3D Object Detection: The History, Present and Future

Charles Qi

2/2/2021

Agenda

- Background; The definition and applications of 3D object detection.
- The history and recent developments of 3D detection algorithms.
- Future directions of 3D detection research.
- Q&A, discussion.

Prior knowledge:
Although not required, it will be helpful to have some understanding of deep learning methods in 2D object detection. You can learn more about it here:
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

The big picture: A.I. applications from the virtual world to the physical world

1998 — Google was founded.
(Rise of the Internet)

Google

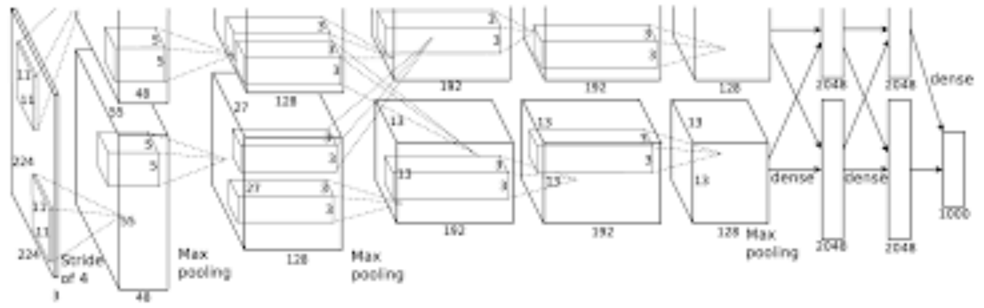
2004 — Facebook was founded.
(Rise of the Internet)

facebook

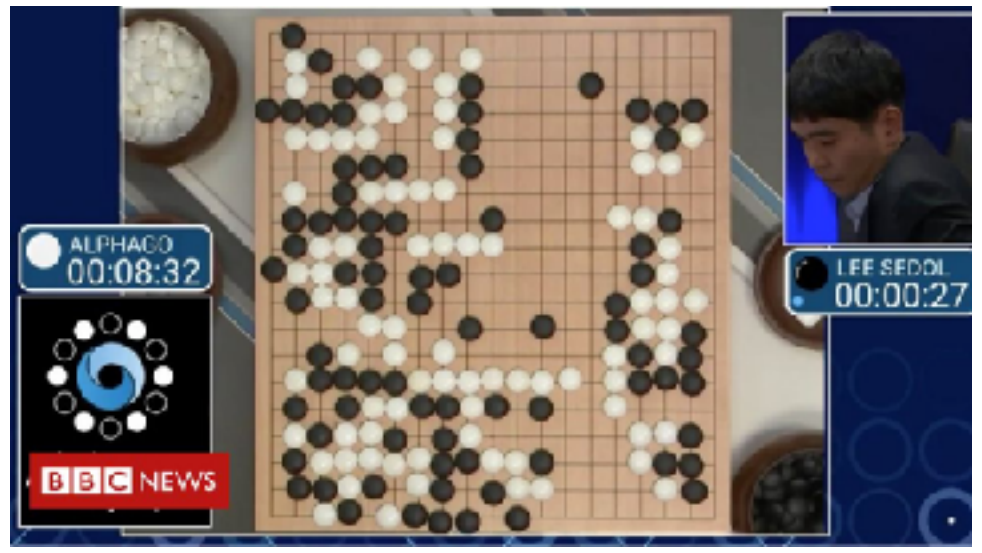
2007 — The release of the first iPhone
(Rise of the smart mobile devices)

2009 — The ImageNet was released.

2012 — The AlexNet breakthrough
(Rise of the deep learning for computer vision)



2016 — AlphaGo defeated world champion Lee Sedol
(A major milestone of A.I.)



The big picture: A.I. applications from the virtual world to the physical world

2005 2006 2009 2010 2013 2015 2016

The 2005 DARPA Grand Challenge.
(A milestone in autonomous driving)

Willow Garage

Willow Garage was founded.
(ROS, PCL, PR2 robots were invented;
Lab shut down in 2014)

Google's self-driving car
project started.

The Kinect was released.

The KITTI dataset was released.

The ShapeNet dataset was released.

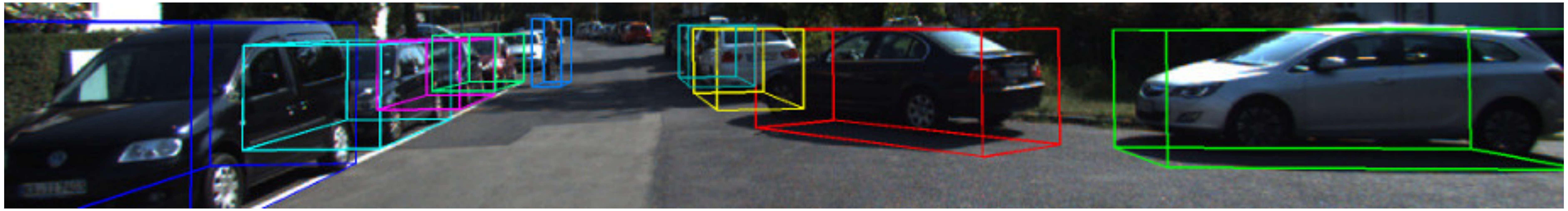
A series of deep neural networks
for 3D data were invented.
(Rise of 3D deep learning)

We are yet to reach the time where A.I. is widely applied to the "robots" in the physical world.
3D deep learning and 3D object detection are the core technologies towards that future!

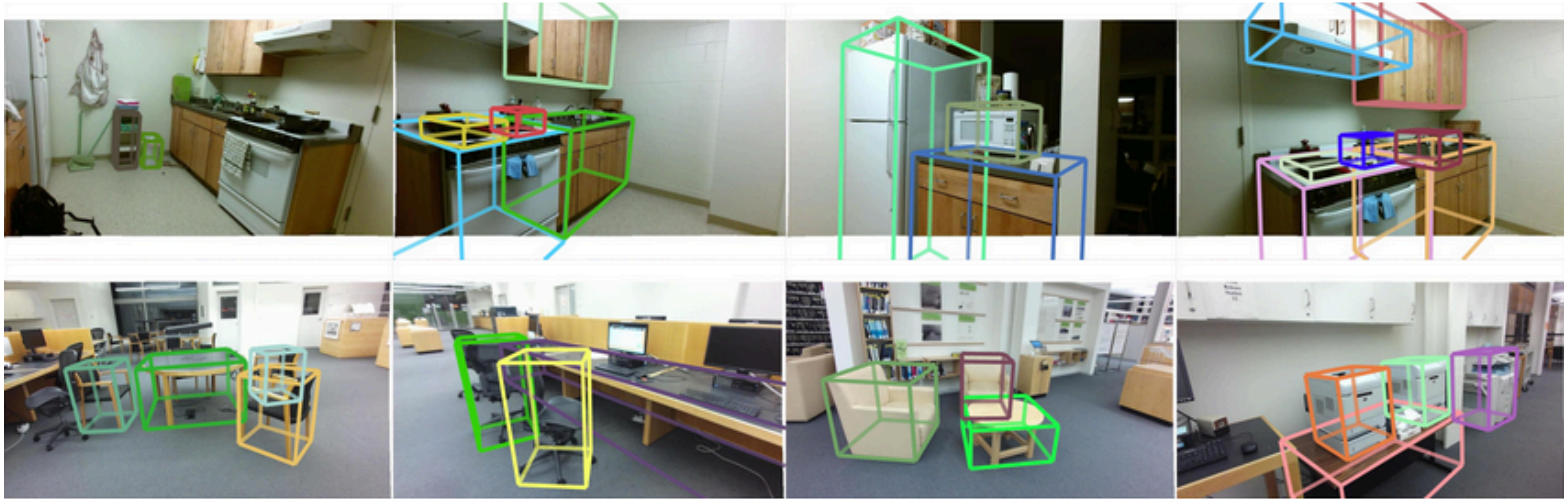
What is 3D object detection?

- **Input**: sensor data of a 3D scene (RGB/depth/radar images).
- **Output**: localization, shape and semantics of the 3D objects in the scene (amodal 3D bounding boxes).

KITTI:



SUN RGB-D:



What is 3D object detection?

- **Input**: sensor data of a 3D scene (RGB/depth/radar images).
- **Output**: localization, shape and semantics of the 3D objects in the scene (amodal 3D bounding boxes).

KITTI:

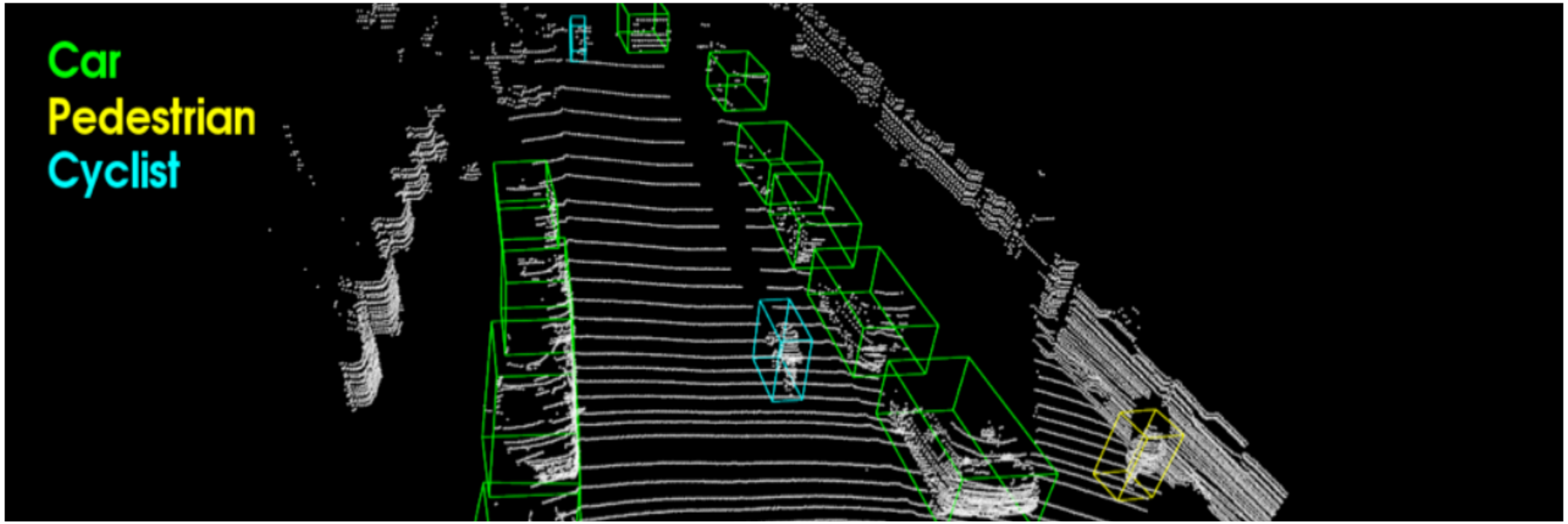


Figure from VoxelNet [Zhou et al. 2018]

What is 3D object detection?

- **Input**: sensor data of a 3D scene. (RGB/depth/radar images).
  - E.g. point clouds (N, 3+C) where the channels are x,y,z and features like color, intensity etc.
- **Output**: localization, shape and semantics of the 3D objects in the scene.
  - E.g. upright 3D amodal, oriented bounding boxes (K, 7) with center xyz, length, width, height, heading; semantic classes (K,) and box scores (K,).

KITTI:

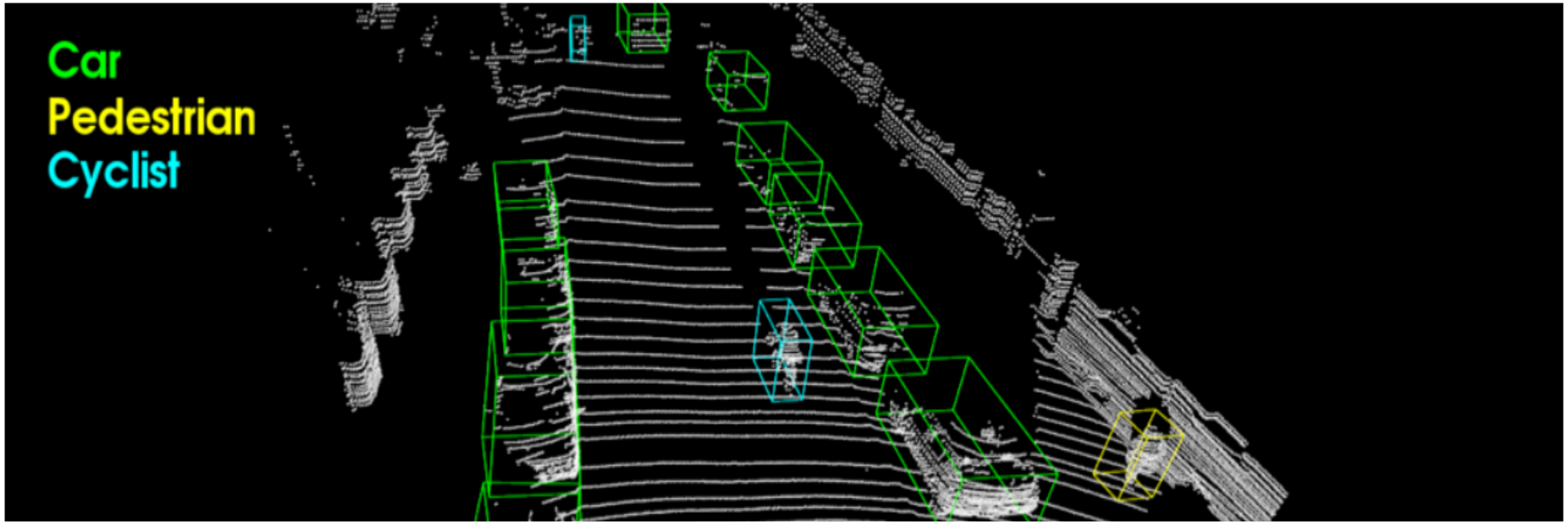


Figure from VoxelNet [Zhou et al. 2018]

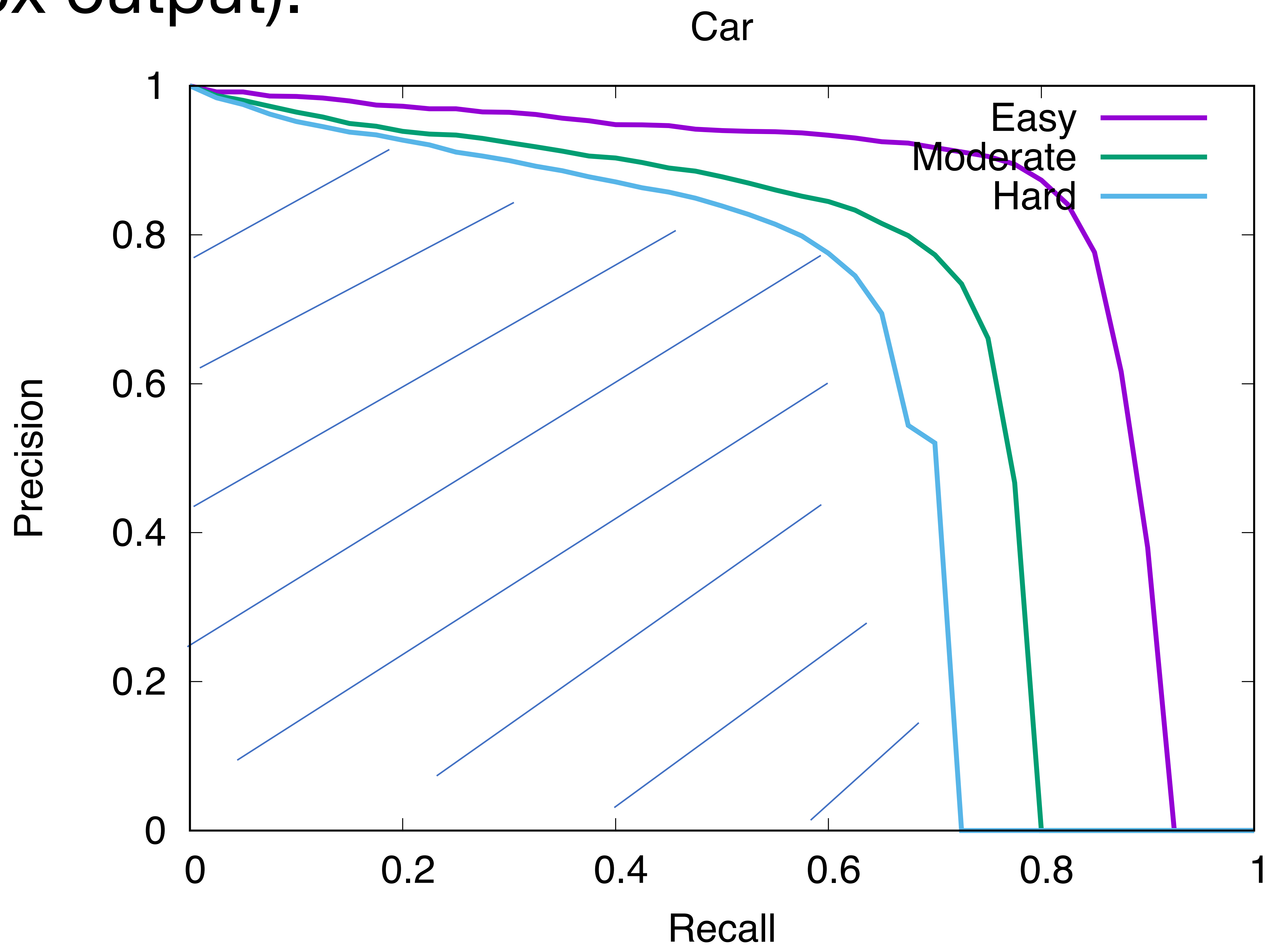
What is 3D object detection?

Evaluation Metric: Average Precision (AP) with a 3D Intersection over Union (IoU) threshold (assuming 3D bounding box output).

For each score threshold, we get an “operation point” on the PR curve — all predictions with scores higher than the threshold are considered “positive” detections while all others with scores lower than the threshold is considered “negative” detections.

By scanning through the score thresholds e.g. from 0 to 1, we get the PR curve.

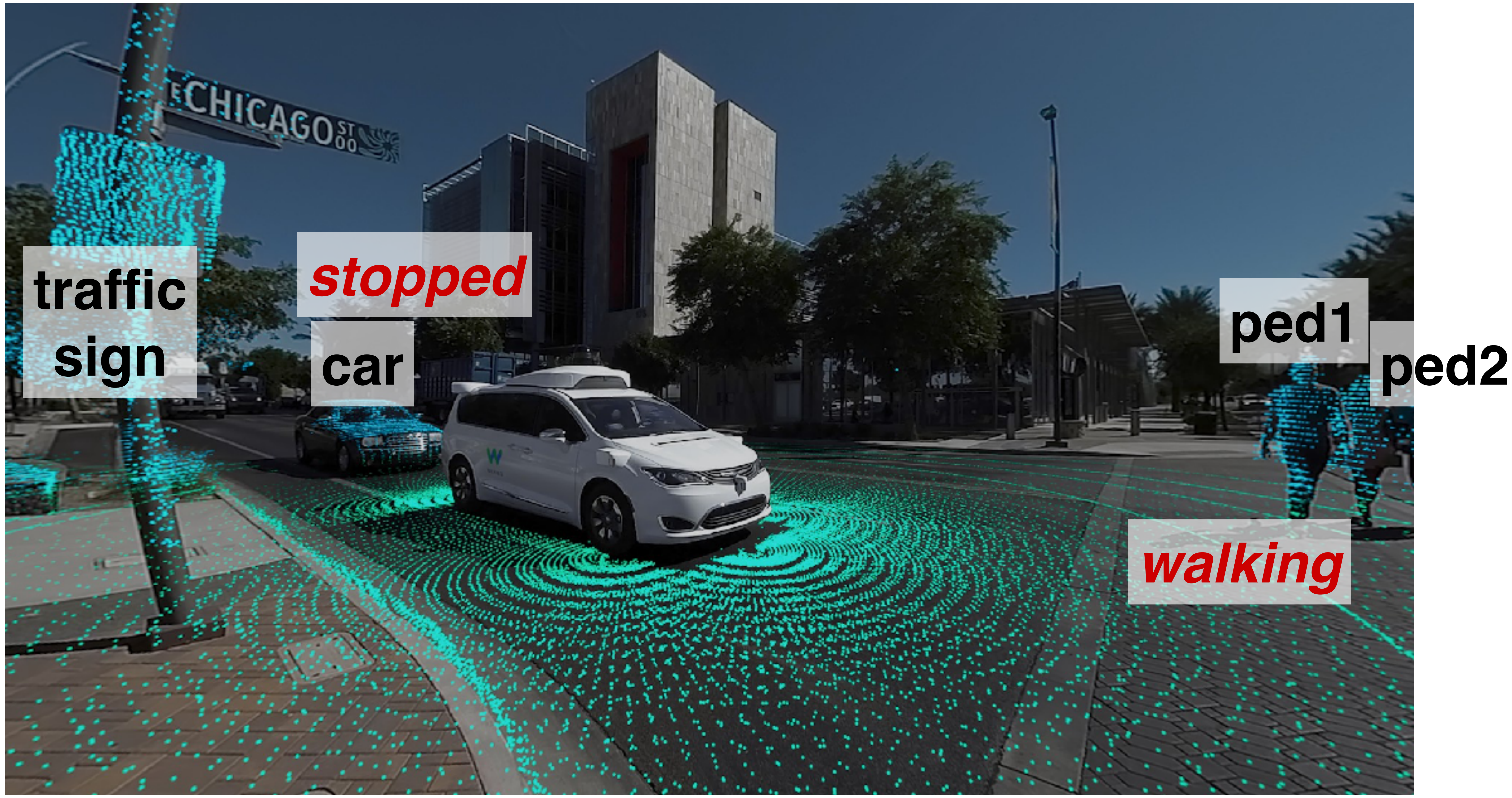
Average precision is the area under the curve.



A detailed explanation of the Average Precision metric:
https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173

Applications of 3D object detection

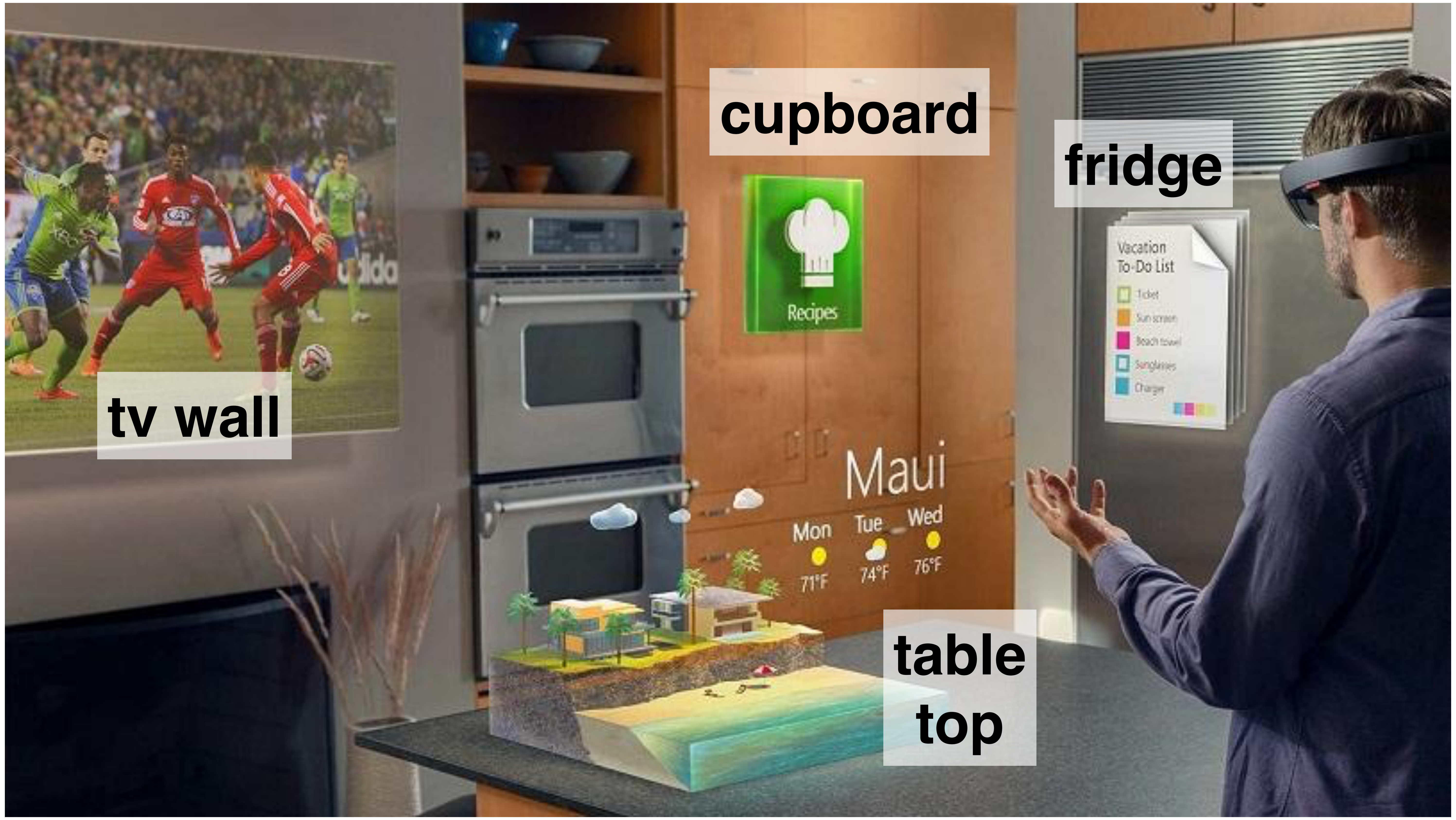
Autonomous Driving



source: Waymo

Applications of 3D object detection

Augmented Reality



source: Microsoft HoloLens

Applications of 3D object detection

Robot Grasping

source: NVIDIA

The history of 3D object detection

Pre deep learning:

Template-based:
Generalized Hough Voting (2010) [1]
Local/global descriptor+matching+ICP (2012) [2]

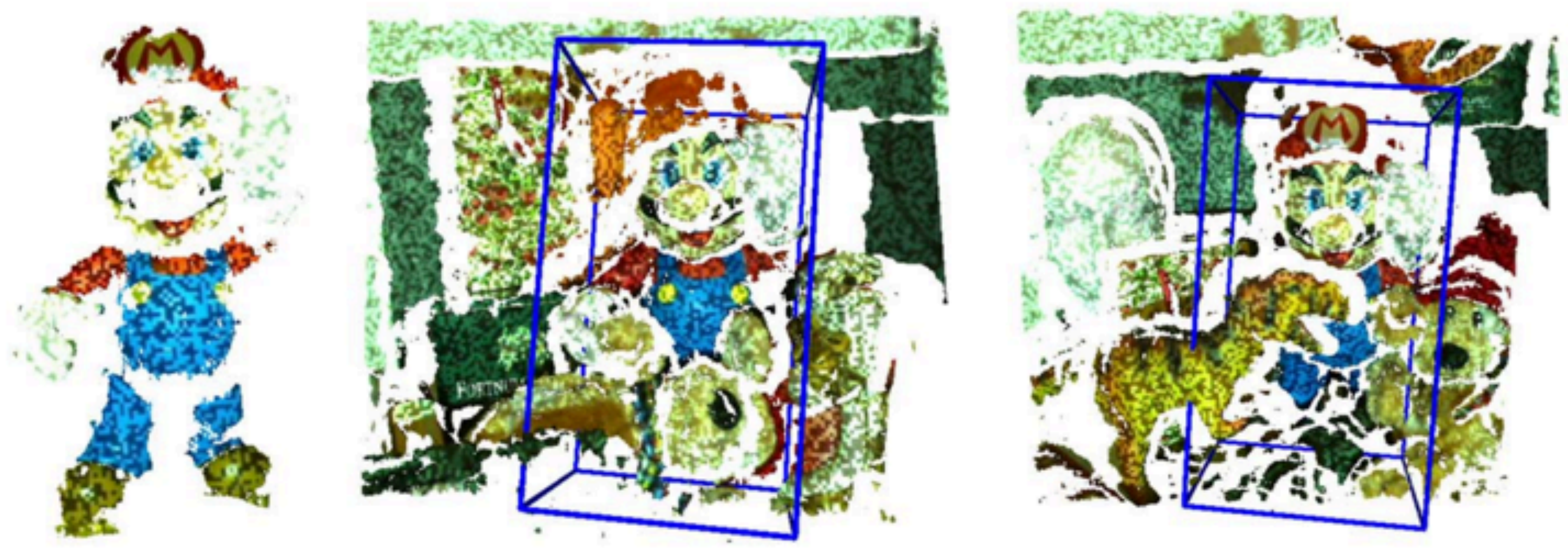


Figure from [1]

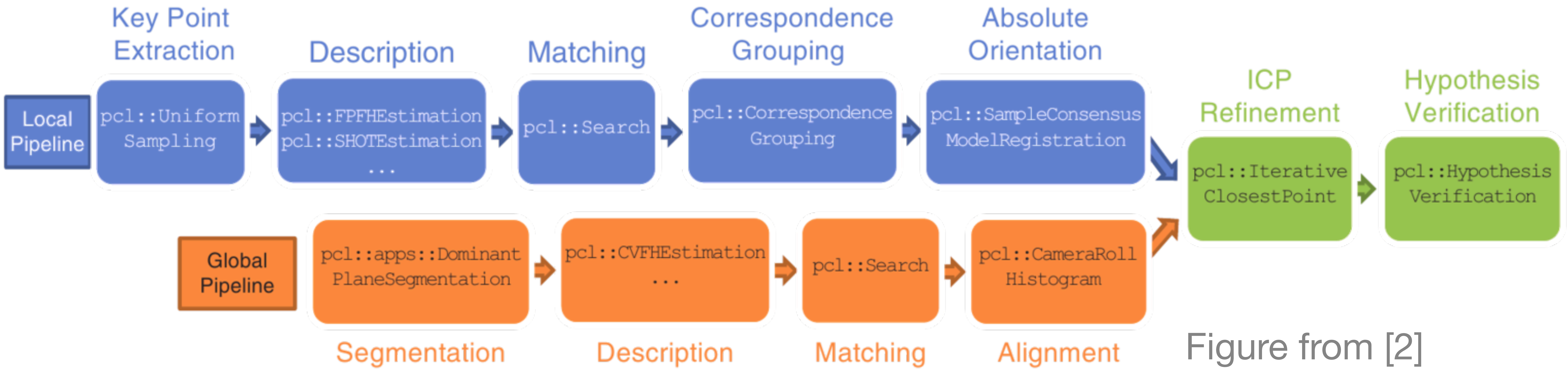


Figure from [2]

The history of 3D object detection

Pre deep learning:

Clustering-based:
Object Discovery in 3D scenes via Shape Analysis (2013) [3]
Data-driven “objectness” score prediction.

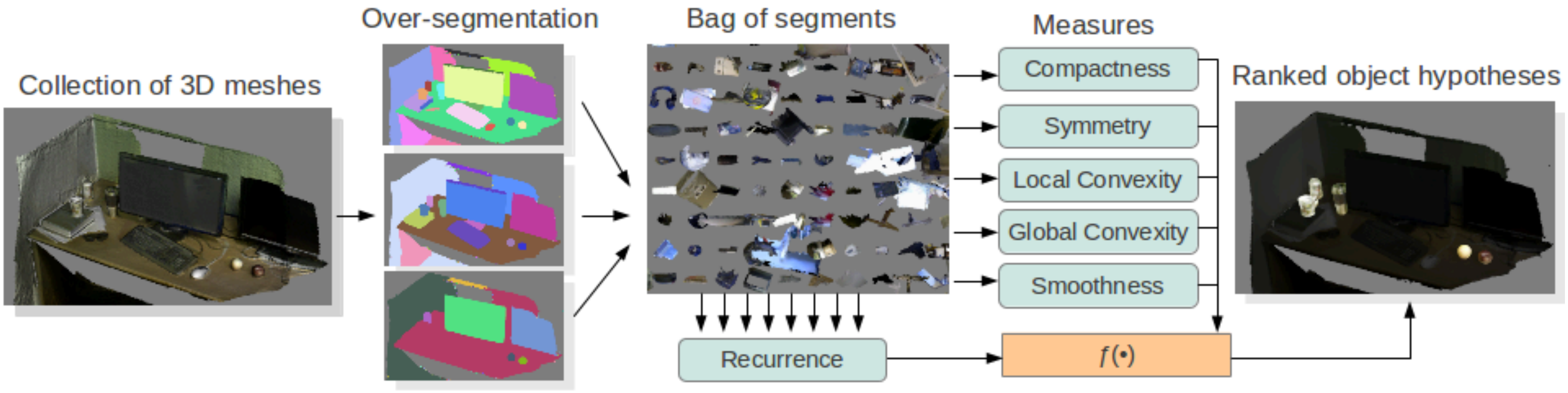
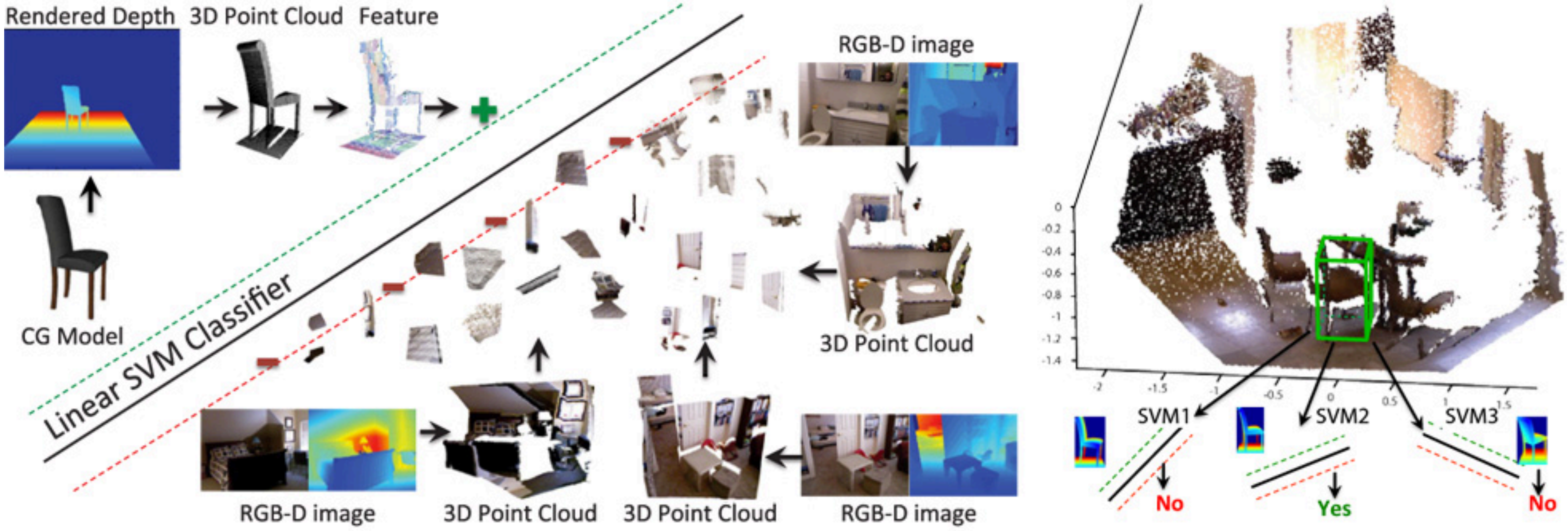


Figure from [3]

The history of 3D object detection

Pre deep learning:

Sliding window based:
Sliding Shapes for 3D Object Detection in Depth Images (2014) [4]



(a) Training each 3D exemplar detector independently. (b) Testing with exemplars.

