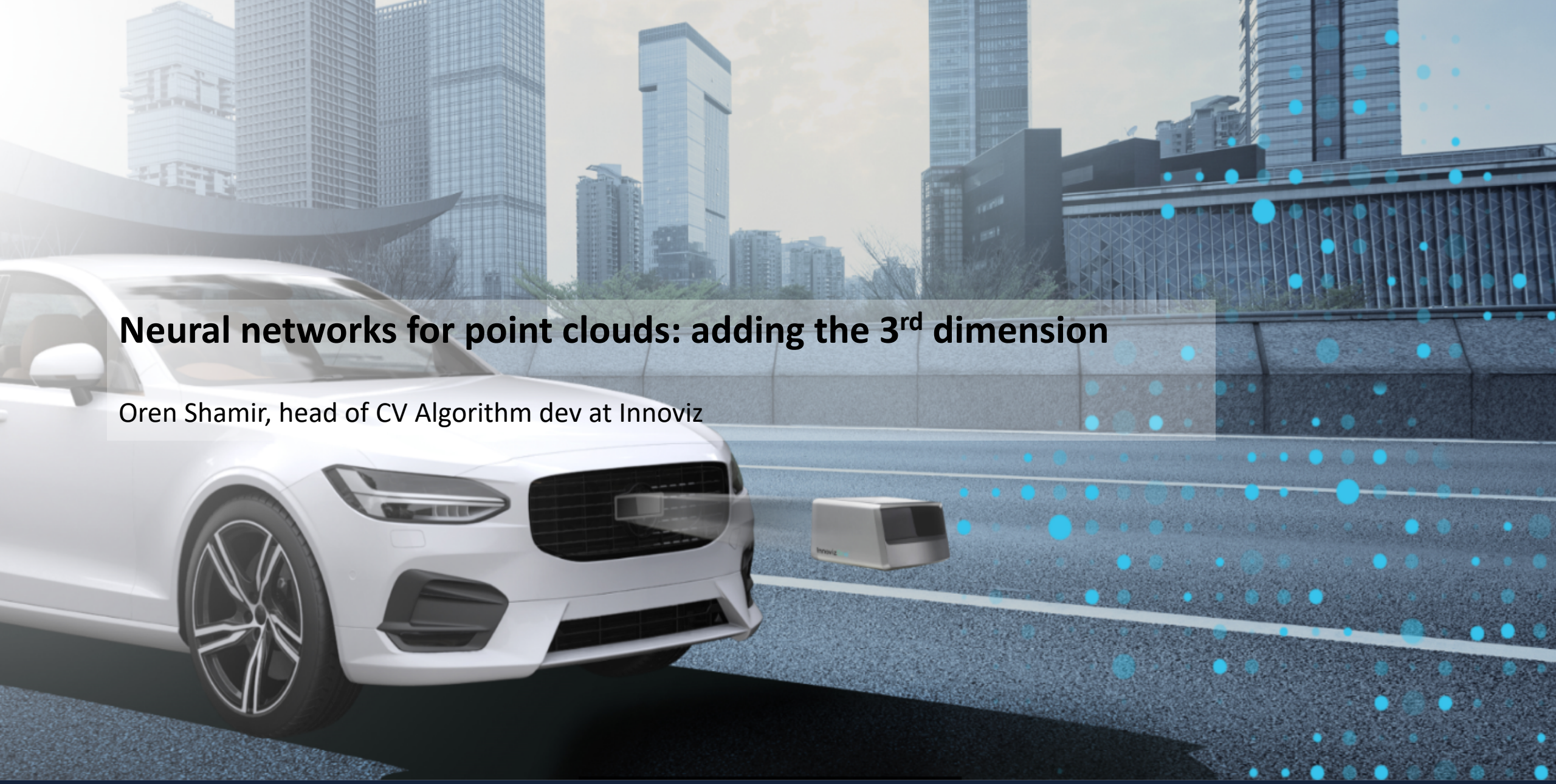


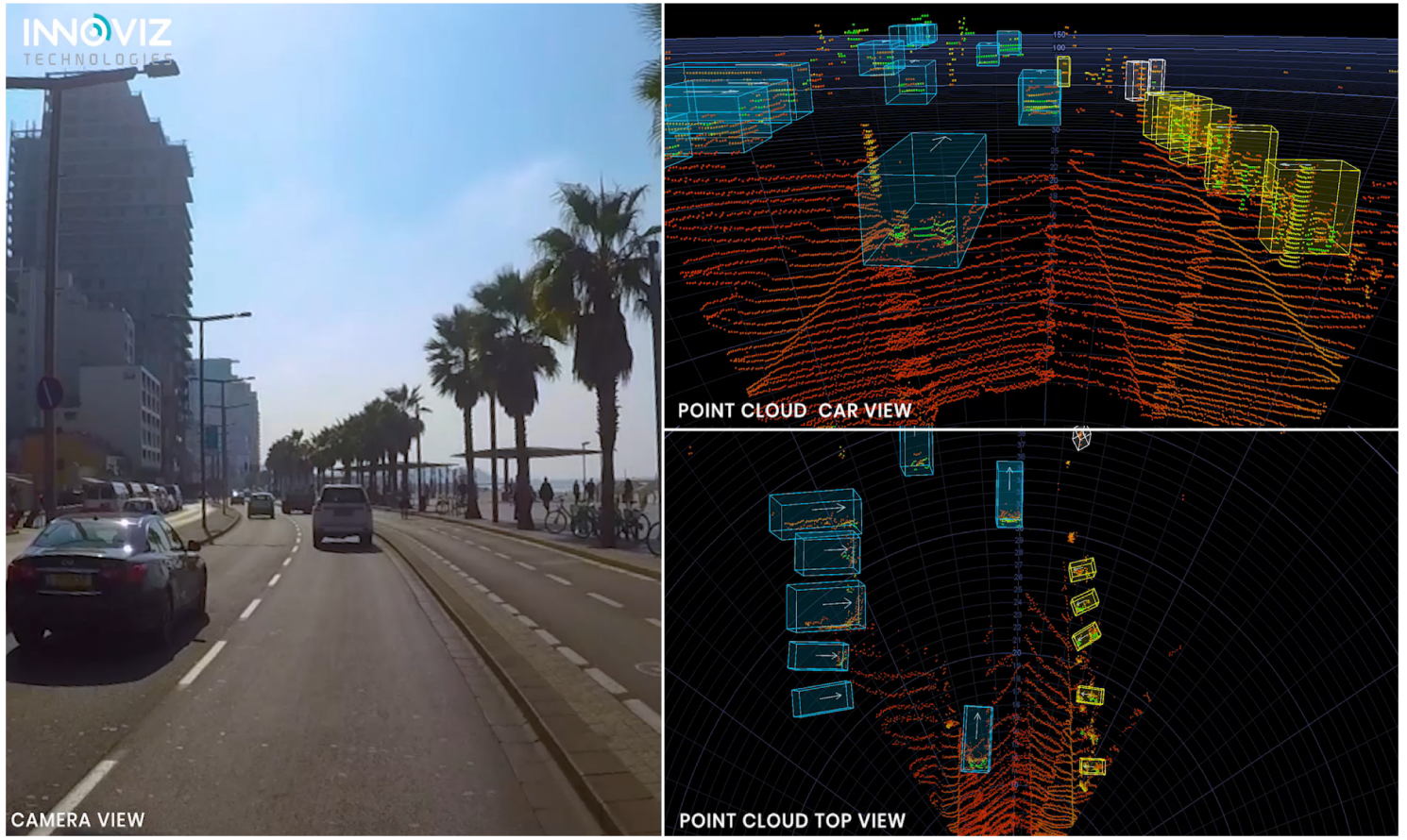
Neural networks for point clouds: adding the 3rd dimension

Oren Shamir, head of CV Algorithm dev at Innoviz

INNOVIZ TECHNOLOGIES

Innoviz Technologies

- Founded in 2016
- ~200 employees
- Developing a solid-state LiDAR
- CV stack over 3D data for perception on autonomous vehicles



Source: Innoviz lidar

TOC

- Introduction to 3D data
- Representations and insights
- NN architectures for 3D data

