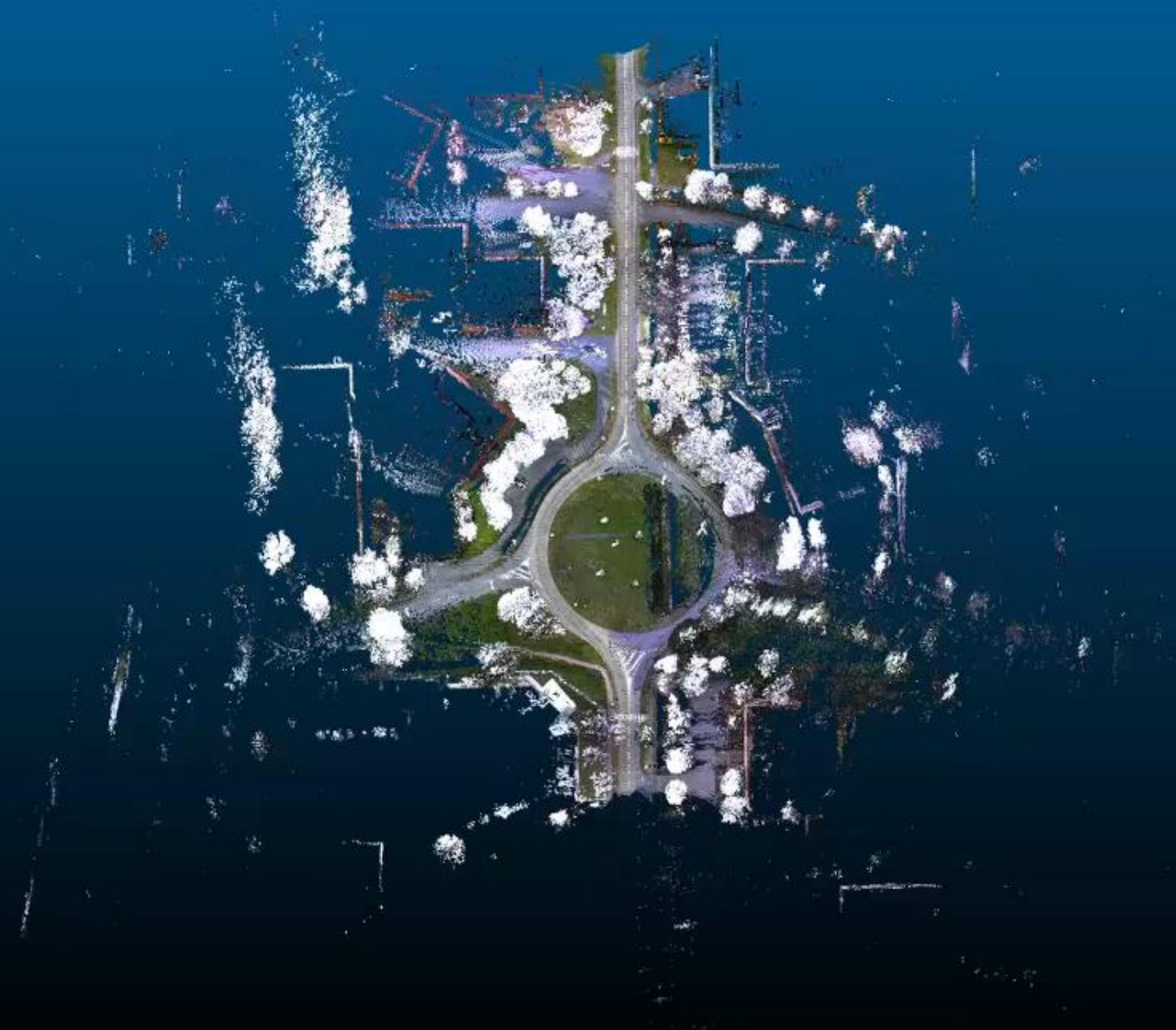
I. Introduction
II. Analyse de surfaces et Reconstruction 3D
III. Segmentation 3D
IV. Recalage 3D

Valorisation de la donnée 3D

- Nous savons obtenir de l'information 3D sous la forme d'un nuage de points à l'aide d'un système de numérisation 3D (c.f. Partie I : Perception 3D).
- Comment valoriser cette donnée?
  - Visualisation 3D
  - Reconstruction de modèles 3D
  - Building Information Modeling
  - Clustering 3D
  - Dimensionnement 3D
  - Détection d'objets 3D
  - Modèle numérique de terrain (2.5D ou 3D)
  - Navigation Autonome
  - …

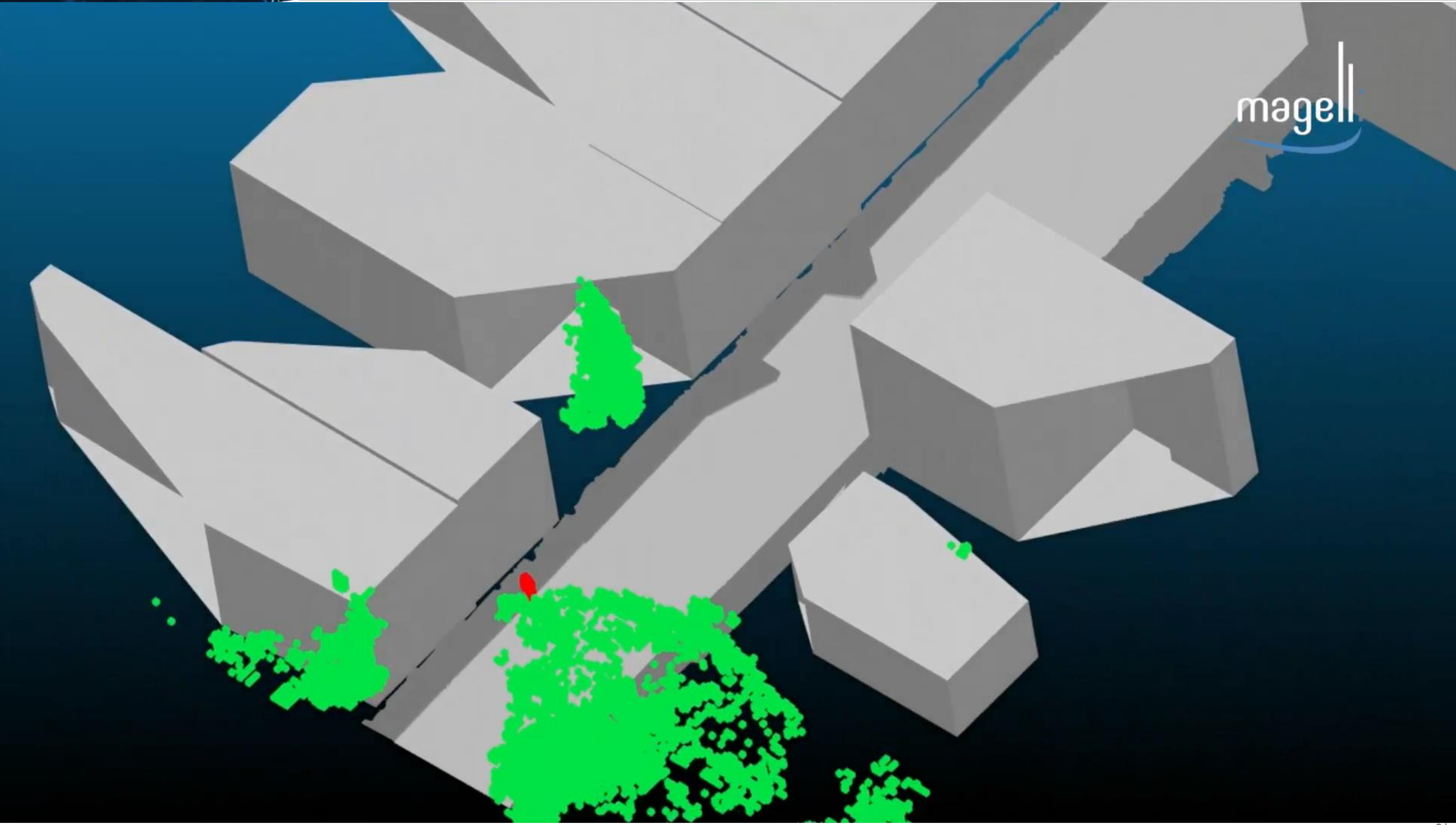
- Comment passer d'un nuage de points à une donnée à valeur ajoutée?
    - Analyse de surfaces 3D
    - Reconstruction 3D
    - Segmentation 3D
    - Recalage 3D
    - …

Visualisation 3D



Reconstruction de modèles 3D

Building Information Modeling



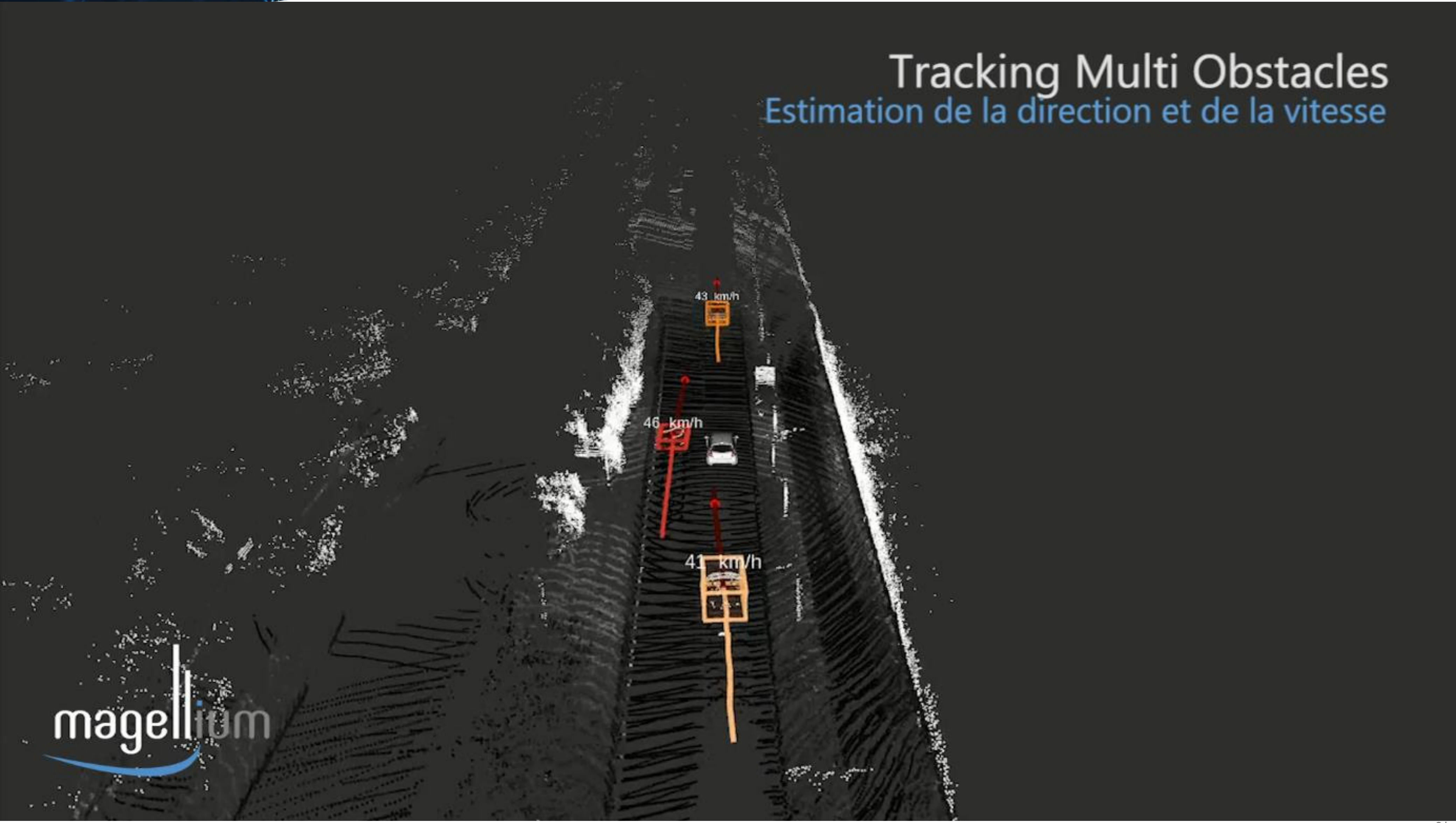
Segmentation sémantique

SEGMENTATION SÉMANTIQUE DE DONNÉES LIDAR

RÉSULTAT 1 : Nos réseaux de neurones segmentent les données LiDAR en continu

À chaque point est attribué une classe dont le code couleur est affiché en haut à gauche. L'image en bas à gauche est issue de la caméra contextuelle et n'est pas utilisée par le réseau.

