A1

Short version

Some computer vision conferences: List

- Conference on Computer Vision and Pattern Recognition (CVPR)
- International Conference on Computer Vision (ICCV)
- European Conference on Computer Vision (ECCV)
- Special Interest Group on GRAPHics and Interactive Techniques (SIGGRAPH)
- Neural Information Processing Systems (NIPS)
- International Conference on Machine Learning (ICML)
- International Conference on Pattern Recognition (ICPR)

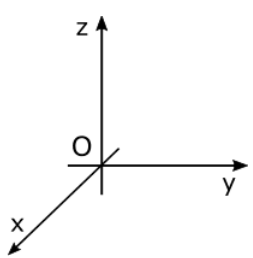
Some robotics conferences: List

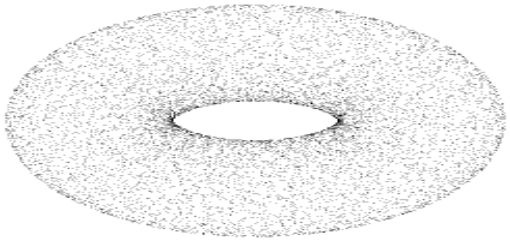
- IEEE International Conference on Robotics and Automation (ICRA)
- IEEE International Conference on Intelligent Robots and Systems (IROS)
- Robotics: Science and Systems (RSS)
- IEEE International Conference on Intelligent Transportation Systems (ITS)
- IEEE Intelligent Vehicles Symposium (IEEE IV)

Conferences deadlines

- (CVPR): Mid November
- (ICCV): Mid March
- (ECCV): Beginning March
- (SIGGRAPH): Mid January
- (NIPS): Beginning January
- (ICML): Beginning February
- (ICPR): Mid April
- (ICRA): Mid September
- **(IROS): Beginning March**
- (RSS): End January
- (ITS): Mid June
- (IEEE IV): Beginning February
- **…**

Point cloud, centroid, normals

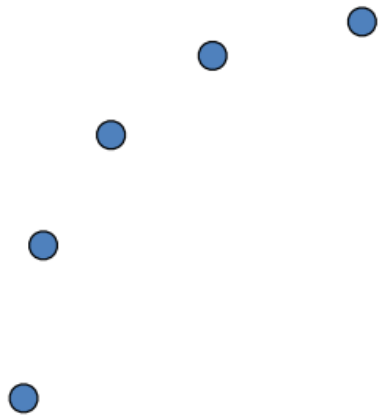




- Point cloud

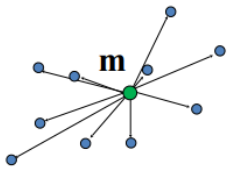
$x = \{x_1, \ldots, x_N\}$

$x_i = (a, b, c) \in R^3$

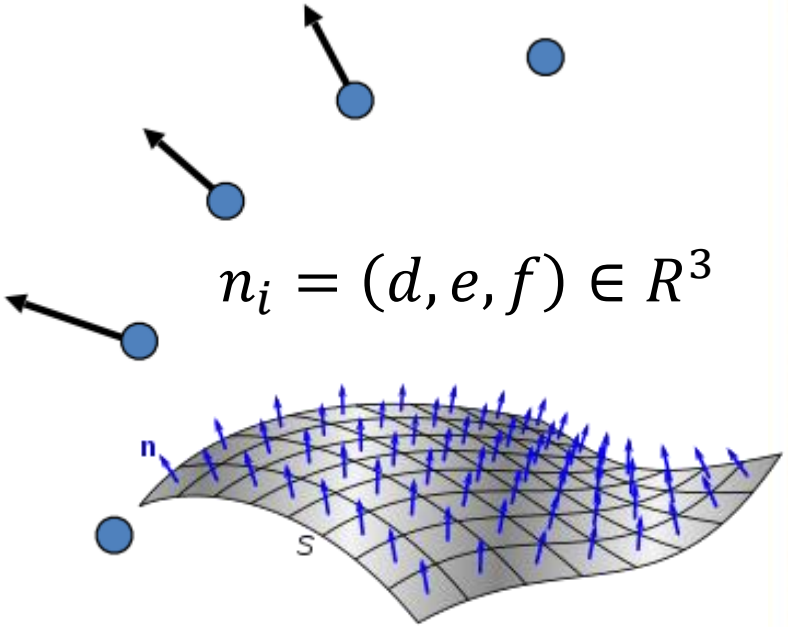


- Centroid

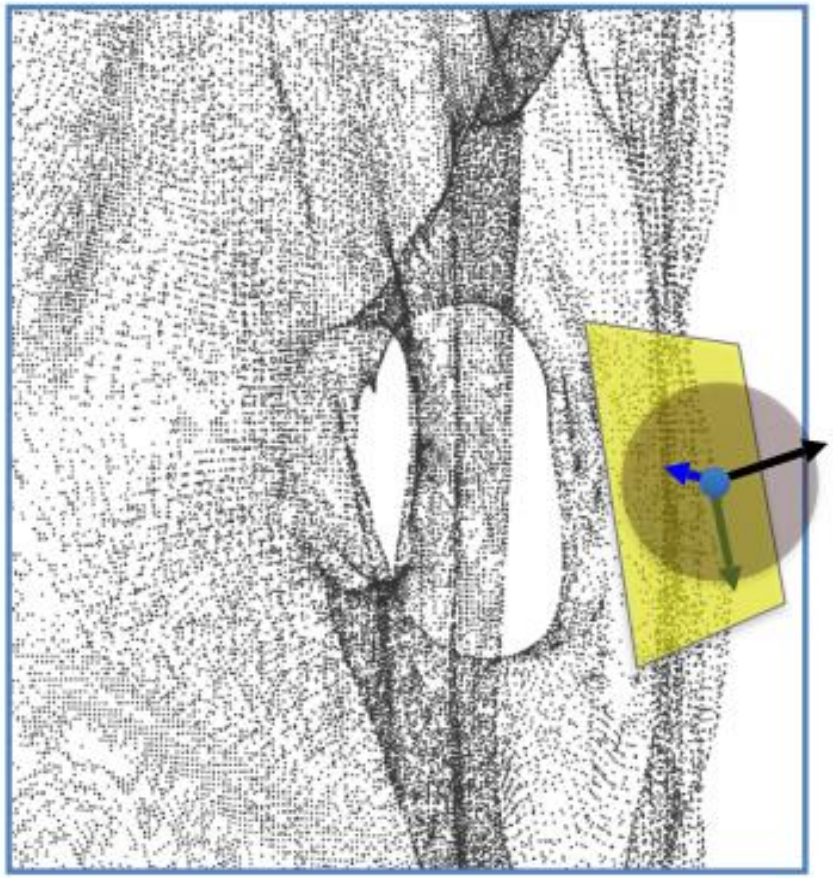
$$\mathbf{m} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$



