EuroSDR

European Spatial Data Research

# Deep Learning for Remote Sensing – EduServ Course

Sébastien Lefèvre, Loic Landrieu

Mars 2020

www.eurosdr.net

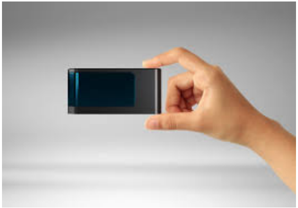
# Presentation outline

# Outline

# Presentation Layout

1. Deep Learning on 3D Point Clouds
   - Motivation
   - Traditional Approaches
   - Deep-Learning Approaches
   - Scaling Segmentation
   - In Practice
   - Bibliography

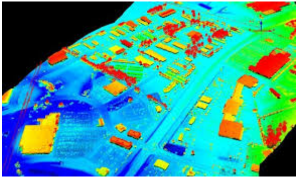
- LiDAR are getting cheaper :$100k\$ \rightarrow 2k\$$ in a few years.

**credit:** velodynelidar, green car congress

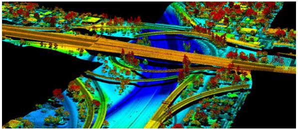
# An Exciting time for 3D Analysis

- LiDAR are getting cheaper :100$k$\$ $\rightarrow$ 2$k$\$ in a few years.
- Also coming: solid state LiDAR (cheap, fast and resilient), single photon LiDAR (unmatched acquisition density).

**credit:** velodynelidar, spar3d

- LiDAR are getting cheaper :$100k\$ \rightarrow 2k\$$ in a few years.
- Also coming: solid state LiDAR (cheap, fast and resilient), single photon LiDAR (unmatched acquisition density).
- **Major industrial application:** autonomous driving, virtual models, land survey...
- **Also to come:** major advances in automatic analysis of 3D data.
- Rapid progress in harware and methodology + major applications = **a booming field.**



**credit:** tuck mapping solutions, clearpath robotics

# Analysis of 3D point clouds

- **Classification**: classify the point cloud among class set $\mathcal{K}$:

$$P \mapsto \mathcal{K}$$

- **Partition**: cluster the point cloud in $C$ parts/object:

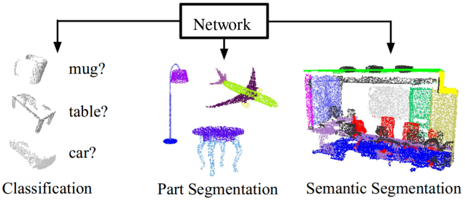
$$P_i \mapsto [1, \cdots, C]$$

- **Semantic Segmentation**: classify each point of a point cloud between $K$ classes:

$$P_i \mapsto [1, \cdots, K]$$

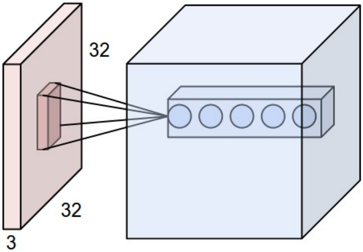
- **Instance Segmentation**: cluster the point cloud into semantically characterized objects:

$$P_i \mapsto [1, \cdots, C]$$
$$[1, \cdots, C] \mapsto [1, \cdots, K]$$



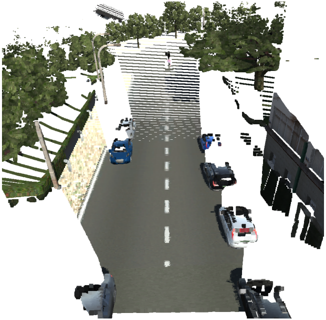

**credit:** Qi et. al. 2017a

- *Overheard at a vision conference:* "Computer Vision is a solved problem" [not my words].
- **Fact:** CNNs are very efficient at extracting visual features in images.
- **Fact:** has allowed tremedous improvements in many image-based tasks.
- **Fact:** In 2019, (almost) no vision paper on images rely on traditional methods.
- **When is this "revolution" coming for the 3D community?**




credit : pubs.sciepub.com/ajmm

# What Makes 3D Analysis so Hard?



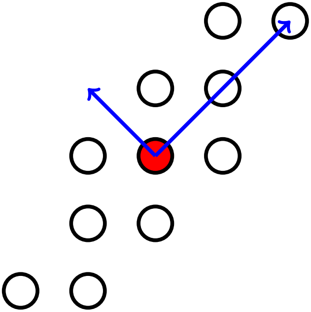
- **3D data is much harder than images:**
- Data volume considerable.
- Lack of grid-structure.
- Permutation-invariance.
- Sparsity.
- Highly variable density.
- Acquisition artifacts.
- Occlusions.
- Batching issues.

**credit:** Gaidon2016, Engelmann2017, Hackel2017

# Presentation Layout

- **Step 1:** compute point features based on neighborhood

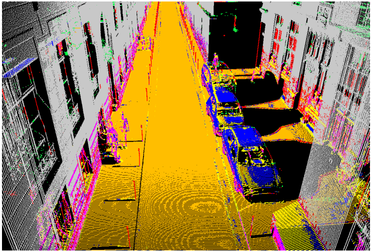
Demantke2011



$$\mathrm{Lin} = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1}}$$

$$\mathrm{Pla} = \frac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}}$$

$$\mathrm{Sca} = \frac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}}$$



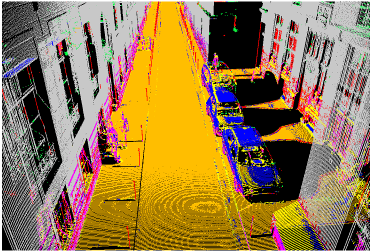
- **Step 1:** compute point features based on neighborhood
- **Step 2:** classification (RF, SVM, etc...)

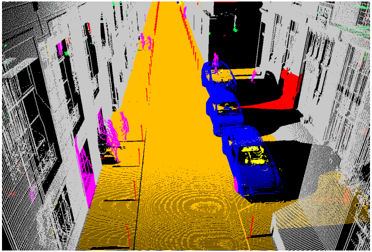
Weimann2015

**credit:** landrieu et. al. , JSPRS 2017

# Pointwise classification

- **Step 1:** compute point features based on neighborhood
- **Step 2:** classification (RF, SVM, etc...)
- **Step 3:** smoothing to increase spatial regularity (with CRFs, MRFs, graph-structured optimization, etc...)



