Computer Vision:
3D detection and segmentation

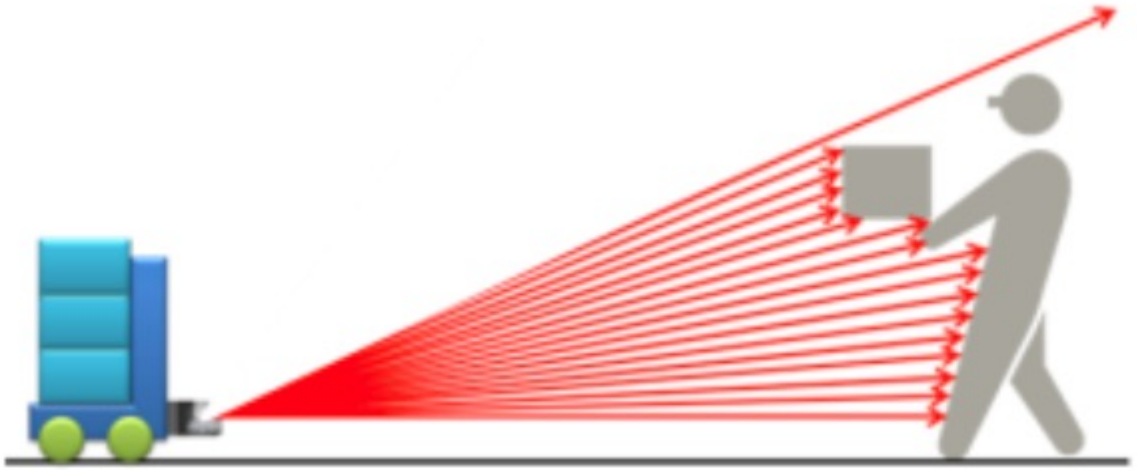
Siheng Chen 陈思衡

3D Sensing

3D Sensing

3D scanner

LiDAR



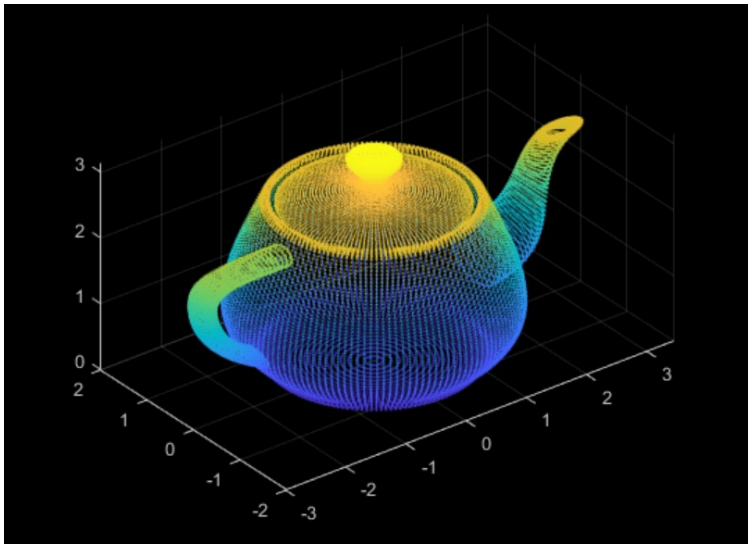
3D segmentation

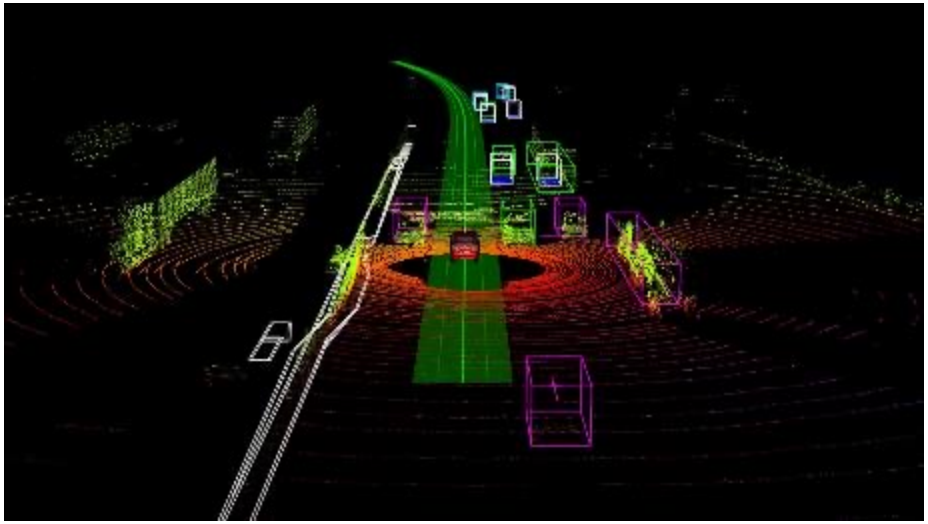
3D detection

Data representation

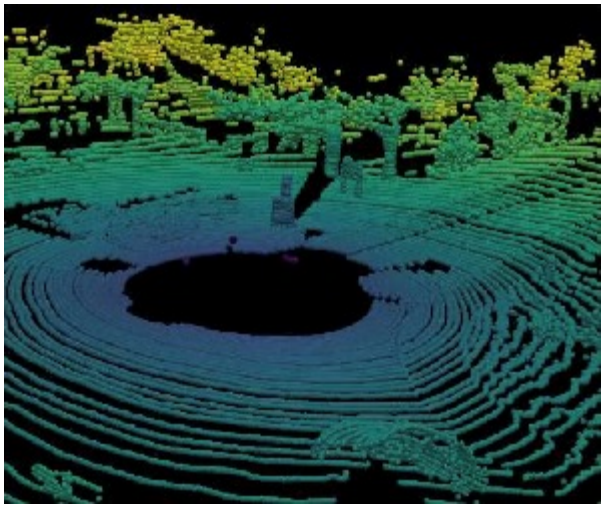
A **point cloud** is a set of data points in space. The points may represent a 3D shape or object. Each point position has its set of Cartesian coordinates (X, Y, Z).

3D Point Cloud are collected through sensors like **LiDAR**.

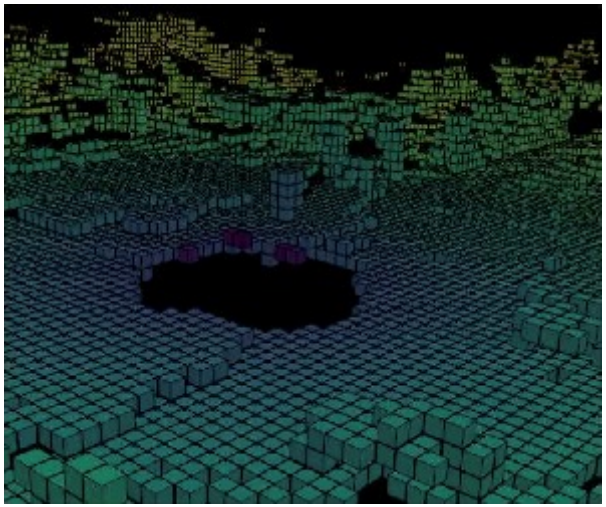




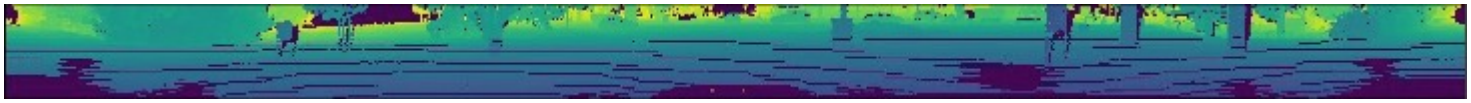
Data representation



Raw points



3D voxel

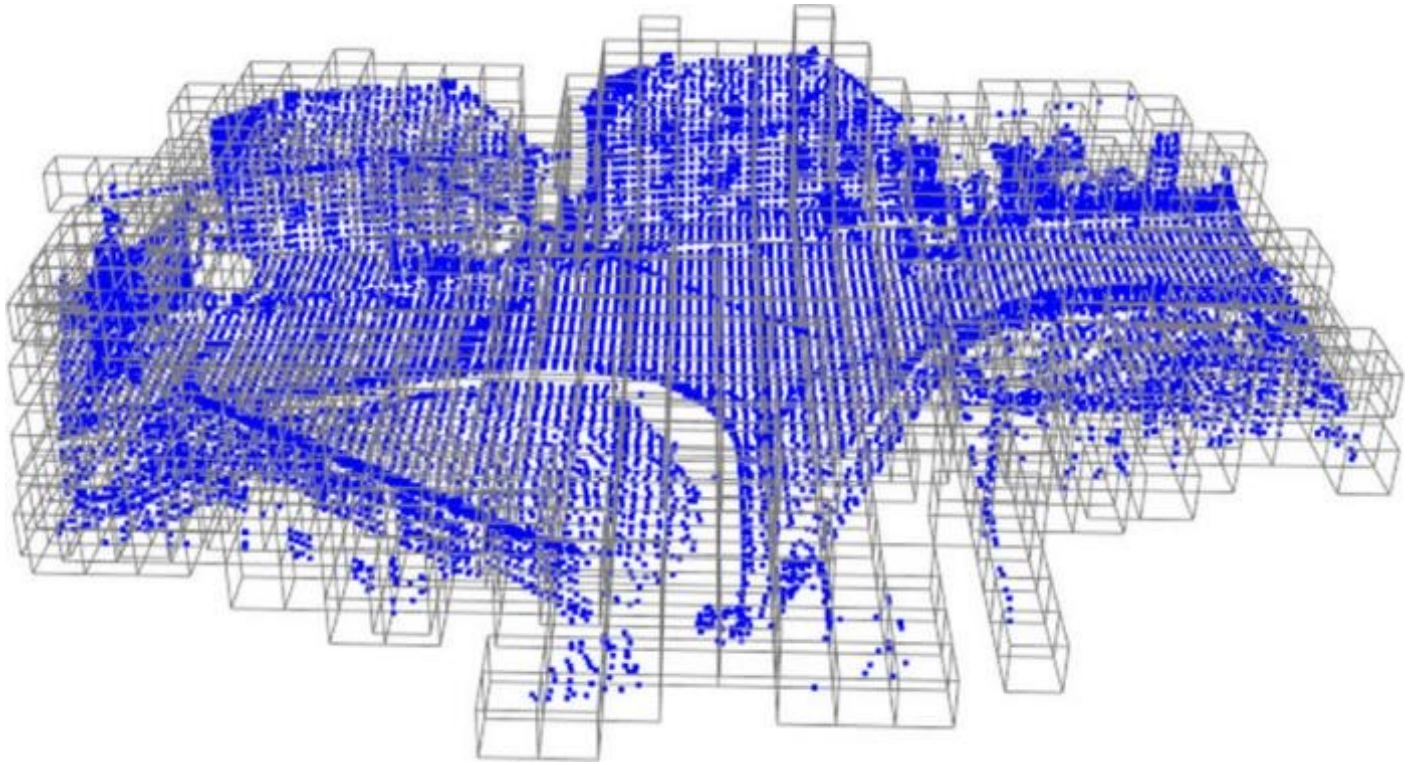


Range view

Voxel-based representation

Point cloud -> voxel

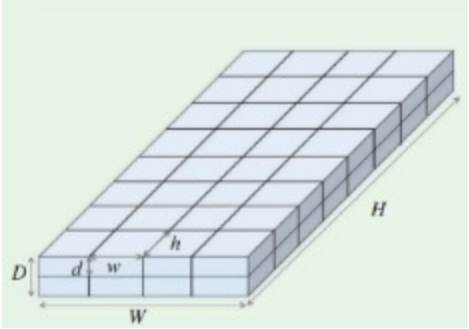
Put the point cloud into the regular grids



Occupancy grid: value = 1 if points in grid else 0

Voxel-based representation

- A **voxel** is a volume element that represents a specific grid value in 3D space.



- Voxel-based approaches divide the point clouds into equally spaced 3D voxels in the **three-dimensional cartesian coordinate system**.
- Then, feature learning can be applied to extract features of a group of points within each voxel.
- This representation scheme results in the reduction of the point cloud's dimension, saving memory resources.

3D Convolution based on Voxel

Limitation

For a scene with 20m x 20m x 5m:

if voxel size = 0.1m

total 200*200*50=2,000,000 voxels

3D Convolution based on Voxel

Limitation

For a scene with 20m x 20m x 5m:

if voxel size = 0.1m

total 200*200*50=2,000,000 voxels

A 256x256 resolution image

total 256*256=65,536 pixels

Compared with 2D convolution, the computation cost of 3D convolution is too high!

3D Convolution based on Voxel

Limitation

For a scene with 20m x 20m x 5m:

if reduce total voxels to $\sim 10^4$

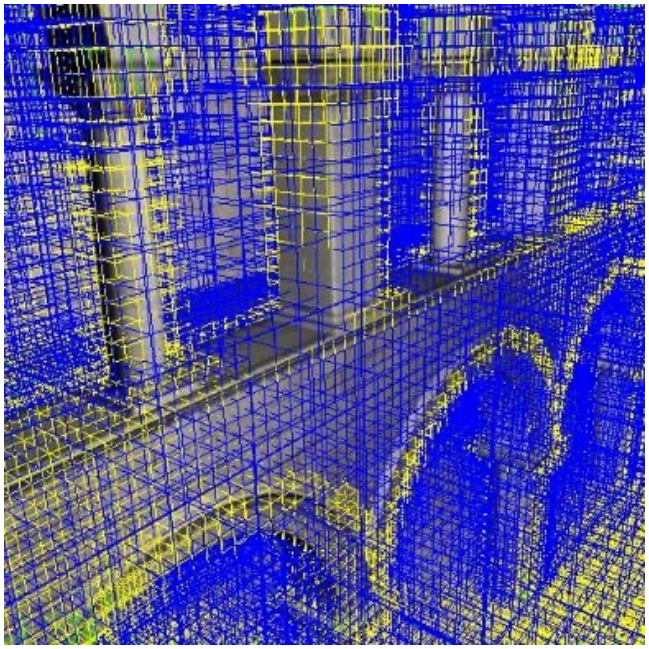
voxel size ≈ 0.5m

too coarse!

3D Convolution based on Voxel

Observation:

Sparsity: more than 90% of the occupancy grid is empty, and make little contribution to the point cloud segmentation



Problem: The voxels in blue is empty voxels and the voxel in yellow is non-empty.