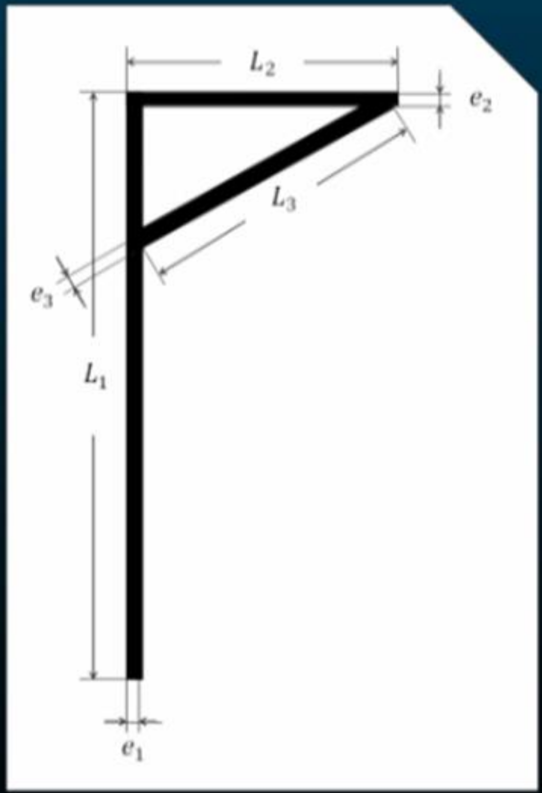
Application véhicule autonome



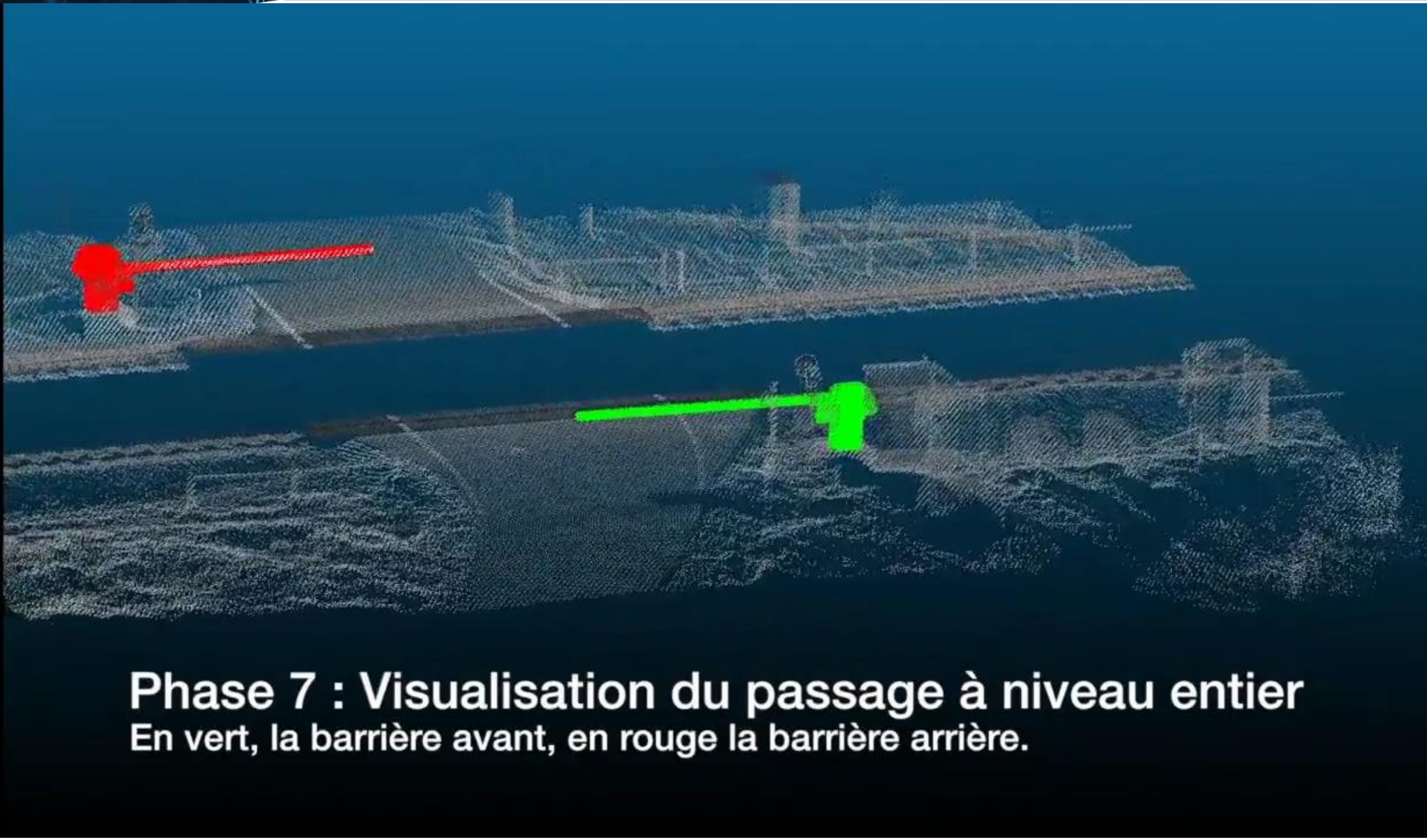
Dimensionnement 3D

Détection de lignes dans le nuage

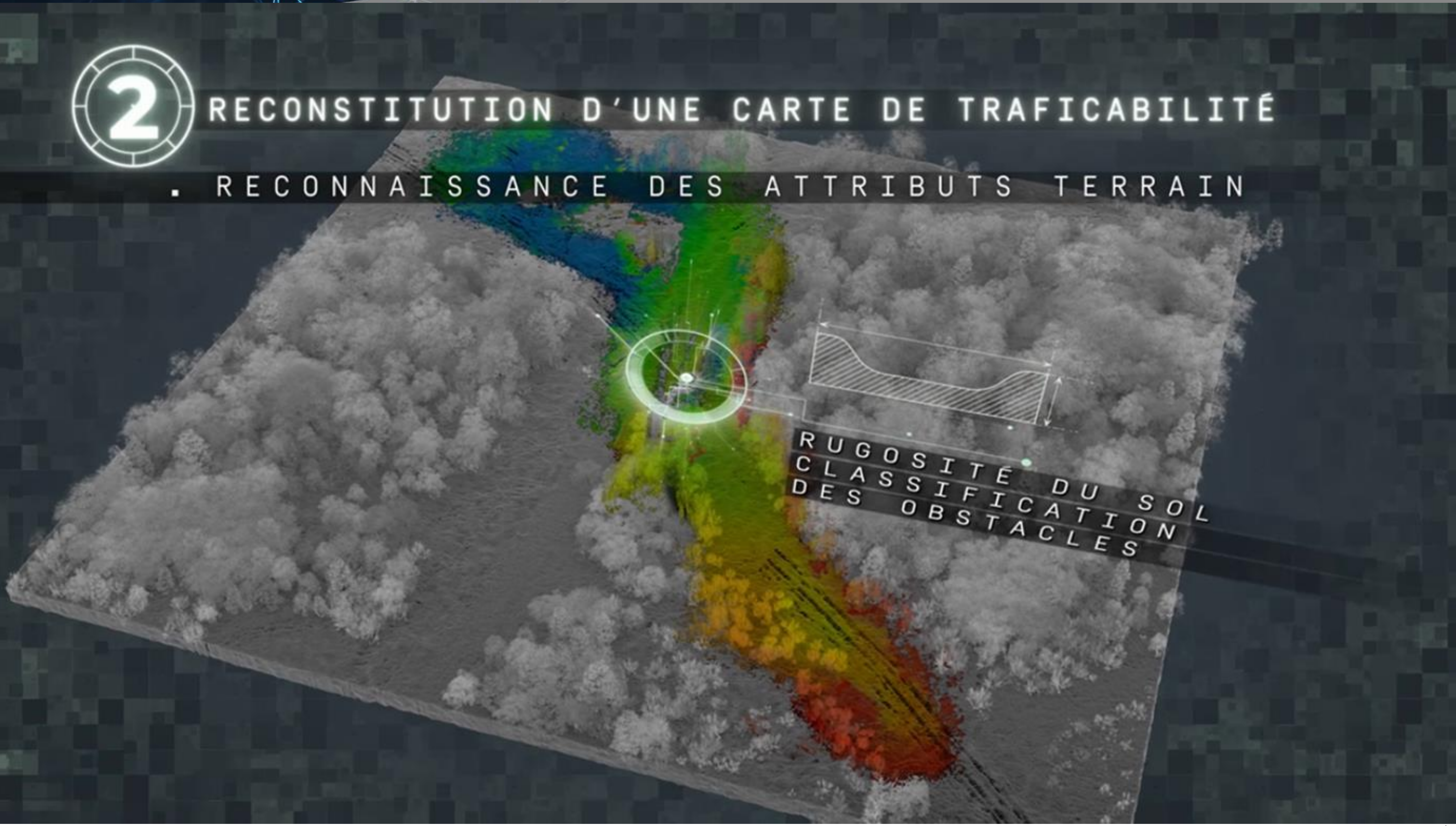
Basée sur l'algorithme de RANSAC



Détection d'objets 3D

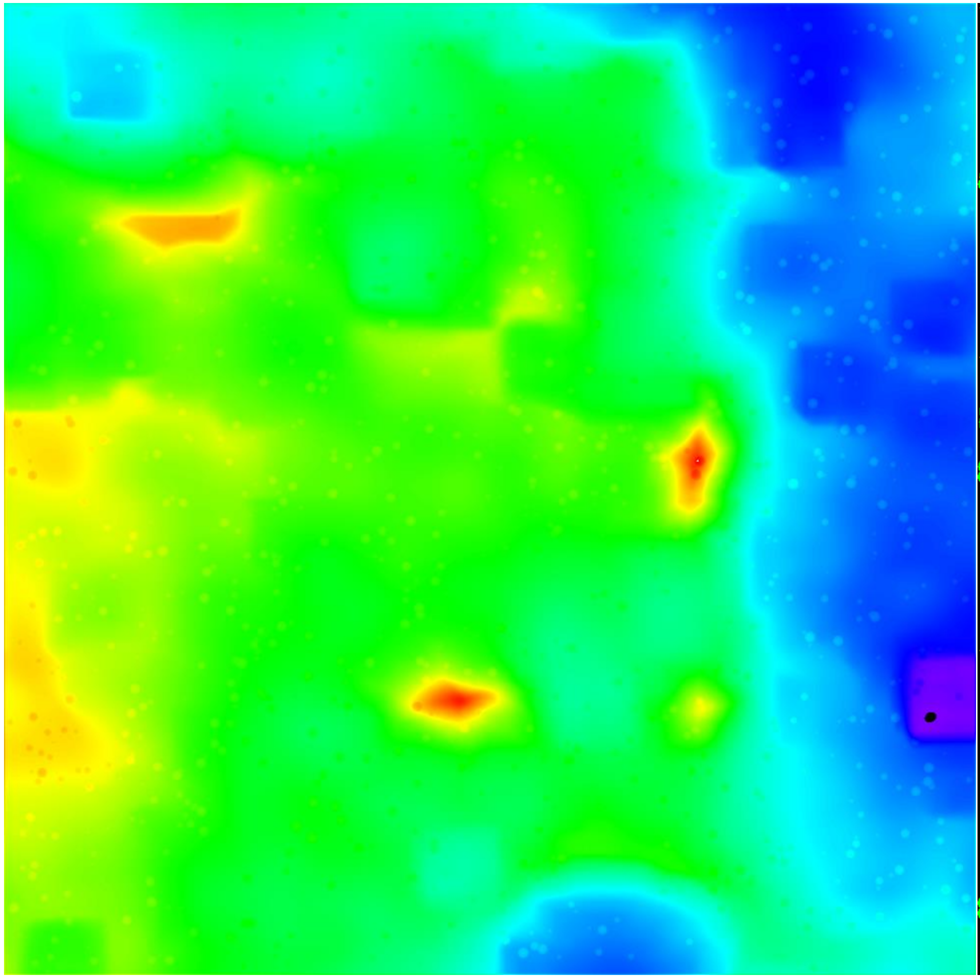


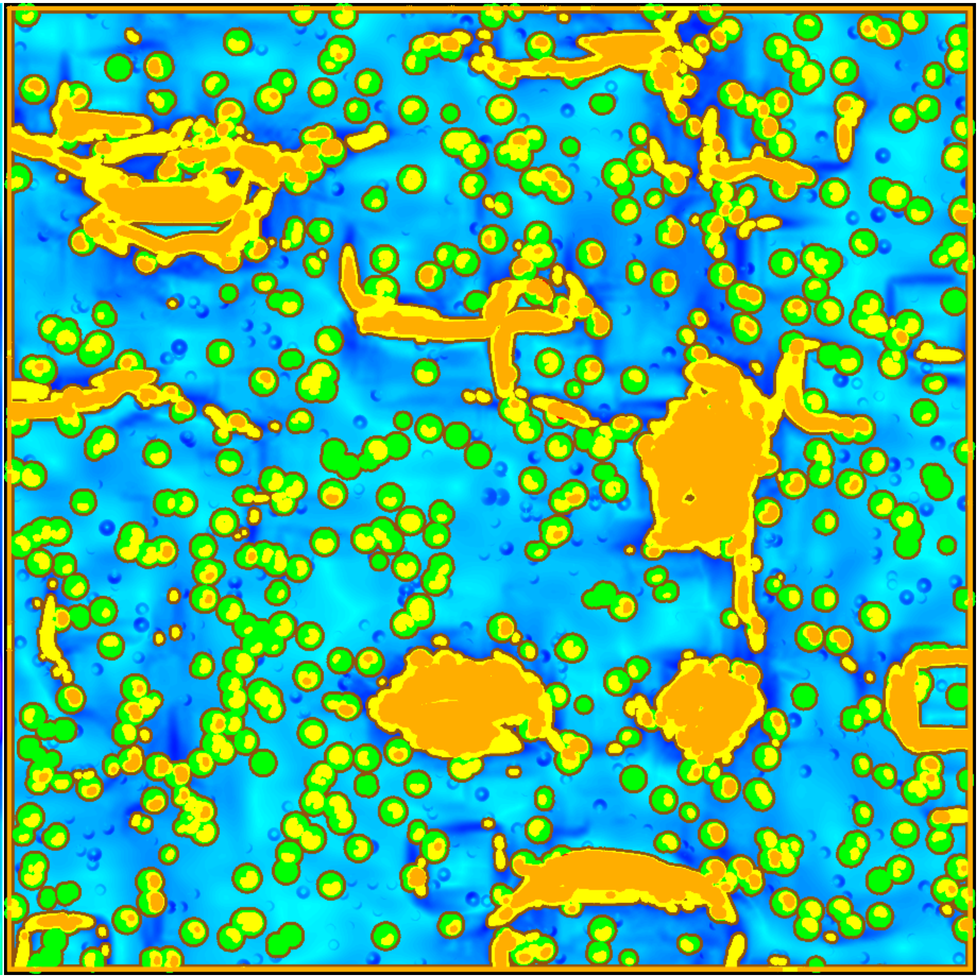
MNT et Trafficabilité



magellium artal group

MNT & Carte de Navigation



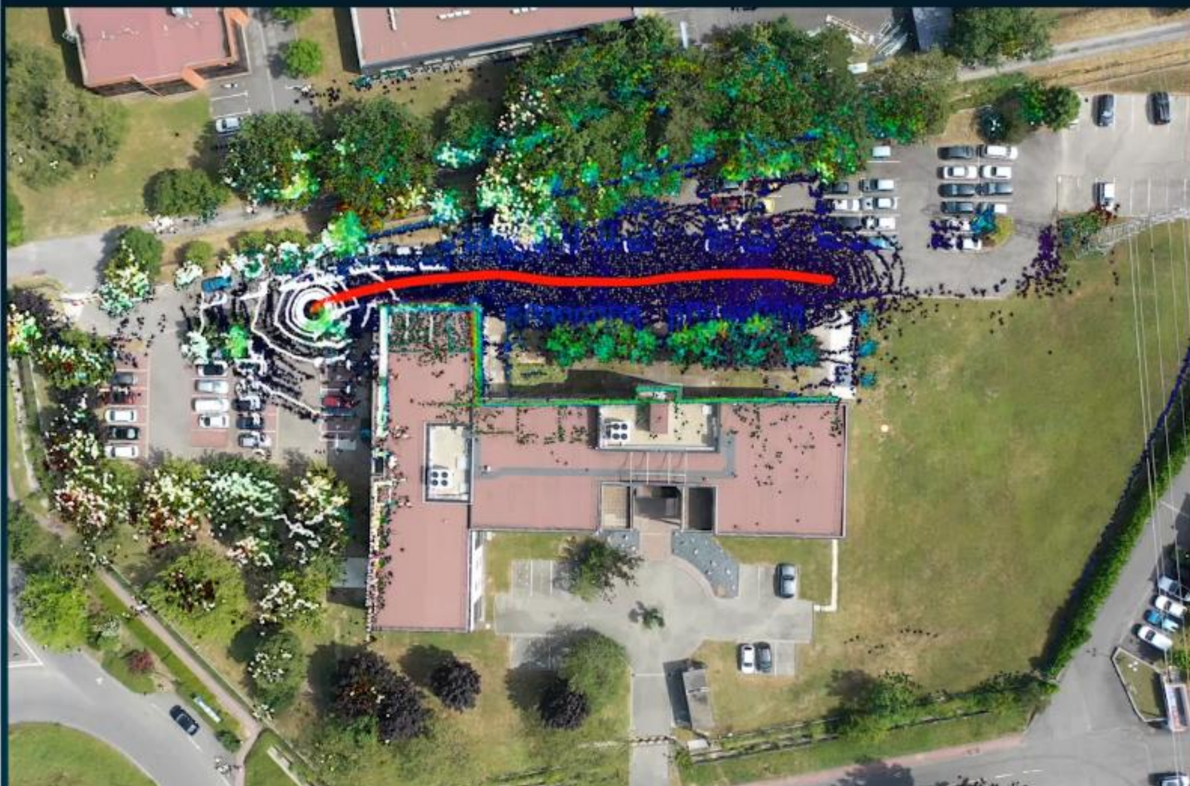


AUTOMATIC HD MAP DESIGN

OBJECTIVE : Using onboard LiDAR and Camera fusion to create High Definition Maps

3D estimation of detected lines and traffic signs are used to create the HD Map of the current lane. The perception is performed online whereas the HD Map design can be completed offline.

