Nuages de Points et Modélisation 3D

4 - Machine learning I

Overview

Machine learning courses

- Surface reconstruction

- Descriptors and machine learning
- Image based processing

Today

- Geometric deep learning

ML course 2

- Convolutional and Transformer based architectures

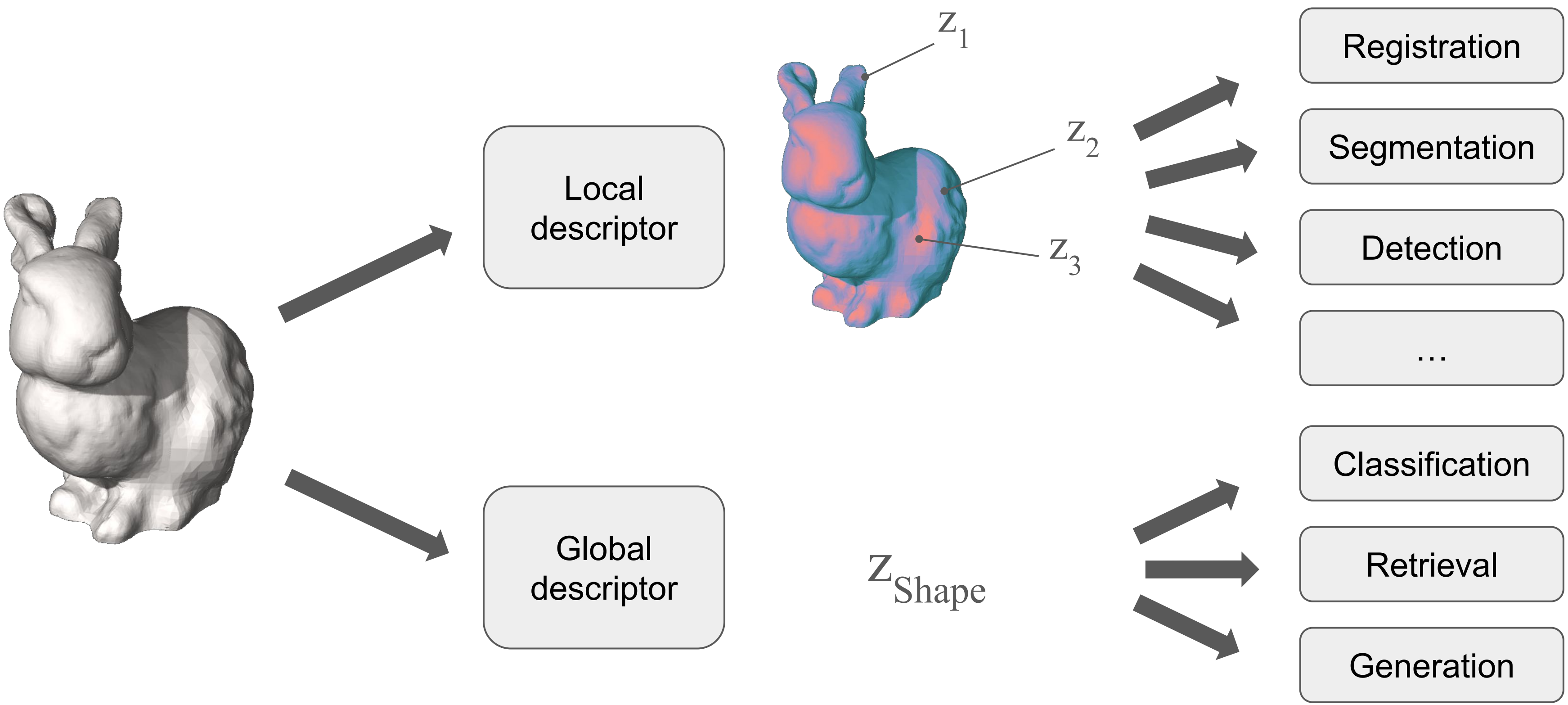
ML course 3

- Tasks and corresponding architectures

ML course 4

I - Descriptors and machine learning

Descriptors



I - Descriptors and machine learning

A - Local descriptors

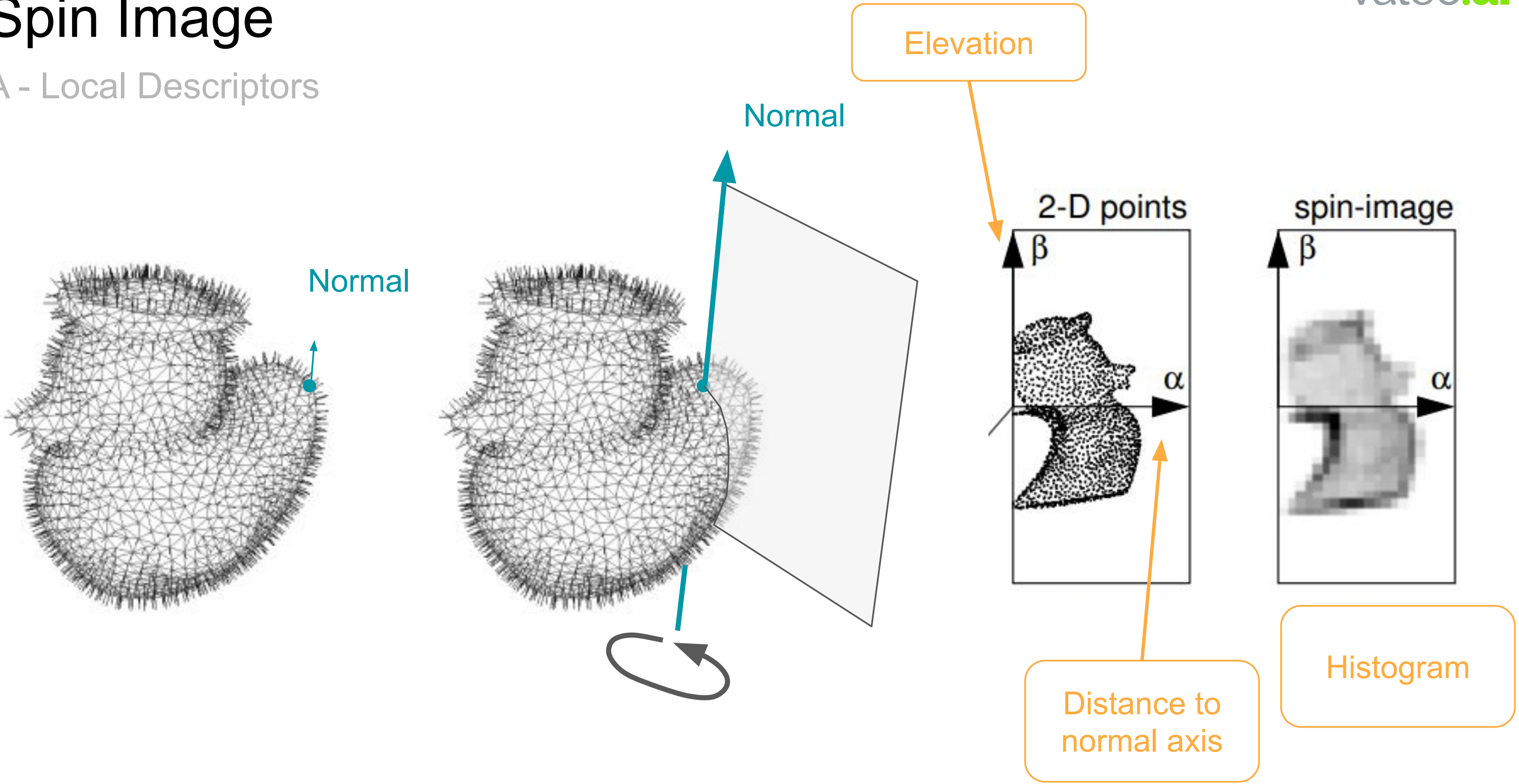
Spin Image

A - Local Descriptors

Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes, A. E. Johnson and M. Hebert, 1999

Spin Image

A - Local Descriptors



Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes, A. E. Johnson and M. Hebert, 1999

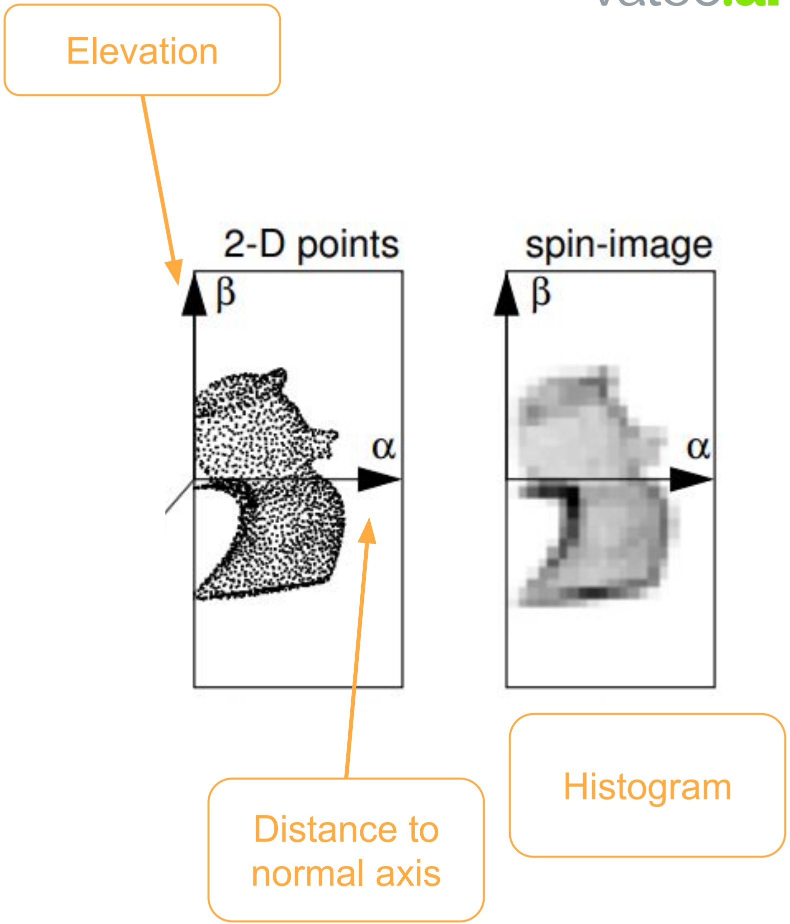
Spin Image

A - Local Descriptors

Given normal **n** at point **p**

For each point of interest:

- Compute alpha and beta:
  beta=<n, q-p>
  alpha = ||(q-p) - beta n||
- Accumulte in the histogram image



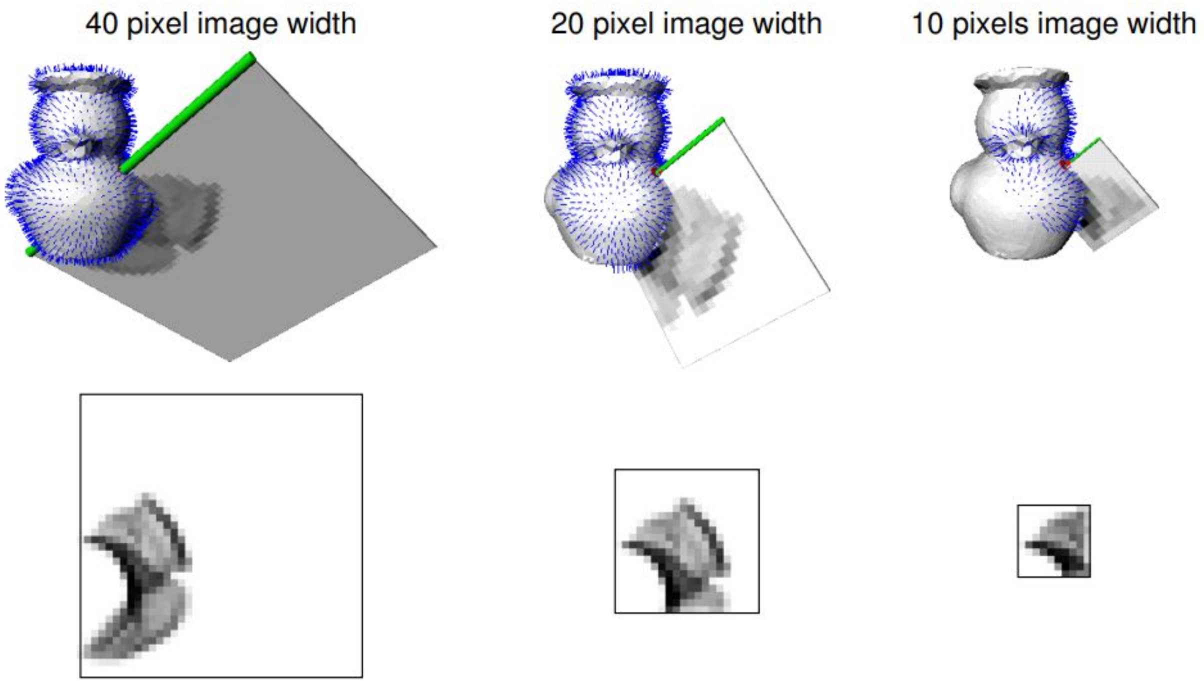
Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes, A. E. Johnson and M. Hebert, 1999

Spin Image

A - Local Descriptors

Influence of the accumulator size

- Defines the size of the observed neighborhood
- More or less context



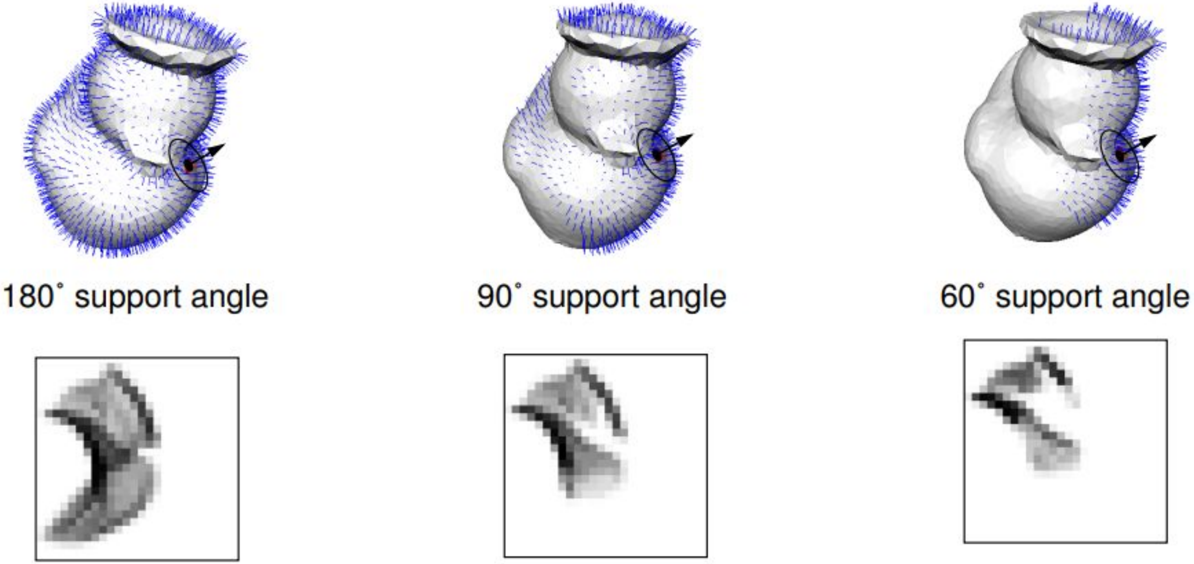
Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes, A. E. Johnson and M. Hebert, 1999