Solution: Convolution on nonzero voxels!

3D Convolution based on Voxel

Convolution on nonzero voxels

Sparse convolution

- Only compute kernel response on nonzero voxels
- Drawback: “submanifold” dilation (reduce sparsity in output voxels)

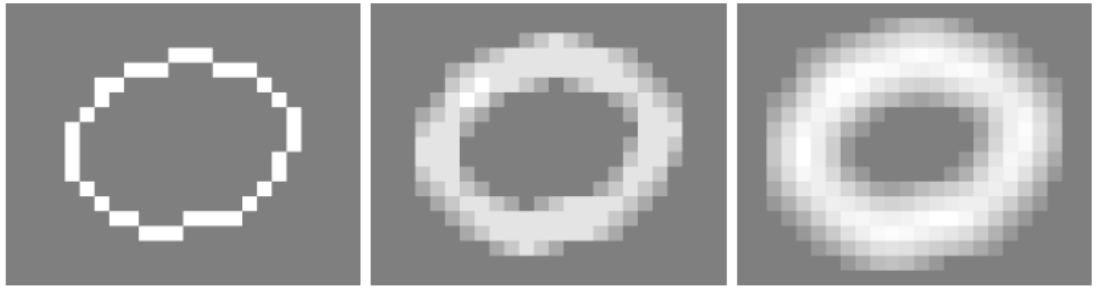


Figure 2: Example of “submanifold” dilation. **Left:** Original curve. **Middle:** Result of applying a regular 3×3 convolution with weights $1/9$. **Right:** Result of applying the same convolution again. Regular convolutions substantially reduce the feature sparsity with each convolutional layer.

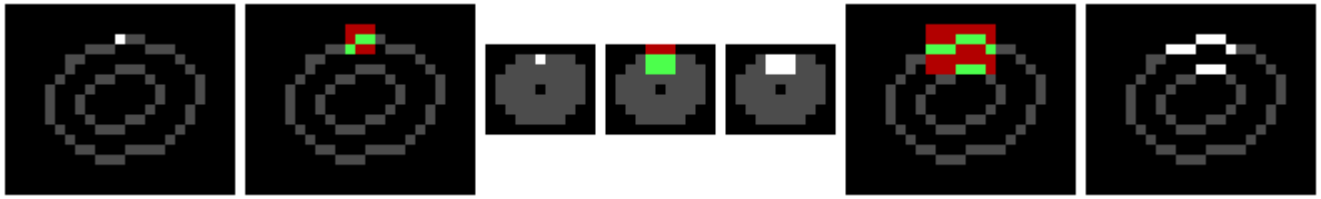
Graham B, Engelcke M, Van Der Maaten L. 3d semantic segmentation with submanifold sparse convolutional networks CVPR 2018

3D Convolution based on Voxel

Convolution on nonzero voxels

Improvement: submanifold sparse convolution

- Only compute kernel response on nonzero voxels
- Only the nonzero voxels in input will be computed in output

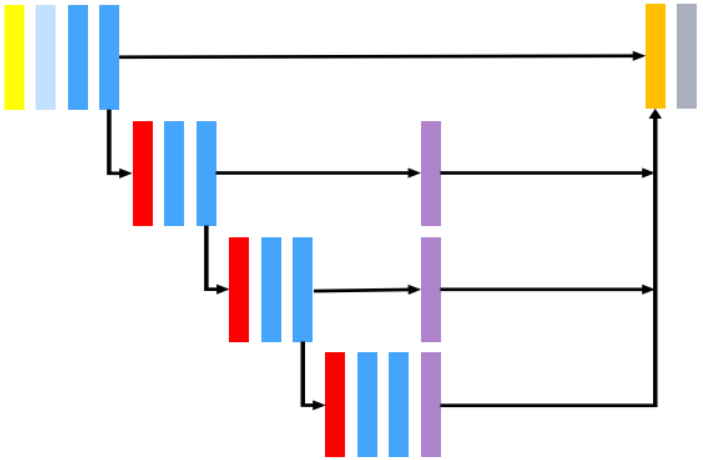


Graham B, Engelcke M, Van Der Maaten L. 3d semantic segmentation with submanifold sparse convolutional networks CVPR 2018

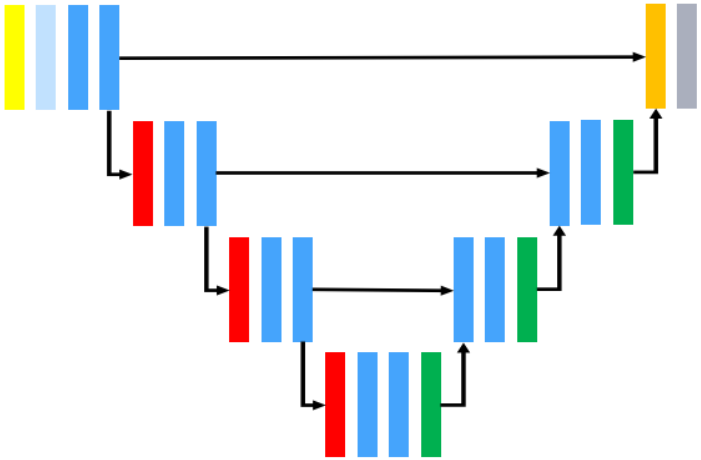
Voxel-based segmentation

Build semantic segmentation network with submanifold sparse convolution

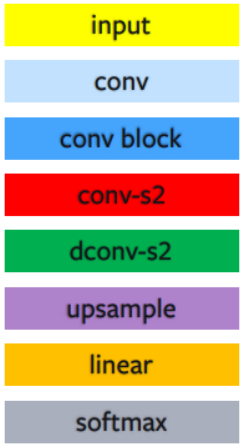
- Basic block: submanifold convolution
- Architecture: FCN, U-Net



(a) Submanifold sparse FCN.



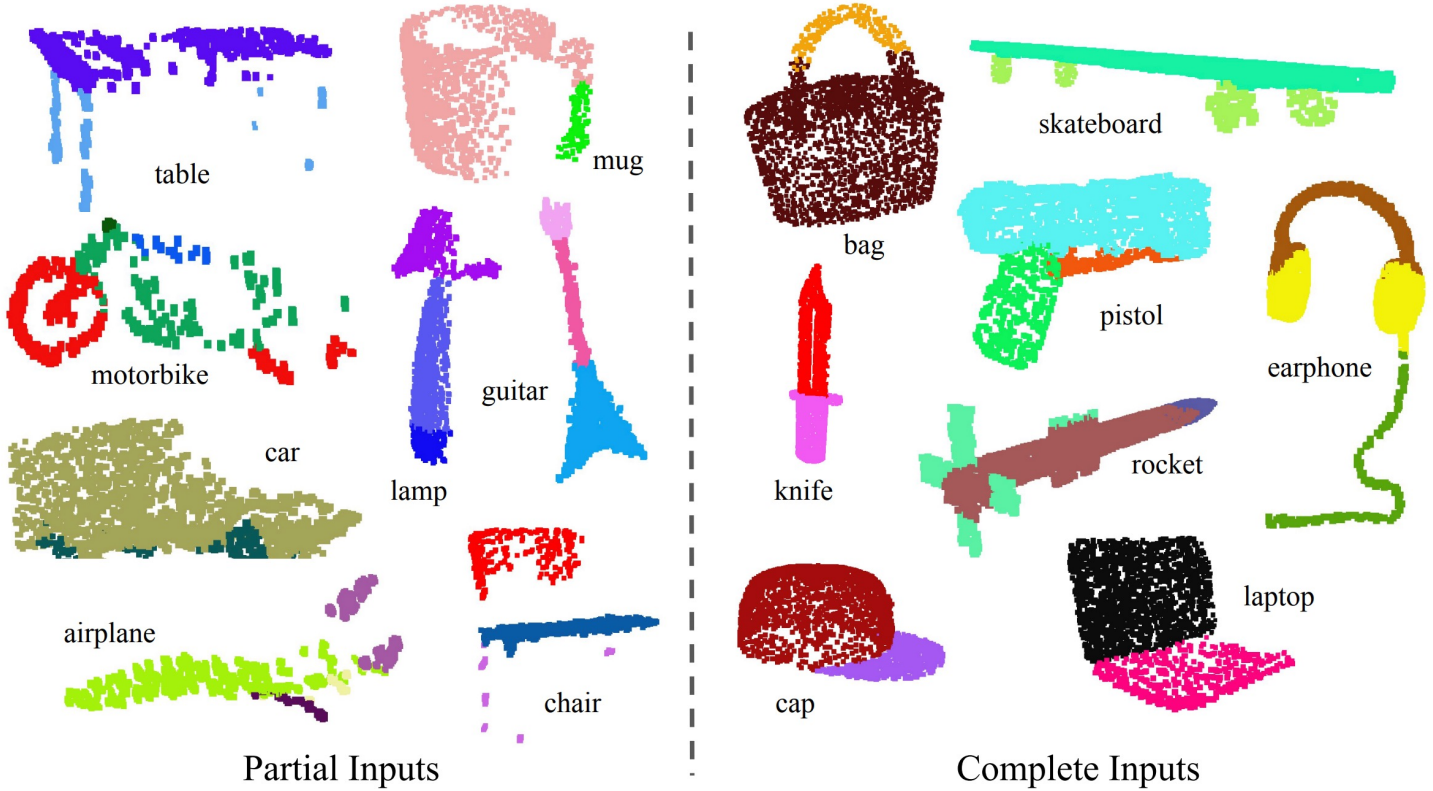
(b) Submanifold sparse U-Net.



Graham B, Engelcke M, Van Der Maaten L. 3d semantic segmentation with submanifold sparse convolutional networks CVPR 2018

Voxel-based segmentation

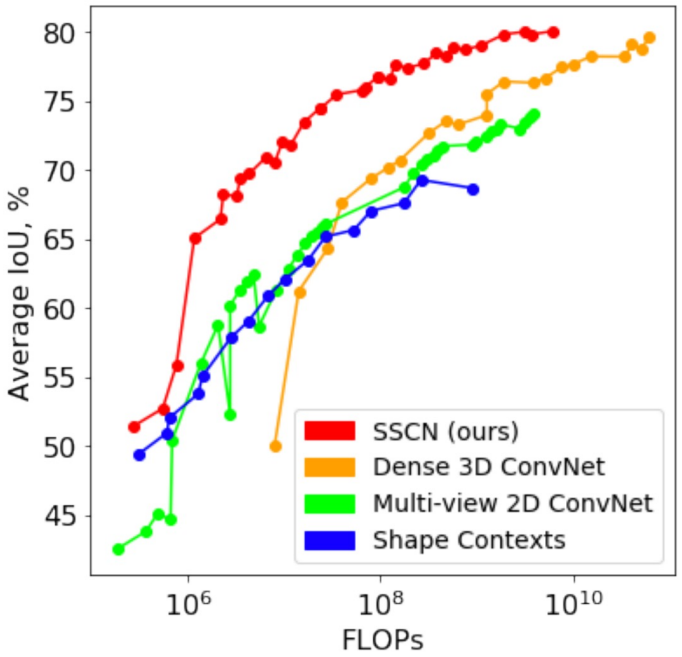
ShapeNet dataset



Graham B, Engelcke M, Van Der Maaten L. 3d semantic segmentation with submanifold sparse convolutional networks CVPR 2018

Voxel-based segmentation

Experimental results



Method	Average IoU
NN matching with Chamfer distance	77.57%
Synchronized Spectral CNN [11]	84.74%
Pd-Network (extension of Kd-Network [10])	85.49%
Densely Connected PointNet (extension of [17])	84.32%
PointCNN	82.29%
Submanifold SparseConvNet (Section 6.5)	**85.98 %**

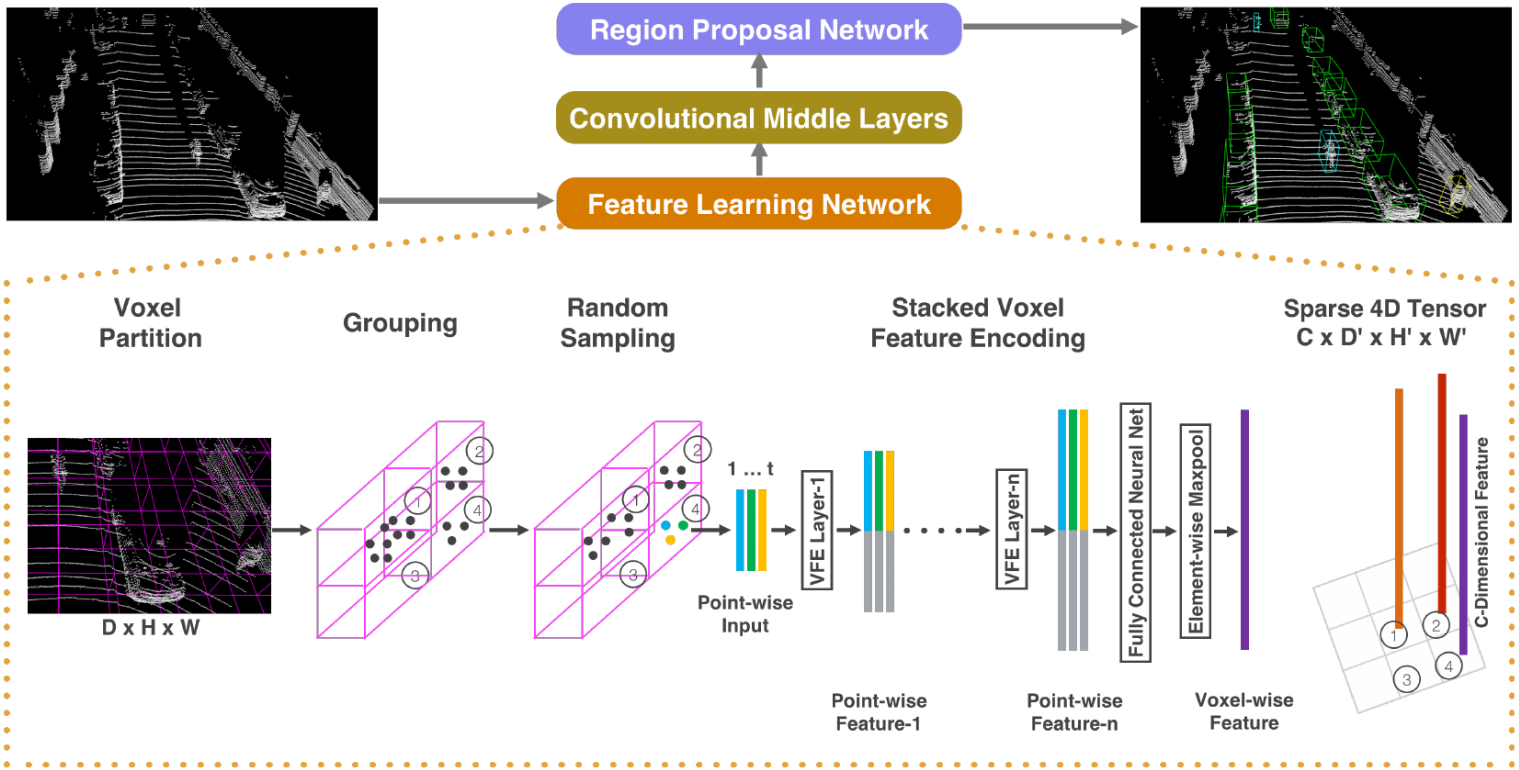
Table 1: Average intersection-over-union (IoU) of six approaches on the test set of a recent part-based segmentation competition on ShapeNet [23]. Higher is better. Our SSCNs outperform all alternative approaches.