Figure from [4]

The deep learning era of 3d object detection

Three factors prepared us for this phase:

- **The rise of 3D sensors and access to large-scale 3D datasets**
  - Commercial depth cameras, Lidars.
  - KITTI, SUN RGB-D, ShapeNet, ScanNet…
- **The progresses of 2D object detectors**
  - R-CNN (deep nets for classification) -> Fast R-CNN (parallel processing) -> Faster R-CNN (region proposal network)…
- **The rise of 3D deep learning**
  - A series of novel deep neural network architectures for 3D data (MVCNN, 3DCNNs, PointNet, PointNet++ etc.) has been invented.

The deep learning era of 3d object detection

A first serious attempt of using deep nets for 3d detection:

Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images (2016) [5]
- 3D CNNs for Faster-RCNN style region proposal.

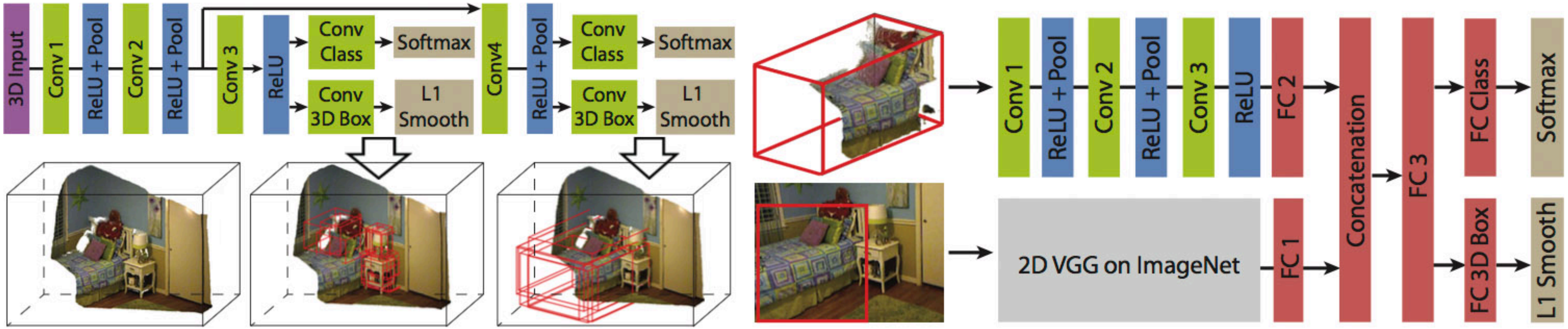
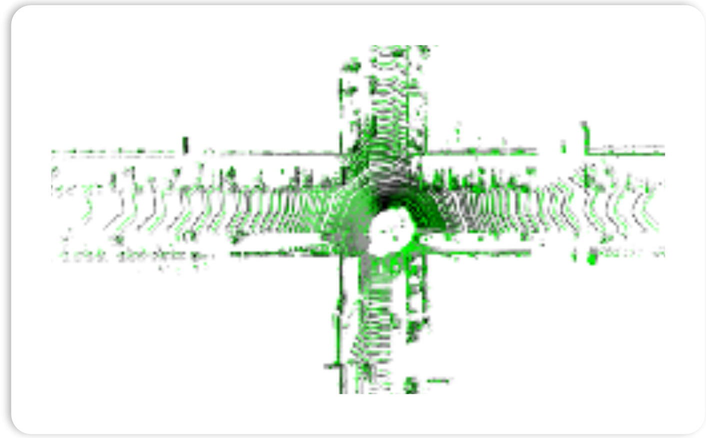
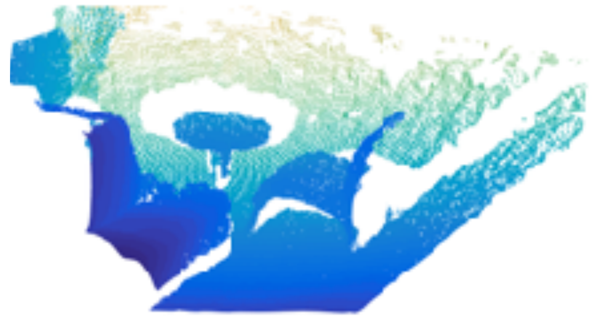


Figure from [5]

Con: 3D CNNs are very costly in both memory and time.

The deep learning era of 3d object detection

Solutions

Image-driven	Dimension reduction	Leveraging Sparsity in 3D
Monocular view detectors Frustum-based detectors	Bird’s eye view detectors	Point set deep nets Sparse 3D conv, GNNs
		

E.g.: **Frustum PointNets [6]** **PointPillars [7]** **VoteNet [8]**

Image-driven 3D object detection

- Key idea: Leverage mature 2D object detectors to propose objects from RGB images.

Monocular or stereo view based

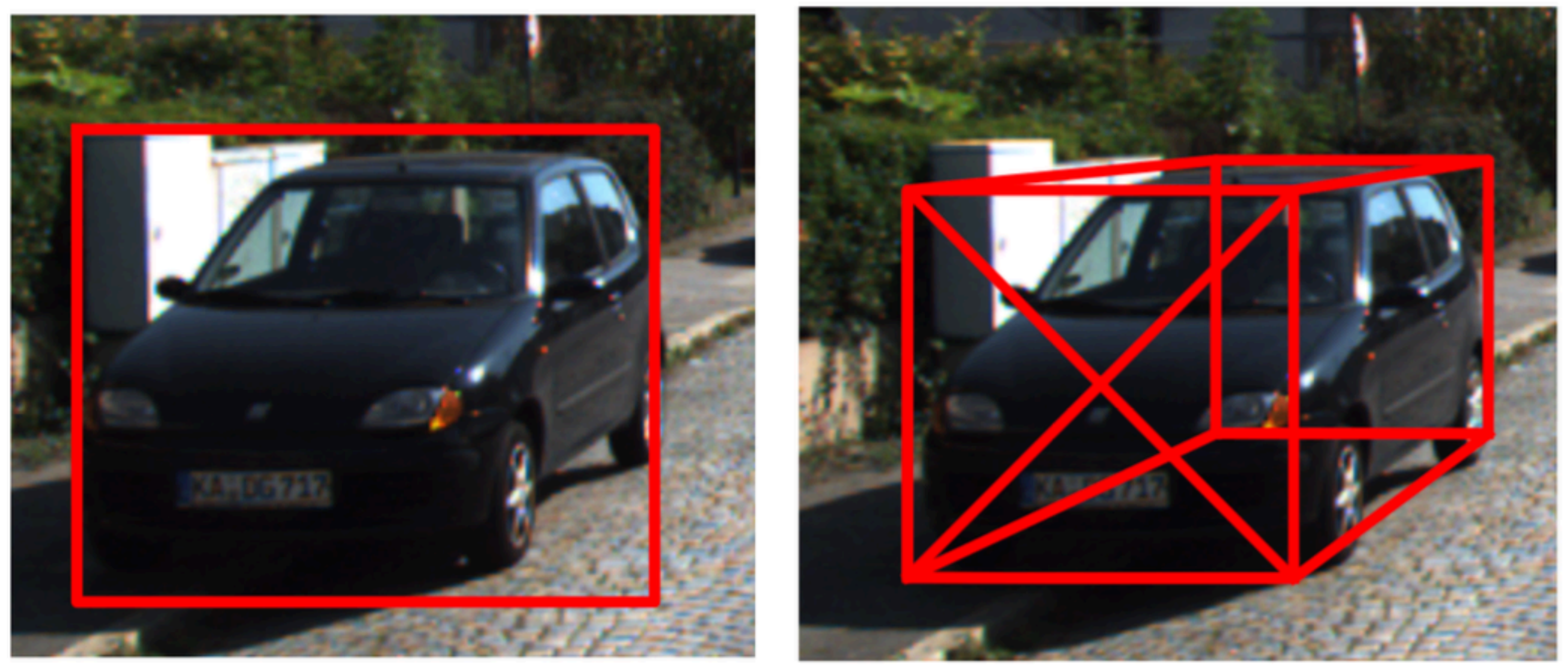
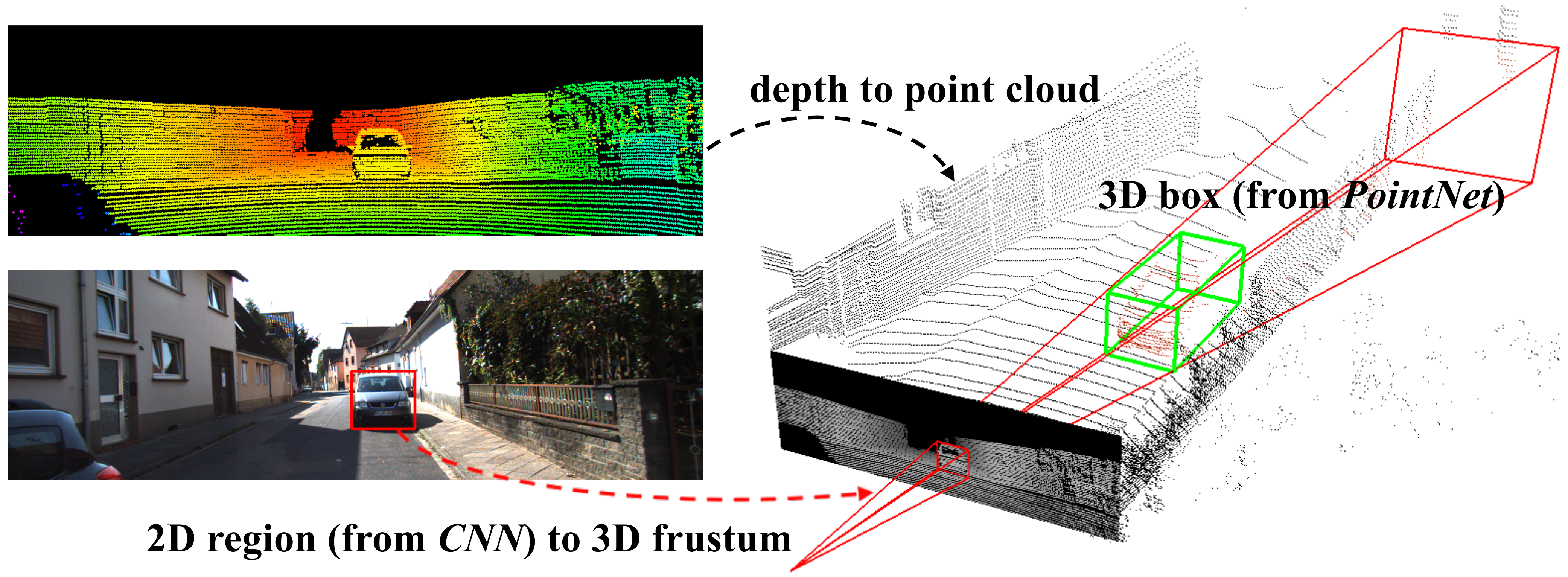


Figure from [9]

3d bounding box estimation using deep learning and geometry (2017) [9]
Pseudo-lidar (2019) [10]
Objects as Points (2019) [11]

RGB-D data based



Frustum PointNets [6]

Frustum PointNets for 3D Object Detection from RGB-D Data

Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, Leonidas Guibas.

CVPR 2018

arXiv:1711.08488v2 [cs.CV] 13 Apr 2018

Frustum PointNets for 3D Object Detection from RGB-D Data

Charles R. Qi[1*] Wei Liu[2] Chenxia Wu[2] Hao Su[3] Leonidas J. Guibas[1]
[1]Stanford University [2]Nuro, Inc. [3]UC San Diego

Abstract

In this work, we study 3D object detection from RGB-D data in both indoor and outdoor scenes. While previous methods focus on images or 3D voxels, often obscuring natural 3D patterns and invariances of 3D data, we directly operate on raw point clouds by popping up RGB-D scans. However, a key challenge of this approach is how to efficiently localize objects in point clouds of large-scale scenes (region proposal). Instead of solely relying on 3D proposals, our method leverages both mature 2D object detectors and advanced 3D deep learning for object localization, achieving efficiency as well as high recall for even small objects. Benefited from learning directly in raw point clouds, our method is also able to precisely estimate 3D bounding boxes even under strong occlusion or with very sparse points. Evaluated on KITTI and SUN RGB-D 3D detection benchmarks, our method outperforms the state of the art by remarkable margins while having real-time capability.

1. Introduction

Recently, great progress has been made on 2D image understanding tasks, such as object detection [13] and instance segmentation [14]. However, beyond getting 2D bounding boxes or pixel masks, *3D understanding* is eagerly in demand in many applications such as autonomous driving and augmented reality (AR). With the popularity of 3D sensors deployed on mobile devices and autonomous vehicles, more and more 3D data is captured and processed. In this work, we study one of the most important 3D perception tasks – 3D object detection, which classifies the object category and estimates *oriented 3D bounding boxes* of physical objects from 3D sensor data.

While 3D sensor data is often in the form of point clouds, how to represent point cloud and what deep net architectures to use for 3D object detection remains an open problem. Most existing works convert 3D point clouds to images by projection [36, 26] or to volumetric grids by quantization [40, 23, 26] and then apply convolutional networks.

*Majority of the work done as an intern at Nuro, Inc.

Figure 1. **3D object detection pipeline.** Given RGB-D data, we first generate 2D object region proposals in the RGB image using a CNN. Each 2D region is then extruded to a *3D viewing frustum* in which we get a point cloud from depth data. Finally, our frustum PointNet predicts a (oriented and amodal) 3D bounding box for the object from the points in frustum.

This data representation transformation, however, may obscure natural 3D patterns and invariances of the data. Recently, a number of papers have proposed to process point clouds directly without converting them to other formats. For example, [25, 27] proposed new types of deep net architectures, called *PointNets*, which have shown superior performance and efficiency in several 3D understanding tasks such as object classification and semantic segmentation.

While PointNets are capable of classifying a whole point cloud or predicting a semantic class for each point in a point cloud, it is unclear how this architecture can be used for instance-level 3D object detection. Towards this goal, we have to address one key challenge: how to efficiently propose possible locations of 3D objects in a 3D space. Imitating the practice in image detection, it is straightforward to enumerate candidate 3D boxes by sliding windows [8] or by 3D region proposal networks such as [33]. However, the computational complexity of 3D search typically grows cubically with respect to resolution and becomes too expensive for large scenes or real-time applications such as autonomous driving.

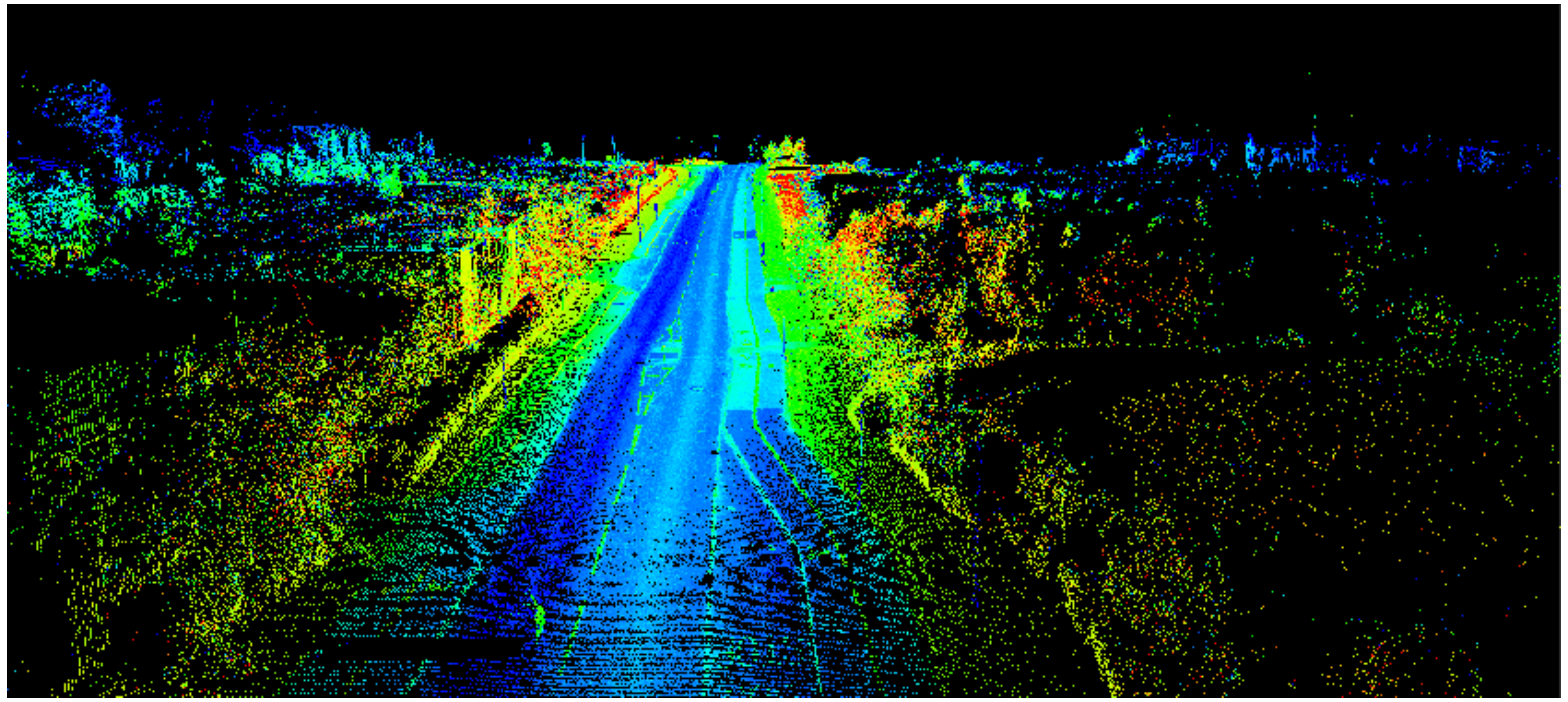
Instead, in this work, we reduce the search space following the dimension reduction principle: we take the advantage of mature 2D object detectors (Fig. 1). First, we extract the 3D bounding frustum of an object by extruding 2D bounding boxes from image detectors. Then, within the 3D space trimmed by each of the 3D frustums, we consecutively perform 3D object instance segmentation and *amodal*

1

Images and Point Clouds

RGB images

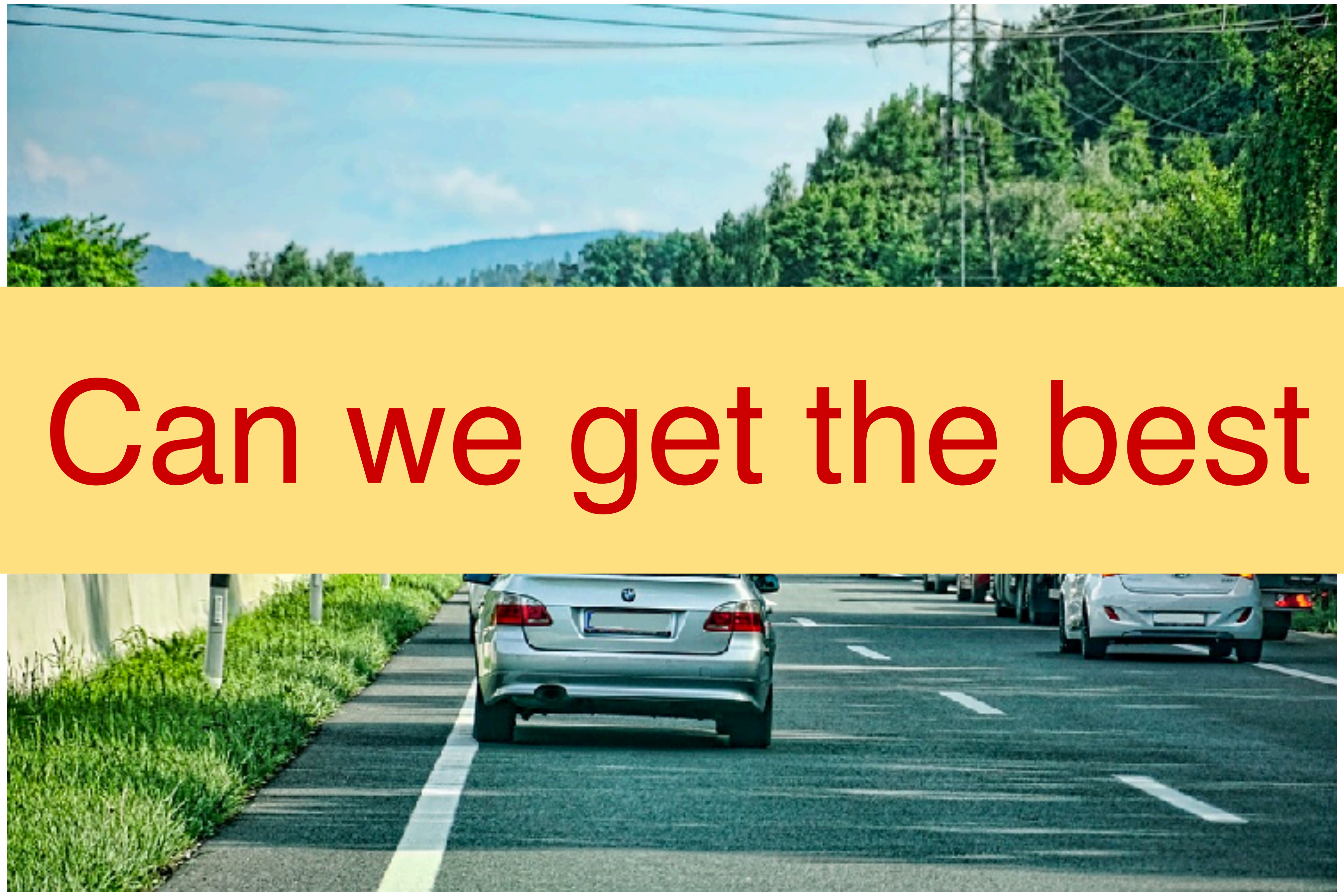
High resolution
Rich textures

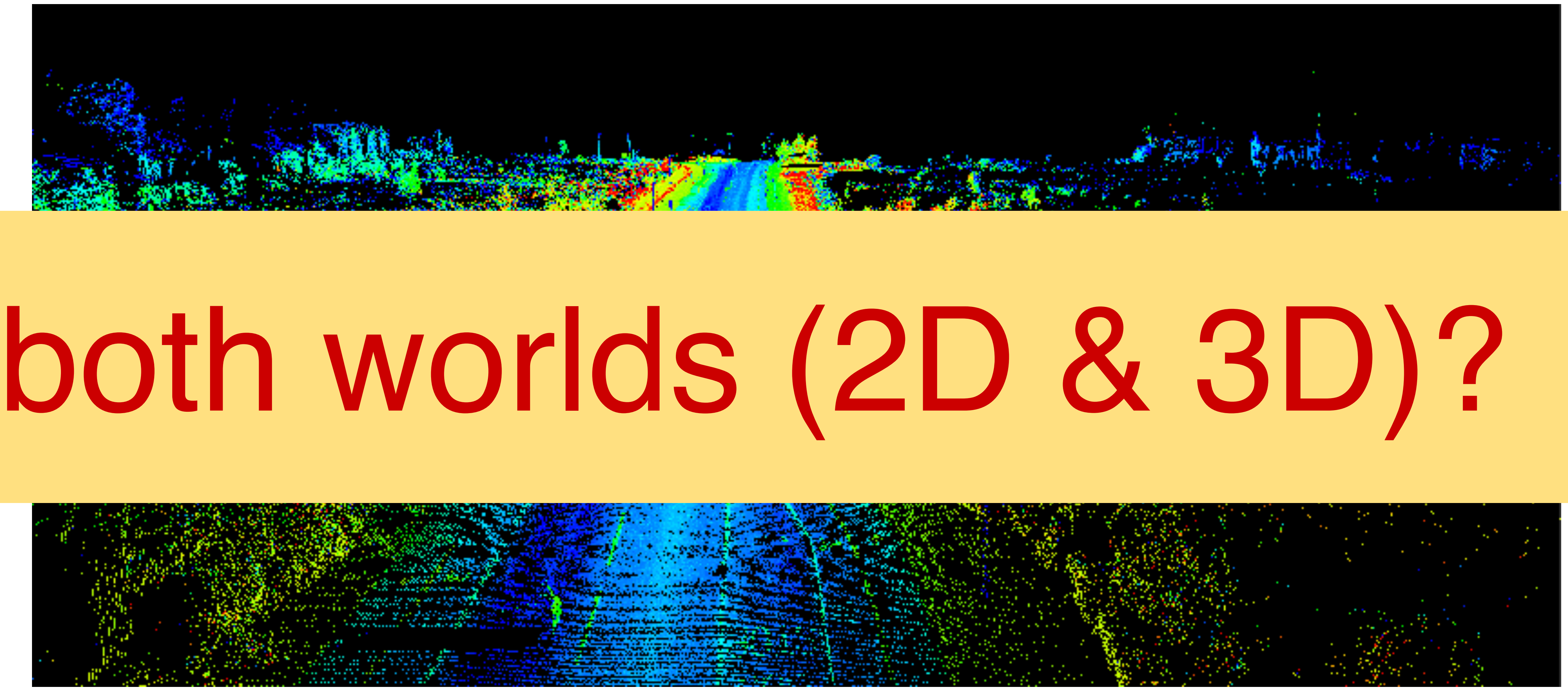


Lidar point clouds

Accurate depth
Accurate 3D geometry

Images and Point Clouds





Can we get the best of both worlds (2D & 3D)?

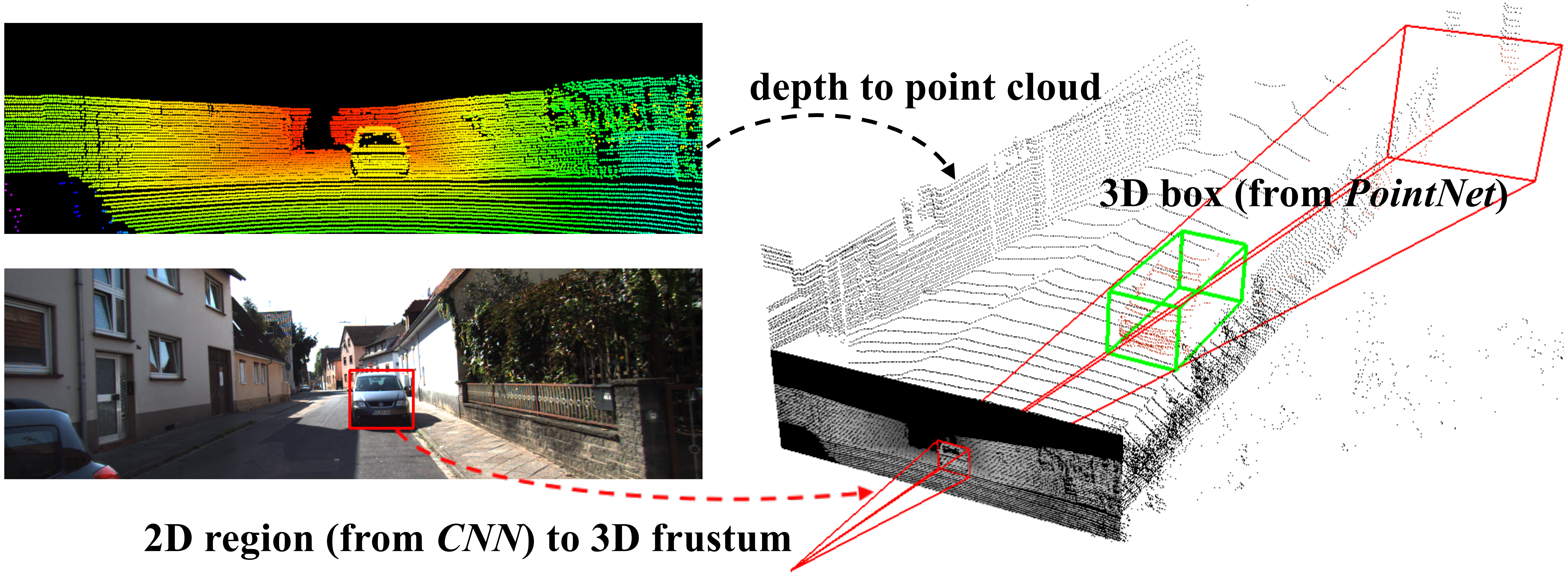
RGB images

High resolution
Rich textures

Lidar point clouds

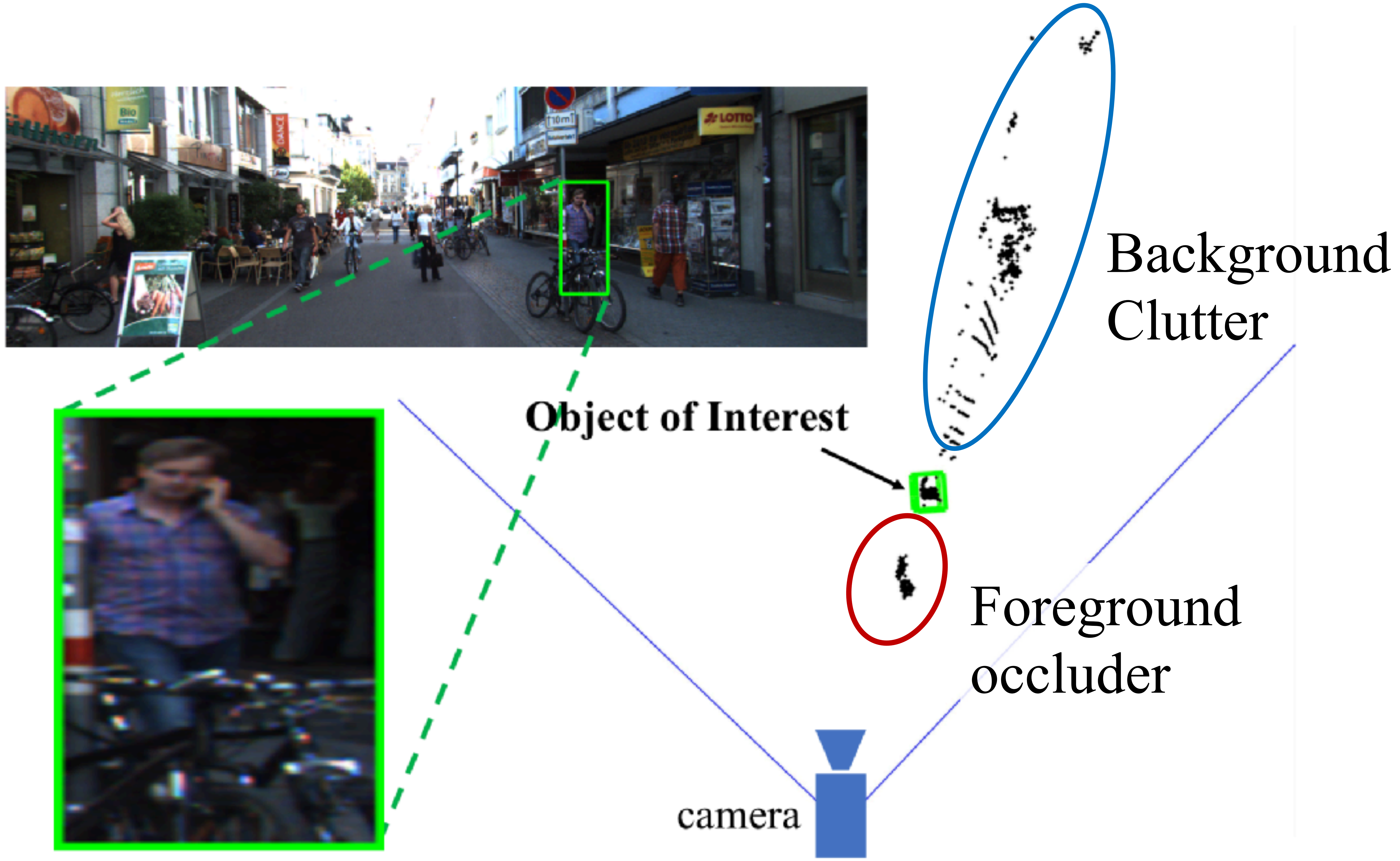
Accurate depth
Accurate 3D geometry

Frustum PointNets for 3D Object Detection



+ ***Leveraging mature 2D detectors*** *for region proposal. greatly reducing 3D search space.*
+ ***3D deep learning*** *for accurate object localization in frustum point clouds.*

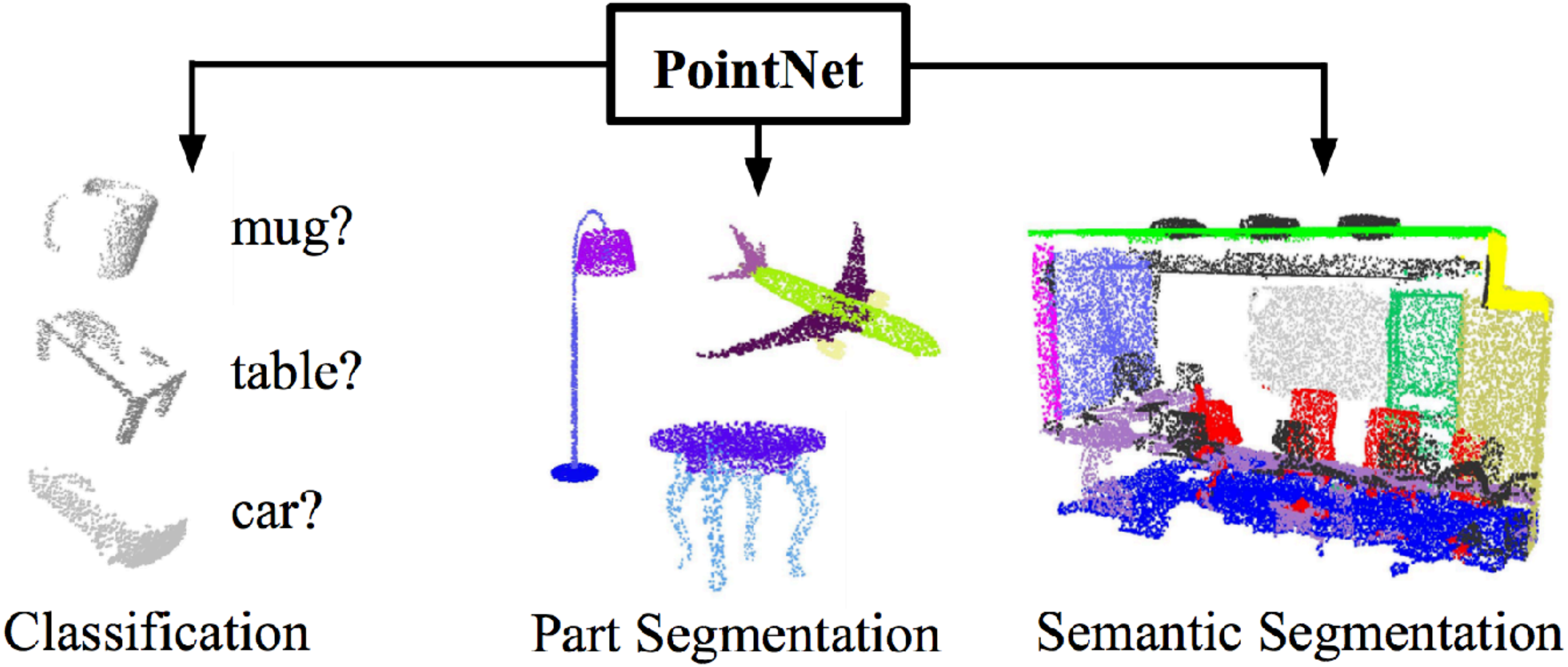
Frustum-based 3D Object Detection: Challenges



- Occlusions and clutters are common in frustum point clouds
- Large range of point depths

Frustum PointNets

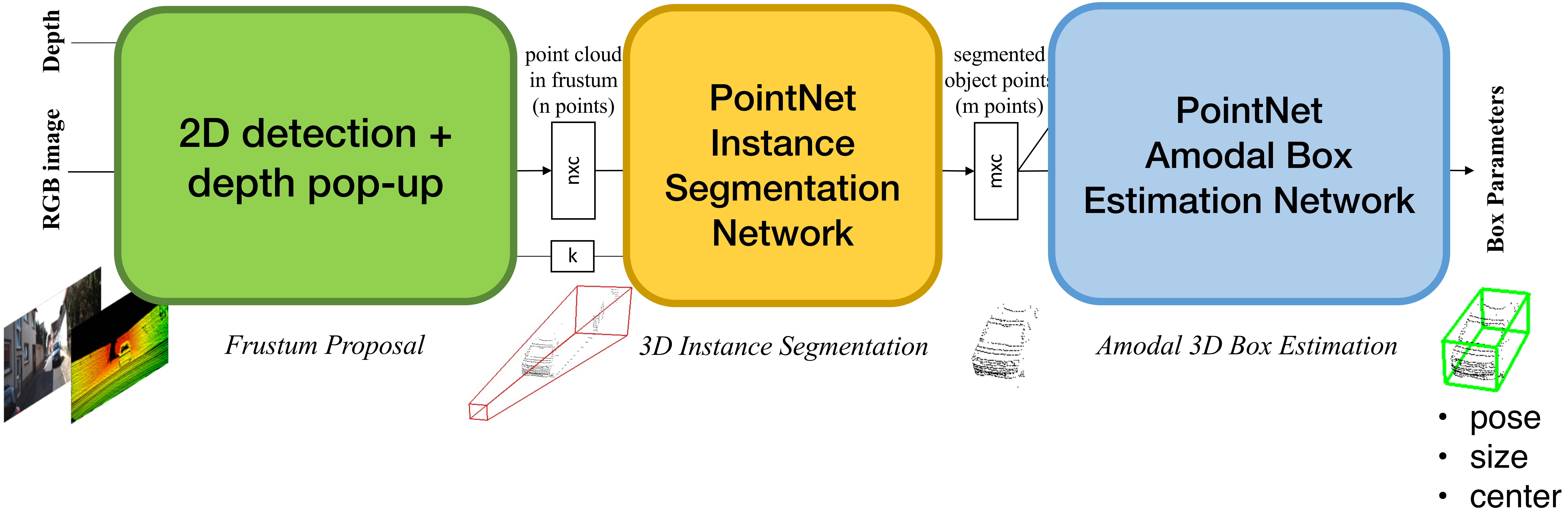
Use **PointNets** for **data-driven** object detection in frustums.