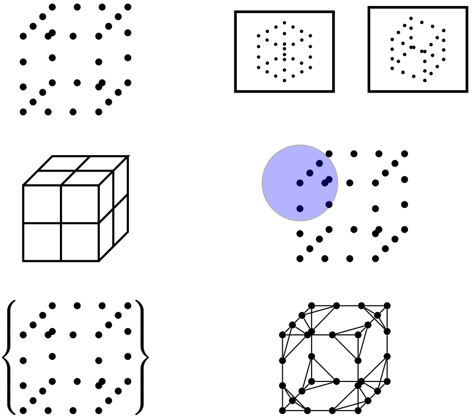


Landrieu et. al. 2017a JSPRS

# Presentation Layout

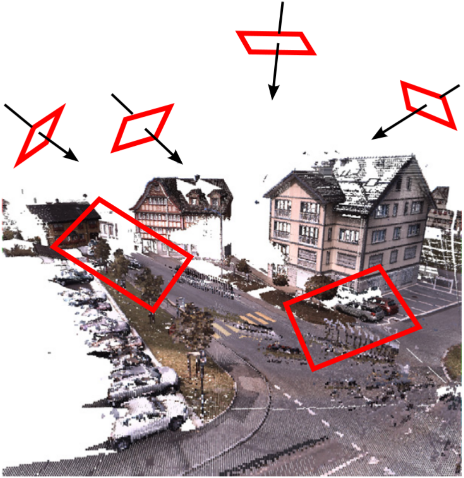
1. Deep Learning on 3D Point Clouds
   - Motivation
   - Traditional Approaches
   - Deep-Learning Approaches
   - Scaling Segmentation
   - In Practice
   - Bibliography

- Different ways of handling 3D point clouds:
- As a set of images,
- As a voxel occupancy grid,
- As a convolution-enabling structure,
- As an non-ordered set of points,
- As a graph structure.



- **Idea:** CNNs work great for images. Can we use images for 3D?
- **SnapNet:**
  - surface reconstruction,
  - *virtual snapshots*,
  - semantic segmentation of snapshots with CNNs,
  - project prediction back to point clouds.

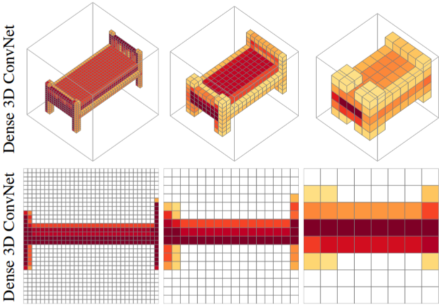
Boulch et. al. 2017



**credit:** Boulch et. al. 2017

# Voxel-Based Methods

- **Idea:** generalize image convolutions to 3D regular grids.
- Voxelization + 3D convNets **Guiotte2020**
- **Problem:** inefficient representation, loss of invariance, costly (cubic).

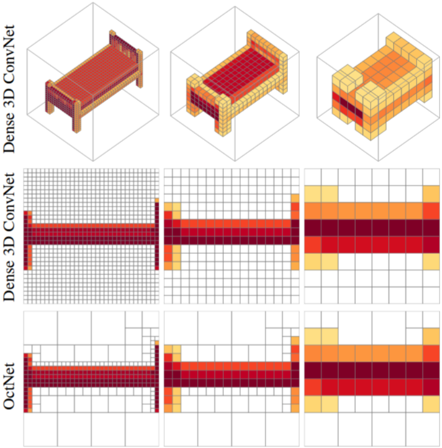



Wu2015

**credit:** Riegler2017, Tchapmi2017, Jampani2017

# Voxel-Based Methods

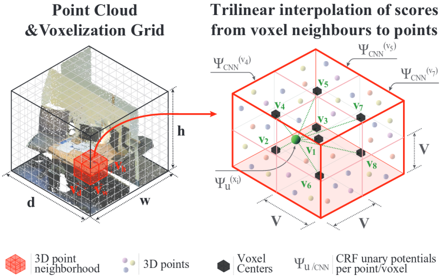
- **Idea:** generalize image convolutions to 3D regular grids.
- Voxelization + 3D convNets **Guiotte2020**
- **Problem:** inefficient representation, loss of invariance, costly (cubic).
- **Idea 1:** OctNet, OctTree based approach




Wu2015 , Riegler2017

**credit:** Riegler2017, Tchapmi2017, Jampani2017

- **Idea:** generalize image convolutions to 3D regular grids.
- Voxelization + 3D convNets **Guiotte2020**
- **Problem:** inefficient representation, loss of invariance, costly (cubic).
- **Idea 1:** OctNet, OctTree based approach
- **Idea 2:** SegCloud, large voxels, subvoxel predictions with CRFs.

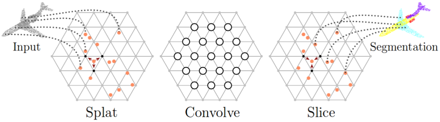



Wu2015 , Riegler2017 , Tchapmi2017, Jampani2018.

**credit:** Riegler2017, Tchapmi2017, Jampani2017

# Voxel-Based Methods

- **Idea:** generalize image convolutions to 3D regular grids.
- Voxelization + 3D convNets **Guiotte2020**
- **Problem:** inefficient representation, loss of invariance, costly (cubic).
- **Idea 1:** OctNet, OctTree based approach
- **Idea 2:** SegCloud, large voxels, subvoxel predictions with CRFs.
- **Idea 3:** SplatNet, MinkowskiNet sparse convolutions, indexed with hashmaps.

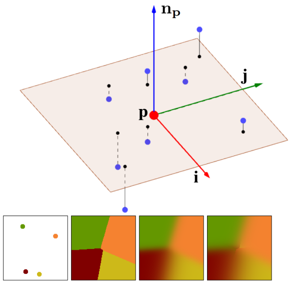



Wu2015 , Riegler2017 , Tchapmi2017, Jampani2018.

**credit:** Riegler2017, Tchapmi2017, Jampani2017

# 3D Convolution-Based Methods

- **Idea:** generalize 2D convolutions to *spatial* convolutions.
- **Tangent Convolution:** 2D convolution in the tangent space of each point.
- Can now used 2D convolutions to learn spatial features.