Graham B, Engelcke M, Van Der Maaten L. 3d semantic segmentation with submanifold sparse convolutional networks CVPR 2018

Voxel-based detection: VoxelNet

Network architecture:

Encode points into voxel representation and use 3D convolutions

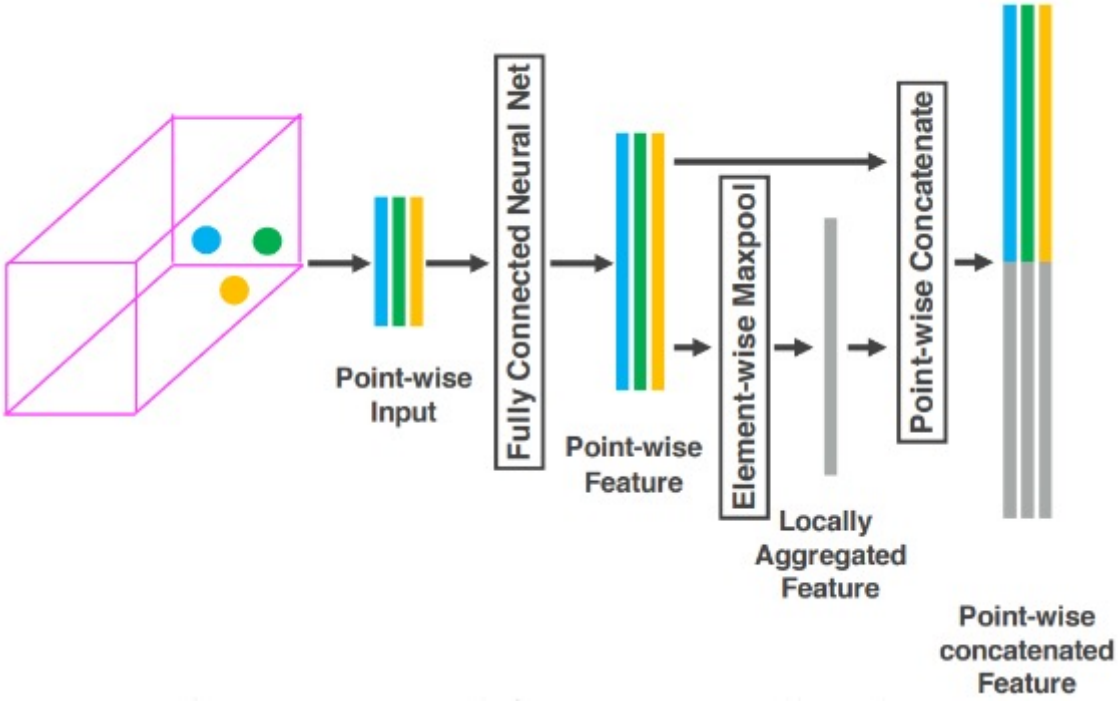


Zhou, Tuzel: VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. CVPR, 2018

Voxel-based detection: VoxelNet

Voxel **F**eature **E**ncoding(VFE) Layer is the key of VoxelNet.

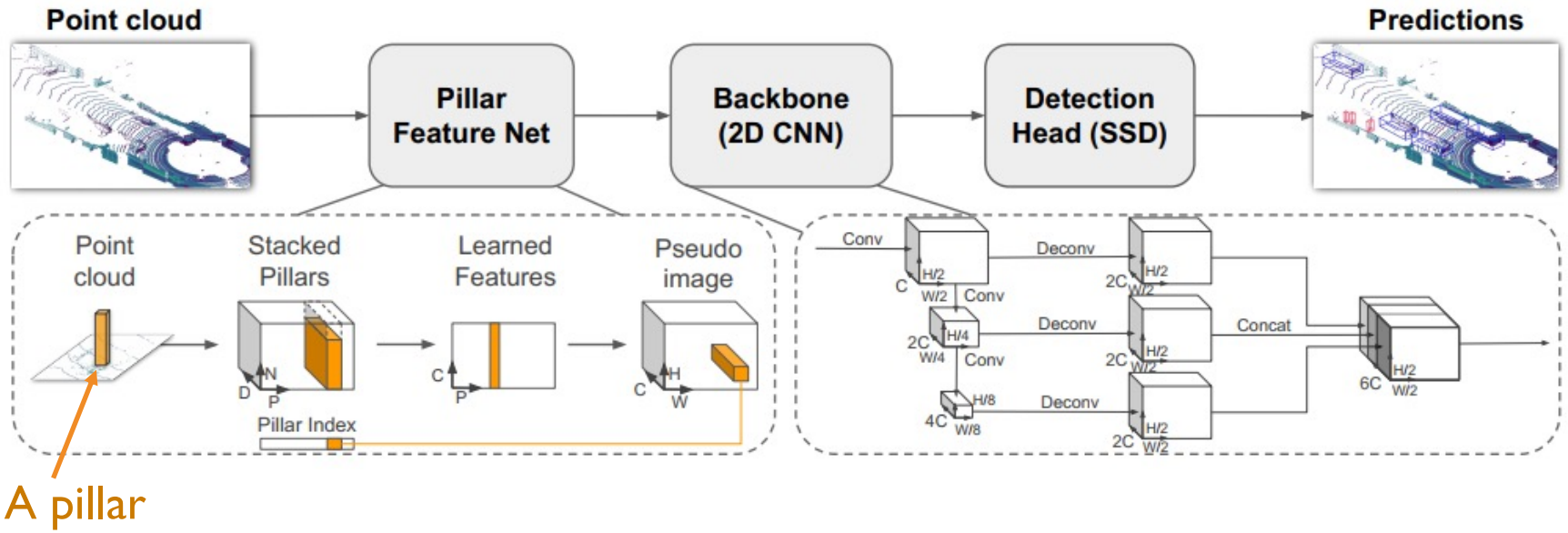
- Extract **point-wise features** in each voxel with FCN net
- Aggregate point-wise features in the same voxel with maxpooling and concatenate it with point-wise features to get **voxel-level feature**



Zhou, Tuzel: VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. CVPR, 2018

Voxel-based detection: PointPillars

Utilizes PointNets to learn a representation of point clouds organized in vertical columns (**pillars**)

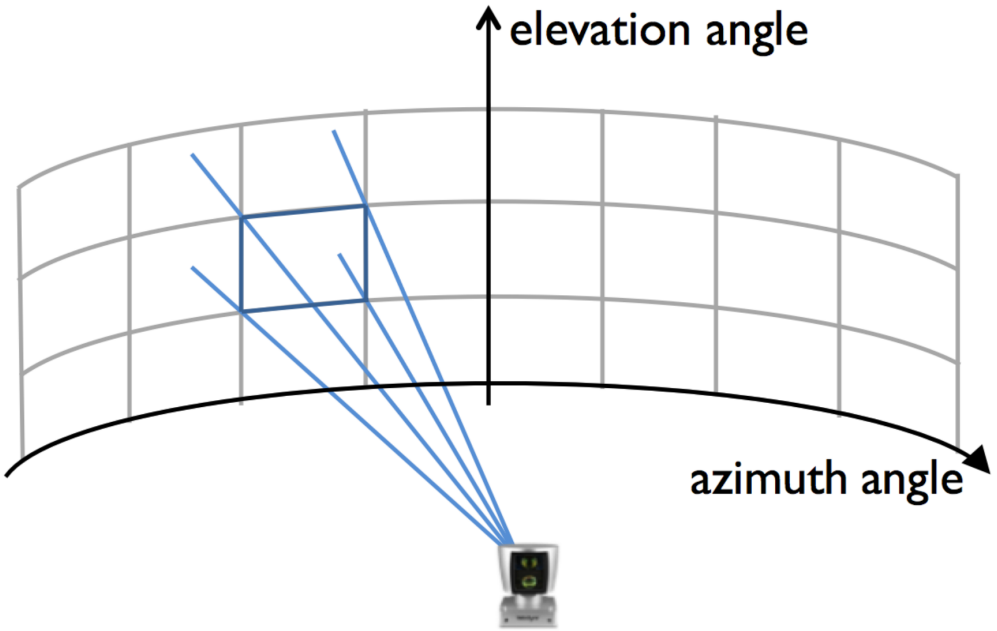


- A feature encoder network that converts a **point cloud** to a **sparse pseudo image**.
- A 2D convolutional backbone to process the pseudo-image into **high-level representation**.
- A detection head that detects and regresses **3D boxes**.

Lang A H, Vora S, Caesar H, et al. Pointpillars: Fast encoders for object detection from point clouds. CVPR 2019

Front-view-based representation

- Real-time LiDAR sweep is essentially a series of range measurements from a single location with certain angular field of view
- We can approximately organize the 3D points in a real-time LiDAR to a 2D range-view image.
- Each pixel in the range-view image corresponds to a frustum in the 3D space. The pixel value is the range from the LiDAR to the closest 3D point inside the frustum.



Front-view-based segmentation

RangeNet++: Fast and Accurate LiDAR Semantic Segmentation

Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss

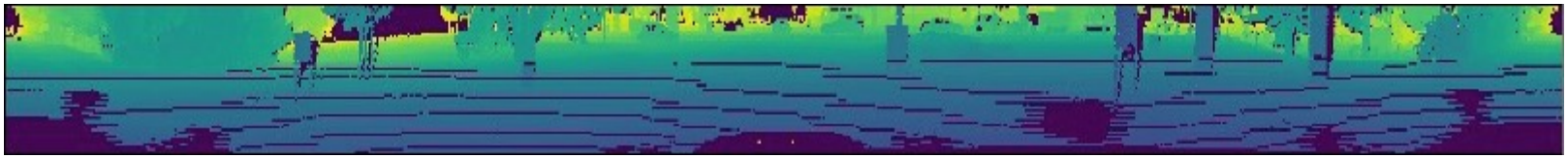
UNIVERSITÄT BONN

Andres Milioto; Ignacio Vizzo; Jens Behley; Cyrill Stachniss, RangeNet++: Fast and Accurate LiDAR Semantic Segmentation, IROS 2019

Front-view-based segmentation: RangeNet++

Intuition:

- Deep learning point cloud semantic segmentation model based on range image (**projected image**)
- Core idea: project point cloud to image, do segmentation at image, and then re-project result back to point cloud.

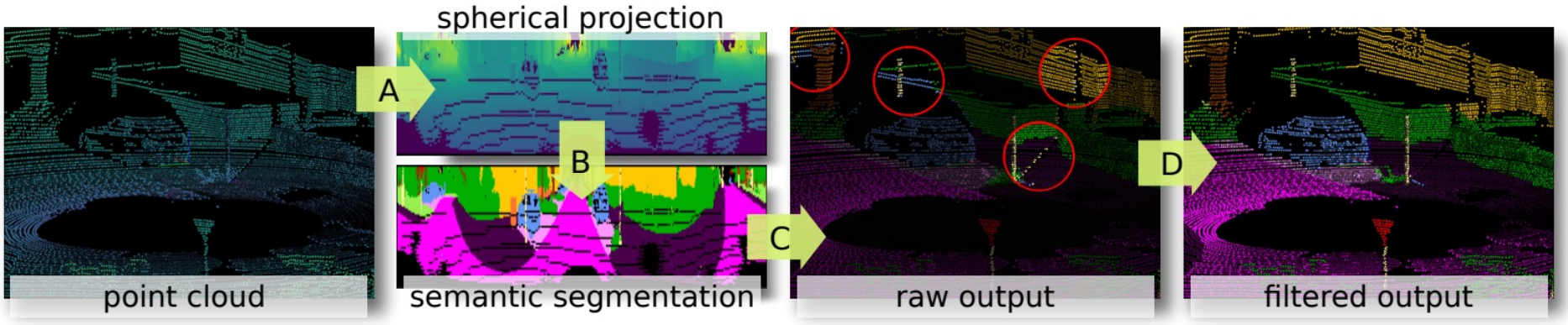


Range-view image

Andres Milioto; Ignacio Vizzo; Jens Behley; Cyrill Stachniss, RangeNet++: Fast and Accurate LiDAR Semantic Segmentation, IROS 2019

Front-view-based segmentation: RangeNet++

Pipeline



A. Project point cloud into range image
B. Image-level semantic segmentation
C. Re-project pixel labels to points
D. Post process

Andres Milioto; Ignacio Vizzo; Jens Behley; Cyrill Stachniss, RangeNet++: Fast and Accurate LiDAR Semantic Segmentation, IROS 2019