Source: https://www.template.net/design-templates/3d/3d-street-art-painting/

Why is 3D sensor data important

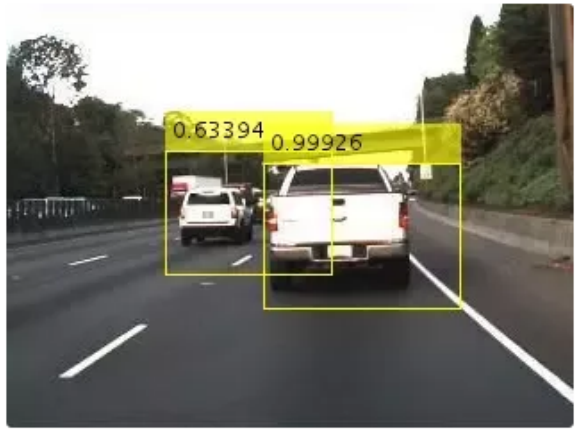
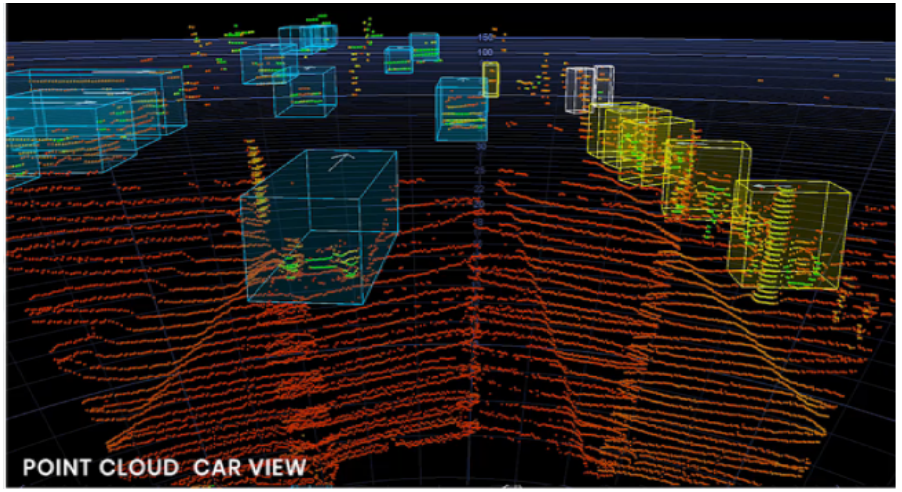
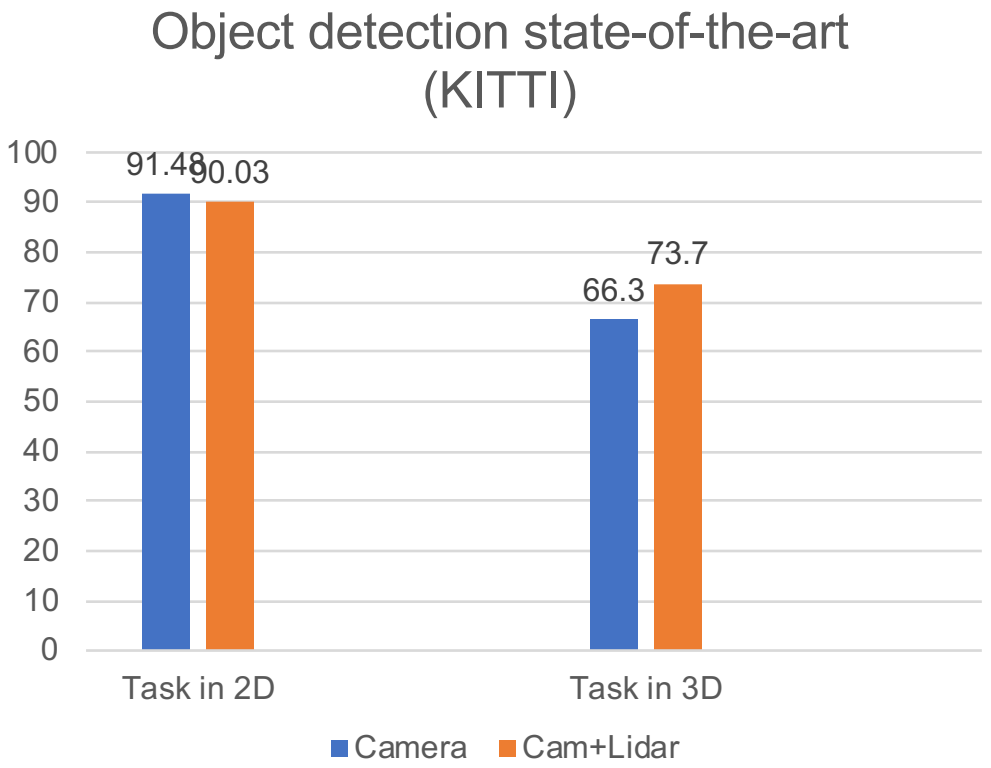


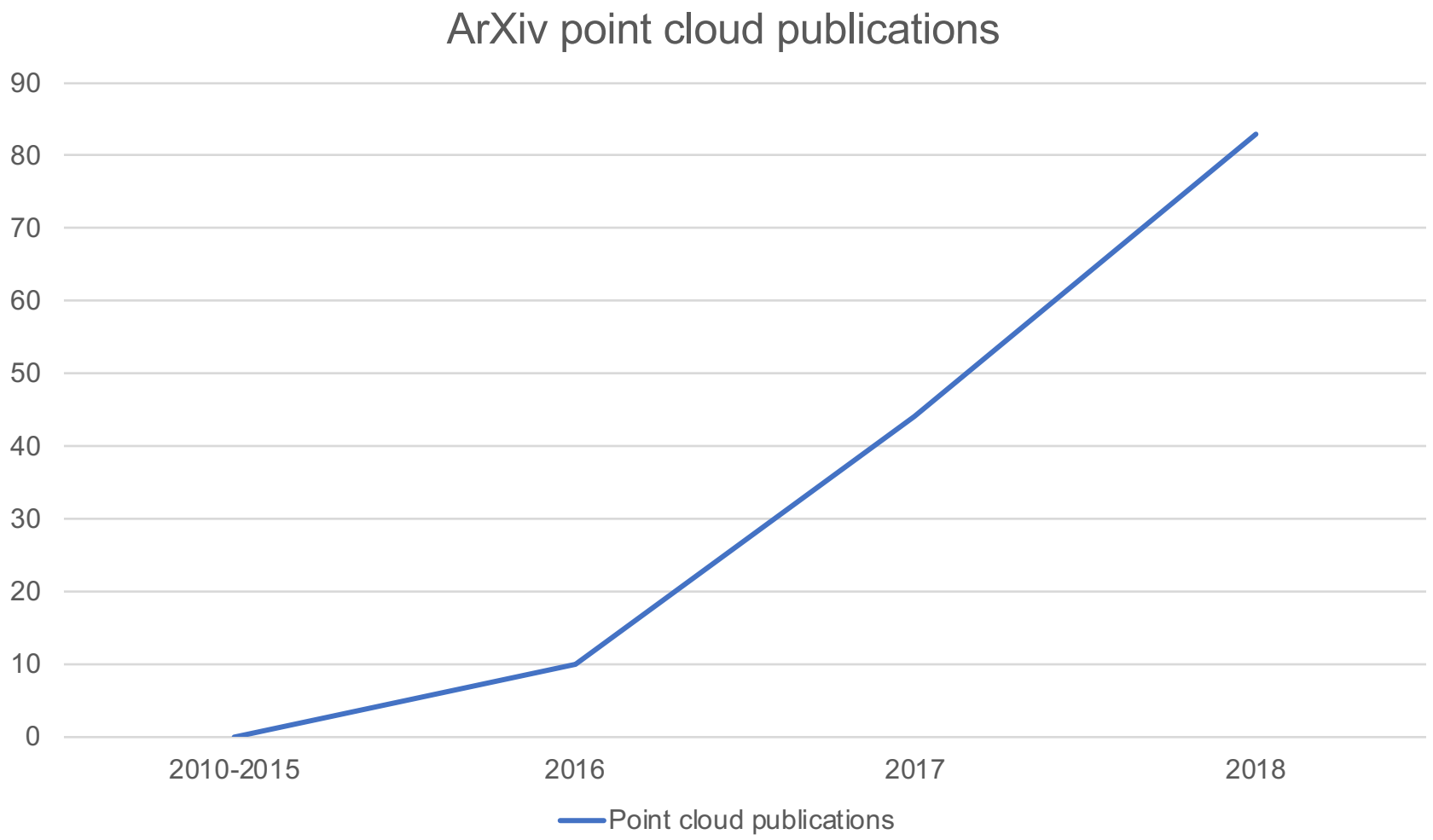
Image source: Features - Computer Vision System Toolbox



Source: Innoviz lidar

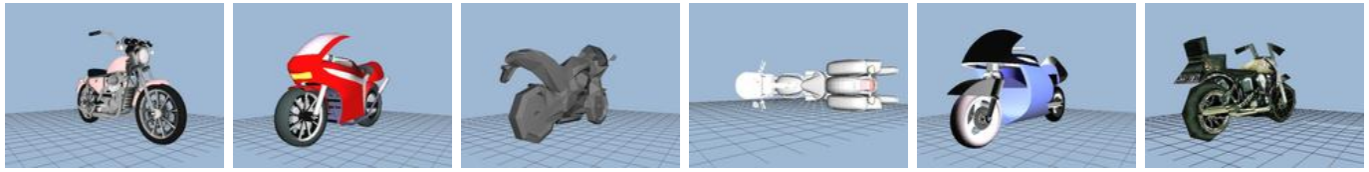


Research trend on 3D data processing



3D data sources

- CAD models
  - Full 3D data (including self occlusions)
  - Polygon / curve representation

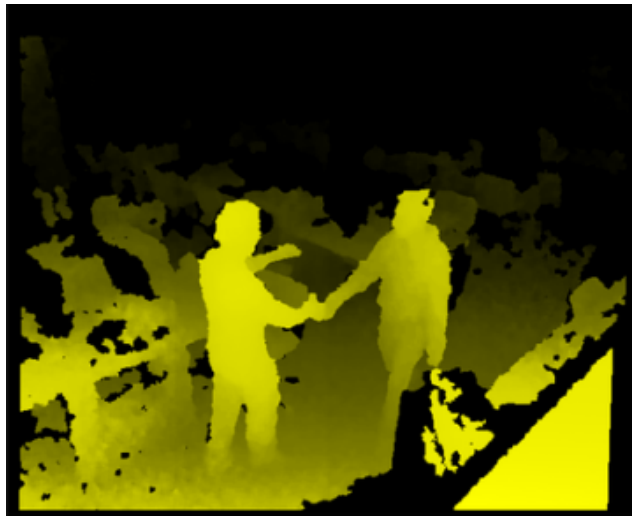


Source: http://modelnet.cs.princeton.edu/

- Sensor acquired
  - Typically only view data from one perspective, resulting in self-occlusions (exception is object scanning)
    Also referred to sometimes as '2.5D' data
  - Main industries: medical, automotive, mapping, VR
  - Sensor types – Stereo, RGB-D cameras (e.g. Kinect), fMRI, CT, lidar, radar
  - Sensor technologies – TOF, structured light, phase shift
  - High variability of data properties depending on technology and sensor

