

# Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review

Yaodong Cui, *Student Member, IEEE*, Ren Chen, Wenbo Chu, Long Chen<sup>ID</sup>, *Senior Member, IEEE*, Daxin Tian<sup>ID</sup>, *Senior Member, IEEE*, Ying Li<sup>ID</sup>, *Graduate Student Member, IEEE*, and Dongpu Cao<sup>ID</sup>, *Member, IEEE*

**Abstract**—Autonomous vehicles were experiencing rapid development in the past few years. However, achieving full autonomy is not a trivial task, due to the nature of the complex and dynamic driving environment. Therefore, autonomous vehicles are equipped with a suite of different sensors to ensure robust, accurate environmental perception. In particular, the camera-LiDAR fusion is becoming an emerging research theme. However, so far there has been no critical review that focuses on deep-learning-based camera-LiDAR fusion methods. To bridge this gap and motivate future research, this article devotes to review recent deep-learning-based data fusion approaches that leverage both image and point cloud. This review gives a brief overview of deep learning on image and point cloud data processing. Followed by in-depth reviews of camera-LiDAR fusion methods in depth completion, object detection, semantic segmentation, tracking and online cross-sensor calibration, which are organized based on their respective fusion levels. Furthermore, we compare these methods on publicly available datasets. Finally, we identified gaps and over-looked challenges between current academic researches and real-world applications. Based on these observations, we provide our insights and point out promising research directions.

**Index Terms**—Camera-LiDAR fusion, sensor fusion, depth completion, object detection, semantic segmentation, tracking, deep learning.

## I. INTRODUCTION

RECENT breakthroughs in deep learning and sensor technologies have motivated rapid development of autonomous driving technology, which could potentially

Manuscript received April 9, 2020; revised August 27, 2020; accepted September 8, 2020. The Associate Editor for this article was S.-H. Kong. (*Corresponding author: Dongpu Cao*)

Yaodong Cui, Ren Chen, Ying Li, and Dongpu Cao are with the Waterloo Cognitive Autonomous Driving (CogDrive) Laboratory, Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: yaodong.cui@uwaterloo.ca; andrews.rchen@gmail.com; y2424li@uwaterloo.ca; dongpu@uwaterloo.ca).

Wenbo Chu is with China Intelligent and Connected Vehicles (Beijing) Research Institute Company, Ltd., Beijing 100176, China (e-mail: chuwenbo@china-icv.cn).

Long Chen is with the School of Data and Computer Science, Sun Yat-sen University, Zhuhai 519082, China, and also with Waytous Inc., Qingdao 266109, China (e-mail: chenl46@mail.sysu.edu.cn).

Daxin Tian is with the Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems and Safety Control, Beijing Advanced Innovation Center for Big Data and Brain Computing, School of Transportation Science and Engineering, Beihang University, Beijing 100191, China (e-mail: dtian@buaa.edu.cn).

Digital Object Identifier 10.1109/TITS.2020.3023541

improve road safety, traffic efficiency and personal mobility [1]–[3]. However, technical challenges and the cost of exteroceptive sensors have constrained current applications of autonomous driving systems to confined and controlled environments in small quantities. One critical challenge is to obtain an adequately accurate understanding of the vehicle’s 3D surrounding environment in real-time. To this end, sensor fusion, which leverages multiple types of sensors with complementary characteristics to enhance perception and reduce cost, has become an emerging research theme.