- Normal Estimation: Assign a normal vector n at each point cloud point x



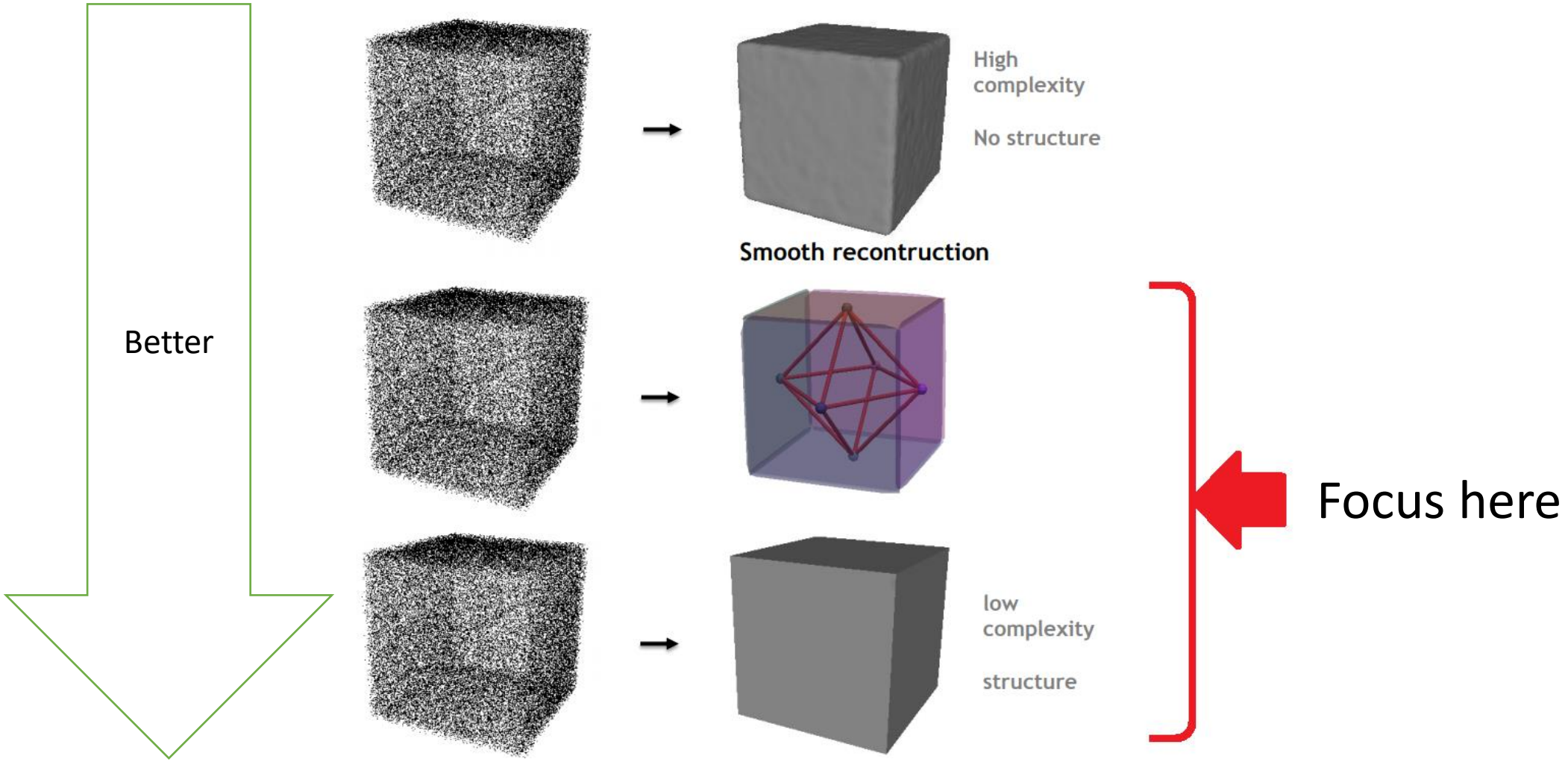
$n_i = (d, e, f) \in R^3$



https://cs.nyu.edu/~panozzo/gp/04%20-%20Normal%20Estimation,%20Curves.pdf

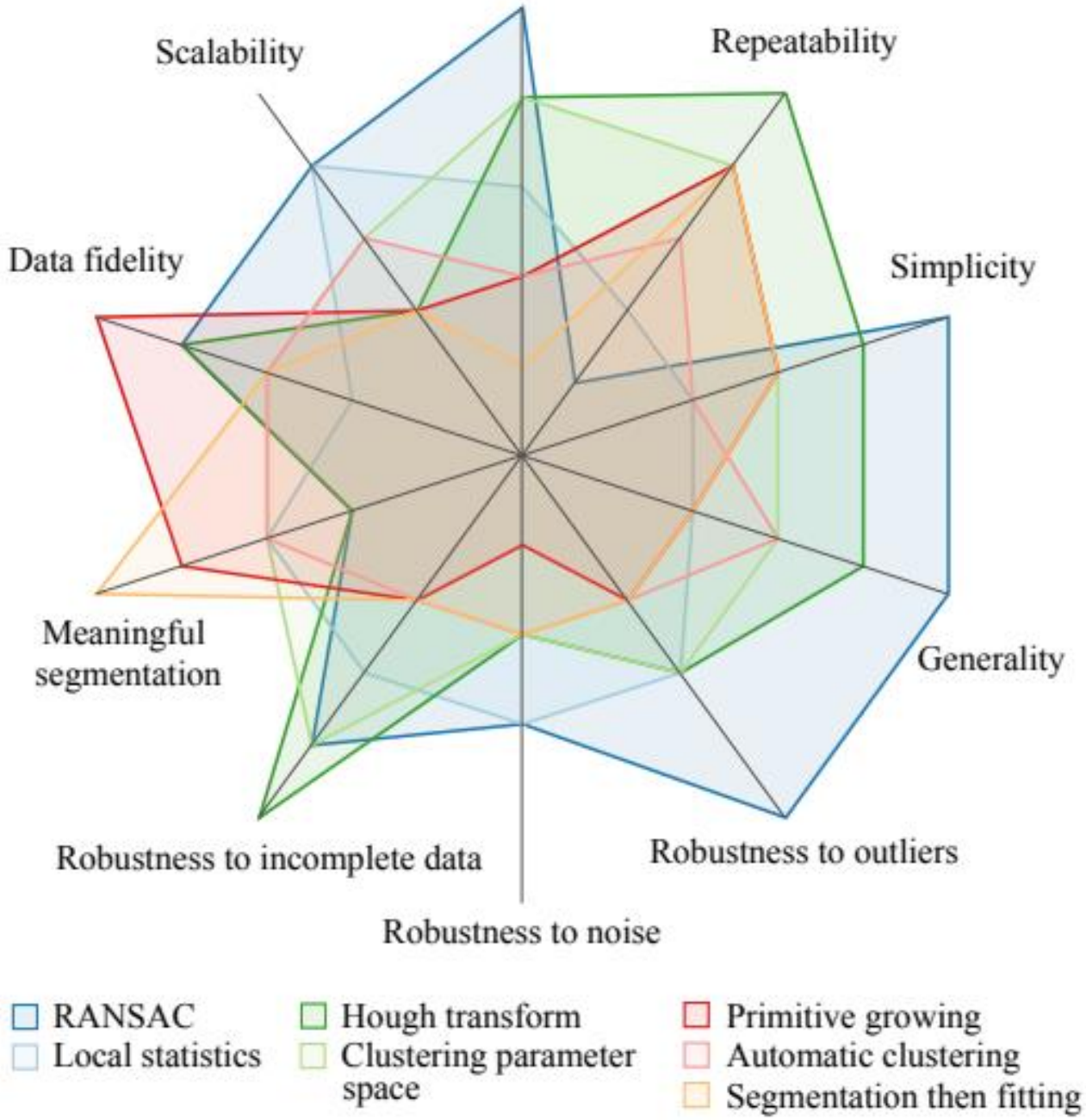
[Ref: Primitive-based surface reconstruction Florent Lafarge Inria Sophia Antipolis – Mediterranee] https://team.inria.fr/titane/files/2015/04/surface_reconstruction_primitive-based.pdf

Primitive-based surface reconstruction



Techniques

- **Stochastic**: RANSAC, local statistics;
- **Parameter spaces**: Hough-like voting methods, parameter space clustering;
- **Other clustering techniques**: primitive-driven region growing, Lloyd-like automatic clustering, primitive-oblivious segmentation followed by fitting.



Kaiser, Adrien, Jose Alonso Ybanez Zepeda, and Tamy Boubekeur. "A survey of simple geometric primitives detection methods for captured 3d data." *Computer Graphics Forum*. Vol. 38. No. 1. 2019.

Techniques (2020)

LIST OF REVIEWED PRIMITIVE EXTRACTION METHODS

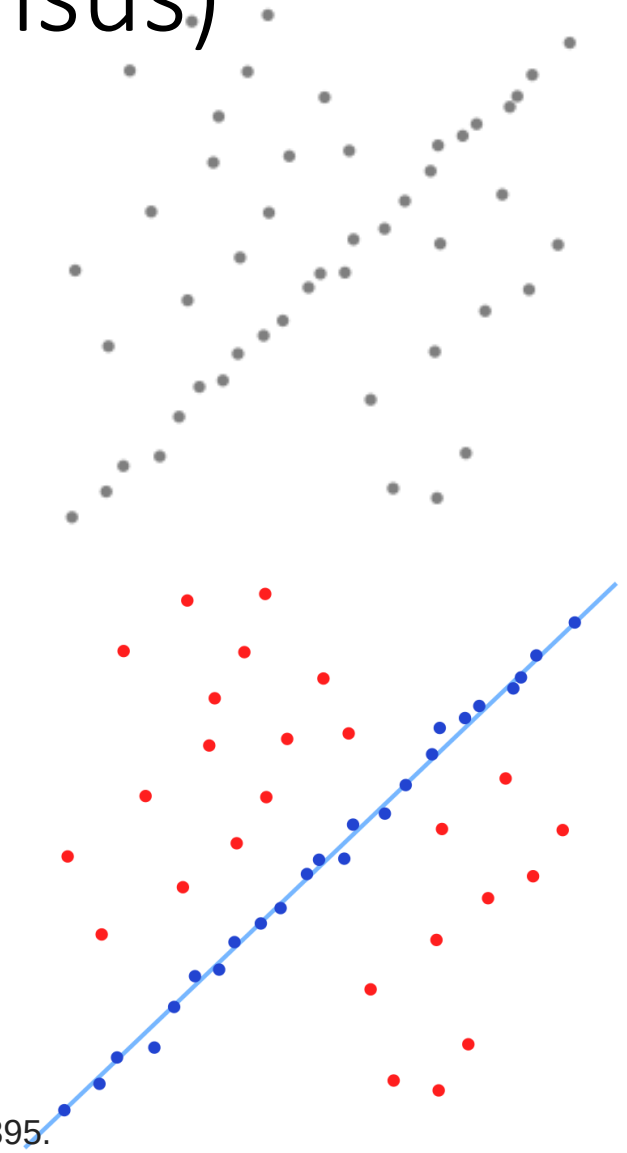
Methods		Descriptions	Strengths	Limitations
Region Growing [41], [43], [84]–[94], [152]		Merge adjacent points into segments based on feature differences	* Easy to implement * Flexible in adding rules	* Parameter tuning * Locally optimal * Sensitive to noise
Accumulation [94]–[96], [98]–[104], [106]–[109]		Voting in parameter spaces where maxima are used to identify primitives	* Easy to implement * Robust to noise * Robust to missing data	* Voting space construction * Space/time complexity * Locally optimal
Hypothesis and Selection	RANSAC [62], [100], [112]–[120], [122]–[127]	Generate potential models among which the one with most inliers is selected	* Easy to access * Robust to outliers	* Degeneracy * Locally optimal
	Energy-based Optimization [128], [129], [131]–[138]	Generate guesses and select models by optimizing an objective function	* Add prior knowledge * Globally optimal solutions * Flexible in adding rules	* Time consuming * Parameter tuning * Little lidar experience
	Preference Analysis [139]–[142]	Select models based on the preference of each point	* Robust to outliers * A global view of guesses	* Time consuming * Numerous guesses * Little lidar experience
Clustering [34], [68], [72], [73], [143]–[151]		Using existing clustering algorithms	* Many available methods	* Density sensitive * Need further refinements * Often local optimal

Geometric Primitives in LiDAR Point Clouds: A Review Shaobo Xia , Dong Chen , Member, IEEE, Ruisheng Wang , Senior Member, IEEE, Jonathan Li , Senior Member, IEEE, and Xinchang Zhang

RANSAC (RANdom SAmple Consensus)

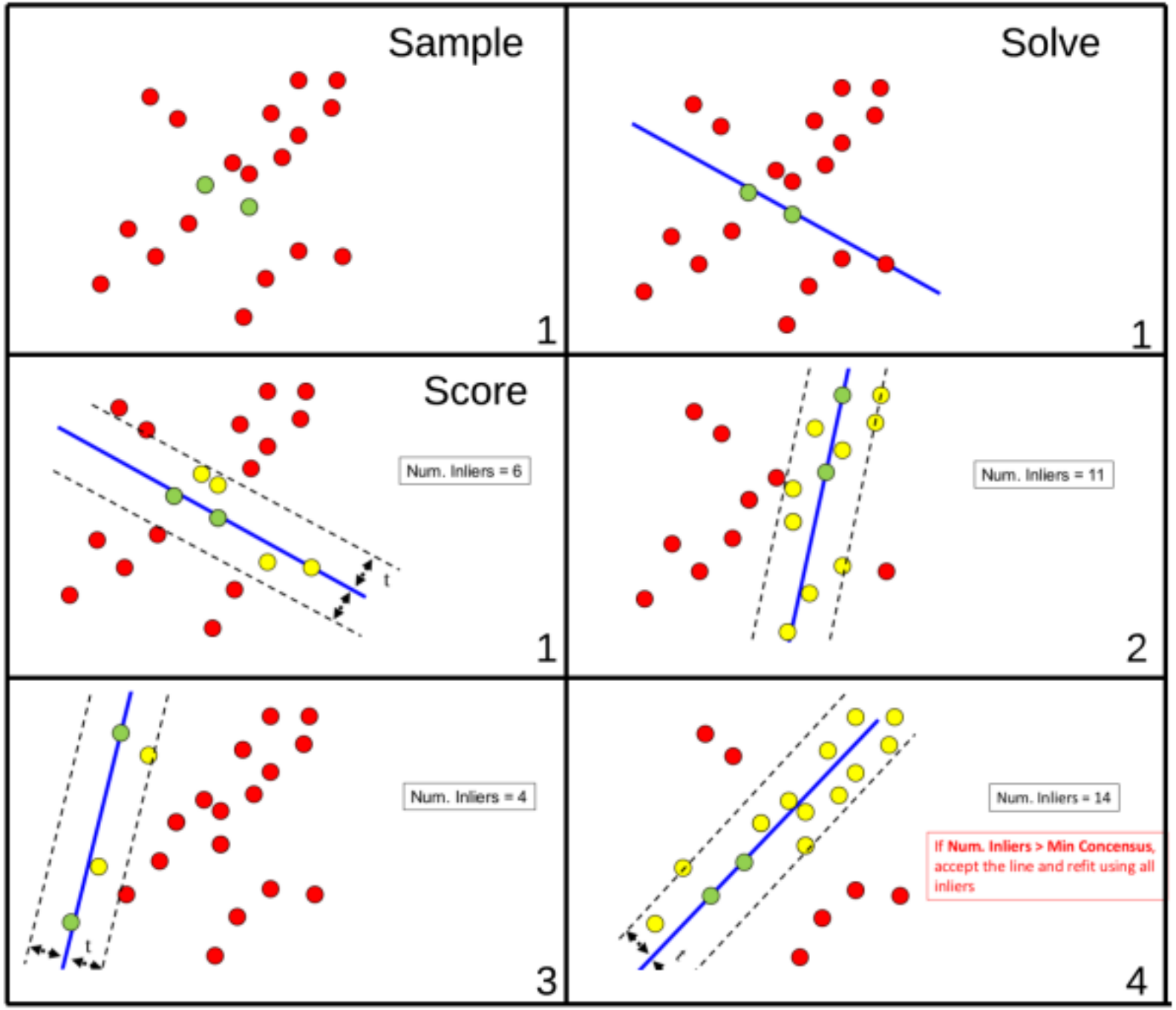
1. Sample (randomly) the number of points required to fit the primitive (in this case $n = 2$)
2. Solve for primitive parameters using samples (in this case m and q in $y = mx + q$)
3. Score by the fraction of inliers within a preset threshold δ of the primitive (in this case the number of points in the area)

- Repeat these 3 steps until the best primitive is found with high confidence
- Blue points = Inlier(s)
- Red points = Outlier(s) -> “bad” points