Source: Innoviz lidar

Common 3D data representations

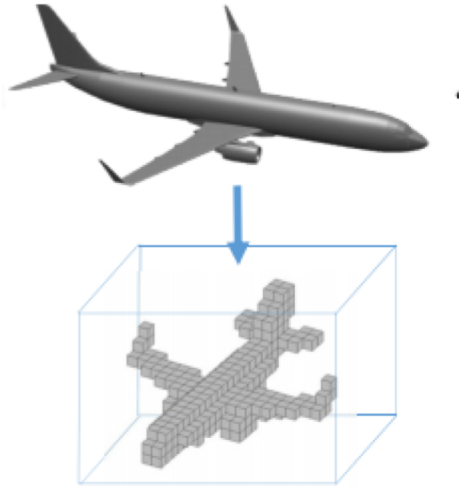


Depth map

2D matrix, depth as value

W*H*bpp

Source: RGB-D-KingstonRGB-D-dataset

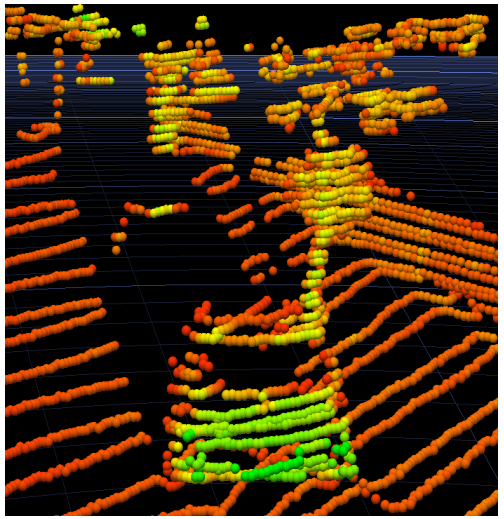


Voxels

3D matrix, occupancy as value

W*H*D

Source:Volumetric and Multi-View CNNs for Object Classification on 3D Data, Charles R. Qi et al



Point cloud

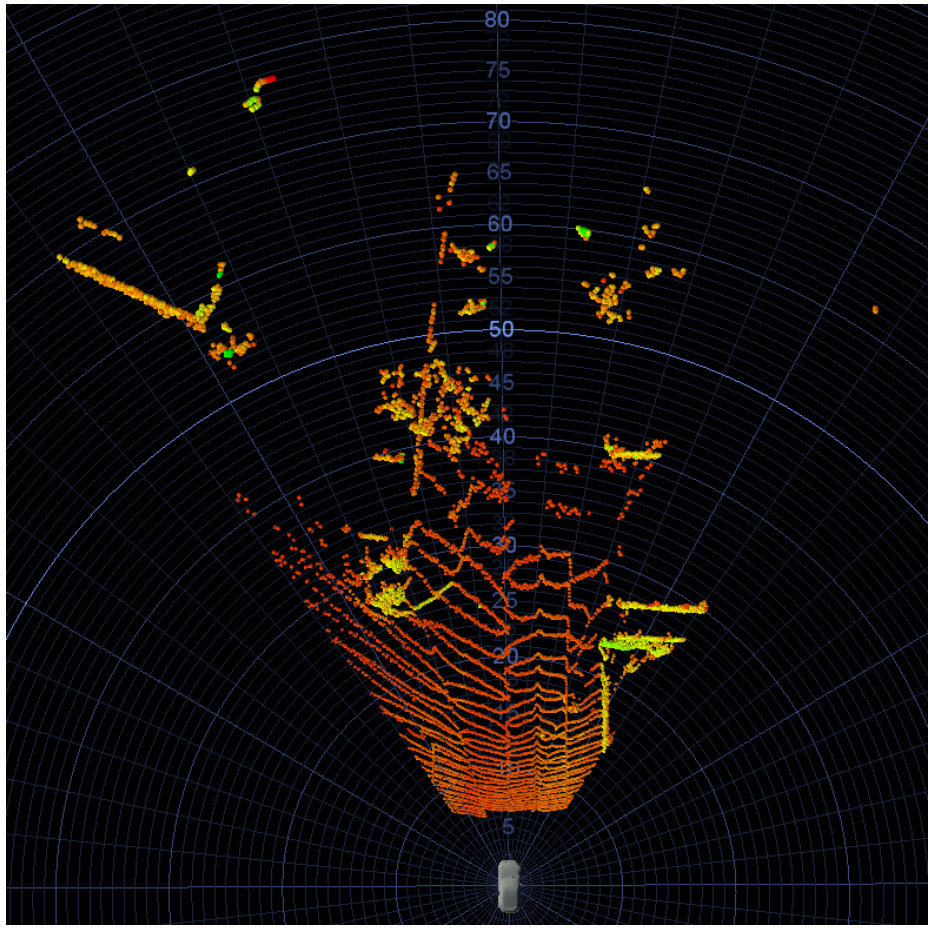
1D matrix, location as value

N*3

Source: Innoviz lidar

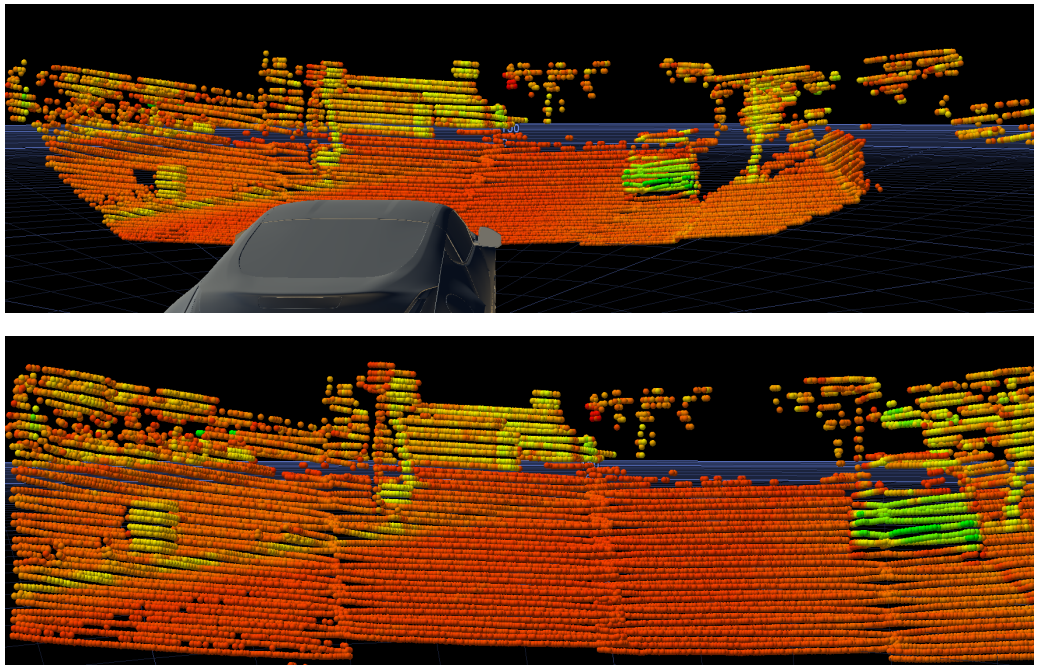
Challenges

- Data & Annotation
  - Few open datasets, many of which are synthetic
  - Few existing 3D annotation tools
  - Strong expertise needed to perform manual 3D annotations
- Sparsity – typically, most of the 3D space of interest is unoccupied, resulting in inefficiency
- Constant angular resolution, but variable resolution in world coordinates
- Irregular grids
- variable data size
- Meaning of measurement changes with distance



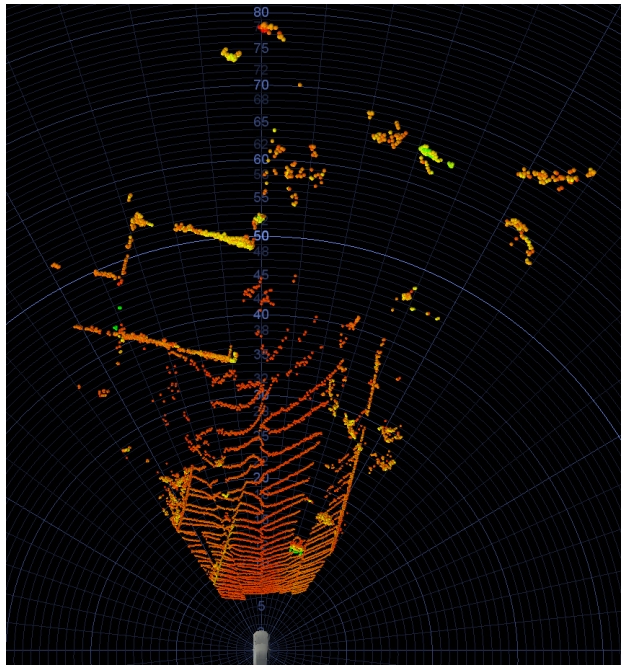
Source: Innoviz lidar

2D projections of 3D data



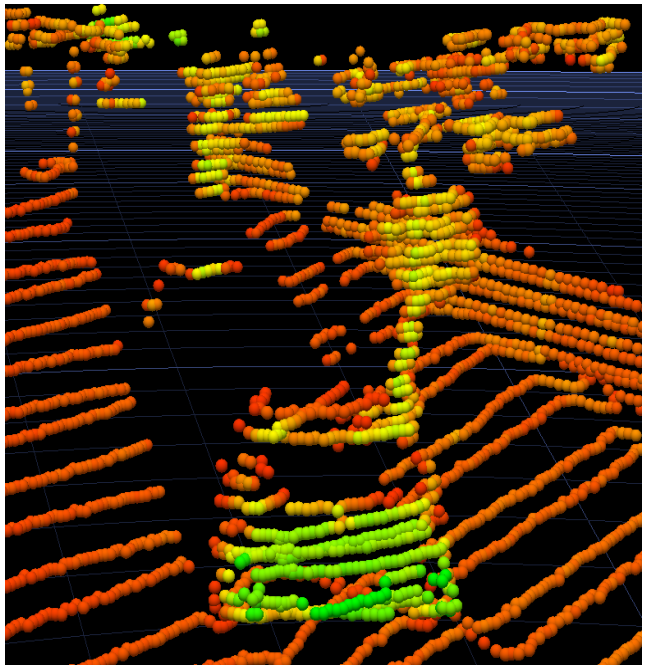
Front view (FV)