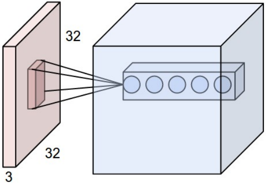

**credit:** Tatarchenko2018, Li2018

- Generalizing Spatial convolutions.

  $(i,j)$ point to embed, $f$ features, $K$ kernel, $c_d \in \mathbb{R}^d$ kernel weights.

  2D convolutions on images:

$$(\mathcal{C} * f)(i,j) = \sum_{d \in K} c_d f[(i,j) + d] \; .$$




credit : pubs.sciepub.com/ajmm

# 3D Convolution-Based Methods II

- Generalizing Spatial convolutions.

  $(i, j)$ point to embed, $f$ features, $K$ kernel, $c_d \in \mathbb{R}^d$ kernel weights.

  2D convolutions on images:

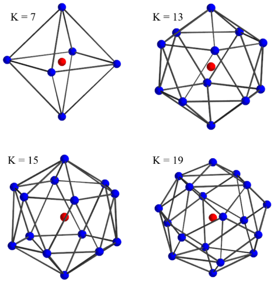
$$(\mathcal{C} * f)(i, j) = \sum_{d \in K} c_d f[(i, j) + d] \ .$$

- $x$ spatial coordinates, $N_x$: neighbors of $x$, $y_j \in K$ spatial coordinate of kernel points.

  3D convolutions on point clouds:

$$(\mathcal{C} * f)(x) = \sum_{j \in K} \sum_{i \in N_x} g(x_i, y_j) c_j f_i \ .$$

- Kernels can be fixed or learnt.
- $g(x, y) = \max\left(0, 1 - \frac{1}{\sigma}\|x - y\|\right)$.
- Or $g$ can be a learnt function as well.




**credit:** Thomas etal. 2019, Boulch 2019

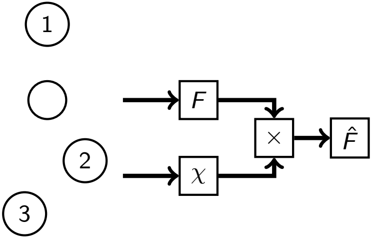
# 3D Convolution-Based Methods III

- **PointCNN :** $\chi$-convolutions: generalized convolutions for unordered inputs.
- **Principle:** the network learns how to permute *ordered* inputs:

  $x =$ concatenated features of neighbors sorted by distance.

  $F, \chi = f(x)$.

  $\hat{F} = F \times \chi$.
- The invariance is learnt.

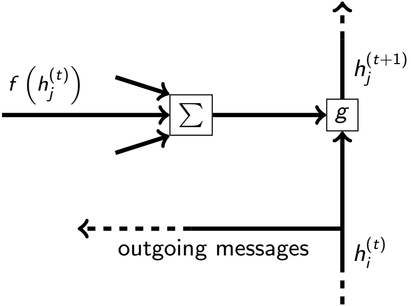



**credit:** Tatarchenko2018

# Graph-Neural Network

- **Idea:** Generalize convolutions to the general graph setting.
- For example: k-nearest neighbors graph.
- **Principle:** Each point maintain a hidden state $h_i$ influenced by its neighbors.
- GNN **Qi2017**: iterative message-passing algorithm witha mapping $f$ and RNN $g$:

$$h_i^{(t+1)} = g(\sum_{j \to i} f(h_i^{(t)}), h_i^{(t)}).$$



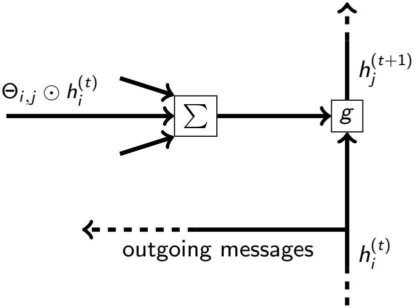

Qi2017, Simonovski2017

- **Idea:** Generalize convolutions to the general graph setting.
- For example: k-nearest neighbors graph.
- **Principle:** Each point maintain a hidden state $h_i$ influenced by its neighbors.
- GNN **Qi2017**: iterative message-passing algorithm witha mapping $f$ and RNN $g$:

$$h_i^{(t+1)} = g(\sum_{j \to i} f(h_i^{(t)}), h_i^{(t)}).$$

- ECC **Simonovski2017** messages are conditioned by edge features:

$$h_i^{(t+1)} = g(\sum_{j \to i} \Theta_{i,j} \odot h_i^{(t)}, h_i^{(t)}).$$

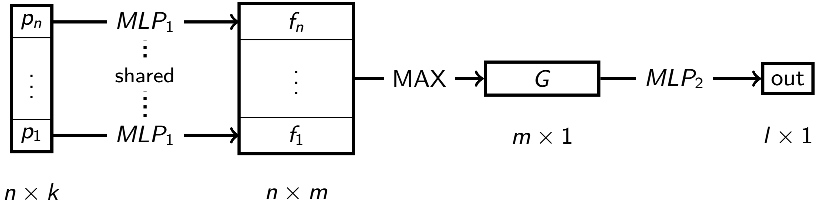



Qi2017, Simonovski2017

# PointNet - Set-Based Approach

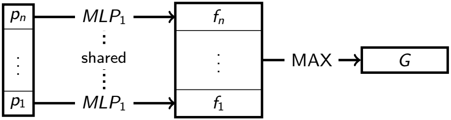
- A cornerstone of modern 3D analysis.
- **A fondamental constraint:** permutation-invariant inputs.
- **Solution:** $n$: number of points, $k$ size of observations, $m$ size of intermediary embeddings, $l$ size of desired output:

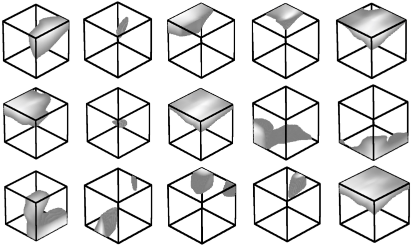
- Independantly compute point features $\{f_i\}_{i=1}^n$ through function $MLP_1 : \mathbb{R}^k \mapsto \mathbb{R}^m$
- Computes permutation invariant global embedding $[G]_j = \max_{i=1\cdots n}[f_i]_j, j = 1 \cdots m$
- Process this global with function with $MLP_3 : \mathbb{R}^m \mapsto \mathbb{R}^l$