Front-view-based segmentation: RangeNet++

Spherical projection

$$\text{point } \mathbf{p}_i = (x, y, z) \rightarrow \text{image coordinates } (u, v)$$

$$\Pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$$

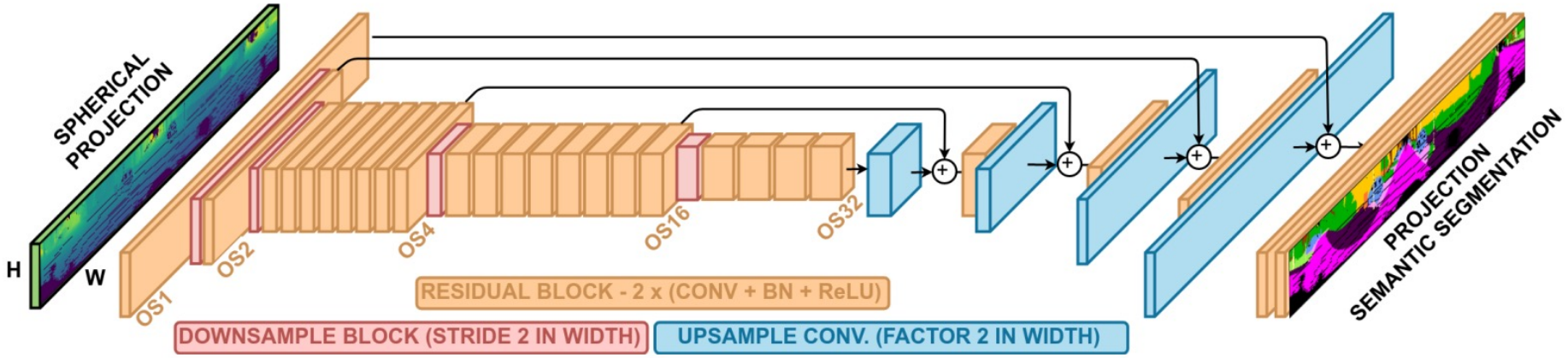
$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\left[1 - \arctan(y, x)\,\pi^{-1}\right] & w \\ \left[1 - \left(\arcsin(z\,r^{-1}) + \mathrm{f}_{\mathrm{up}}\right)\mathrm{f}^{-1}\right] & h \end{pmatrix},$$

Andres Milioto; Ignacio Vizzo; Jens Behley; Cyrill Stachniss, RangeNet++: Fast and Accurate LiDAR Semantic Segmentation, IROS 2019

Front-view-based segmentation: RangeNet++

Fully convolutional semantic segmentation

hour-glass-shaped architecture (multi-scale, skip-connection)



only perform downsampling in **horizontal** direction

Andres Milioto; Ignacio Vizzo; Jens Behley; Cyrill Stachniss, RangeNet++: Fast and Accurate LiDAR Semantic Segmentation, IROS 2019

Front-view-based segmentation: RangeNet++

Point cloud reconstruction from range image

use (u, v) pairs to index the range image, obtain the segmentation label for each point

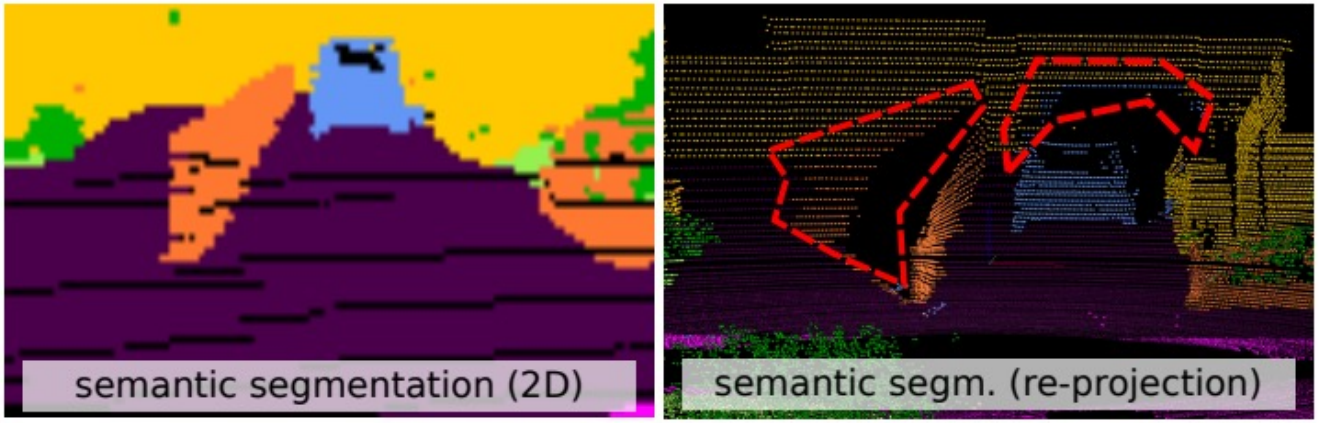


Fig. 4: Illustration of the label re-projection problem. Both the fence and the car in the range image (left) were given the proper semantic label, but during the process of sending the semantics back to the original points (right), the labels were also projected as "shadows".

Andres Milioto; Ignacio Vizzo; Jens Behley; Cyrill Stachniss, RangeNet++: Fast and Accurate LiDAR Semantic Segmentation, IROS 2019

Front-view-based segmentation: RangeNet++

Point cloud post-processing

kNN

For each points p_i

1) select S*S points from local window in range image

2) select k nearest points (distance to p_i < max_dist) from the selected points in 1)

3) vote with the k points in 2) to obtain the label of p_i

Andres Milioto; Ignacio Vizzo; Jens Behley; Cyrill Stachniss, RangeNet++: Fast and Accurate LiDAR Semantic Segmentation, IROS 2019

Front-view-based segmentation: RangeNet++

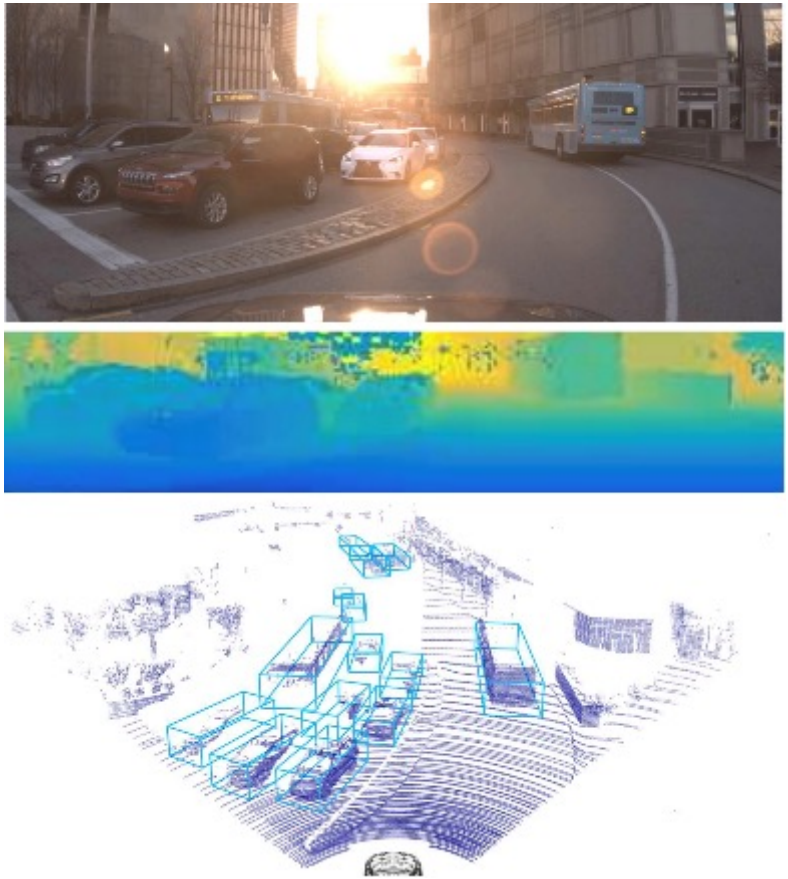
Experimental results

TABLE I: IoU [%] on test set (sequences 11 to 21). RangeNet21 and RangeNet53 represent the new baselines with augmented Darknet backbones (21 and 53 respectively), and the versions with (++) are treated with our fast point cloud post-processing based on range.

Approach	Size	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mean IoU	Scans/sec
Pointnet [14]	50000pts	46.3	1.3	0.3	0.1	0.8	0.2	0.2	0.0	61.6	15.8	35.7	1.4	41.4	12.9	31.0	4.6	17.6	2.4	3.7	14.6	2
Pointnet++ [15]		53.7	1.9	0.2	0.9	0.2	0.9	1.0	0.0	72.0	18.7	41.8	5.6	62.3	16.9	46.5	13.8	30.0	6.0	8.9	20.1	0.1
SPGraph [10]		68.3	0.9	4.5	0.9	0.8	1.0	6.0	0.0	49.5	1.7	24.2	0.3	68.2	22.5	59.2	27.2	17.0	18.3	10.5	20.0	0.2
SPLATNet [19]		66.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	70.4	0.8	41.5	0.0	68.7	27.8	72.3	35.9	35.8	13.8	0.0	22.8	1
TangentConv [20]		86.8	1.3	12.7	11.6	10.2	17.1	20.2	0.5	82.9	15.2	61.7	9.0	82.8	44.2	75.5	42.5	55.5	30.2	22.2	35.9	0.3
SqueezeSeg [21]	64 × 2048 px	68.8	16.0	4.1	3.3	3.6	12.9	13.1	0.9	85.4	26.9	54.3	4.5	57.4	29.0	60.0	24.3	53.7	17.5	24.5	29.5	**66**
SqueezeSeg-CRF [21]		68.3	18.1	5.1	4.1	4.8	16.5	17.3	1.2	84.9	28.4	54.7	04.6	61.5	29.2	59.6	25.5	54.7	11.2	36.3	30.8	55
SqueezeSegV2 [22]		81.8	18.5	17.9	13.4	14.0	20.1	25.1	3.9	88.6	45.8	67.6	17.7	73.7	41.1	71.8	35.8	60.2	20.2	36.3	39.7	50
SqueezeSegV2-CRF [22]		82.7	21.0	22.6	14.5	15.9	20.2	24.3	2.9	88.5	42.4	65.5	18.7	73.8	41.0	68.5	36.9	58.9	12.9	41.0	39.6	40
RangeNet21 [Ours]		85.4	26.2	26.5	18.6	15.6	31.8	33.6	4.0	91.4	57.0	74.0	26.4	81.9	52.3	77.6	48.4	63.6	36.0	50.0	47.4	20
RangeNet53 [Ours]	64 × 2048 px	86.4	24.5	32.7	25.5	22.6	36.2	33.6	4.7	**91.8**	64.8	74.6	**27.9**	84.1	55.0	78.3	50.1	64.0	38.9	52.2	49.9	13
	64 × 1024 px	84.6	20.0	25.3	24.8	17.3	27.5	27.7	7.1	90.4	51.8	72.1	22.8	80.4	50.0	75.1	46.0	62.7	33.4	43.4	45.4	25
	64 × 512 px	81.0	9.9	11.7	19.3	7.9	16.8	25.8	2.5	90.1	49.9	69.4	2.0	76.0	45.5	74.2	38.8	62.7	25.5	38.1	39.3	52
RangeNet53++ [Ours+kNN]	64 × 2048 px	**91.4**	**25.7**	**34.4**	**25.7**	**23.0**	**38.3**	**38.8**	**4.8**	**91.8**	**65.0**	**75.2**	27.8	**87.4**	**58.6**	**80.5**	**55.1**	**64.6**	**47.9**	**55.9**	**52.2**	12
	64 × 1024 px	90.3	20.6	27.1	25.2	17.6	29.6	34.2	7.1	90.4	52.3	72.7	22.8	83.9	53.3	77.7	52.5	63.7	43.8	47.2	48.0	21
	64 × 512 px	87.4	9.9	12.4	19.6	7.9	18.1	29.5	2.5	90.0	50.7	70.0	2.0	80.2	48.9	77.1	45.7	64.1	37.1	42.0	41.9	38

Andres Milioto; Ignacio Vizzo; Jens Behley; Cyrill Stachniss, RangeNet++: Fast and Accurate LiDAR Semantic Segmentation, IROS 2019

Front-view-based detection



RGB image

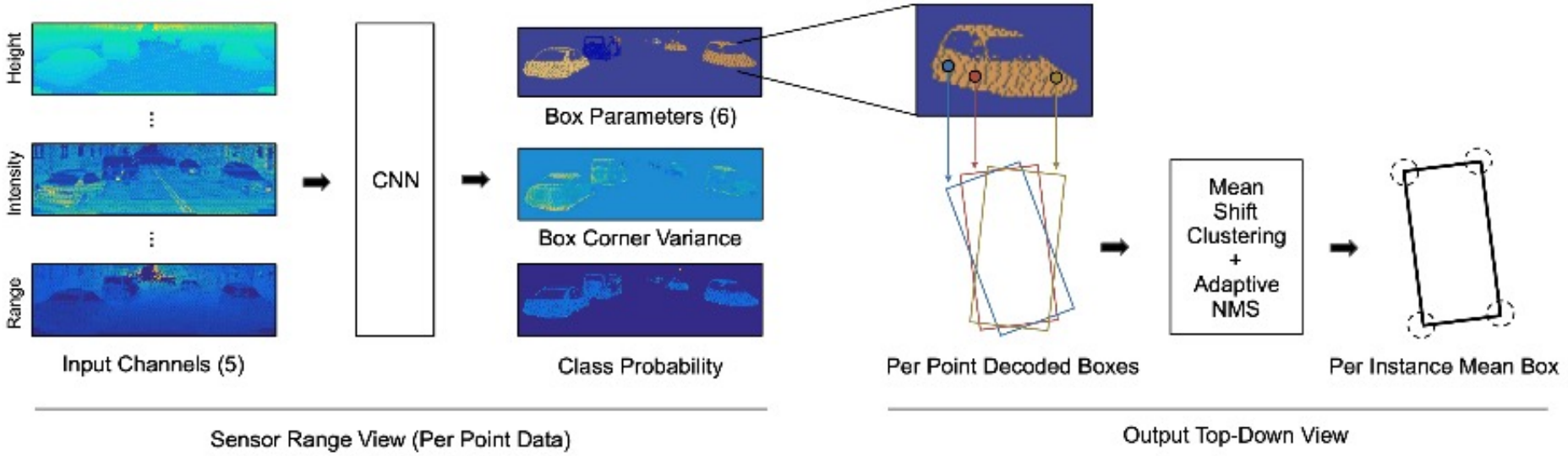
Front-view LiDAR image

3D point cloud

Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, Carl K. Wellington, LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving, CVPR 2019

Front-view-based detection

Network architecture



Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, Carl K. Wellington, LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving, CVPR 2019

Front-view-based detection

Experimental results

- The KITTI object detection benchmark contains 7,481 training sweeps and 7,518 testing sweeps

Table 3: Runtime Performance on KITTI

Method	Forward Pass (ms)	Total (ms)
LaserNet (Ours)	**12**	**30**
PIXOR [28]	35	62
PIXOR++ [27]	35	62
VoxelNet [30]	190	225
MV3D [30]	-	360
AVOD [15]	80	100
F-PointNet [22]	-	170
ContFuse [17]	60	-