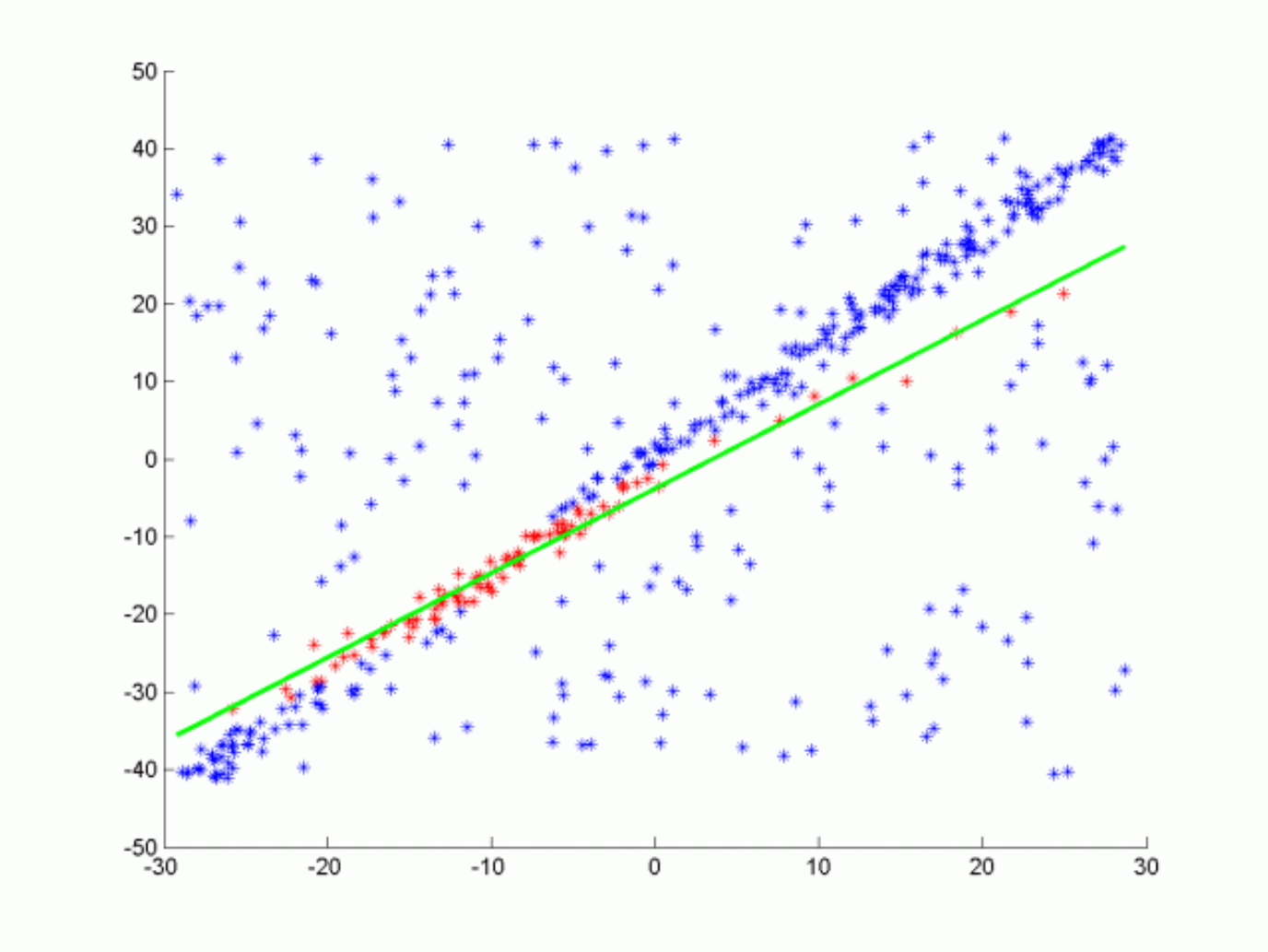


- Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." *Communications of the ACM* 24.6 (1981): 381-395.

RANSAC


Sample
1
Solve
1
Score
Num. Inliers = 6
t
1
Num. Inliers = 11
2
Num. Inliers = 4
t
3
Num. Inliers = 14
If Num. Inliers > Min Concensus, accept the line and refit using all inliers
4

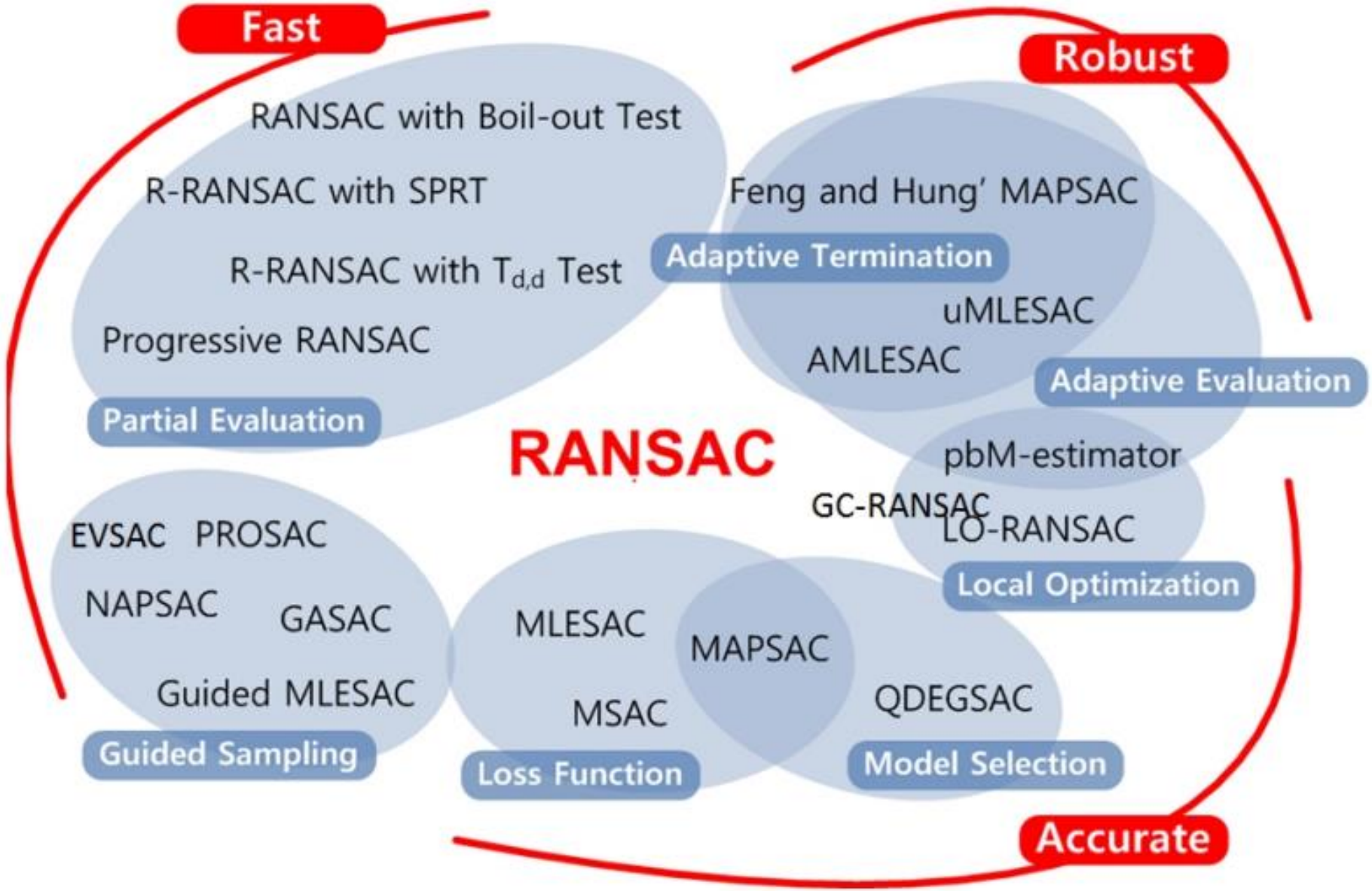
RANSAC (simulation)



RANSAC Algorithm in the context of geometric primitive detection in an oriented point cloud.

Algorithm 1 RANSAC

Input: List of vertices and associated normals
while there are too many unassigned vertices **do**
    for N times **do**
        Randomly select three vertices
        Compute the parameters of a primitive that sweeps them
        Compute the number of inliers for this primitive that are close enough to it
    end for
    Keep the parameters with the most inliers assigned to it and remove them from the pointcloud
end while
Output: List of primitive parameters and the corresponding inliers.



Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation Yuxing Xie, Jiaojiao Tian, Member, IEEE and Xiao Xiang Zhu, Senior Member, IEEE

Region Growing

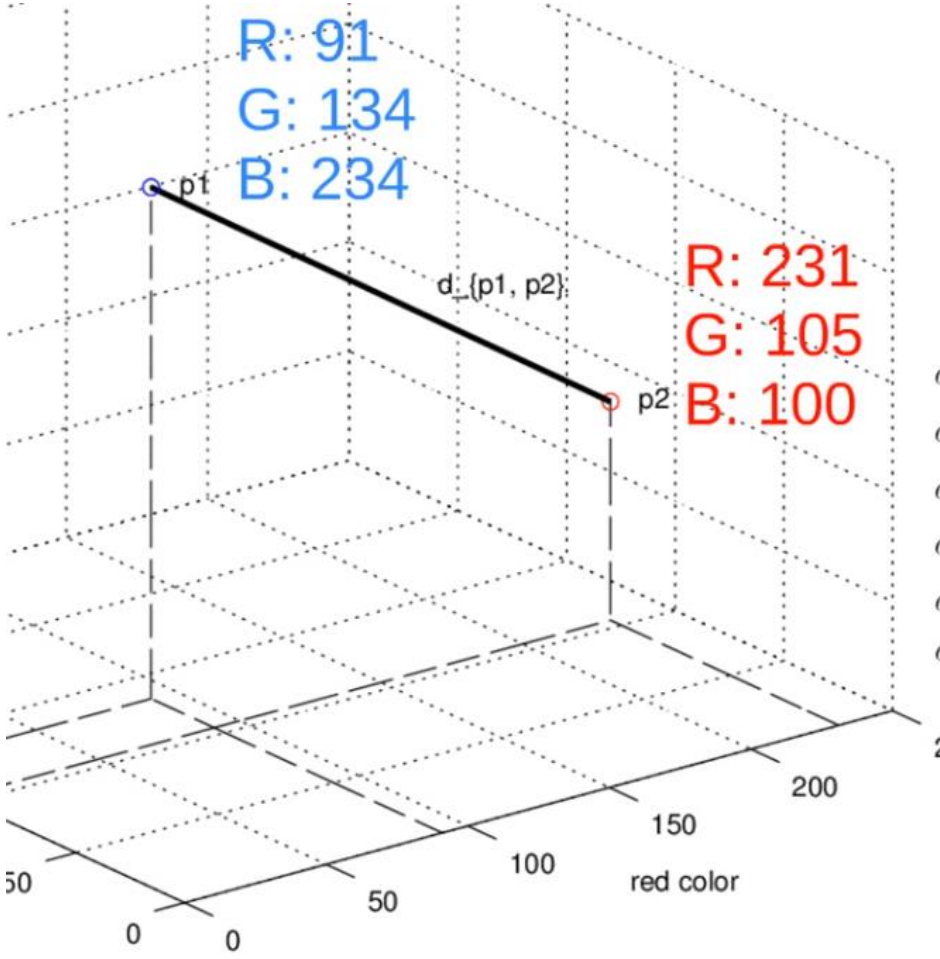
- **Clustering** method
- The region growing algorithm is used to extract connected components in a depth map or a point cloud with neighborhood information (e.g., **k-nearest neighbors**)
- A label is assigned to a seed sample and its neighbors are iteratively analyzed and assigned the seed's label if their **characteristics** are **similar enough** to the seed's

color, depth or normal orientation

are considered similar given a threshold T

[A Survey of Simple Geometric Primitives Detection Methods for Captured 3D Data Adrien Kaiser+,∗ Jose Alonso Ybanez Zepeda∗ Tamy Boubekeur+]

Region Growing (Example)

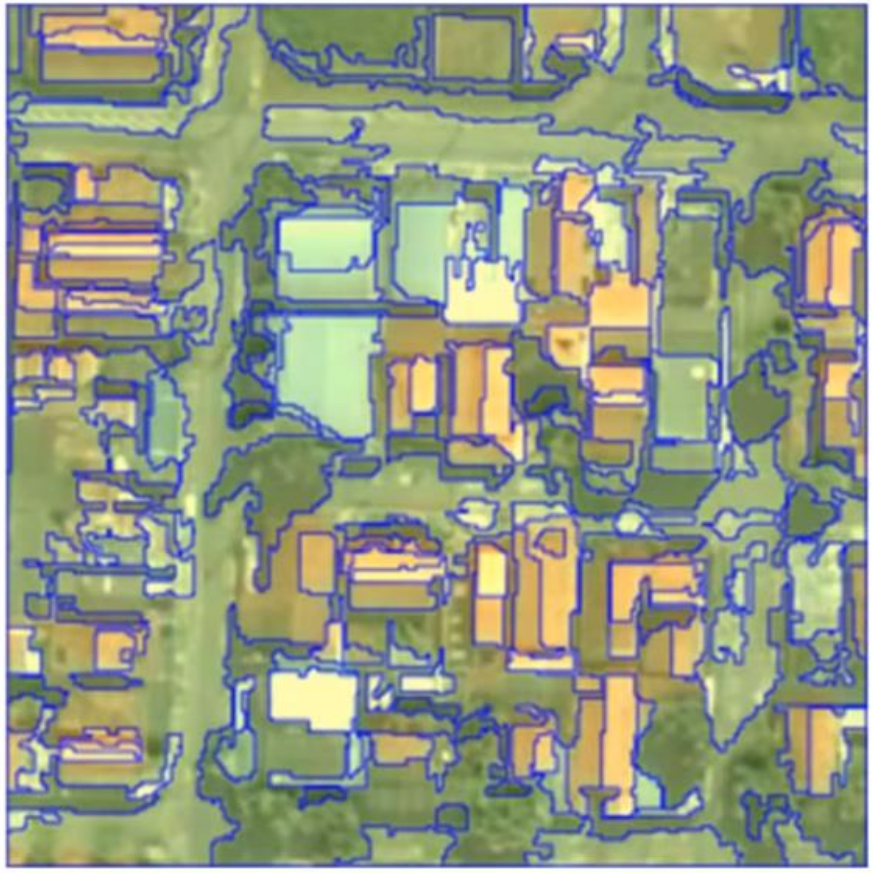


- Define distance between two pixels

$$d_{p_1,p_2} = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}$$
$$d_{p_1,p_2} = \sqrt{(91 - 231)^2 + (134 - 105)^2 + (234 - 100)^2}$$
$$d_{p_1,p_2} = \sqrt{(-140)^2 + (29)^2 + (134)^2}$$
$$d_{p_1,p_2} = \sqrt{19600 + 841 + 17956}$$
$$d_{p_1,p_2} = \sqrt{38397}$$
$$d_{p_1,p_2} \approx 195.951$$