In particular, recent deep learning advances have significantly improved the performance of camera-LiDAR fusion algorithms. Cameras and LiDARs have complementary characteristics, which make camera-LiDAR fusion models more effective and popular compared with other sensor fusion configurations (radar-camera, LiDAR-radar, etc.). To be more specific, vision-based perception systems achieve satisfactory performance at low-cost, often outperforming human experts [4], [5]. However, a mono-camera perception system cannot provide reliable 3D geometry, which is essential for autonomous driving [6], [7]. On the other hand, stereo cameras can provide 3D geometry, but do so at high computational cost and struggle in high-occlusion and textureless environments [8]–[10]. Furthermore, camera base perception systems struggle with complex or poor lighting conditions, which limit their all-weather capabilities [11]. Contrarily, LiDAR can provide high-precision 3D geometry and is invariant to ambient light. However, mobile LiDARs are limited by low-resolution (ranging from 16 to 128 channels), low-refresh rates (10Hz), severe weather conditions (heavy rain, fog and snow) and high cost. To mitigate these challenges, many works combined these two complementary sensors and demonstrated significant performance advantages than a-modal approaches. Therefore, this article focuses on reviewing current deep learning fusion strategies for camera-LiDAR fusion.

Camera-LiDAR fusion is not a trivial task. First of all, cameras record the real-world by projecting it to the image plane, whereas the point cloud preserves the 3D geometry. Furthermore, in terms of data structure, the point cloud is irregular, orderless and continuous, while the image is regular, ordered and discrete. These characteristics differences between the point cloud and the image lead to different feature extraction methodologies. In Figure 1, a comparison

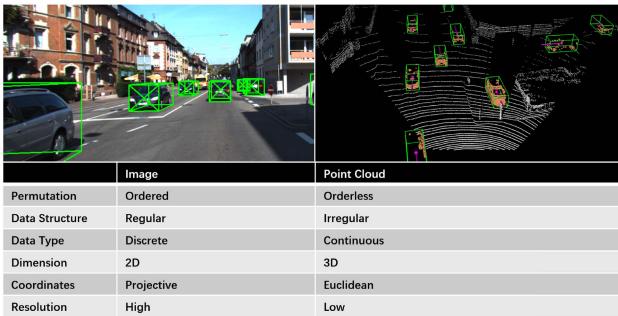


Fig. 1. A comparison between image data and point cloud data.

between the characteristics of image and point are shown.

Previous reviews [12], [13] on deep learning methods for multi-modal data fusion covered a broad range of sensors, including radars, cameras, LiDARs, Ultrasonics, IMU, Odometers, the GNSS and HD Maps. This article focuses on camera-LiDAR fusion only and therefore is able to present more detailed reviews on individual methods. Furthermore, we cover a broader range of perception related topics (depth completion, dynamic and stationary object detection, semantic segmentation, tracking and online cross-sensor calibration) that are interconnected and are not fully included in the previous reviews [13]. The contribution of this article is summarized as the following:

- To the best of our knowledge, this article is the first survey focusing on deep learning based image and point cloud fusion approaches in autonomous driving, including depth completion, dynamic and stationary object detection, semantic segmentation, tracking and online cross-sensor calibration.
- This article organizes and reviews methods based on their fusion methodologies. Furthermore, this article presented the most up-to-date (2014–2020) overviews and performance comparisons of the state-of-the-art camera-LiDAR fusion methods.
- This article raises overlooked open questions, such as open-set detection and sensor-agnostic framework, that are critical for the real-world deployment of autonomous driving technology. Moreover, summaries of trends and possible research directions on open challenges are presented.

This article first provides a brief overview of deep learning methods on image and point cloud data in Section II. In Sections III to VIII, reviews on camera-LiDAR based depth completion, dynamic object detection, stationary object detection, semantic segmentation, object tracking and online sensor calibration are presented respectively. Trends, open challenges and promising directions are discussed in Section VII. Finally, a summary is given in Section VIII. Figure 2 presents the overall structures of this survey and the corresponding topics.

## II. A BRIEF REVIEW OF DEEP LEARNING

### A. Deep Learning on Image

Convolutional Neural Networks (CNNs) are one of the most efficient and powerful deep learning models for image

processing and understanding. Compared to the Multi-Layer-Perceptron (MLP), the CNN is shift-invariant, contains fewer weights and exploits hierarchical patterns, making it highly efficient for image semantic extraction. Hidden layers of a CNN consist of a hierarchy of convolutional layers, batch normalization layers, activation layers and pooling layers, which are trained end-to-end. This hierarchical structure extracts image features with increasing abstract levels and receptive fields, enabling the learning of high-level semantics.