Very fast

Table 4: BEV Object Detection Performance on KITTI

Method	Input	Vehicle $AP_{0.7}$		
		Easy	Moderate	Hard
LaserNet (Ours)	LiDAR	78.25	73.77	66.47
PIXOR [28]	LiDAR	81.70	77.05	72.95
PIXOR++ [27]	LiDAR	**89.38**	83.70	**77.97**
VoxelNet [30]	LiDAR	89.35	79.26	77.39
MV3D [5]	LiDAR+RGB	86.02	76.90	68.49
AVOD [15]	LiDAR+RGB	88.53	83.79	77.90
F-PointNet [22]	LiDAR+RGB	88.70	84.00	75.33
ContFuse [17]	LiDAR+RGB	88.81	**85.83**	77.33

Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, Carl K. Wellington, LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving, CVPR 2019

Front-view-based detection

Experimental results

- The entire dataset contains 1.2 million sweeps for training, and 5,969 sweeps for validation

Table 1: BEV Object Detection Performance on ATG4D

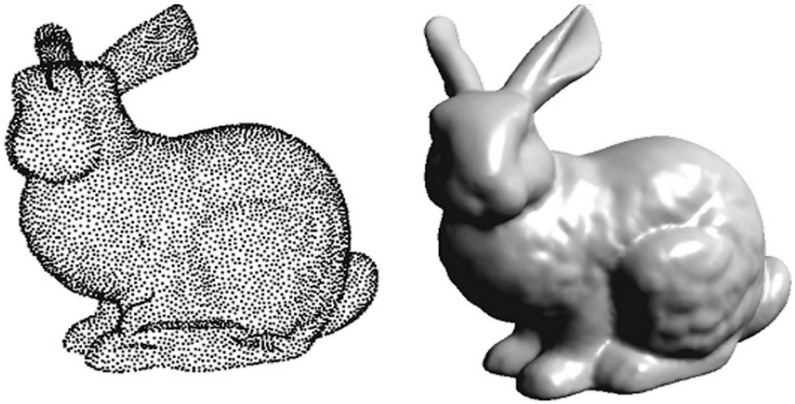
Method	Input	Vehicle $AP_{0.7}$				Bike $AP_{0.5}$				Pedestrian $AP_{0.5}$			
		0-70m	0-30m	30-50m	50-70m	0-70m	0-30m	30-50m	50-70m	0-70m	0-30m	30-50m	50-70m
LaserNet (Ours)	LiDAR	**85.34**	**95.02**	**84.42**	67.65	**61.93**	**74.62**	51.37	40.95	**80.37**	**88.02**	**77.85**	**65.75**
PIXOR [28]	LiDAR	80.99	93.34	80.20	60.19	-	-	-	-	-	-	-	-
PIXOR++ [27]	LiDAR	82.63	93.80	82.34	63.42	-	-	-	-	-	-	-	-
ContFuse [17]	LiDAR	83.13	93.08	82.48	65.53	57.27	68.08	48.83	38.26	73.51	80.60	71.68	59.12
ContFuse [17]	LiDAR+RGB	85.17	93.86	84.41	**69.83**	61.13	72.01	**52.60**	**43.03**	76.84	82.97	75.54	64.19

Perform well when the training dataset is huge

Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, Carl K. Wellington, LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving, CVPR 2019

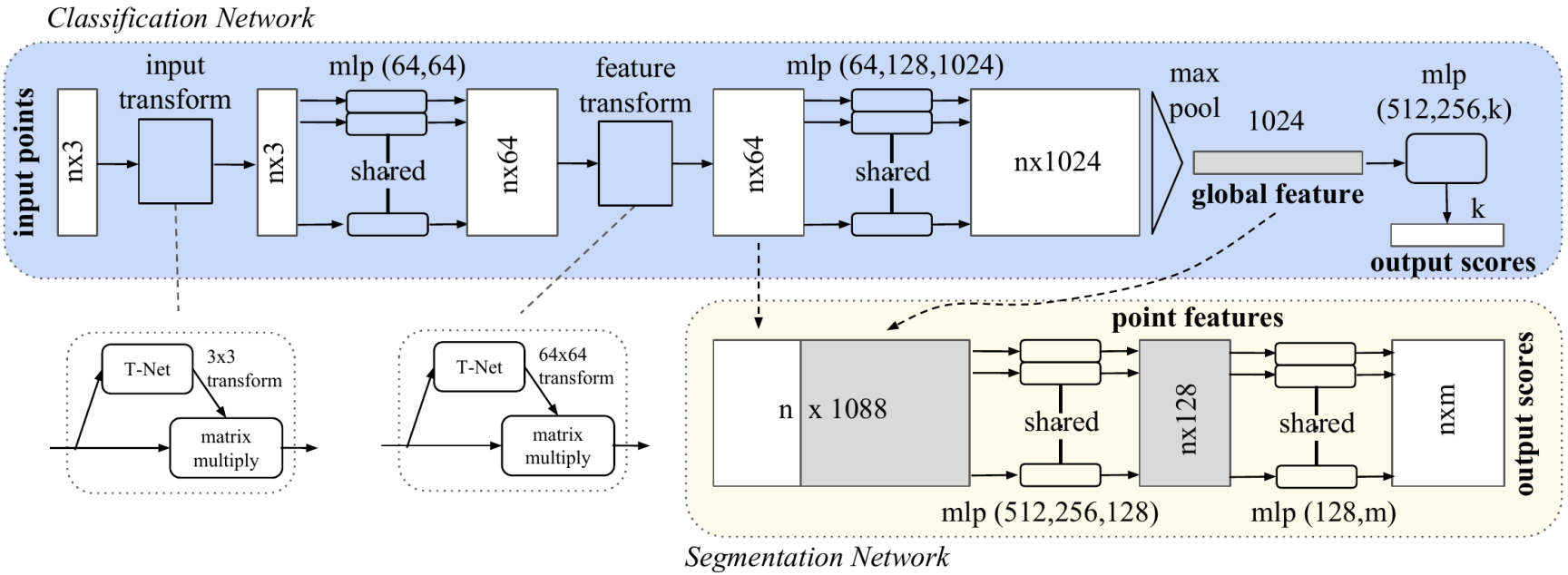
Point-based representation

Point-based methods **directly process the input point clouds** and produce a sparse representation.



- **Unordered**: point cloud is a set of points without specific order.
- **Interaction among points**: points are not isolated, and neighboring points form a meaningful subset.
- **Invariance under transformations：** the learned representation of the point set should be invariant to certain transformations

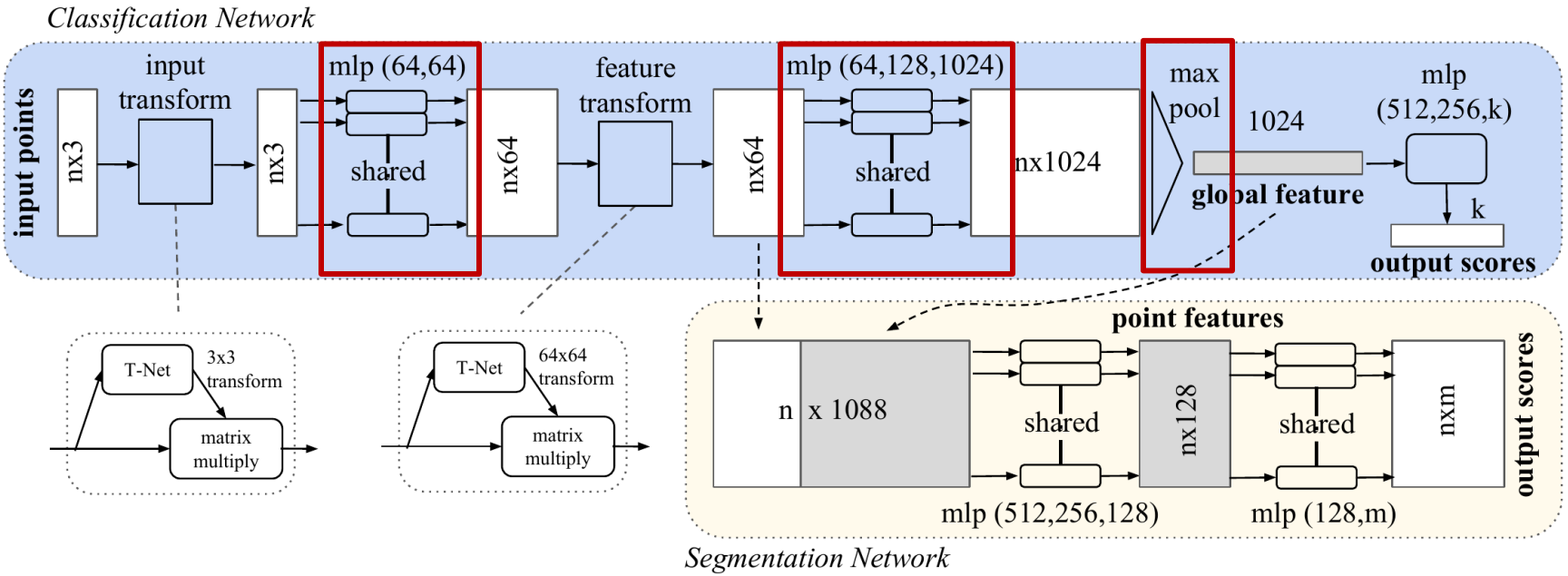
PointNet



1. takes n points as input (X,Y,Z) ;
2. applies input and feature transformations ;
3. aggregates point features by max pooling ;
4. concatenates global and local features ;
5. Output scores

Qi, Su, Mo and Guibas: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. CVPR, 2017.

PointNet

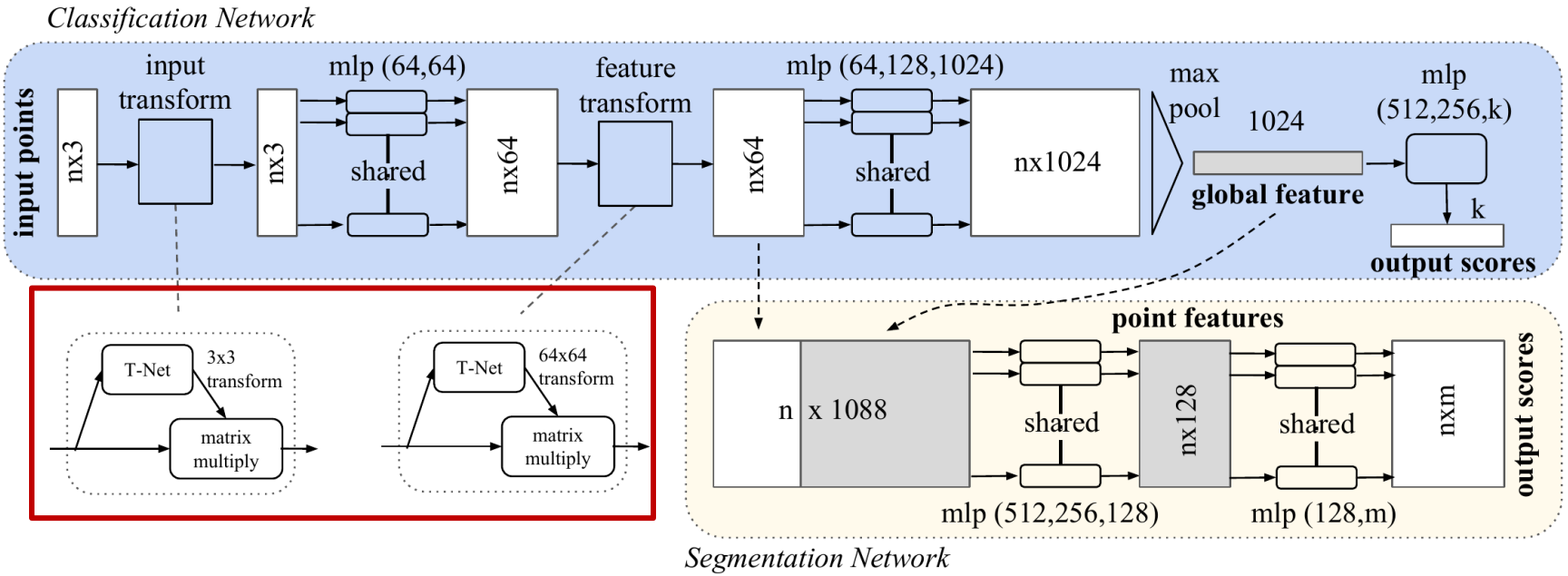


Symmetry Function for Unordered Input

- multi-layer perceptron network
- max pooling function

Qi, Su, Mo and Guibas: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. CVPR, 2017.

PointNet



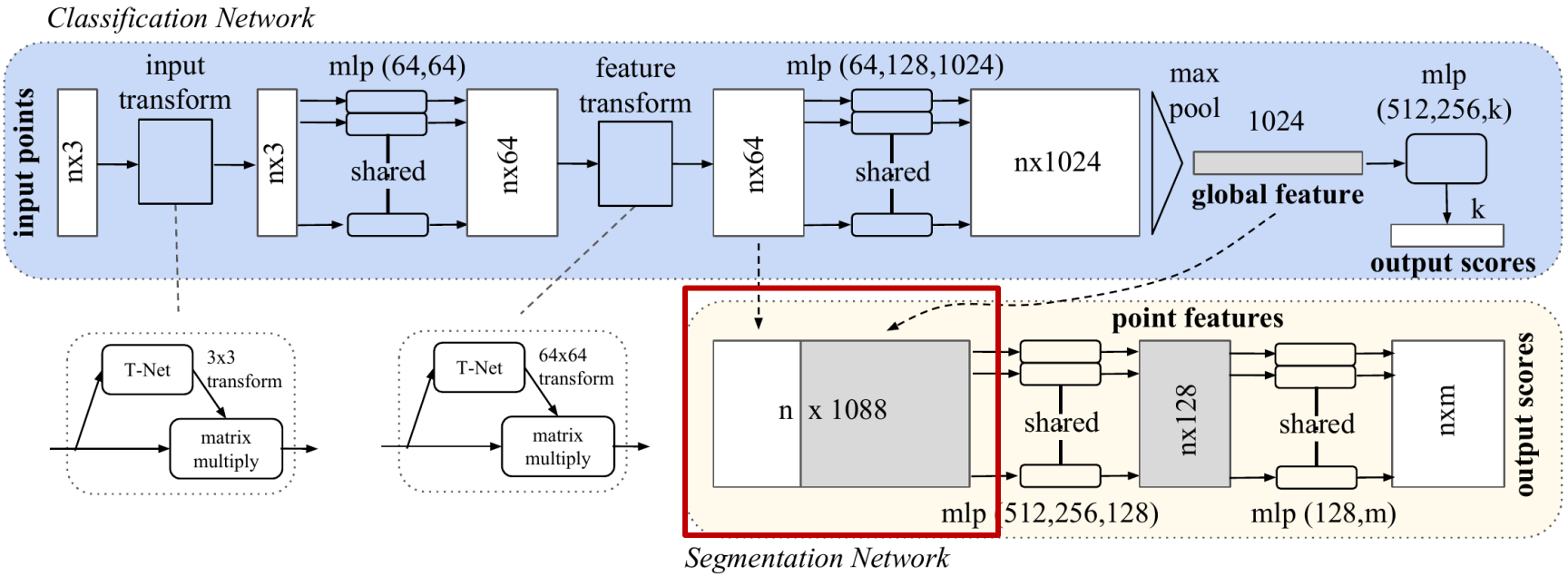
Joint Alignment Network:

align all input set to a canonical space before feature extraction.

predict an affine transformation matrix by a mini-network (**T-net**) and directly apply this transformation to the coordinates of input points.