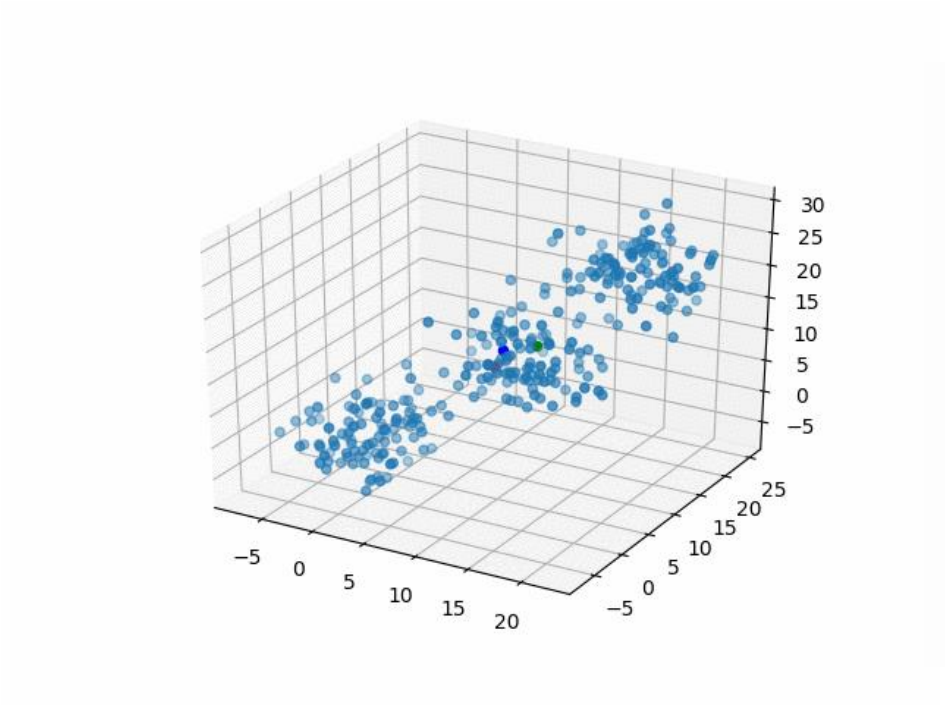
Region Growing (Example)

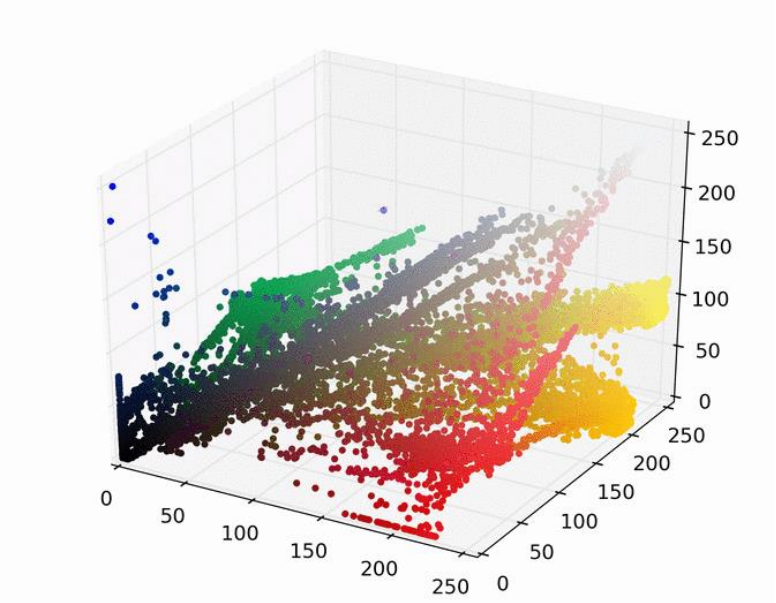
- Image segmentation using region growing. Haralick and Shapiro (1985)
- https://www.youtube.com/watch?v=VaR21S8ewCQ



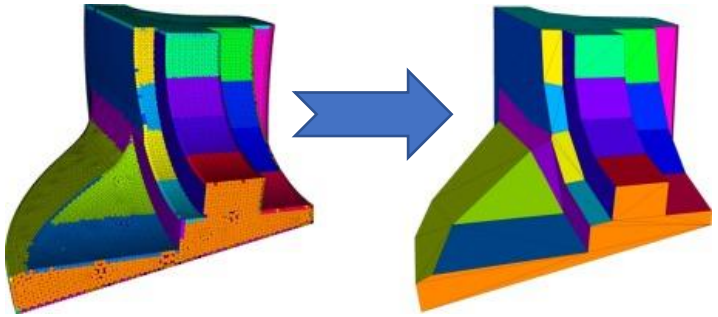
Region Growing: Automatic Clustering

- Automatic clustering methods in machine learning are often based on Lloyd's clustering algorithm [LLOYD 1982] (Lloyd-like automatic clustering): K-means clustering [MacQueen 1967], Mean Shift clustering [Fukunaga and Hostetler 1975], …

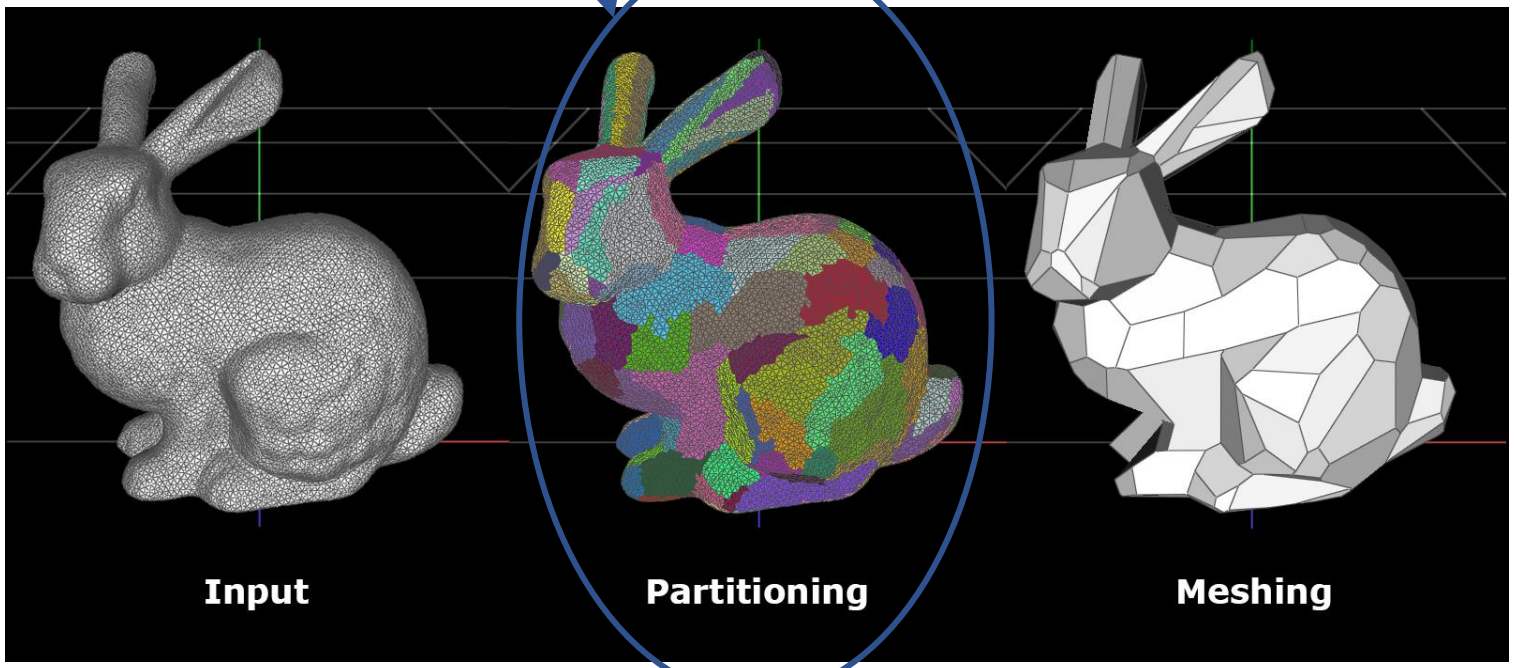




Automatic Clustering (Example)

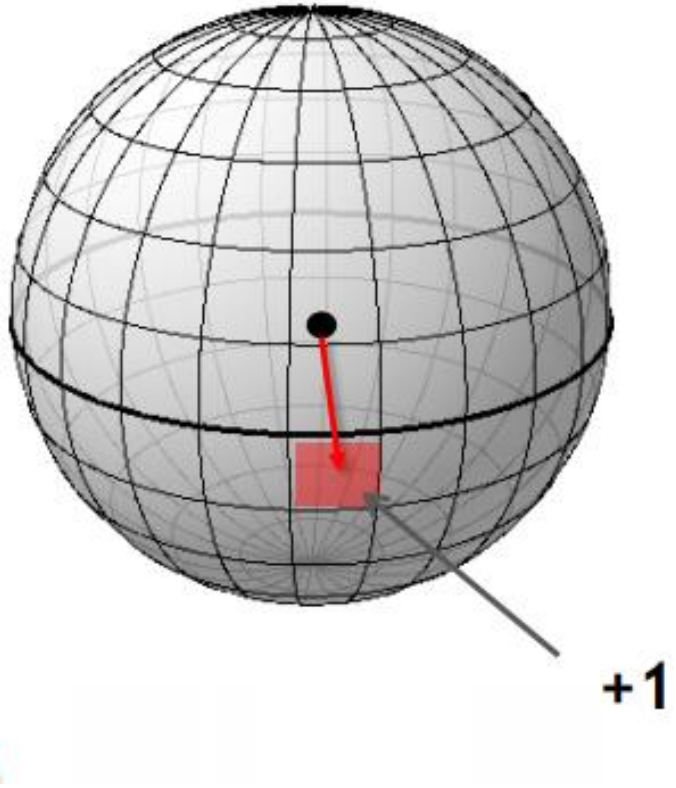


- Example of applications: *Variational Shape Approximation* (VSA)
- VSA aims to find a fixed number of planar proxies to simplify a meshed object with the best possible approximation



Accumulation methods: Gaussian sphere

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



https://team.inria.fr/titane/files/2015/04/surface_reconstruction_primitive-based.pdf

https://publik.tuwien.ac.at/files/PubDat_220568.pdf

Point Cloud Segmentation (Example)