Spin Image

A - Local Descriptors

Filter on the normal angle

- Retain meaningfull points

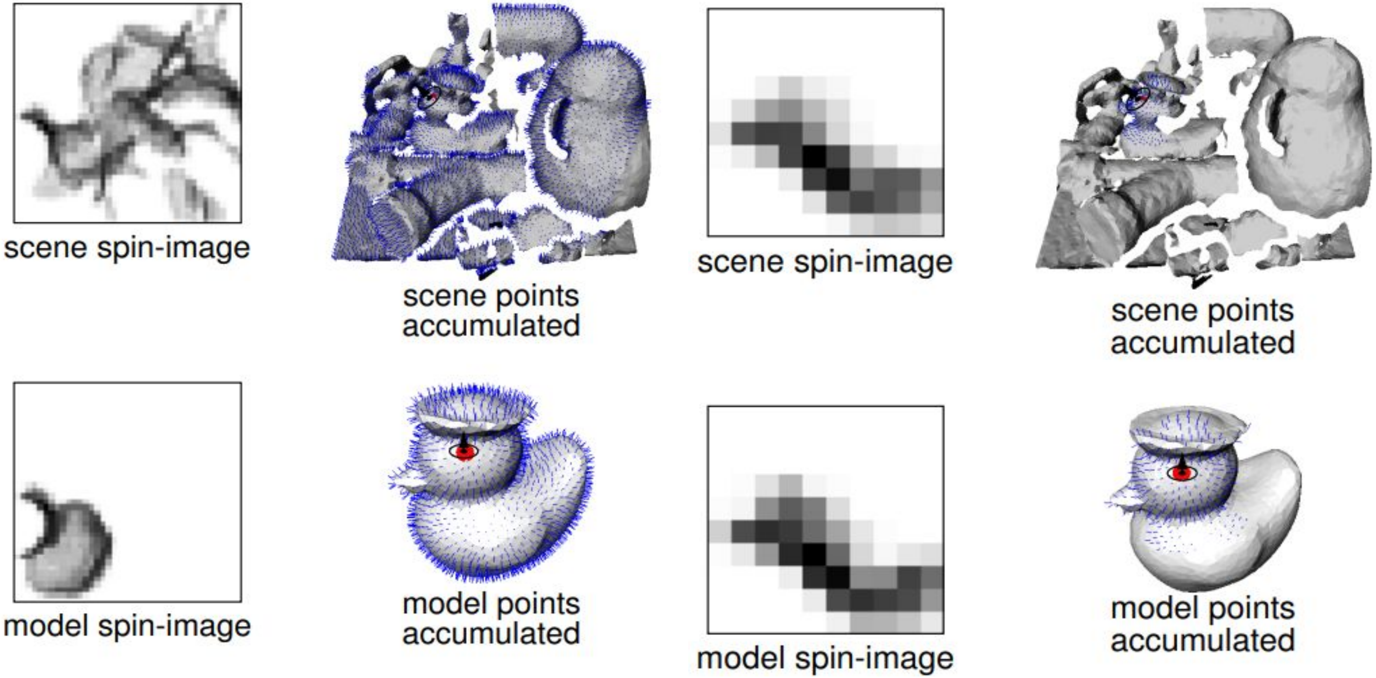


Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes, A. E. Johnson and M. Hebert, 1999

Spin Image

A - Local Descriptors

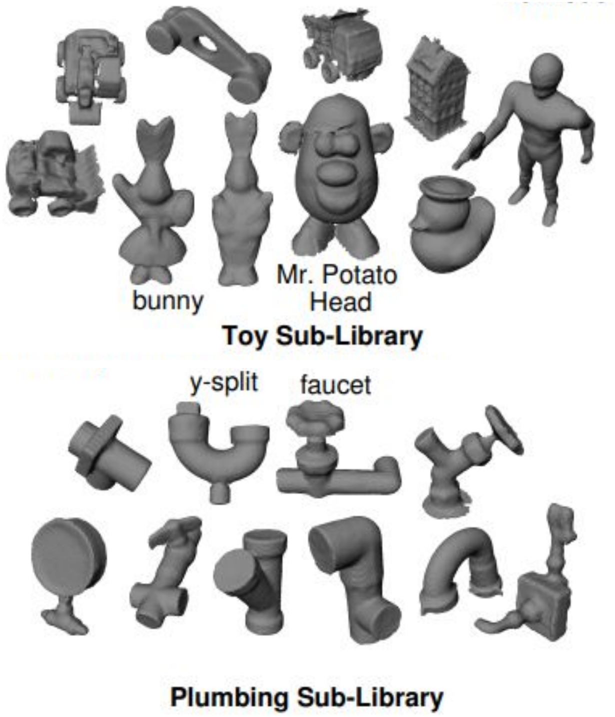
Influence of histogram for descriptor matching



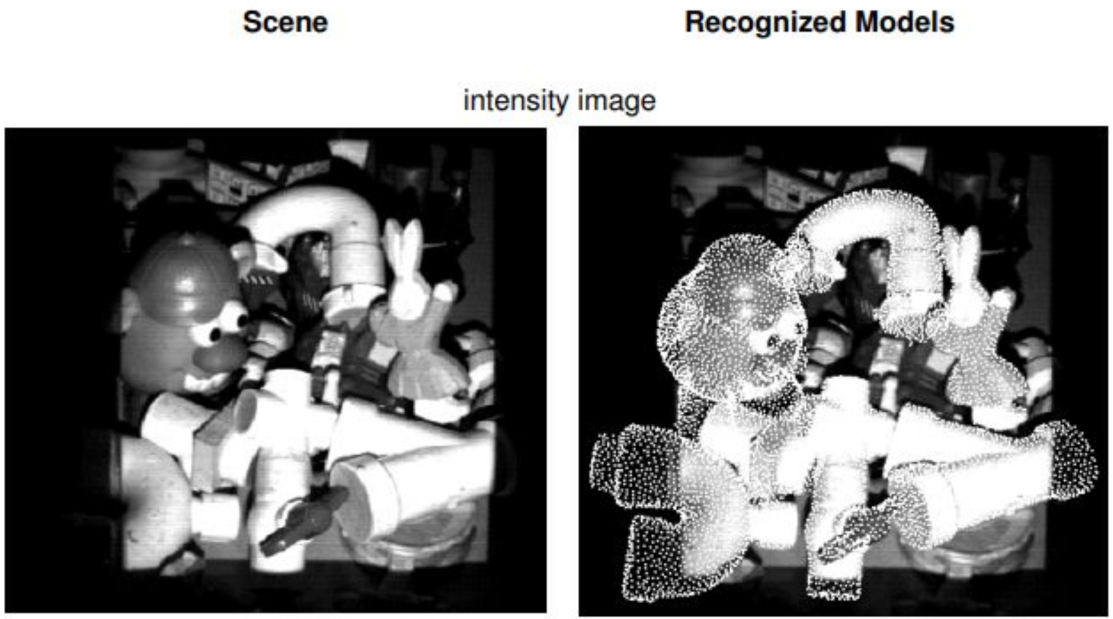
Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes, A. E. Johnson and M. Hebert, 1999

Spin Image

A - Local Descriptors



Object retrieval in 3D point clouds



Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes, A. E. Johnson and M. Hebert, 1999

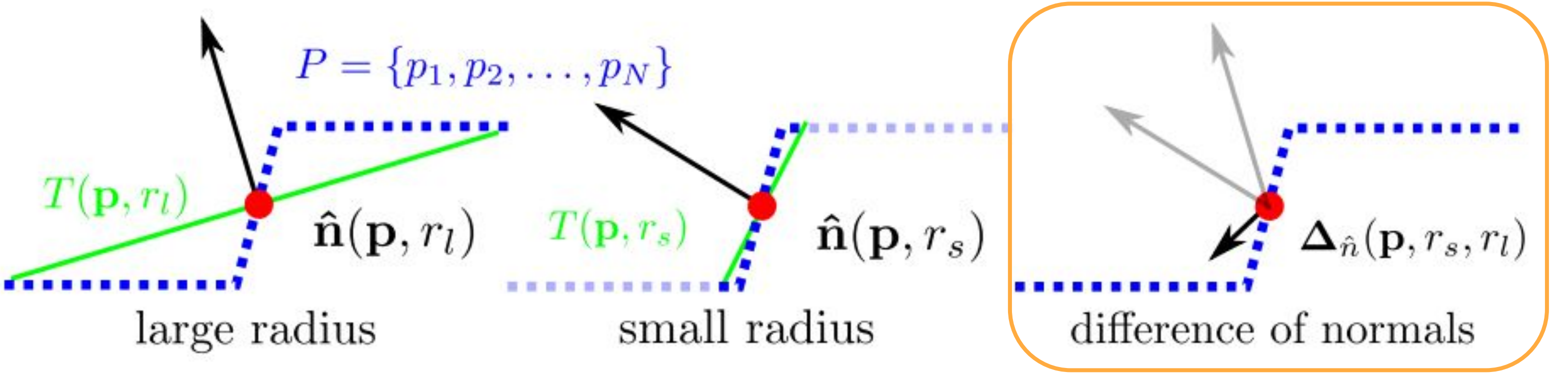
DoN - Difference of Normals

A - Local Descriptors

Observation:

Normals estimated with radius vary with **r** and the smoothness of the surface

⇒ use this variation a descriptor



Difference of Normals as a Multi-Scale Operator in Unorganized Point Clouds, Yani Ioannou, Babak Taati, Robin Harrap, Michael Greenspan 3DIMPVT

DoN - Difference of Normals

A - Local Descriptors

More formally:

$$\Delta_{\hat{\mathbf{n}}}(\mathbf{p}, r_1, r_2) = \frac{\hat{\mathbf{n}}(\mathbf{p}, r_1) - \hat{\mathbf{n}}(\mathbf{p}, r_2)}{2}$$

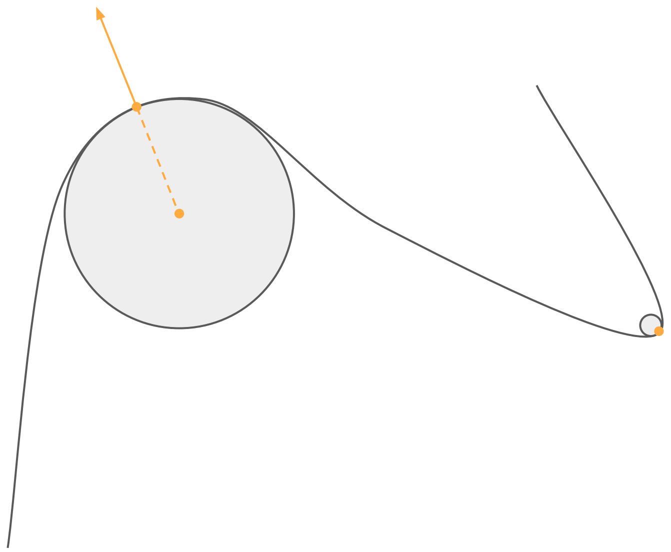
r1 an r2 are parameters.

More formally, it is an approximation of the **curvature** of the surface at given radius

Difference of Normals as a Multi-Scale Operator in Unorganized Point Clouds, Yani Ioannou, Babak Taati, Robin Harrap, Michael Greenspan 3DIMPVT

Curvature

A - Local Descriptors



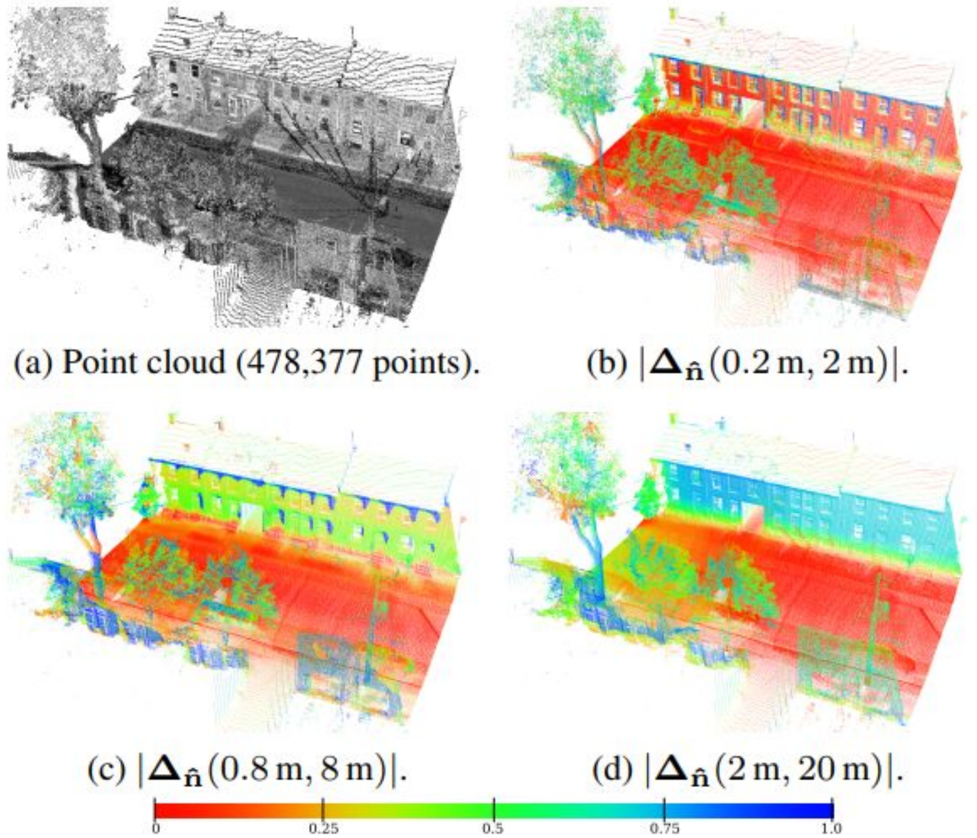
Curvature:

radius of maximal the sphere that can be fitted on the concave side.