PointNet [Qi et al. CVPR 2017] *PointNet++ [Qi et al. NIPS 2017]*

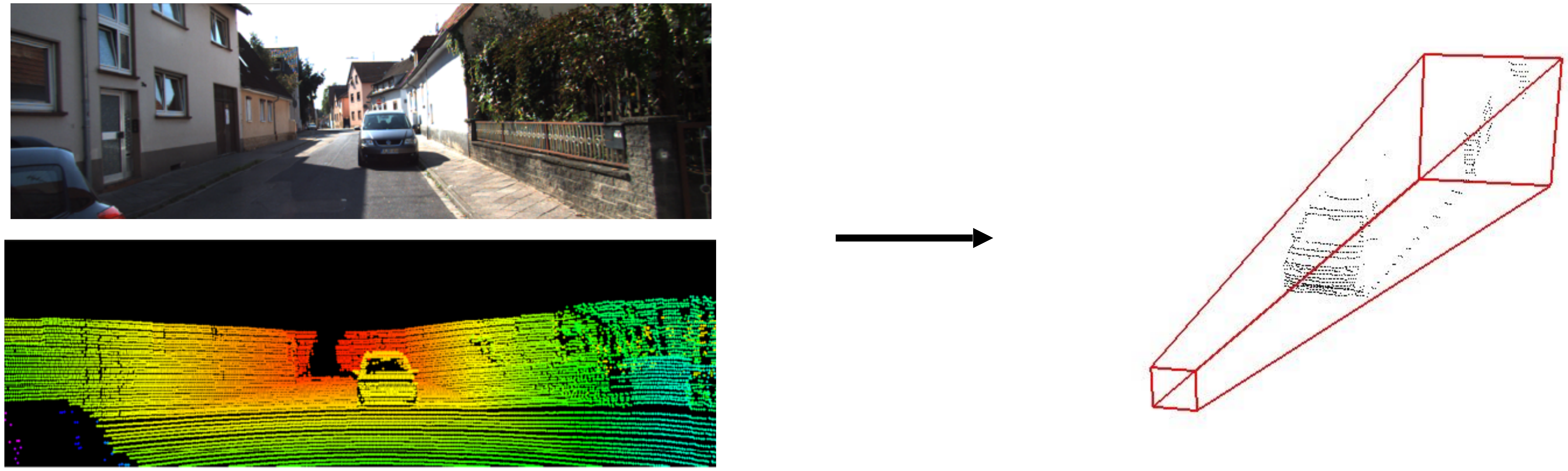
Frustum PointNets

Use **PointNets** for **data-driven** object detection in frustums.



Frustum Proposal

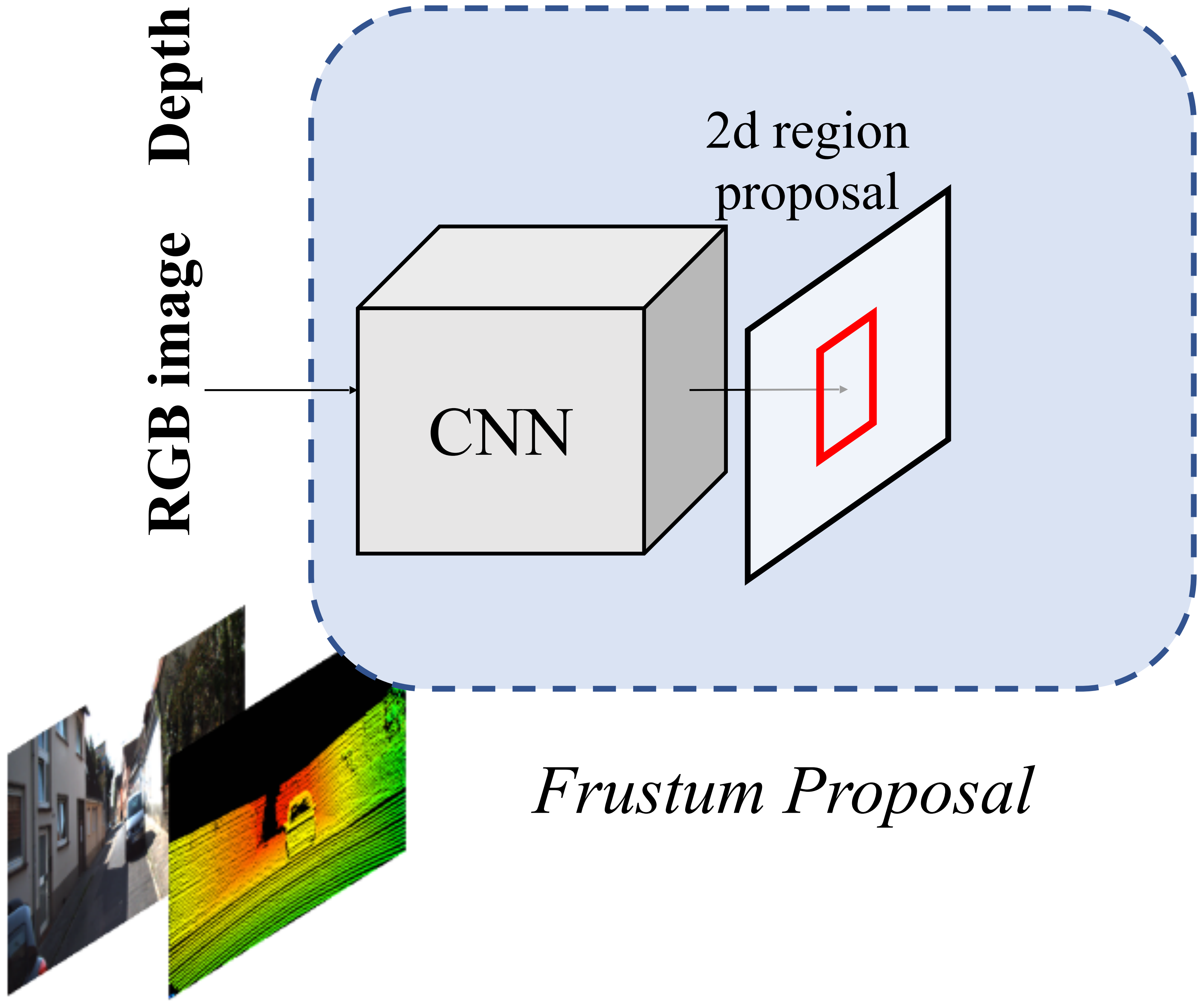
Propose 3D frustums by 2D region proposals in images and depth pop-up



Frustum Proposal

Input: RGB-D data

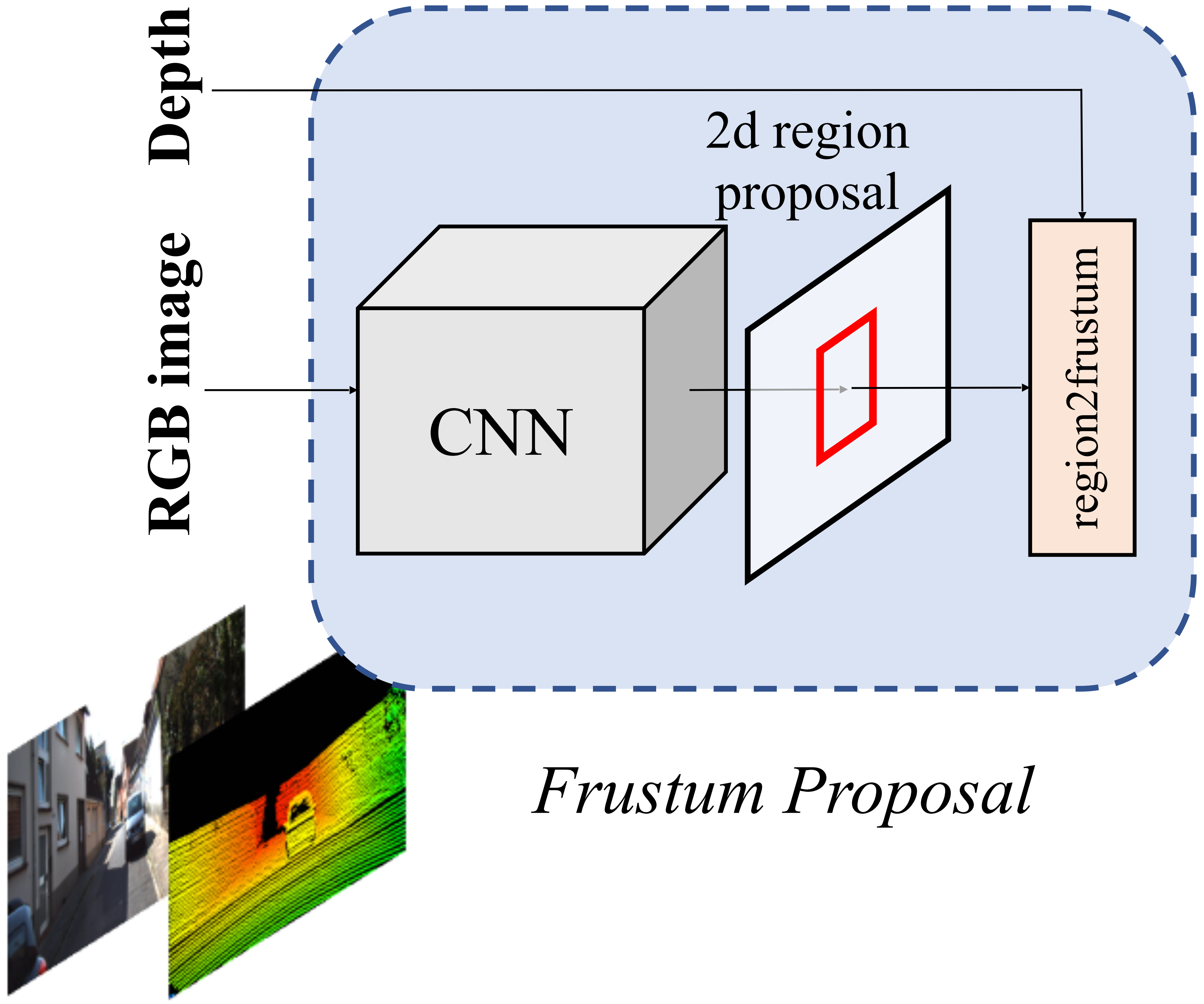
Frustum Proposal



Input: RGB-D data

Image region proposal using a 2D detector on RGB images (high resolution)

Frustum Proposal

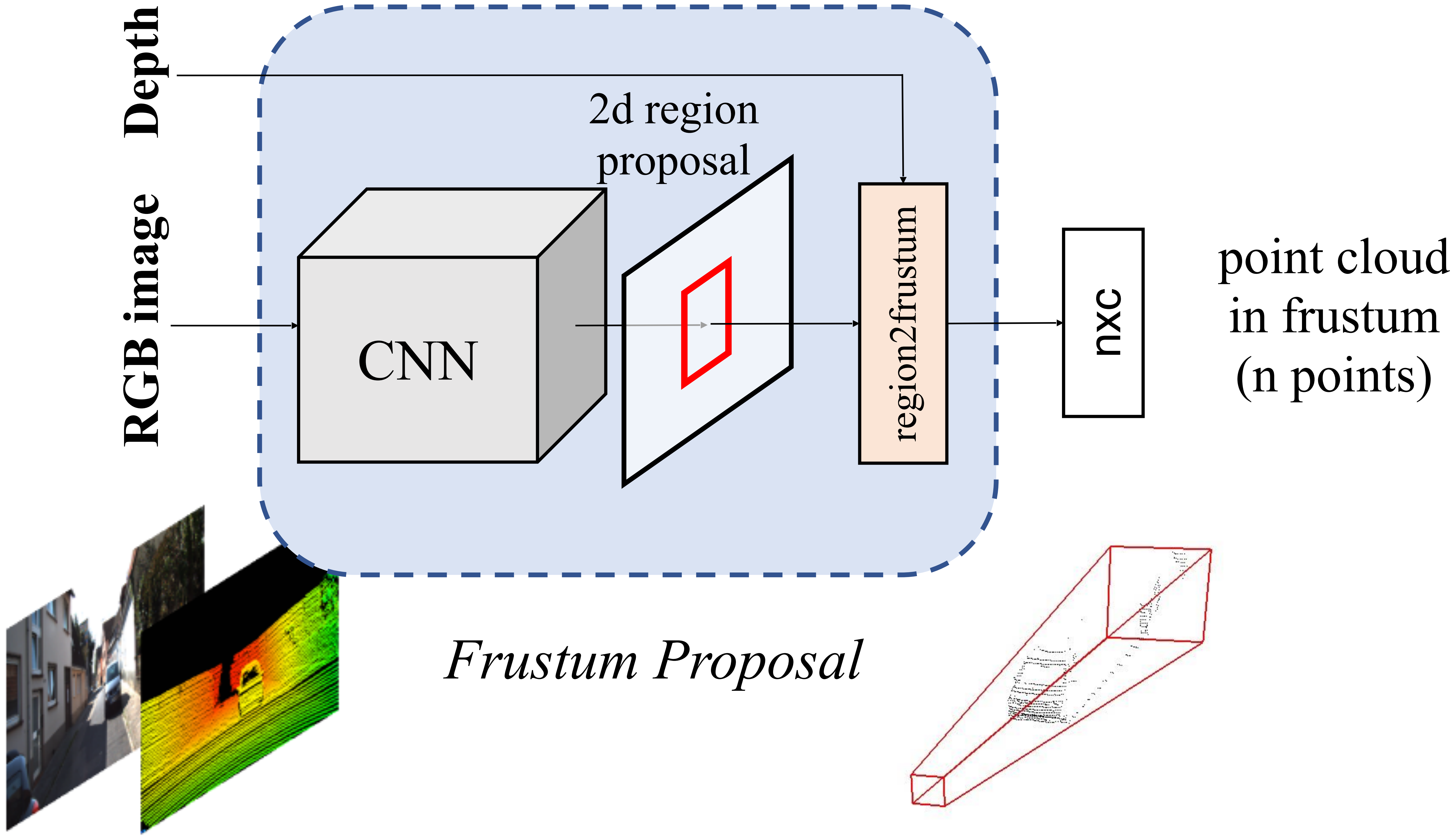


Input: RGB-D data

Image region proposal using a 2D detector on RGB images (high resolution)

Frustum proposal by lifting a 2D region into a 3D frustum.

Frustum Proposal



Input: RGB-D data

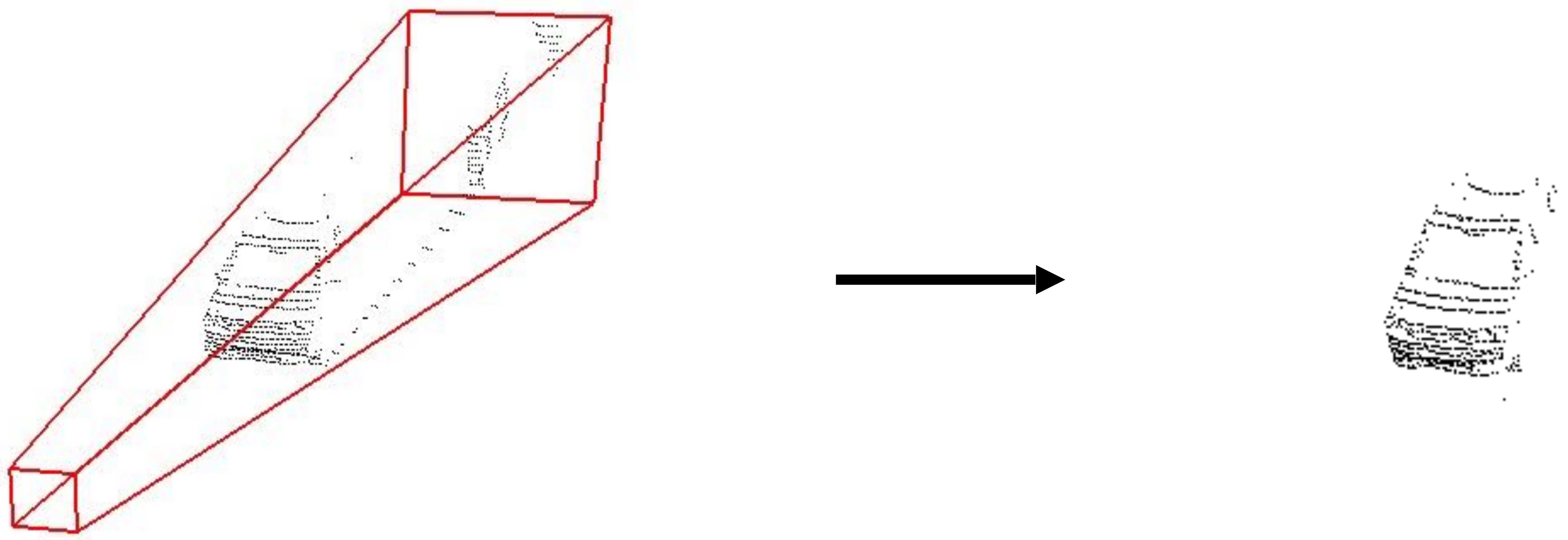
Image region proposal using a 2D detector on RGB images (high resolution)

Frustum proposal by lifting a 2D region into a 3D frustum.

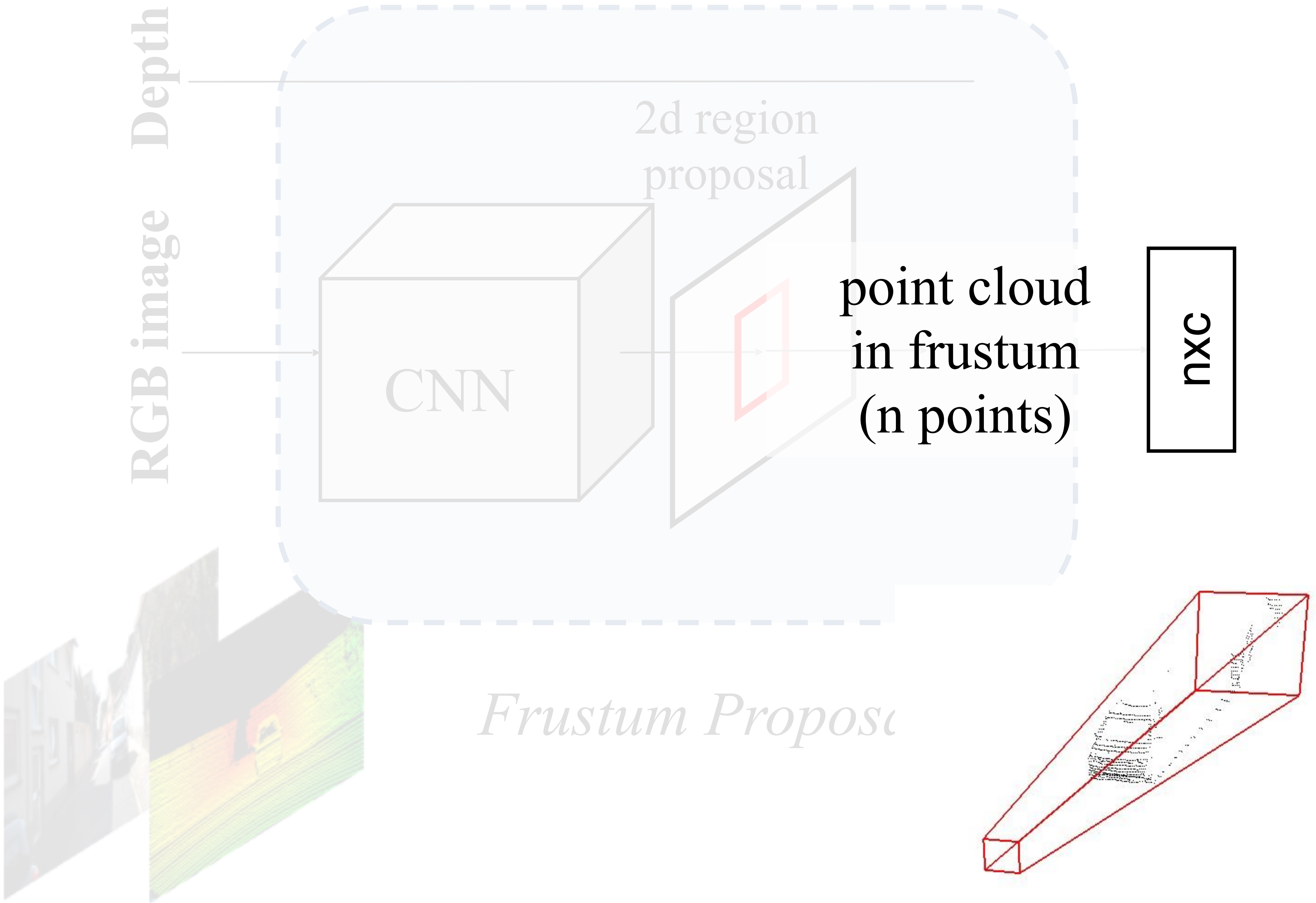
Points in the frustum are extracted and are called a *frustum point cloud*.

3D Instance Segmentation in Frustums

Localize objects in frustums by point cloud segmentation.

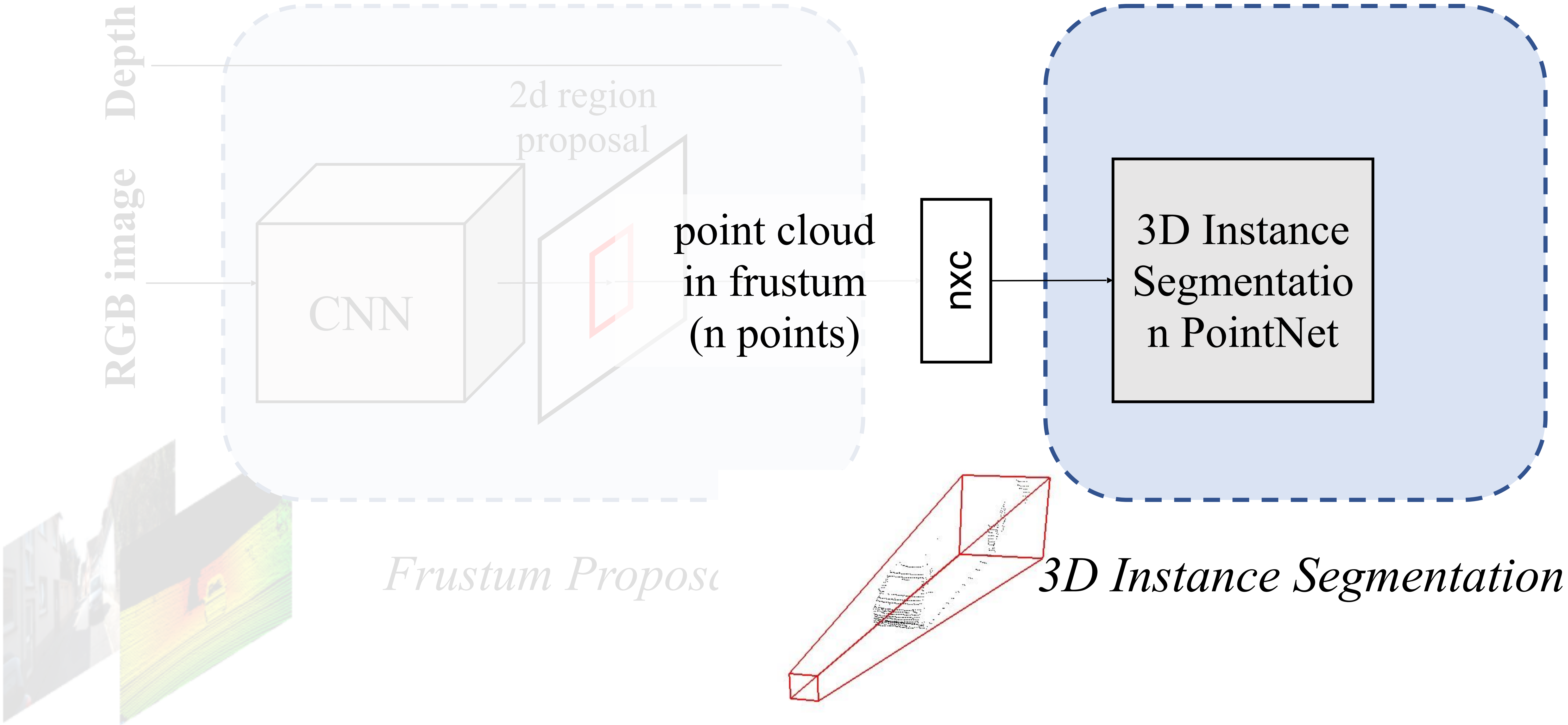


3D Instance Segmentation in Frustums



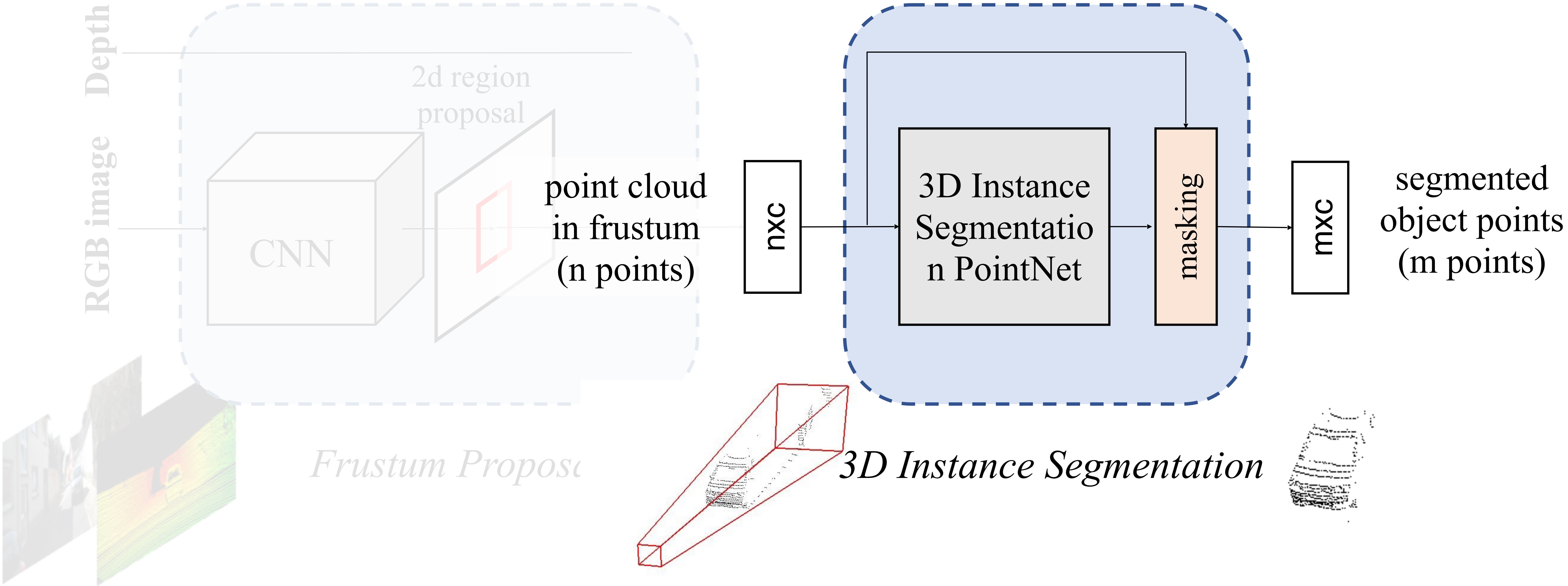
Input: frustum point cloud

3D Instance Segmentation in Frustums



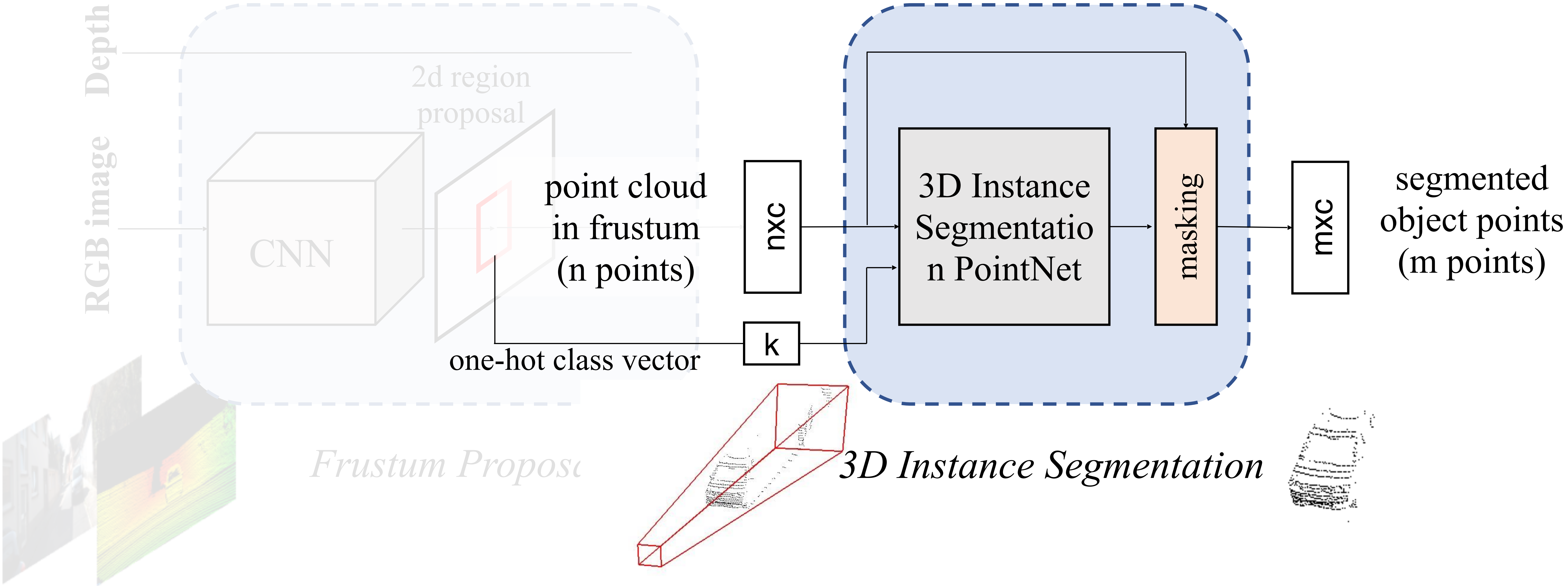
Input: frustum point cloud
Point cloud binary segmentation with PointNet: object of interest v.s. others

3D Instance Segmentation in Frustums



Input: frustum point cloud
Point cloud binary segmentation with PointNet: object of interest v.s. others
Points that are classified as object points are extracted for the next step.

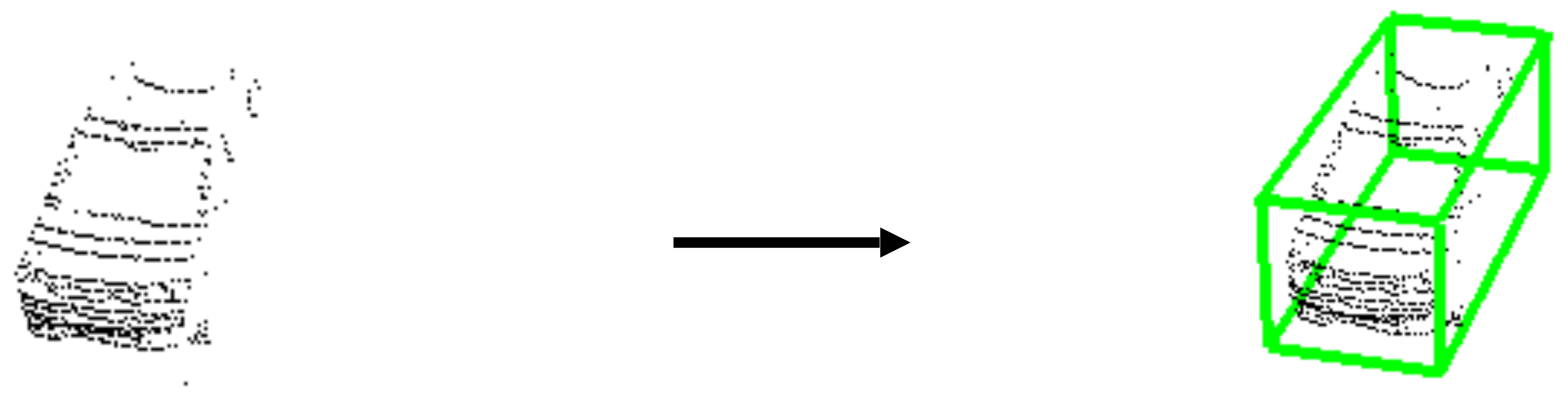
3D Instance Segmentation in Frustums



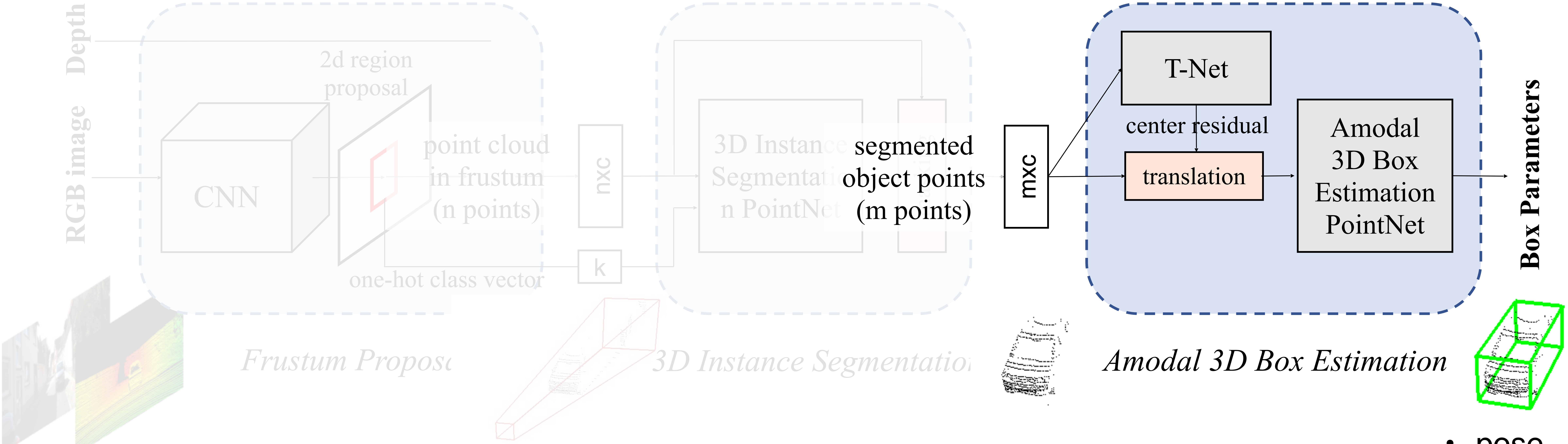
Input: frustum point cloud
Point cloud binary segmentation with PointNet: object of interest v.s. others
Points that are classified as object points are extracted for the next step.