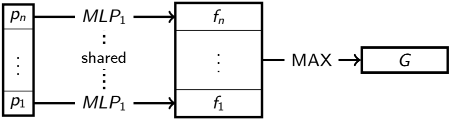

Qi et. al.2017a

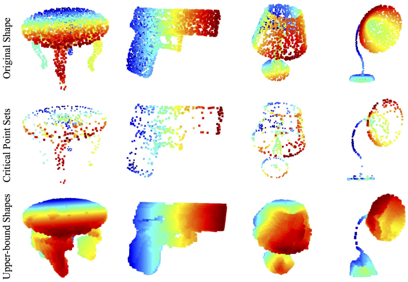
- **Point function**: Visualization of $[MLP_1]_j$ for some $j$. the full part represents high values ($\geq 0.5$) in the unit cube.
- Each coordinates of $MLP_1$ is sensitive to different positions: acts as a learned spatial descriptor.



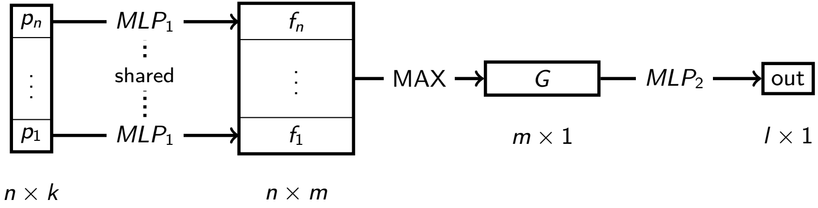
- **Point function**: Visualization of $[MLP_1]_j$ for some $j$. the full part represents high values ($\geq 0.5$) in the unit cube.
- Each coordinates of $MLP_1$ is sensitive to different positions: acts as a learned spatial descriptor.
- **Critical Set**: points with an influence in $G$. Makes up a shape *skeletton*.
- **Upper Bound Shape**: maximal point cloud with exactly the same global embedding $G$.
- All 3 shapes have the exact same embedding.



# PointNet for Cloud embedding

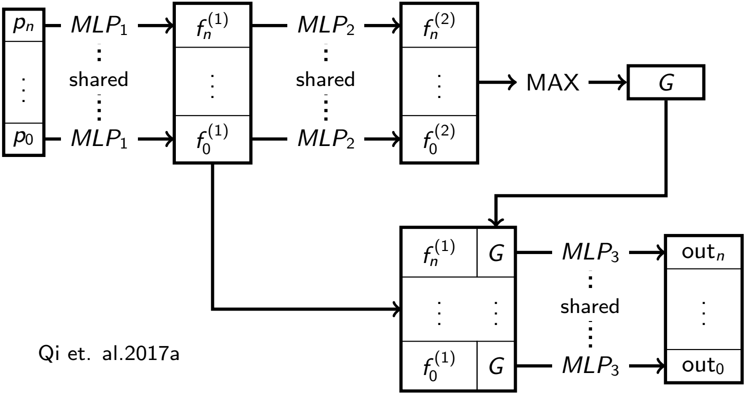
- $p_i \in \mathbb{R}^k$: point features (coordinate, color, etc...)
- $f_i = MLP_1(p_i) \in \mathbb{R}^m, \forall i = 1 \cdots n$
- $G_j = \max_{i=1\cdots n}([f_i]_j) \; \forall j = 1 \cdots m$
- $\text{out} = MLP_2(G) \in \mathbb{R}^l$




Qi et. al.2017a

# PointNet for Semantic Segmentation

- $f_i^{(1)} = MLP_1(p_i) \in \mathbb{R}^{m_1} \ \forall i = 1 \cdots n$
- $f_i^{(2)} = MLP_2(f_i^{(1)}) \in \mathbb{R}^{m_2} \ \forall i = 1 \cdots n$
- $[G_j]_j = \max_{i=1 \cdots n}([f_i^{(2)}]_j) \ \forall j = 1 \cdots m_2$
- $\text{out}_i = MLP_3([G, f_i^{(1)}]) \in \mathbb{R}^{l} \ \forall i = 1 \cdots n$

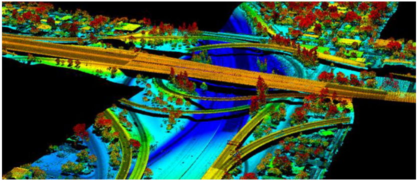



Qi et. al.2017a

# Presentation Layout

# Why we need to scale

- **Problem:** best approaches are very memory-hungry and the data volumes are huge.
- Previous methods only works with a few thousands points.
- **Naive strategies:**
  - **Aggressive subsampling**: loses a lot of information.
  - **Sliding windows**: loses the global structure.



**credit:** tuck mapping solution

# PointNet++

- Pyramid structure for multi-scale feature extraction.
- Initialization:

$$f_i^{(0)} = \text{initial features: xyz, RGB, etc...}$$

- Spatial Pooling:

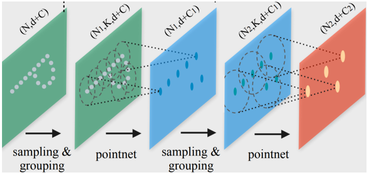
$$P^{(t+1)} = \text{subsample}(P^{(t)})$$

- Correspondence

$$N_i = \text{neighborhood of } i \in P^{(t+1)} \text{ in } P^{(t)}$$

- Next layer:

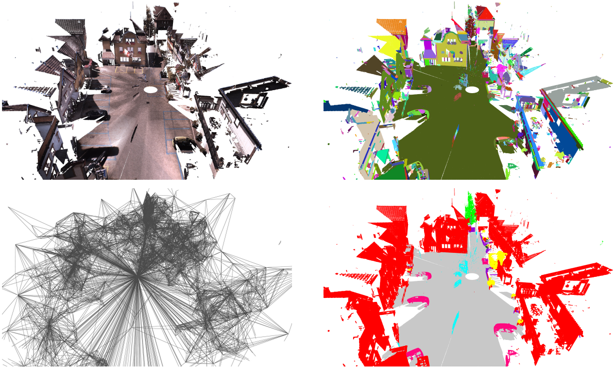
$$f_i^{(t+1)} = PointNet^{(t)}\left(\{f_j^{(t)}\}_{j \in N_i}\right)$$




- From local to global with with increasingly abstract features.