Qi, Su, Mo and Guibas: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. CVPR, 2017.

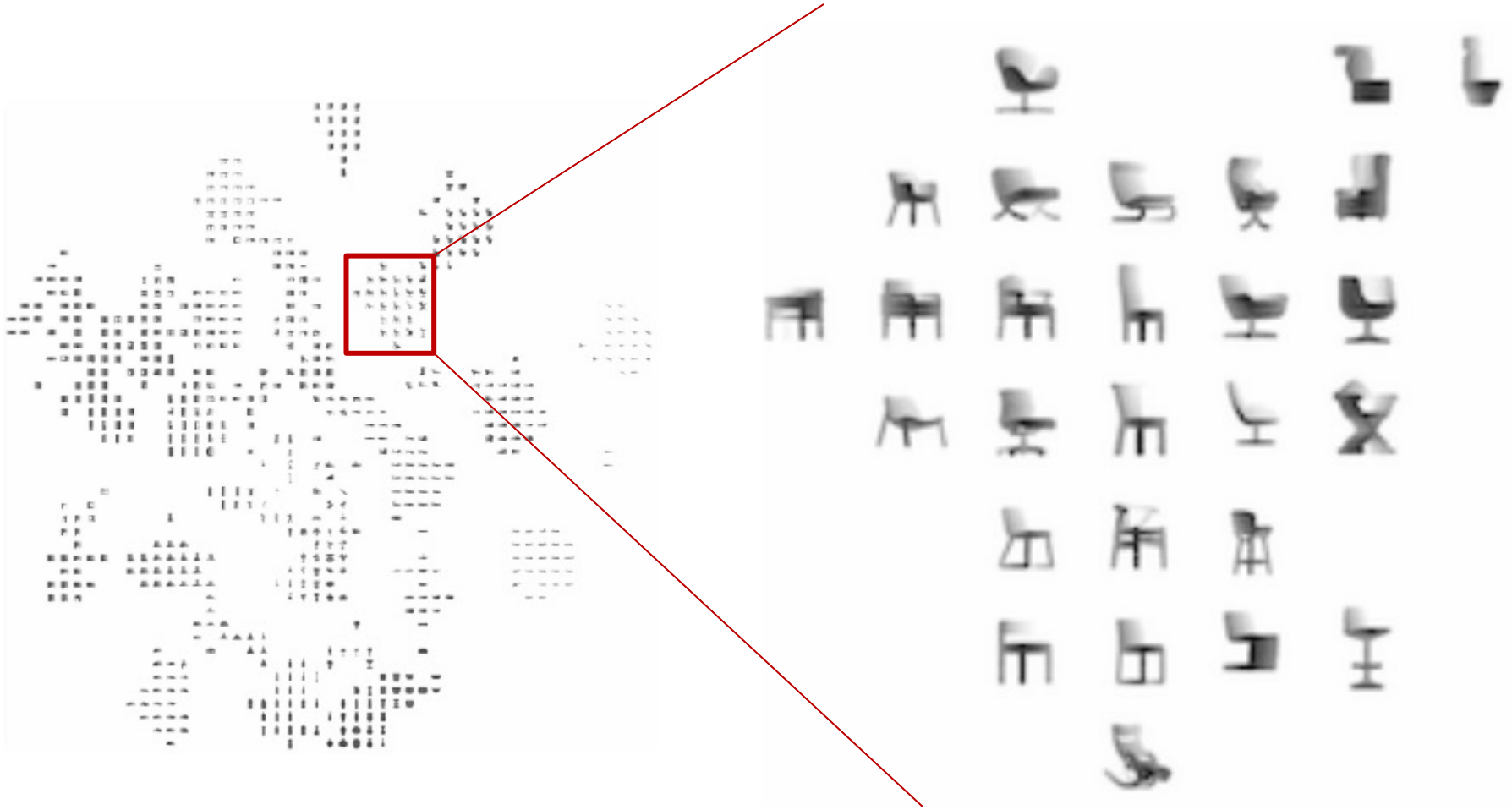
PointNet



Local and Global Information Aggregation:
After computing the global point cloud feature vector, feed it back to per point features by **concatenating** the global feature with each of the point features.

Qi, Su, Mo and Guibas: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. CVPR, 2017.

2D embedding of learnt shape global features



Qi, Su, Mo and Guibas: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. CVPR, 2017.

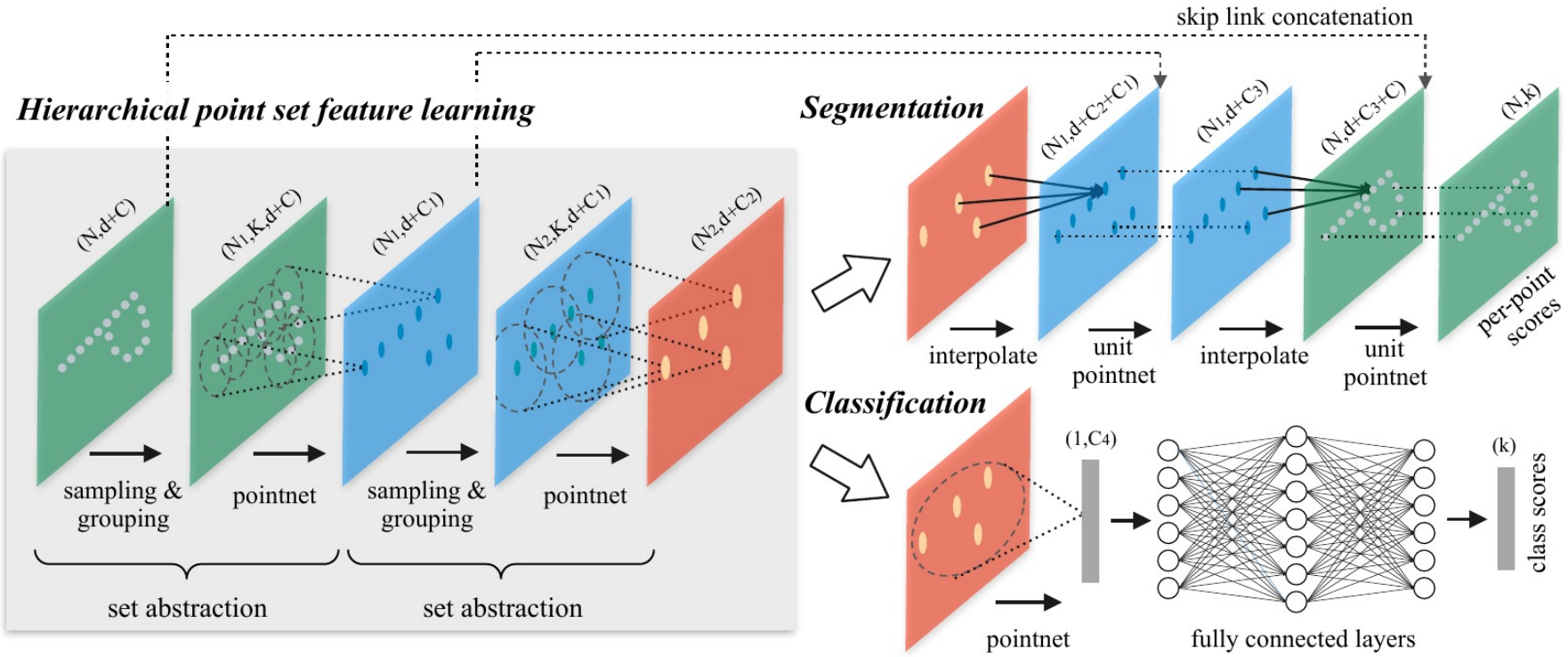
Any weakness of PointNet ?

Any weakness of PointNet ?

PointNet does not capture local structures and depends on global coordinates

→ PointNet ++

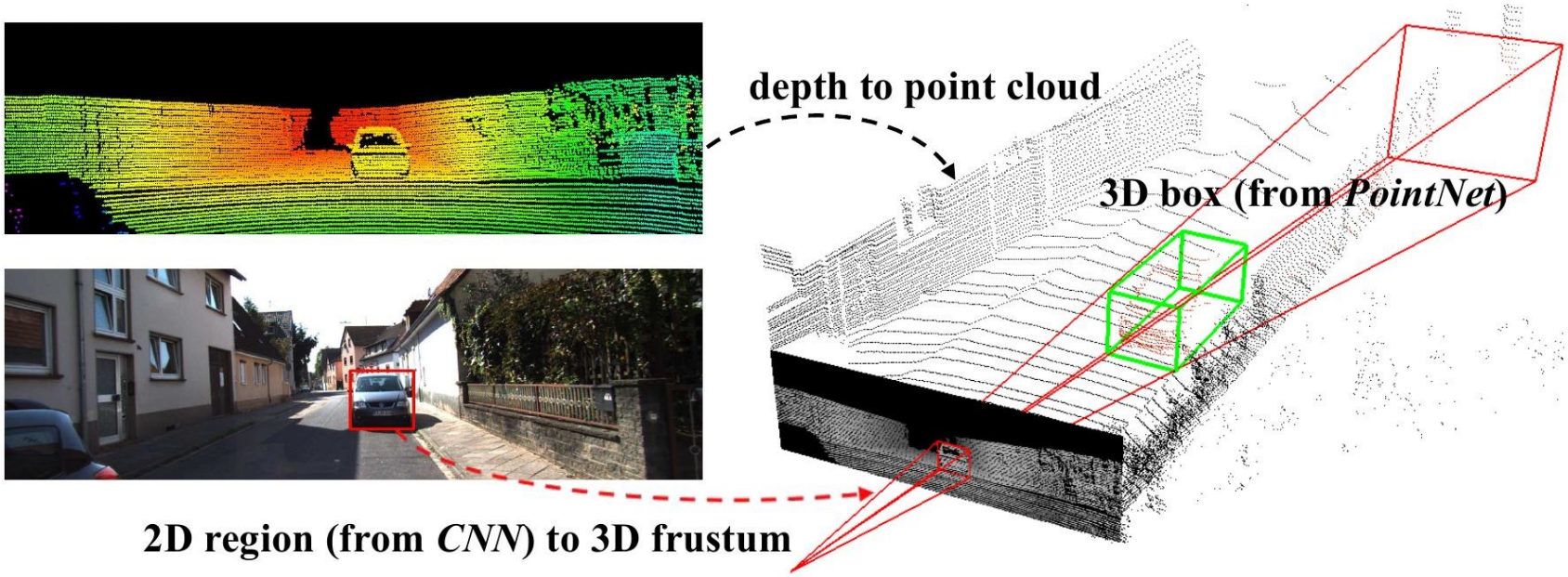
PointNet ++



- PointNet++ applies PointNet recursively on nested partitions of the point set
- It learns features with increasing contextual scales) → "Multi-scale PointNet"

Qi, Yi, Su and Guibas: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. NeurIPS, 2017.

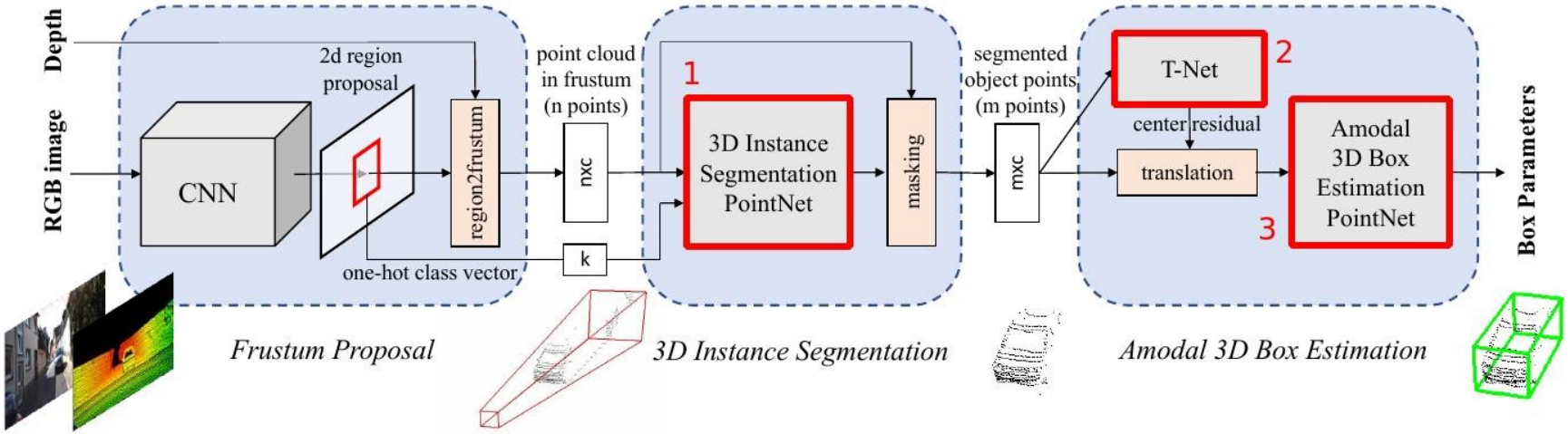
Point-based detection: Frustum PointNet



1. Generate 2D object proposals in the RGB image
2. Extract point cloud from 3D object frustum
3. Predict 3D bounding box from points in frustum via PointNet