Amodal 3D Bounding Box Estimation

Estimate 3D bounding boxes from segmented object point clouds.



Amodal 3D Bounding Box Estimation

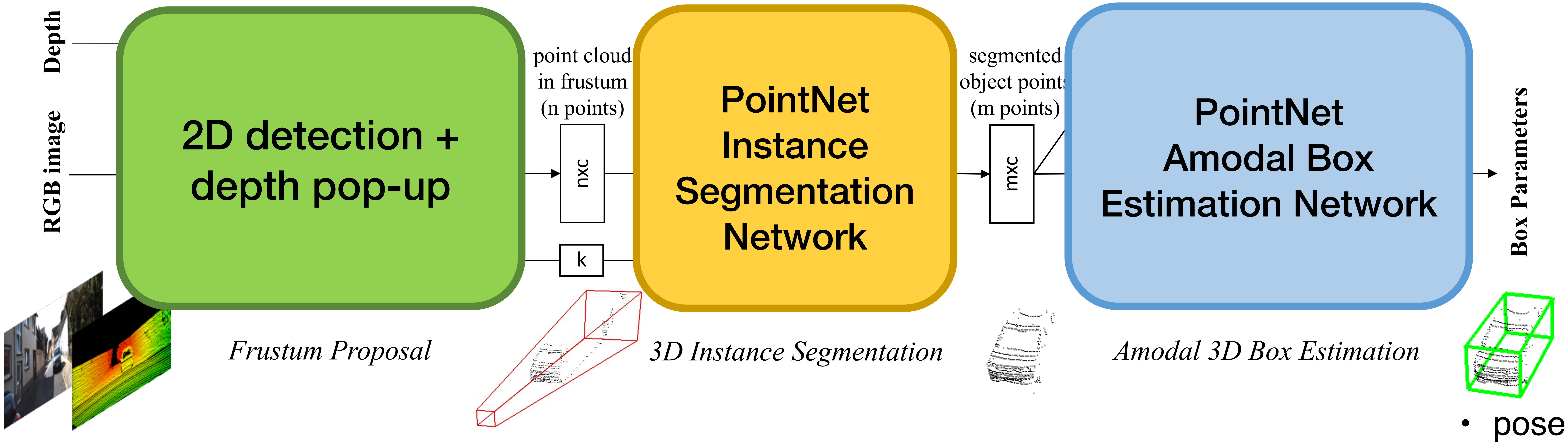


- pose
- size
- center

Input: object point cloud

A regression PointNet estimates amodal 3D bounding box for the object

Frustum PointNets

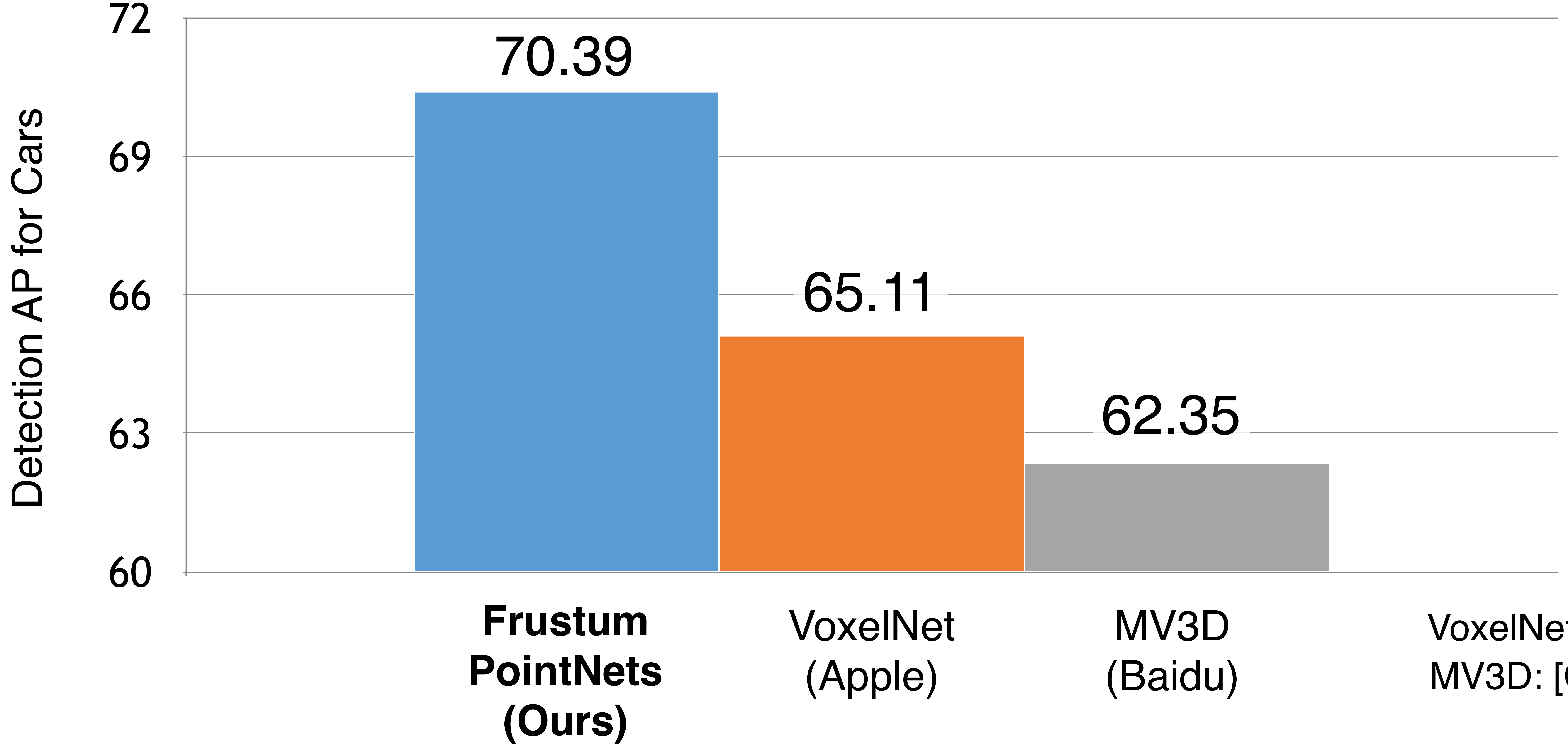


- pose
- size
- center

KITTI Results: Quantitative

Leading performance on KITTI benchmark

(at the time of publication)



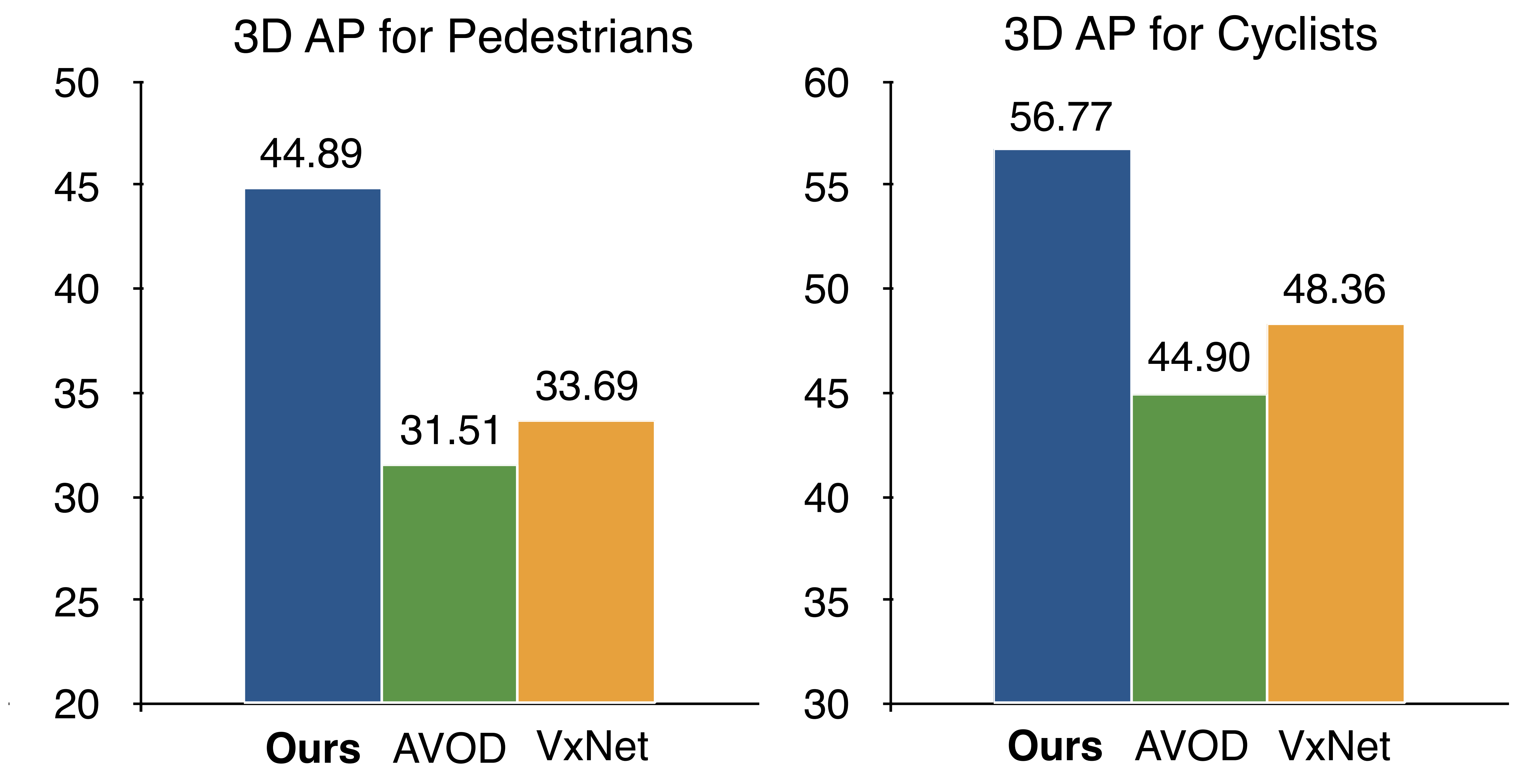
VoxelNet: [Zhou et al. 2018]
MV3D: [Chen et al. 2017]

KITTI Results: Quantitative

Leading performance on KITTI benchmark

(at the time of publication)

Especially leading at smaller objects (pedestrians and cyclists) – hard to localize with 3D proposals only.



AVOD: [Ku et al. 2018]
VxNet: [Zhou et al. 2017]

Frustum PointNets: Key to our Success

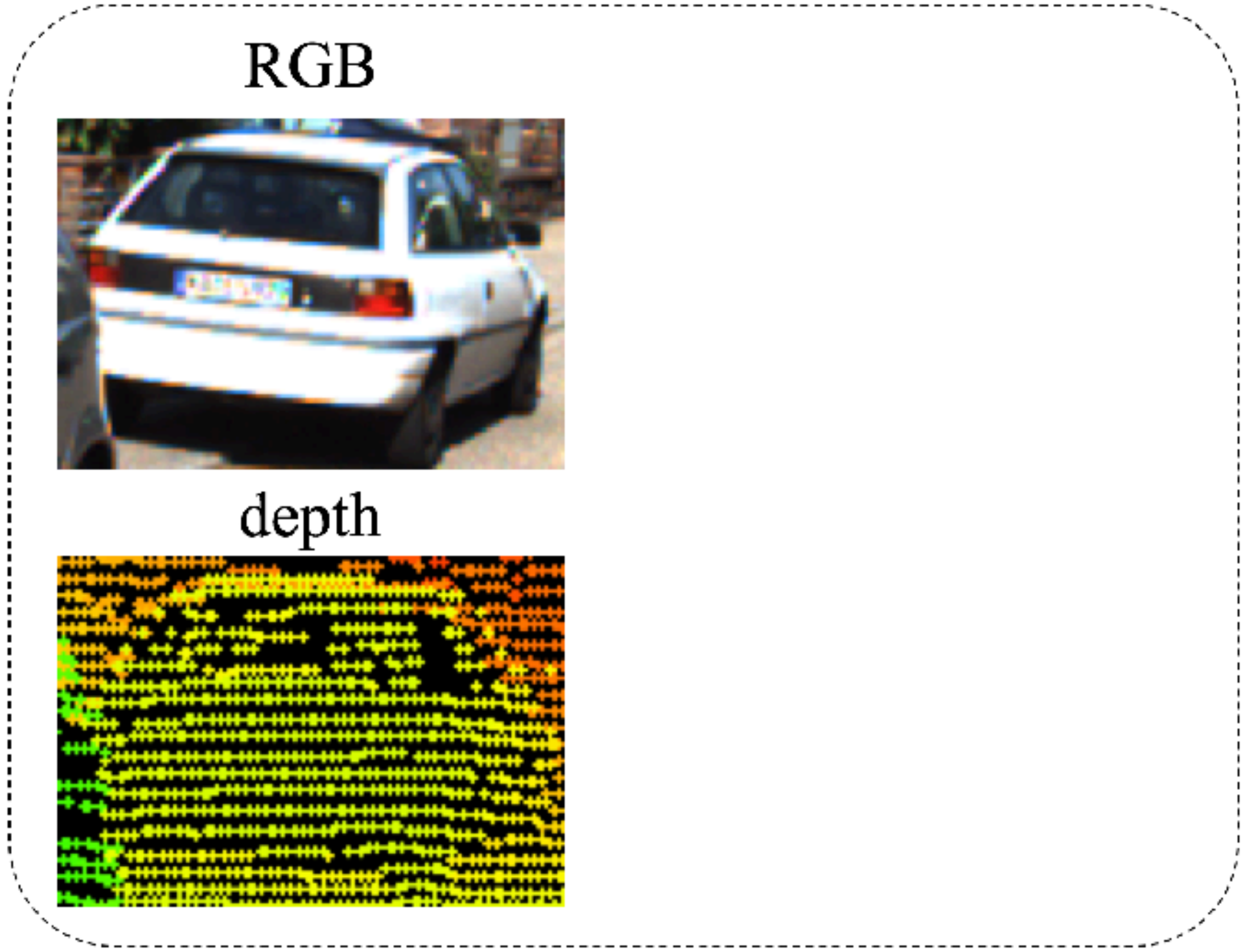
- **Representation matters — 2D v.s. 3D**

  Instance segmentation: depth range maps v.s. point clouds.

Frustum PointNets: Key to our Success

- **Representation matters — 2D v.s. 3D**

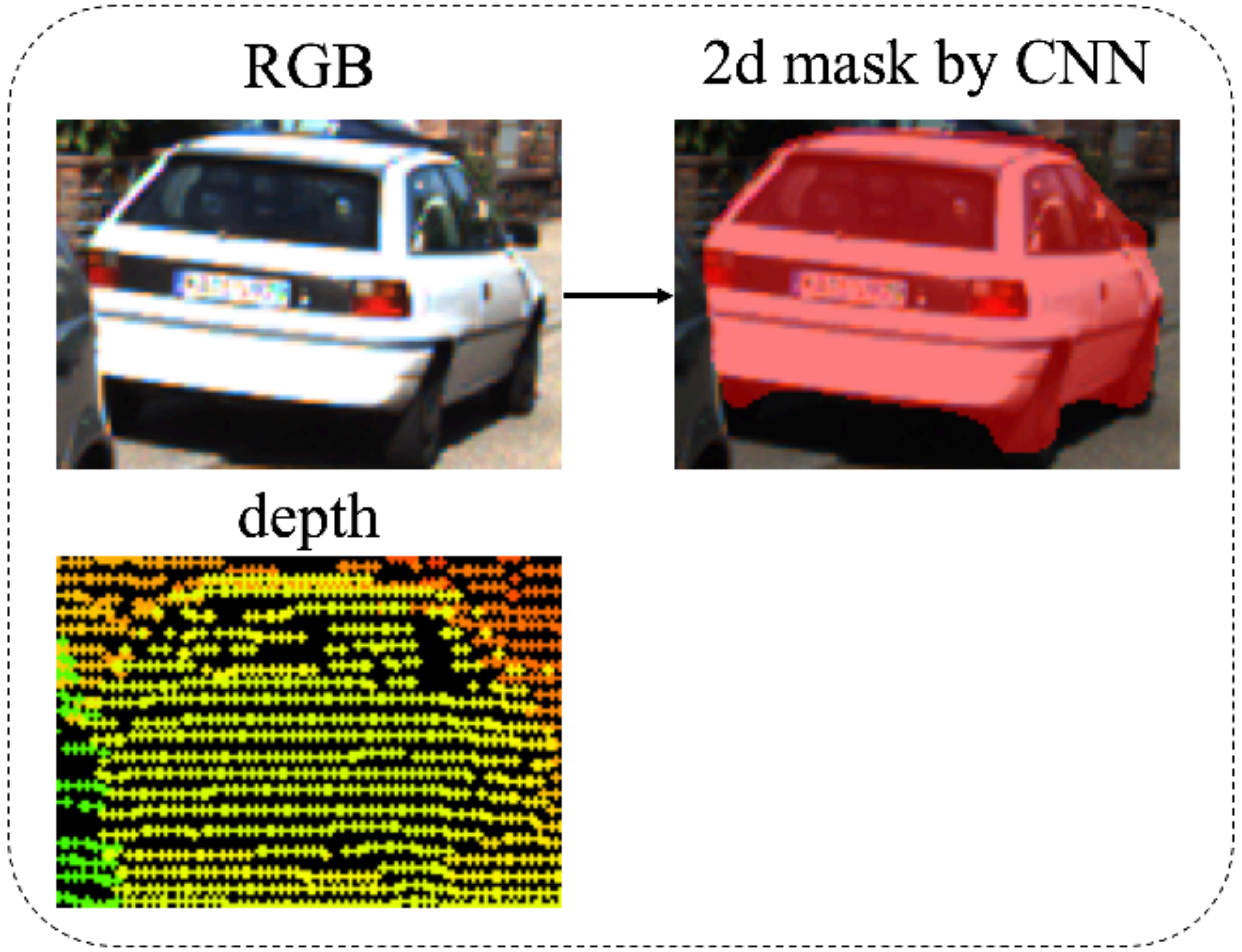
Instance segmentation: depth range maps v.s. point clouds.



Frustum PointNets: Key to our Success

- **Representation matters — 2D v.s. 3D**

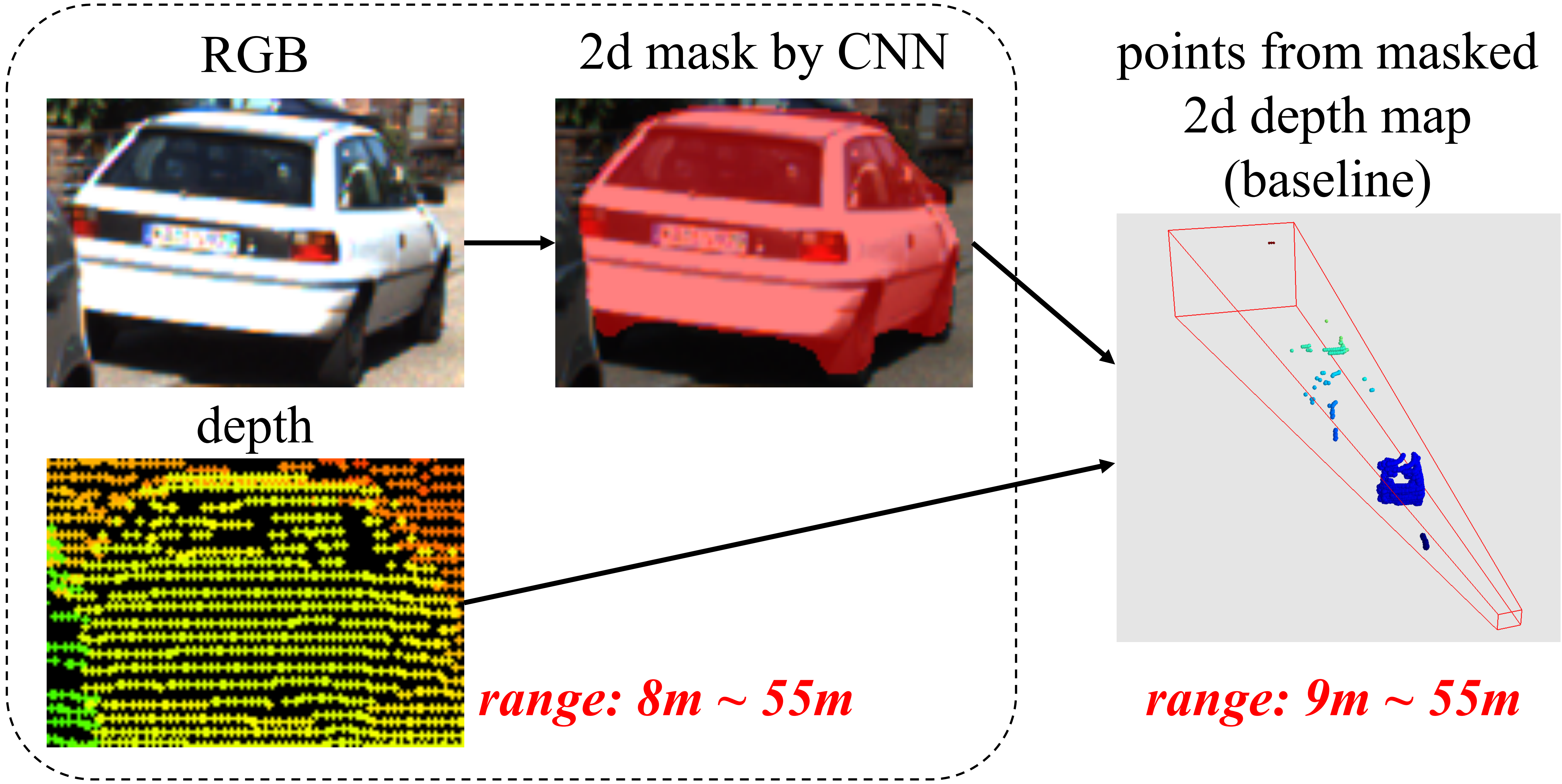
Instance segmentation: depth range maps v.s. point clouds.



Frustum PointNets: Key to our Success

- **Representation matters — 2D v.s. 3D**

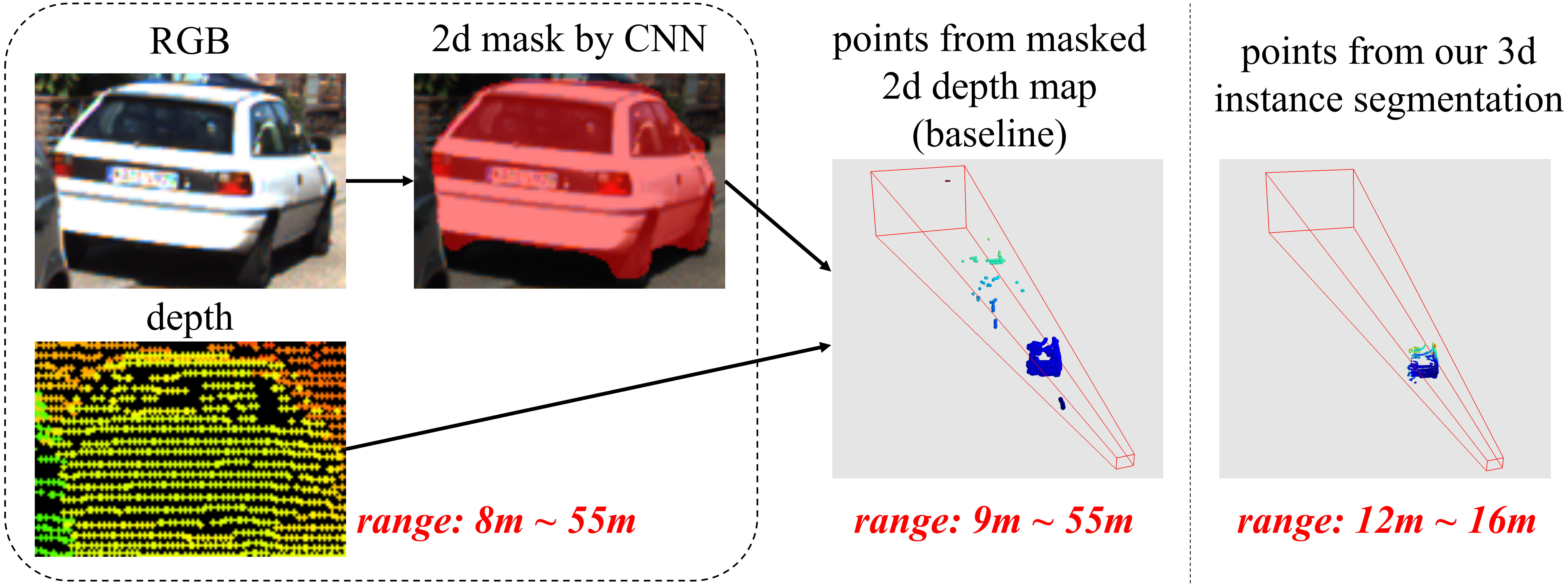
Instance segmentation: depth range maps v.s. point clouds.



Frustum PointNets: Key to our Success

- **Representation matters — 2D v.s. 3D**

  Instance segmentation: depth range maps v.s. point clouds.



Frustum PointNets: Key to our Success

- **Representation matters — 2D v.s. 3D**

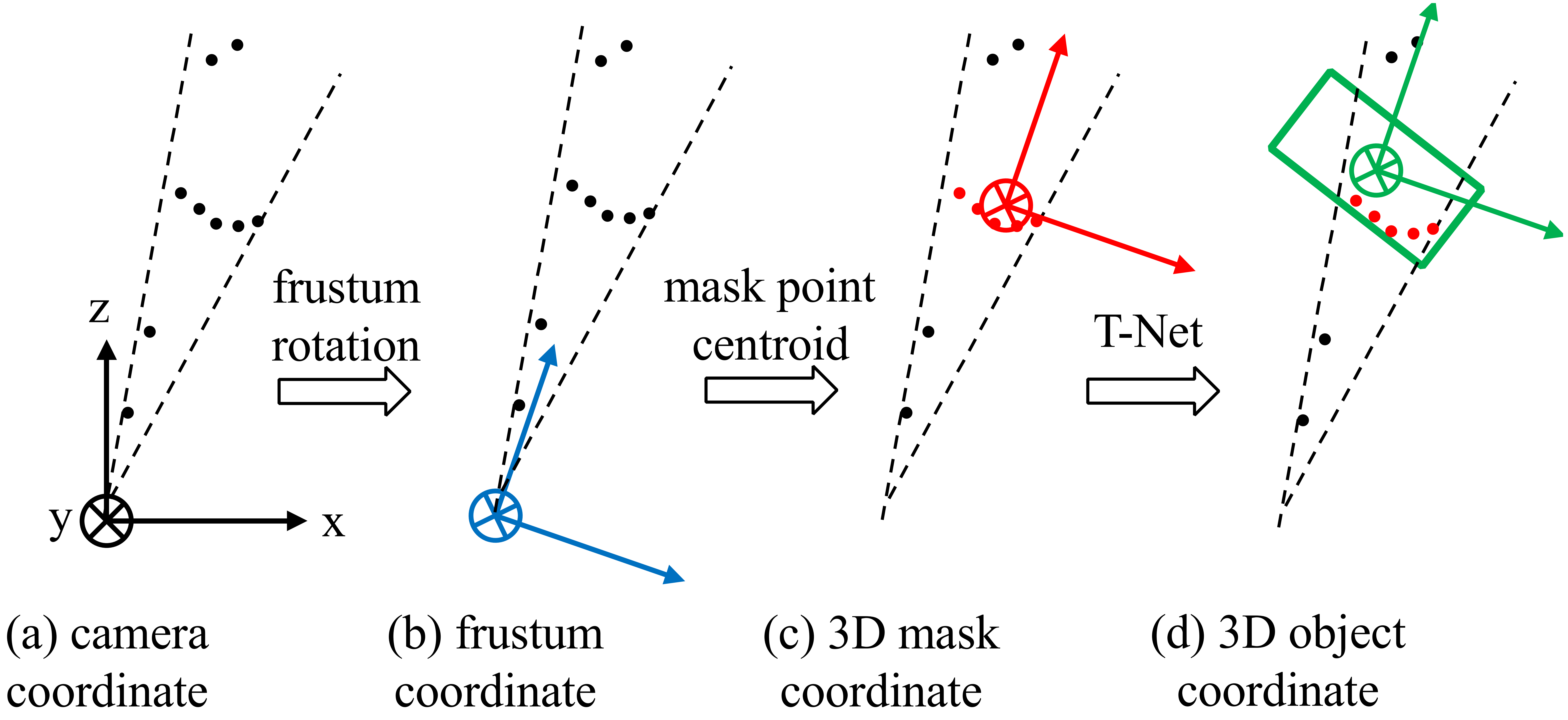
Effects of depth representation

network arch.	mask	depth representation	accuracy
ConvNet	-	image	18.3
ConvNet	2D	image	27.4
PointNet	-	point cloud	33.5
PointNet	2D	point cloud	61.6
PointNet	3D	point cloud	**74.3**
PointNet	2D+3D	point cloud	70.0

dataset: KITTI; metric: 3D bounding box estimation accuracy (%) under IoU 0.7

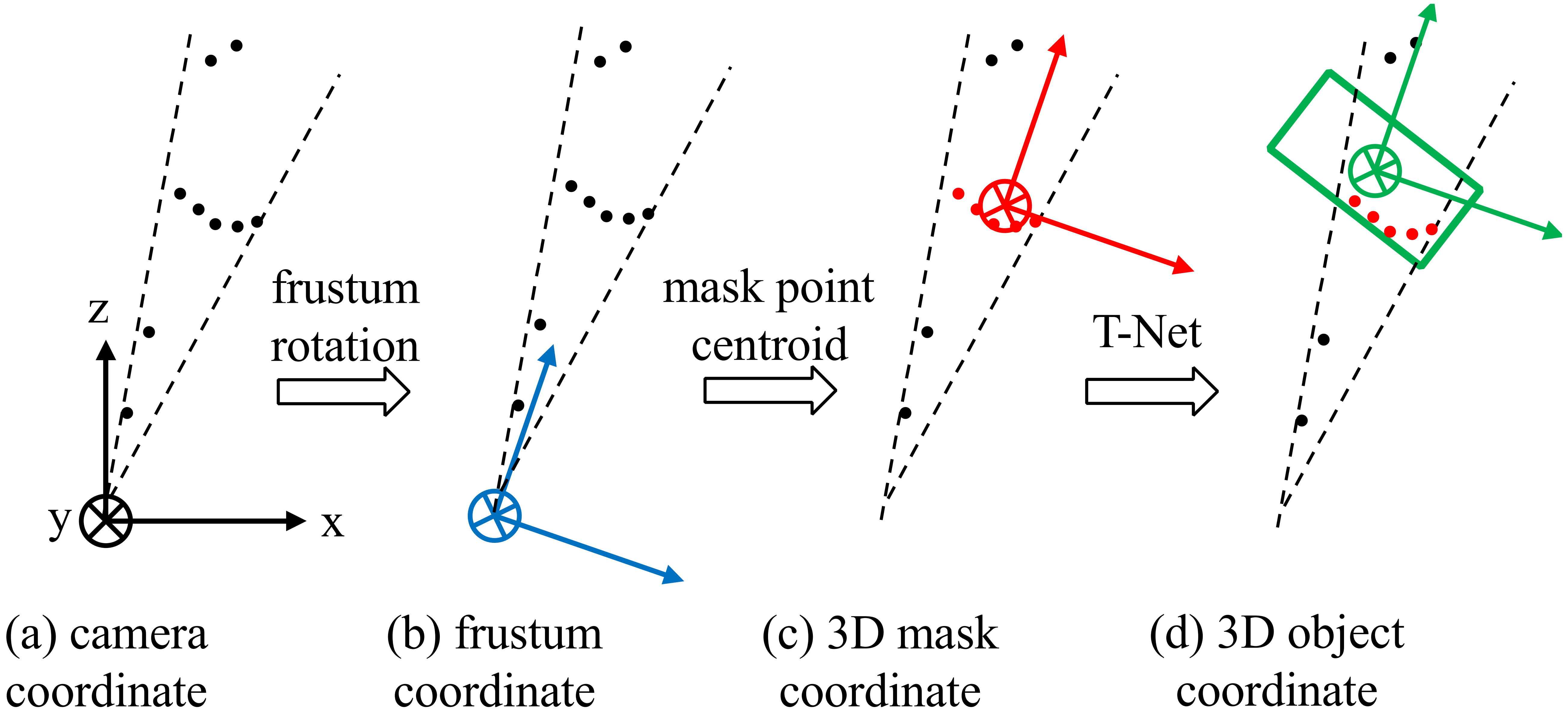
Frustum PointNets: Key to our Success

- **Canonicalize the problem with coordinate transformations**



Frustum PointNets: Key to our Success

- **Canonicalize** **the problem with coordinate transformations**



frustum rot.	mask centralize	t-net	accuracy
-	-	-	12.5
√	-	-	48.1
-	√	-	64.6
√	√	-	71.5
√	√	√	**74.3**

dataset: KITTI; metric: 3D bounding box estimation accuracy (%) under IoU 0.7

Frustum PointNets: Key to our Success

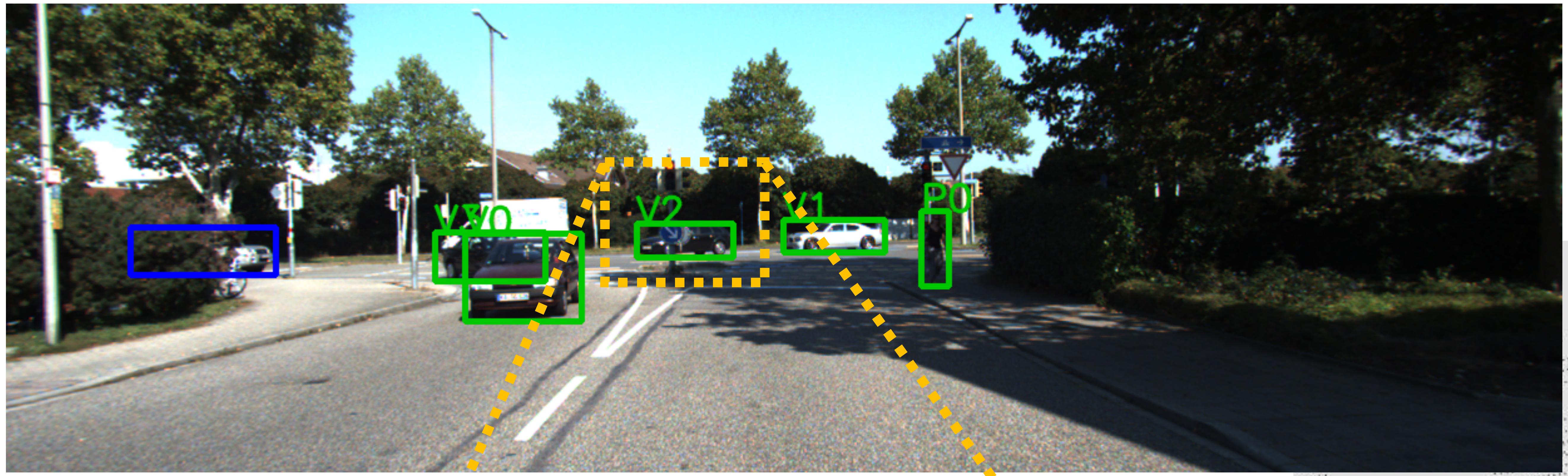
Respect and exploit 3D

- **Representation matters** — using 3D representation and 3D deep learning for the 3D problem.
- **Canonicalize the problem** — exploiting geometric transformations in point clouds.