**credit:** Qi et. al.2017b

- **Observation:** $n_{\text{points}} \gg n_{\text{objects}}$.
- Partition scene into superpoints with simple shapes.
- Only a few superpoints, context leveraging with powerful graph methods.

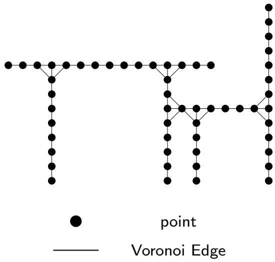


Landrieu&Simonovski, CVPR 2018

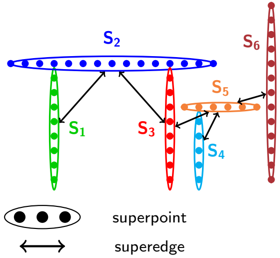
# Pipeline

| Step | Complexity | Algorithm |
|---|---|---|
| **Geometric Partition**<br>into simple shapes | very high<br>$10^8$ points | $\ell_0$-cut pursuit |
| **Superpoint embedding**<br>learning shape descriptors | low<br>subsampling to 128 points | PointNet |
| **Contextual Segmentation**<br>leveraging the global structure | very low<br>$\sim$ 1000 vertices | ECC<br>with GRUs |

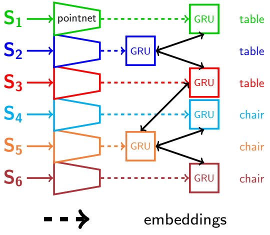
# Pipeline



(a) Point cloud



(b) Superpoint graph



(c) Convolution Network

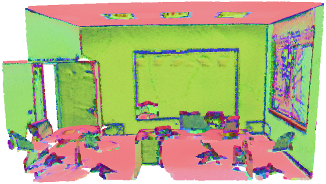
# Superpoint Partition

$$f^\star = \text{argmin}_{f \in \mathbb{R}^{C \times m}} \sum_{i \in C} \|f_i - e_i\|^2 + \sum_{(i,j) \in E} w_{i,j}[f_i \neq f_j],$$

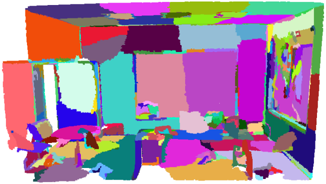
- $e \in \mathbb{R}^{C \times m}$ : handcrafted descriptors of the local geometry/radiometry.
- $f^\star$ : piecewise constant approximation of $e$.
- Solutions can be efficiently approximated with $\ell_0$-cut pursuit.
- Superpoints: constant connected components of $f^\star$.
- **Problem:** any errors made in the partition will carry in the prediction...

Landrieu & obozinski, SIIMS 2017, Raguet & Landrieu, ICML 2018, 2019

Input Point Cloud



Learned Embedding



Oversegmentation

True Objects

1) Train a neural network to produce points embeddings with high contrast at the border of objects...
2) ... which serve as inputs of a segmentation algorithm.

→ Requires 5 times less superpoints than unsupervized approaches.

Landrieu, CVPR 2019

# Adjacency Graph

- $G = (C, E)$ adjacency graph
- $E_{\text{inter}}$ : set of inter-object edges
- $E_{\text{intra}}$ : set of intra-object edges
- We want high contrast at $E_{\text{inter}}$ and homogeneous value at $E_{\text{intra}}$
- If we get $E_{\text{inter}}$ right, then we get objects' borders.
  almost!

# Generalized Minimal Partition Problem

- $e_i$ embeddings of the local geometry/radiometry
- Superpoints: connected component of $f^\star$ a **piecewise-constant approximation** of $e_i$.

$$f^\star = \operatorname*{arg\,min}_{f \in \mathbb{R}^{C \times m}} \sum_{i \in C} \|f_i - e_i\|^2 + \sum_{(i,j) \in E} w_{i,j} \left[f_i \neq f_j\right],$$

- Superpoints: regions with homogeneous embeddings
- **Problem:** a non-convex, nondifferentiable, noncontinuous problem
- Good approximations can be computed with $\ell_0$-cut pursuit [Landrieu & Obozinski SIIMS 2018, Raguet & Landrieu, ICML2018,2019]

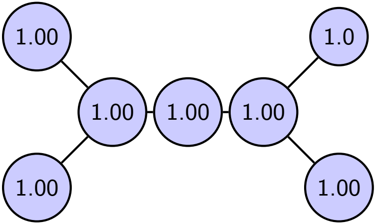
# The Problem With the GMPP

$$f^\star = \underset{f \in \mathbb{R}^{C \times m}}{\arg\min} \sum_{i \in C} \|f_i - e_i\|^2 + \sum_{(i,j) \in E} w_{i,j} \left[ f_i \neq f_j \right],$$

- Let consider our pipeline:
  - Let $x$ be the parameters of the Local Point Embedder
  - Let $e(x)$ be the resulting embeddings
  - Let $f^\star(e(x))$ be the solution of the GMPP
  - Let $CCC$ the constant connected component operator on $G$
  - The superpoints are: $\mathcal{S} = CCC(f^\star(e(x)))$
- Let $M(\mathcal{S})$ be a measure of how good an oversegmentation is (implementing purity, border recall, etc...)
- **Naive Approach:** $\ell(x) = -M(CCC(f^\star(e(x))))$
- To backpropagate we need: $\frac{\partial CCC}{\partial f^\star}$ and $\frac{\partial f^\star}{\partial e}$
- **Problem:** Those functions are **not differentiable** (at all).