DoN - Difference of Normals

A - Local Descriptors

Practical influence of the radius size



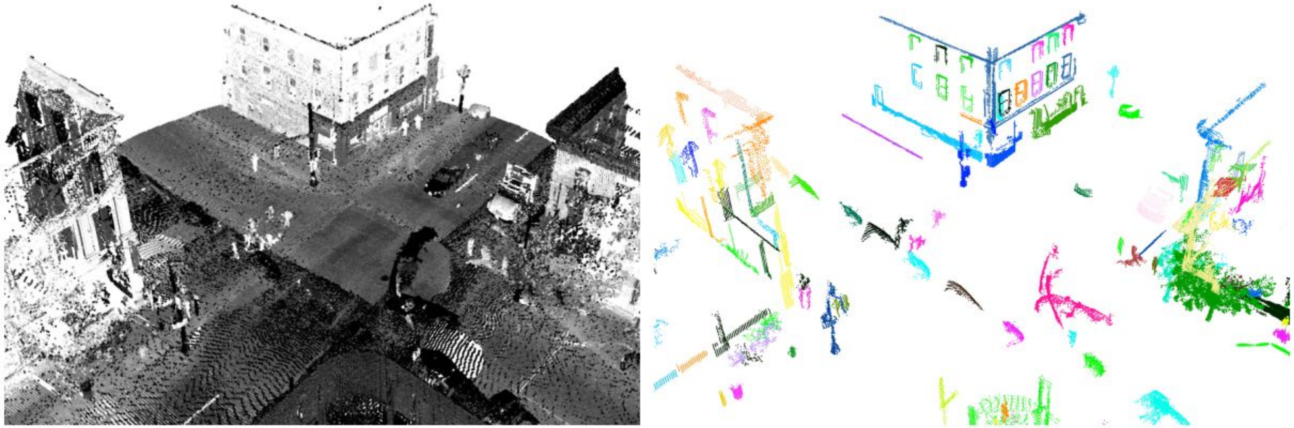
Difference of Normals as a Multi-Scale Operator in Unorganized Point Clouds, Yani Ioannou, Babak Taati, Robin Harrap, Michael Greenspan 3DIMPVT

DoN - Difference of Normals

A - Local Descriptors

Usage for unsupervised segmentation:

- Put a threshold on the norm of the DoN
- Apply **clustering**



(a) Original point cloud (620,820 points). (b) Clusters found in $|\mathbf{\Delta}_{\hat{\mathbf{n}}}(0.2\,\mathrm{m}, 2\,\mathrm{m})| \geq 0.25$.

(c) Person cluster. (d) Traffic light cluster. (e) Window cluster. (f) Car cluster. (g) Tree cluster.

Difference of Normals as a Multi-Scale Operator in Unorganized Point Clouds, Yani Ioannou, Babak Taati, Robin Harrap, Michael Greenspan 3DIMPVT

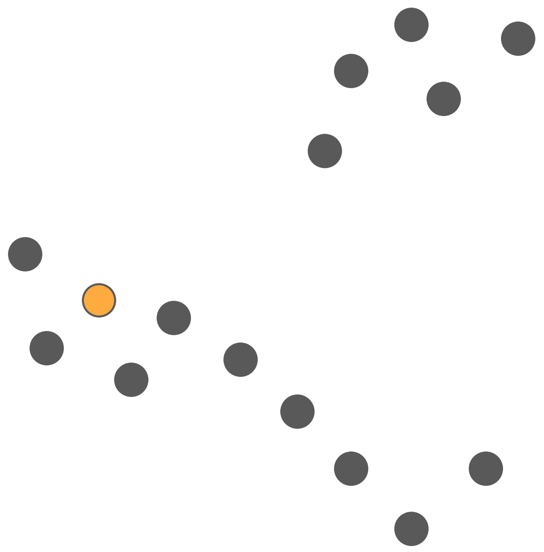
Apparte on clustering

Euclidean Cluster Extraction

Clustering

for every point, $\boldsymbol{p}_i \in P$ perform the following steps:

- If already visited, skip
- Else:
  - Initiate cluster
  - add $\boldsymbol{p}_i$ to the current queue Q;

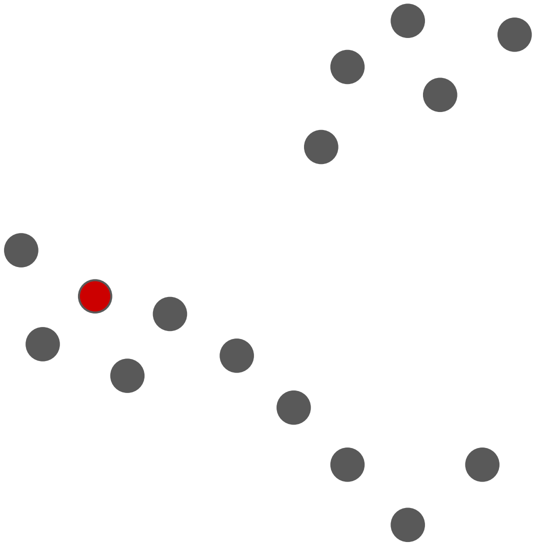


Euclidean Cluster Extraction

Clustering

for every point, $\boldsymbol{p}_i \in P$ perform the following steps:

- If already visited, skip
- Else:
  - Initiate cluster
  - add $\boldsymbol{p}_i$ to the current queue Q;
  - While Q is not empty:
    - Dequeue first element q
    - Add q to cluster C

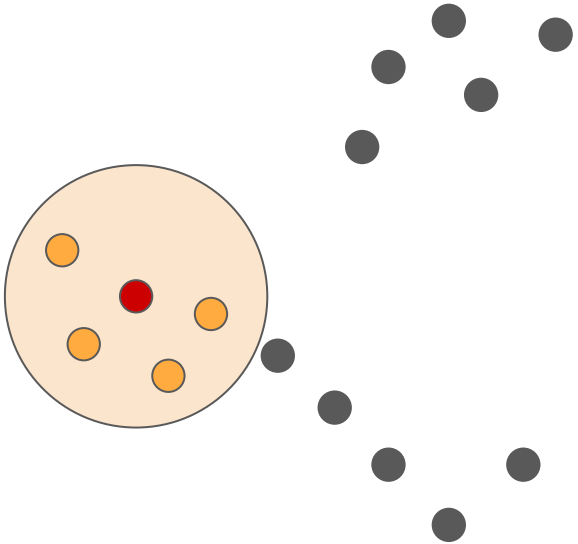


Euclidean Cluster Extraction

Clustering

for every point, $\boldsymbol{p}_i \in P$ perform the following steps:

- If already visited, skip
- Else:
  - Initiate cluster
  - add $\boldsymbol{p}_i$ to the current queue Q;
  - While Q is not empty:
    - Dequeue first element q
    - Add q to cluster C
    - search for the set neighbors of q in a sphere with radius r;
    - for every neighbor ,
      - if point not already processed
        - Add to queue Q



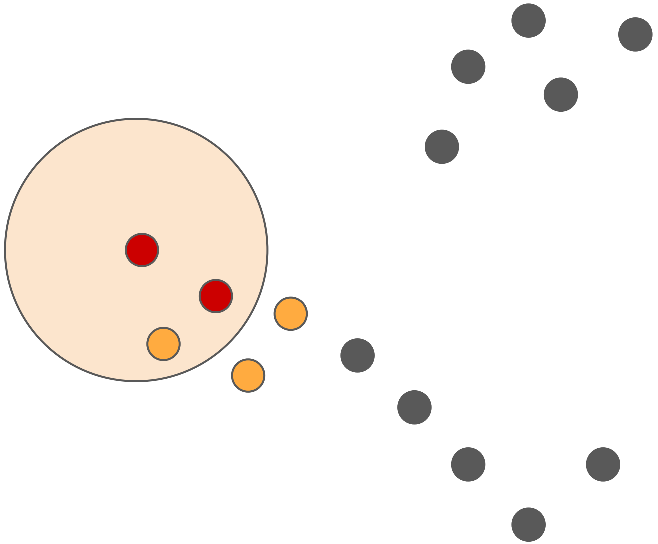
3D is here: Point Cloud Library (PCL), R. B. Rusu and S. Cousins, 2011

Euclidean Cluster Extraction

Clustering

for every point, $\boldsymbol{p}_i \in P$ perform the following steps:

- If already visited, skip
- Else:
  - Initiate cluster
  - add $\boldsymbol{p}_i$ to the current queue Q;
  - While Q is not empty:
    - Dequeue first element q
    - Add q to cluster C
    - search for the set neighbors of q in a sphere with radius r;
    - for every neighbor ,
      - if point not already processed
        - Add to queue Q



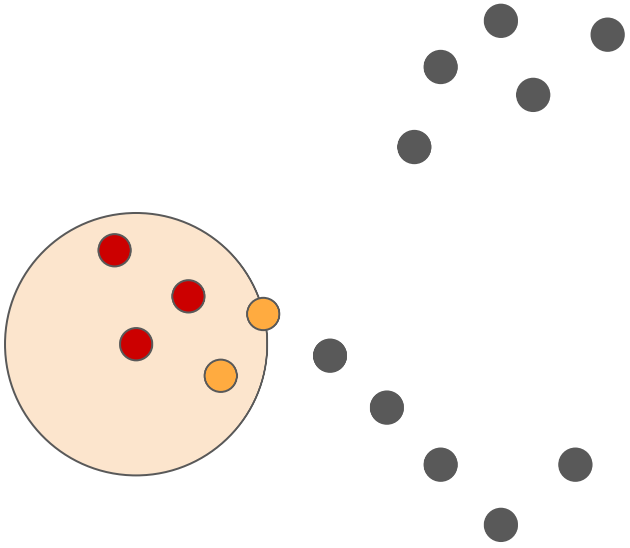
3D is here: Point Cloud Library (PCL), R. B. Rusu and S. Cousins, 2011

Euclidean Cluster Extraction

Clustering

for every point, $\boldsymbol{p}_i \in P$ perform the following steps:

- If already visited, skip
- Else:
  - Initiate cluster
  - add $\boldsymbol{p}_i$ to the current queue Q;
  - While Q is not empty:
    - Dequeue first element q
    - Add q to cluster C
    - search for the set neighbors of q in a sphere with radius r;
    - for every neighbor ,
      - if point not already processed
        - Add to queue Q



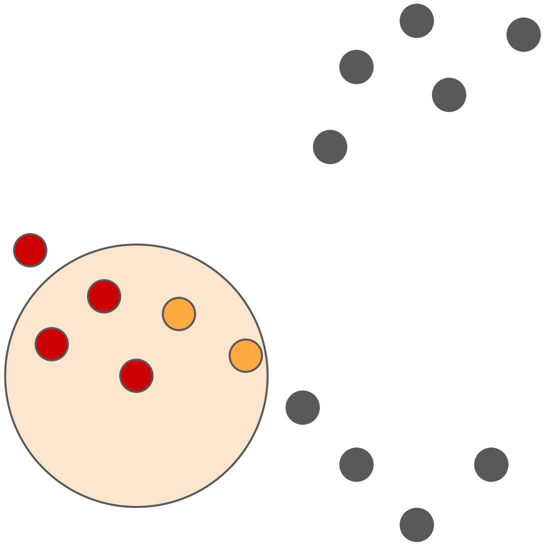
3D is here: Point Cloud Library (PCL), R. B. Rusu and S. Cousins, 2011

Euclidean Cluster Extraction

Clustering

for every point, $\boldsymbol{p}_i \in P$ perform the following steps:

- If already visited, skip
- Else:
  - Initiate cluster
  - add $\boldsymbol{p}_i$ to the current queue Q;
  - While Q is not empty:
    - Dequeue first element q
    - Add q to cluster C
    - search for the set neighbors of q in a sphere with radius r;
    - for every neighbor ,
      - if point not already processed
        - Add to queue Q



3D is here: Point Cloud Library (PCL), R. B. Rusu and S. Cousins, 2011

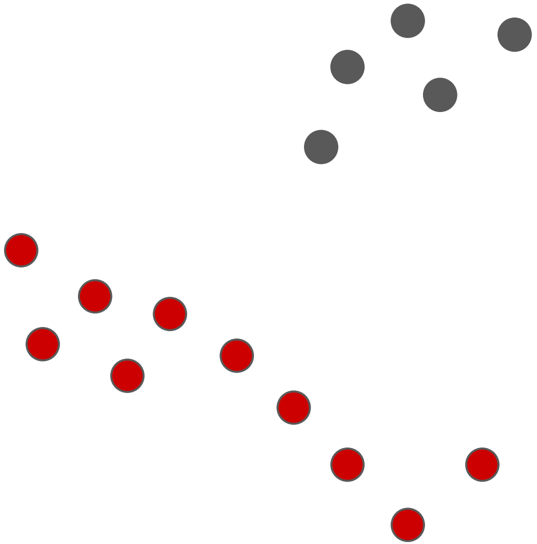
Euclidean Cluster Extraction

valeo.ai

Clustering

for every point, $\boldsymbol{p}_i \in P$ perform the following steps:

- If already visited, skip
- Else:
  - Initiate cluster
  - add $\boldsymbol{p}_i$ to the current queue Q;
  - While Q is not empty:
    - Dequeue first element q
    - Add q to cluster C
    - search for the set neighbors of q in a sphere with radius r;
    - for every neighbor ,
      - if point not already processed
        - Add to queue Q



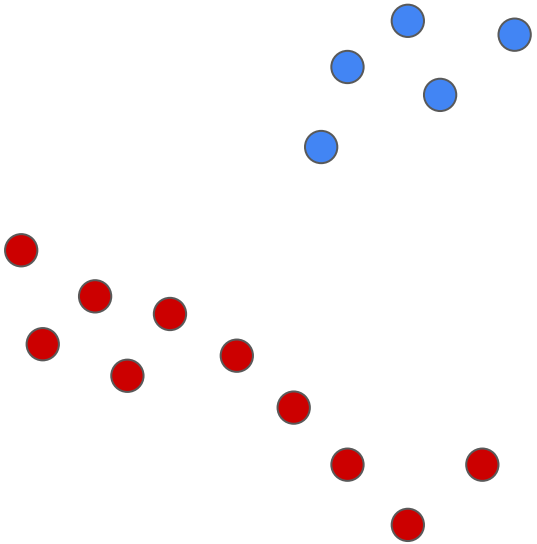
3D is here: Point Cloud Library (PCL), R. B. Rusu and S. Cousins, 2011

Euclidean Cluster Extraction

Clustering

for every point, $\boldsymbol{p}_i \in P$ perform the following steps:

- If already visited, skip
- Else:
  - Initiate cluster
  - add $\boldsymbol{p}_i$ to the current queue Q;
  - While Q is not empty:
    - Dequeue first element q
    - Add q to cluster C
    - search for the set neighbors of q in a sphere with radius r;
    - for every neighbor ,
      - if point not already processed
        - Add to queue Q



3D is here: Point Cloud Library (PCL), R. B. Rusu and S. Cousins, 2011

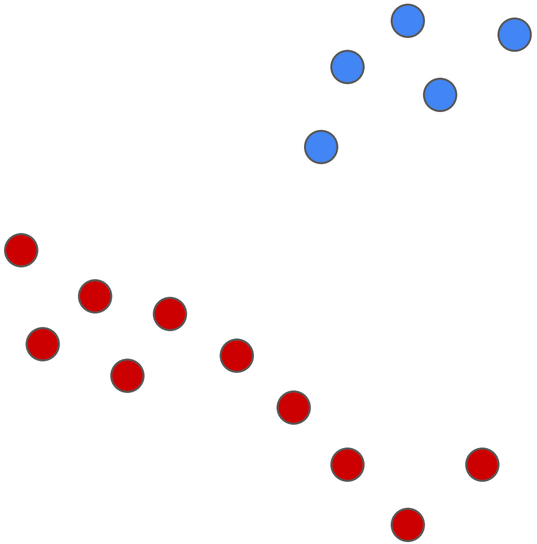
Euclidean Cluster Extraction

Clustering

It is a **region growing** algorithm in n dimensions.