Qi, Liu, Wu, Su and Guibas: Frustum PointNets for 3D Object Detection From RGB-D Data. CVPR 2018

Point-based detection: Frustum PointNet

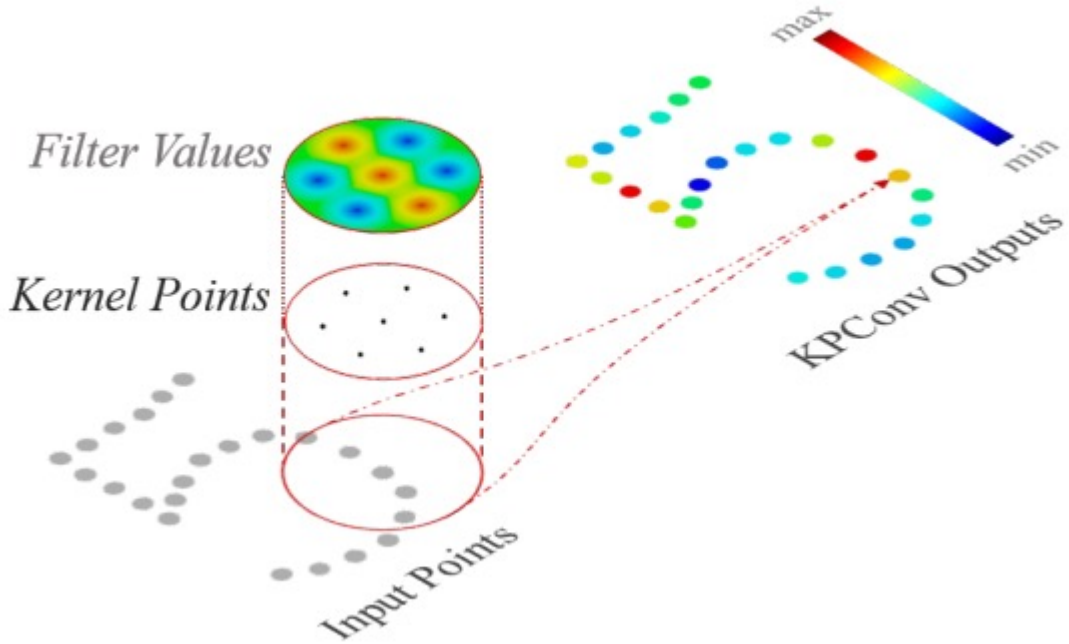


1. Frustum proposal generation
2. Segmentation of RGB point cloud (object vs. background)
3. Translation of points such that centroid aligns with 3D box center
4. 3D bounding box regression

Qi, Liu, Wu, Su and Guibas: Frustum PointNets for 3D Object Detection From RGB-D Data. CVPR 2018

Point-based segmentation: KPConv

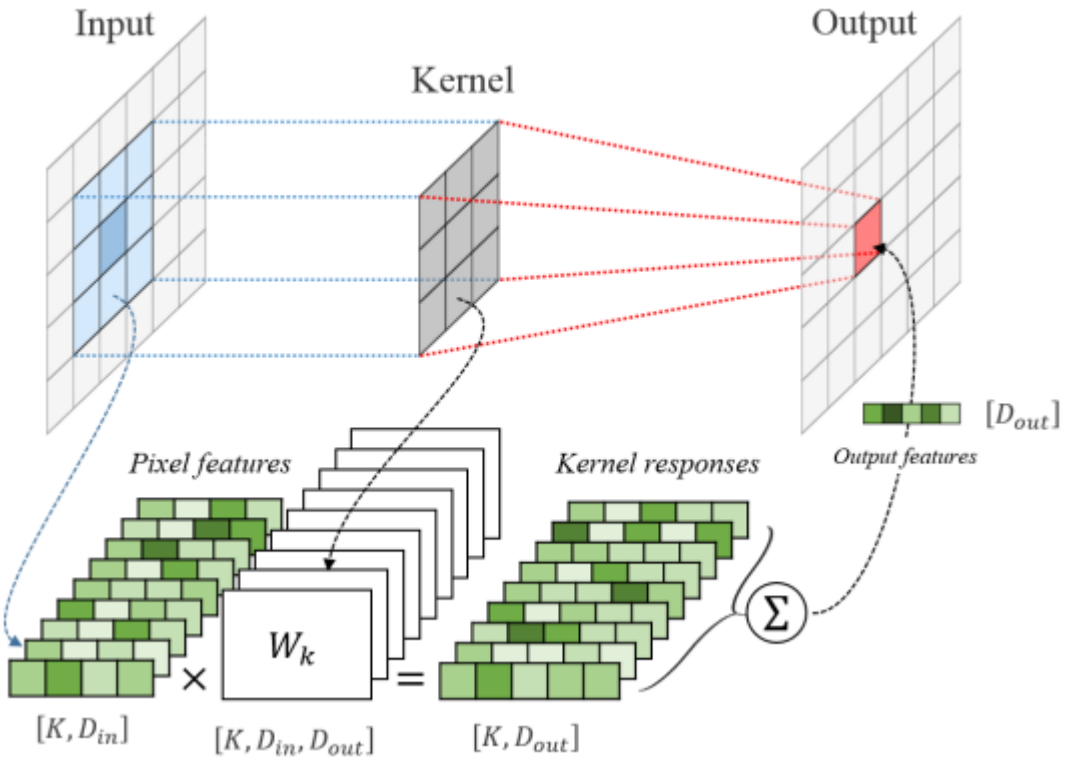
KPConv: Flexible and Deformable Convolution for Point Clouds
Define convolution operation on **raw points**



Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, Leonidas J. Guibas, KPConv: Flexible and Deformable Convolution for Point Clouds, ICCV 2019

Point-based segmentation: KPConv

Image-based convolution

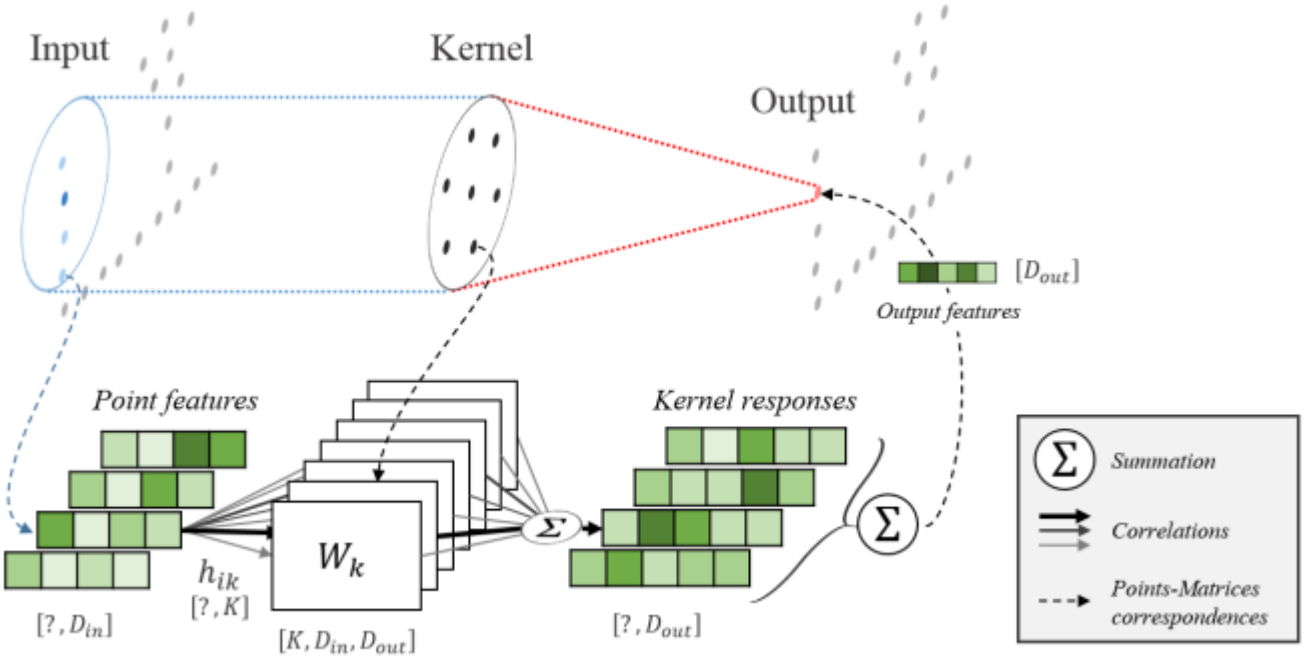


Aggregate the kernel response of local pixel

Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, Leonidas J. Guibas, KPConv: Flexible and Deformable Convolution for Point Clouds, ICCV 2019

Point-based segmentation: KPConv

KPConv



Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, Leonidas J. Guibas, KPConv: Flexible and Deformable Convolution for Point Clouds, ICCV 2019

Point-based segmentation: KPConv

Comparison

Similarity:

- Aggregate local kernel response

Difference:

- Calculation of kernel response
- Neighbor selection

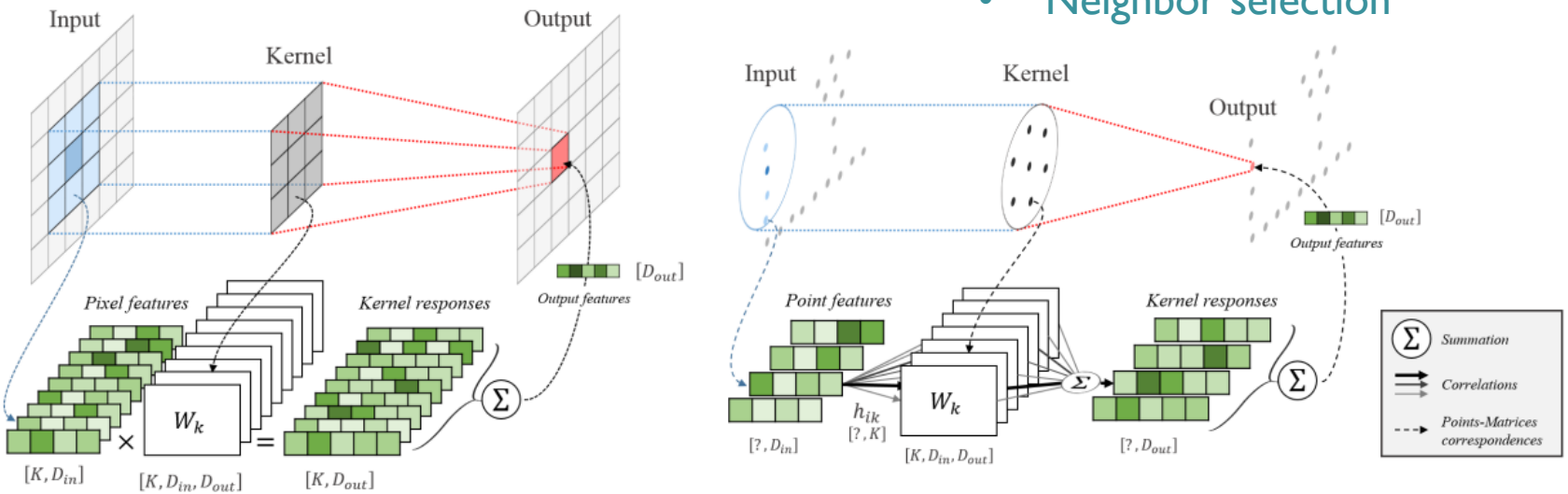


Figure 2. Comparison between an image convolution (left) and a KPConv (right) on 2D points for a simpler illustration. In the image, each pixel feature vector is multiplied by a weight matrix $(W_k)_{k<K}$ assigned by the alignment of the kernel with the image. In KPConv, input points are not aligned with kernel points, and their number can vary. Therefore, each point feature f_i is multiplied by all the kernel weight matrices, with a correlation coefficient h_{ik} depending on its relative position to kernel points.

Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, Leonidas J. Guibas, KPConv: Flexible and Deformable Convolution for Point Clouds, ICCV 2019

Point-based segmentation: KPConv

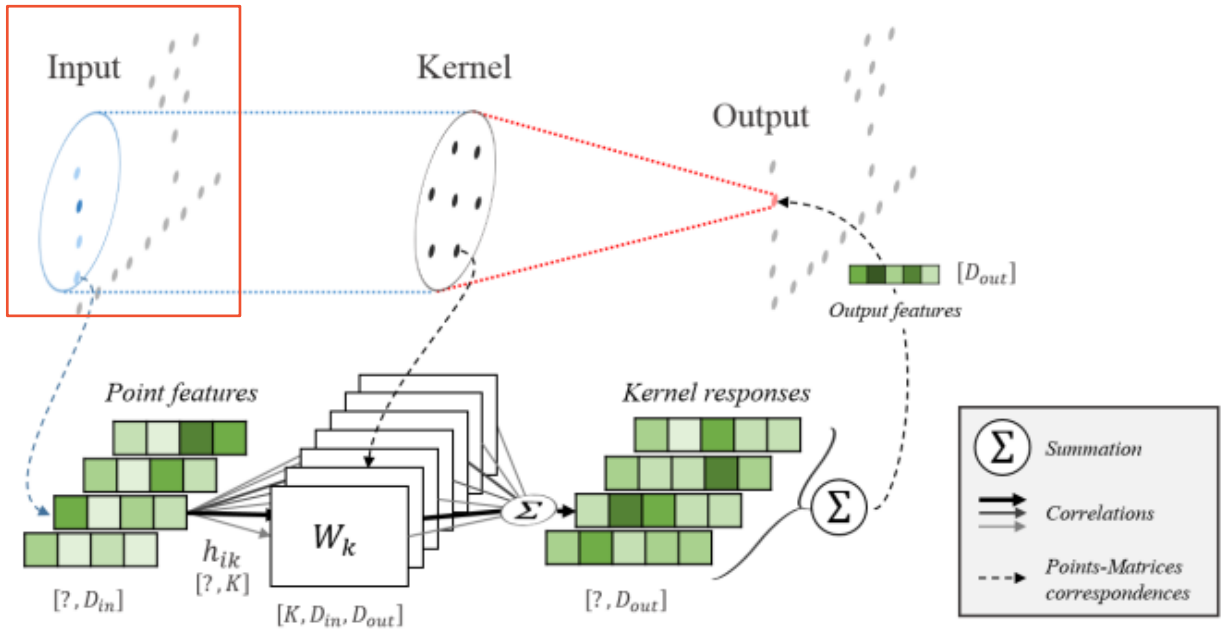
Kernel function

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x) f_i$$

x_i point coordinates
f_i point feature

$$\mathcal{N}_x = \{ x_i \in \mathcal{P} \mid \|x_i - x\| \leqslant r \}$$

Neighbor selection: ball query



Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, Leonidas J. Guibas, KPConv: Flexible and Deformable Convolution for Point Clouds, ICCV 2019