### B. Deep Learning on Point Cloud

The point cloud is a set of data points, which are LiDAR's measurements of the detected object's surface. In terms of data structure, the point cloud is sparse, irregular, orderless and continuous. Point cloud encodes information in 3D structures and in per-point features (reflective intensities, color, normal, etc.,), which is invariant to scale, rigid transformation and permutation. These characteristics made feature extractions on the point cloud challenging for existing deep learning models, which require the modifications of existing models or developing new models. Therefore, this section focuses on introducing common methodologies for point cloud processing.

*1) Volumetric Representation Based:* The volumetric representation partitions the point cloud into a fixed-resolution 3D grid, where features of each grid/voxel are hand-crafted or learned. This representation is compatible with standard 3D convolution [14]–[16]. Several techniques have been proposed in [17] to reduce over-fittings, orientation sensitivity and capture internal structures of objects. However, the volumetric representation loses spatial resolution and fine-grained 3D geometry during voxelization which limits its performance. Furthermore, attempts to increase its spatial resolution (denser voxels) would cause computation and memory footprint to grow cubically, making it unscalable.

*2) Index/Tree Representation Based:* To alleviate constraints between high spatial resolution and computational costs, adapted-resolution partition methods that leverage tree-like data structures, such as kd-tree [18], [19], octree [20]–[22] are proposed. By dividing the point cloud into a series of unbalanced trees, regions can be partitioned based on their point densities. This allows regions with lower point densities to have lower resolutions, which reduce unnecessary computation and memory footprint. Point features are extracted along with the pre-built tree structure.

*3) 2D Views Representation Based:* 2D views/multi-views are generated by projecting the point cloud to multiple 2D view planes. These rendered multi-view images can be processed by standard 2D convolutions and features from these views are aggregated via view-pooling layers [23]. Thus, the permutation-invariant problem is solved by transforming point cloud to images and the translation-invariant is achieved by aggregating features from different views. Qi *et al.* [17] combined the volumetric representation with multi-views generated via sphere rendering. Unfortunately, 2D views methods lose the 3D geometry information during the view rendering and struggle with per-point label prediction [19].

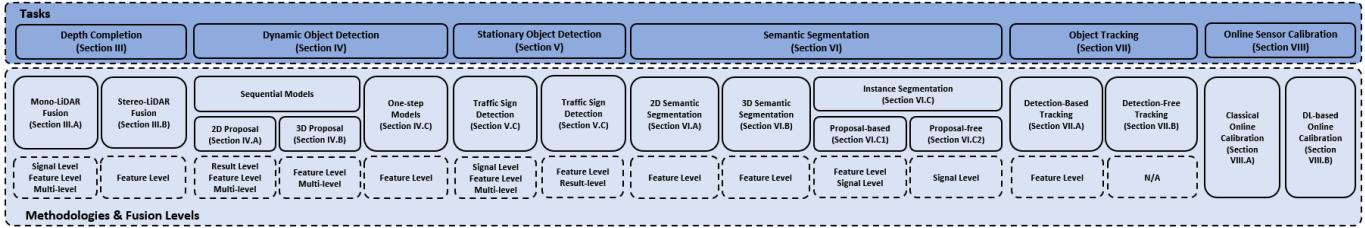


Fig. 2. Tasks related to image and point cloud fusion based perception and their corresponding sections.

4) *Graph Representation Based*: Point clouds can be represented as graphs and convolution-like operation can be implemented on graphs in the spatial or spectral domain [24]–[26]. For graph-convolution in the spatial domain, operations are carried out by MLPs on spatially neighbouring points. Spectral-domain graph-convolutions extend convolutions as spectral filtering on graphs through the Laplacian Spectrum [27]–[29].