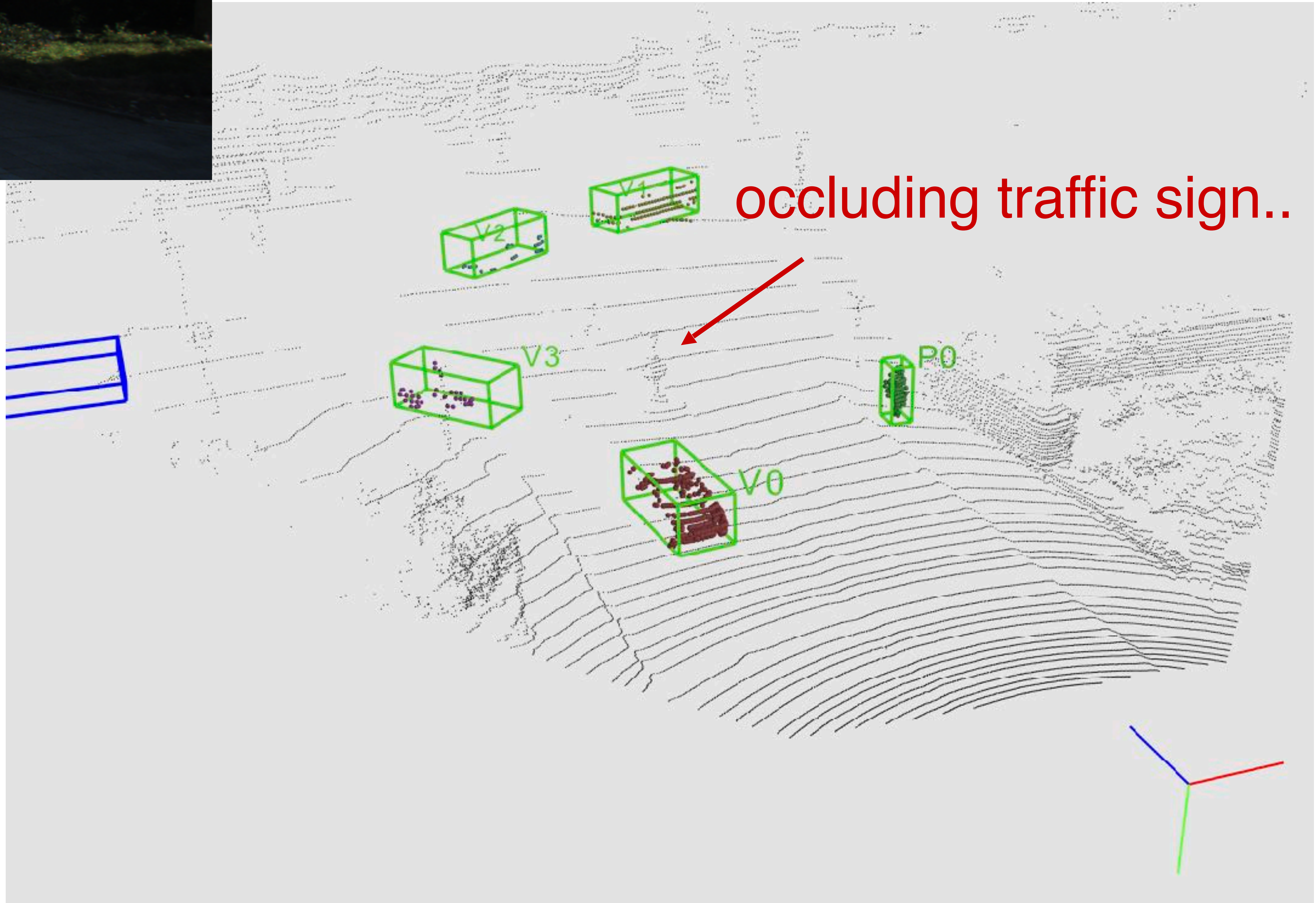
KITTI Results: Qualitative

Remarkable box estimation accuracy even with a dozen of points or with very partial point clouds.

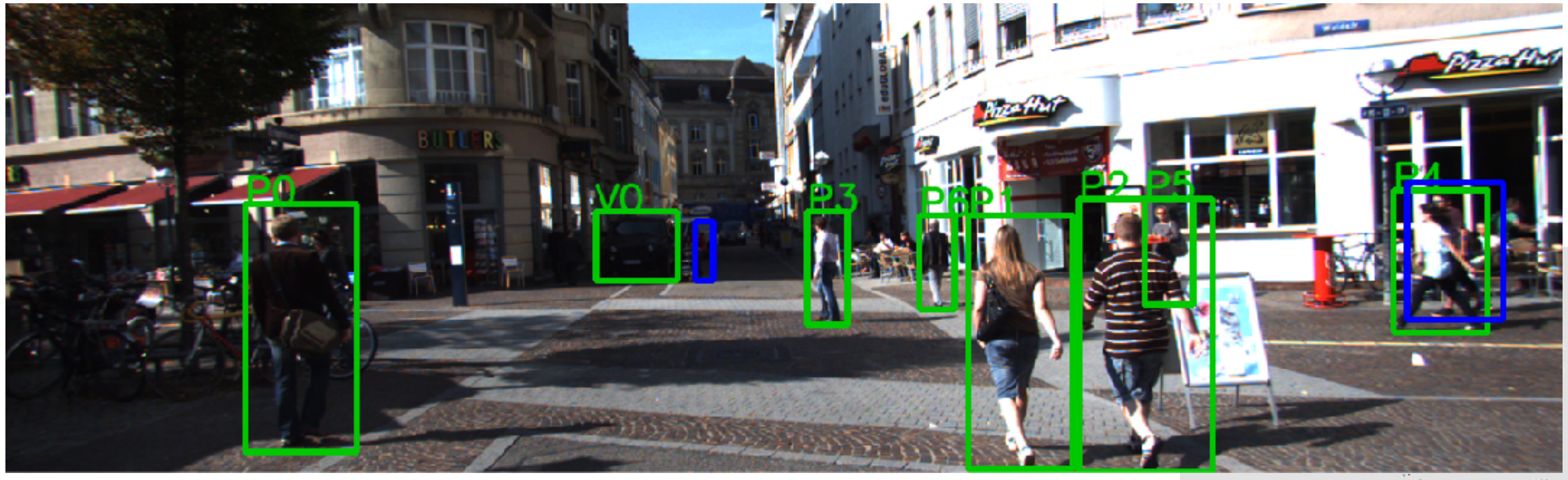
KITTI Results: Qualitative

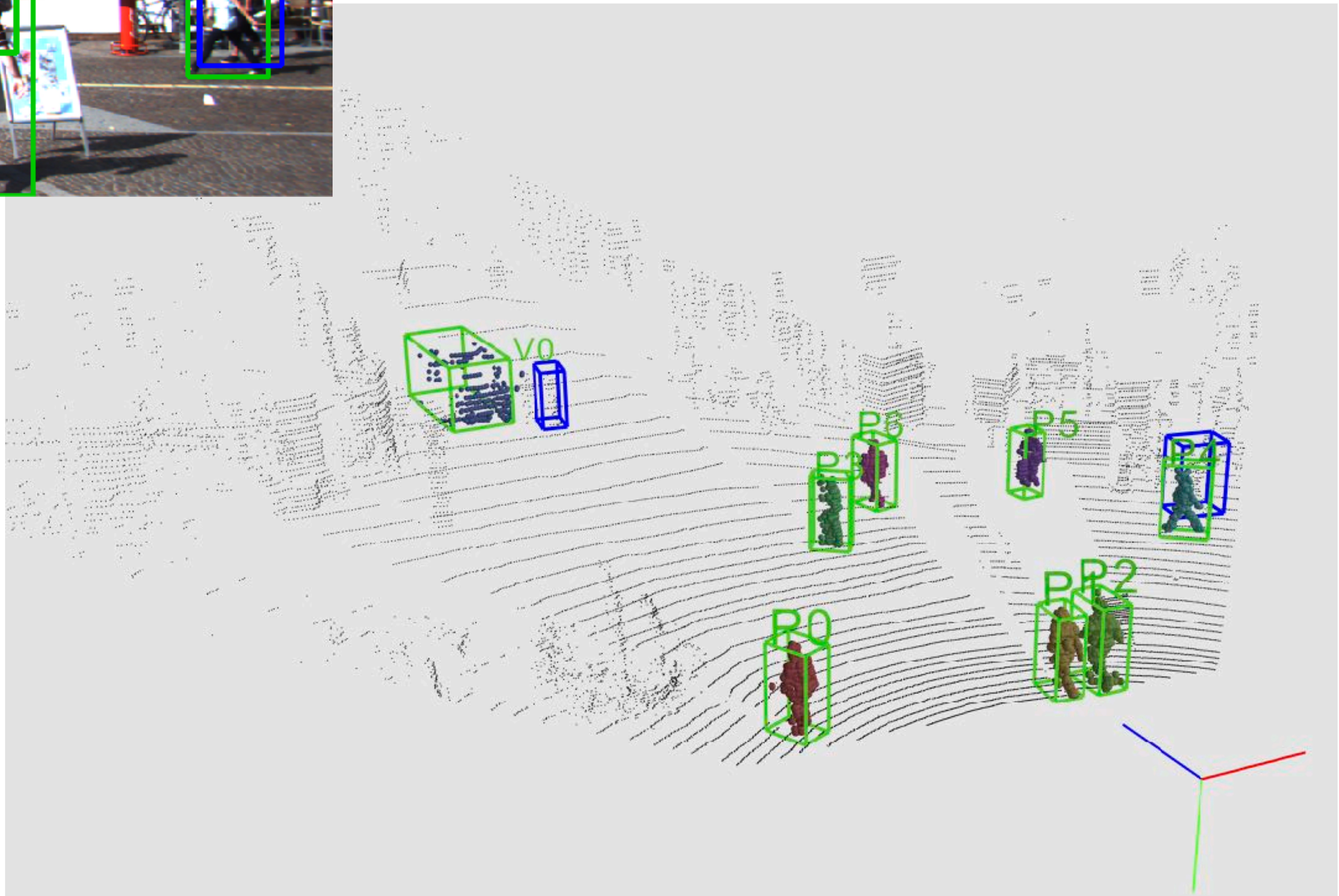


Correct segmentation in point clouds with heavy occlusion.



KITTI Results: Qualitative

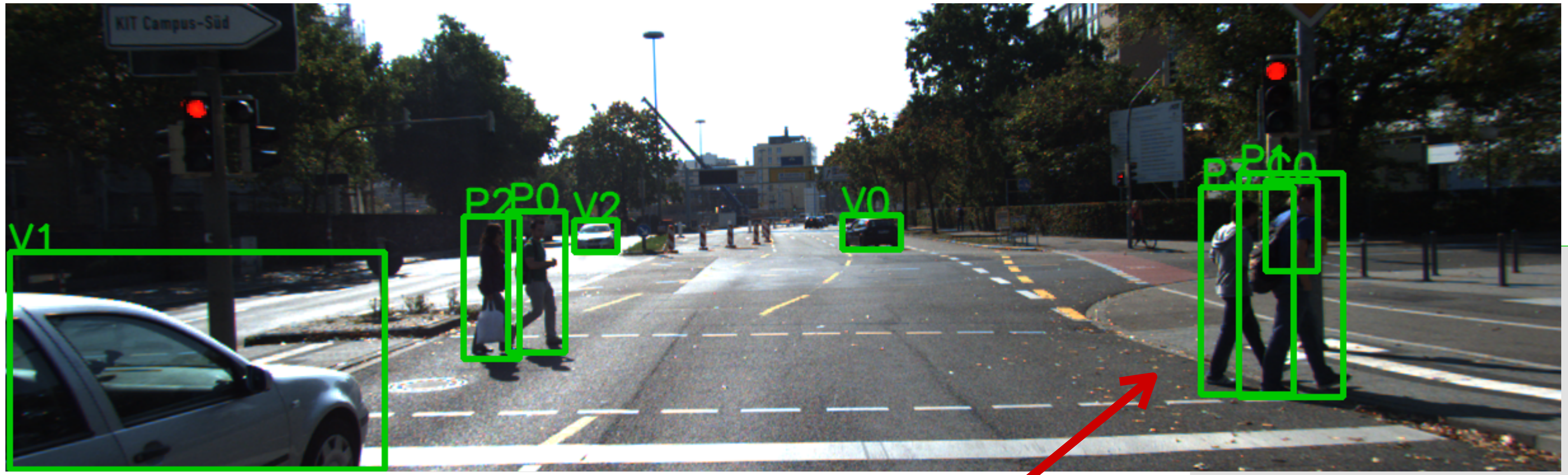




Missing 2D detection results in no 3D detection

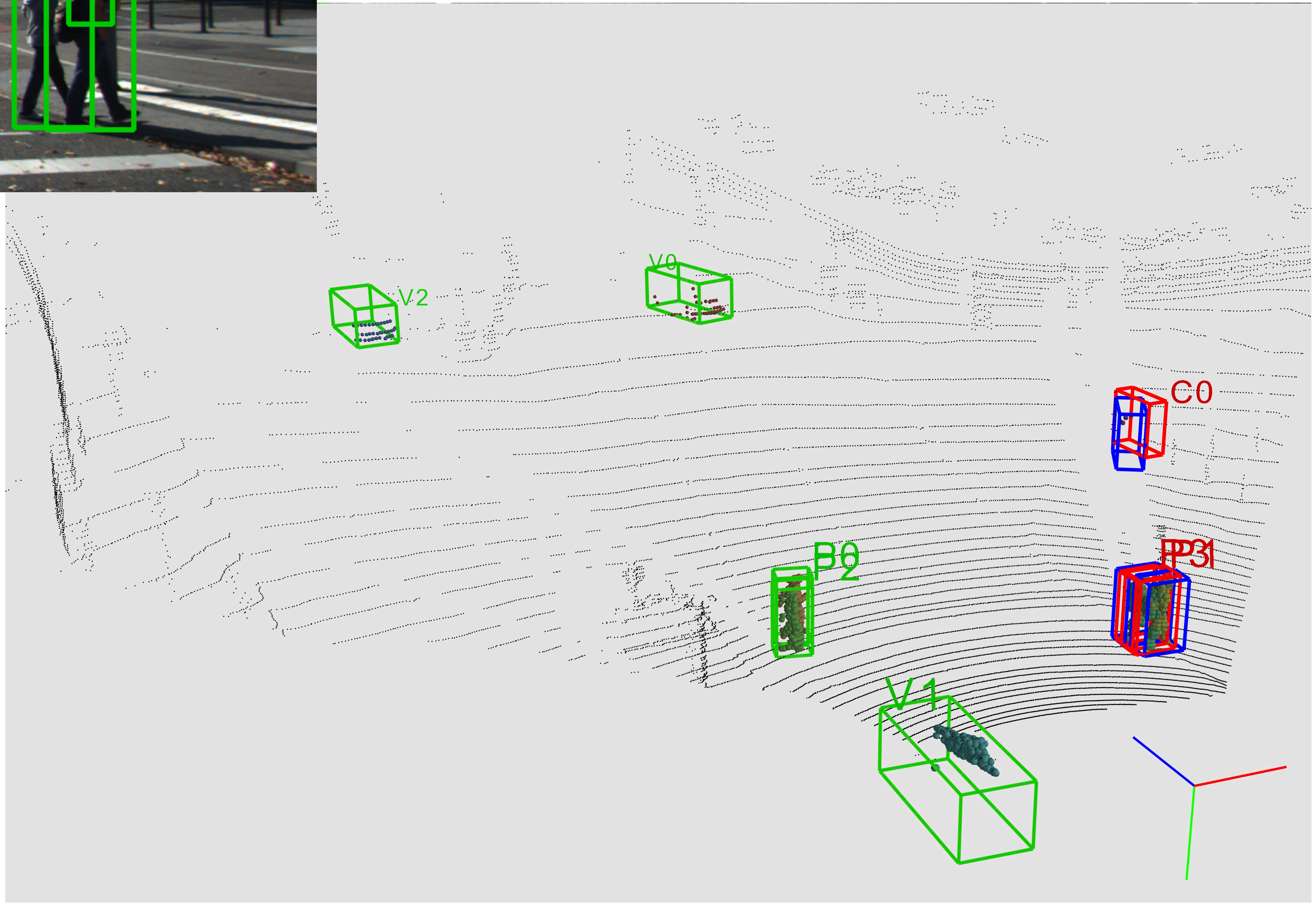
Multiple ways of proposal could help (e.g. bird's eye view, multiple 2D proposal networks)

KITTI Results: Qualitative



Very strong occlusion.

Challenging case for instance segmentation (multiple close-by objects in a single frustum)



Limitation of the Frustum PointNets

- Hard dependence on 2D detections: will miss objects due to strong occlusions in 2D views or unfavorable illumination conditions.
- No support of multiple 3D proposals in a frustum.

Solution: ***object proposal from 3D point clouds.***

(VoteNet & ImVoteNet)

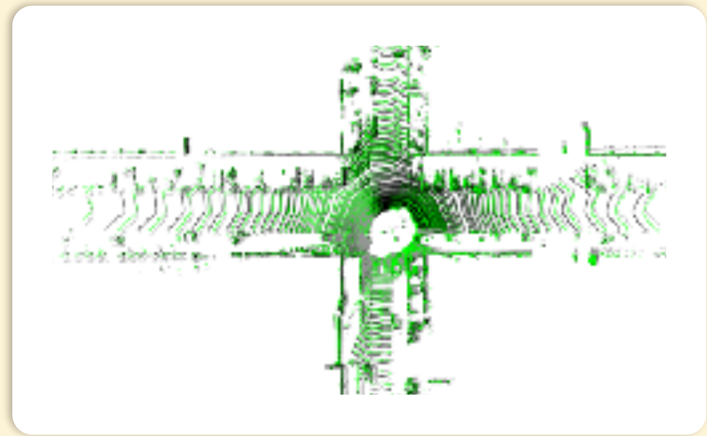
The deep learning era of 3d object detection

Image-driven

Monocular view detectors
Frustum-based detectors

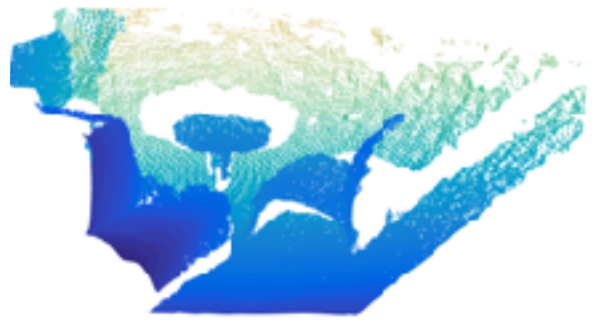
Dimension reduction

Bird’s eye view detectors



Leveraging Sparsity in 3D

Point set deep nets
Sparse 3D conv, GNNs



E.g.:

Frustum PointNets [6]

PointPillars [7]

VoteNet [8]

Bird's eye view 3D object detector

- Key idea: Converting the 3D learning problem to a 2D learning problem.

Volumetric and Multi-View CNNs for Object Classification on 3D Data (CVPR'16) by Qi et al. [12]

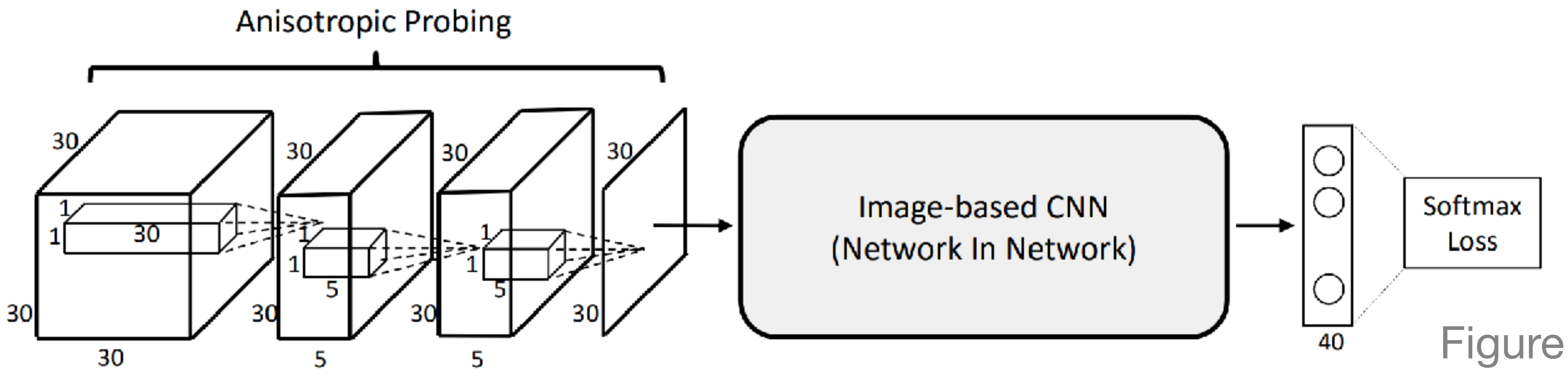


Figure from [12]

3D CNN with Anisotropic Probing kernels.
We **use an elongated kernel to convolve the 3D cube and aggregate information to a 2D plane.**
Then we use a 2D NIN (NIN-CIFAR10 [23]) to classify the 2D projection of the original 3D shape.

Bird's eye view 3D object detector

- Key idea: Converting the 3D learning problem to a 2D learning problem.

The work that started the KITTI 3D object detection challenge:

Multi-View 3D Object Detection Network for Autonomous Driving (2017) [13]
  - Hand designed features are used to convert a 3D scene point cloud to a bird's eye image.

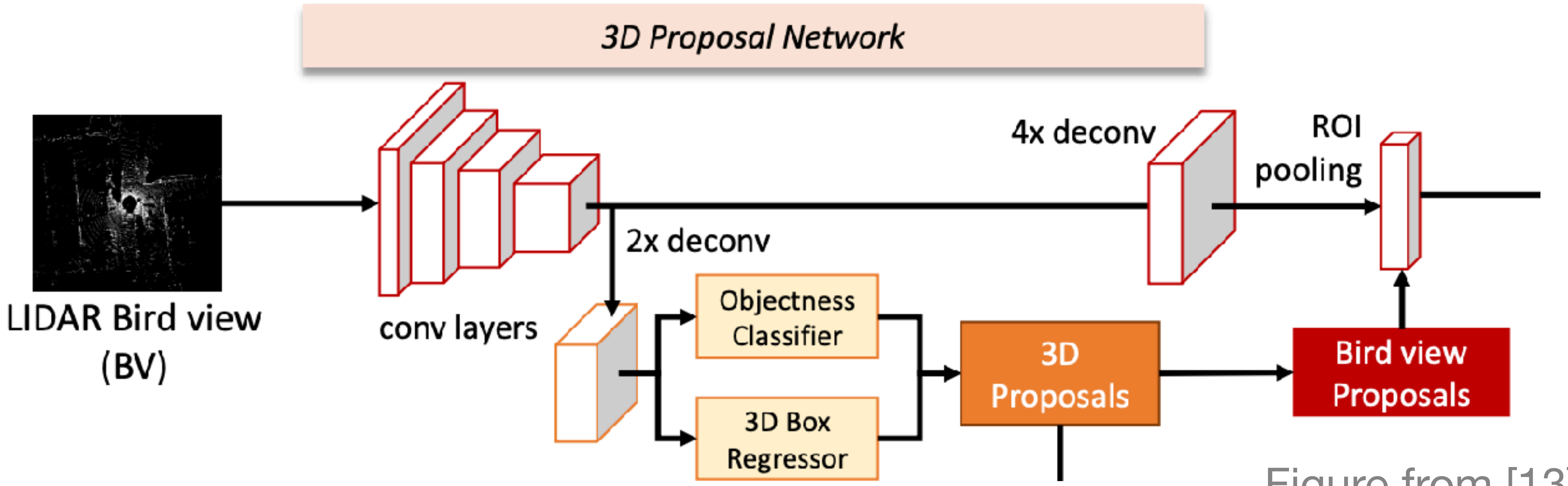


Figure from [13]

Bird’s eye view 3D object detector

- From hand designed projection to data-driven projection (with PointNet like architectures).

Voxelnet: End-to-end learning for point cloud based 3d object detection (2018) [14]
Pointpillars: Fast encoders for object detection from point clouds (2019) [7]

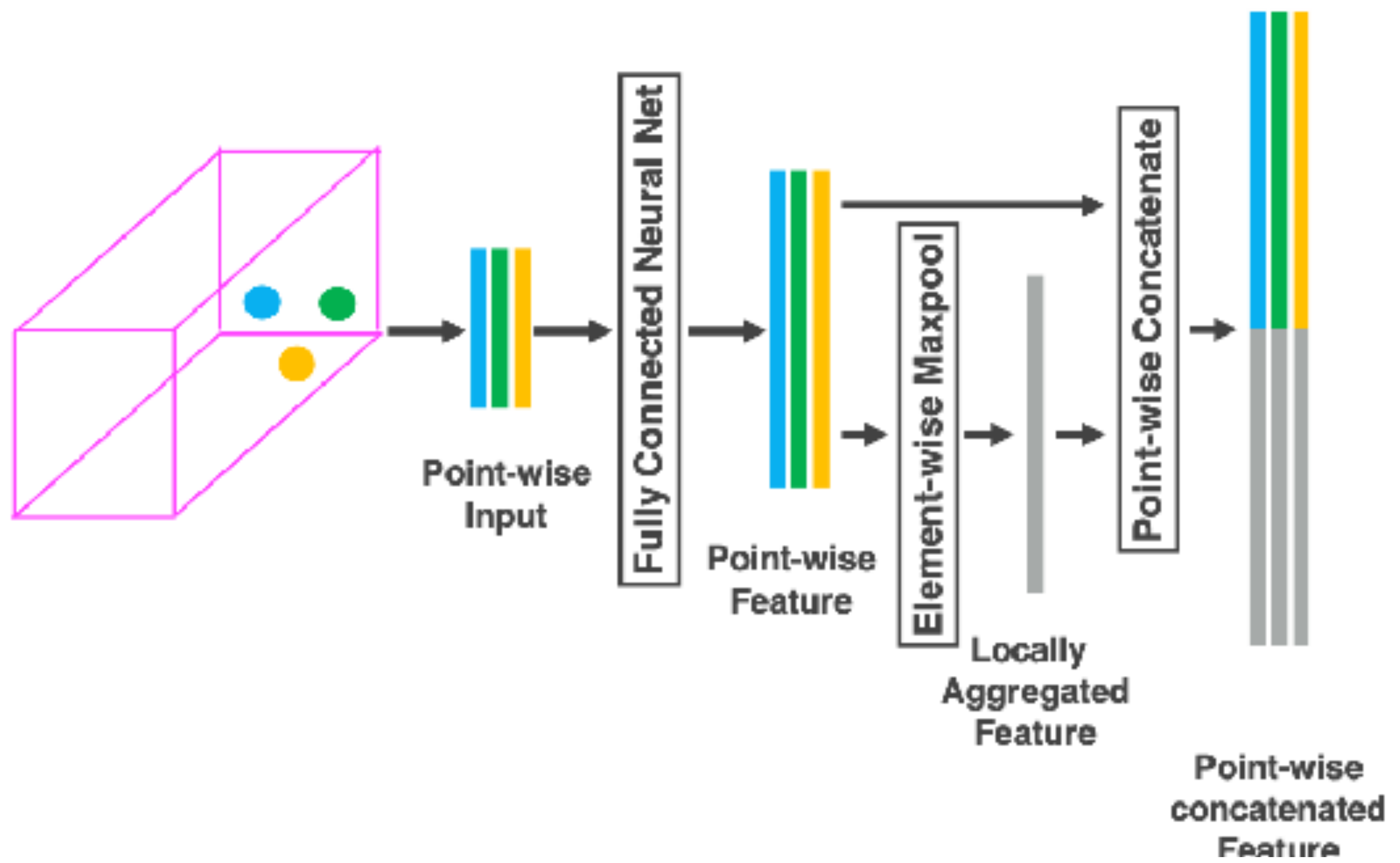


Figure from [14]

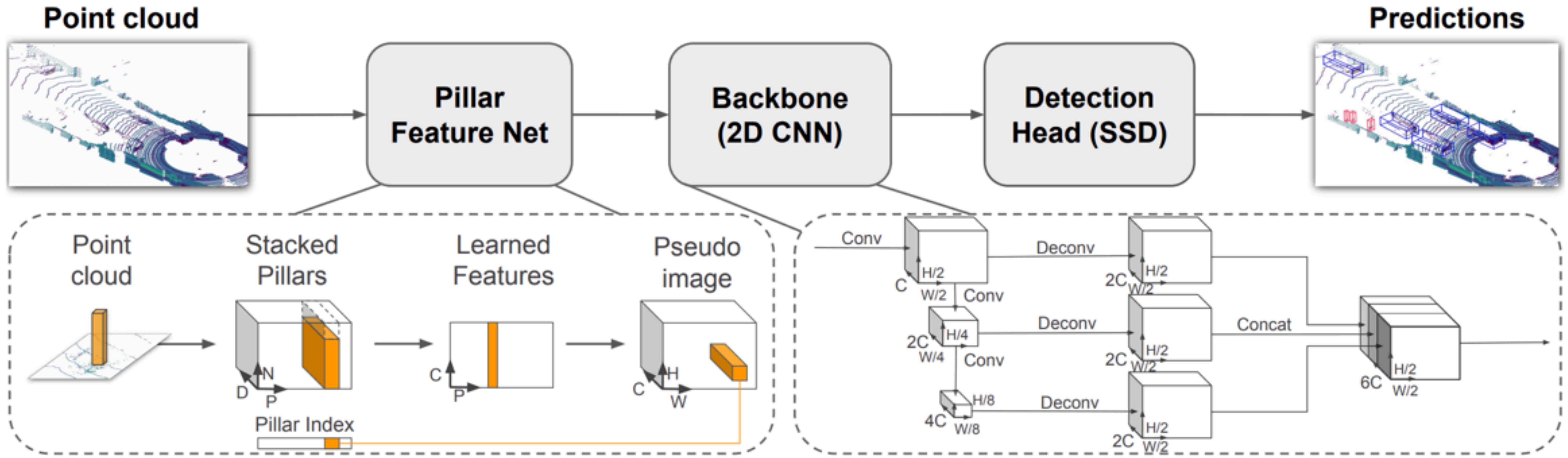
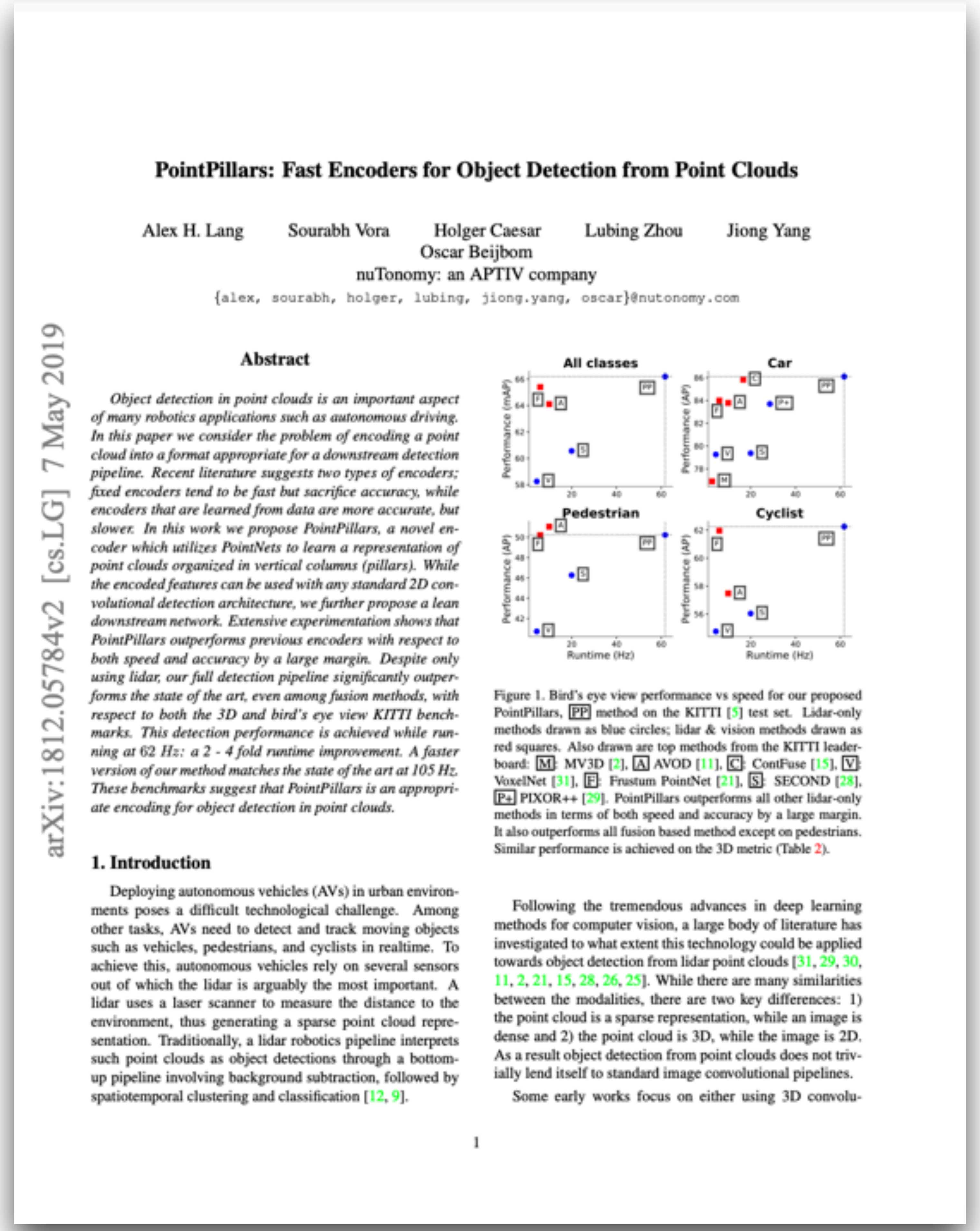


Figure from [7]

PointPillars: Fast Encoders for Object Detection from Point Clouds

Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, Oscar Beijbom.