Source: Innoviz lidar



Birds eye view (BEV)

Source: Innoviz lidar



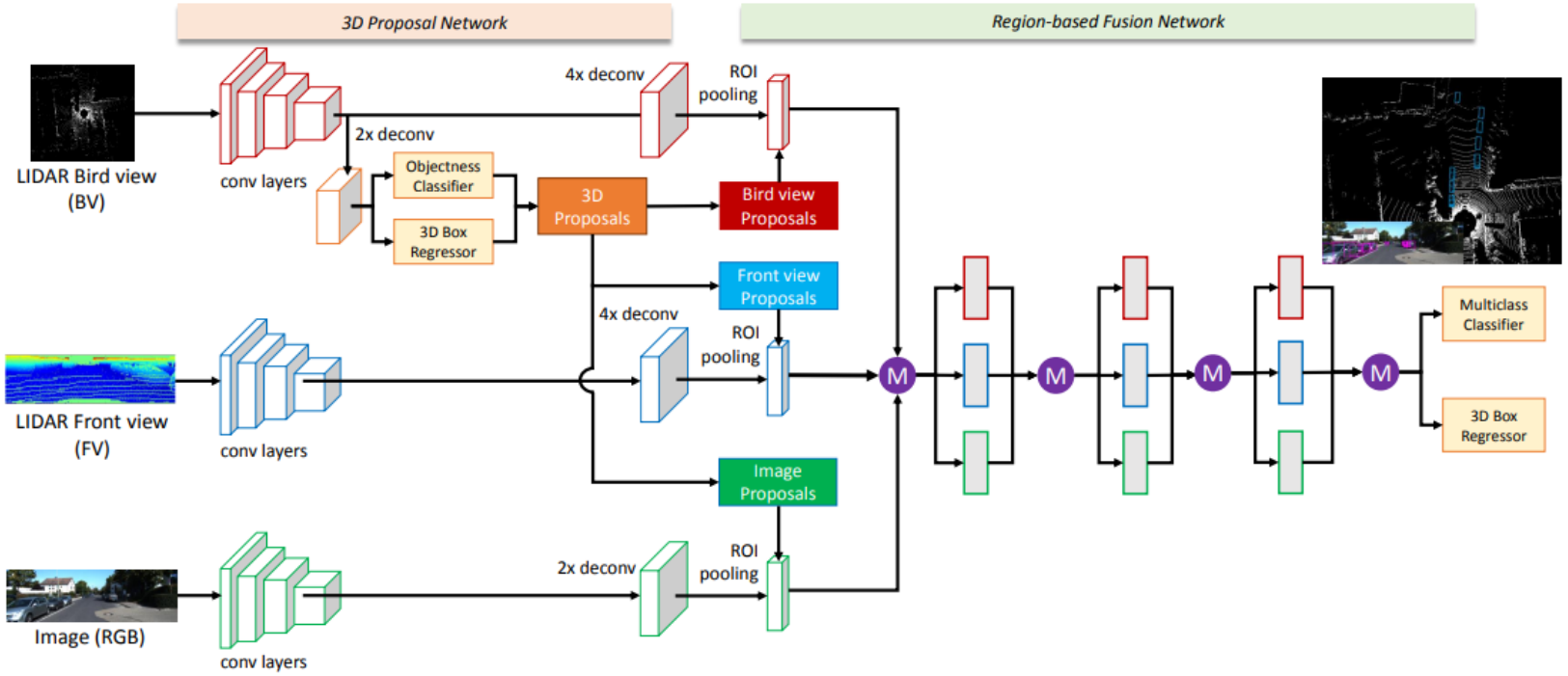
Custom view

Source: Innoviz lidar

* Showing reflectivity / intensity channel (projections can include other layers of data)

MV3D

- Xiaozhi Chen , Huimin Ma , Ji Wan , Bo Li , Tian Xia: Multi-View 3D Object Detection Network for Autonomous Driving In CVPR, 2017
- Multi-view approach – multiple, predefined 2D projections of 3D data
- Predefined projections – BEV + FV, fused with RGB data



MV3D

- RPN is based on BEV features
- BEV features include height maps, intensity information
- FV and RGB image used for 2nd stage of detection
- FV features include intensity, distance and height maps

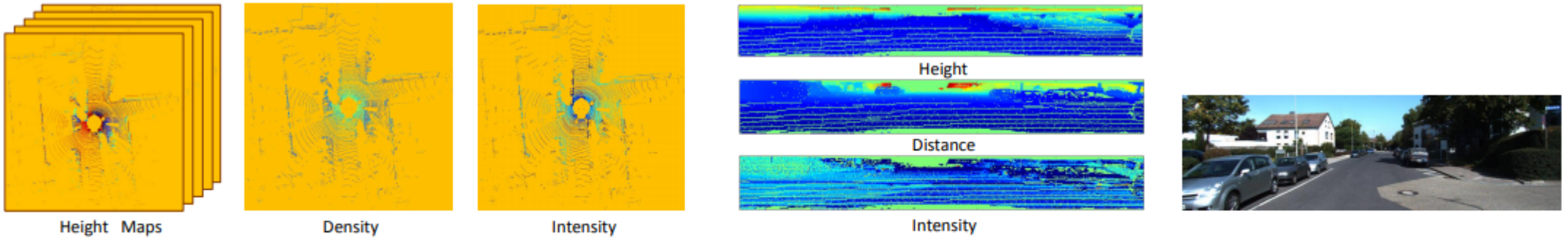


Figure 2: Input features of the MV3D network.

Source: Multi-View 3D Object Detection Network for Autonomous Driving, cvpr 2017

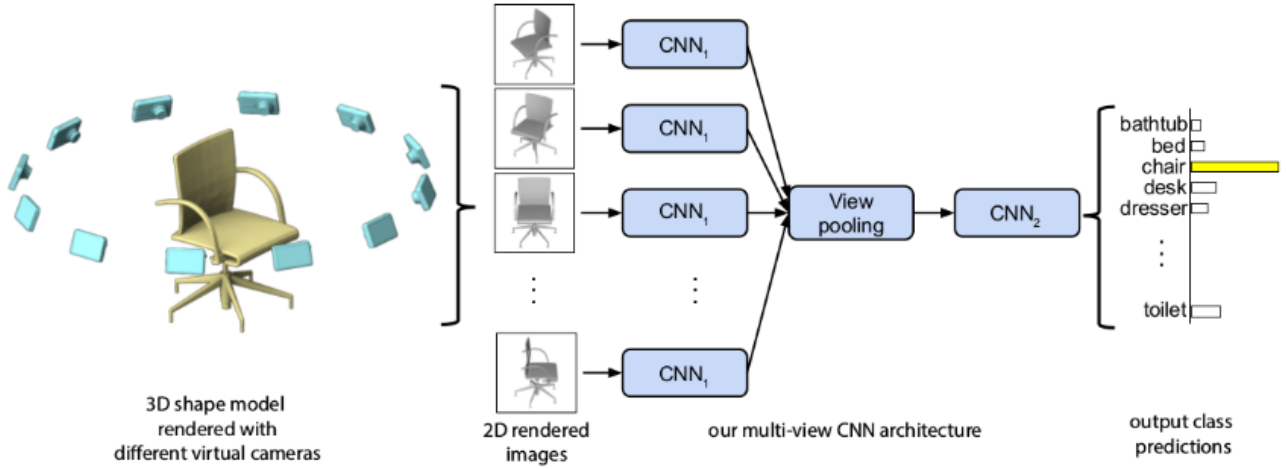
MV3D

Advantages

- Leverages architectures and optimizations done for 2D networks
- Computationally reasonable
- Takes advantage of advantageous properties of different views

Issues

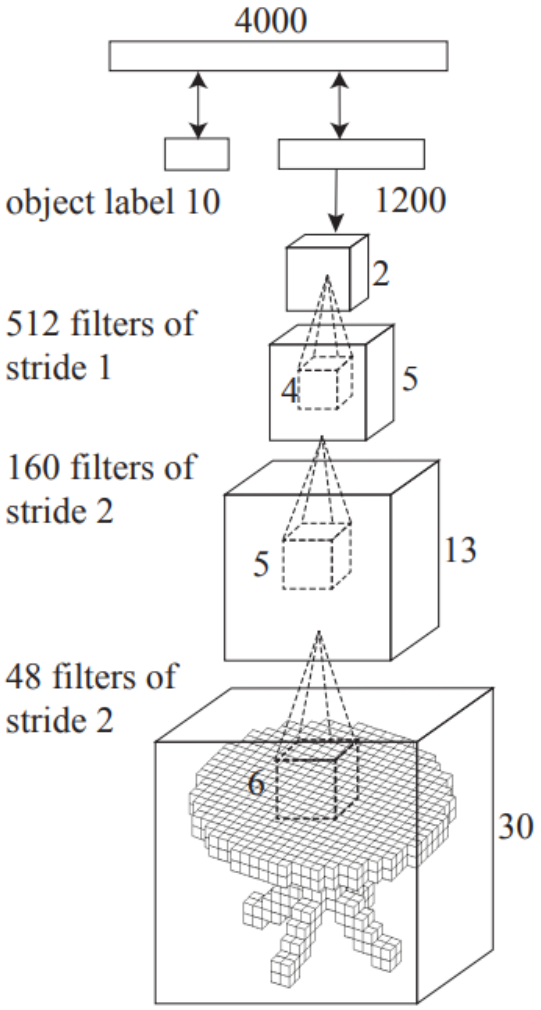
- Specific views and representations (features) are a form of feature engineering
- Sparsity (in the sense of missing data) common in some views



Source: http://vis-www.cs.umass.edu/mvcnn/

3D convolutions

- Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In CVPR 2015
- Represent the 3D data as a volumetric occupancy grids (voxels)
- Turning 2.5D to 3D by **marking each voxel as surface, free space or occluded**
- 3D convolutional layers:
  - Can learn local (in 3D) spatial filters
  - Multiple stacked layers enlarge 3D receptive field and represent increasingly complex features
  - 3D layers include convolution, pooling layers
- Introduced ModelNet – a CAD model dataset still used as a benchmark for classification of 3D shapes