5) *Point Representation Based*: Point representation based methods consume the point cloud without transforming it into an intermediate data representation. Early works in this direction employ shared Multi-Layer Perceptrons (MLPs) to process point cloud [30], [31], while recent works concentrated on defining specialized convolution operations for points [32]–[38].

One of the pioneering works of direct learning on point clouds is the PointNet [30], [31], which employs an independent T-Net module to align point clouds and shared MLPs to process individual points for per-point feature extraction. The computation complexity of the PointNet increases linearly with the number of inputs, making it more scalable compared with volumetric based methods. To achieve permutation invariant, point-wise features are extracted by shared MLPs which are identical for all points. These features are aggregated by symmetric operations (i.e. max-pooling), which are also permutation invariant. The feature extraction process of the PointNet is defined as:

$$g(\{x_1, \dots, x_n\}) \approx f_{sym}(h(x_1), \dots, h(x_n)) \quad (1)$$

where  $x$  represents input points,  $h$  represents the per-point feature extraction function (i.e. shared MLPs),  $f_{sym}$  represents a symmetric function (i.e. max-polling) and  $g$  is a general function that we want to approximate.

However, the PointNet fails to extract local inter-point geometry at different levels. To mitigate this challenge, Qi *et al.* [30] extended the PointNet to extract features from different levels by grouping points into multiple sets and apply PointNets locally. To reduce the computational and memory cost of the PointNet++ [30], the RandLA-Net [39] stacked the random point sampling modules and attention-based local feature aggregation modules hierarchically to progressively increase receptive field while maintaining high efficiency.

Unlike PointNet-based methods, the spatial relationship between points is explicitly modelled in point-wise convolutions. Point-wise convolutions aim to generalize the standard 2D discrete convolution to the continuous 3D space. The main challenge is to replace the discrete weight filter in

standard convolution with a continuous weight function. This continuous weight function is approximated using MLPs in PointConv [40] and correlation functions in KPConv [38] and PCNN [33]. More specifically, the PCNN [33] defines convolution kernels as 3D points with weights. A Gaussian correlation function that takes the coordinates of the kernel point and input point is used to calculate the weighting matrix at any given 3D coordinates. The KPConv [38] follows this idea but instead uses a linear correlation function. Furthermore, KPConvs [38] are applied on local point patches hierarchically, which are similar to the concepts of standard CNNs. This general point-wise convolution  $\mathcal{F}$  at an input point  $x \in \mathbb{R}^3$  in 3D continuous space is defined as:

$$(\mathcal{F} * h)(x) = \sum_{x_i \in \mathcal{N}_x} h(x_i - x) f_i \quad (2)$$

where  $h$  is the per-point kernel function which calculates the weighting matrix given the coordinates of input points and kernel points.  $x_i$  and  $f_i$  are the  $i$ th neighboring points of  $x$  and their corresponding features (intensity, color etc.).  $\mathcal{N}_x$  are all the neighboring points of the input point  $x$ , which are determined using KNN or radius neighborhoods.

### III. DEPTH COMPLETION

Depth Completion aims to up-sample sparse irregular depth to dense regular depth, which facilitates the down-stream perception module. Depth completion can reduce the drastic uneven distributions of points in a LiDAR scan. For instance, far-away objects represented by a hand full of points are up-sampled to match their closer counterparts. To achieve this, high-resolution images are often employed to guide the 3D depth up-sampling. The depth completion task can be represented as:

$$w^* = \arg \min_w \mathcal{L}(f(x; w), G) \quad (3)$$

where the network  $f(\cdot)$  parametrized by  $w$ , predicts the ground truth  $G$ , given the input  $x$ . The loss function is represented as  $\mathcal{L}(\cdot, \cdot)$ .