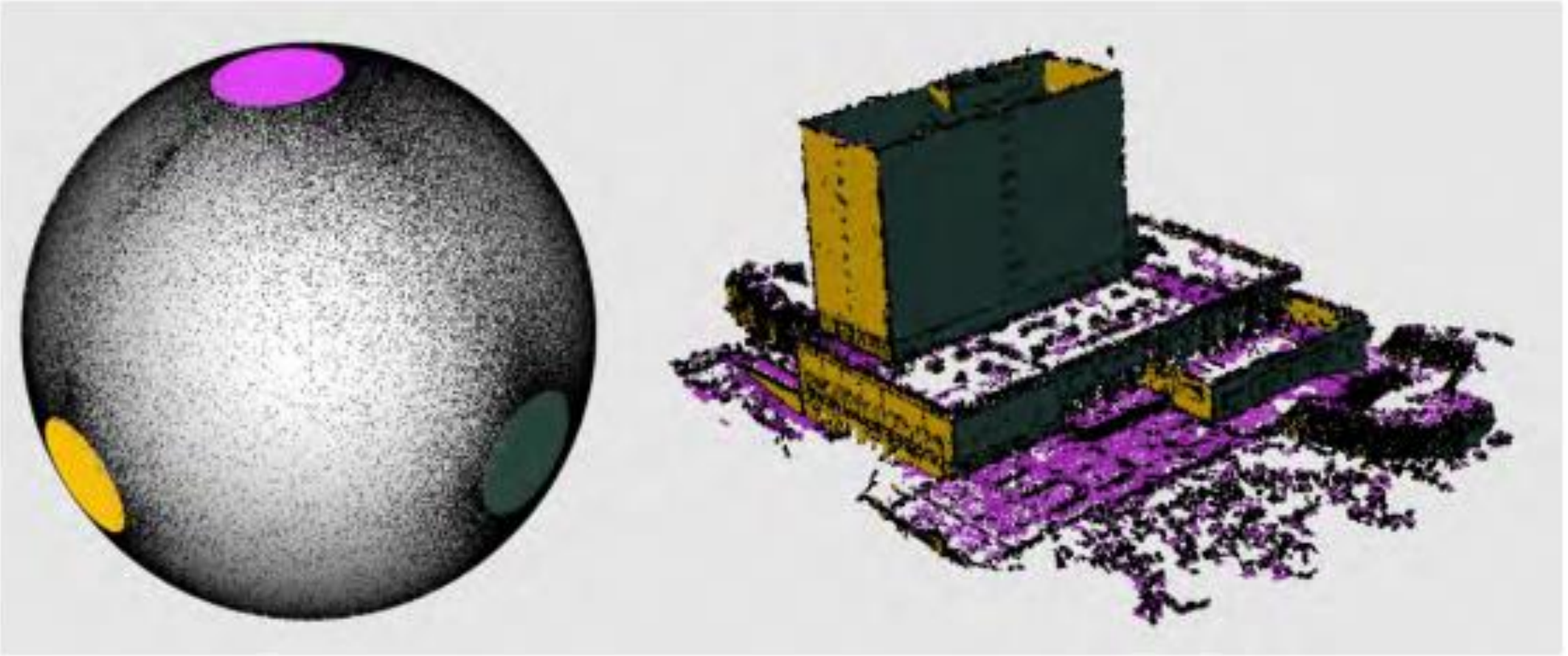
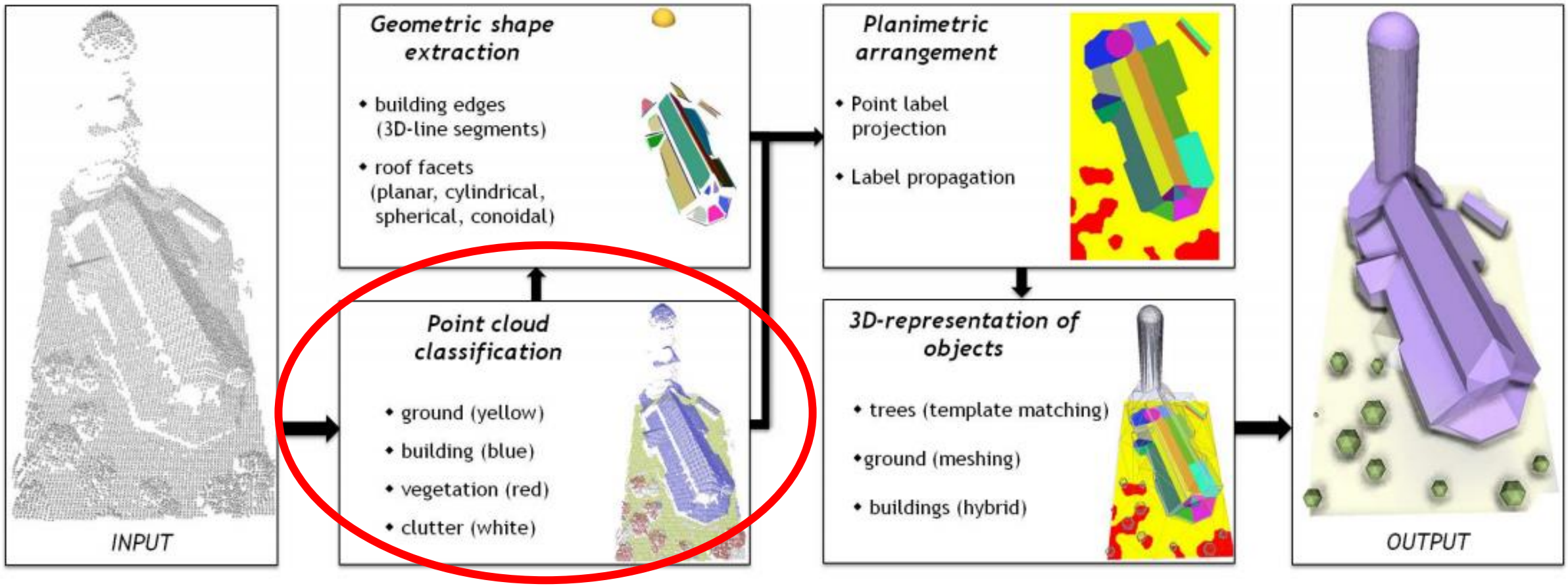


Figure 4.2: Three orthogonal main axes have been identified in this scene. The colored areas on the normal sphere (left image) denote all points belonging to one axis cluster. The clustered points are shown with the same colors in the 3D view (right image).

Segmentation then fitting (example)

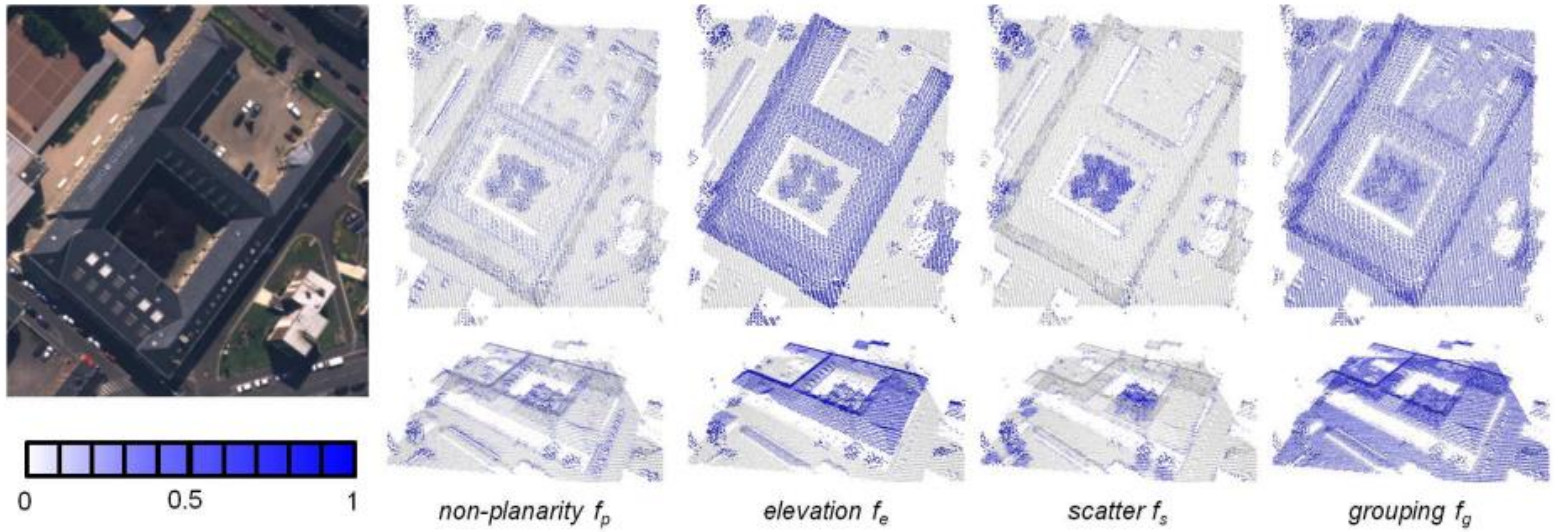


Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation Florent Lafarge, Clément Mallet

Another example: 3D All The Way: Semantic Segmentation of Urban Scenes From Start to End in 3D Andelo Martinovi ¯ c´ ♠1 Jan Knopp ♠1 Hayko Riemenschneider2 Luc Van Gool

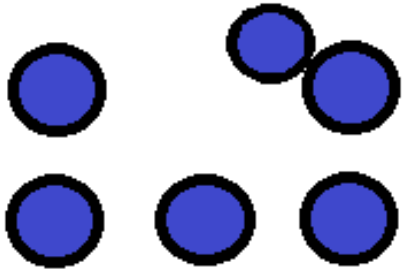
Segmentation then fitting (example)

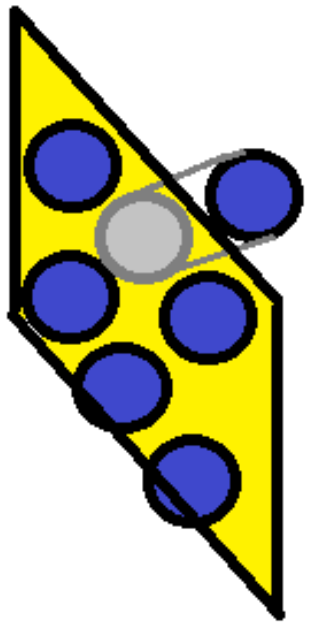
- Approach from: *Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation Florent Lafarge, Clément Mallet*
- 4 Labels: building, vegetation, ground and clutter (cars, fences, wires, roof antennas, cranes). Label water may be included in some applications.
- Discriminative features (in this work):
  - fp: Local non-planarity
  - fe: Elevation fe
  - fs: Scatter
  - fg: Regular grouping



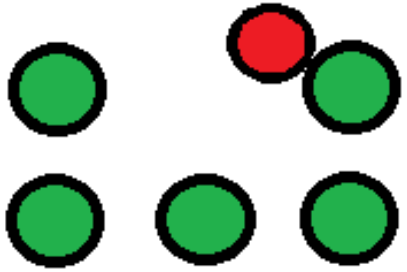
fp = Local non-planarity

- Represents the **quadratic distance** between the point and the optimal 3D-plane computed among its **neighbors**.
- Low values typically correspond to ground and building roofs.





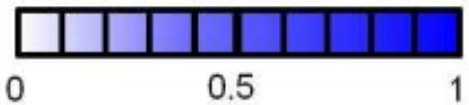
$$f_p = distance^2(P_i, LocalPlan_i)$$

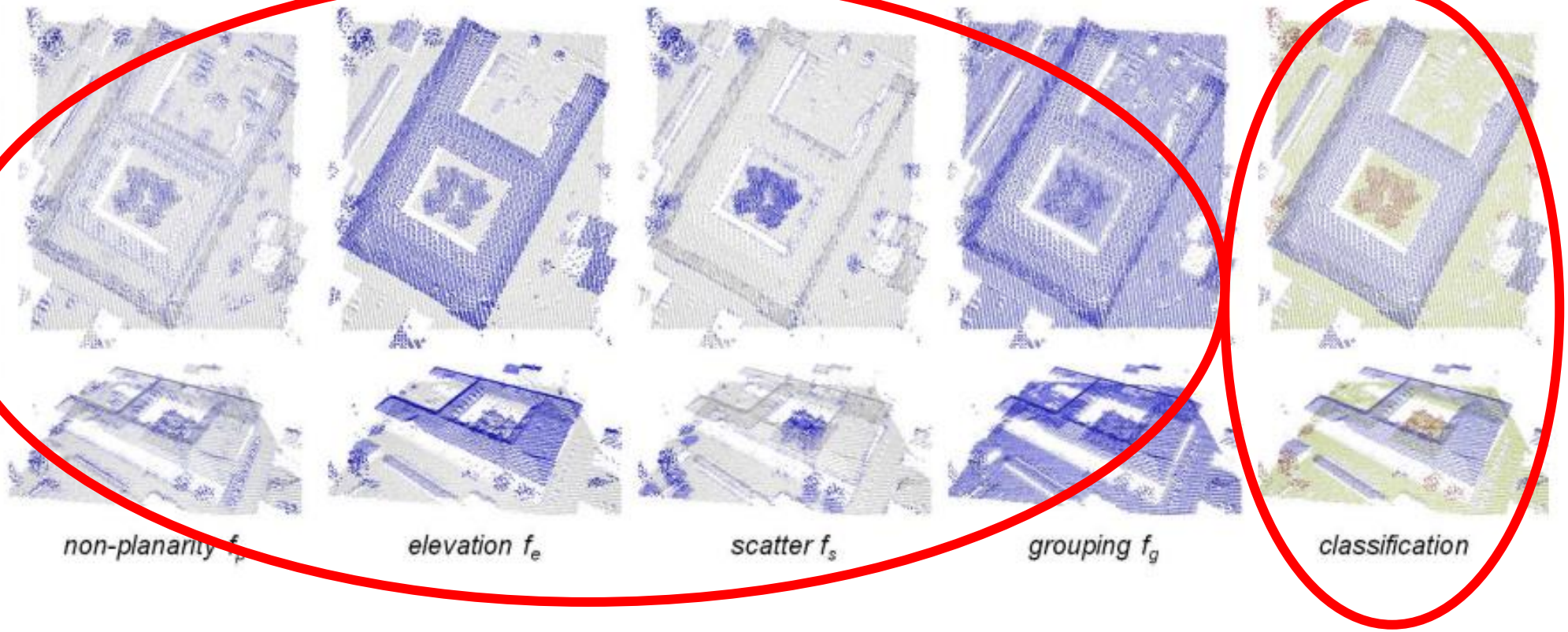


- **Low fp**
- **High fp**

Non-supervised classification

$$E_{di}(x_i) = \begin{cases} (1-f_e)\cdot f_p \cdot f_s & \text{if } x_i = building \\ (1-f_e)\cdot (1-f_p)\cdot (1-f_s) & \text{if } x_i = vegetation \\ f_e \cdot f_p \cdot f_s & \text{if } x_i = ground \\ (1-f_p)\cdot f_s \cdot f_g & \text{if } x_i = clutter \end{cases}$$