Can be further extended:

- Planarity (restimate regression plane)
- Normal filtering
- More generally descriptor filtering



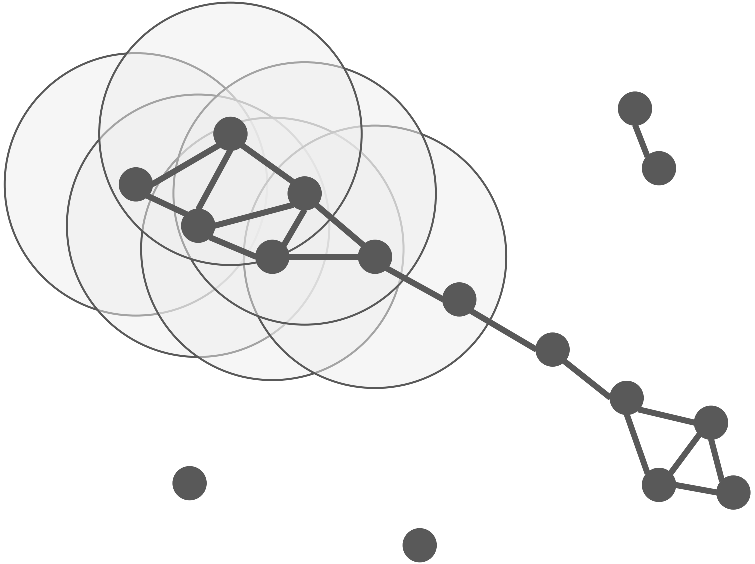
3D is here: Point Cloud Library (PCL), R. B. Rusu and S. Cousins, 2011

DBSCAN: Density-based spatial clustering of applications with noise

Clustering

Algorithm:

- Build the graph of for neighborhoods of radius r

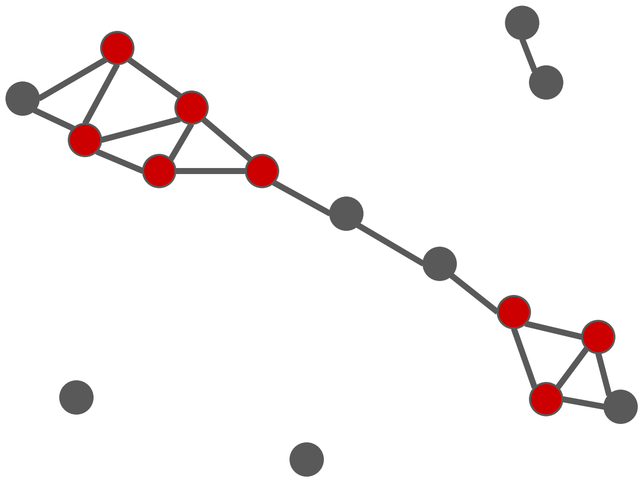


DBSCAN: Density-based spatial clustering of applications with noise

Clustering

Algorithm:

- Build the graph of for neighborhoods of radius r
- Search the core points: points that have minPts points in their neighborhood (including itself)



Density-based spatial clustering of applications with noise, M Ester, HP Kriegel, J Sander, X Xu, 1996

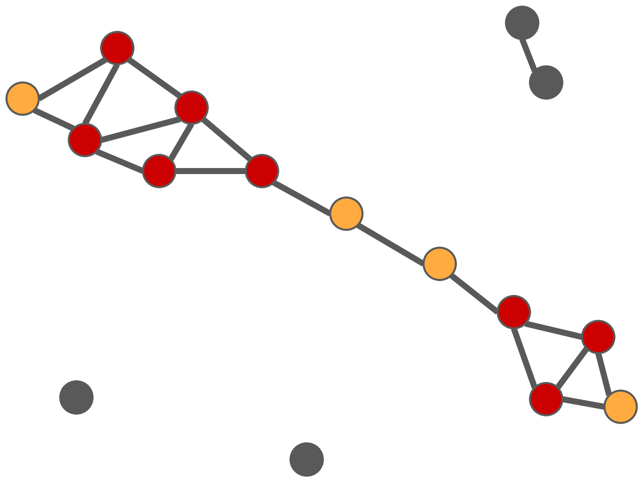
DBSCAN: Density-based spatial clustering of applications with noise

Clustering

Algorithm:

- Build the graph of for neighborhoods of radius r
- Search the core points: points that have minPts points in their neighborhood (including itself)
- Reachable point q if there is path $(p, p_1,...,p_n,q)$ where p_i are core points

minPts=4

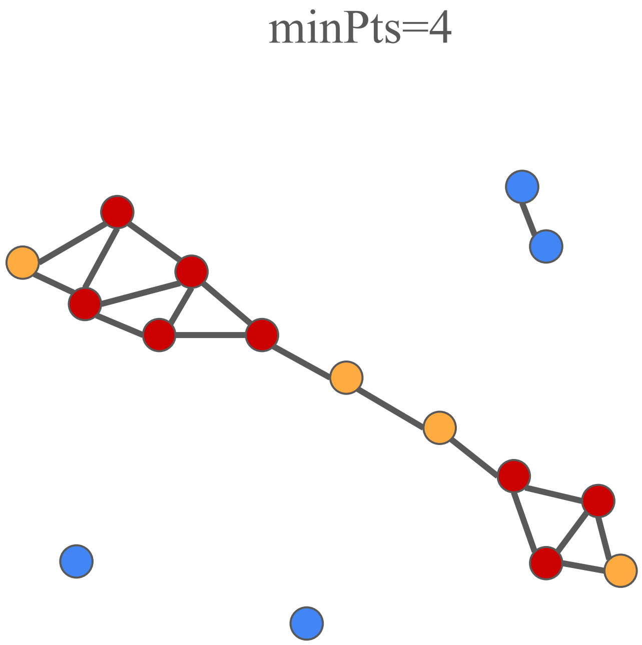


DBSCAN: Density-based spatial clustering of applications with noise

Clustering

Algorithm:

- Build the graph of for neighborhoods of radius r
- Search the core points: points that have minPts points in their neighborhood (including itself)
- Reachable point q if there is path $(p, p_1,\ldots,p_n,q)$ where p_i are core points
- Outliers: isolated points



Density-based spatial clustering of applications with noise, M Ester, HP Kriegel, J Sander, X Xu, 1996

DBSCAN: Density-based spatial clustering of applications with noise

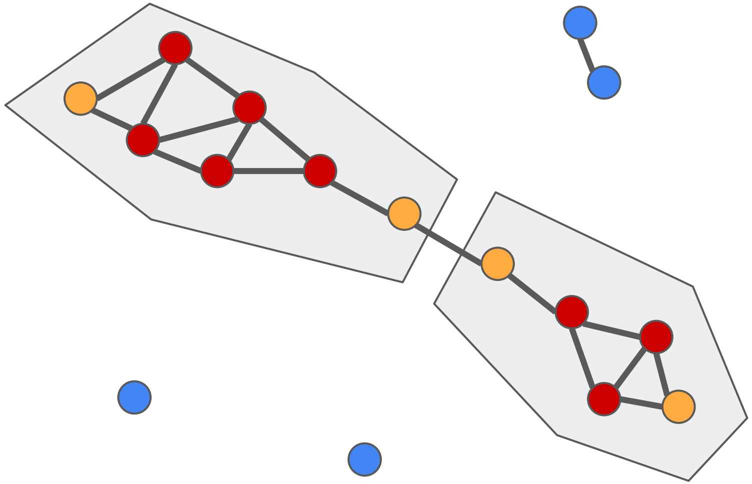
Clustering

Algorithm:

- Build the graph of for neighborhoods of radius r
- Search the core points: points that have minPts points in their neighborhood (including itself)
- Reachable point q if there is path $(p, p_1,\dots,p_n,q)$ where p_i are core points
- Outliers: isolated points

Cluster of p is all the reachable points from p.

minPts=4



Density-based spatial clustering of applications with noise, M Ester, HP Kriegel, J Sander, X Xu, 1996

K-means

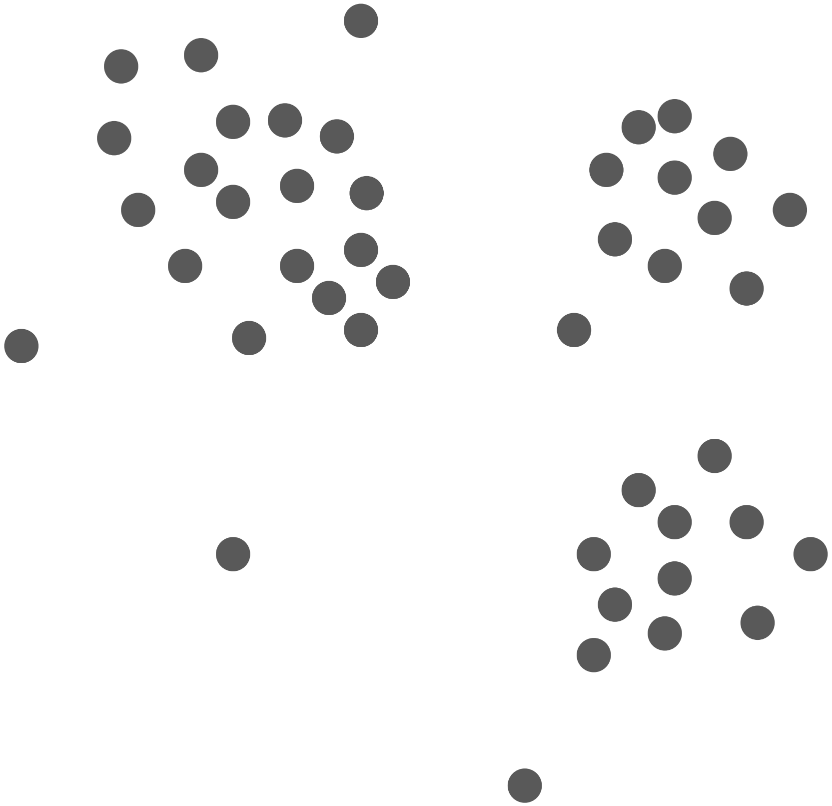
Clustering

One of the most used clustering algorithm.

Parameter: k, number of cluster

Advantage:

- Simple
- Proof of convergence

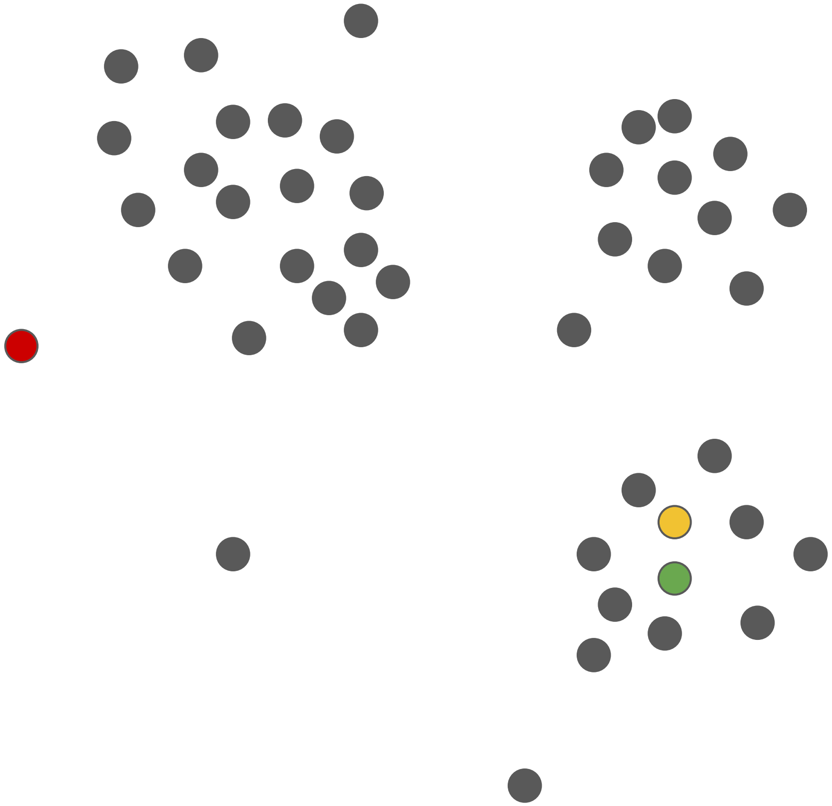


K-means

Clustering

Initialization:

Select random k points

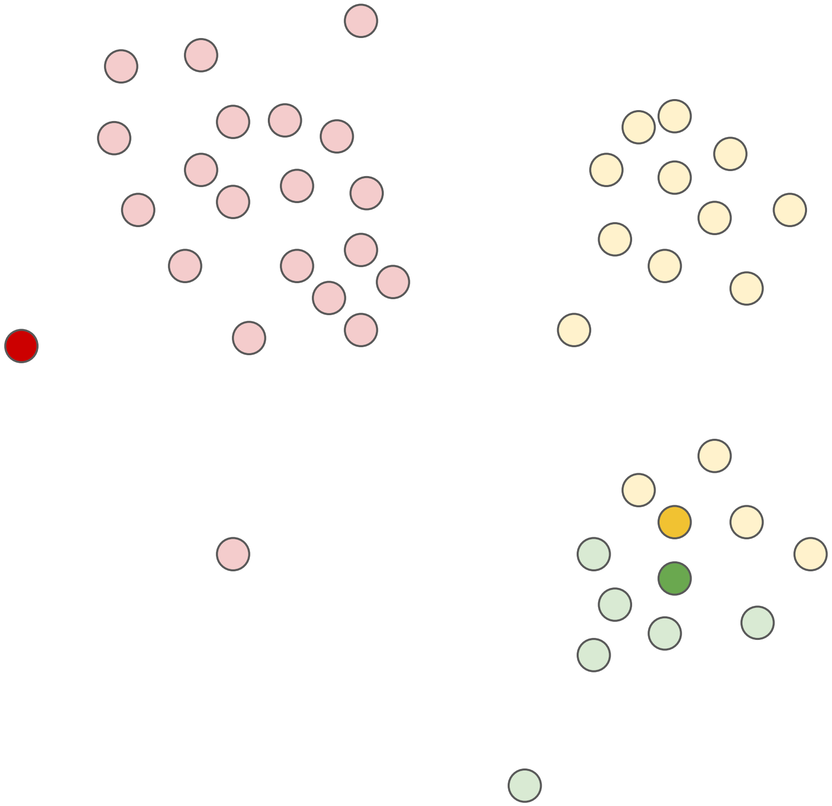


K-means

Clustering

Initialization:
Select random k points

Step 1:
Assign each point to its closest center

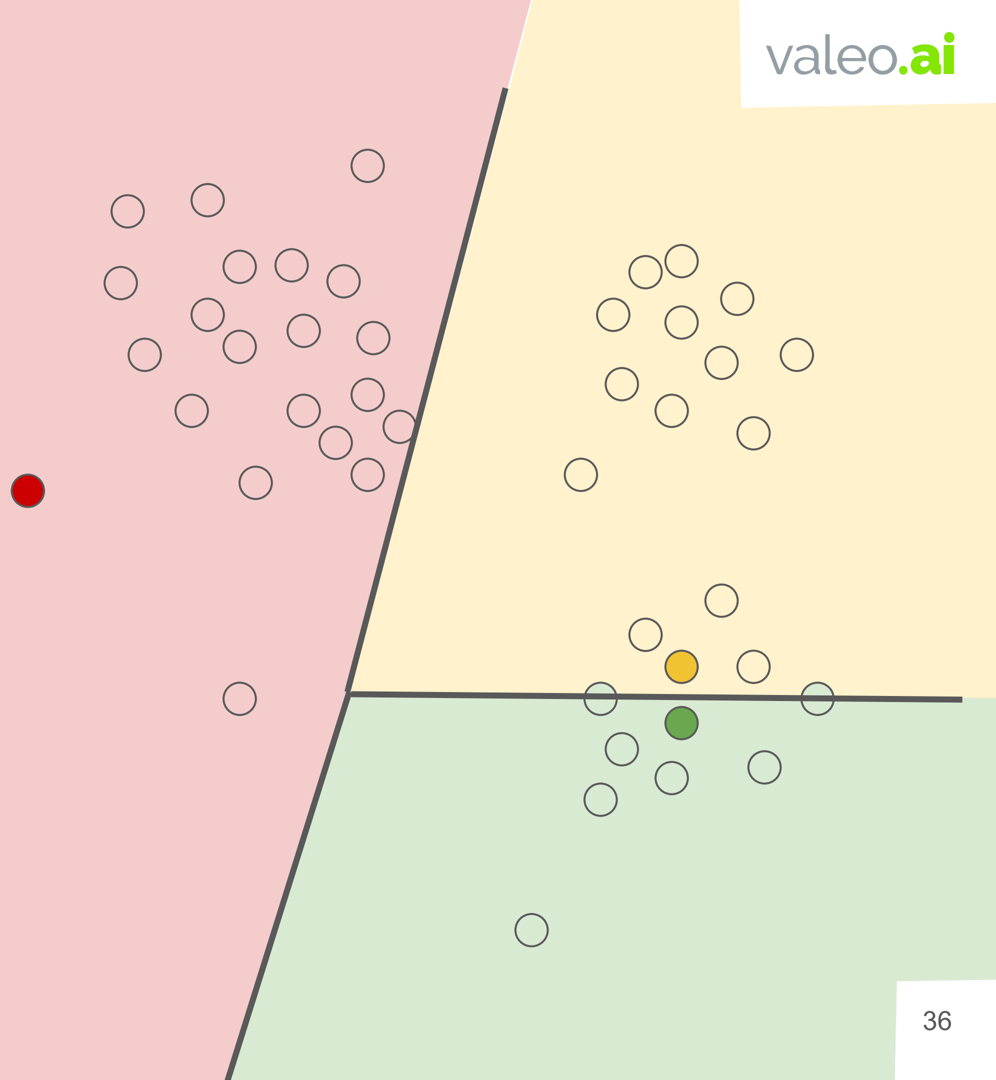


K-means

Clustering

Initialization:
Select random k points

Step 1:
Assign each point to its closest center
(Assignation based on Voronoï cells)



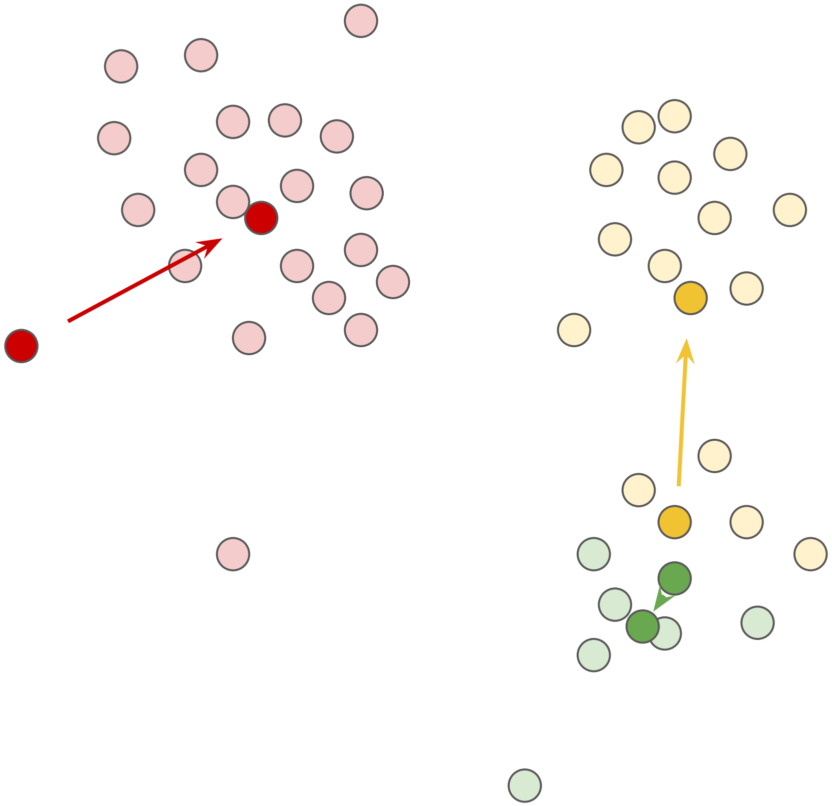
K-means

Clustering

Initialization:
Select random k points

Step 1:
Assign each point to its closest center

Step 2:
Compute new centroids



K-means

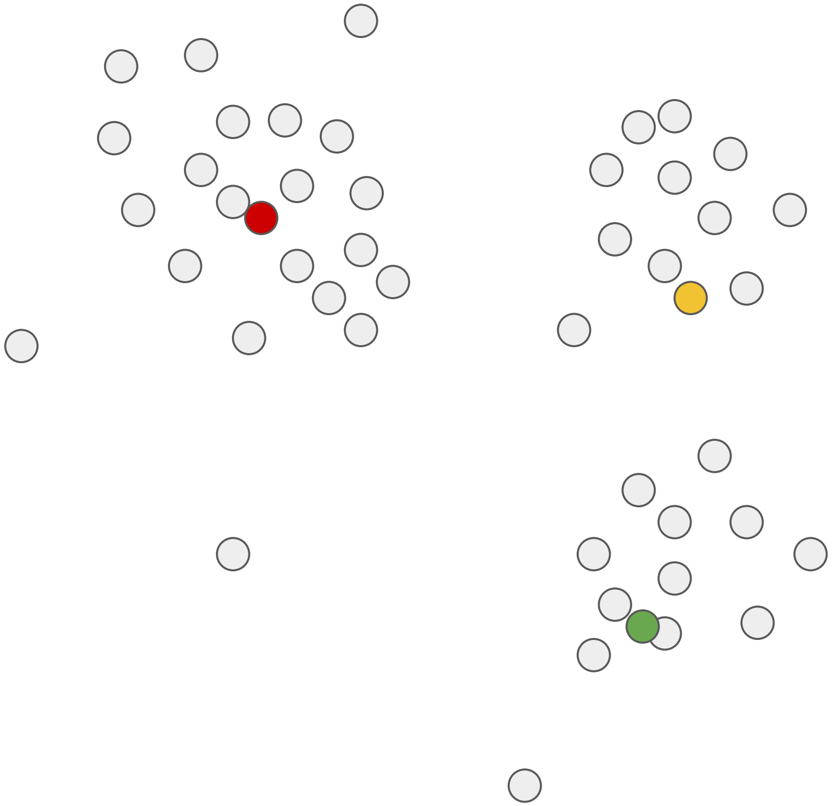
Clustering

Initialization:
Select random k points

Step 1:
Assign each point to its closest center

Step 2:
Compute new centroids

Iterate until convergence (no point changes cluster)



K-means

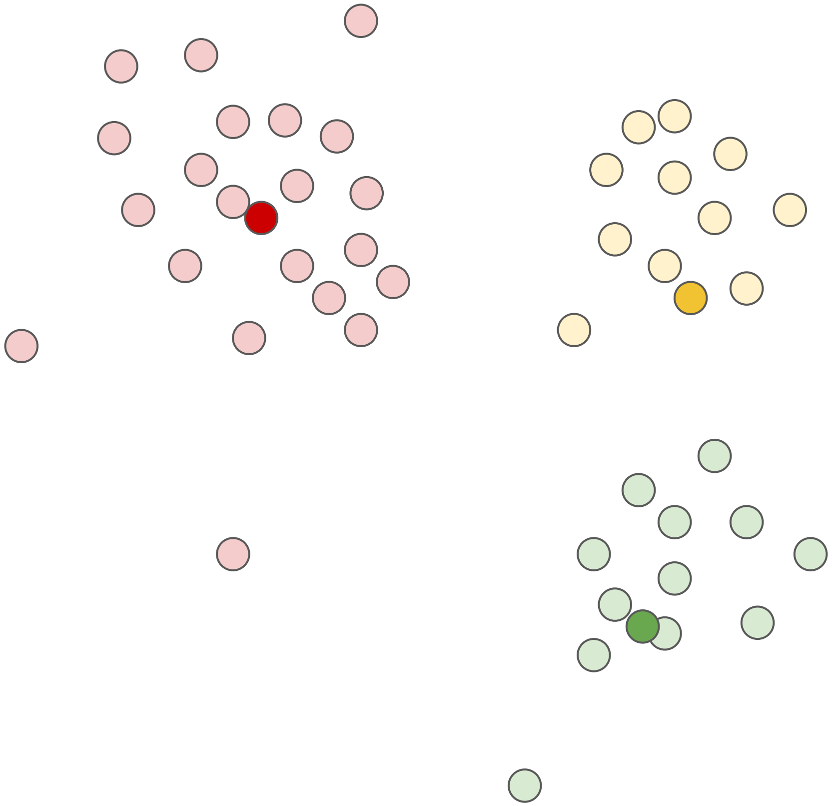
Clustering

Initialization:
Select random k points

Step 1:
Assign each point to its closest center

Step 2:
Compute new centroids

Iterate until convergence (no point changes cluster)



K-means

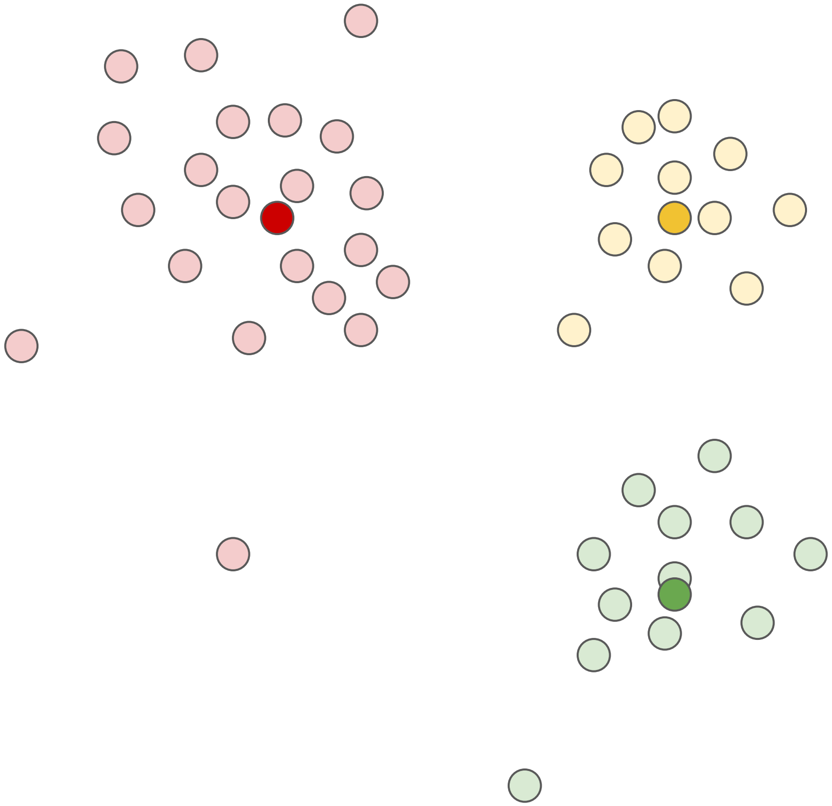
Clustering

Initialization:
Select random k points

Step 1:
Assign each point to its closest center

Step 2:
Compute new centroids

Iterate until convergence (no point changes cluster)



K-means

Clustering

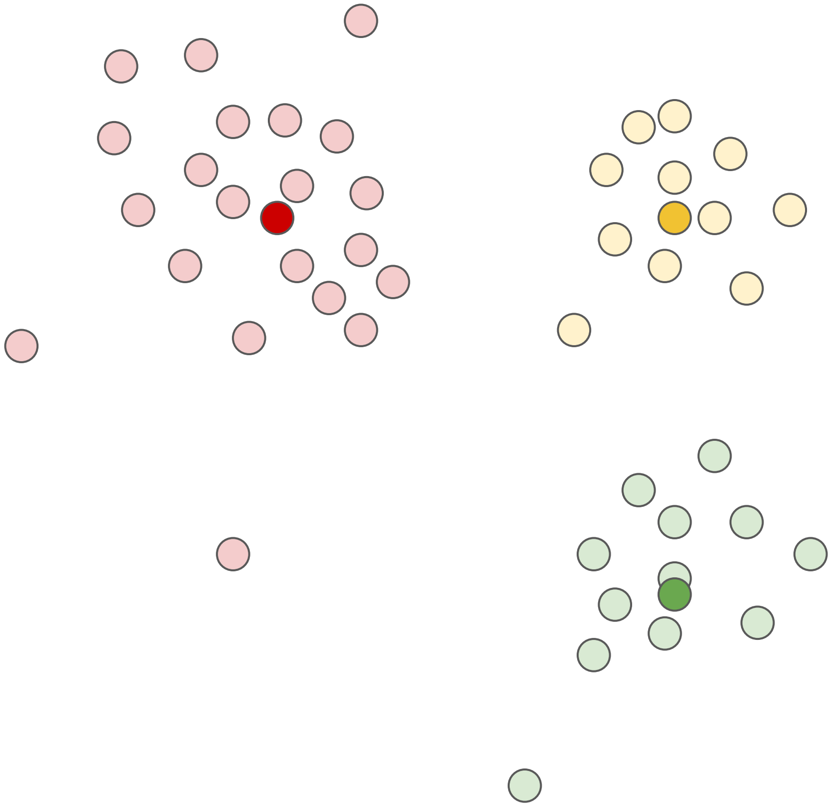
Advantages:

- Simple
- Any dimension
- Only on parameter
- Will converge

Disadvantages

- Convergence may be slow
  → bad initialization
- Need to know k

Mutilple method for estimating the number of clusters, better initialization…



Coming back to descriptors

Covariance-based descriptors

A - Local Descriptors

A single point → poor information (only 3 coordinates)

Local descriptors → neighborhood

Previous course

Normal estimation using neighborhood (K-nearest neighbors)

Construction of the covariance matrix

Hugues Thomas et al. “Semantic classification of 3D point clouds with multiscale spherical neighborhoods”. 3DV, 2018

Covariance-based descriptors

A - Local Descriptors

Covariance matrix (again)

Build the covariance matrix for the neighborhood N of a point q:

- Average:

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

Covariance:

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

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

Hugues Thomas et al. “Semantic classification of 3D point clouds with multiscale spherical neighborhoods”. 3DV, 2018

Covariance-based descriptors

A - Local Descriptors

valeo.ai

Covariance matrix (again)

- Compute covariance matrix

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

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

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

$$\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$$

$$\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in \mathbb{R}^3$$

Hugues Thomas et al. “Semantic classification of 3D point clouds with multiscale spherical neighborhoods”. 3DV, 2018

Covariance-based descriptors

A - Local Descriptors

Covariance matrix (again)

- Compute covariance matrix

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

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

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

$$\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$$

$$\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in \mathbb{R}^3$$

Hugues Thomas et al. “Semantic classification of 3D point clouds with multiscale spherical neighborhoods”. 3DV, 2018

Covariance-based descriptors

A - Local Descriptors

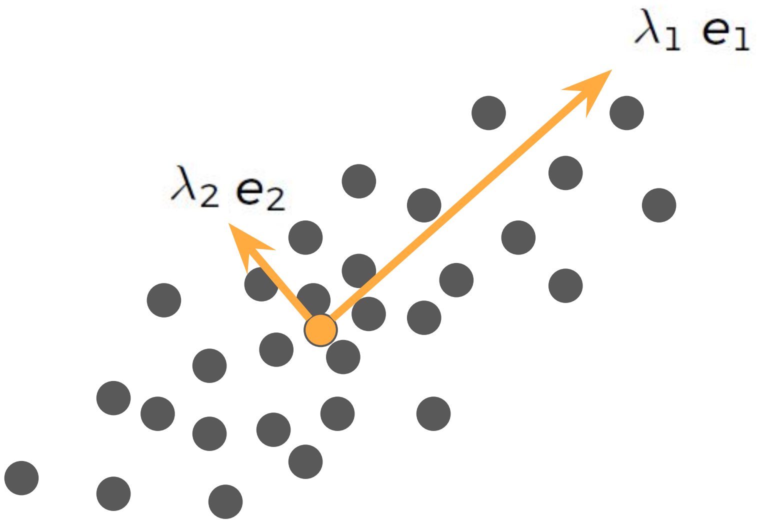
What are $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$?

PCA spread

Orthogonal basis $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in \mathbb{R}^3$

$\mathbf{e}_3$ is the normal

⇒ there is more information than just the normal

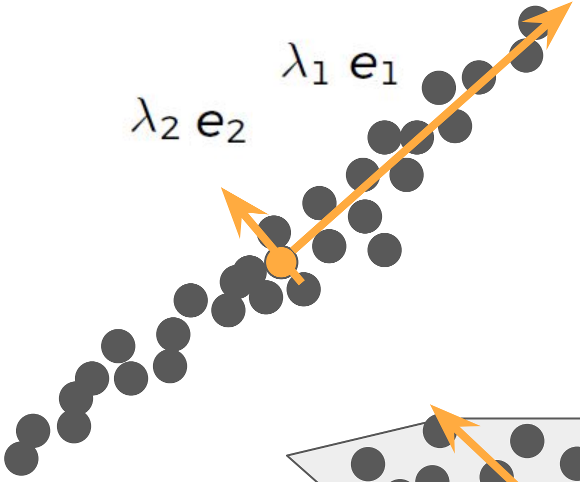


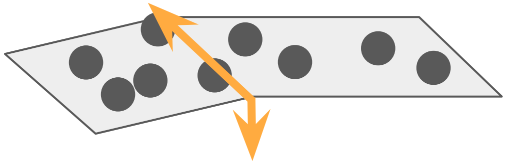
Hugues Thomas et al. “Semantic classification of 3D point clouds with multiscale spherical neighborhoods”. 3DV, 2018

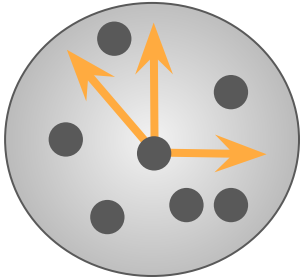
Covariance-based descriptors

A - Local Descriptors

Sum of eigenvalues	$\sum \lambda_i$
Omnivariance	$\left(\prod \lambda_i\right)^{\frac{1}{3}}$
Eigenentropy	$-\sum \lambda_i \ln(\lambda_i)$
Linearity	$(\lambda_1 - \lambda_2)/\lambda_1$
Planarity	$(\lambda_2 - \lambda_3)/\lambda_1$
Sphericity	λ_3/λ_1
Change of curvature	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$







Hugues Thomas et al. “Semantic classification of 3D point clouds with multiscale spherical neighborhoods”. 3DV, 2018

Covariance-based descriptors

A - Local Descriptors

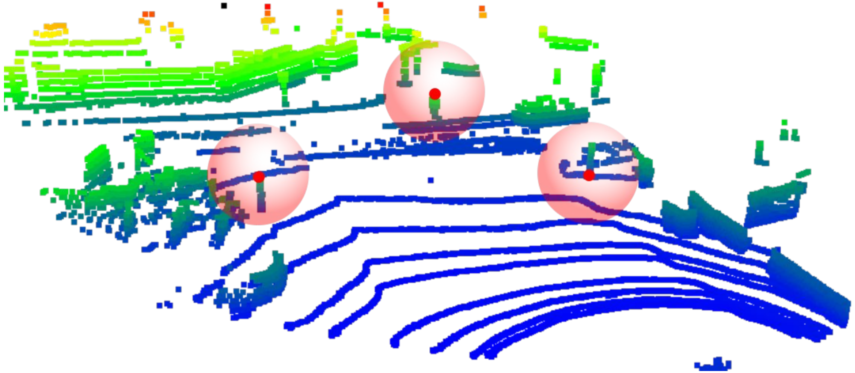
Sum of eigenvalues	$\sum \lambda_i$
Omnivariance	$\left(\prod \lambda_i\right)^{\frac{1}{3}}$
Eigenentropy	$-\sum \lambda_i \ln(\lambda_i)$
Linearity	$(\lambda_1 - \lambda_2)/\lambda_1$
Planarity	$(\lambda_2 - \lambda_3)/\lambda_1$
Sphericity	λ_3/λ_1
Change of curvature	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$

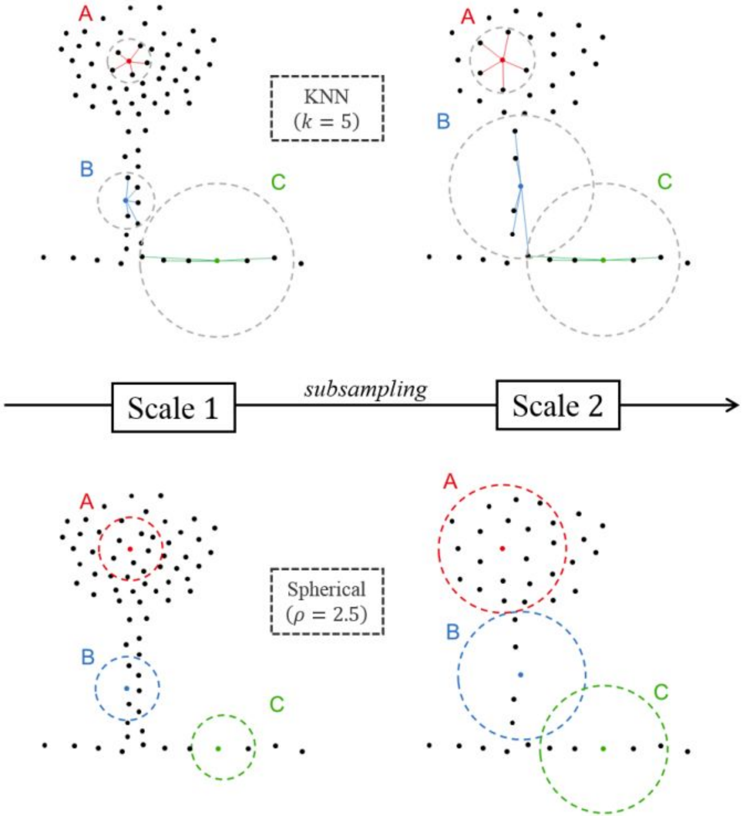
Verticality (x2)	$\left\lvert \frac{\pi}{2} - \text{angle}(\mathbf{e}_i, \mathbf{e}_z) \right\rvert_{i \in (0,2)}$
Absolute moment (x6)	$\frac{1}{\lvert\mathcal{N}\rvert} \left\lvert \sum \langle \mathbf{p} - \mathbf{p}_0, \mathbf{e}_i \rangle^k \right\rvert_{i \in (0,1,2)}$
Vertical moment (x2)	$\frac{1}{\lvert\mathcal{N}\rvert} \sum \langle \mathbf{p} - \mathbf{p}_0, \mathbf{e}_z \rangle^k$
Number of points	$\lvert\mathcal{N}\rvert$
Average color (x3)	$\frac{1}{\lvert\mathcal{N}\rvert} \sum c$
Color variance (x3)	$\frac{1}{\lvert\mathcal{N}\rvert - 1} \sum (c - \bar{c})^2$

Hugues Thomas et al. “Semantic classification of 3D point clouds with multiscale spherical neighborhoods”. 3DV, 2018

Covariance-based descriptors

A - Local Descriptors





Hugues Thomas et al. “Semantic classification of 3D point clouds with multiscale spherical neighborhoods”. 3DV, 2018

Covariance-based descriptors

A - Local Descriptors

Pros

- Fast to compute
- Memory efficient
- Simple classifier
  (SVM, random forests, MLP...)
- Requires limited amount of data for training

Cons

- No long distance relations
- Limited by the descriptors:
  - The system cannot infer the most suited descriptors
- Difficulty to create dataset specific descriptors

Hugues Thomas et al. “Semantic classification of 3D point clouds with multiscale spherical neighborhoods”. 3DV, 2018

I - Descriptors and machine learning

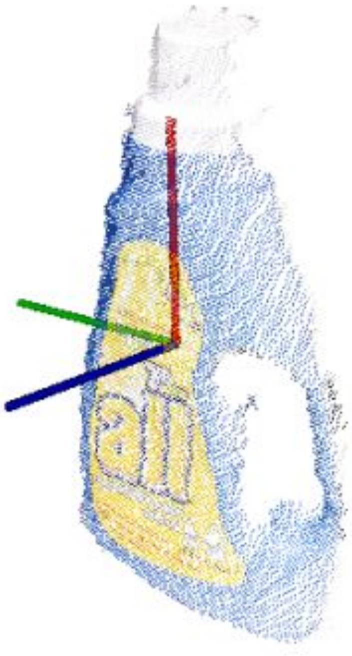
B - Global descriptors

GASD - Globally Aligned Spatial Distribution

B - Global descriptors

Descriptor for a complete shape
→ shape classification

Step 1: compute global orientation (PCA)



do Monte et al. "An efficient global point cloud descriptor for object recognition and pose estimation." *SIBGRAPI*, 2016.

GASD - Globally Aligned Spatial Distribution

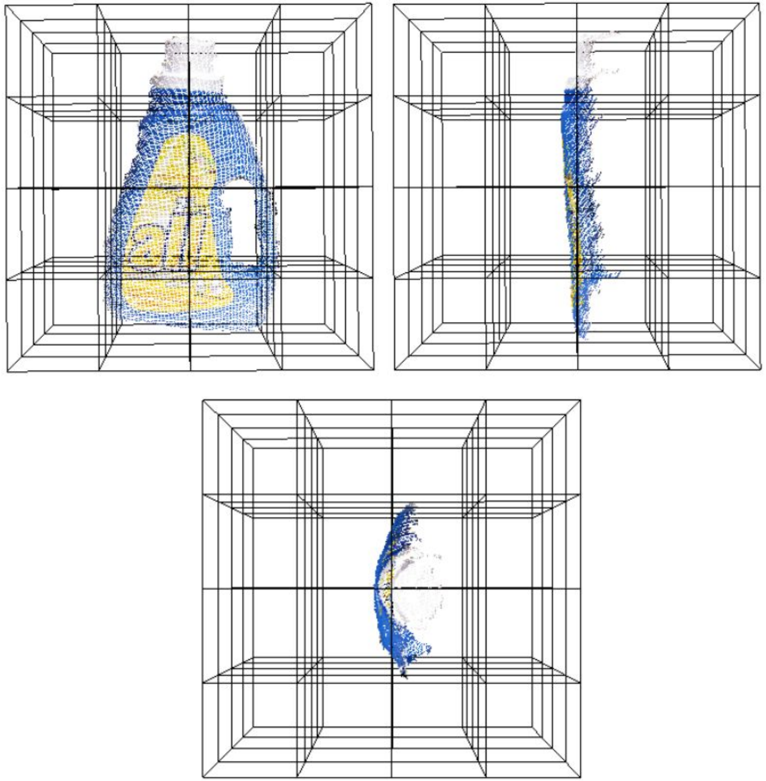
B - Global descriptors

Descriptor for a complete shape
→ shape classification

Step 1: compute global orientation (PCA)

Step 2:

- Normalized histogram of the count of points in each cell
- Average color per cell



do Monte et al. "An efficient global point cloud descriptor for object recognition and pose estimation." *SIBGRAPI*, 2016.

GASD - Globally Aligned Spatial Distribution

B - Global descriptors

Descriptor for a complete shape
→ shape classification

Fig. 14. GASD fails to recognize a partially occluded blue detergent bottle.