Figure 3 gives the timeline of depth completion models and their corresponding fusion levels. The comparative results of depth completion models on the KITTI depth completion benchmark [41] are listed in Table I.

#### A. Mono Camera and LiDAR Fusion

The idea behind image-guided depth completion is that dense RGB/color information contains relevant 3D geometry.

TABLE I

COMPARATIVE RESULTS ON KITTI DEPTH COMPLETION BENCHMARK [41]. 'LL', 'S', 'SS', 'U' STAND FOR LEARNING SCHEME, SUPERVISED, SELF-SUPERVISED AND UNSUPERVISED RESPECTIVELY. 'RGB', 'sD', 'Co', 'S', STAND FOR RGB IMAGE DATA, SPARSE DEPTH, CONFIDENCE AND STEREO DISPARITY. 'iRMSE', 'iMAE', 'RMSE', 'MAE' STAND FOR ROOT MEAN SQUARED ERROR OF THE INVERSE DEPTH [1/KM], MEAN ABSOLUTE ERROR OF THE INVERSE DEPTH [1/KM], ROOT MEAN SQUARED ERROR [MM], MEAN ABSOLUTE ERROR [MM]

Methods	Fusion Level	LL	Input	Models	Hardware	Run Time	Model Size	iRMSE	iMAE	RMSE	MAE
Mono-LiDAR Fusion	Signal Level	S	RGB+sD	Sparse2Dense [42]	Tesla P100	0.08s	12M	4.07	1.57	1299.85	350.32
		SS	RGB+sD	Sparse2Dense++ [43]	Tesla V100	0.08s	22M	2.80	1.21	814.73	249.95
		S	RGB+sD	CSPN [44]	Titan X	1s	25M	2.93	1.15	1019.64	279.46
		S	RGB+sD	CSPN++ [45]	Tesla P40	0.2s	-	2.07	0.90	743.69	209.28
	Feature Level	S	RGB+sD	Spade-RGBsD [46]	N/A	-	-	2.17	0.95	917.64	234.81
		S	RGB+sD+Co	NConv-CNN [47]	Tesla V100	0.02s	0.5M	2.60	1.03	829.98	233.26
		S	RGB+sD	GuideNet [48]	GTX 1080Ti	0.14s	-	2.25	0.99	736.24	218.83
	Multi	U	RGB+sD	RGBguide [49]	Tesla V100	0.02s	0.35M	2.19	0.93	772.87	215.02
Stereo-LiDAR	Feature Level	S	sD+S	HDE-Net [9]	Titan X	0.05s	4.2M	-	-	-	-
		U	sD+S	LidarStereoNet [50]	Titan X	-	-	-	-	-	-
		SS	sD+S	LiStereo [51]	Titan X	-	-	2.19	1.10	832.16	283.91

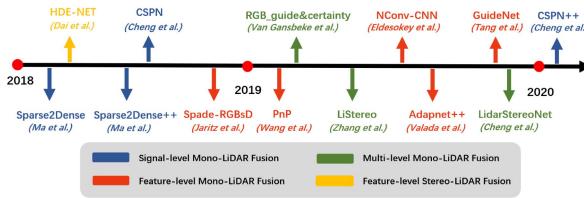


Fig. 3. Timeline of depth completion models and their corresponding fusion levels.

Therefore, images can be leveraged as a reference for depth up-sampling.

1) *Signal-Level Fusion*: In 2018, Ma and Karaman [42] presented a ResNet [52] based autoencoder network that leverages RGB-D images (i.e. images concatenated with sparse depth maps) to predict dense depth maps. However, this method requires pixel-level depth ground truth, which is difficult to obtain. To solve this issue, Ma *et al.* [43] presented a model-based self-supervised framework that only requires a sequence of images and sparse depth images for training. This self-supervision is achieved by employing sparse depth constrain, photometric loss and smoothness loss. However, this approach assumes objects to be stationary. Furthermore, the resulting depth output is blurry and input depth may not be preserved.