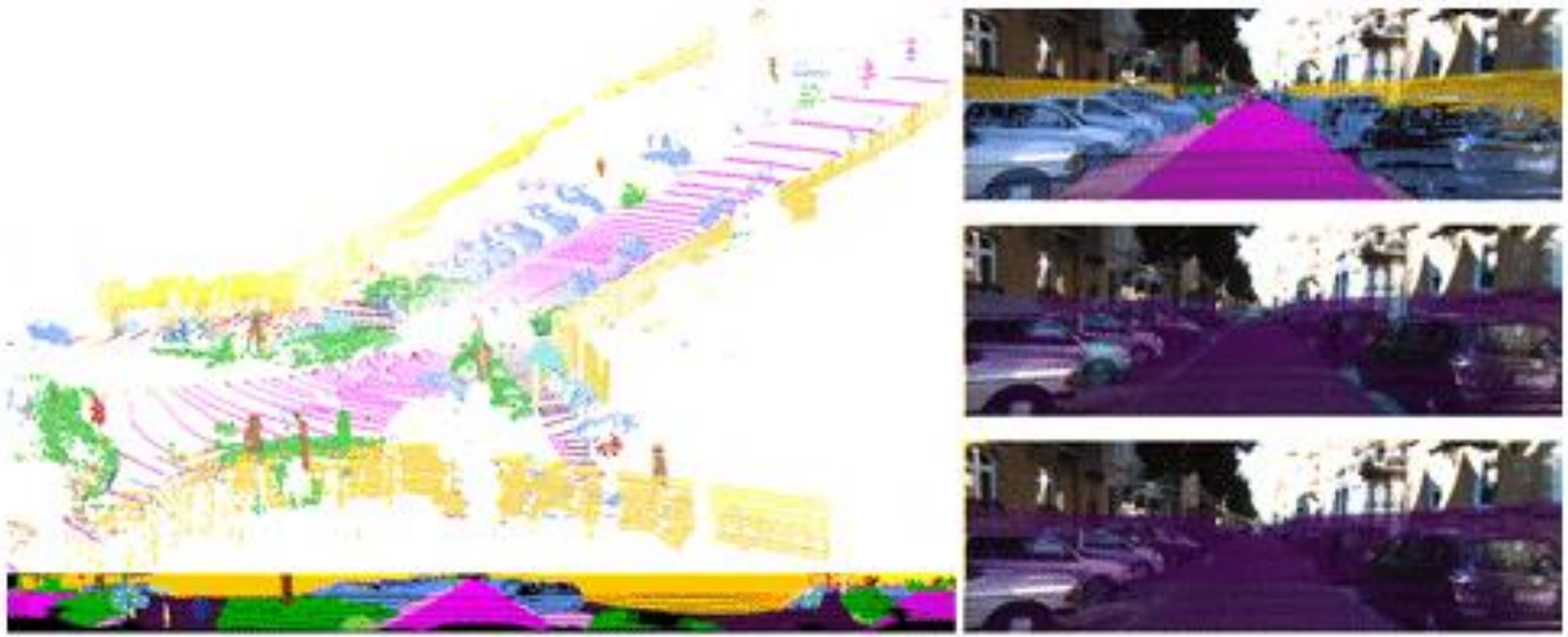


https://arxiv.org/pdf/1908.08854.pdf

PCS vs PCSS

- PCS (Point cloud segmentation): aims at grouping raw 3D points into non overlapping regions
  - Edge-based methods
  - region growing
  - model fitting (Hough Transform, RANSAC…)
  - Unsupervised clustering-based methods (K-means, Mean-Shift, …)
- PCSS (Point cloud semantic segmentation): aims at generating semantic information for every point, and are not limited to clustering
  - Regular supervised machine learning (No Deep-Learning)
  - Deep Learning
  - Hybrid methods

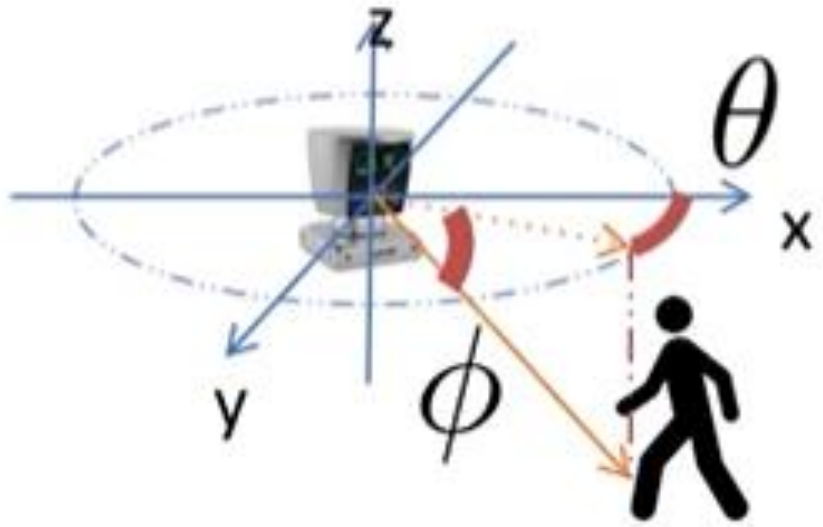
SalsaNext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds



Similar works at: https://aksoyeren.github.io/

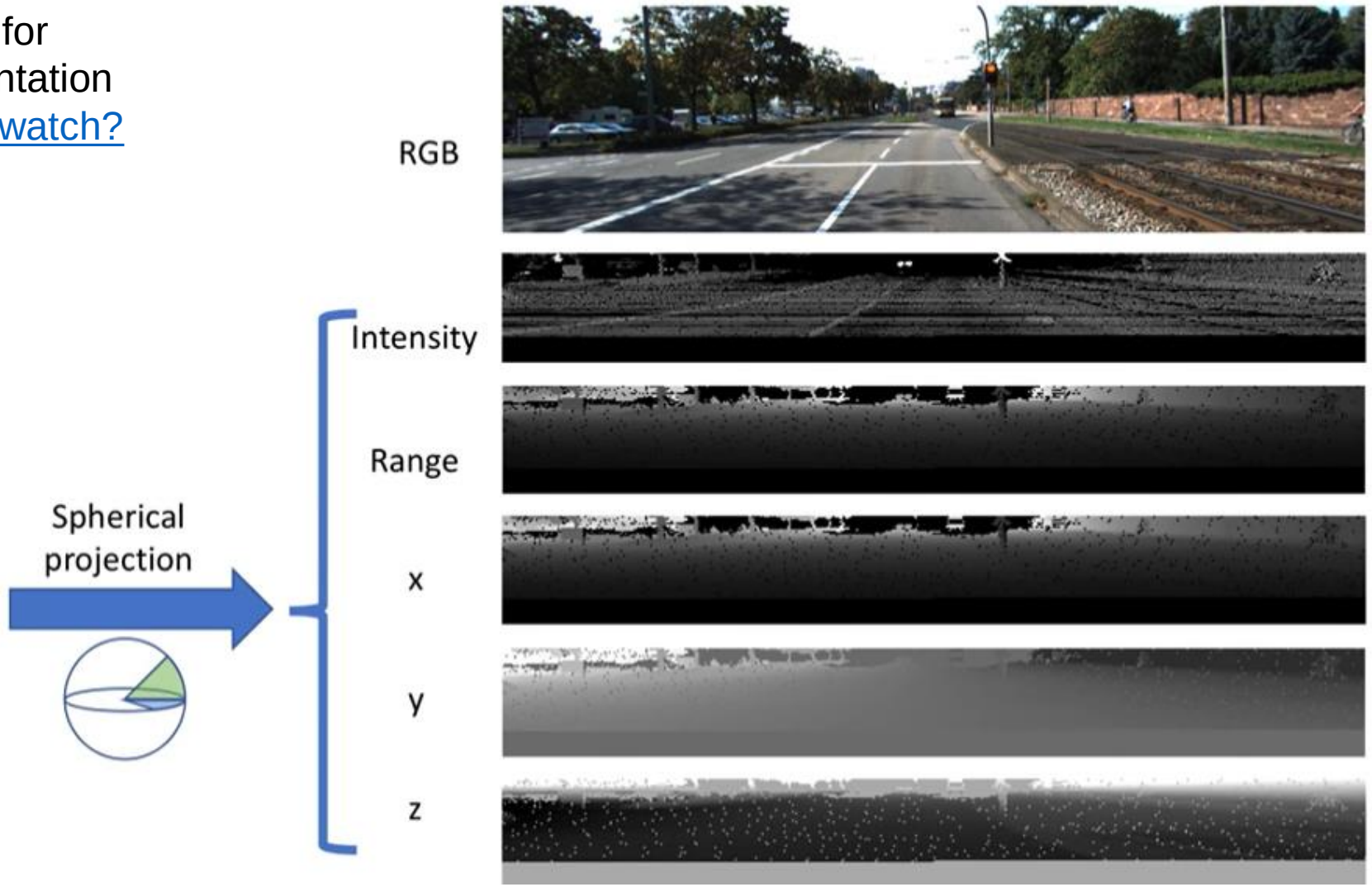
Spherical-Front-View (SFV)

- A point (x, y, z) → (i, θ, ϕ)
  - θ, ϕ angles, i intensity
- In the 2D spherical grid image, each point is mapped to the coordinates (u, v)
  - $u = \lfloor \theta / \Delta\theta \rfloor$
  - $v = \lfloor \phi / \Delta\phi \rfloor$
  - $\Delta\theta, \Delta\phi$ → discretization resolutions
- θ, ϕ are azimuth and zenith angles:
  - $\theta = \arcsin \frac{z}{\sqrt{x^2+y^2+z^2}}$, $\phi = \arcsin \frac{y}{\sqrt{x^2+y^2}}$