NAVIGATION AUTONOME BASÉE LIDAR



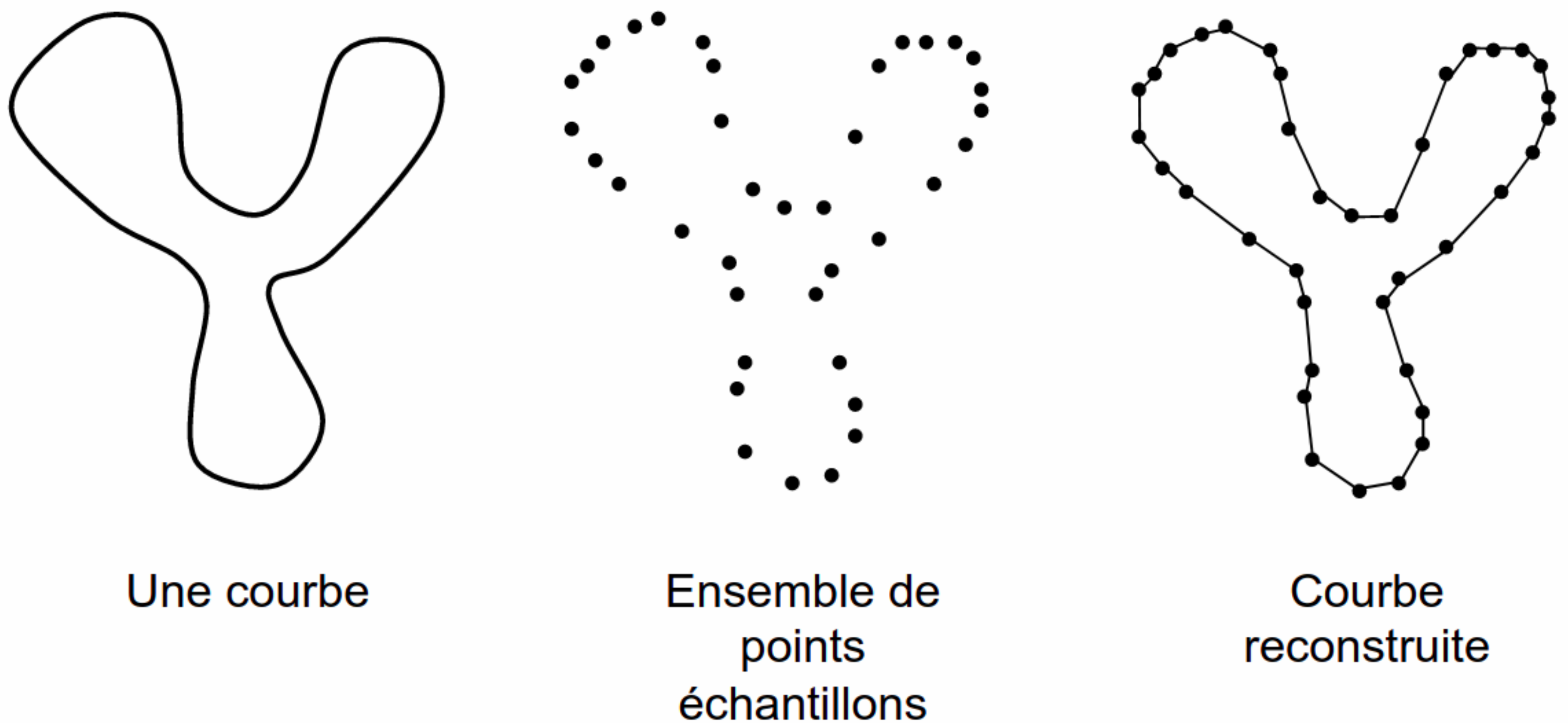
Cartographie et localisation 3D simultanées

La plateforme robotique mobile contrôlée manuellement cartographie l'environnement en 3D grâce à son LiDAR 360°.

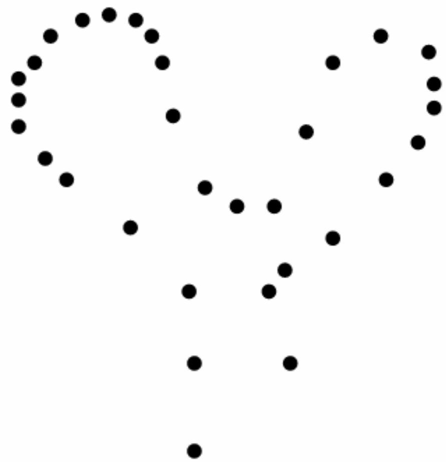
Partie II: Analyse de l’environnement 3D

I. Introduction
II. Analyse de surfaces et Reconstruction 3D
III. Segmentation 3D
IV. Recalage 3D

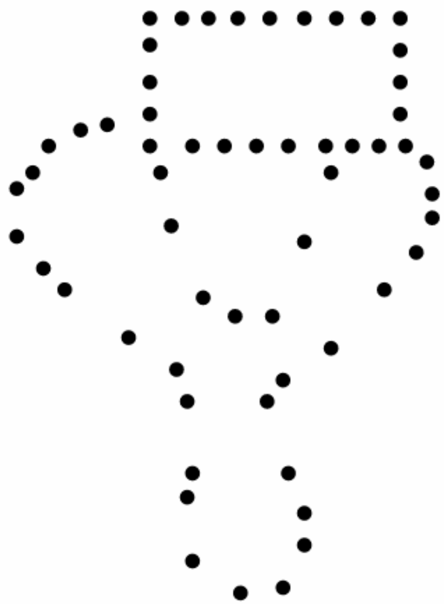
- Exemple de la problématique en 2D:



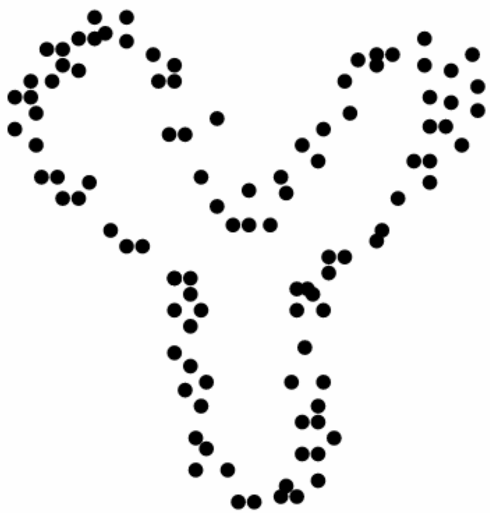
- Dans la réalité, le problème n’est pas si simple:



Densité non homogène

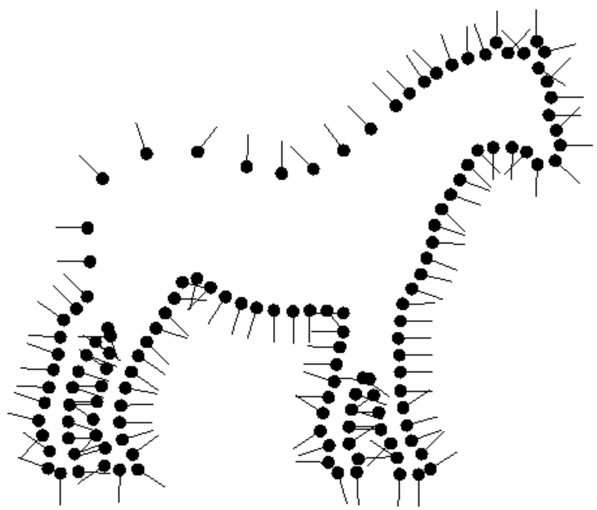


Superposition de courbes

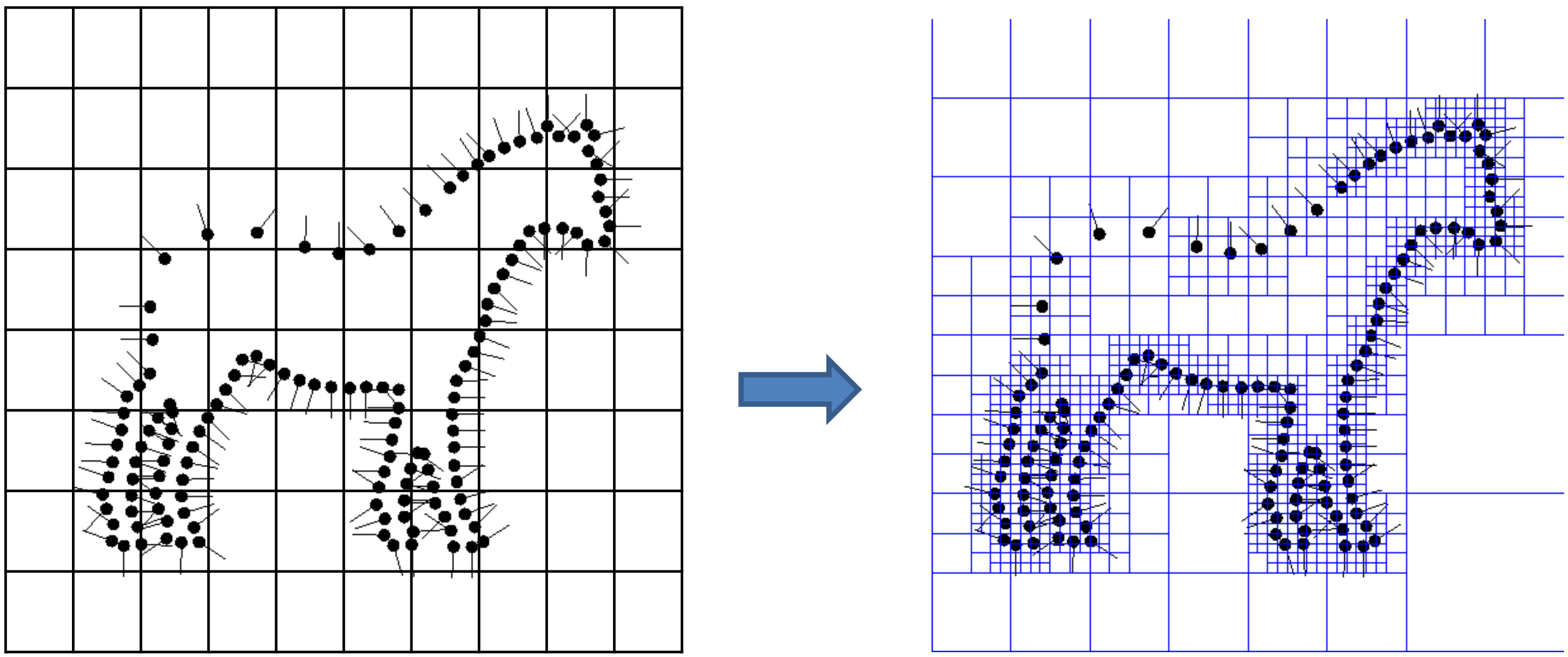


Bruit de mesure

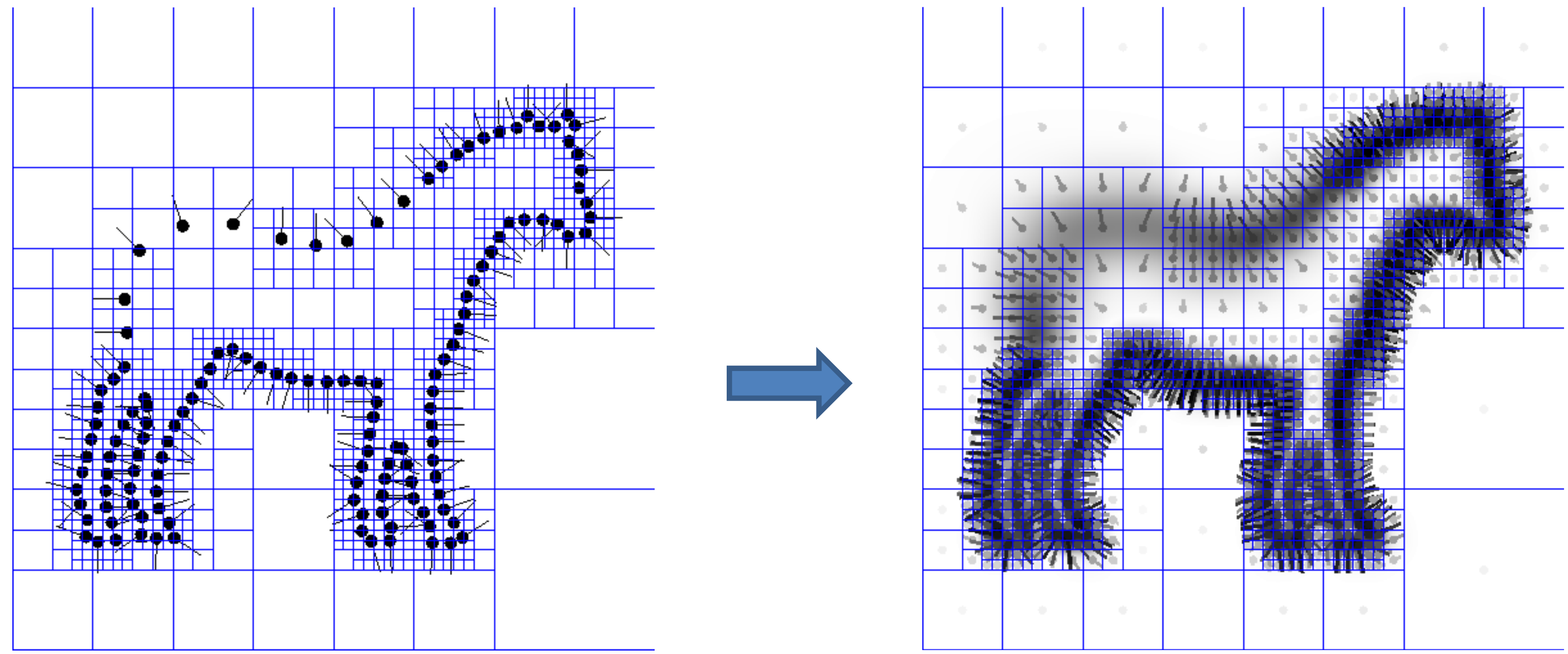
- Calcul des normales:



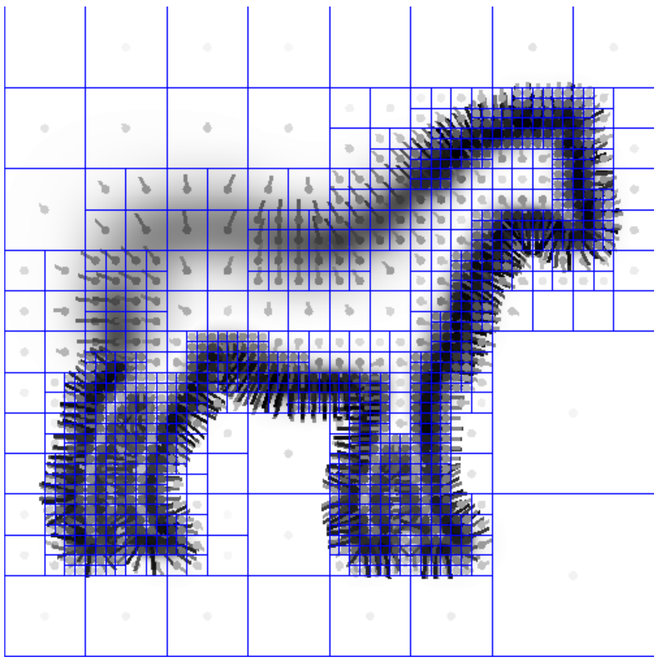
- Discrétisation Octree (1 point par cellule)



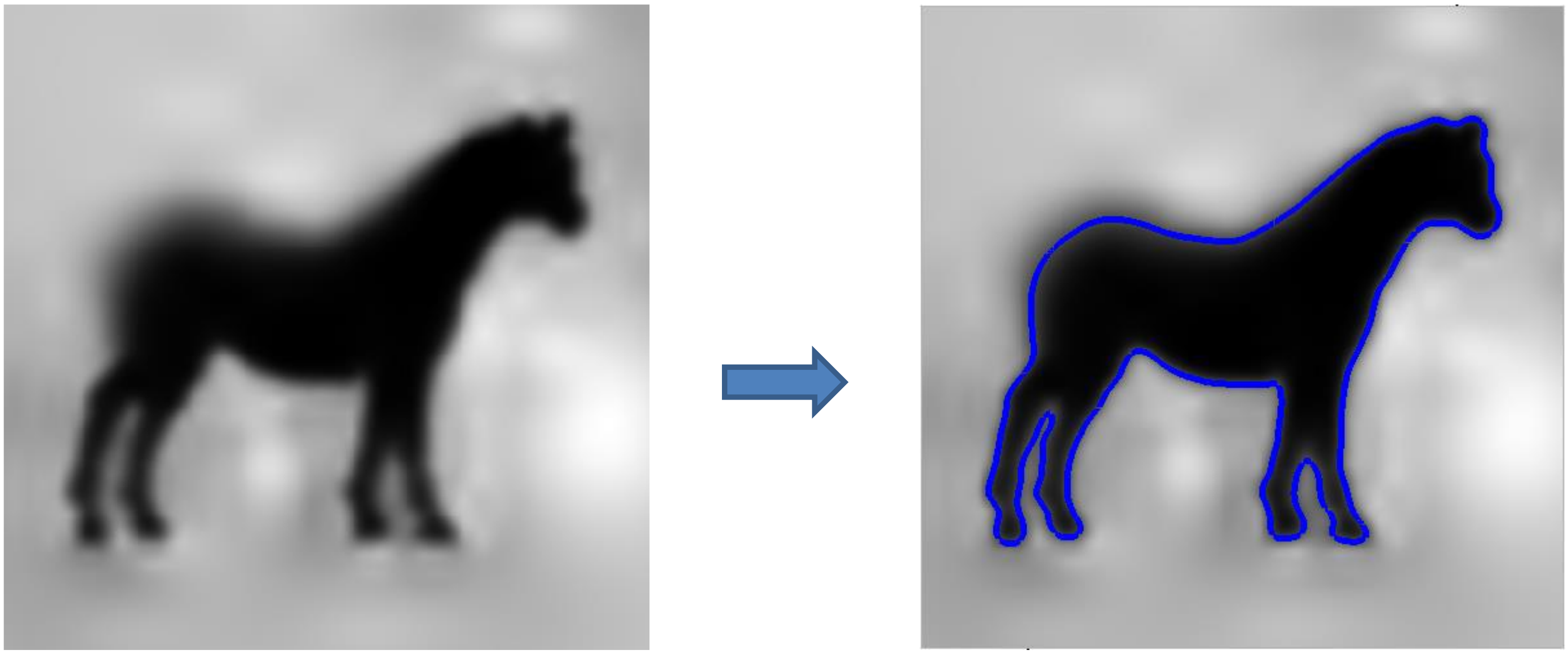
- Calcul d’un champ de vecteurs



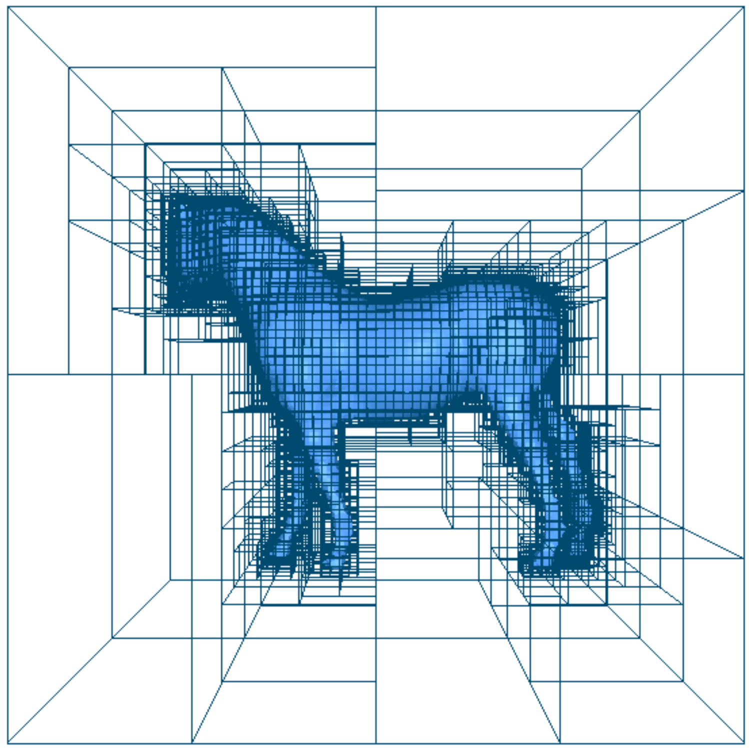
- Calcul de la fonction indicatrice
  - Fonction dont les valeurs inférieures à zero sont à l’extérieur et les valeurs positives à l’intérieur