To generate a sharp dense depth map in real-time, Cheng *et al.* [44] fed RGB-D images to a convolutional spatial propagation network (CSPN). This CSPN aims to extract the image-dependent affinity matrix directly, producing significantly better results in key measurements with lesser run-time. In CSPN++, Cheng *et al.* [45] proposed to dynamically select convolutional kernel sizes and iterations to reduce computation. Furthermore, CSPN++ employs weighted assembling to boost its performance.

2) *Feature-Level Fusion*: Jaritz *et al.* [46] presented an autoencoder network that can either perform depth completion or semantic segmentation from sparse depth maps and images without applying validity masks. Images and sparse depth maps are first processed by two parallel NASNet-based encoders [53] before fusing them into the shared decoder. This approach can achieve decent performance with very sparse

depth inputs (8-channel LiDAR). Wang *et al.* [54] designed an integrable module (PnP) that leverages the sparse depth map to improve the performance of existing image-based depth prediction networks. This PnP module leverages gradient calculated from sparse depth to update the intermediate feature map produced by the existing depth prediction network. Eldesokey *et al.* [47] presented a framework for unguided depth completion that processes images and very sparse depth maps in parallel and combine them in a shared decoder. Furthermore, normalized convolutions are used to process highly sparse depth and to propagate confidence. Valada *et al.* [55] extended one-stage feature-level fusion to multiple-stages of varying depth of the network. Similarly, GuideNet [48] fuse image features to sparse depth features at different stages of the encoder to guide the up-sampling of sparse depths, which achieves top performance in the KITTI depth completion benchmark. The constrain of these approaches is the lack of large-scale datasets that have dense depth ground truth.

3) *Multi-Level Fusion*: Van Gansbeke *et al.* [49] further combines signal-level fusion and feature-level fusion in an image-guided depth completion network. The network consists of a global and a local branch to process RGB-D data and depth data in parallel before fusing them based on the confidence maps.

### B. Stereo Cameras and LiDAR Fusion

Compared with the RGB image, dense depth disparity from stereo cameras contains richer ground truth 3D geometry. On the other hand, LiDAR depth is sparse but of higher accuracy. These complementary characteristics enable stereo-LiDAR fusion based depth completion models to produce a more accurate dense depth. However, it is worth noting that stereo cameras have limited range and struggles in high-occlusion, texture-less environments, making them less ideal for autonomous driving.

1) *Feature-Level Fusion*: One of the pioneering works is from Park *et al.* [9], in which high-precision dense disparity map is computed from dense stereo disparity and point cloud using a two-stage CNN. The first stage of the CNN takes LiDAR and stereo disparity to produce a fused disparity. In the second stage, this fused disparity and left

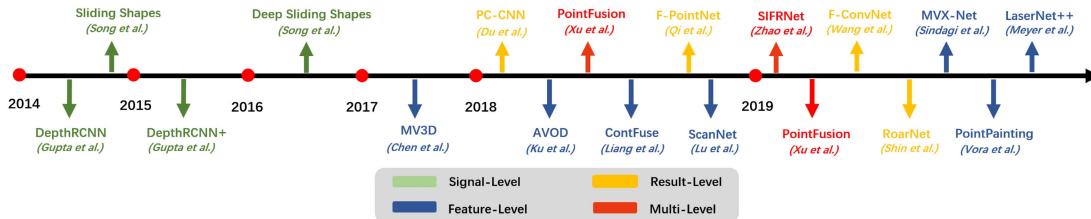


Fig. 4. Timeline of 3D object detection networks and their corresponding fusion levels.