# Non-differentiability of the naive pipeline

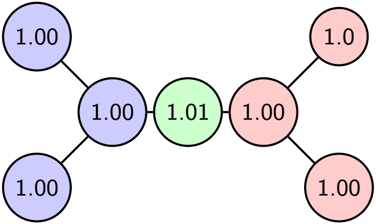
- Non differentiability of the CCC operator



# Non-differentiability of the naive pipeline

- Non differentiability of the CCC operator

= Tiny changes - large consequence



# Non-differentiability of the naive pipeline

$$f^\star = \arg\min \|f_0 - e_0\|^2 + \|x_1 - e_1\|^2 + 0.5[f_0 \neq f_1]$$

- Non differentiability of the CCC operator

$=$ Tiny changes - large consequence

- Non differentiability of $f^\star(e)$

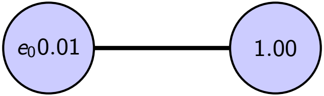
# Non-differentiability of the naive pipeline

$$f^\star = \arg\min \|f_0 - e_0\|^2 + \|x_1 - e_1\|^2 + 0.5[f_0 \neq f_1]$$

- Non differentiability of the CCC operator

= Tiny changes - large consequence

- Non differentiability of $f^\star(e)$




$$f_0^\star = 0.505, \quad f_1^\star = 0.505$$

# Non-differentiability of the naive pipeline

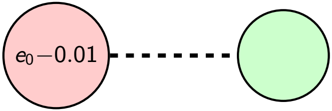
$$f^\star = \arg\min \|f_0 - e_0\|^2 + \|x_1 - e_1\|^2 + 0.5[f_0 \neq f_1]$$

- Non differentiability of the CCC operator

= Tiny changes - large consequence

- Non differentiability of $f^\star(e)$

= non-continuous w.r.t inputs




$$f_1^\star = -0.01, \quad f_1^\star = 1.00$$

## Graph-Structured Contrastive Loss

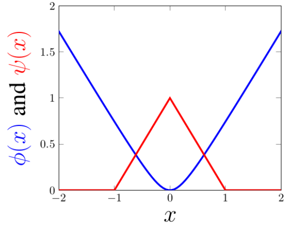
- We propose a *surrogate* loss to learn meaningful embeddings

$$\ell(e) = \frac{1}{|E|}\left(\sum_{(i,j)\in E_{\text{intra}}} \phi\left(e_i - e_j\right) + \sum_{(i,j)\in E_{\text{inter}}} \mu_{i,j}\psi\left(e_i - e_j\right)\right),$$

- $\phi$ minimum at 0, $\psi$ maximum at 0

$$\begin{aligned}\phi(x) &= \delta(\sqrt{\|x\|^2/\delta^2+1}-1)\\ \psi(x) &= \max\left(1-\|x\|, 0\right)\end{aligned}$$

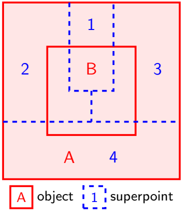
- Promotes homogeneity within objects and contrast at their borders
- $\mu_{i,j}$ : weight of inter-edges

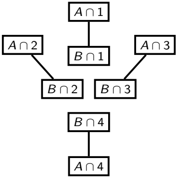


# Cross-Partition Weighting Strategy

- The role of $\mu_{i,j}$ is critical
- Indeed, one single missed inter-edge can have drastic consequence on the purity!
- **Objective:** $\mu_{i,j}$ represents the importance of $(i, j)$
- We propose: the **cross-partition weighting strategy**
- Based on the cross-partition between $\mathcal{O}$ objects and superpoints $\mathcal{S}$:

$$\mathcal{C} = \{O \cap S \mid O \in \mathcal{O}, S \in \mathcal{S}\}$$
$$\mathcal{E} = \{\{(i,j) \in (U \times V) \cap E_{\text{inter}}\} \mid U, V \in \mathcal{C}\}\,.$$



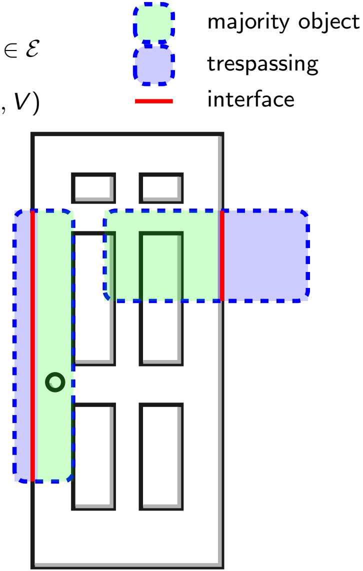




cross-segmentation graph $\mathcal{G}$

# Cross-Partition Weighting Strategy, cont'd

$$\mu_{U,V} = \mu \frac{\min(|U|,|V|)}{|(U,V)|} \qquad \text{for } (U,V) \in \mathcal{E}$$

$$\mu_{i,j} = \mu_{U,V} \qquad \text{for all } (i,j) \in (U,V)$$



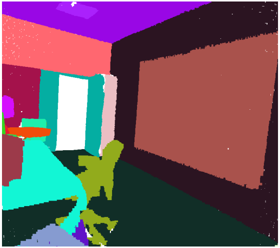

- $\min(|U|,|V|)$ : how many *trespassing* points *i.e.* not in majority object $\sim$ undersegmentation error
- $|(U,V)|$ : length of the interface $\sim$ number of terms in $\ell$

$$\sum_{(i,j)\in(U,V)} \mu_{i,j} = \mu \min(|U|,|V|)$$

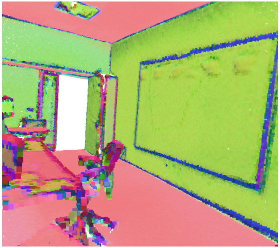
- Sum of weights of edges constituting a superedge $\propto$ number of trespassing points = undersegmentation error.

# Illustration

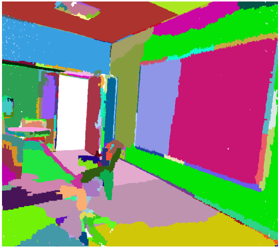
Input cloud



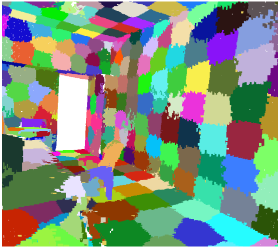
Ground truth objects



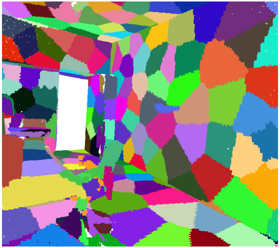
LCE embeddings



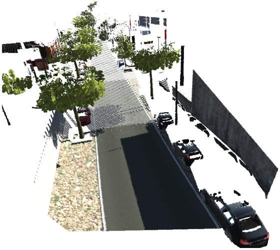
Graph-LCE (ours)