Fig. 7. Object recognition and pose estimation example: input scene (left), results obtained using ESF (center) and GASD-SI (right).

do Monte et al. "An efficient global point cloud descriptor for object recognition and pose estimation." *SIBGRAPI*, 2016.

Global descriptors

In practice:

This approach can be used with any descriptor (use accumulator)

Warning:

- Orientation is a hard problem (missing elements, partial view…)

Current trend would be to use augmentations

- Use multiple rotation
- Masking

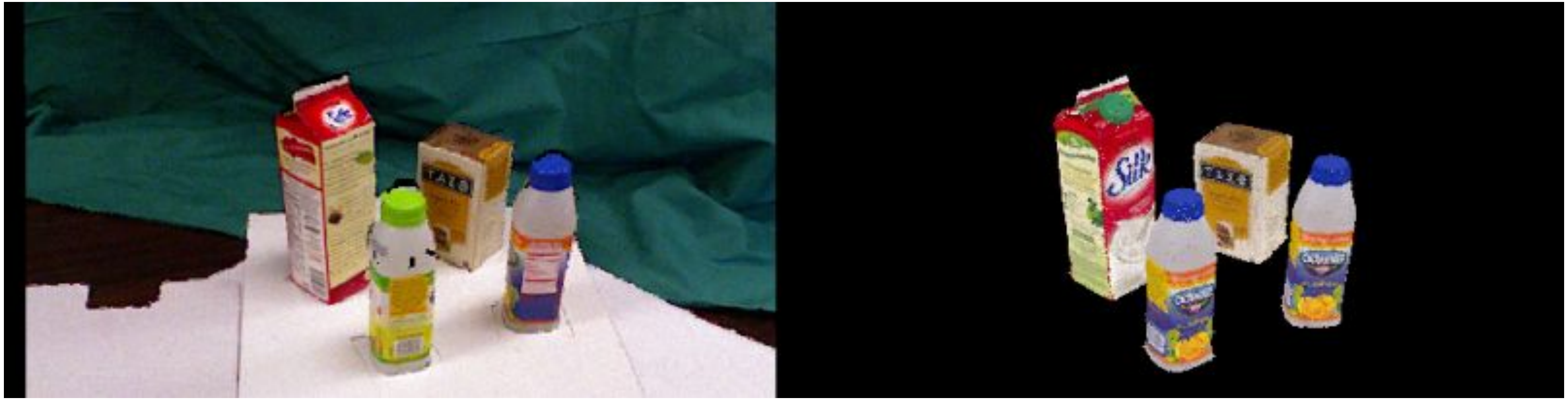
I - Descriptors and machine learning

C - Downstream tasks

Object retrieval

C - Downstream tasks

Given a descriptor look for the k-closest in a database



Classification

C - Downstream tasks

Associate each shape with a label

- Expert decision (by hand)
- ML: SVMs, RandomForest, MLP…

Requires a **labeled train set**

Classification

C - Downstream tasks

Associate each shape with a label

- Expert decision (by hand)
- ML: SVMs, RandomForest, MLP…

Requires a **labeled train set**

ModelNet 10

Semantic segmentation

C - Downstream tasks

Associate each point with a label

- Expert decision (by hand)
- ML: SVMs, RandomForest, MLP…

Requires a **labeled train set**

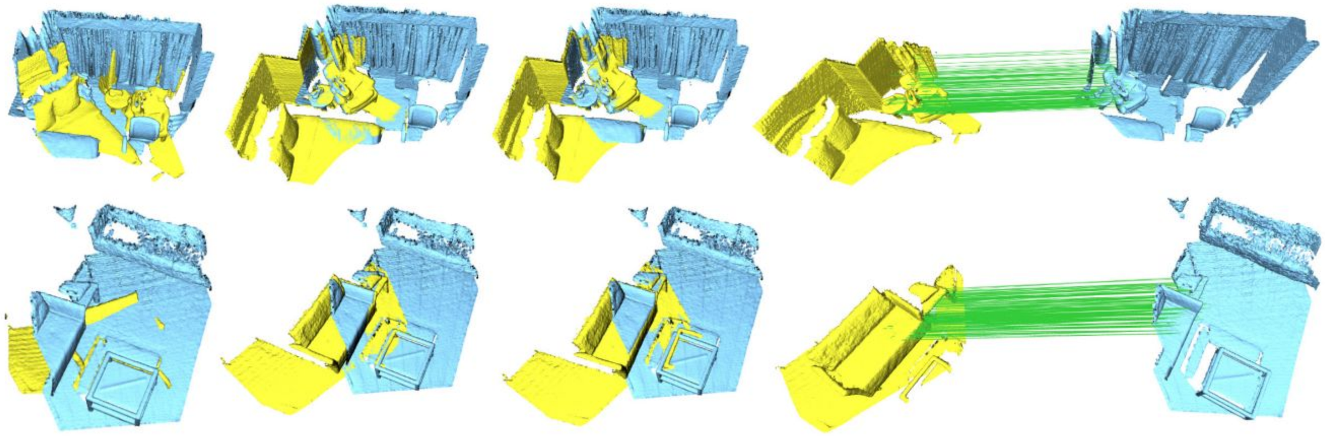
NPM3D

Point matching

C - Downstream tasks

Match local descriptor from one point cloud to another

Estimate transformation (e.g., RANSAC)



II - Image-based approaches

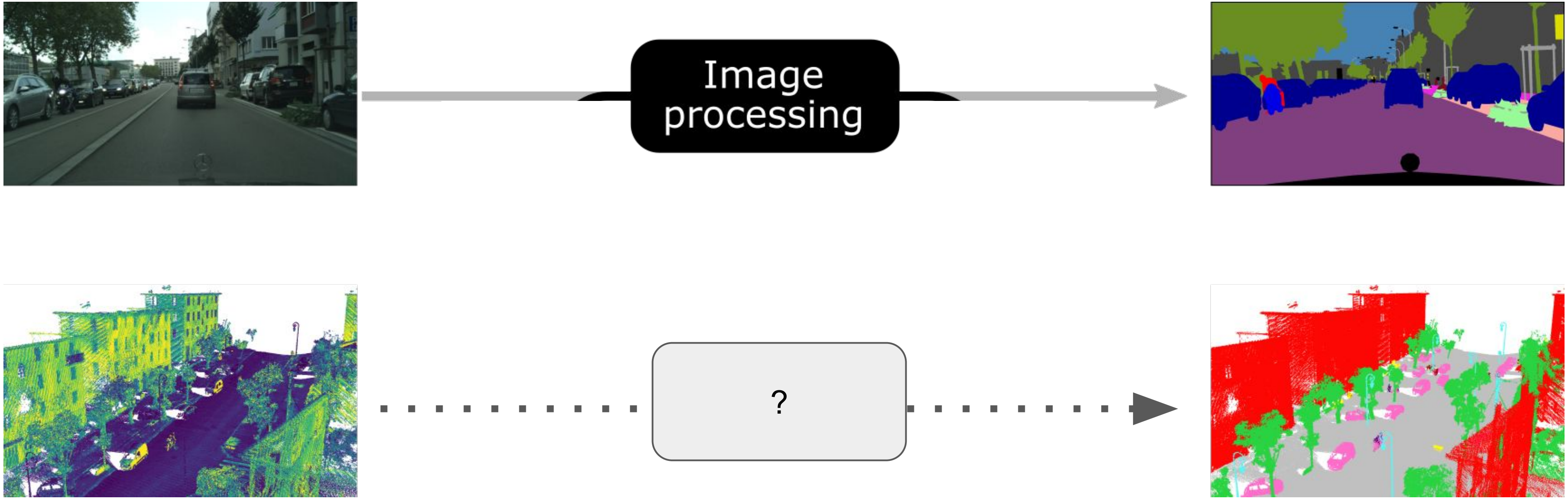
Idea

Image-based approaches

Image processing is a well studied problem

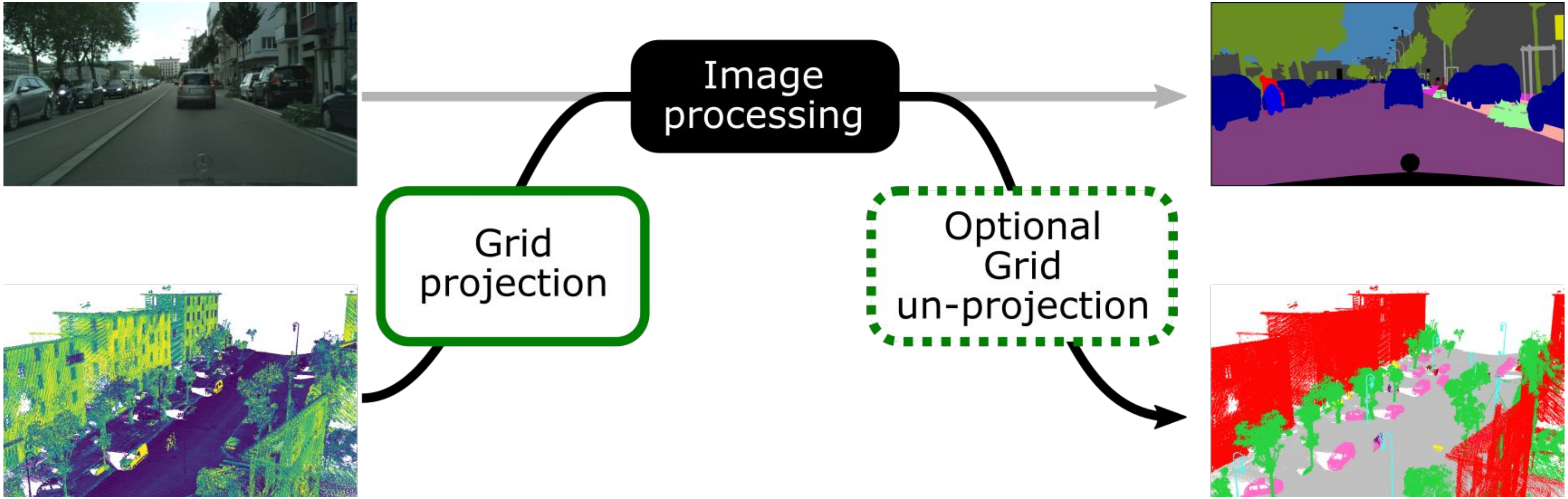
Idea

Image-based approaches



Idea

Image-based approaches



Regular grid projections

Image-based approaches

Images are pixels arrays

Implicit neighborhoods

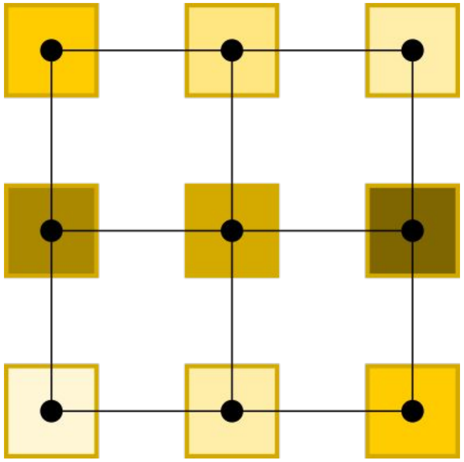
Information is in the color and relative position of the pixels

Thanks to this grid structure

Optimized network architectures

Fast (hardware optimization)

Relatively low memory cost



2D projections

Image-based approaches

2D convolution for an image patch centered on pixel n:

$$\mathbf{h}[n] = \sum_{f \in \{1,\ldots,C\}} \sum_{m \in \{-M/2,\ldots,M/2\}^2} \mathbf{K}_f[m]\, \mathbf{f}_f[n+m]$$

With **f**: input features and **K**: convolution kernel

And new architectures for images:

Vision transformers, MLP Mixers, …

Regular grid projections

Image-based approaches

Images are pixels arrays

Implicit neighborhoods

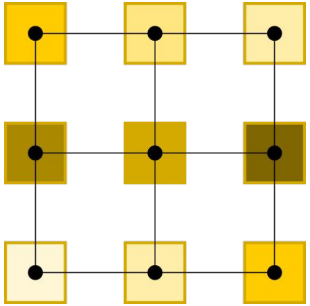
Information is in the color and relative position of the pixels

Thanks to this grid structure

Optimized network architectures

Fast (hardware optimization)

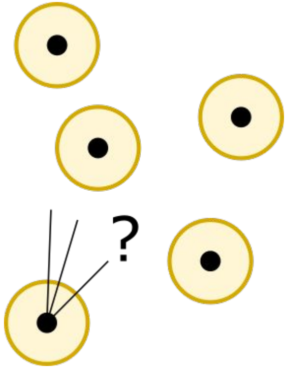
Relatively low memory cost



Point clouds:

~~Implicit neighborhoods~~

~~Information is in the color and relative position of the pixels~~



Idea

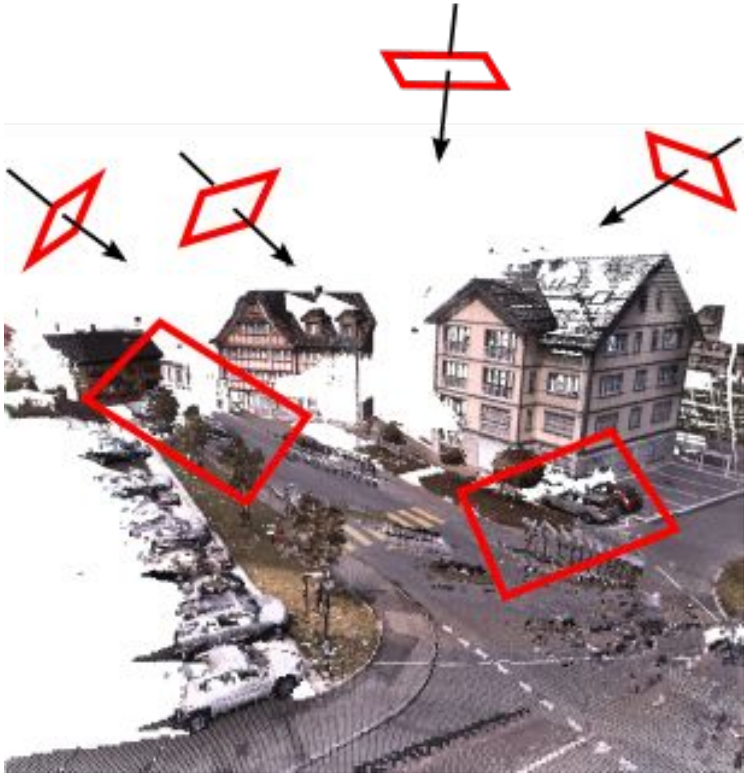
Find a way to create grid data from point cloud