TABLE II

COMPARATIVE RESULTS ON KITTI 3D OBJECT DETECTION BENCHMARK (MODERATE DIFFICULTIES) [56]. THE IOU REQUIREMENT FOR CAR, PEDESTRIANS AND CYCLISTS IS 70%, 50% AND 50% RESPECTIVELY. 'FL', 'FT', 'RT' 'MS', 'PCR' STAND FOR FUSION LEVEL, FUSION TYPE, RUN-TIME, MODEL SIZE (NUMBER OF PARAMETERS) AND POINT CLOUD REPRESENTATION RESPECTIVELY

Methods	FL	FT	PCR	Models	Hardware	RT	MS	Cars			Pedestrians			
								Cars	AP <sub>3D</sub> %	AP <sub>3D</sub> %	Pedestrians	AP <sub>3D</sub> %	Cyclists	AP <sub>3D</sub> %
Sequential Models	2D Proposal based	Result	ROI	Points Points Voxels Points Points	F-PointNet [57] F-ConvNet [65] PC-CNN [58] RoarNet [59] IPOD [61]	GTX1080 GTX1080Ti - GTX1080Ti GTX1080Ti	0.17s 0.47s - 0.1s 0.2s	7M 30M 276M - 24M	69.79 76.39 73.79 73.04 72.57	42.15 43.38 - - 44.68	56.12 65.07 - - 53.46			
			Feature	Point	Multiple	PointPainting [66]	GTX1080	0.4s	-	71.70	40.97	63.78		
		Multi	ROI	Points Points	PointFusion [62] SIFRNet [63]	GTX1080	1.3s	6M	63.00	28.04	29.42			60.34
	3D Proposal based	Feature	ROI	2D views	MV3D [67]	Titan X	0.36s	240M	63.63	-	-			
			ROI	2D views	AVOD-FPN [68]	Titan Xp	0.08s	38M	71.76	42.27	50.55			
			ROI	2D views	SCANet [69]	GTX1080Ti	0.17s	-	68.12	37.93	53.38			
			Point	2D views	ContFuse [70]	GTX1080	0.06s	-	68.78	-	-			
			Point	Voxels	MVX-Net [71]	-	-	-	77.43	-	-			45.00
			Pixel	2D views	BEVF [72]	-	-	-	-	-	-			
One-Step Models	Feature	Pixel	2D views	MMF [73]	GTX1080	0.08s	-	-	77.43	-	-			
				LaserNet++ [74]	TITAN Xp	0.04s	-	-	-	-	-			

RGB image is fused in the feature space to predict the final high-precision disparity. Finally, the 3D scene is reconstructed from this high-precision disparity. The bottleneck of this approach is the lack of large-scale annotated stereo-LiDAR datasets. The LidarStereoNet [50] averted this difficulty with an unsupervised learning scheme, which employs image warping/photometric loss, sparse depth loss, smoothness loss and plane fitting loss for end-to-end training. Furthermore, the introduction of 'feedback loop' makes the LidarStereoNet robust against noisy point cloud and sensor misalignment. Similarly, Zhang *et al.* [51] presented a self-supervised scheme for depth completion. The loss function consists of sparse depth, photometric and smoothness loss.

#### IV. DYNAMIC OBJECT DETECTION

Object detection (3D) aims to locate, classify and estimate oriented bounding boxes in the 3D space. This section is devoted to dynamic object detection, which includes common dynamic road objects (car, pedestrian, cyclist, etc.). There are two main approaches for object detection: sequential and one-step. Sequential based models consist of a proposal stage and a 3D bounding box (bbox) regression stage in the chronological order. In the proposal stage, regions that may contain objects of interest are proposed. In the bbox regression stage, these proposals are classified based on the region-wise features extracted from 3D geometry. However, the performance of sequential fusion is limited by each stage. On the other hand, one-step models consist of one stage, where 2D and 3D data are processed in a parallel manner.