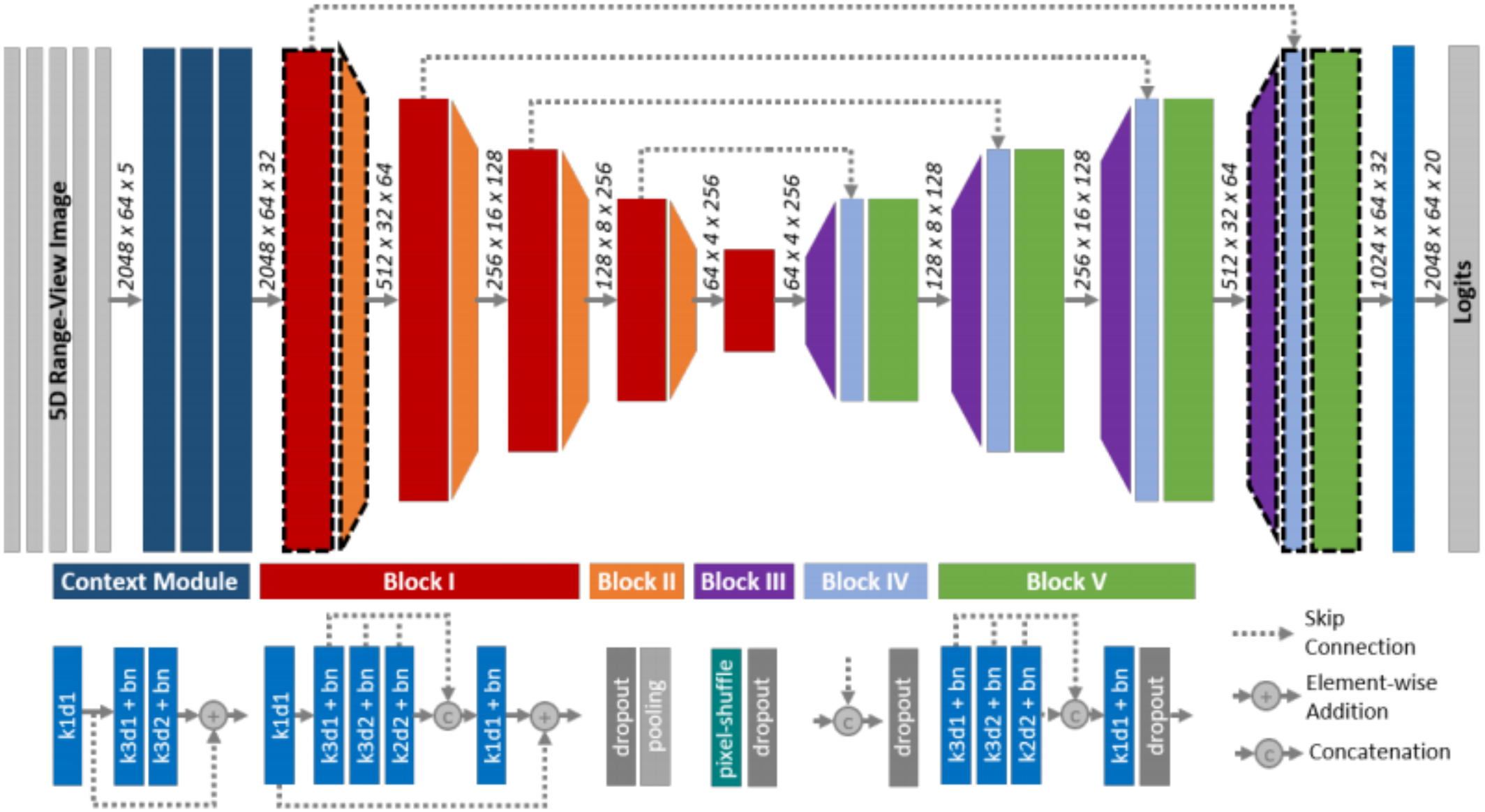


Multiple 2D channels

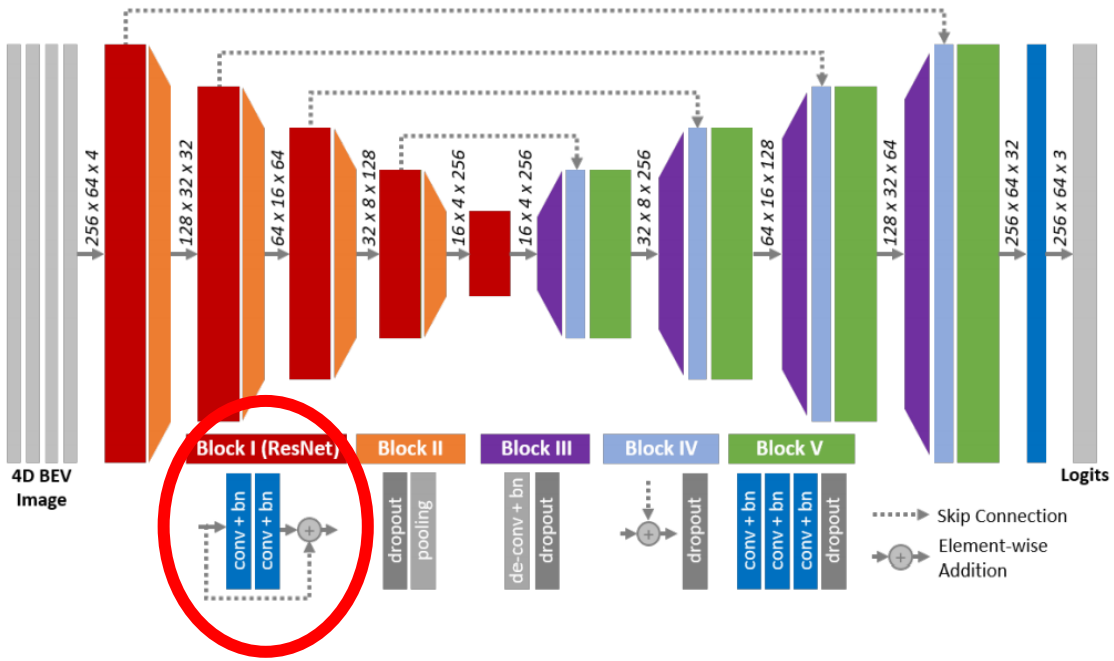
SqueezeSeg demo: CNN for LiDAR point cloud segmentation
https://www.youtube.com/watch?v=Xyn5Zd3lm6s

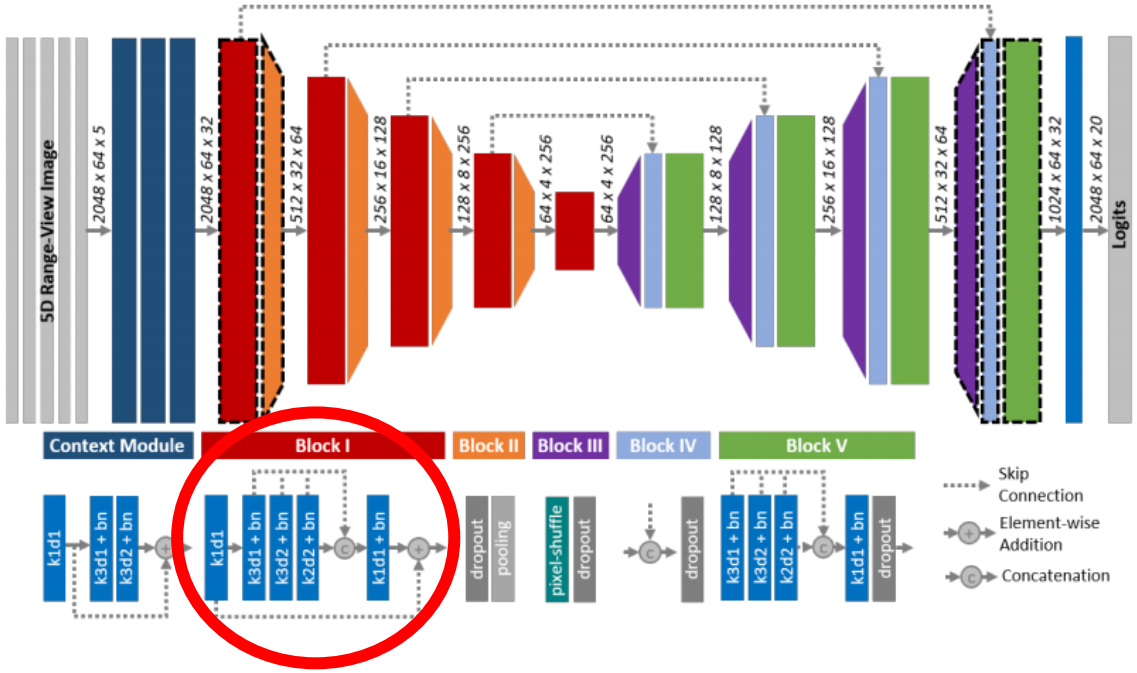


SalsaNext


5D Range-View Image
2048 x 64 x 5
2048 x 64 x 32
512 x 32 x 64
256 x 16 x 128
128 x 8 x 256
64 x 4 x 256
64 x 4 x 256
128 x 8 x 128
256 x 16 x 128
512 x 32 x 64
1024 x 64 x 32
2048 x 64 x 20
Logits
Context Module
Block I
Block II
Block III
Block IV
Block V
k1d1
k3d1 + bn
k3d2 + bn
k2d2 + bn
dropout
pooling
pixel-shuffle
Skip Connection
Element-wise Addition
Concatenation

ResNet (Residual Network)





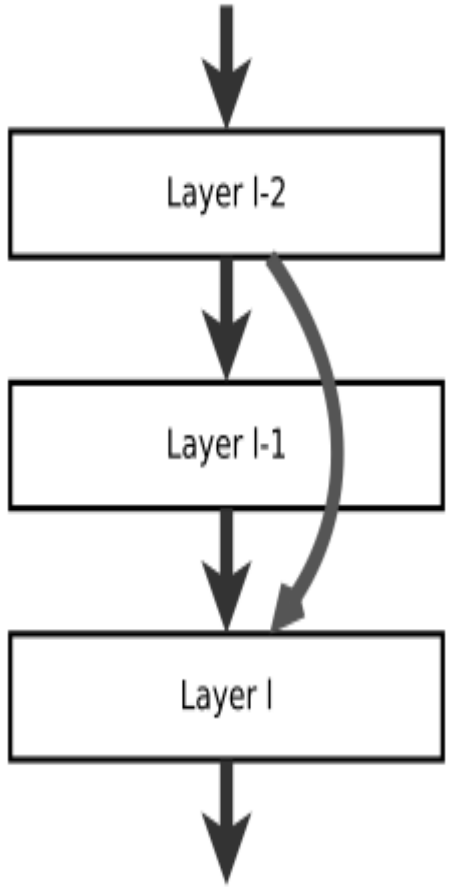
- Typical ResNet models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

ResNet (Residual Network)

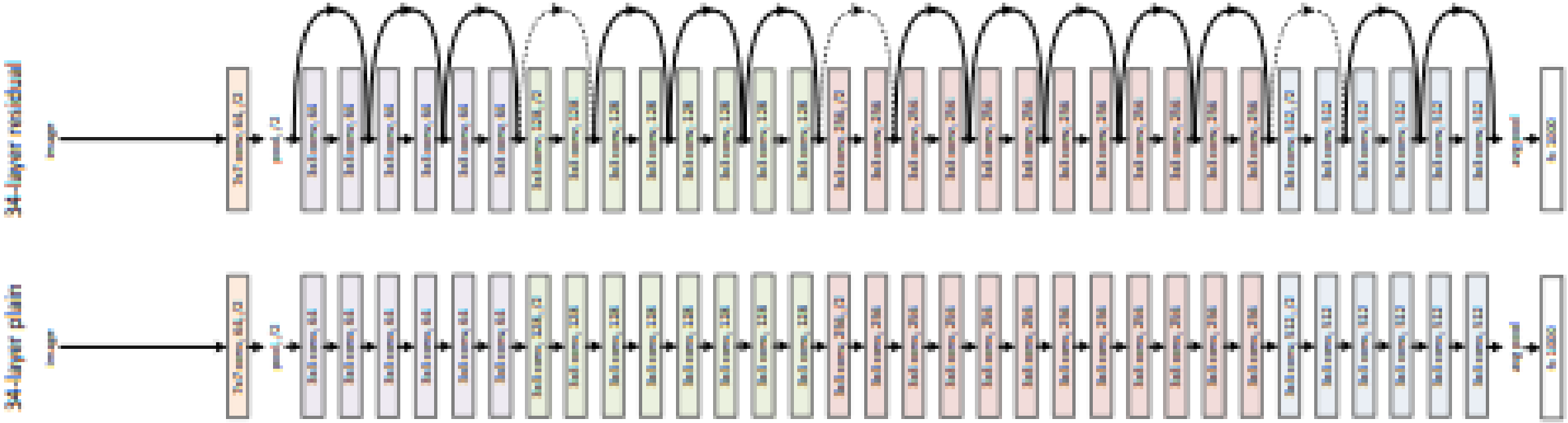
- Plain networks:
- a^i Input
- $z^{i+1} = W^{i+1}a^i + b^{i+1}$
- $a^{i+1} = g(z^{i+1})$
  - e.g., ReLU: $g(x) = x^+$
- $z^{i+2} = W^{i+2}a^{i+1} + b^{i+2}$
- $a^{i+2} = g(z^{i+2})$ Output

- Residual networks:
- a^i Input
- $z^{i+1} = W^{i+1}a^i + b^{i+1}$
- $a^{i+1} = g(z^{i+1})$
  - e.g., ReLU: $g(x) = x^+$
- $z^{i+2} = W^{i+2}a^{i+1} + b^{i+2}$
- $a^{i+2} = g(z^{i+2} + a^i)$ Output