2D projections

Image-based approaches

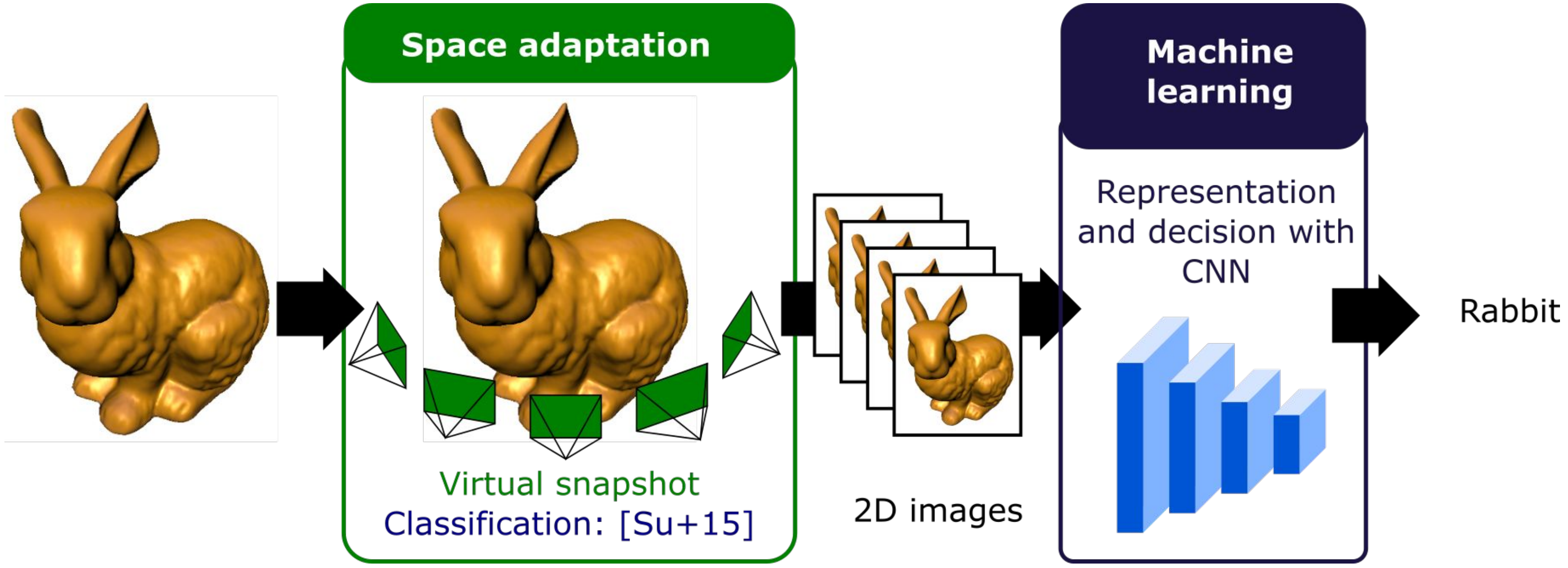
Generate images representing the scene

- Use a 3D renderer
- Take virtual snapshots of the scene
- Work in the image



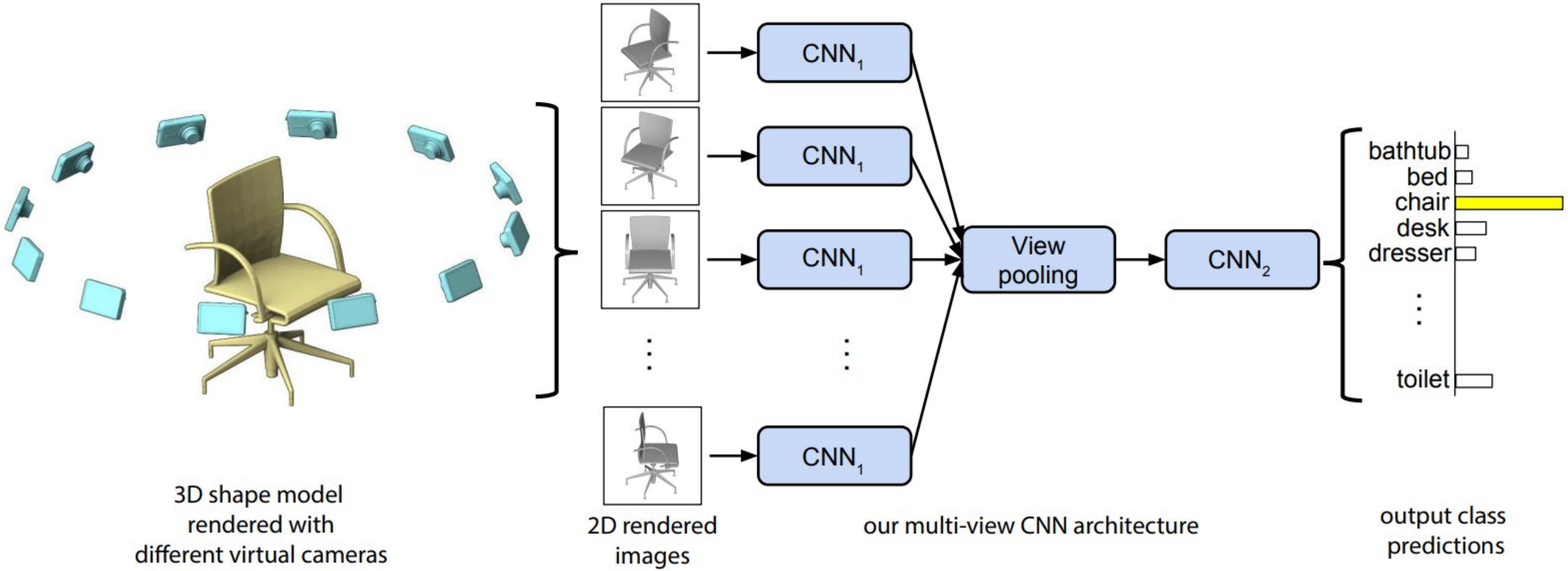
Classification pipeline

Image-based approaches



Classification pipeline

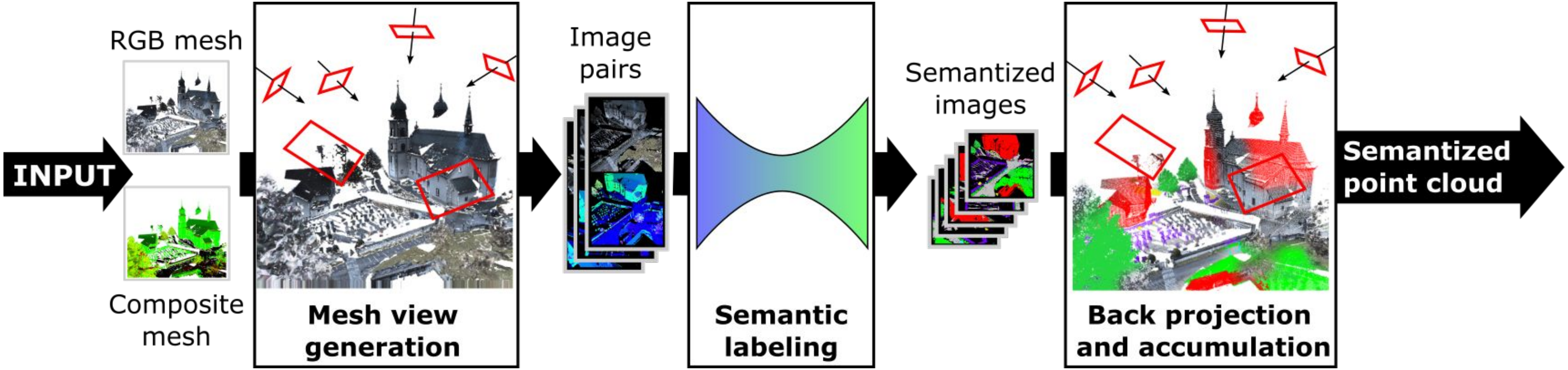
Image-based approaches



Hang Su et al. “Multi-view convolutional neural networks for 3D shape recognition”. In: Proceedings of the IEEE international conference on computer vision. 2015, pp. 945–953

Semantic segmentation pipeline

Image-based approaches



Alexandre Boulch et al. “SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks”. In: Computers & Graphics (2017)

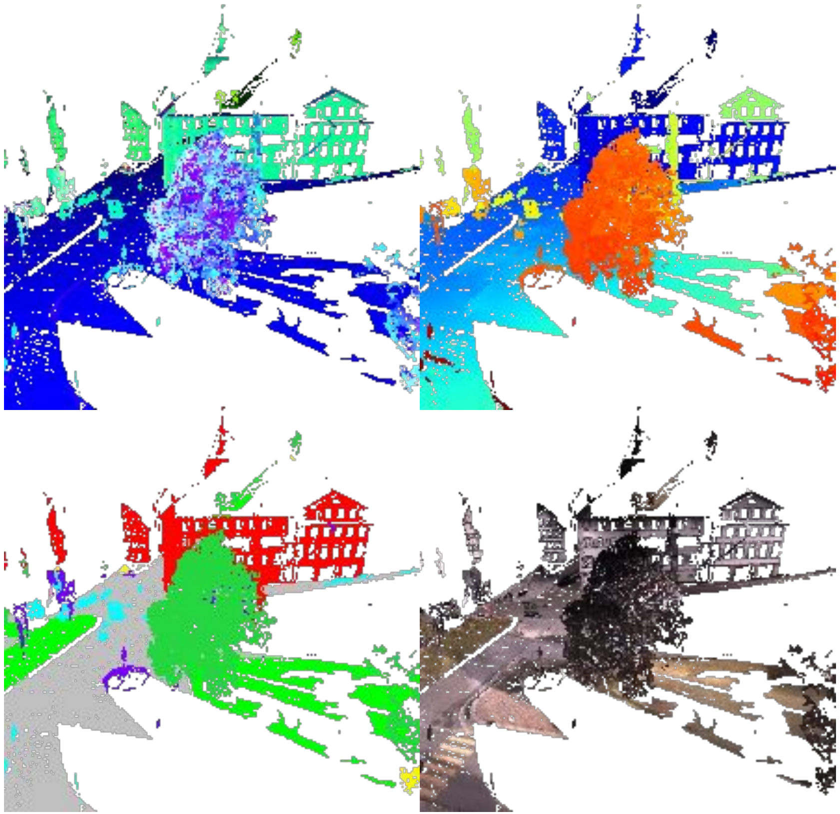
SnapNet

Image-based approaches

Reprojection trick

Generate a snapshot of the scene with fake colors corresponding to point ids

- Allow to generate different snapshots (w / wo colors, geometric features, ground truth at training...)
- Easy reprojection of the results on the original points



Alexandre Boulch et al. “SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks”. In: Computers & Graphics (2017)

SnapNet - advantages and limitations

Image-based approaches

Pros

- Benefit from architectures from image processing
- Use of pre-trained models from large image datasets
- Unlimited number of snapshot a given scene (straightforward data augmentation)

Cons

- Good snapshot strategy, which can vary from a dataset to another
- Requires a mesh

Alexandre Boulch et al. “SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks”. In: Computers & Graphics (2017)

Range projection

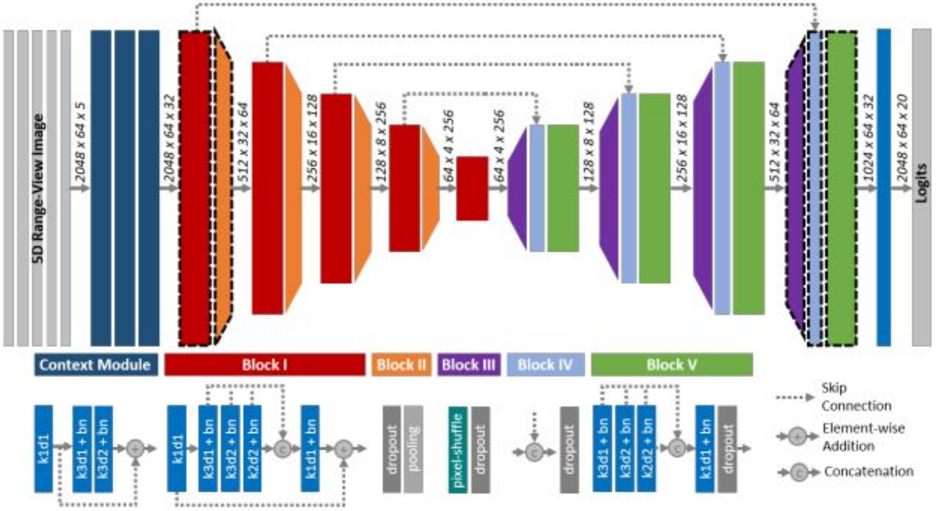
Image-based approaches

Exploit sensor information to produce images

SalsaNext

Image-based approaches

Use image backbone (U-Net) for semantic segmentation



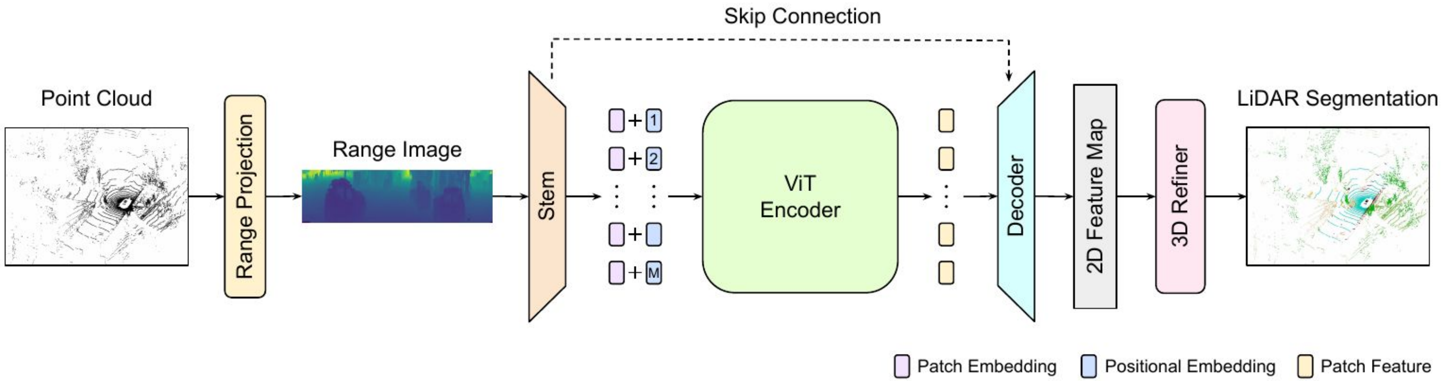
Cortinhal, Tiago et al. "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds." ISVC 2020.

RangeViT

Image-based approaches

LiDAR segmentation based on range images & vision transformers (ViTs)

- Unify architectures in LiDAR and image domain
- Leverage image pre-trained ViTs for LiDAR segmentation



Angelika Ando et al., "RangeViT: Towards Vision Transformers for 3D Semantic Segmentation in Autonomous Driving" CVPR 2023

Practical session

Machine learning 1

Practical session

- Implement covariance based local descriptors.
- Train a classifier
- Evaluate on validation point cloud