Point-based segmentation: KPConv

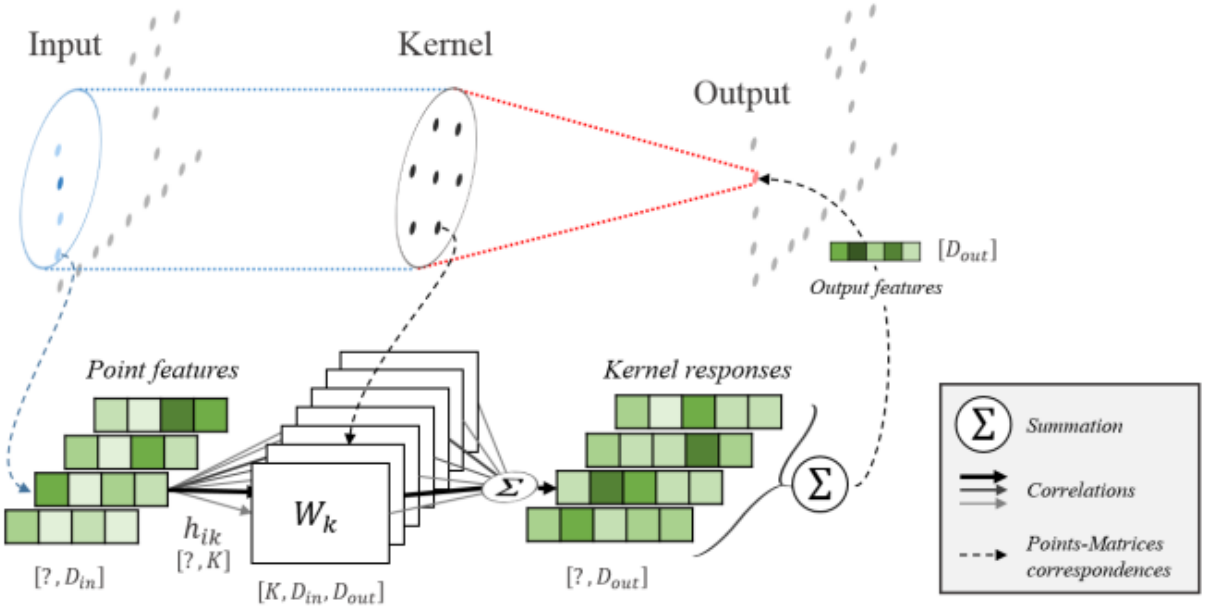
Kernel function

$$g(y_i) = \sum_{k<K} h\left(y_i, \widetilde{x}_k\right) W_k \qquad \{\widetilde{x}_k \mid k < K\}$$

$$h\left(y_i, \widetilde{x}_k\right) = \max\left(0, 1 - \frac{\|y_i - \widetilde{x}_k\|}{\sigma}\right)$$

K kernel points with learnable weights

σ is the influence distance of the kernel points



Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, Leonidas J. Guibas, KPConv: Flexible and Deformable Convolution for Point Clouds, ICCV 2019

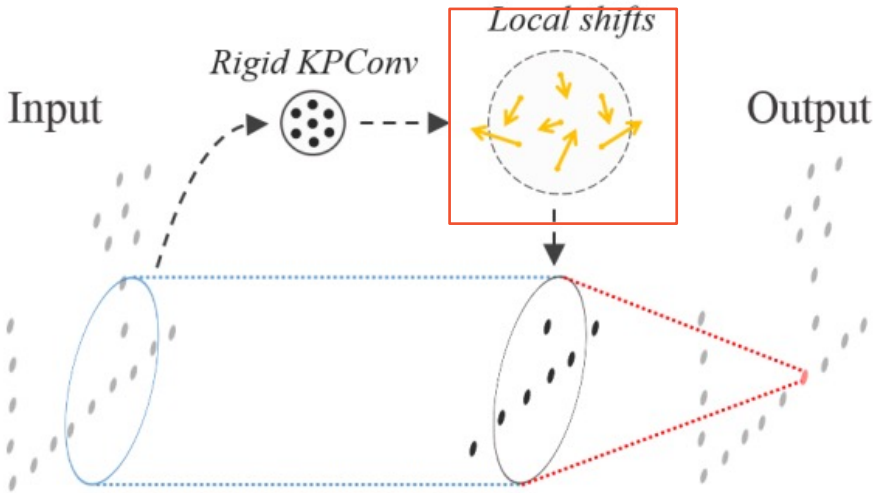
Point-based segmentation: KPConv

Deformable KPConv

The location of kernel points is predicted by a subnetwork

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g_{deform}(x - x_i, \Delta(x)) f_i$$

$$g_{deform}(y_i, \Delta(x)) = \sum_{k<K} h\left(y_i, \widetilde{x}_k + \Delta_k(x)\right) W_k$$



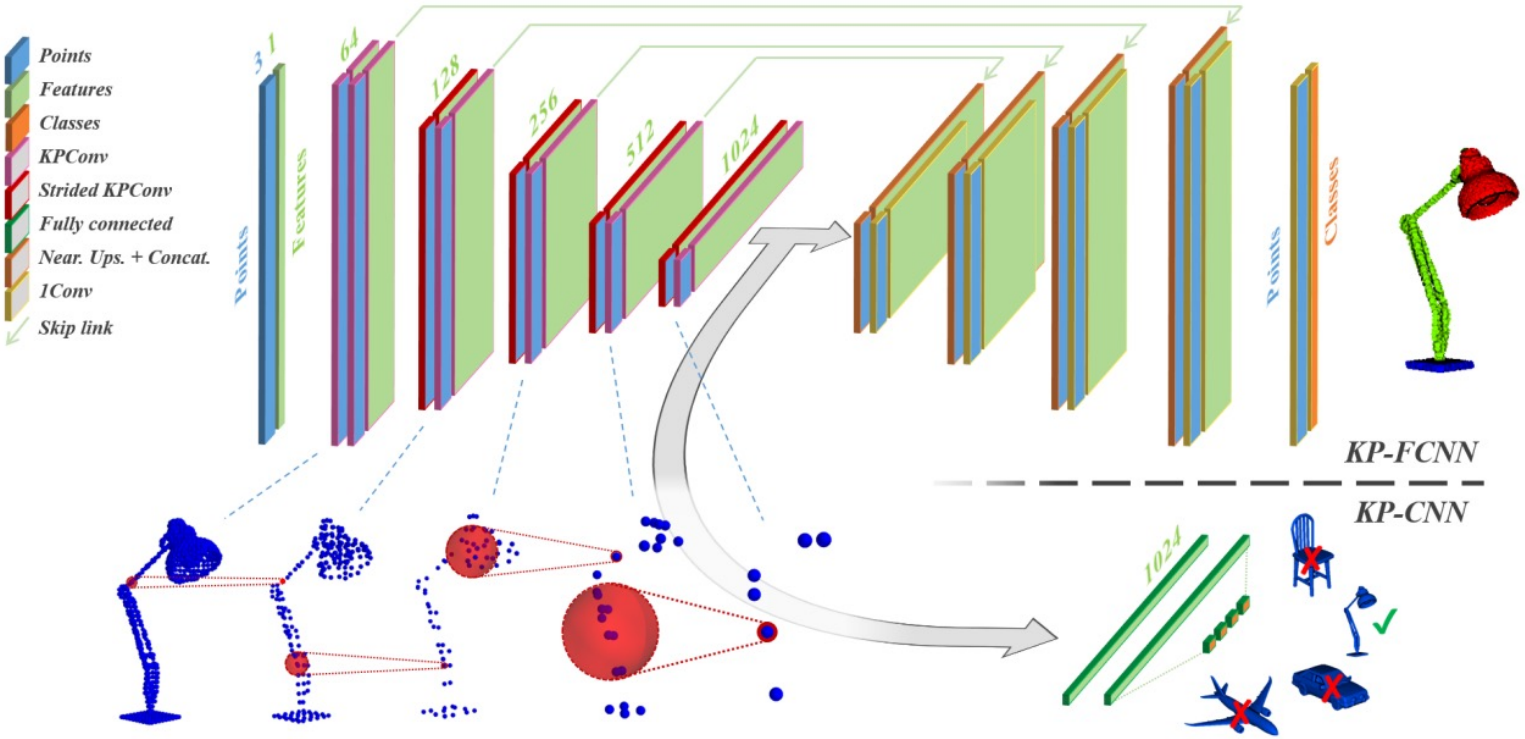
Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, Leonidas J. Guibas, KPConv: Flexible and Deformable Convolution for Point Clouds, ICCV 2019

Point-based segmentation: KPConv

Overall architecture

- Downsample: grid sample
- Upsample: nearest upsample

UNet architecture with Strided KPConv and Upsample layer for semantic segmentation



Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, Leonidas J. Guibas, KPConv: Flexible and Deformable Convolution for Point Clouds, ICCV 2019

Point-based segmentation: KPConv

Experimental results

	ModelNet40	ShapeNetPart	
Methods	OA	mcIoU	mIoU
SPLATNet [34]	-	83.7	85.4
SGPN [42]	-	82.8	85.8
3DmFV-Net [9]	91.6	81.0	84.3
SynSpecCNN [48]	-	82.0	84.7
RSNet [15]	-	81.4	84.9
SpecGCN [40]	91.5	-	85.4
PointNet++ [27]	90.7	81.9	85.1
SO-Net [19]	90.9	81.0	84.9
PCNN by Ext [2]	92.3	81.8	85.1
SpiderCNN [45]	90.5	82.4	85.3
MCConv [13]	90.9	-	85.9
FlexConv [10]	90.2	84.7	85.0
PointCNN [20]	92.2	84.6	86.1
DGCNN [43]	92.2	85.0	84.7
SubSparseCNN [9]	-	83.3	86.0
KPConv *rigid*	**92.9**	85.0	86.2
KPConv *deform*	92.7	**85.1**	**86.4**

3D Shape Classification and Segmentation

Methods	Scannet	Sem3D	S3DIS	PL3D
Pointnet [26]	-	-	41.1	-
Pointnet++ [27]	33.9	-	-	-
SnapNet [4]	-	59.1	-	-
SPLATNet [34]	39.3	-	-	-
SegCloud [37]	-	61.3	48.9	-
RF_MSSF [38]	-	62.7	49.8	56.3
Eff3DConv [50]	-	-	51.8	-
TangentConv [36]	43.8	-	52.6	-
MSDVN [30]	-	65.3	54.7	66.9
RSNet [15]	-	-	56.5	-
FCPN [28]	44.7	-	-	-
PointCNN [20]	45.8	-	57.3	-
PCNN [2]	49.8	-	-	-
SPGraph [17]	-	73.2	58.0	-
ParamConv [41]	-	-	58.3	-
SubSparseCNN [9]	**72.5**	-	-	-
KPConv *rigid*	68.6	**74.6**	65.4	72.3
KPConv *deform*	68.4	73.1	**67.1**	**75.9**

3D scene segmentation

Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, Leonidas J. Guibas, KPConv: Flexible and Deformable Convolution for Point Clouds, ICCV 2019

Some datasets for 3D object detection

KITTI:

- 3,712 training samples
- 3,769 validation samples
- 7,518 testing samples

Nuscenes:

- 28k training samples
- 6k validation samples
- 6k testing samples

Waymo open dataset

- 798 training sequences with around 158, 361 LiDAR samples
- 202 validation sequences with 40, 077 LiDAR samples

Some datasets for 3D segmentation

SemanticKITTI:

- 28 outdoor classes
- 11 sequences for training and validation
- 11 sequences for testing

nuScenes:

- 32 outdoor classes
- 850 scenes for training and validation
- 150 scenes for testing

S3DIS:

- 13 indoor classes
- 5 areas for training
- 1 area for validation

Some famous codebase

- Point Cloud Library (PCL): https://github.com/PointCloudLibrary/pcl
- Spconv: https://github.com/traveller59/spconv
- Det3D: https://github.com/poodarchu/Det3D
- Mmdetection3d: https://github.com/open-mmlab/mmdetection3d

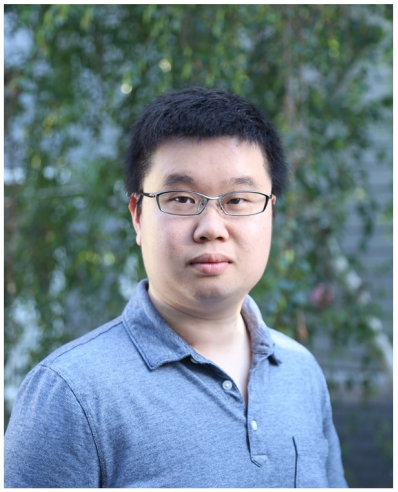
Next lecture:

NVIDIA

RESEARCH | RESEARCH AREAS | PUBLICATIONS | AI PLAYGROUND | TEAM | CAREERS | LICENSING

PEOPLE

Zhiding Yu



Zhiding Yu is a Senior Research Scientist at NVIDIA. Before joining NVIDIA in 2018, he received a Ph.D. in ECE from Carnegie Mellon University in 2017, and an M.Phil. in ECE from The Hong Kong University of Science and Technology in 2012. His research interests mainly focus on deep representation learning, weakly/self-supervised learning, transfer learning and deep structured prediction, with their applications to vision and robotics problems. He is a winner of the Domain Adaptation for Semantic Segmentation Track, WAD Challenge at CVPR 2018. He is an author of the best student paper at ISCSLP 2014 and the best paper awards at WACV 2015 / BMVC 2020. His intern work on deep facial expression recognition at Microsoft Research won first runner-up at the EmotiW-SFEW Challenge 2015 and was integrated into the Microsoft Emotion Recognition API under the Microsoft Azure Cognitive Services.

For more information, please visit his Homepage.

Research Area(s):

- Computer Vision
- Robotics

Main Field of Interest:

Artificial Intelligence and Machine Learning

Google Scholar:

https://scholar.google.com/citations?user=1VI_oYUAAAAJ&hl=en

Thank you very much!

sihengc@sjtu.edu.cn