K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

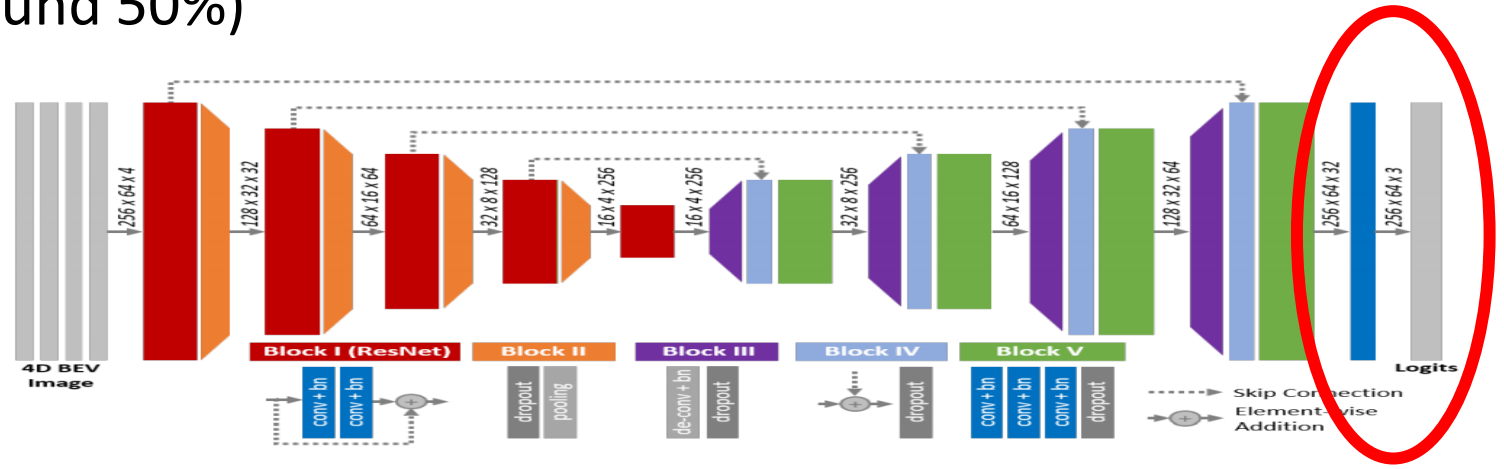
ResNet (Residual Network)



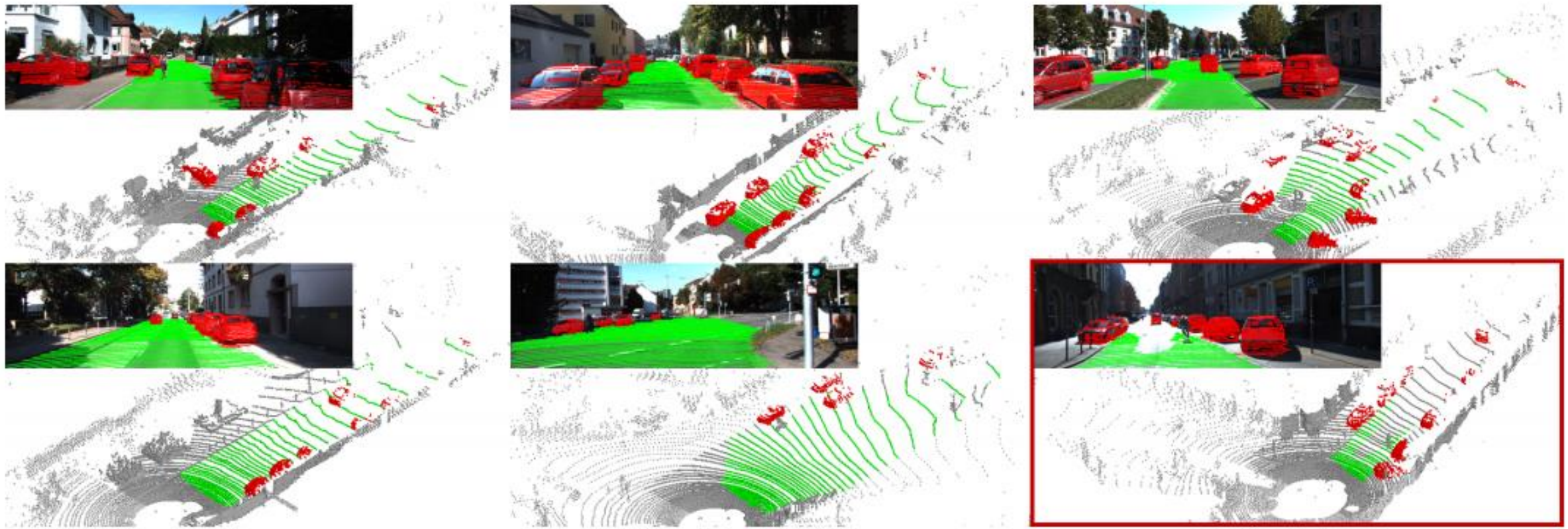
K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

Final layers

- The next layer applies 1×1 convolution to have **3 channels** which corresponds to the total number of **semantic classes** (i.e. road, vehicle, and background).
- Finally, the output feature map is fed to a **soft-max classifier** to obtain **pixel-wise** classification
  - Soft-max: assign a probability to each pixel (e.g., road 20%, vehicle 30%, background 50%)



Sample qualitative results of SalsaNet



Test Dataset: Semantic KITTI

- A large dataset to propel research on laser-based semantic segmentation
- Based on KITTI (Karlsruhe, Germany)

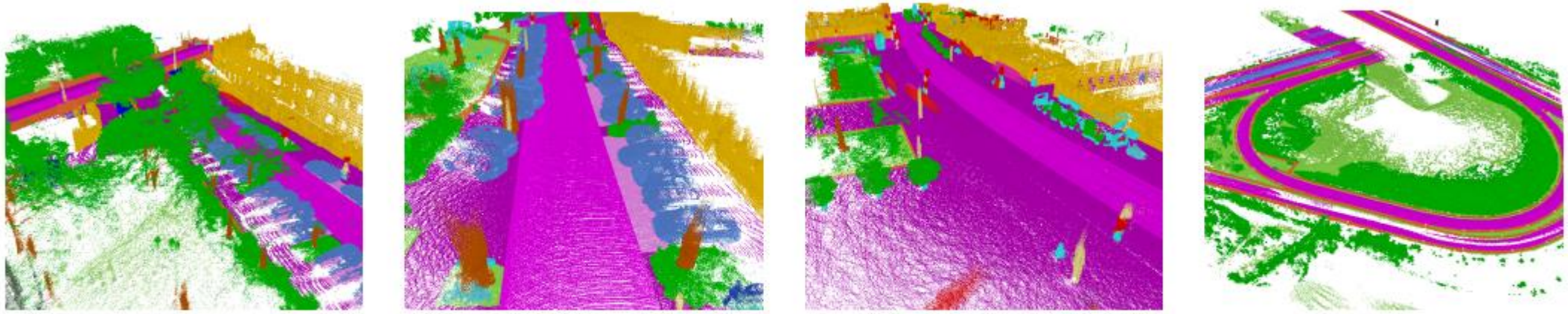
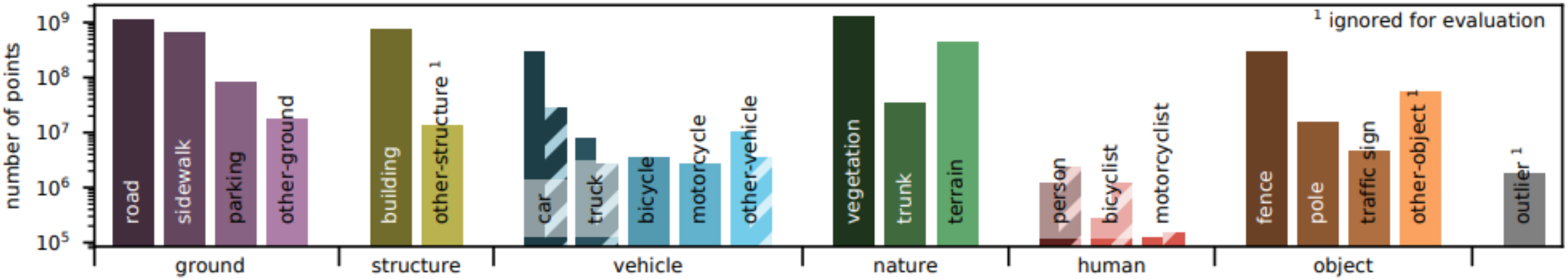
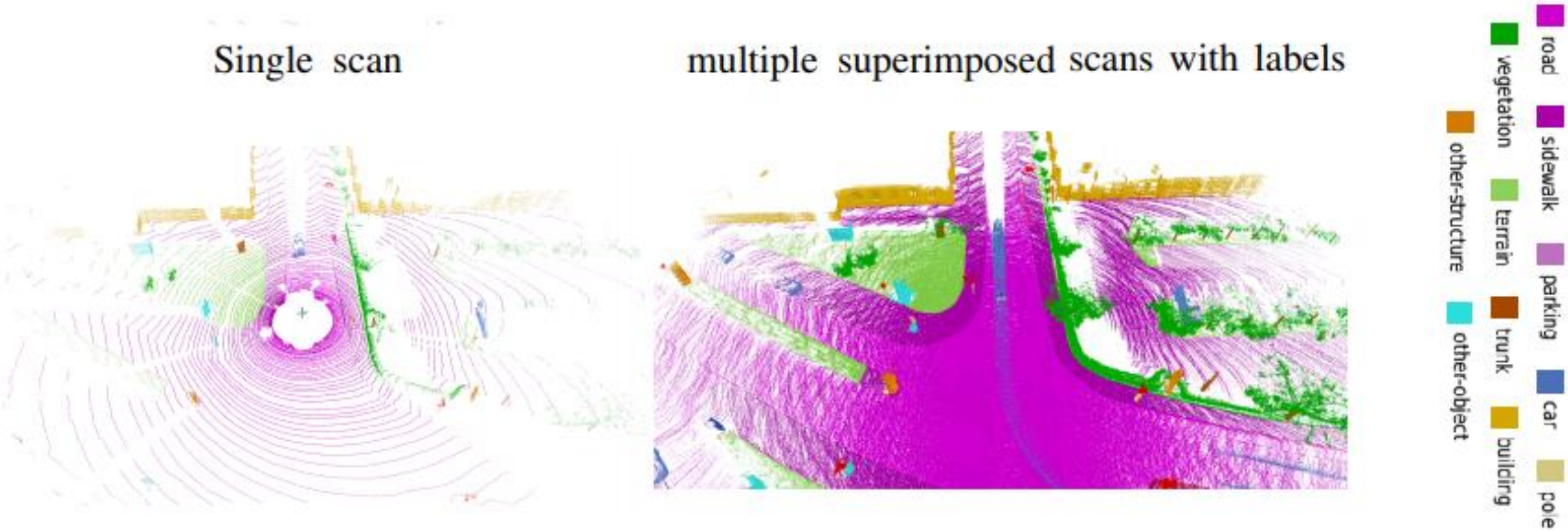


Figure 1: Our dataset provides dense annotations for each scan of all sequences from the KITTI Odometry Benchmark [19]. Here, we show multiple scans aggregated using pose information estimated by a SLAM approach.

SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences Jens Behley∗ Martin Garbade∗ Andres Milioto Jan Quenzel Sven Behnke Cyrill Stachniss Juergen Gall University of Bonn, Germany www.semantic-kitti.org


number of points
10^9
10^8
10^7
10^6
10^5
[1] ignored for evaluation
road
sidewalk
parking
other-ground
building
other-structure [1]
car
truck
bicycle
motorcycle
other-vehicle
vegetation
trunk
terrain
person
bicyclist
motorcyclist
fence
pole
traffic sign
other-object [1]
outlier [1]
ground
structure
vehicle
nature
human
object


Single scan
multiple superimposed scans with labels
road
sidewalk
parking
car
pole
building
trunk
terrain
vegetation
other-structure
other-object

Evaluation metrics

- To assess the labeling performance, we rely on the commonly applied mean Jaccard Index or mean intersection-over-union (mIoU) metric over all classes, given by:

$$\frac{1}{C}\sum_{c=1}^{C}\frac{\mathrm{TP}_c}{\mathrm{TP}_c+\mathrm{FP}_c+\mathrm{FN}_c}$$

- Where:
  - C number of all classes
  - TPc, FPc, and FNc correspond to the number of true positive, false positive, and false negative predictions for class c

Versione completa

Link alle slide complete (github): https://github.com/teobellu/pdf-collection/blob/main/A1.pptx

Download diretto: https://github.com/teobellu/pdf-collection/raw/main/A1.pptx