Source: 3D Shapenets

3D convolutions

- D. Maturana and S. Scherer: VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. IROS, 2015
- Work on point cloud and RGB-D camera data
- Mitigation for rotation invariance (especially around Z axis) - using augmentation (carefully with 2.5D data)

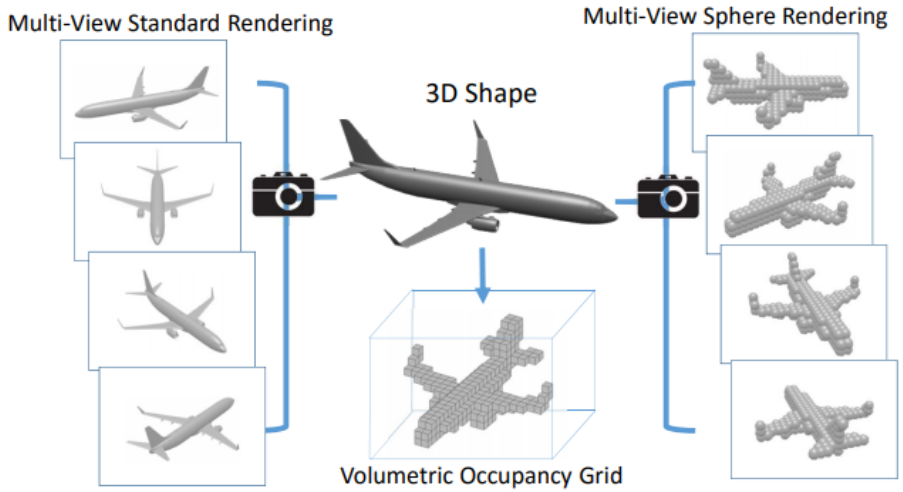
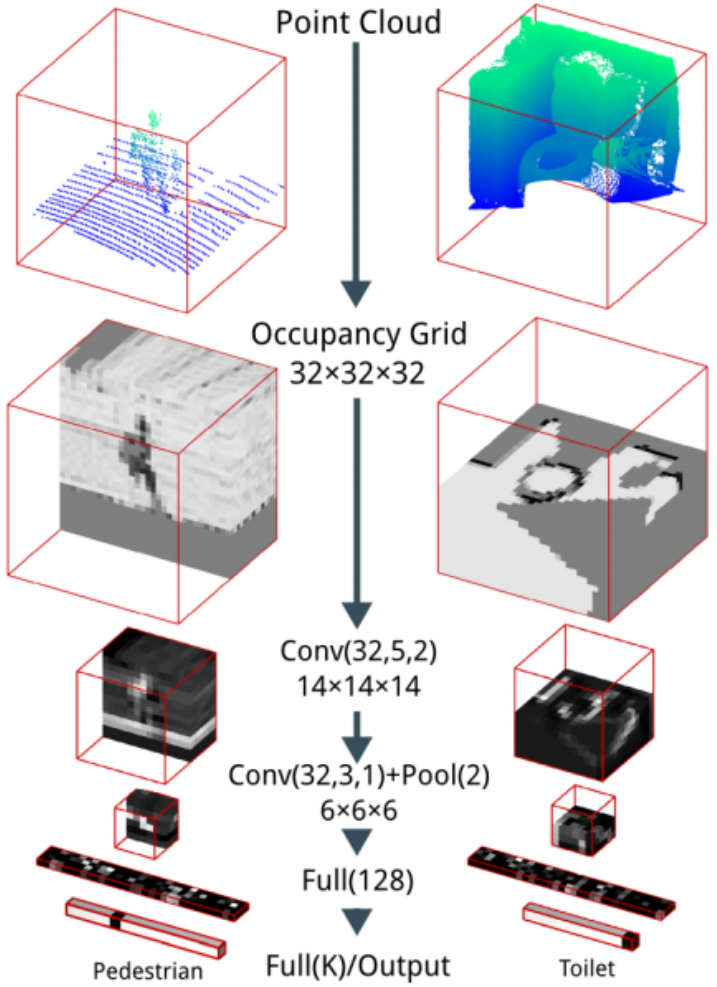


Figure 1. 3D shape representations.

Source: Charles R. Qi, Hao Su et al: Volumetric and Multi-View CNNs for Object Classification on 3D Data



Source: VoxNet

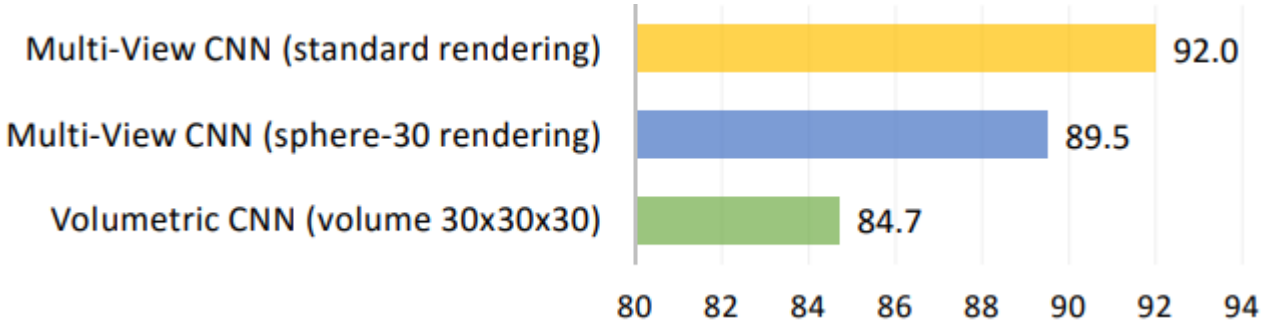
3D convolutions

Advantages

- Natural extension of 2D operations (convolution, pooling)
- Encoding of unknown / occluded areas
- Ability to process data from various 3D sensors (2.5D and 3D)

Issues

- Memory and computational complexity – 227x227 ~= 30x30x30
- Real-life performance is inferior to multi-view approaches



Source: Charles R. Qi, Hao Su et al: Volumetric and Multi-View CNNs for Object Classification on 3D Data

Pointnet

- Charles R. Qi, Hao Su et al: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. CVPR 2017)
- Design of a neural network which directly consumes point clouds
- Given that a point cloud is an **unordered** set of points, each with d properties (coordinates in Euclidian space and additional properties), the desired properties for a neural network working on point clouds should include:
  - Permutation invariance

Represents the same set as

  - Rotation invariance

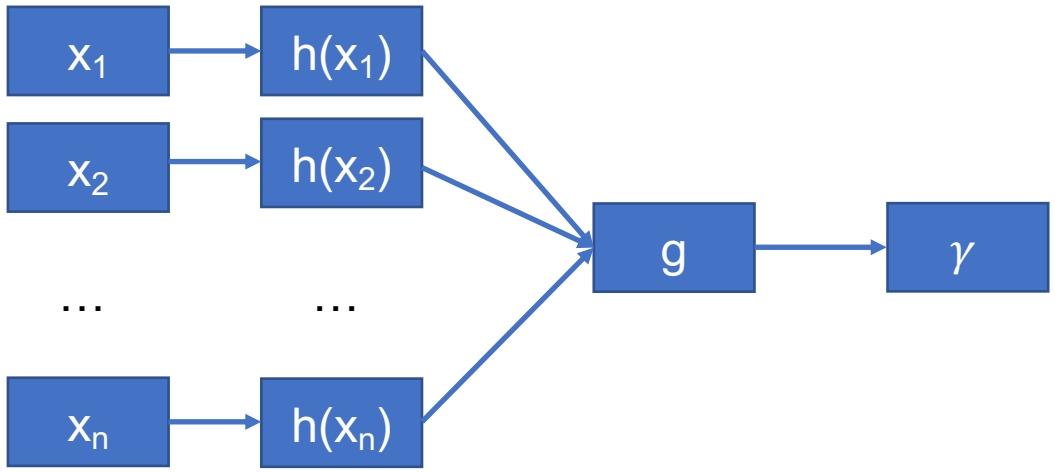
Represents the same object as

Source: https://p3d.in/L7sdw

Pointnet – permutation invariance

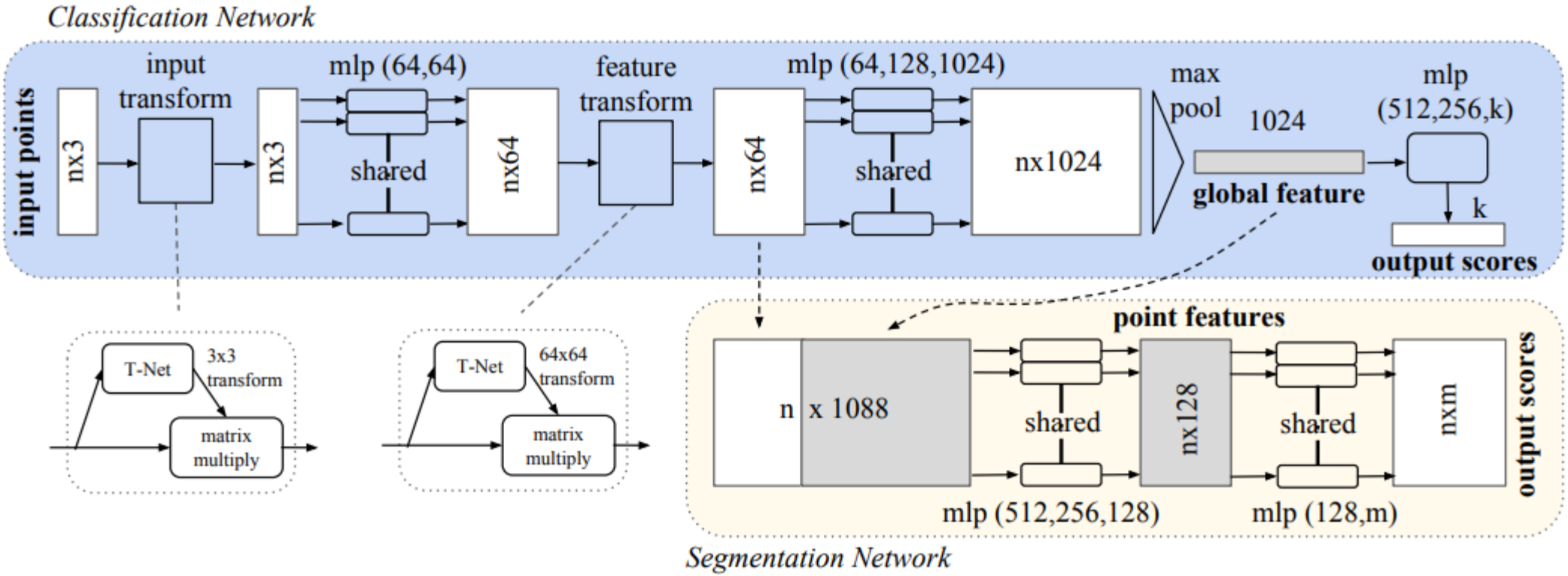
In order to achieve the property of permutation invariance, we note that:

- Given that $f(x_1, \ldots x_n) = \gamma(g(h(x_1),\ldots,h(x_n)))$
- If function g is symmetric → function f is symmetric



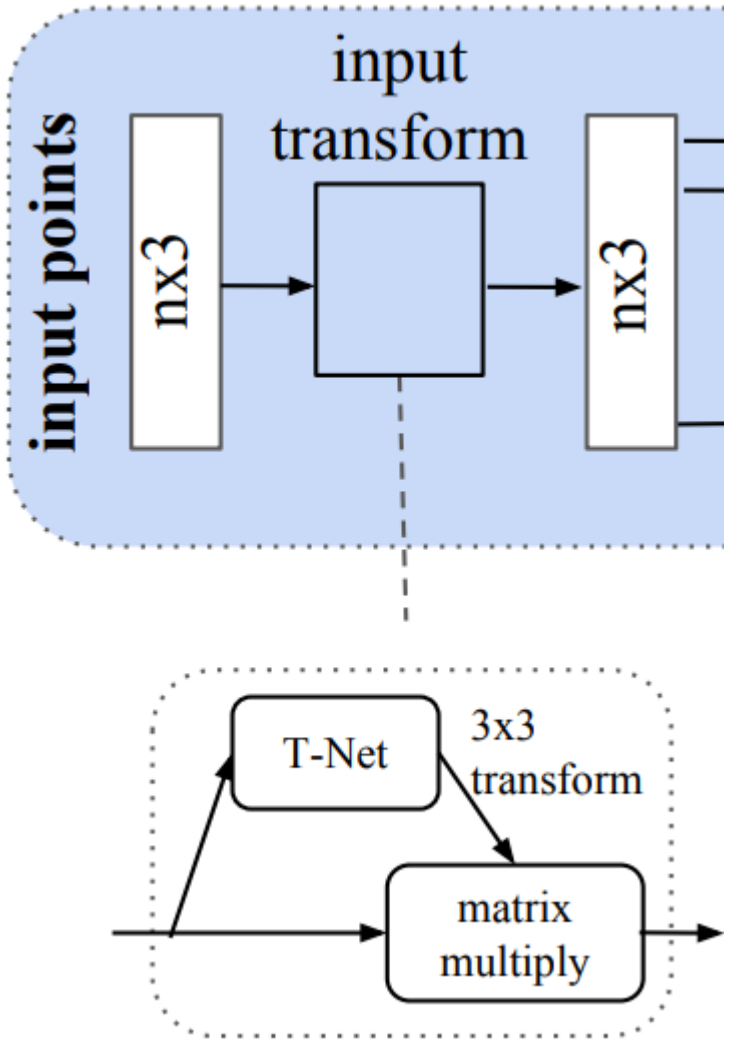
- Symmetric functions (g) explored in Pointnet paper include max, avg

Pointnet architecture



Pointnet – rotation invariance

- Transformation invariance is a property highly desirable for computer vision tasks
- Pointnet aims to transform (e.g. rotate) the point cloud within the network
- A mini-pointnet, denoted as T-net, predicts a transformation matrix T which will transform the point cloud to a canonical viewpoint
- The transformation is then applied to the pointcloud: $P_t=T*P^T$
- The rest of the network now works on a canonically-transformed point cloud, where the transformation is learned inline with the given task
- The article used a 3x3 matrix, but can be extended to allow for translation as well by using a 4x4 matrix



Pointnet

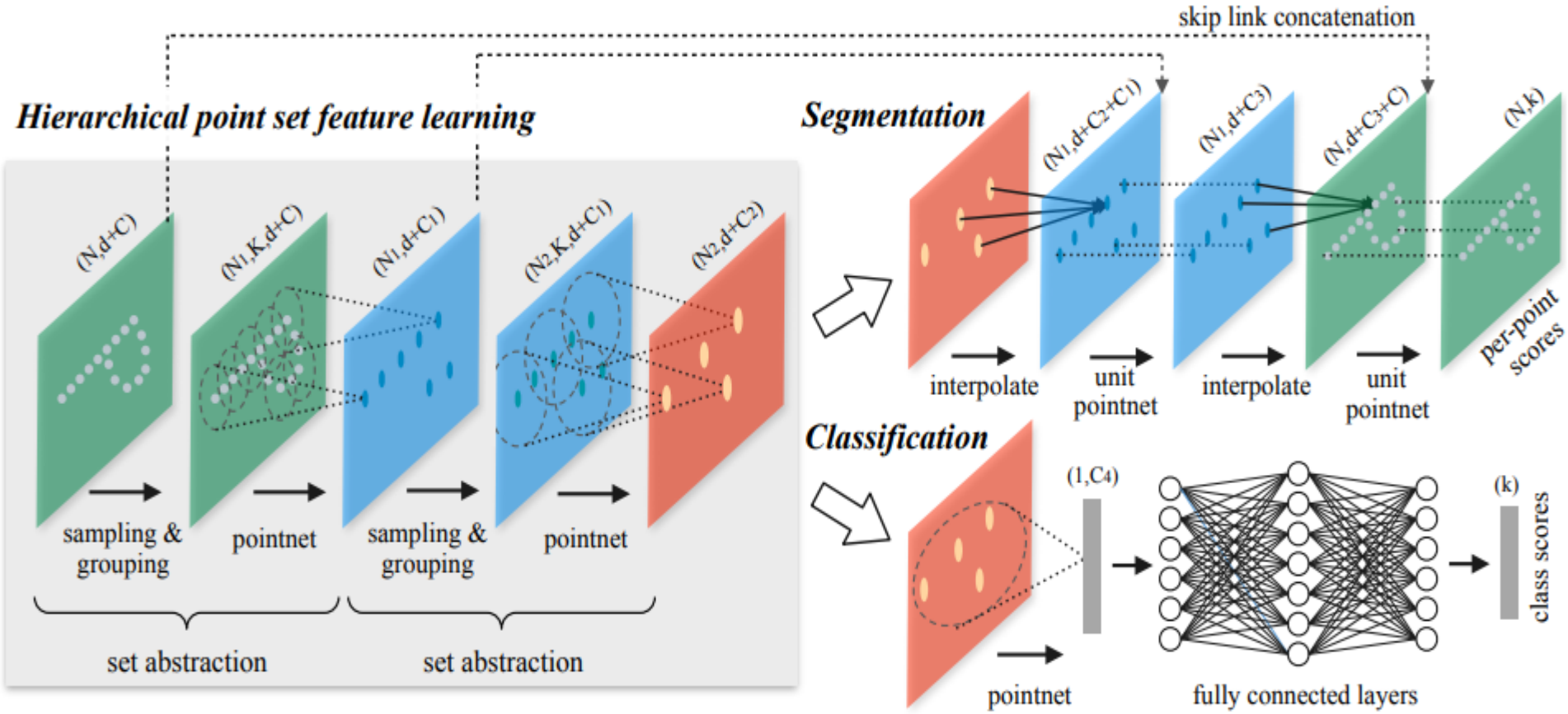
Advantages

- Takes native representation of point clouds
- No feature engineering
- Sparsity does not result a runtime issue
- No quantization
- Common building block for multiple tasks – classification, segmentation, detection