CVPR 2019



arXiv:1812.05784v2 [cs.LG] 7 May 2019

PointPillars: Fast Encoders for Object Detection from Point Clouds

Alex H. Lang Sourabh Vora Holger Caesar Lubing Zhou Jiong Yang
Oscar Beijbom
nuTonomy: an APTIV company
{alex, sourabh, holger, lubing, jiong.yang, oscar}@nutonomy.com

Abstract

Object detection in point clouds is an important aspect of many robotics applications such as autonomous driving. In this paper we consider the problem of encoding a point cloud into a format appropriate for a downstream detection pipeline. Recent literature suggests two types of encoders; fixed encoders tend to be fast but sacrifice accuracy, while encoders that are learned from data are more accurate, but slower. In this work we propose PointPillars, a novel encoder which utilizes PointNets to learn a representation of point clouds organized in vertical columns (pillars). While the encoded features can be used with any standard 2D convolutional detection architecture, we further propose a lean downstream network. Extensive experimentation shows that PointPillars outperforms previous encoders with respect to both speed and accuracy by a large margin. Despite only using lidar, our full detection pipeline significantly outperforms the state of the art, even among fusion methods, with respect to both the 3D and bird's eye view KITTI benchmarks. This detection performance is achieved while running at 62 Hz: a 2 - 4 fold runtime improvement. A faster version of our method matches the state of the art at 105 Hz. These benchmarks suggest that PointPillars is an appropriate encoding for object detection in point clouds.

Figure 1. Bird's eye view performance vs speed for our proposed PointPillars, PP method on the KITTI [5] test set. Lidar-only methods drawn as blue circles; lidar & vision methods drawn as red squares. Also drawn are top methods from the KITTI leaderboard: M: MV3D [2], A AVOD [11], C: ContFuse [15], V: VoxelNet [31], F: Frustum PointNet [21], S: SECOND [28], P+ PIXOR++ [29]. PointPillars outperforms all other lidar-only methods in terms of both speed and accuracy by a large margin. It also outperforms all fusion based method except on pedestrians. Similar performance is achieved on the 3D metric (Table 2).

1. Introduction

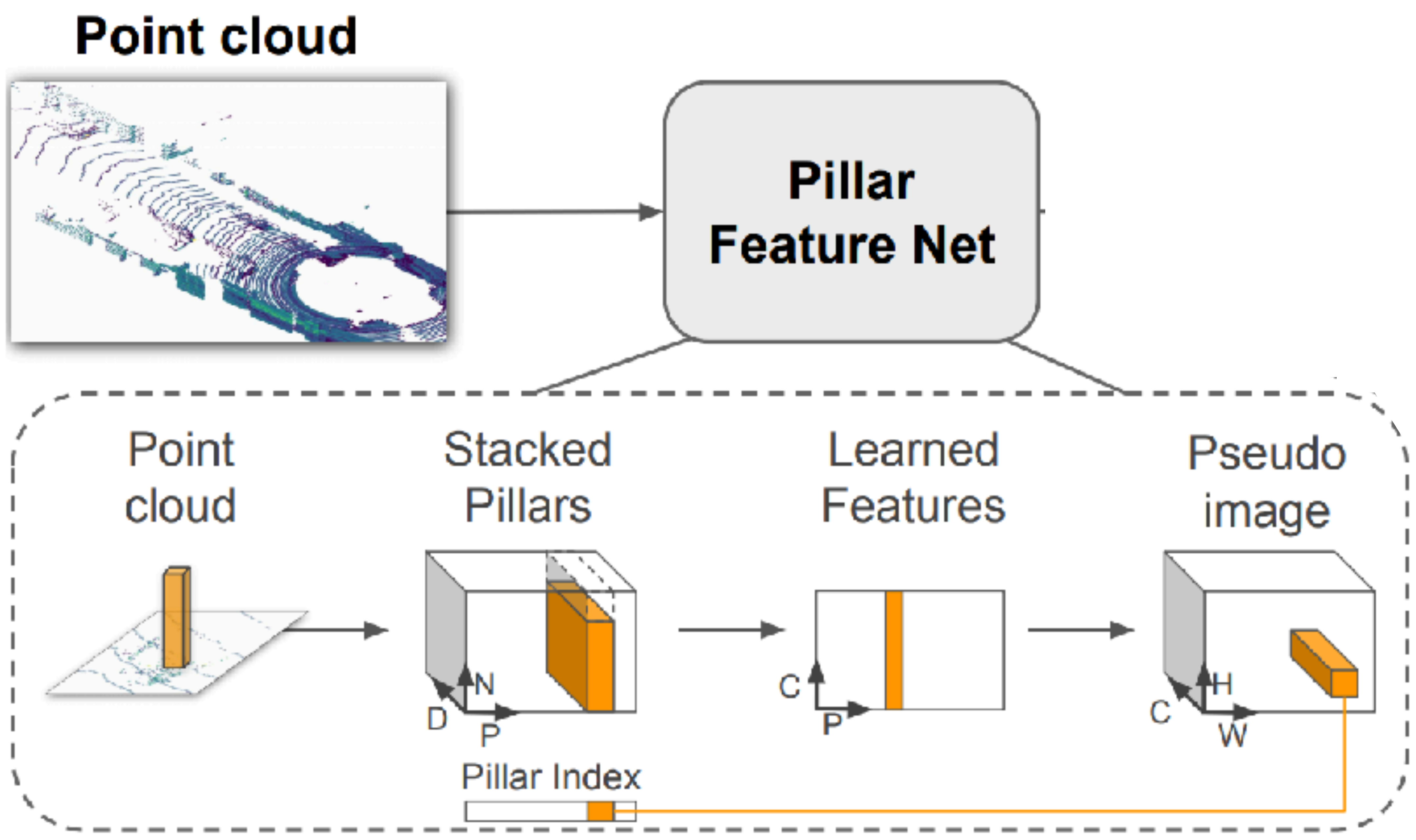
Deploying autonomous vehicles (AVs) in urban environments poses a difficult technological challenge. Among other tasks, AVs need to detect and track moving objects such as vehicles, pedestrians, and cyclists in realtime. To achieve this, autonomous vehicles rely on several sensors out of which the lidar is arguably the most important. A lidar uses a laser scanner to measure the distance to the environment, thus generating a sparse point cloud representation. Traditionally, a lidar robotics pipeline interprets such point clouds as object detections through a bottom-up pipeline involving background subtraction, followed by spatiotemporal clustering and classification [12, 9].

Following the tremendous advances in deep learning methods for computer vision, a large body of literature has investigated to what extent this technology could be applied towards object detection from lidar point clouds [31, 29, 30, 11, 2, 21, 15, 28, 26, 25]. While there are many similarities between the modalities, there are two key differences: 1) the point cloud is a sparse representation, while an image is dense and 2) the point cloud is 3D, while the image is 2D. As a result object detection from point clouds does not trivially lend itself to standard image convolutional pipelines.

Some early works focus on either using 3D convolu-

1

PointPillars



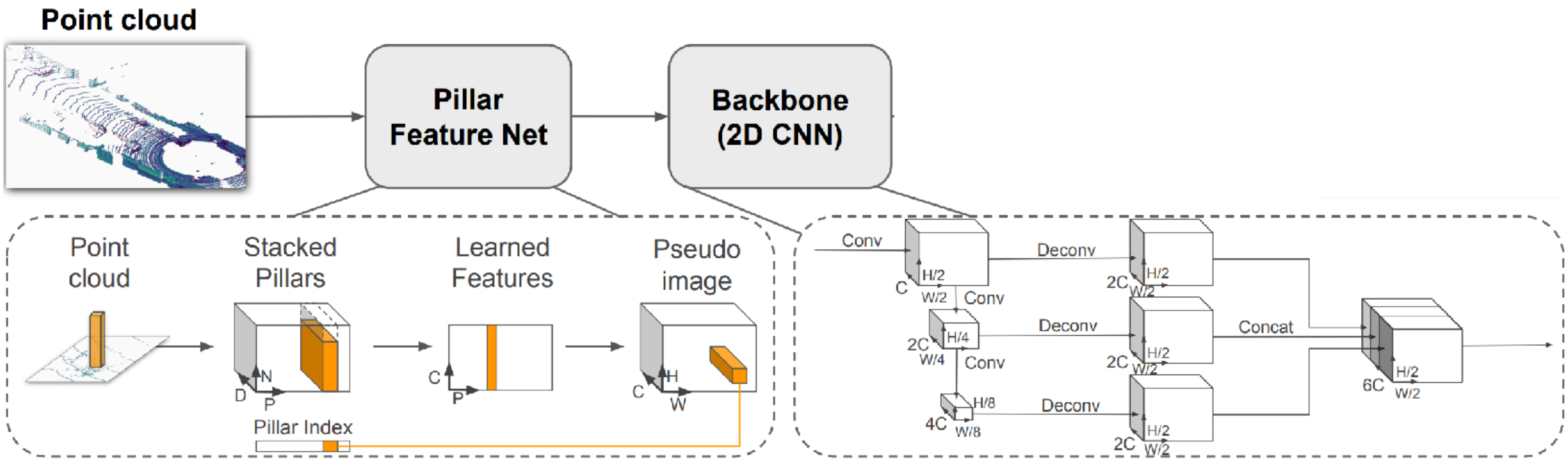
1. Pillar encoding:
B x P x N x D -> PointNet -> B x P x C

2. Scatter the dense features to the top-down view:

Indices: P x 3 (for H, W and B)
Dense pillar features: B x P x C
-> B x H x W x C

Handling sparsity in the top-down image (typically, >90% of the pixels are empty):
B: batch size.
P: the maximum number of non-empty pillars per sample (the buffer size).
N: the maximum number of points to keep per pillar (the buffer size).
D: point dimension/number of channels.

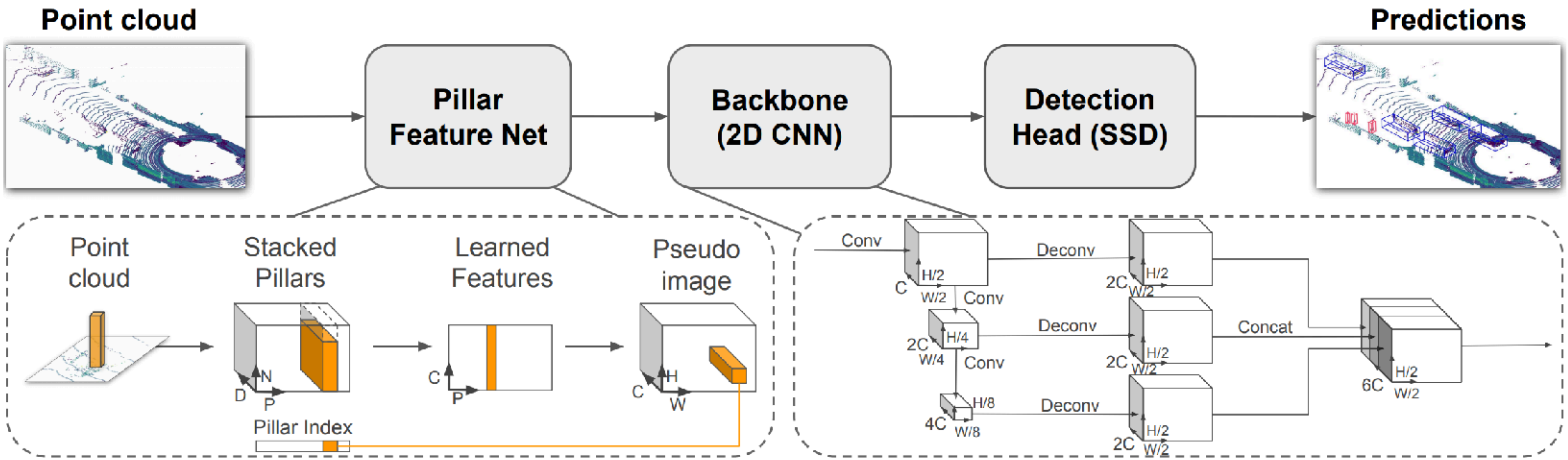
PointPillars



SSD-style backbone 2D CNN

SSD: Single-Stage Detector by Wei Liu et al.

PointPillars



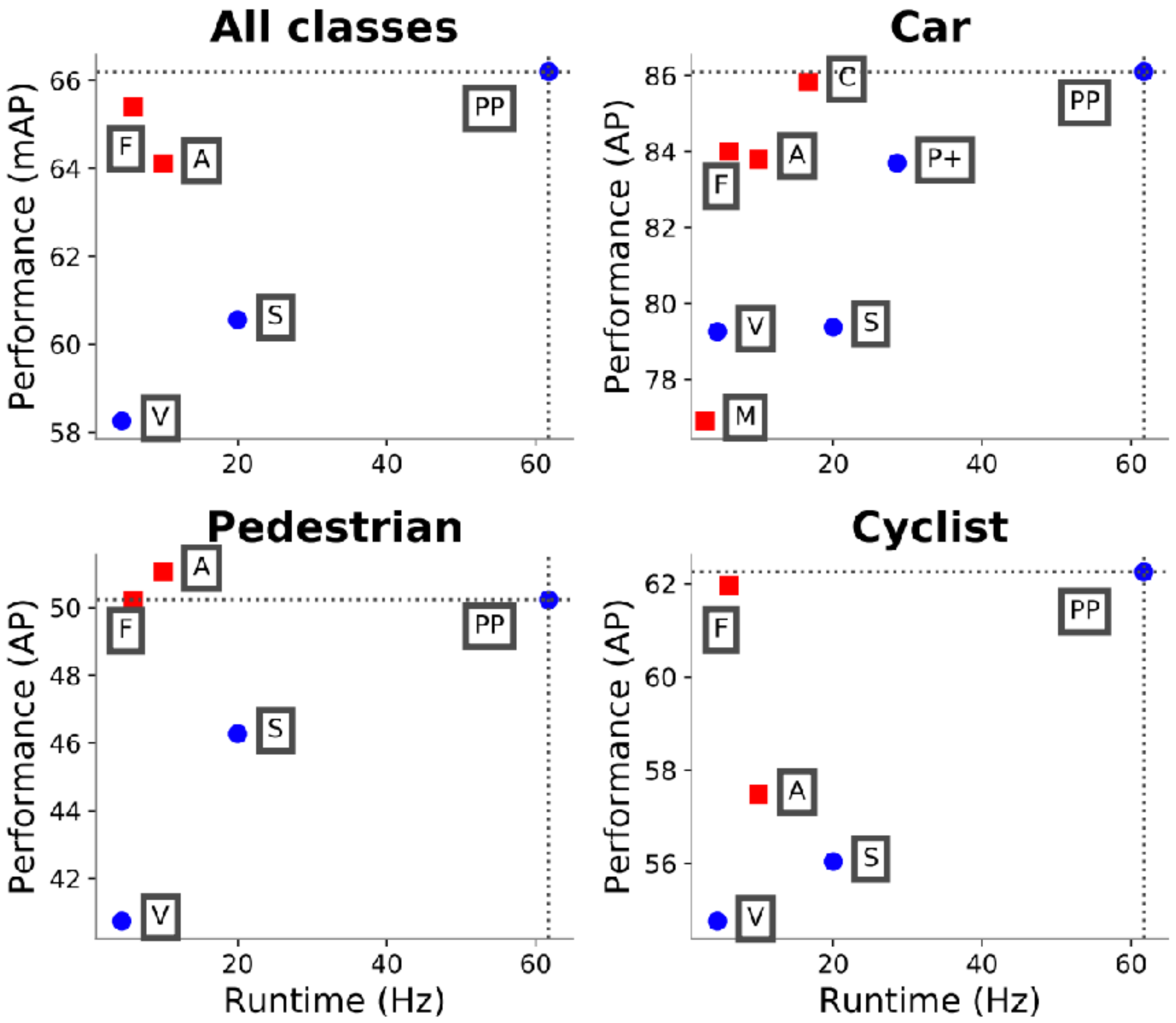
PointPillars

The biggest advantages:

- Inference speed.
- Simplicity.

Weaknesses:

- Assumption of a projection plane (not generalizable to more complex 3d scenes).
- Aggressive compression of the dimension.



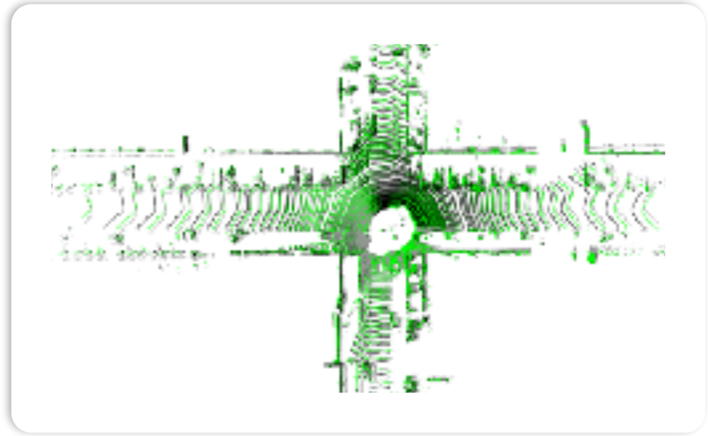
The deep learning era of 3d object detection

Image-driven

Monocular view detectors
Frustum-based detectors

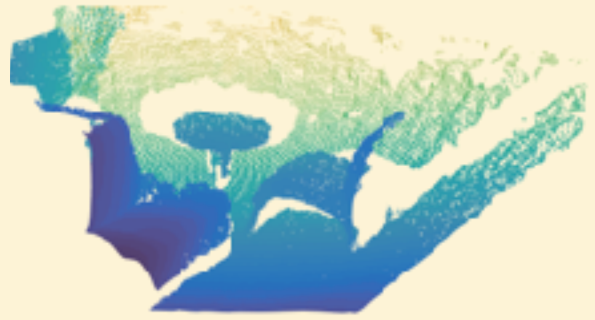
Dimension reduction

Bird's eye view detectors



Leveraging Sparsity in 3D

Point set deep nets
Sparse 3D conv, GNNs



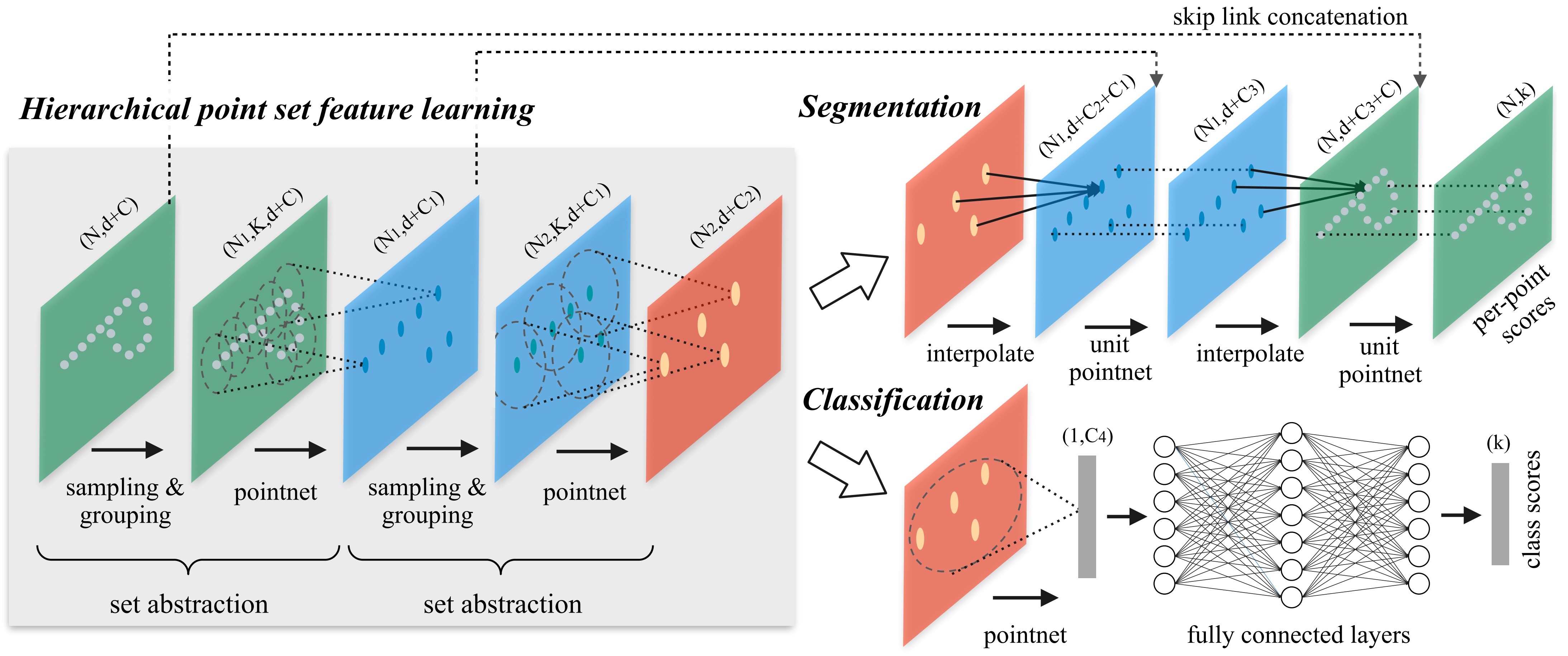
E.g.: **Frustum PointNets [6]** | **PointPillars [7]** | **VoteNet [8]**

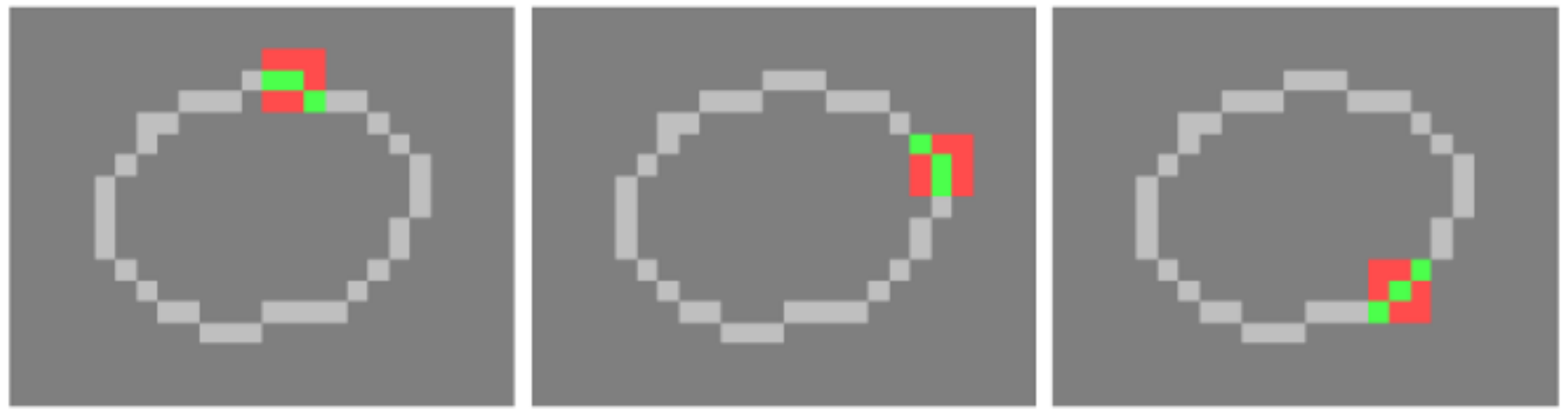
Point cloud based 3D object detectors

- Key idea: Use sparsity aware backbone architectures (e.g. PointNet++, Sparse 3D convnet) and design 3D detection frameworks that leverage sparsity.

Deep Hough Voting for 3D Object Detection in Point Clouds (2019) [8]
PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud (2019) [15]
STD: Sparse-to-Dense 3D Object Detector for Point Cloud (2019) [16]
3DSSD: Point-based 3D Single Stage Object Detector (2020) [17]
Pv-rcnn: Point-voxel feature set abstraction for 3d object detection (2020) [18]

PointNet++





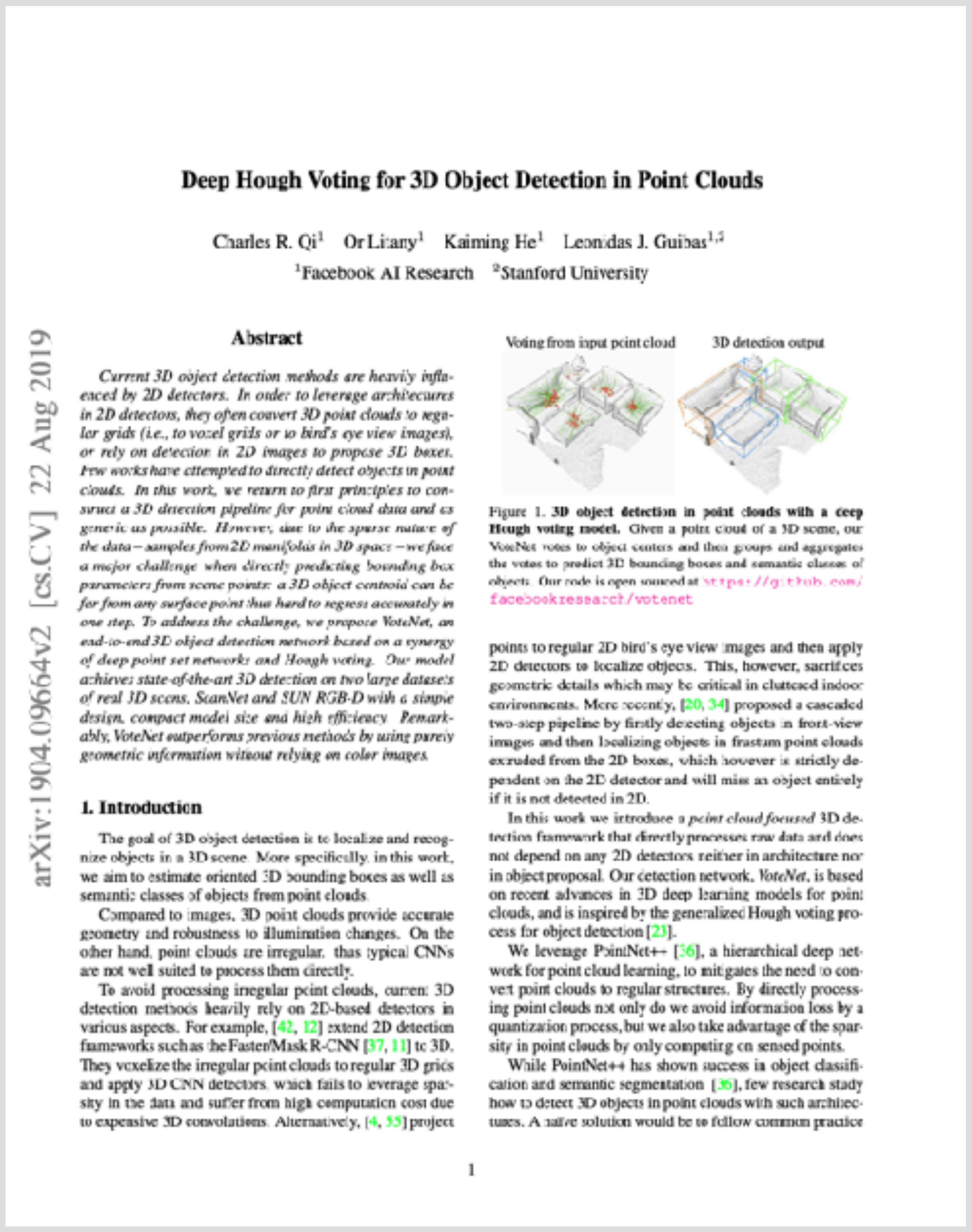
Sub manifold sparse 3d conv

Deep Hough Voting for 3D Object Detection in Point Clouds

Charles R. Qi, Or Litany, Kaiming He, Leonidas Guibas.

ICCV 2019

Best Paper Nominee



arXiv:1904.09664v2 [cs.CV] 22 Aug 2019

Deep Hough Voting for 3D Object Detection in Point Clouds

Charles R. Qi[1] Or Litany[1] Kaiming He[1] Leonidas J. Guibas[1,2]
[1]Facebook AI Research [2]Stanford University

Abstract

Current 3D object detection methods are heavily influenced by 2D detectors. In order to leverage architectures in 2D detectors, they often convert 3D point clouds to regular grids (i.e., to voxel grids or to bird's eye view images), or rely on detection in 2D images to propose 3D boxes. Few works have attempted to directly detect objects in point clouds. In this work, we return to first principles to construct a 3D detection pipeline for point cloud data and as generic as possible. However, due to the sparse nature of the data – samples from 2D manifolds in 3D space – we face a major challenge when directly predicting bounding box parameters from scene points: a 3D object centroid can be far from any surface point thus hard to regress accurately in one step. To address the challenge, we propose VoteNet, an end-to-end 3D object detection network based on a synergy of deep point set networks and Hough voting. Our model achieves state-of-the-art 3D detection on two large datasets of real 3D scans, ScanNet and SUN RGB-D with a simple design, compact model size and high efficiency. Remarkably, VoteNet outperforms previous methods by using purely geometric information without relying on color images.

1. Introduction

The goal of 3D object detection is to localize and recognize objects in a 3D scene. More specifically, in this work, we aim to estimate oriented 3D bounding boxes as well as semantic classes of objects from point clouds.

Compared to images, 3D point clouds provide accurate geometry and robustness to illumination changes. On the other hand, point clouds are irregular. thus typical CNNs are not well suited to process them directly.

To avoid processing irregular point clouds, current 3D detection methods heavily rely on 2D-based detectors in various aspects. For example, [42, 12] extend 2D detection frameworks such as the Faster/Mask R-CNN [37, 11] to 3D. They voxelize the irregular point clouds to regular 3D grids and apply 3D CNN detectors, which fails to leverage sparsity in the data and suffer from high computation cost due to expensive 3D convolutions. Alternatively, [4, 55] project

Figure 1. **3D object detection in point clouds with a deep Hough voting model.** Given a point cloud of a 3D scene, our VoteNet votes to object centers and then groups and aggregates the votes to predict 3D bounding boxes and semantic classes of objects. Our code is open sourced at https://github.com/facebookresearch/votenet

points to regular 2D bird's eye view images and then apply 2D detectors to localize objects. This, however, sacrifices geometric details which may be critical in cluttered indoor environments. More recently, [20, 34] proposed a cascaded two-step pipeline by firstly detecting objects in front-view images and then localizing objects in frustum point clouds extruded from the 2D boxes, which however is strictly dependent on the 2D detector and will miss an object entirely if it is not detected in 2D.

In this work we introduce a *point cloud focused* 3D detection framework that directly processes raw data and does not depend on any 2D detectors neither in architecture nor in object proposal. Our detection network, *VoteNet*, is based on recent advances in 3D deep learning models for point clouds, and is inspired by the generalized Hough voting process for object detection [23].

We leverage PointNet++ [36], a hierarchical deep network for point cloud learning, to mitigates the need to convert point clouds to regular structures. By directly processing point clouds not only do we avoid information loss by a quantization process, but we also take advantage of the sparsity in point clouds by only computing on sensed points.