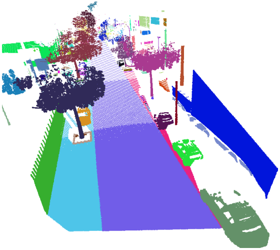
VCCS, Papon *et al.* 2013



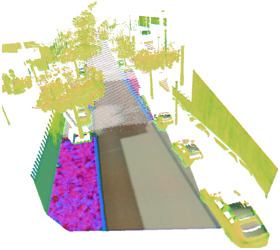
Lin *et al.* 2018



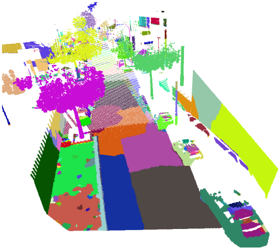
Input cloud



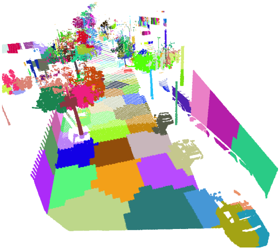
Ground truth objects



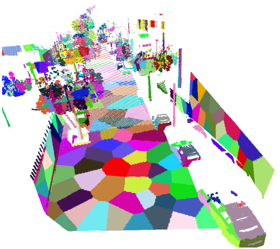
LCE embeddings



Graph-LCE (ours)



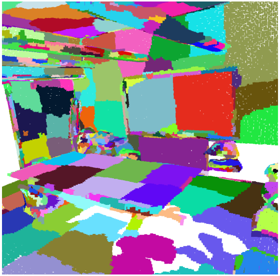
VCCS, Papon *et al.* 2013



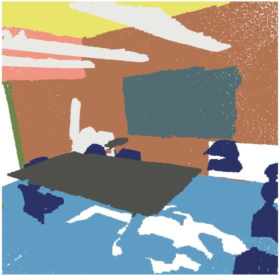
Lin *et al.* 2018

# Presentation Layout

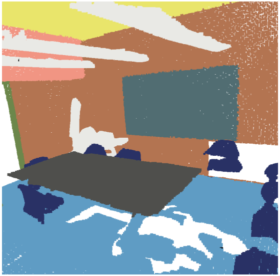
Input Cloud



Oversegmentation



prediction

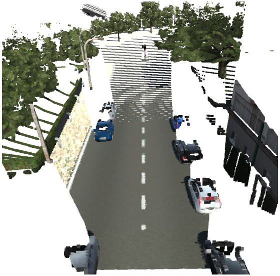


Ground Truth

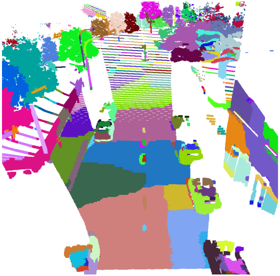
**S3DIS**

- ceiling
- floor
- wall
- column
- beam
- window
- door
- table
- chair
- bookcase
- sofa
- board
- clutter
- unlabelled

# Illustration

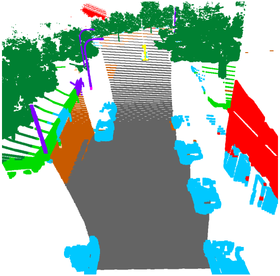


Input Cloud

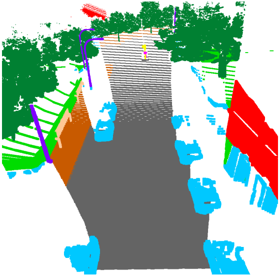


Oversegmentation

**vKITTI**
- terrain
- tree
- vegetation
- building
- road
- guard rail
- traffic sign
- traffic light
- pole
- misc
- truck
- car
- van
- unlabelled



prediction

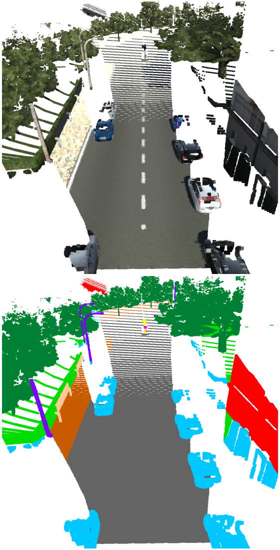


Ground Truth

| **Method** | OA | mAcc | mIoU |
|---|---|---|---|
| 6-fold cross validation | | | |
| PointNet | 78.5 | 66.2 | 47.6 |
| Engelmann *et al.* 2017 | 81.1 | 66.4 | 49.7 |
| PointNet++ | 81.0 | 67.1 | 54.5 |
| Engelmann *et al.* 2018 | 84.0 | 67.8 | 58.3 |
| SPG | 85.5 | 73.0 | 62.1 |
| PointCNN | 88.1 | 75.6 | 65.4 |
| ConvPoint | 88.1 | 75.6 | 68.2 |
| SSP + SPG | 87.9 | 78.3 | 68.4 |
| PointSIFT | 88.7 | - | 70.2 |
| KPConv | **88.8** | **79.1** | **70.6** |
| Fold 5 | | | |
| PointNet | - | 49.0 | 41.1 |
| Engelmann *et al.* | 84.2 | 61.8 | 52.2 |
| pointCNN | 85.9 | 63.9 | 57.3 |
| SPG | 86.4 | 66.5 | 58.0 |
| PCCN | - | 67.0 | 58.3 |
| SSP + SPG | **87.9** | **68.2** | **61.7** |

Table: State-of-the-art 2019: **OA** : Overall Accuracy, **mAcc** : average class accuracy, **mIoU**: average class Intersection over Union.

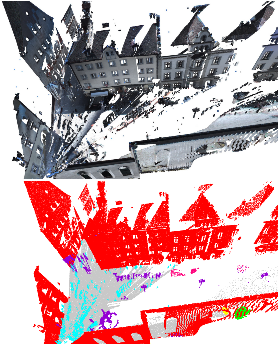
# Quantitative Results on vKITTI3D:



| **Method** | OA | mAcc | mIoU |
|---|---|---|---|
| PointNet | 79.7 | 47.0 | 34.4 |
| Engelmann *et al.* 2018 | 79.7 | 57.6 | 35.6 |
| Engelmann 2017 | 80.6 | 49.7 | 36.2 |
| 3P-RNN | **87.8** | 54.1 | 41.6 |
| SSP + SPG | 84.3 | **67.3** | **52.0** |

Table: STOTA 2019 on vKITTI3D with 6-fold cross validation.