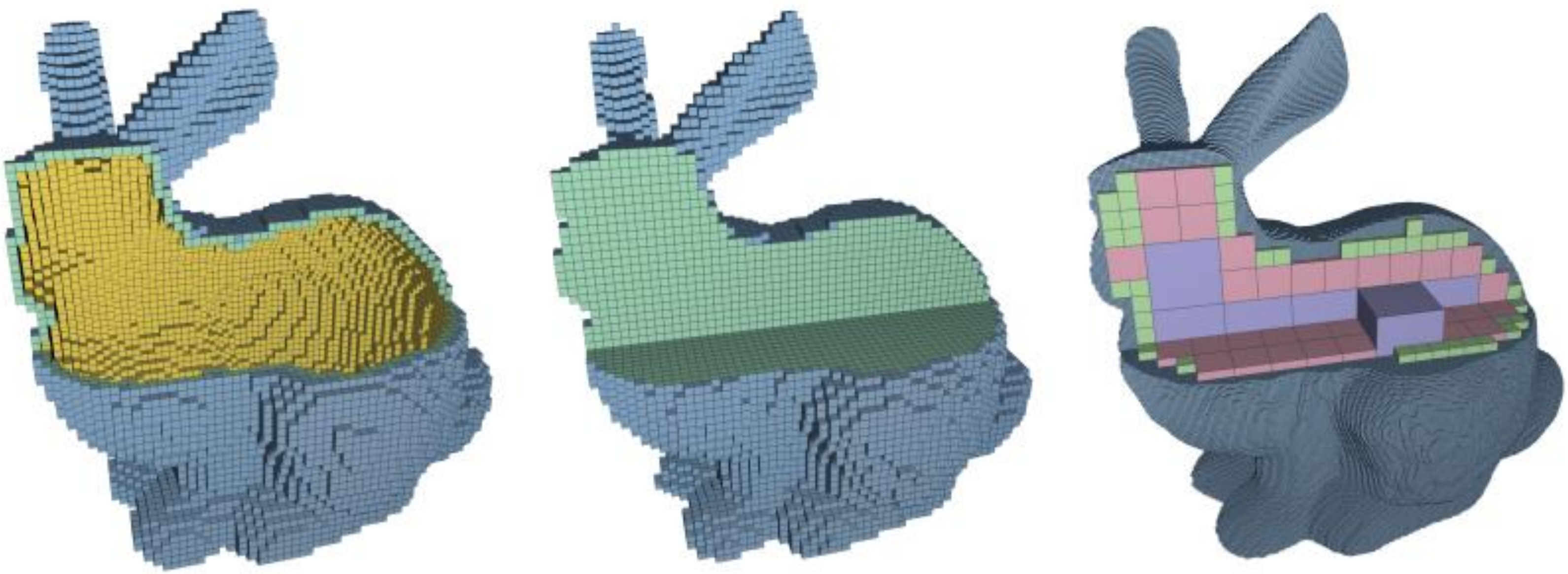
- Extraction de l'isosurface



- Découpage de l'espace en voxels



- Différentes méthodes de voxelisation

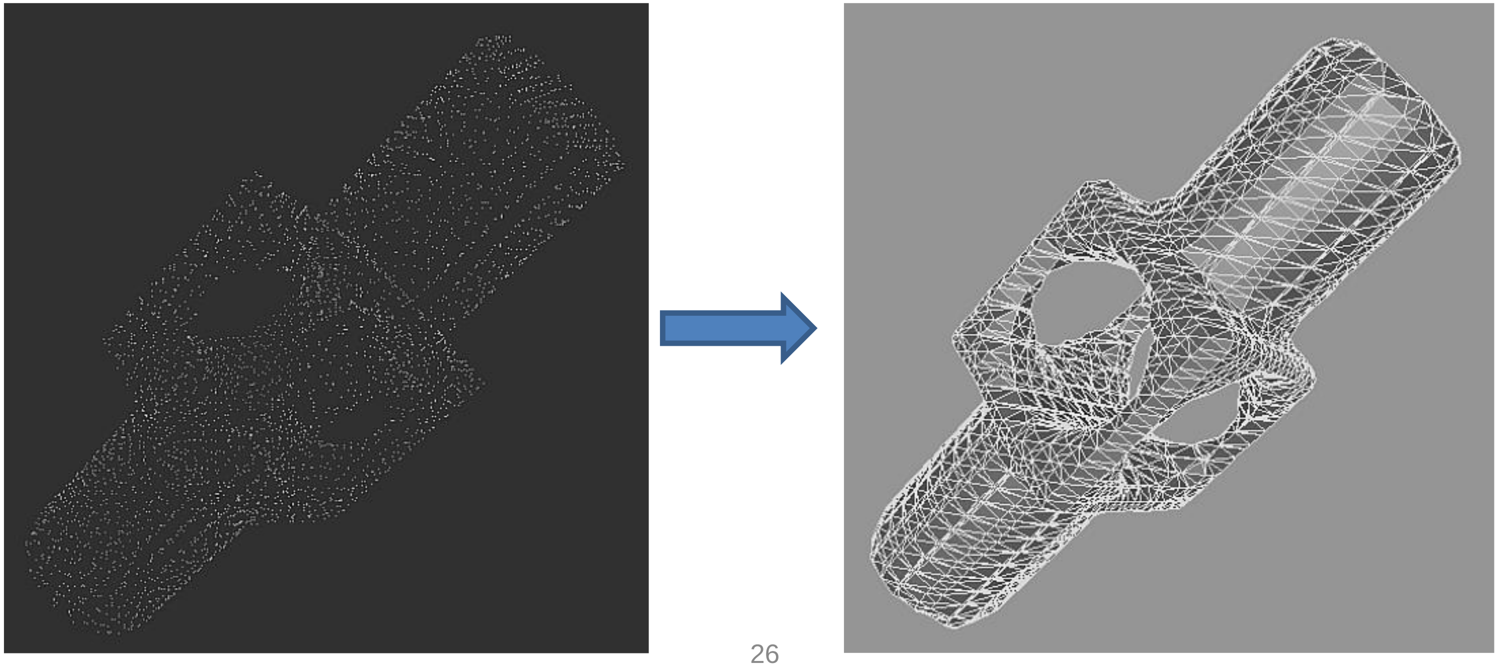


Conservative surface voxelization

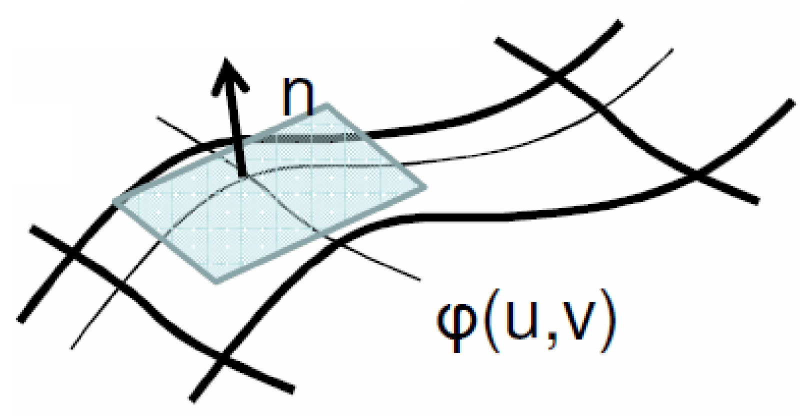
Solid Voxelization

Octree-based sparse solid voxelization

- Ensemble de sommets connectés
  - Maillage polygonal: connexion entre sommets formant des cycles et définissant des polygones
  - Maillage triangulaire: polygone = triangle
  - Plusieurs méthodes (e.g. triangulation de Delaunay)

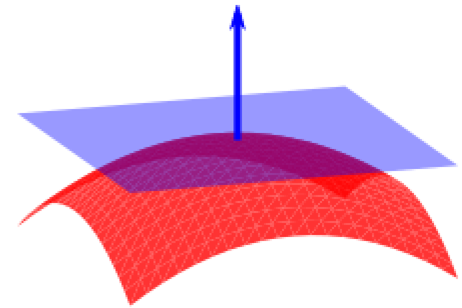


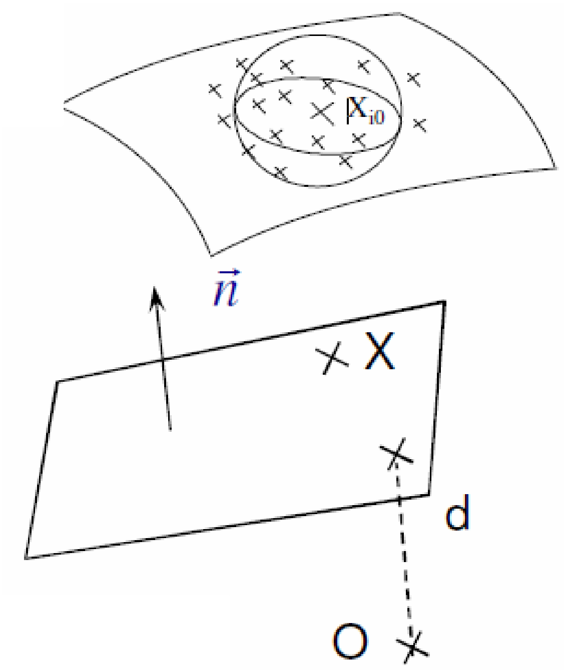
- Equation paramétrique d'une surface:
  - $\varphi : \begin{cases} V \subset \mathfrak{R}^2 \rightarrow S \subset \mathfrak{R}^3 \\ (u, v) \mapsto P = \varphi(u, v) \end{cases}$
- Système de coordonnées locales:
  - $\begin{cases} \varphi_u = \dfrac{\partial \varphi}{\partial u} \\ \varphi_v = \dfrac{\partial \varphi}{\partial v} \end{cases}$
  - La paramétrisation est régulière si φ_u et φ_v sont linéairement indépendants.



- On définit le plan tangent à la surface au point P.
- La normale à la surface en ce point est la normale du plan tangent:
  - $n = \dfrac{\varphi_u \times \varphi_v}{\|\varphi_u \times \varphi_v\|}$

- Calcul de normales par ACP:
  - On cherche le meilleur plan approché dans le voisinage d'un point X_{i0}
    - Les points du voisinage sont notés X_i
  - Equation d'un plan:
    - $n^t X = d, \|n\| = 1$
  - Distance signée d'un point au plan:
    - $d(X_i, P) = n^t X_i - d$





- Résolution
  - Méthode des moindres carrés
  - Résolution de l'équation de minimisation pour le plan
- Fonction à minimiser:
  - $f(n,d) = \sum_{i=1}^{m}(n^t X_i - d)^2$
  - 4 paramètres: (n,d), 1 contrainte: $\|n\| = 1$
- On pose:
  - Barycentre des points: $G = \frac{1}{m}\sum_{i=1}^{m} X_i$
  - Matrice de covariance des points:

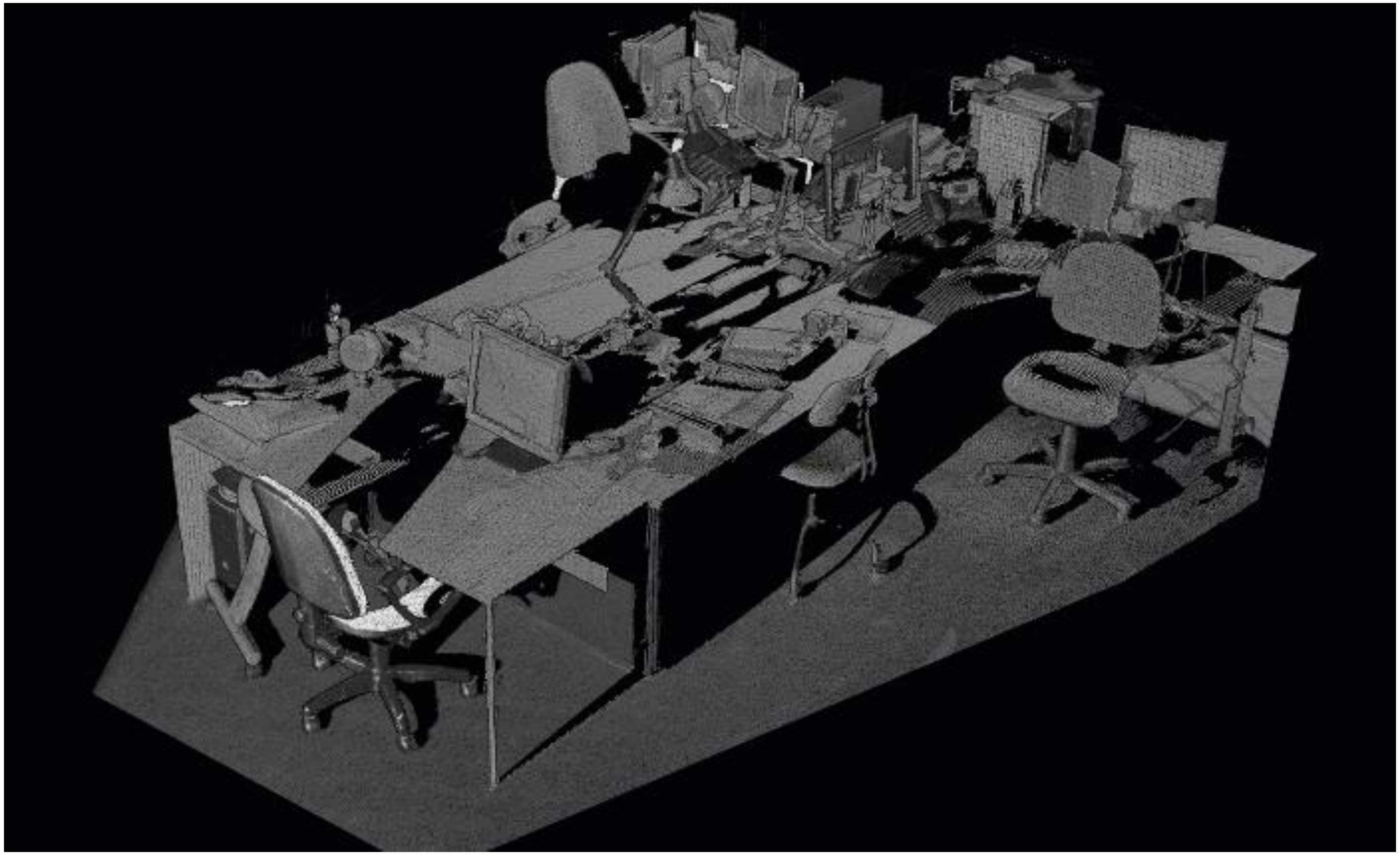
$$M_{cov} = \frac{1}{m}\sum_{i=1}^{m}(X_i - G)(X_i - G)'$$

- Résolution
  - Le meilleur plan approché est défini par:
    - Normale n_{min} :
      - Vecteur propre normé associé à la plus petite valeur propre de M_{cov}
      - NB: indéterminé à un changement de sens près
    - Distance d_{min} :
      - $d_{min} = n^t G$
  - La solution fait appel à l'analyse des directions principales de la matrice de covariance: « Analyse en Composantes Principales » (ACP)

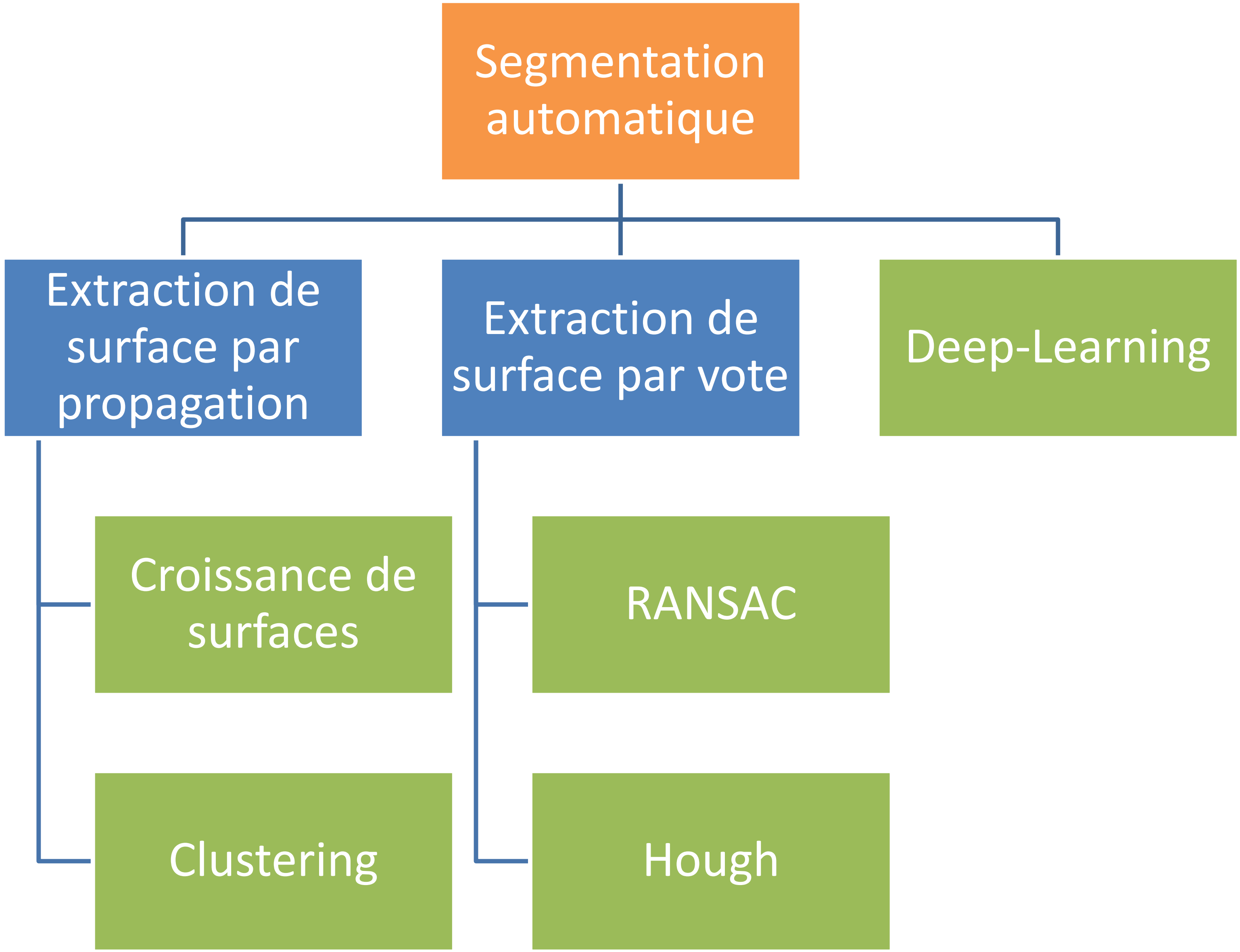
Partie II: Analyse de l’environnement 3D