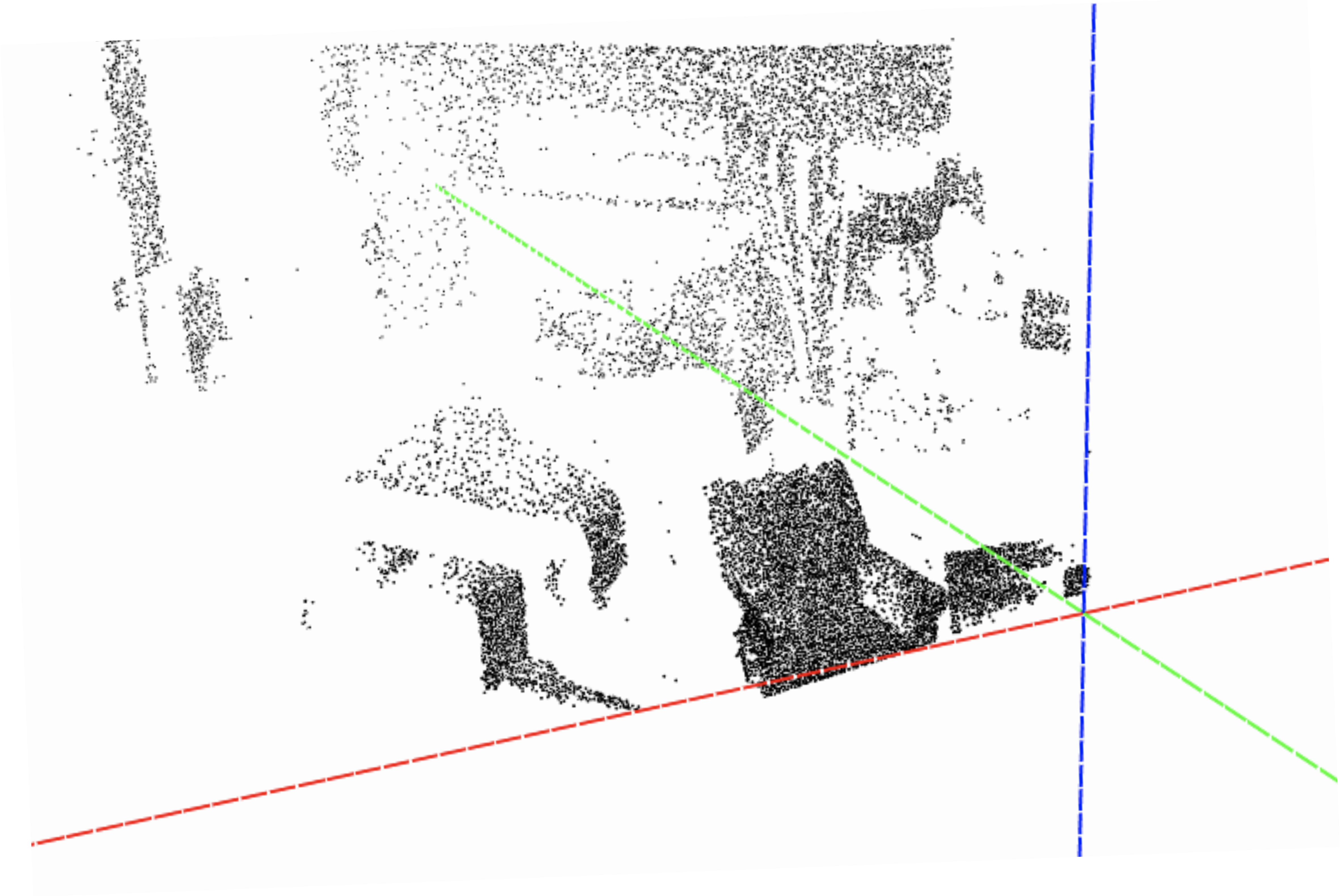
While PointNet++ has shown success in object classification and semantic segmentation [36], few research study how to detect 3D objects in point clouds with such architectures. A naive solution would be to follow common practice

1

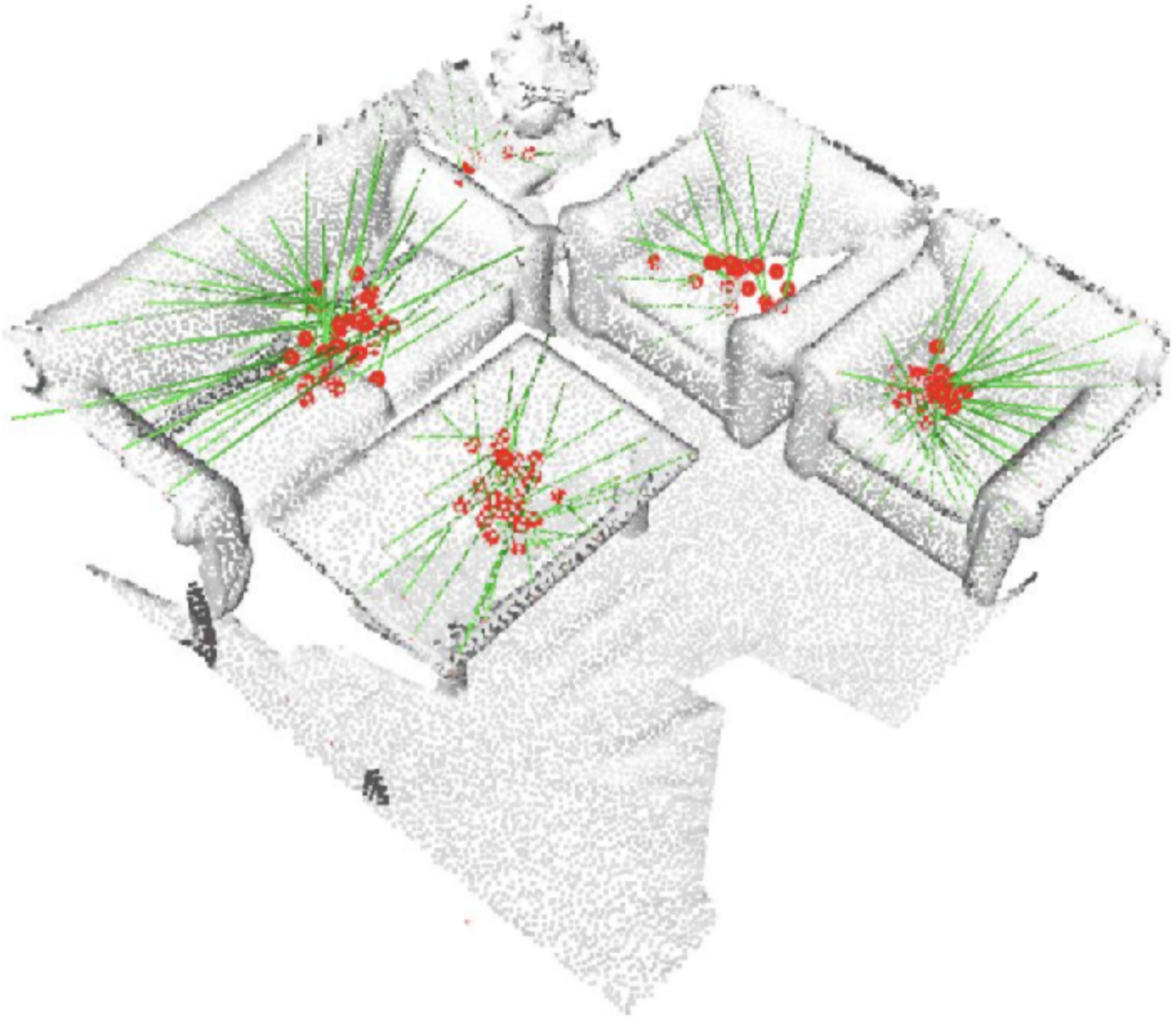
Observation: 2D v.s. 3D

Dense 2D pixel array

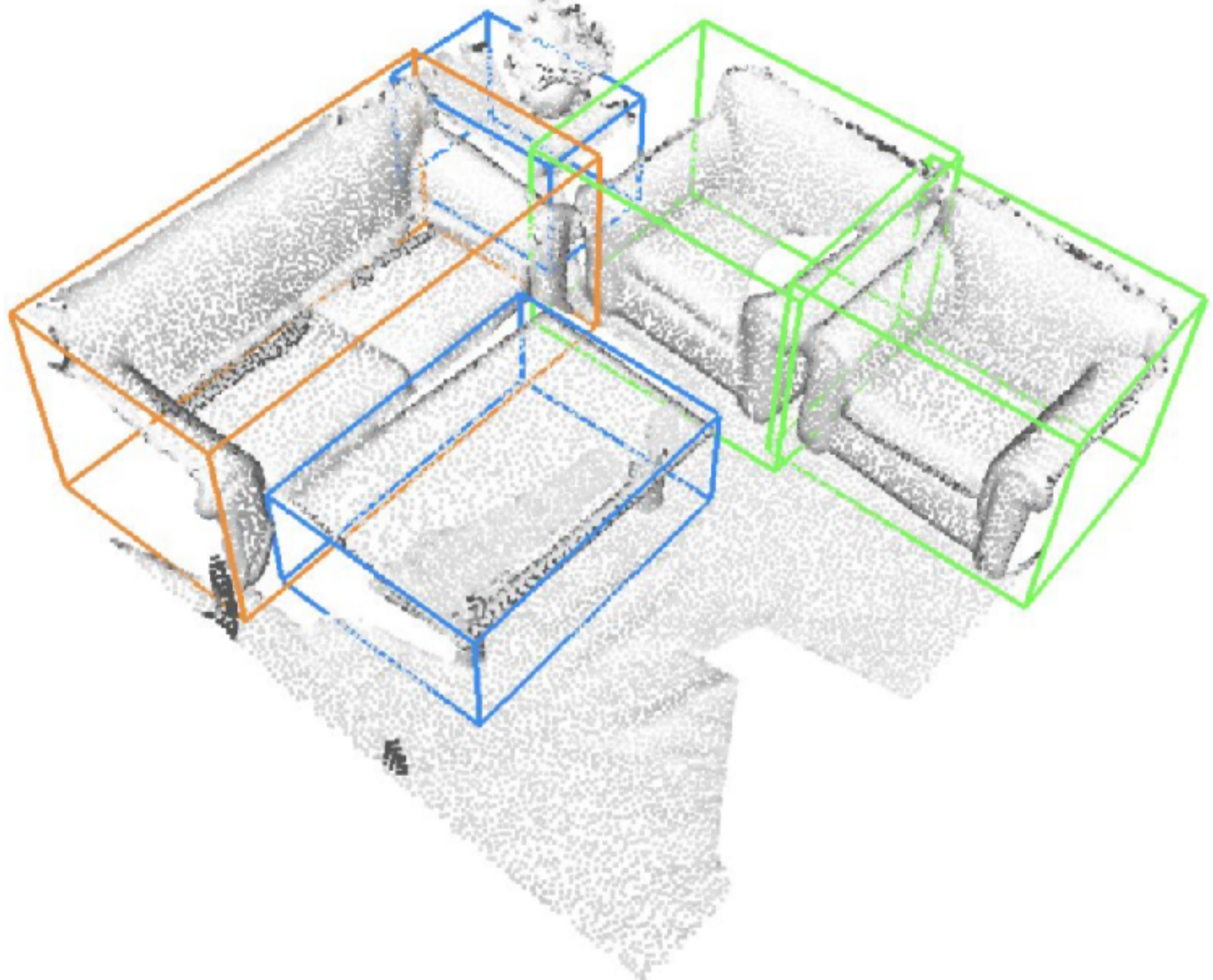
Sparse 3D points
(only on object surfaces)



Our solution: Voting



Voting from surface points



Detected 3D bounding boxes

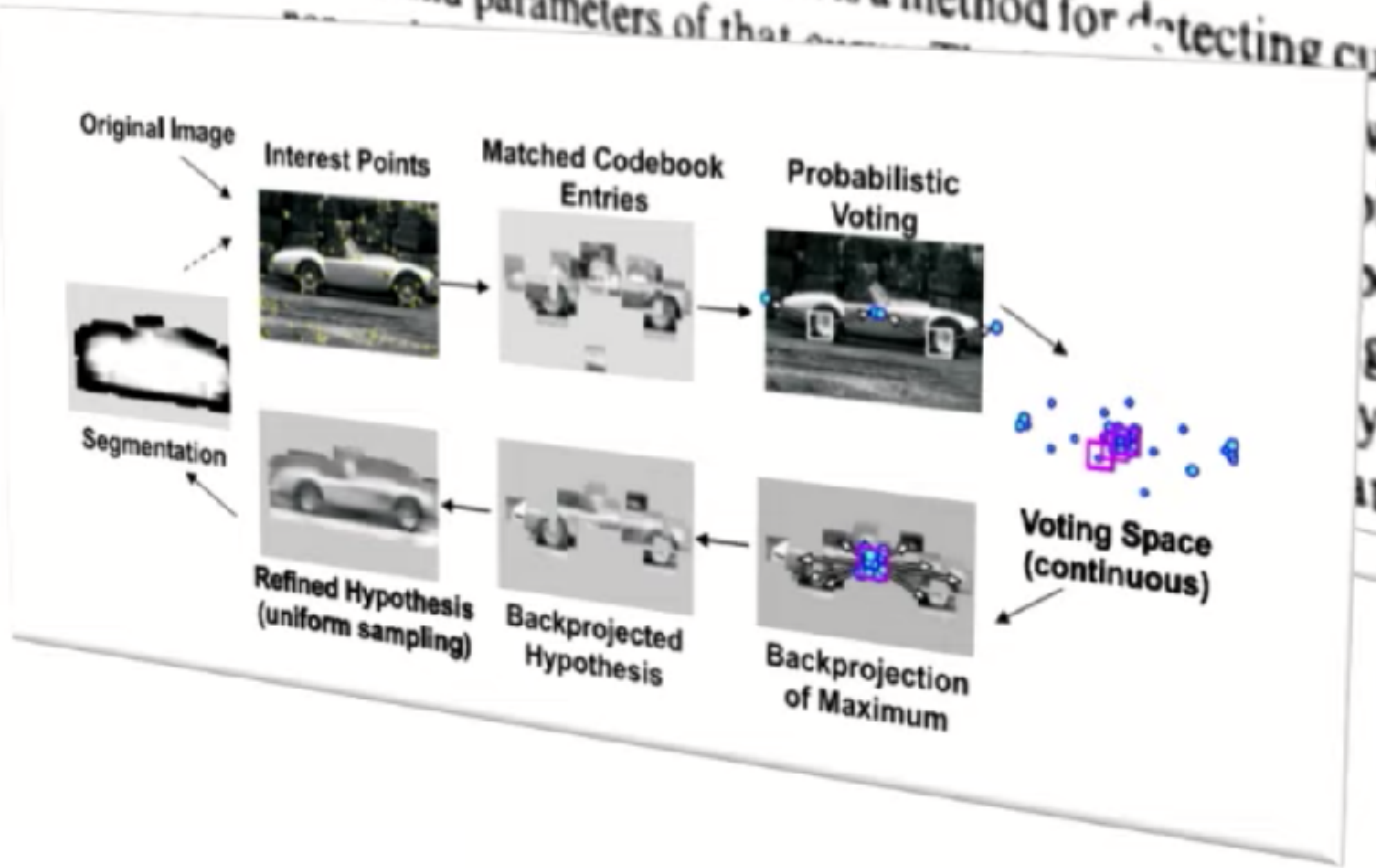
GENERALIZING THE HOUGH TRANSFORM TO DETECT ARBITRARY SHAPES*

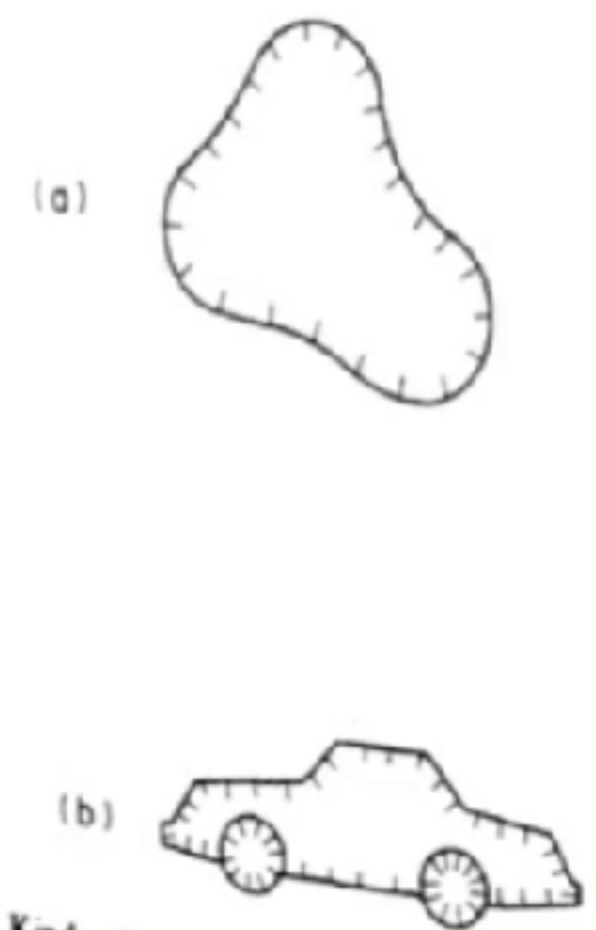
D. H. BALLARD
Computer Science Department, University of Rochester, Rochester, NY 14627, U.S.A.

(*Received 10 October 1979; in revised form 9 September 1980; received for publication 23 September 1980*)

Abstract—The Hough transform is a method for detecting curves by
a curve and parameters of that curve.

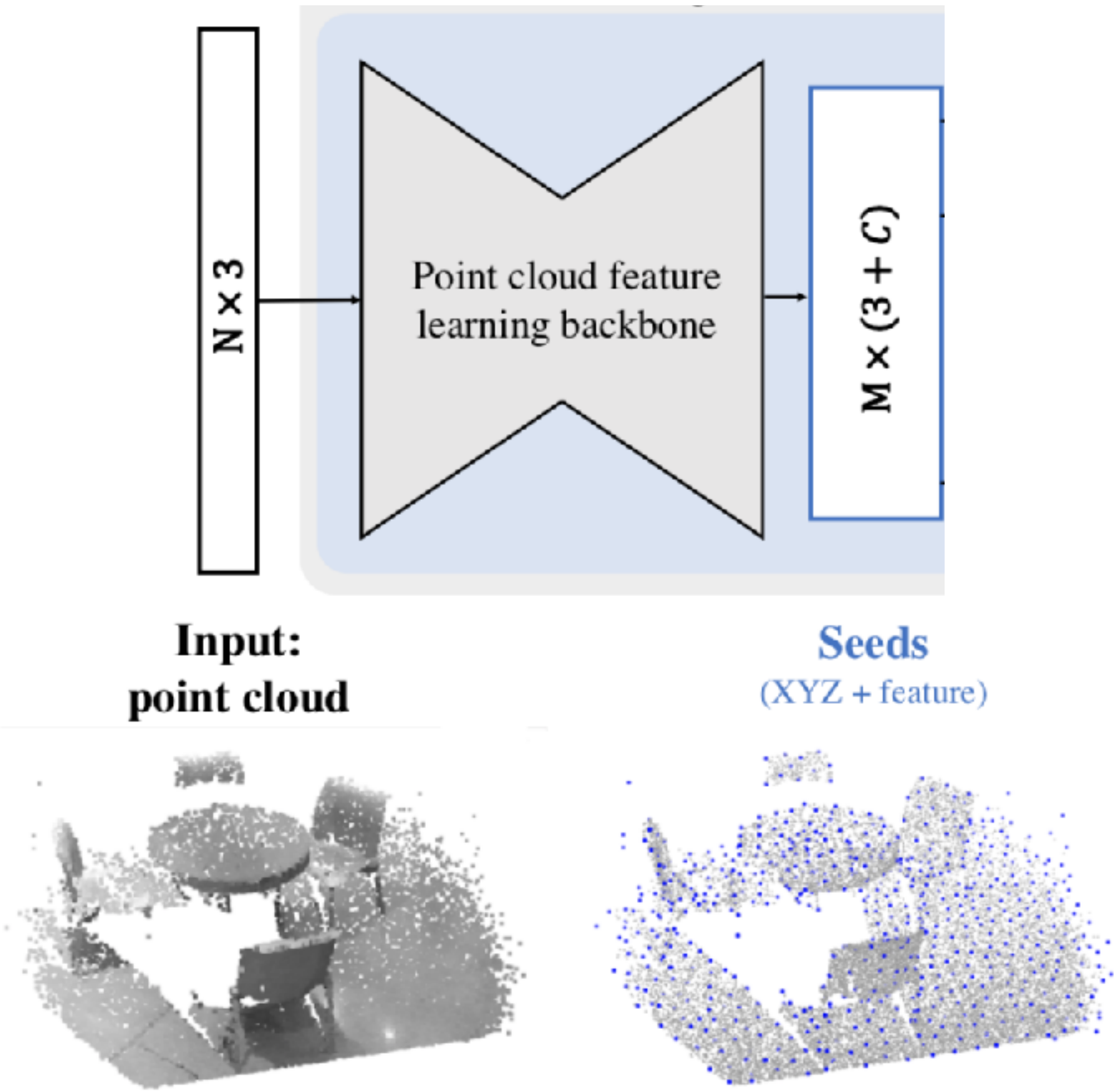
wed how ... points on
binary edg ... s[1,2] and
pecifically ... ralized to
geniously ... as.[6] The
ytic shap ... ons.[7,8,9]
apping ca ... mapping
... es of that



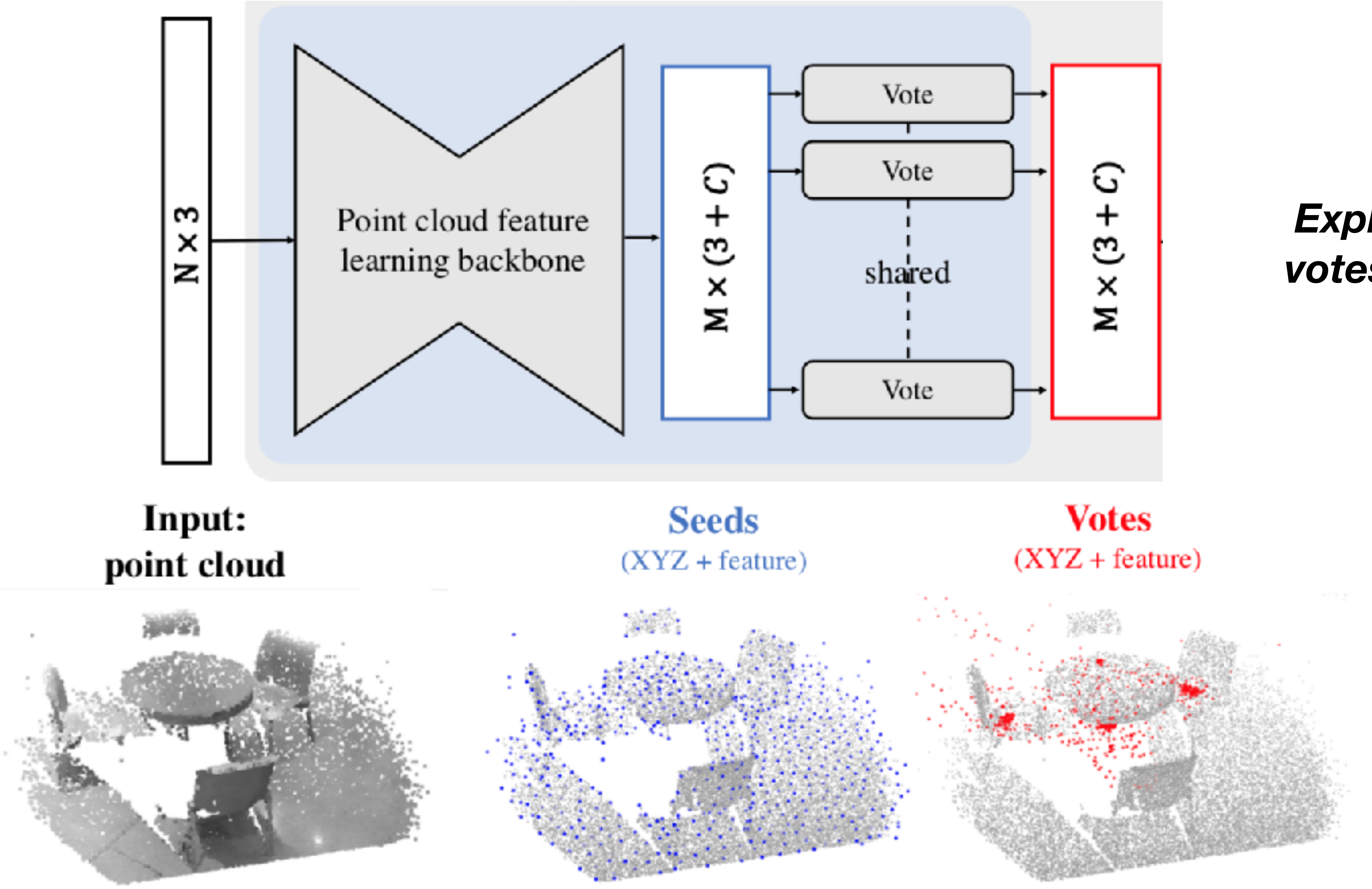


ig. 1. Kinds of shapes detected with generalized Hough transform. (a) Simple shape; (b) composite shape.

Deep Hough voting: Detection pipeline

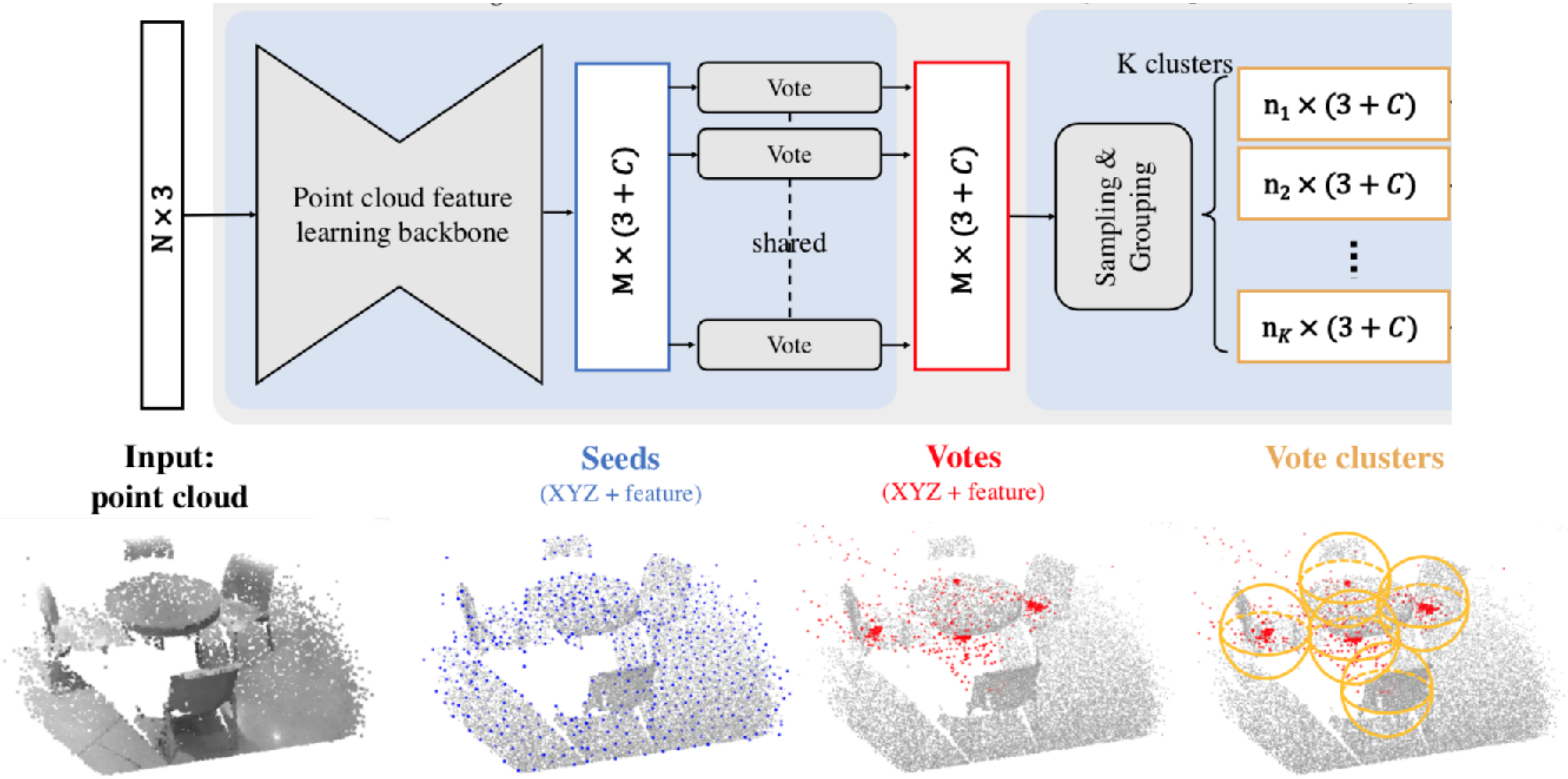


Deep Hough voting: Detection pipeline

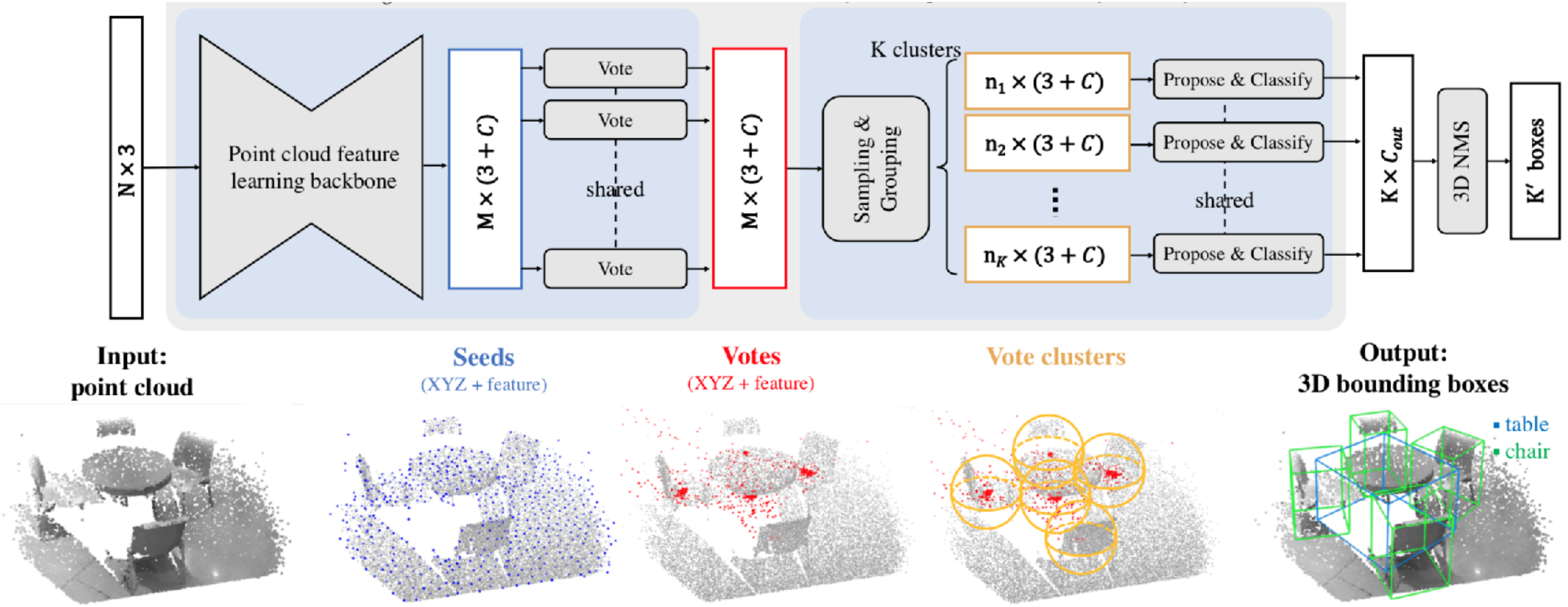


Explicit supervision for the votes (the XYZ translations)