# Quantitative Results on Semantic3D



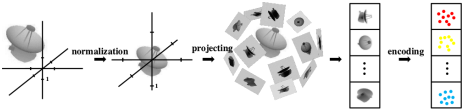
| **Method** | OA | mIoU |
|---|---|---|
| reduced test set: 78 699 329 points | | |
| TMLC-MSR | 86.2 | 54.2 |
| DeePr3SS | 88.9 | 58.5 |
| SnapNet | 88.6 | 59.1 |
| SegCloud | 88.1 | 61.3 |
| SPG | **94.0** | 73.2 |
| KP-FCNN | 92.9 | **74.6** |
| full test set: 2 091 952 018 points | | |
| TMLC-MS | 85.0 | 49.4 |
| SnapNet | 91.0 | 67.4 |
| SPG | 92.9 | 76.2 |
| ConvPoint | **93.4** | **76.5** |

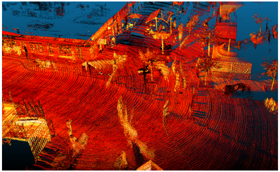
Table: STOTA 2019 on Semantic3D.

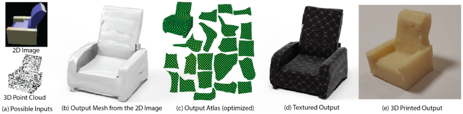
- **Which algorithm to chose in practice:**
- Start with a classical algorithm for baseline / data cleaning
- If enough annotation, PointNet + sliding window.
- If not powerful enough, PointCNN/KPconv/ConvPoint.
- If too slow / loss of global structure: SPG/PointNet++.

- Semi-supervized learning **Zhu2016**.
- Real-time analysis for dynamic 3D data for autonomous driving (ANR READY3D).
- Deep learning for other remote sensing tasks: segmentation, object detection, surface reconstruction. **Groueix2018**.
- Multi-source, multi-task, etc...
- Most codes are now open-source.

  **github/loicland/superpoint-graph**
  420 ★ 136 ⑂





# Presentation Layout

# Bibliography I

**Qi et. al.2017a** Qi, C. R., Su, H., Mo, K., & Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. CVPR, 2017
**Gaidon2016** Gaidon, A., Wang, Q., Cabon, Y., & Vig, E. Virtual worlds as proxy for multi-object tracking analysis. ,CVPR2016.
**Engelmann2017** Engelmann, F., Kontogianni, T., Hermans, A. & Leibe, B. Exploring spatial context for 3d semantic segmentation of point clouds. CVPR, DRMS Workshop, 2017.
**Hackel2017i** Timo Hackel and N. Savinov and L. Ladicky and Jan D. Wegner and K. Schindler and M. Pollefeys, SEMANTIC3D.NET: A new large-scale point cloud classification benchmark,ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,2017
**Armeni2016** Iro Armeni and Ozan Sener and Amir R. Zamir and Helen Jiang and Ioannis Brilakis and Martin Fischer and Silvio Savarese, 3D Semantic Parsing of Large-Scale Indoor Spaces, CVPR, 2016
**Demantke2011** Demantke, J., Mallet, C., David, N. & Vallet, B. Dimensionality based scale selection in 3D lidar point clouds. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2011.

# Bibliography II

**Weinmann2015** Weinmann, M., Jutzi, B., Hinz, S. & Mallet, C., Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS Journal of Photogrammetry and Remote Sensing, 2015.
**Landrieu2017a** Landrieu, L., Raguet, H., Vallet, B., Mallet, C., & Weinmann, M. A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 2017.
**Boulch2017** Boulch, Alexandre, Le Saux, Bertrand, and Audebert, Nicolas, Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks, 3DOR, 2017.
**Wu2015** Wu, Z., Song, S. Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. 3D shapenets: A deep representation for volumetric shapes. CVPR, 2015.
**Riegler2017** Riegler, G., Osman Ulusoy, A., & Geiger, A. Octnet: Learning deep 3d representations at high resolutions, CVPR, 2017.
**Tchapmi2017** Tchapmi, L., Choy, C., Armeni, I., Gwak, J. & Savarese, S., Segcloud: Semantic segmentation of 3d point clouds. 3DV, 2017.
**Jampani2018**, Jampani Su, H., , V. Sun, D., Maji, S., Kalogerakis, E., Yang, M. H., & Kautz, J., Splatnet: Sparse lattice networks for point cloud processing. CVPR2018.
**Tatarchenko2018** Tatarchenko, M., Park, J., Koltun, V., & Zhou, Q. Y. Tangent Convolutions for Dense Prediction in 3D. CVPR, 2018
**Li2018** Li, Y., Bu, R., Sun, M., Wu, W., Di, X., & Chen, B. PointCNN: Convolution On $\chi$-Transformed Points. NIPS, 2018.

# Bibliography III

**Qi2017** Qi, X., Liao, R., Jia, J., Fidler, S., & Urtasun, R. 3D Graph Neural Networks for RGBD Semantic Segmentation. In PCVPR, 2017.
**Simonovsky2017** Simonovsky, M., & Komodakis, N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. CVPR, 2017.
**Landrieu&Simonovski2018** Landrieu, L., & Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. CVPR, 2018
**Landrieu2019** Landrieu, L. & Boussaha, M. Point Cloud Oversegmentation with Graph-Structured Deep Metric Learning, CVPR2019.
**Qi et. al.2017b** Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Advances in Neural Information Processing Systems (pp. 5099-5108).
**Zhu2016** Zhu, Z., Wang, X., Bai, S., Yao, C., & Bai, X.. Deep learning representation using autoencoder for 3D shape retrieval. Neurocomputing, 2016.
**Groueix2018** Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., & Aubry, M. (2018). A papier-mâché approach to learning 3d surface generation. CVPR, 2018.
**Boulch2019** Boulch, A. (2019). Generalizing discrete convolutions for unstructured point clouds. arXiv:1904.02375.
**Guiotte2020** Guiotte, F., Pham, M. T., Dambreville, R., Corpetti, T., & Lefèvre, S. (2020). Semantic segmentation of LiDAR points clouds: Rasterisation beyond Digital Elevation Models.