I. Introduction
II. Analyse de surfaces et Reconstruction 3D
III. Segmentation 3D
IV. Recalage 3D

- Subdiviser (partitionner) le nuage de points 3D en sous-ensembles connexes correspondants à des modèles simples

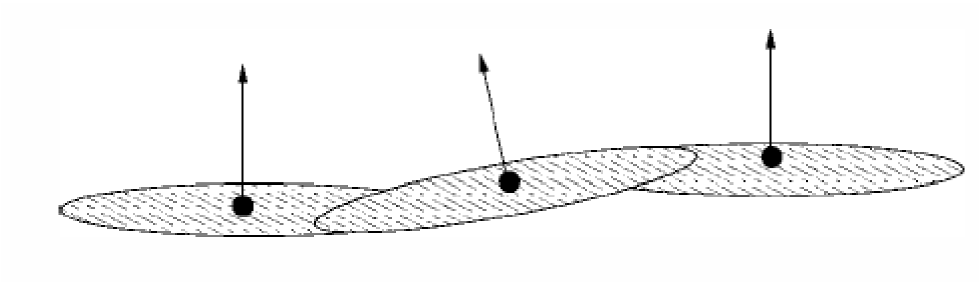


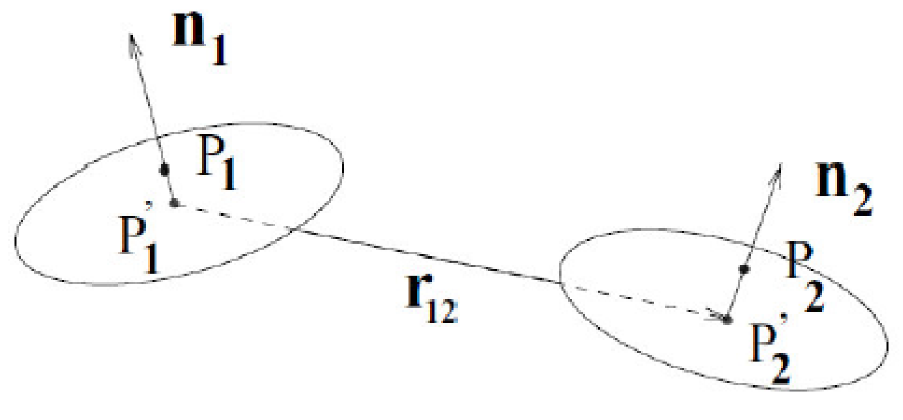
Méthodes de Segmentation



- Principe:
  - À partir de « surfaces germes » ou « graines » (seed surfaces) dans le nuage de point
  - Agrégation progressive des points voisins appartenant à la même surface
- Remarque:
  - Extension de l'algorithme « croissance de régions » pour les images

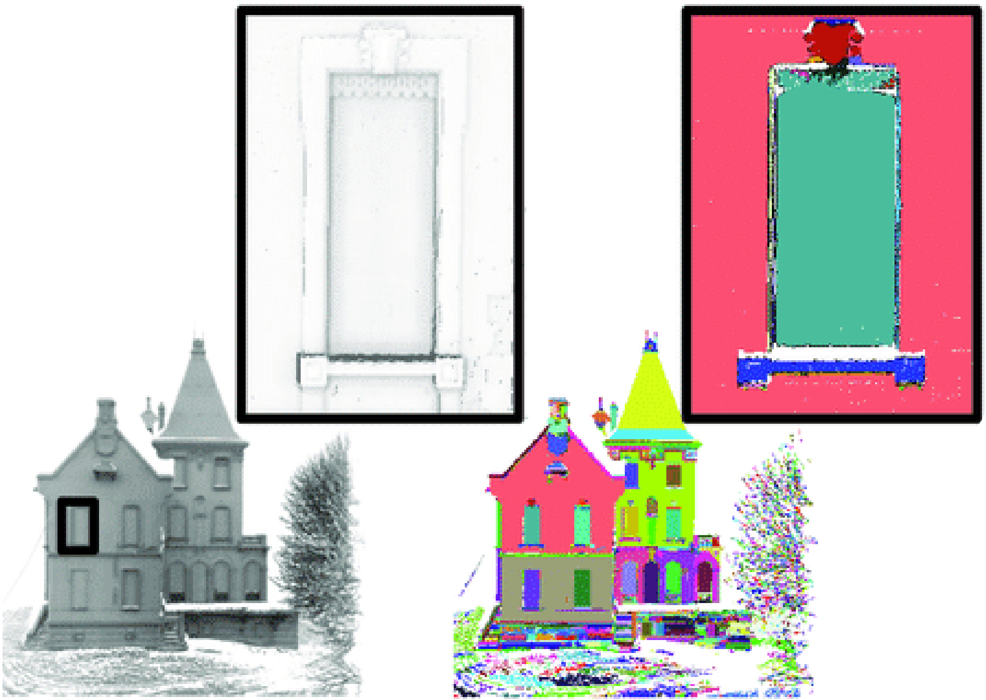
- Pour chaque point, calcul de la normale du plan dans un voisinage:

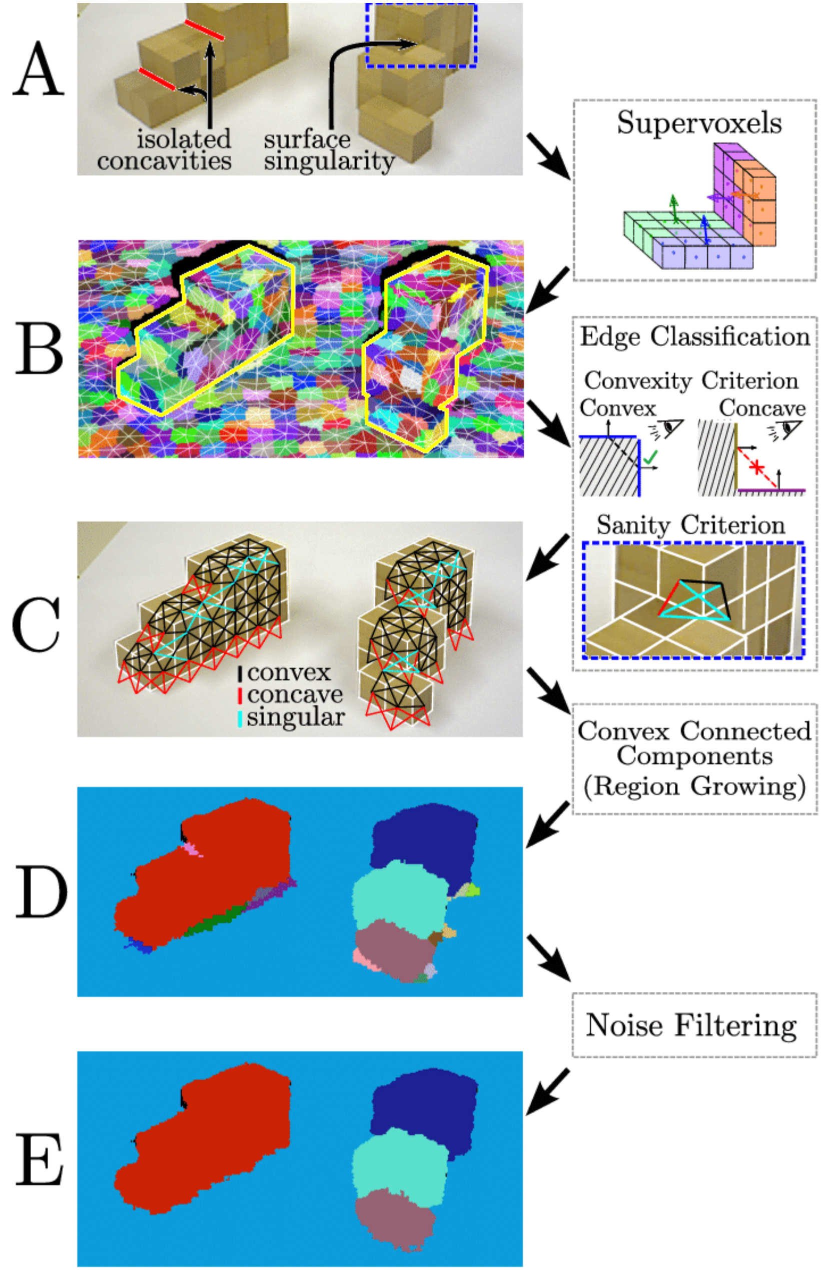




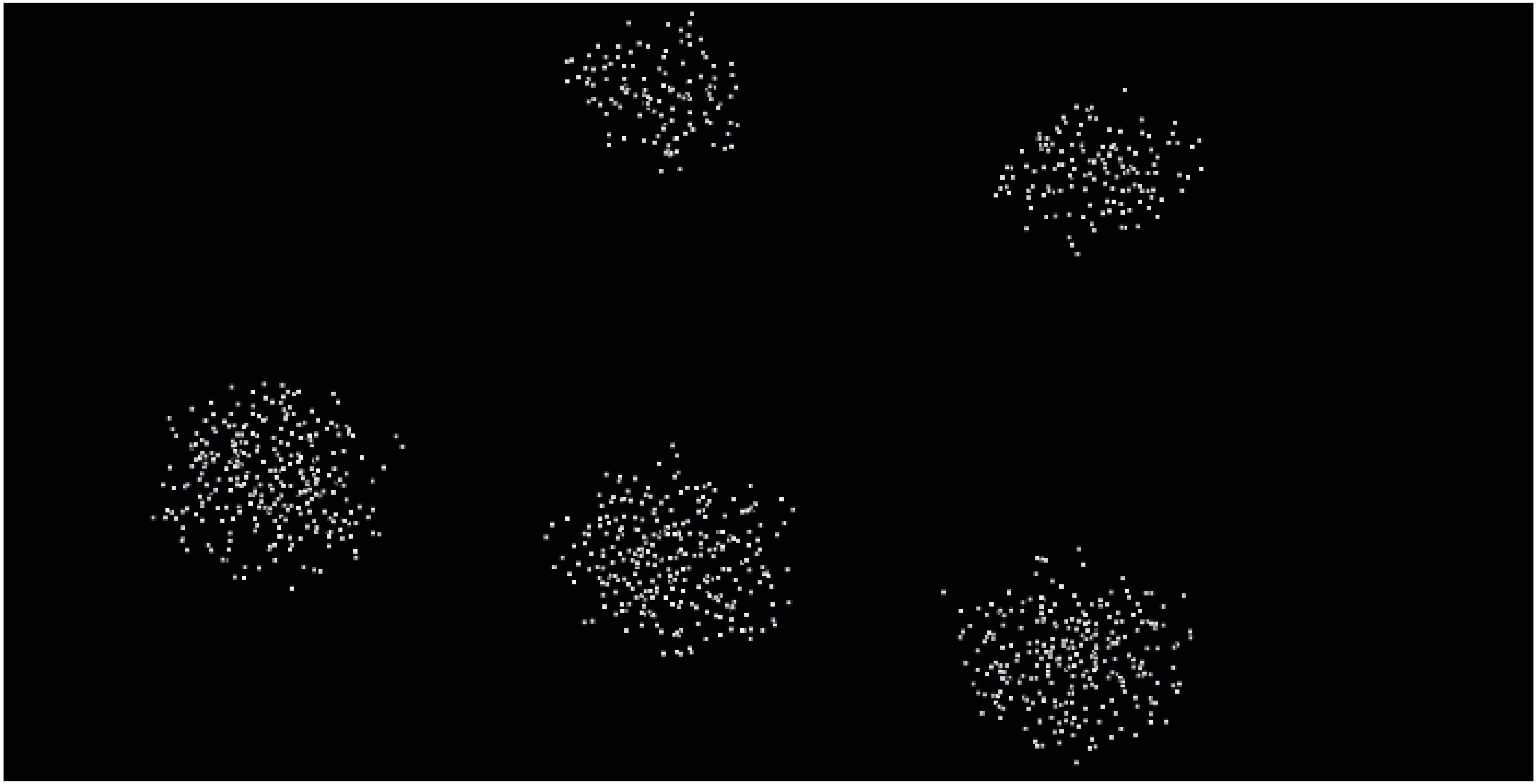
- Critère d'agrégation:
  - Co-normalité: $\alpha = \arccos(n_1 . n_2) \leq \alpha_{seuil}$
  - Coplanarité: $d = \max(|r_{12}.n_1|, |r_{12}.n_2|) \leq d_{seuil}$