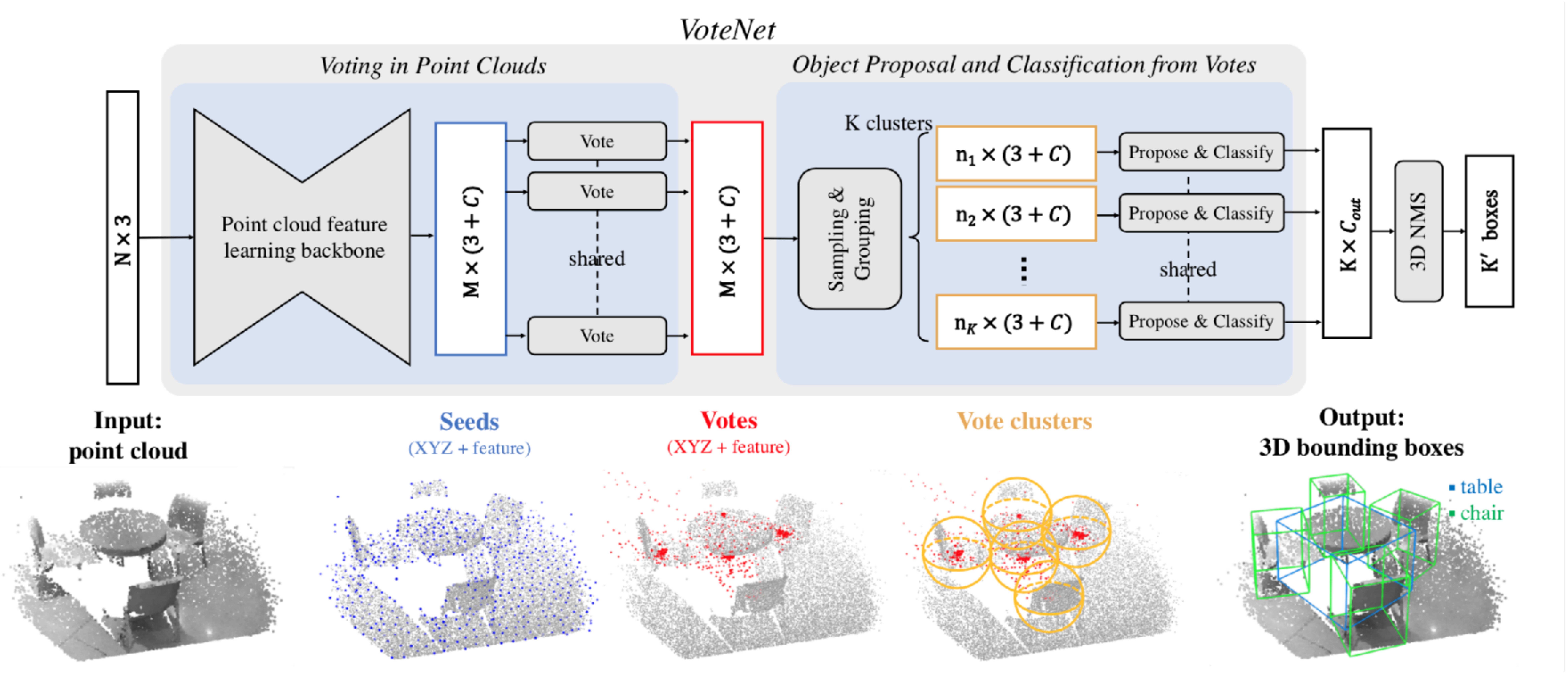
Deep Hough voting: Detection pipeline



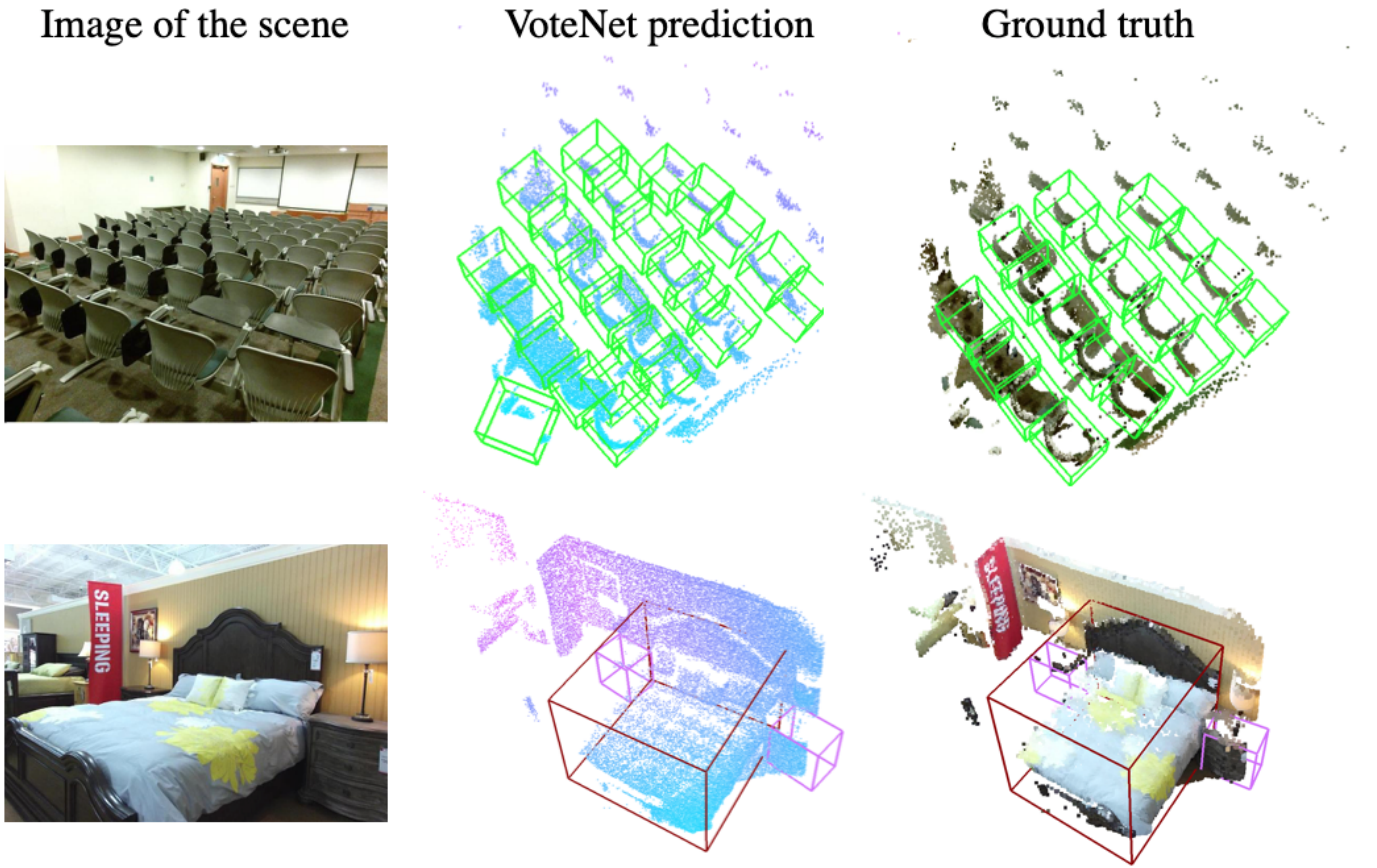
Deep Hough voting: Detection pipeline



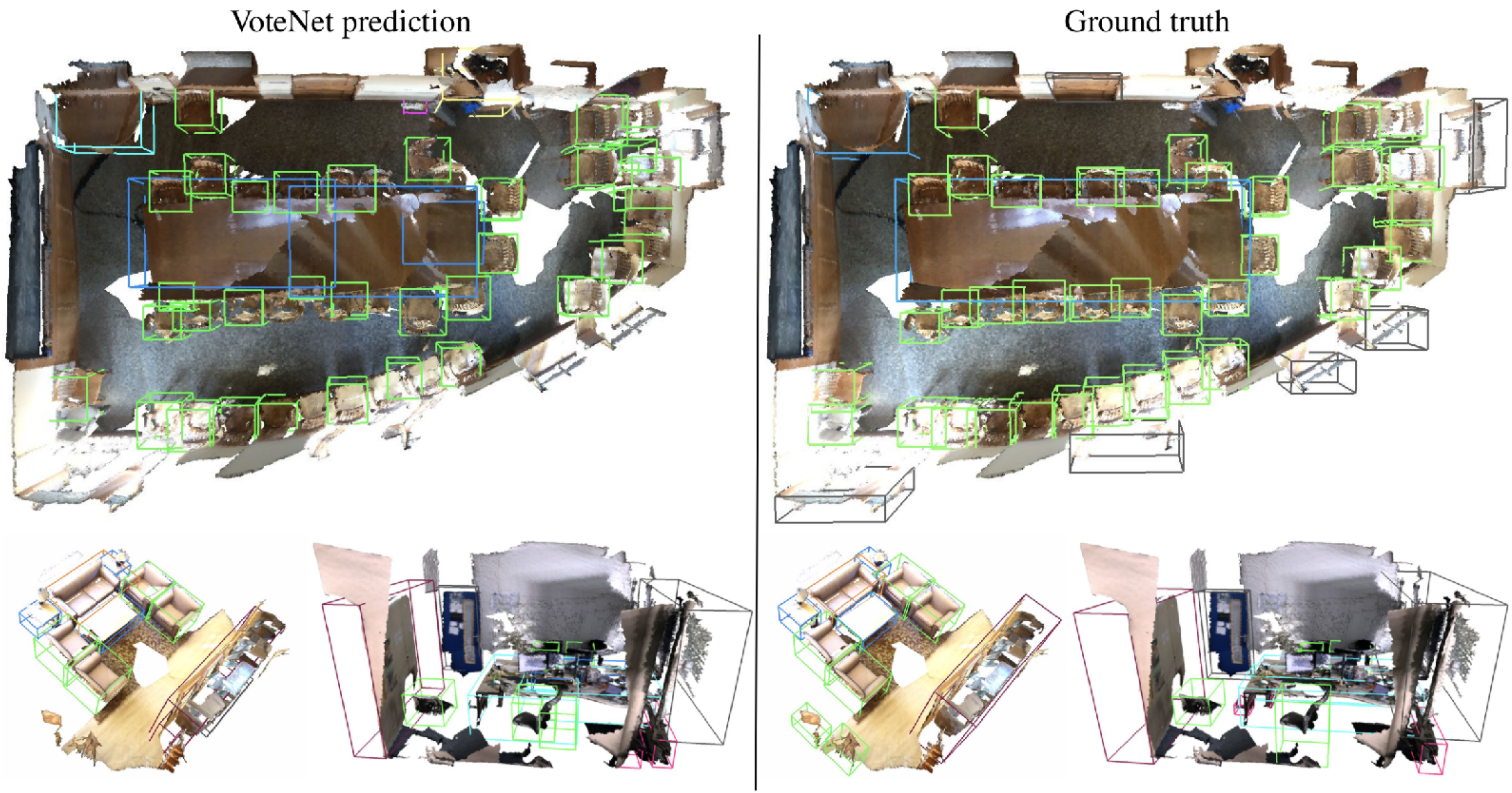
Deep Hough voting: Detection pipeline



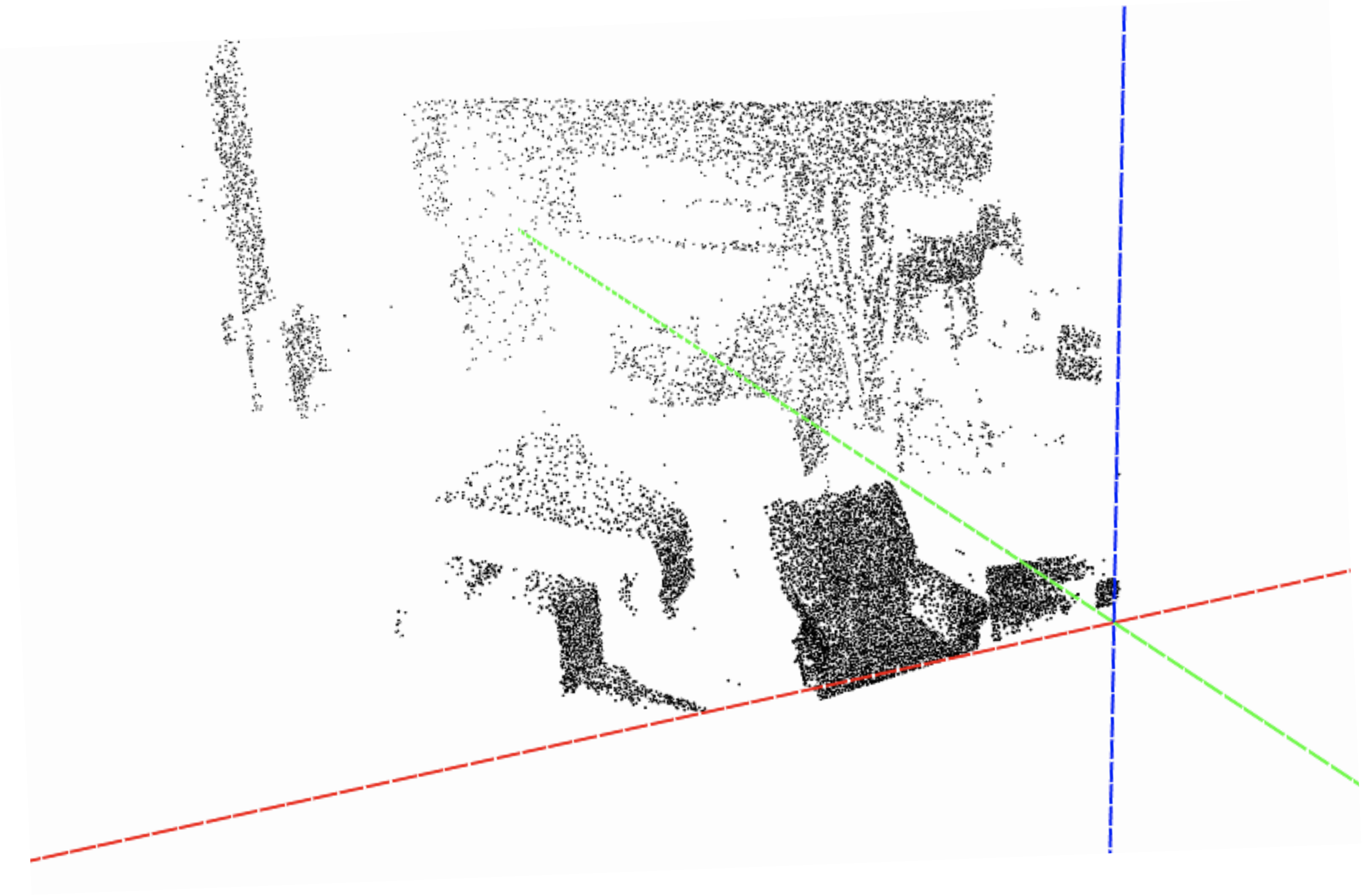
Results: SUN RGB-D (single depth images)



Results: ScanNet (3D reconstructions)



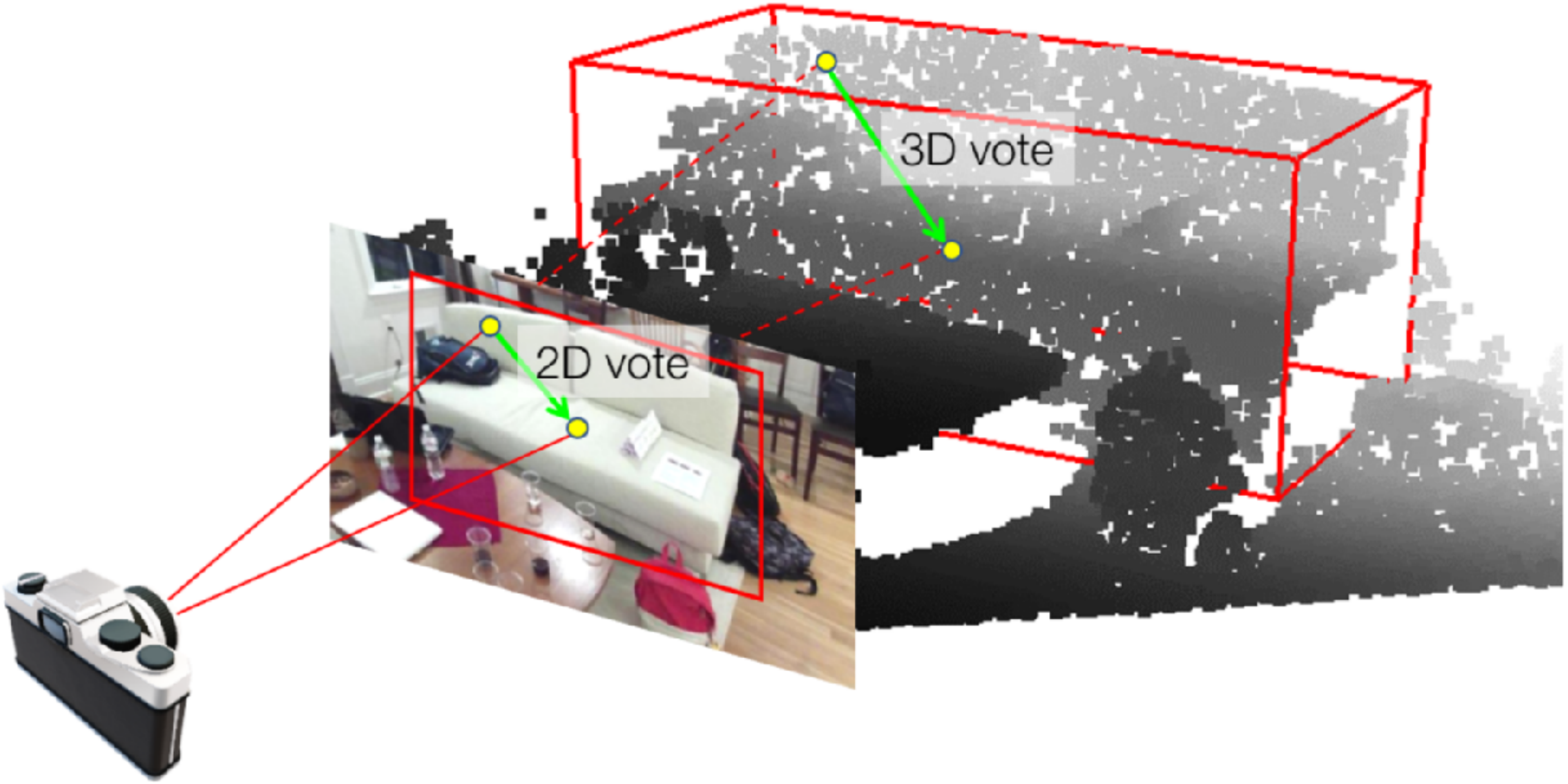
Can images help the VoteNet detection?



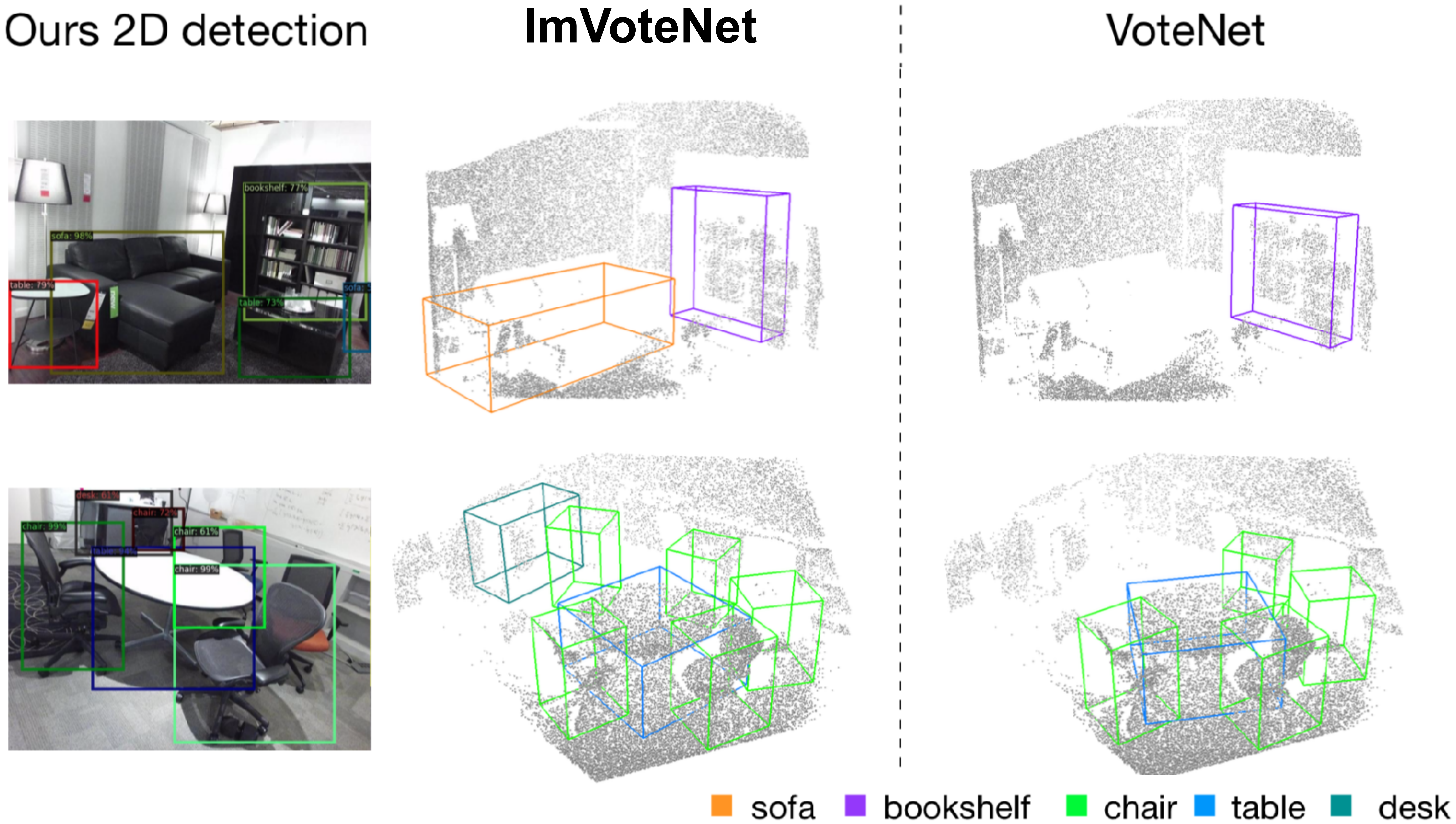
*Images are in **high resolution**, have **rich texture**, and can even provide useful geometric cues for object localization & shape/pose estimation.*

ImVoteNet: Boosting 3D Object Detection in Point Clouds with Image Votes [19]

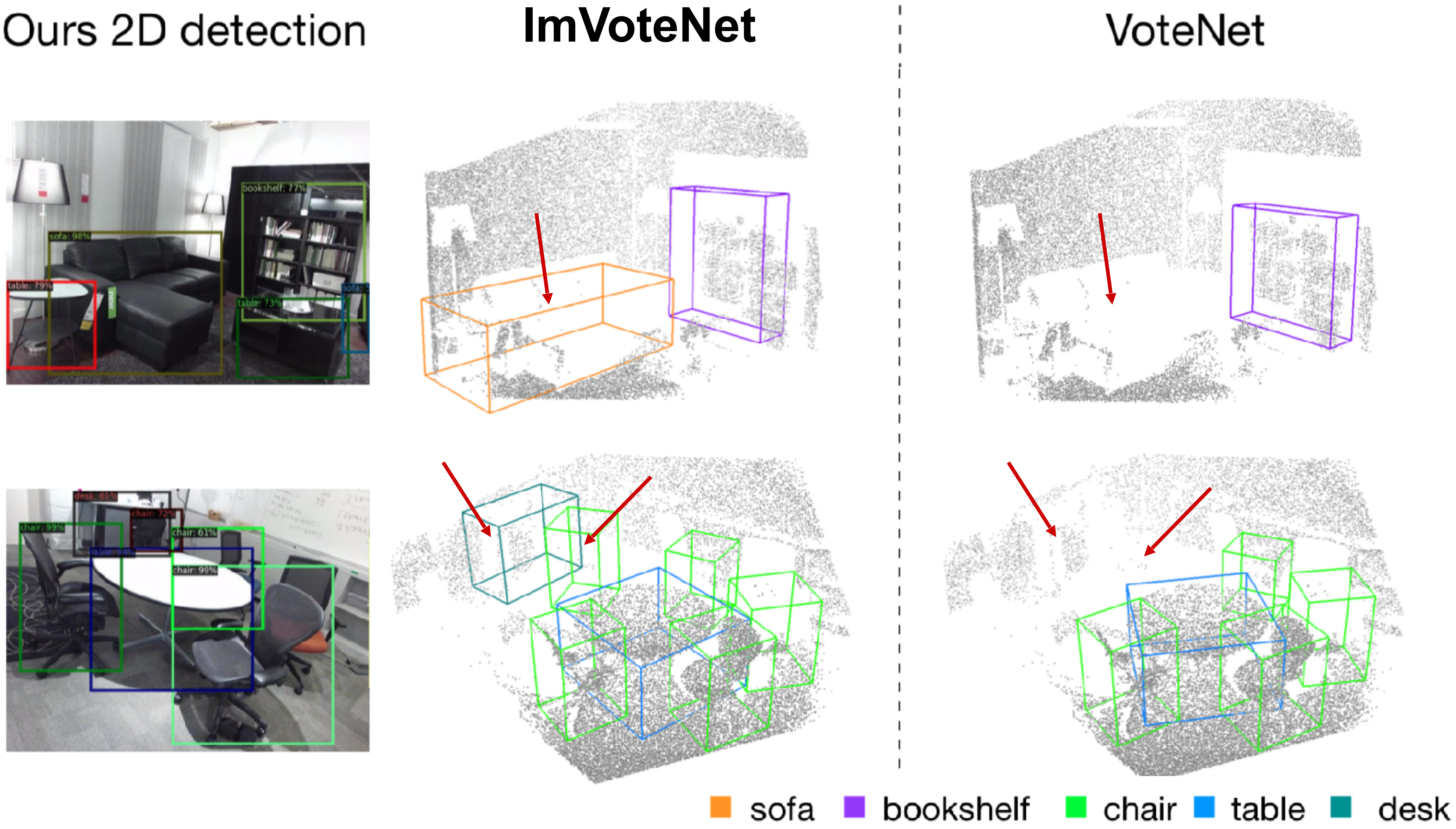
Charles R. Qi*, Xinlei Chen*, Or Litany, Leonidas Guibas. CVPR 2020.



Results on SUN RGB-D



Results on SUN RGB-D



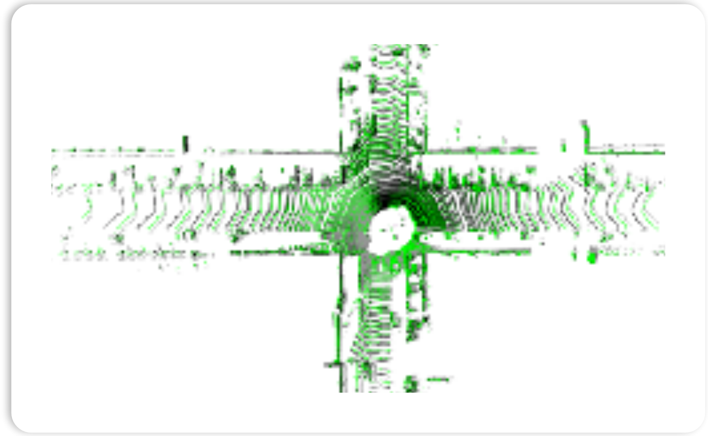
The deep learning era of 3d object detection

Image-driven

Monocular view detectors
Frustum-based detectors

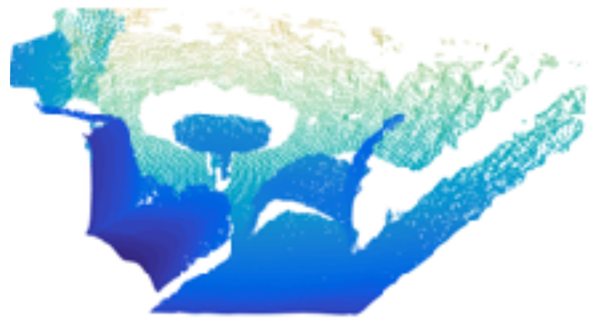
Dimension reduction

Bird's eye view detectors



Leveraging Sparsity in 3D

Point set deep nets
Sparse 3D conv, GNNs



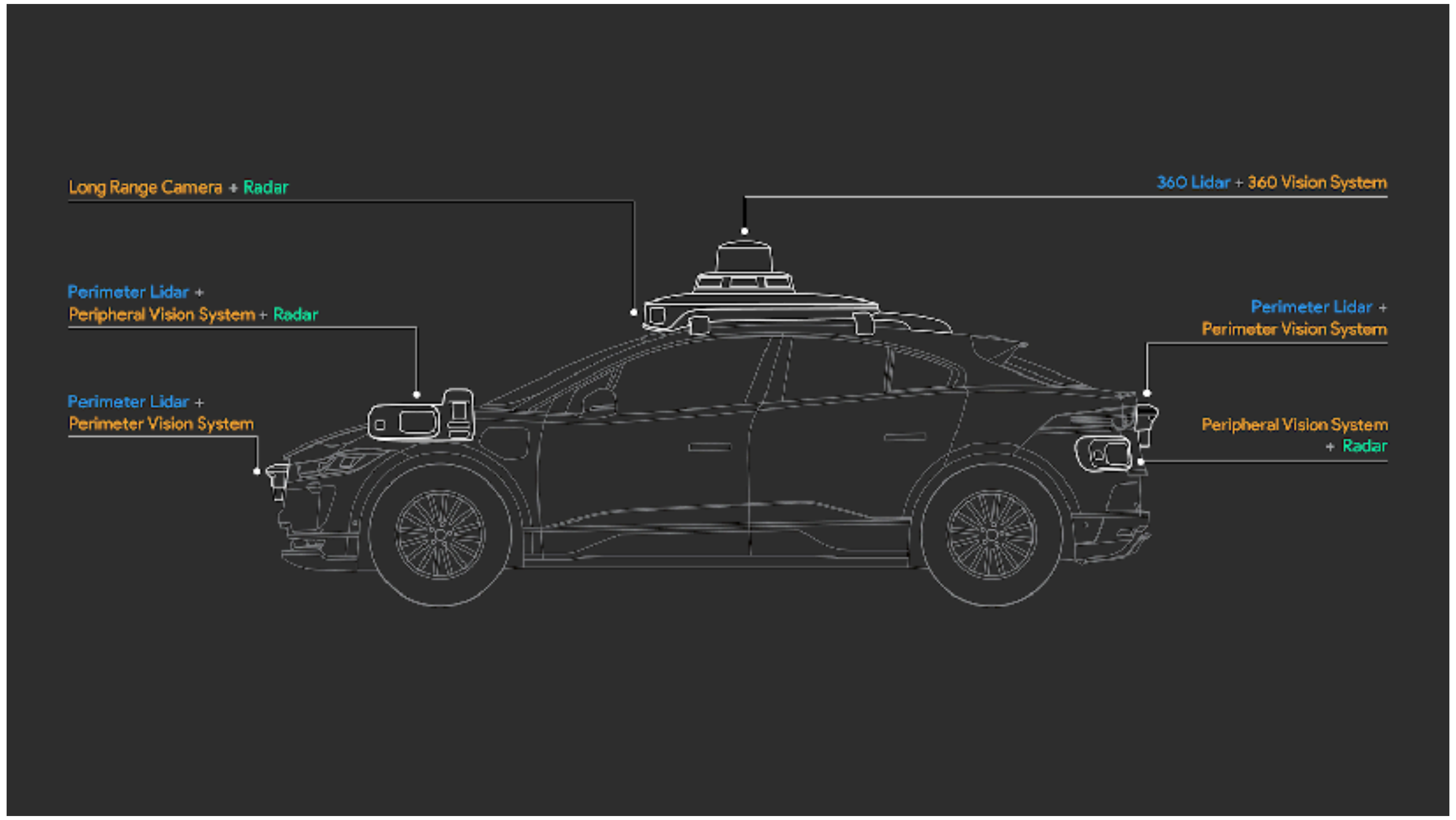
E.g.: **Frustum PointNets [6]** **PointPillars [7]** **VoteNet [8]**

The future of 3D object detection

Input:

Multi-modal input (multi-camera RGB images, Lidar point clouds/depth images, SLAM/SfM point clouds, radar, audio etc.)

Temporal input i.e. sequences.



Source: Waymo (5th generation Waymo driver)

The future of 3D object detection

Machine learning:

Semi-supervised learning
Self-supervised learning
Weakly-supervised learning
Multi-task learning
Adversarial learning
Domain adaptation
Life-long learning
…

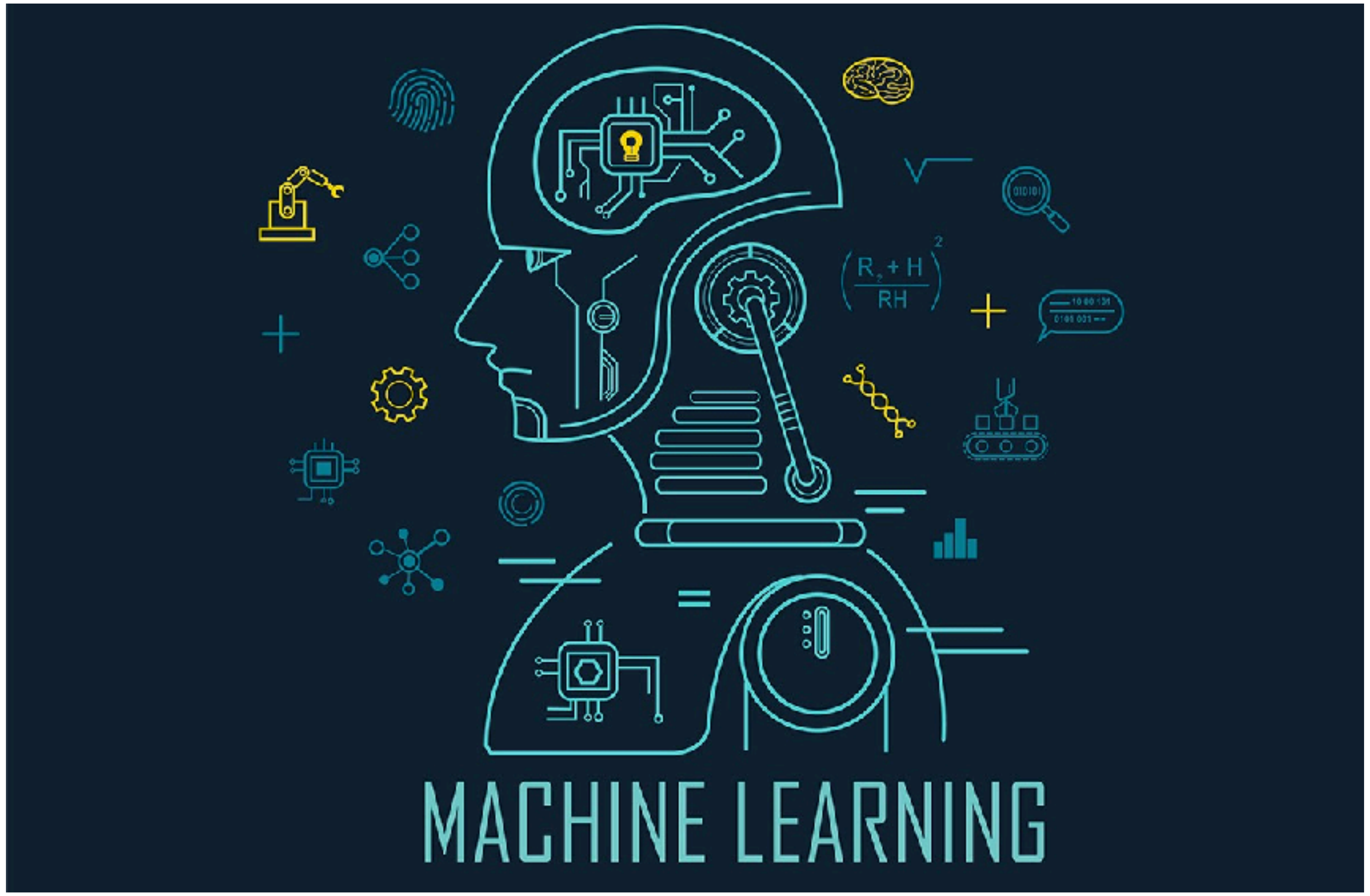


Fig from Anmol Behl

The future of 3D object detection

Robotics:

3D instance detection
6D pose estimation
Template based detection
Few-shot detection
…

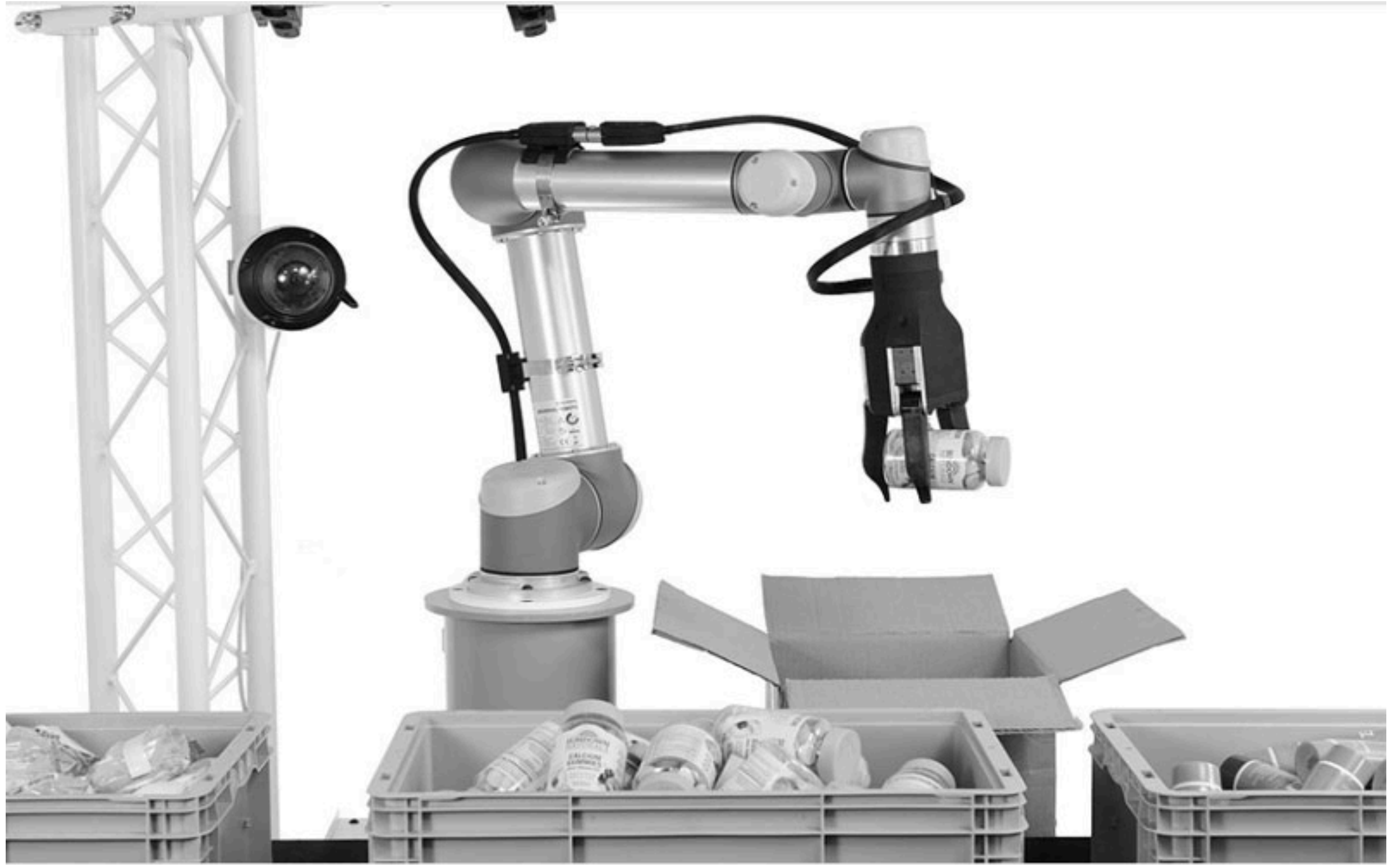


Fig from Frank Tobe

The future of 3D object detection

Continue to push the boundaries

Occluded cases
Long-tail categories
Extreme conditions (no 3d data, bad weather etc.)
…

Source: https://depositphotos.com/vector-images/mountain-climber.html

Summary

- Motivation: A.I. applications in the physical world —> 3D object recognition.
- The history and recent progresses of 3D object detection algorithms.
- Deep dive into three specific 3D object detectors:
  - Frustum PointNets, PointPillar and VoteNet.
- Future research directions of 3D object detection.

Thank you for listening! Q&A time

References

[1] Object recognition in 3D scenes with occlusions and clutter by Hough voting Fourth Pacific-Rim Symposium on Image and Video Technology by Federico et al. IEEE, 2010.
[2] Tutorial: Point Cloud Library: Three-Dimensional Object Recognition and 6 DOF Pose Estimation by Aldoma et al. IEEE Robotics & Automation Magazine 19.3 (2012): 80-91.
[3] Object discovery in 3d scenes via shape analysis by Karpathy et al. *IEEE International Conference on Robotics and Automation*. IEEE, 2013.
[4] Sliding shapes for 3d object detection in depth images by Song et al. *European conference on computer vision*. Springer, Cham, 2014.
[5] Deep sliding shapes for amodal 3d object detection in rgb-d images by Song et al. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.
[6] Frustum pointnets for 3d object detection from rgb-d data by Qi et al. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
[7] Pointpillars: Fast encoders for object detection from point clouds by Lang et al. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
[8] Deep hough voting for 3d object detection in point clouds by Qi et al. *Proceedings of the IEEE International Conference on Computer Vision*. 2019.
[9] 3d bounding box estimation using deep learning and geometry by Mousavian et al. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
[10] Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving by Wang et al. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
[11] Objects as points by Wang et al. 2019.
[12] Volumetric and Multi-View CNNs for Object Classification on 3D Data by Qi et al. CVPR 2016.
[13] Multi-view 3d object detection network for autonomous driving by Chen et al. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
[14] Voxelnet: End-to-end learning for point cloud based 3d object detection by Zhou et al. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
[15] Pointrcnn: 3d object proposal generation and detection from point cloud by Shi et al. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
[16] Std: Sparse-to-dense 3d object detector for point cloud by Yang et al. Proceedings of the IEEE International Conference on Computer Vision. 2019.
[17] 3dssd: Point-based 3d single stage object detector by Yang et al. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.
[18] Pv-rcnn: Point-voxel feature set abstraction for 3d object detection by Shi et al. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
[19] Imvotenet: Boosting 3d object detection in point clouds with image votes by Qi et al. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.