Issues

- Only produces a global descriptor – no gradual aggregation of local data

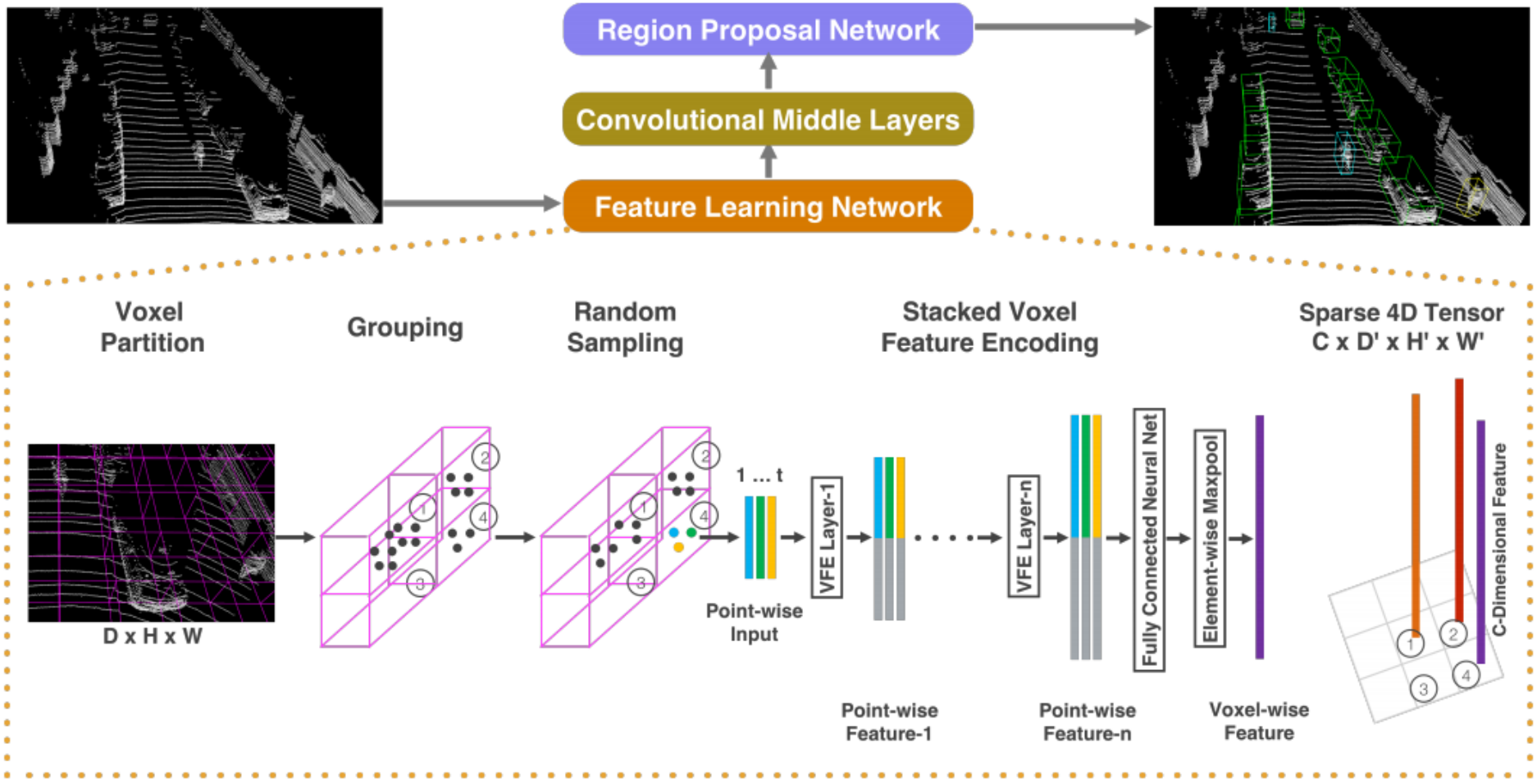
Pointnet++



Method	Input	Accuracy (%)
Subvolume [21]	vox	89.2
MVCNN [26]	img	90.1
PointNet (vanilla) [20]	pc	87.2
PointNet [20]	pc	89.2
Ours	pc	90.7
Ours (with normal)	pc	**91.9**

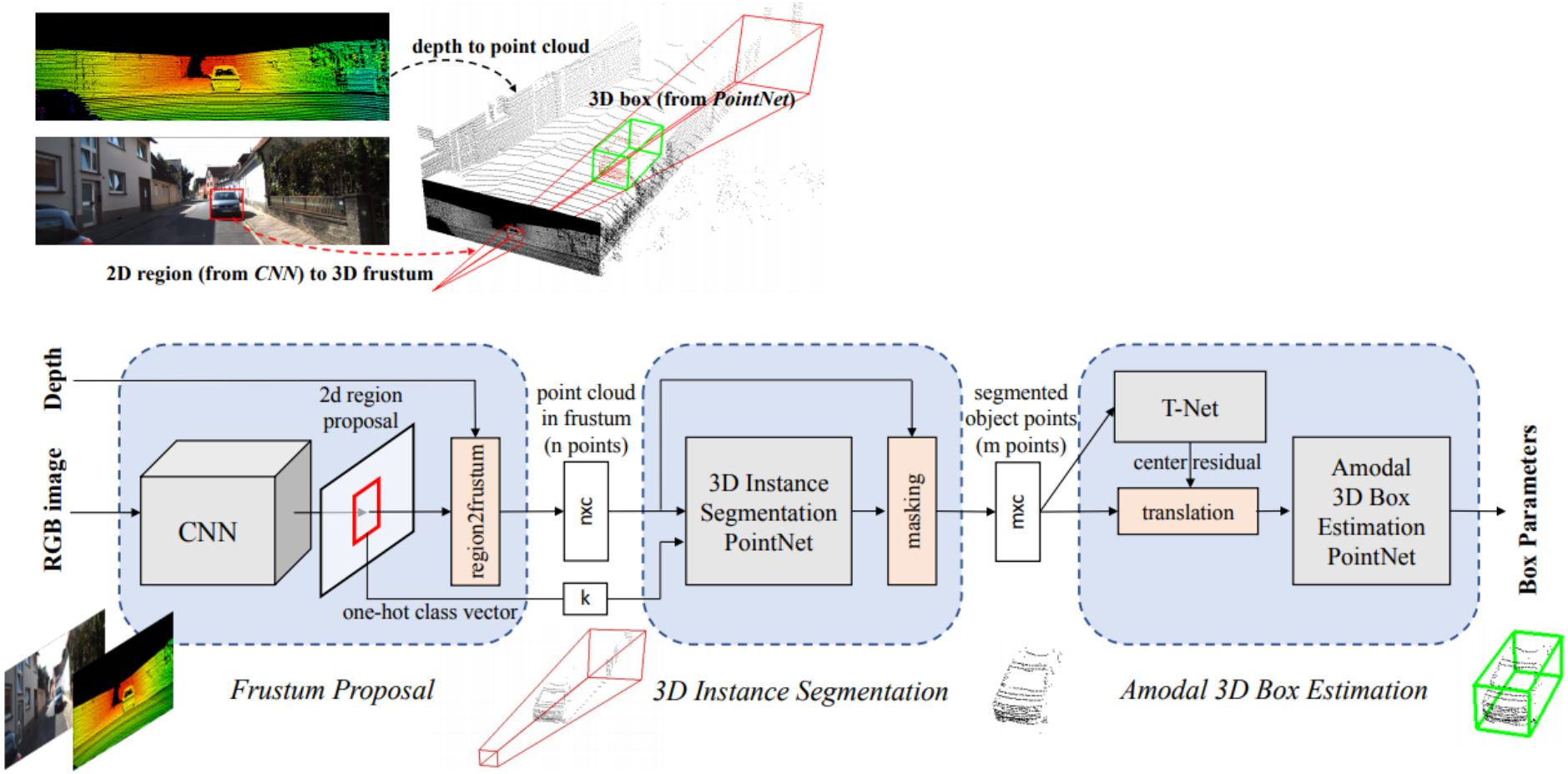
Source: C. R. Qi, L. Yi, H. Su, L. J. Guibas: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space

VoxelNet



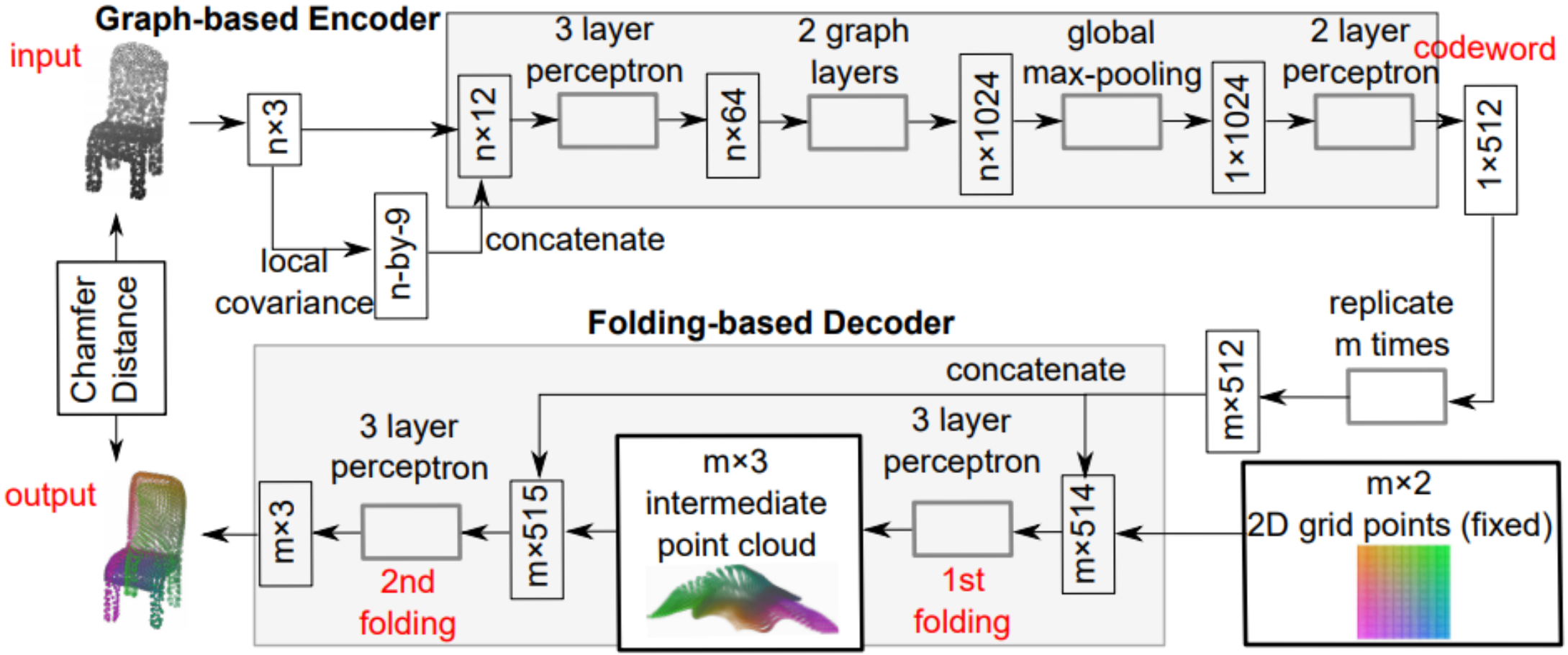
Source: Y. Zhou, O. Tuzel: VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection

Frustrum Pointnets



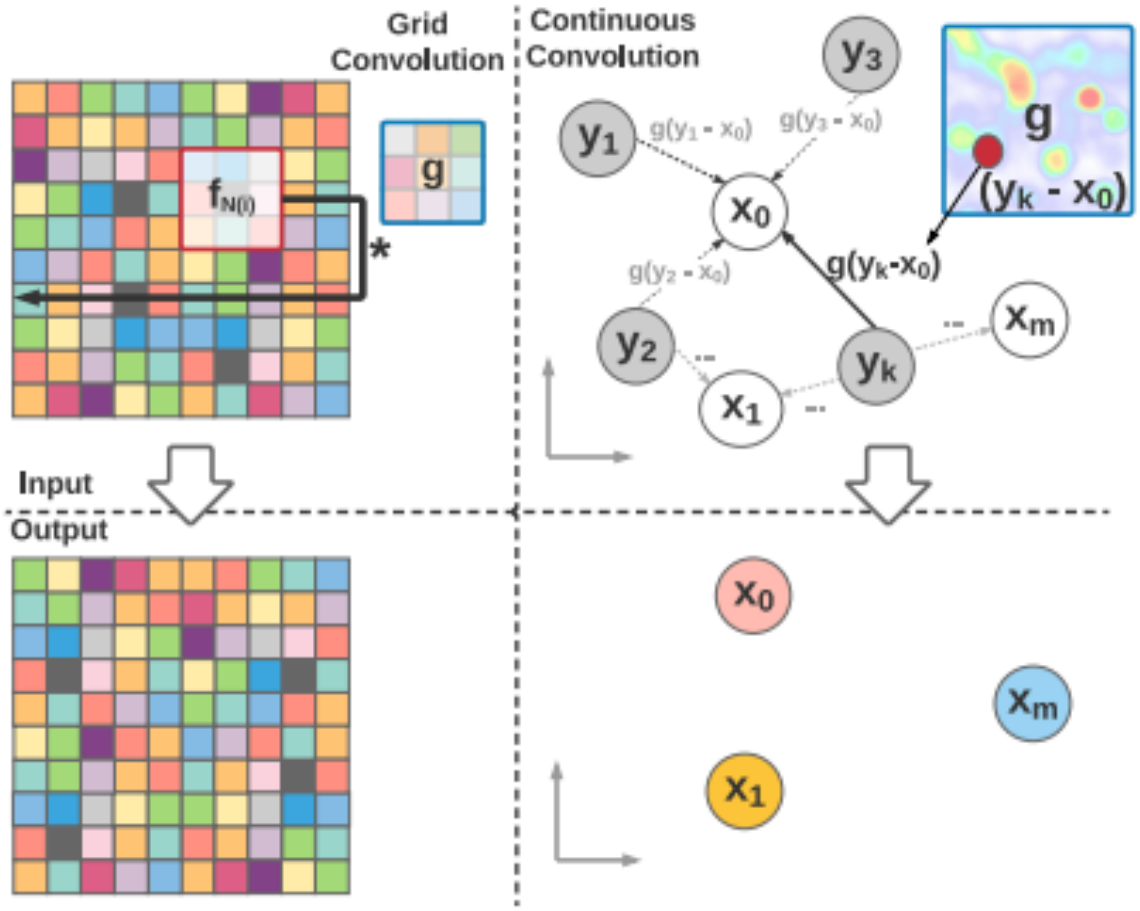
Source: C. R. Qi et al. Frustum PointNets for 3D Object Detection from RGB-D Data

FoldingNet



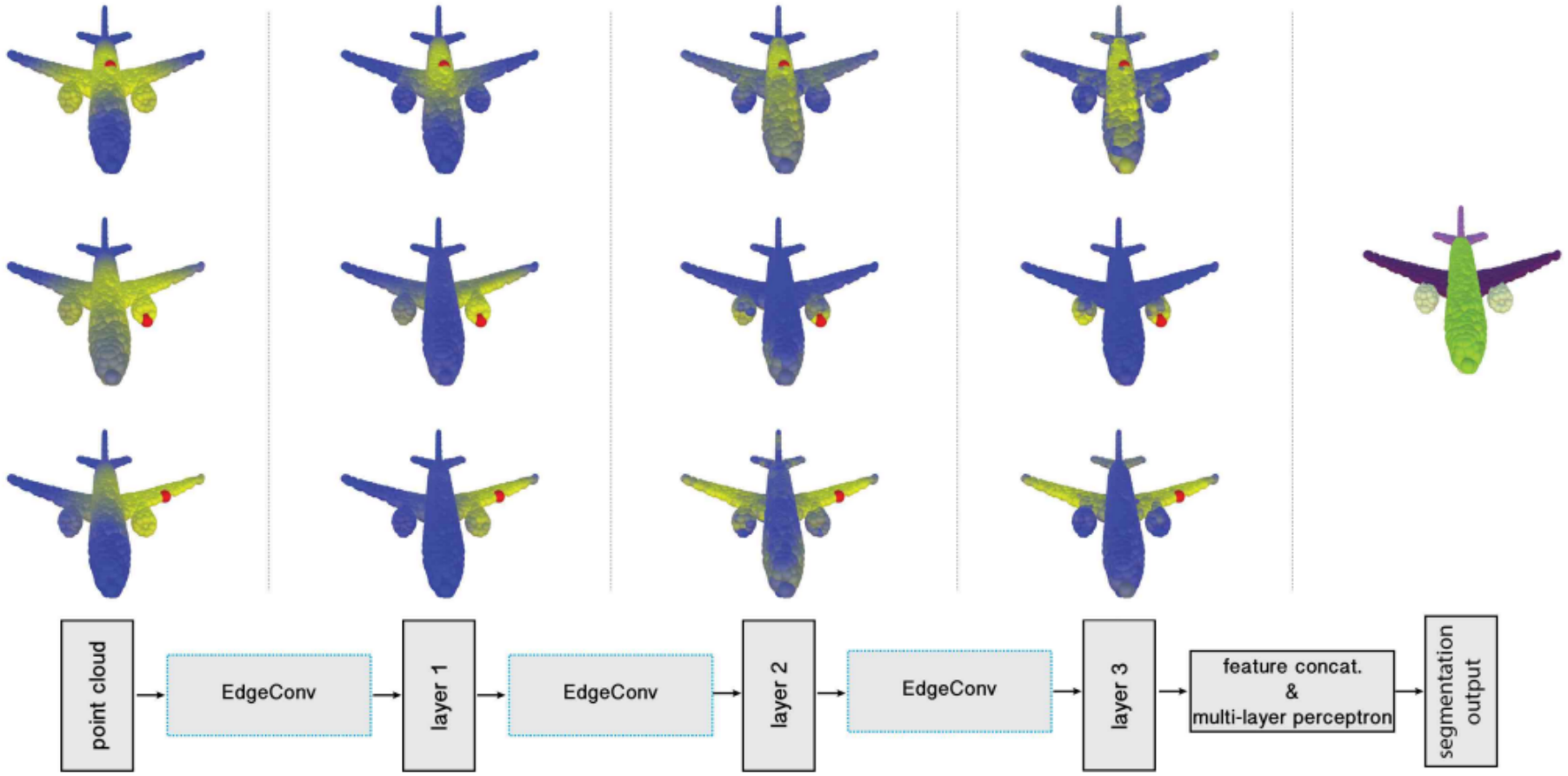
Source: Y. Yang, C. Feng, Y. Shen, D. Tian: FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation (CVPR 18')

Continuous convolutions



Source: S. Wang, S. Suo et al: Deep Parametric Continuous Convolutional Neural Networks

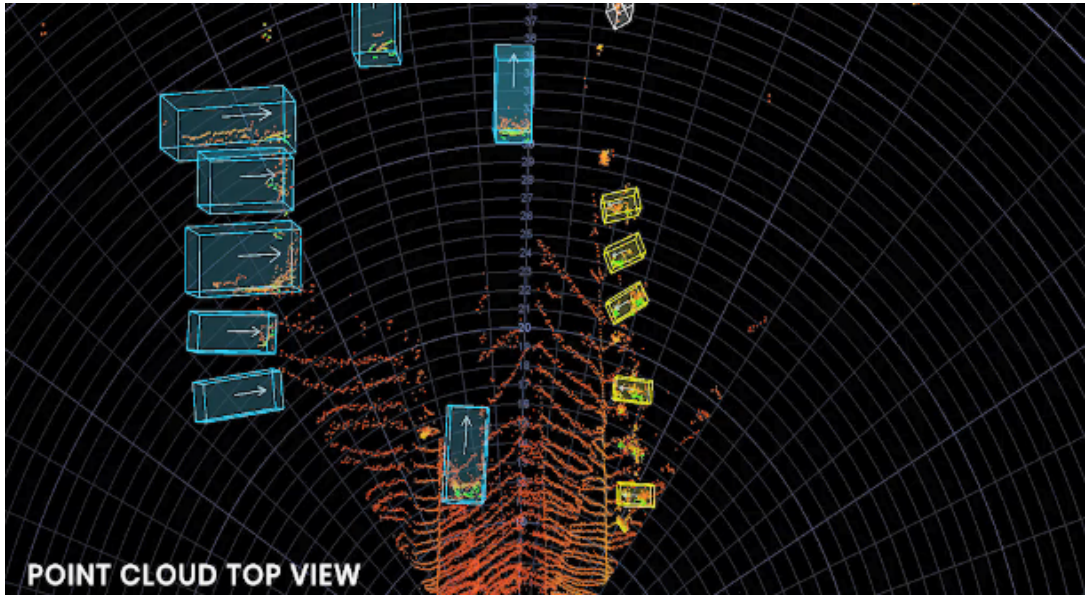
Dynamic graph CNN



Source: Y. Wang, Y. Sun et al: Dynamic Graph CNN for Learning on Point Clouds

Conclusion

- 3D data is here to stay
- Multiple competing approaches, with pros & cons
- Multiple representations of data, often within one network



THANK YOU.