Croissance de surfaces



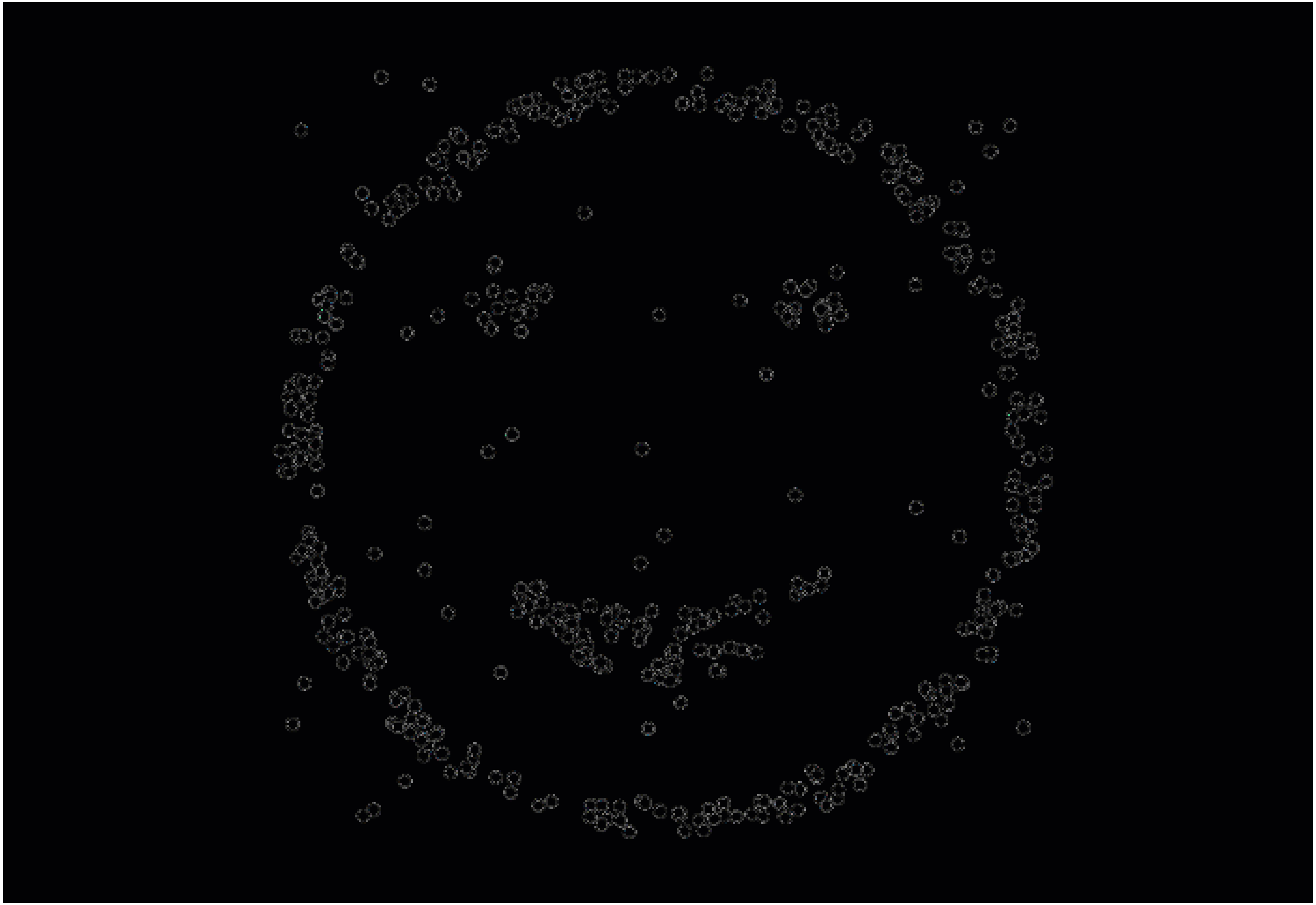


- Regroupemment de données en paquets homogènes selon des caractéristiques communes.



- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est l'un des algorithmes de clustering les plus répandus.
- Principe:
  - Choix d'un point graine pour la région,
  - Identification des voisins du point,
  - Pour chaque point voisin,
    - Si la densité locale de points est suffisante, ajout à la région,
    - Sinon, labellisation en tant que bruit,
  - On continue jusqu'à ne plus pouvoir étendre la région.
- On peut jouer sur deux paramètres:
  - Le rayon de recherche des voisins,
  - La quantité minimale de voisins ou densité.

DBSCAN



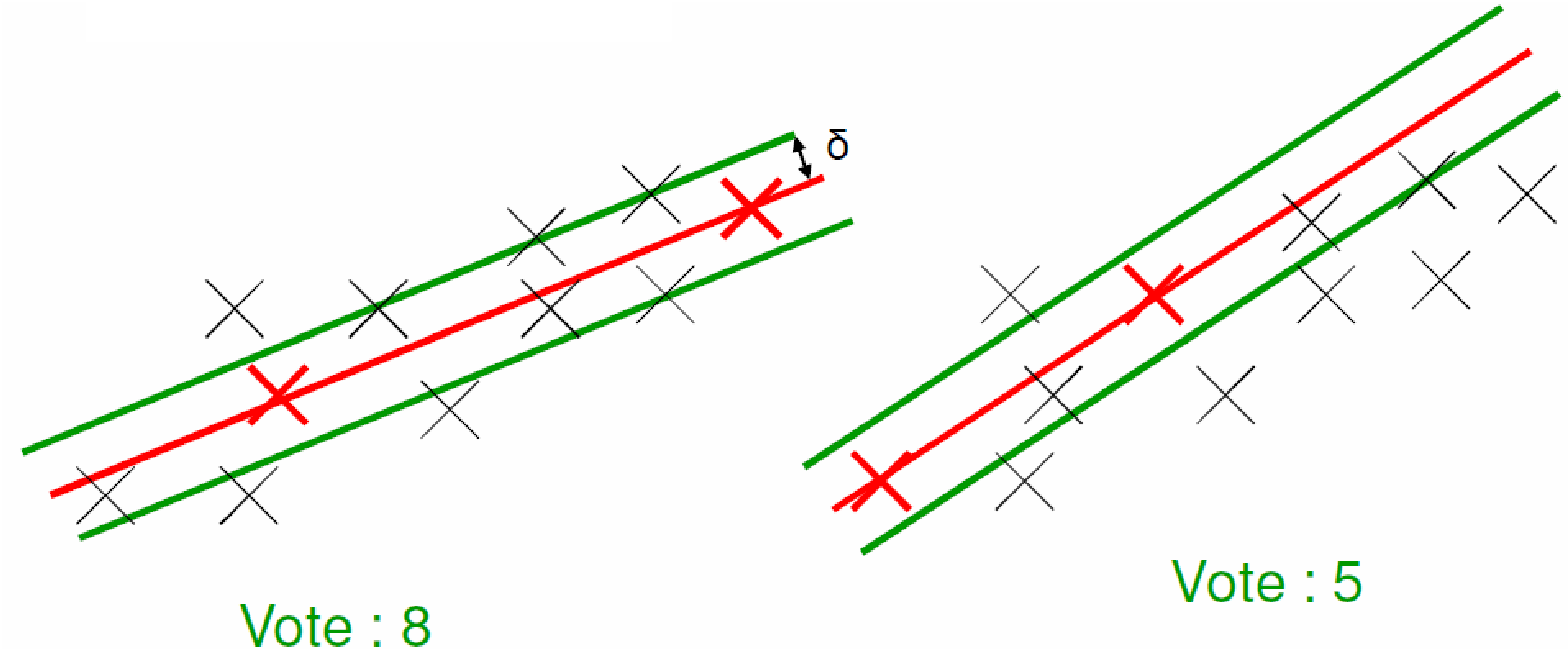
- RANdom SAmple Consensus:
  - Méthode de vote sur des échantillons aléatoires de surfaces
    - Échantillons calculés à partir du nombre minimal de points nécessaires pour définir la surface (quorum)
    - Vote: nombre de points du nuage englobés dans un espace entourant chaque surface
    - Le nombre de votes est décidé en fonction d’un calcul de probabilités

- Très efficace pour les grandes surfaces en nombre inconnu dans un grand nuage de points
- Les surfaces extraites ne sont pas connectées entre elles

- Primitives géométriques et quorum de points:
  - Droite:
    - Quorum = 2 points non alignés
  - Plan:
    - Quorum = 3 points x_i non alignés
    - Le plan est défini par:
      - L'un des points, par exemple x_1
      - Une normale unitaire n, par exemple définie par:

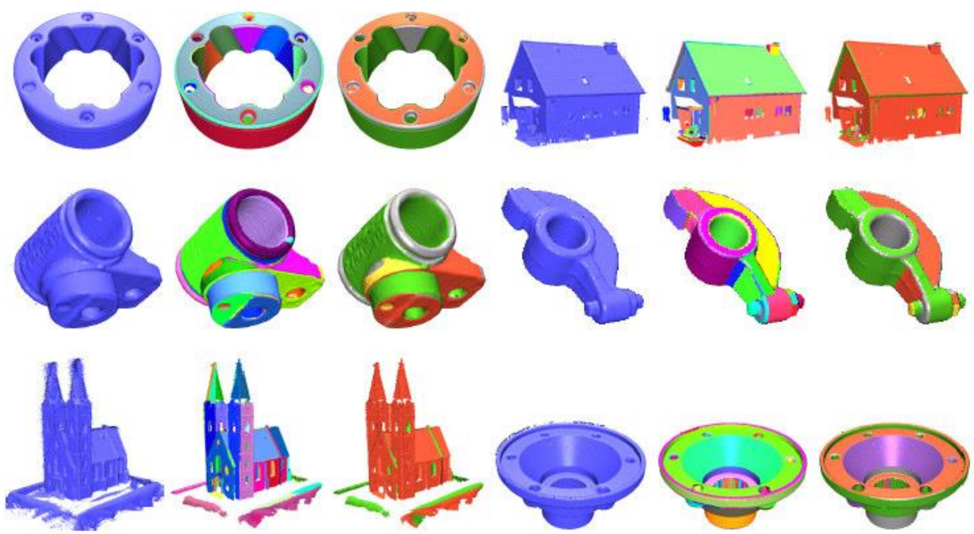
$$n = \frac{(x_2 - x_1)(x_3 - x_1)}{\|(x_2 - x_1)(x_3 - x_1)\|}$$

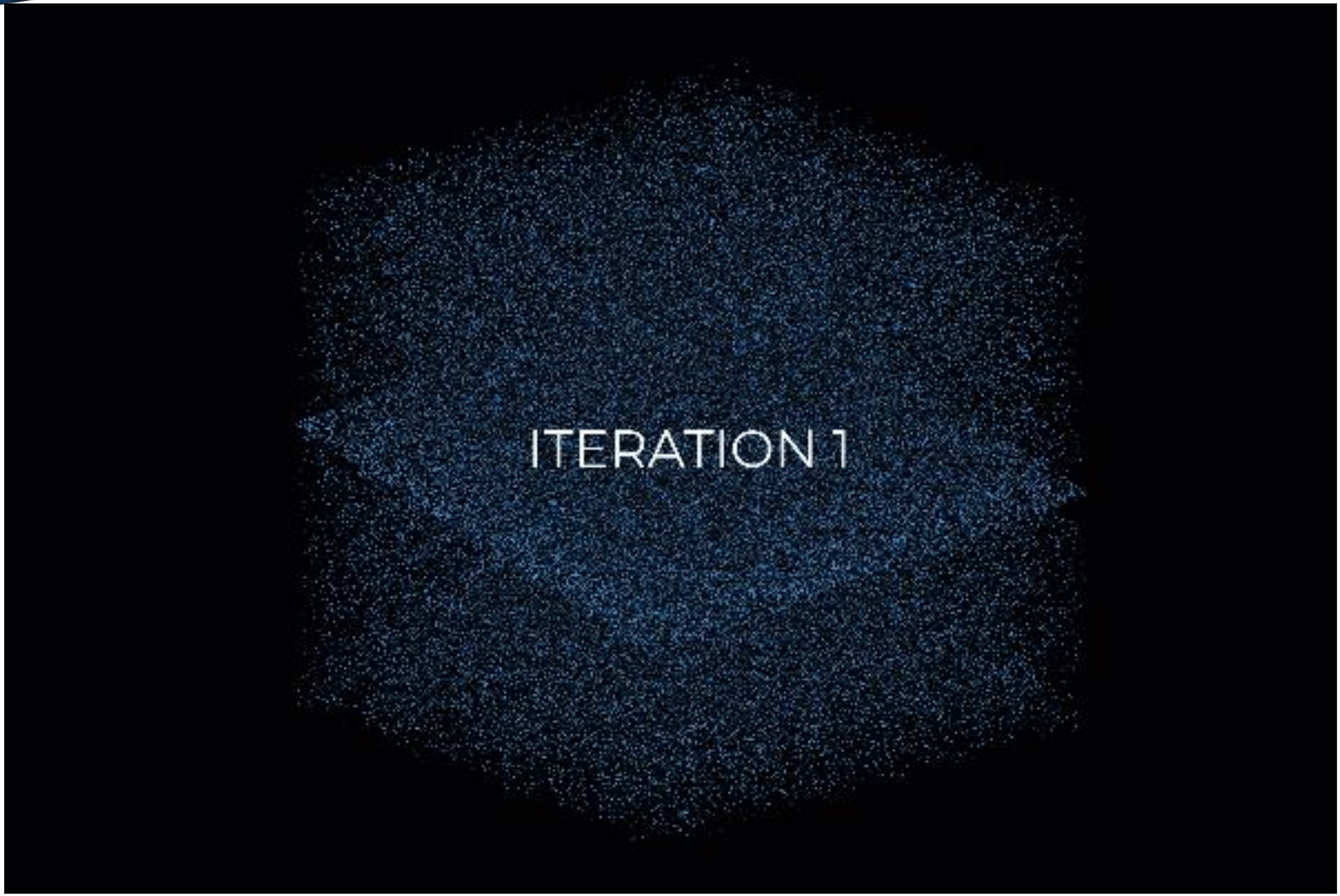
- Vote:
  - Nombre de points compris dans un espace à une certaine distance δ de la surface calculée



- Probabilités:
  - Hypothèses:
    - Plusieurs surfaces possibles, points non bruités
    - N points dans le nuage de points
    - n points appartiennent à la surface recherchée
    - q points pour définir une surface (quorum)
  - Probabilité de trouver la surface recherchée:
    - Avec 1 tirage aléatoire: $p = \left(\frac{n}{N}\right)^q$
    - Avec T tirages aléatoires: $p = 1 - \left(1 - \left(\frac{n}{N}\right)^q\right)^T$

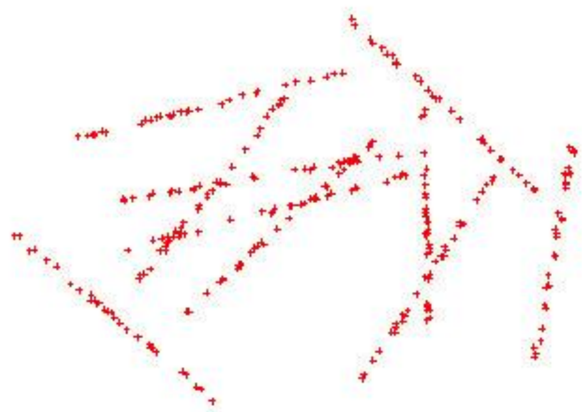
- Nombre de tirages nécessaires:
  - Nombre de tirages aléatoires T_{min} nécessaires pour avoir une probabilité p_t de trouver une surface d'au moins n_{min} points: $T_{min} = \frac{\log(1-p_t)}{\log\left(1-\left(\frac{n_{min}}{N}\right)^q\right)}$
  - En supposant $n_{min} \ll N$: $T_{min} \approx \log\left(\frac{1}{1-p_t}\right)\left(\frac{N}{n_{min}}\right)^q$
- Exemple:



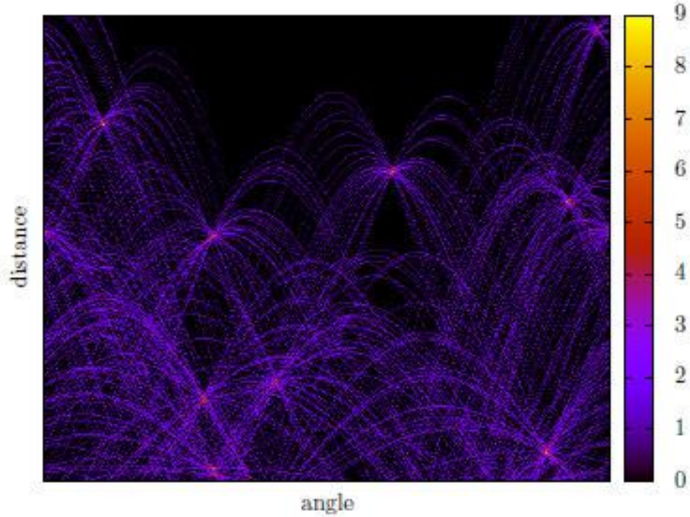

ITERATION 1

- Principe:
  - Méthode de vote dans l'espace discrétisé des paramètres
    - Chaque point du nuage génère des votes de surfaces possibles dans l'espace des paramètres
    - La cellule qui remporte le plus de votes est retenue comme surface
  - Généralisation de la transformée de Hough 2D (image)

- Détection de droites

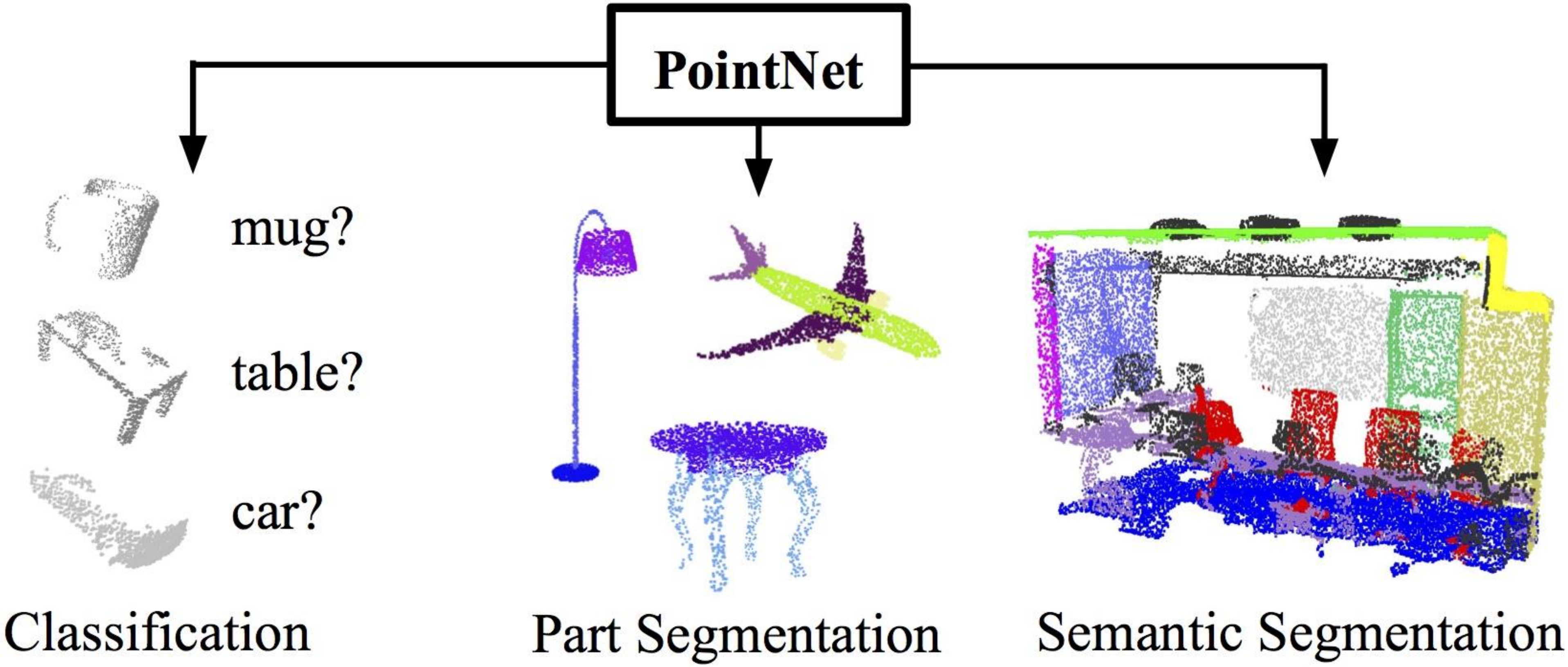


(a) *Nuage de points échantillonnant 10 droites*



(b) *Votes dans l'espace paramétrique des droites*

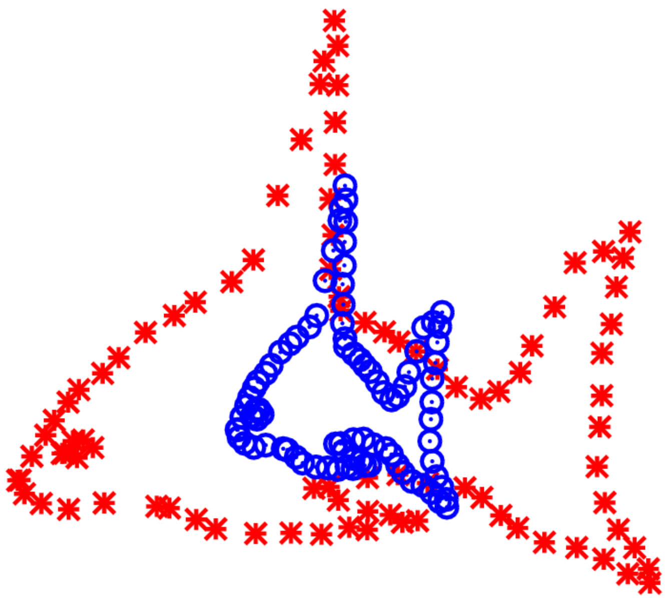
- Problème:
  - Grande dimension de l'espace des paramètres pour les formes complexes
  - Nécessité de simplification: utilisation des normales, etc.


PointNet
mug?
table?
car?
Classification
Part Segmentation
Semantic Segmentation

Partie II: Analyse de l'environnement 3D

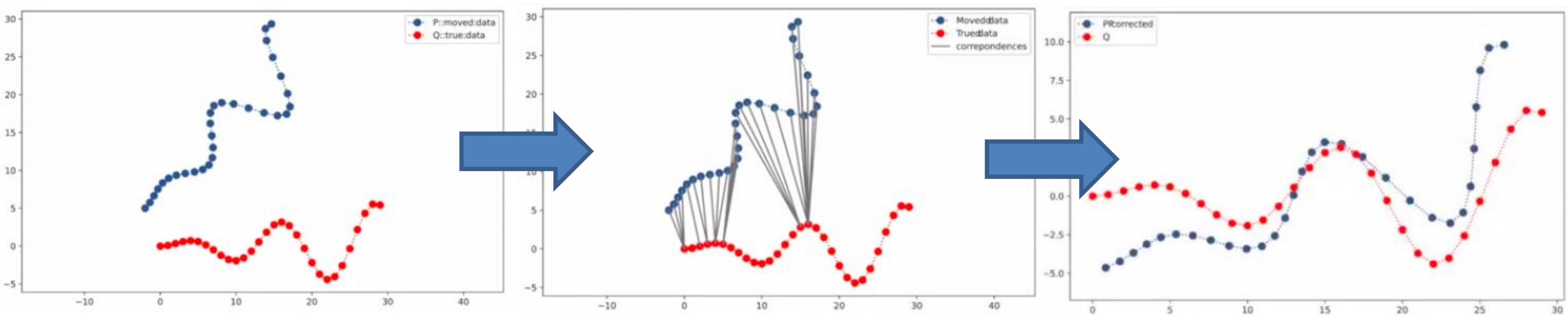
I. Introduction
II. Analyse de surfaces et Reconstruction 3D
III. Segmentation 3D
IV. Recalage 3D

- Point Set Registration, Point Matching:
  - Processus d'alignement de deux jeux de points (2D ou 3D)
  - Recherche de la transformation permettant de projeter les points p_i' du nuage P' sur les point p_i du nuage P.



Iterative Closest Point

- L'algorithme ICP:
  - Détermination de la transformation rigide (R, t) entre les deux nuages de points
  - Principe:
    - Appariement des points du nuage à recaler au point le plus proche dans l'autre nuage (approche de « nearest neighbor »)
    - Calcul de la transformation (R, t) qui minimise la distance entre ces points



Calcul de la transformation (R, t)

- Résolution par la méthode des moindres carrés:
  - On calcule:
    - $f(R, t) = \sum_{i=1}^{n} \|p_i - (R.p_i' + t)\|^2$
    - $f: \begin{cases} SE^3 \rightarrow \mathfrak{R}^+ \\ (R, t) \longmapsto f(R, t) \end{cases}$
  - On cherche:
    - $(R, t)\ t.q.\ (R, t) = argmin_{R,t} f(R, t)$
  - Solution par decomposition en valeurs singulières (SVD):
    - Au minimum de f, s'il existe, on a:

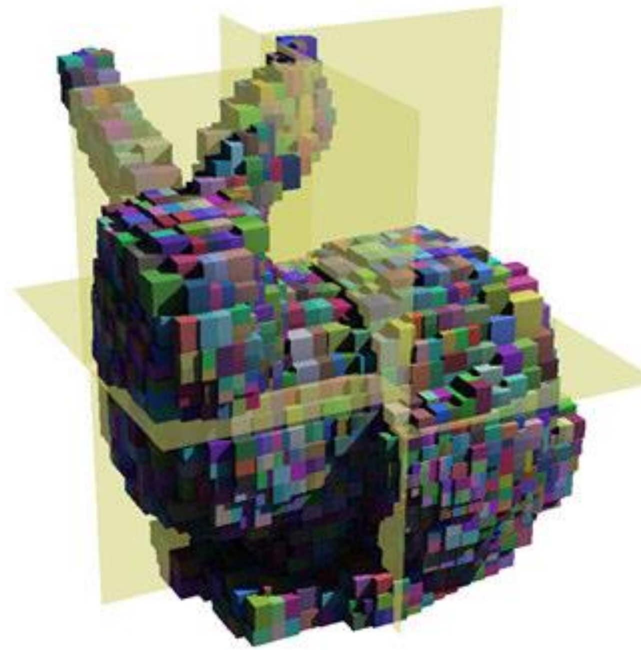
$$\nabla f = 0 \Rightarrow \begin{cases} \dfrac{\partial f}{\partial R} = 0 \\ \dfrac{\partial f}{\partial t} = 0 \end{cases}$$

- Résolution par SVD:
  - Entrée: Jeux de points (P, P')
  - Sortie: Matrice de rotation R, vecteur translation t
  - Algorithme:
    - Déterminer les barycentres $p_m = \frac{1}{n}\sum_{i=1}^{n} p_i$ et $p'_m = \frac{1}{n}\sum_{i=1}^{n} p'_i$
    - Calculer la matrice: $H = \sum_{i=1}^{n} q'_i . q_i^T$ avec $\forall i \in \{i, n\}, \begin{cases} q_i = p_i - p_m \\ q'_i = p'_i - p'_m \end{cases}$
    - Décomposer H en valeurs singulières: $$\exists (U, V, \Sigma) \in M_3(\mathfrak{R})^3 \ t.q. H = U\Sigma V^T$$
    - Calculer $R = VU^T$ et $t = p_m - Rp'_m$

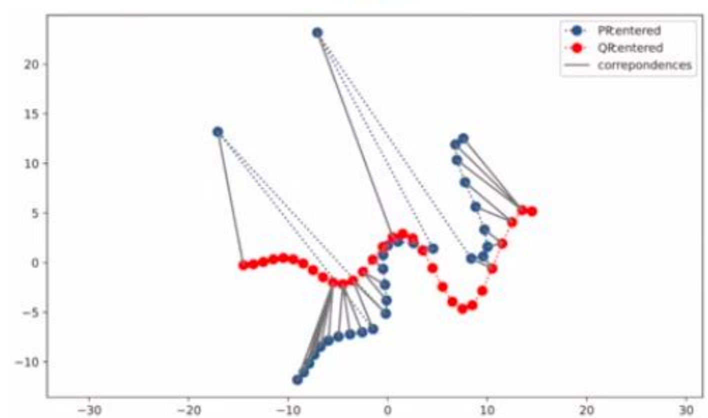
- Pseudo code ICP:
  - Recalage approximatif $P' \rightarrow P$
  - Répéter:
    - Association de données $P' \rightarrow P$
    - Calcul de la transformation (R, t)
    - Application de la transformation au nuage P'
    - Calcul de la distance entre les nuages
  - Tant que:
    - Distance normalisée > seuil
    - Et nombre d'itération < maximum itérations

- Temps de calcul:
  - Appariement en $O(n_1 n_2)$, le reste en $O(n_1 + n_2)$
  - Acceptable pour les petits nuages de points, trop lent pour de gros nuages de points
  - Nécessité de sous-échantilloner
  - Possibilité d'accelération avec ANN (Approximate Nearest Neighbor) : $O(n_1 \log(n_2))$

- Approximate Nearest Neighbor (ANN):
  - Principe:
    - Pré-calcul d'un kd-tree pour partitionnner l'espace
    - Recherche dichotomique avec distance seuil



- Variante ICP:
  - Métrique point à plan (point to plane)
  - Échantillonage: régulier, aléatoire, basé sur les normales…
  - Réjection des outliers
  - …



Merci pour votre attention