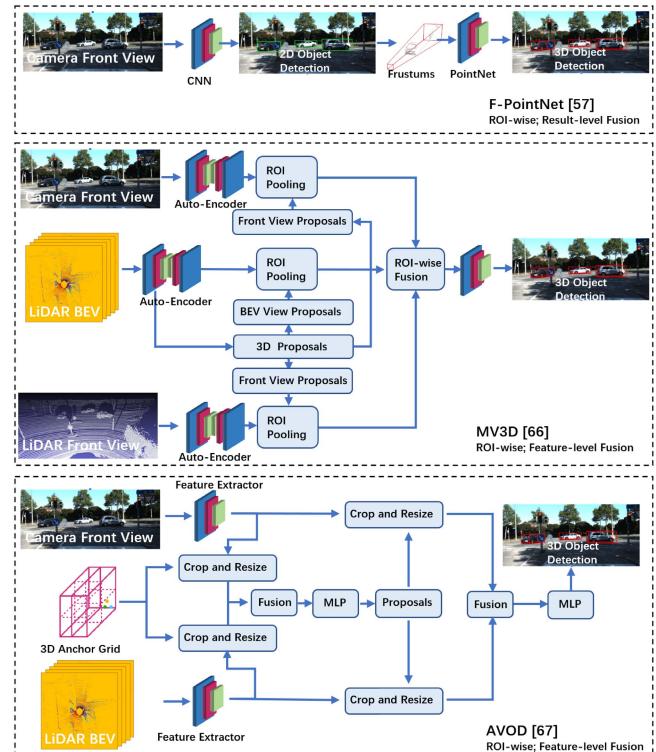


Fig. 5. A comparison between three typical model architectures for dynamic object detection.

The timeline of 3D object detection networks and typical model architectures are shown in Figures 4 & 5. Table II presents the comparative results of 3D object detection models

TABLE III  
A SUMMARIZATION AND COMPARISON BETWEEN DYNAMIC OBJECT DETECTION MODELS

Methods:	Key features:	Advantages:	Disadvantages:
Frustum based [57] [65] [58] [59] [61] [63] [62]	1.Leverage image object detector to generate 2D proposals and project them to form frustum 3D search spaces for 3D object detection. 2.Result-level fusion / Multi-level fusion	3D search space is limited using 2D results to reduce computational cost.	1. Due to sequential result-level fusion, the overall performance is limited by the image detector. 2. Redundant information from multimodal sensors is not leveraged.
Point-fusion based [66]	1.Fuse high-level image semantics point-wise and perform 3D object detection in the fused point cloud. 2.Feature-level fusion.	Solve the resolution mismatch problem between dense RGB and sparse depth via fusing high-level image semantics to points.	1.Image and LiDAR models are highly coupled, which reduce overall reliability and increase development cost. 2.3D search space is not limited which leads to high computational cost.
Multi-view based [67] [68] [69] [70] [73]	1.Generate 3D proposals from BEV and perform 3D bboxes regression on these proposals. 2.Feature-level fusion.	Enable the use of standard 2D convolution and off-the-shelf models, making it much more scalable.	1.Assume all objects are visible in LiDAR's BEV, which is often not the case. 2.Spatial information of small object instances is lost during consecutive convolution operations 3.ROI fusion spoils fine-grained geometric information.
Voxel-based [71] [82]	1.Fuse image semantics voxel-wise and perform 3D bboxes regression using voxel-based network. 2.Feature-level fusion.	Compatible with standard 3D convolution.	1. Spatial resolution and fine-grained 3D geometry information are lost during voxelization. 2. Computation and memory footprint to grow cubically with the resolution, making it unscalable.

on the KITTI 3D Object Detection benchmark [56]. Table III summarizes and compares dynamic object detection models.

#### A. 2D Proposal Based Sequential Models

A 2D proposal based sequential model attempts to utilize 2D image semantics in the proposal stage, which takes advantage of off-the-shelf image processing models. Specifically, these methods leverage the image object detector to generate 2D region proposals, which are projected to the 3D space as detection seeds. There are two projection approaches to translate 2D proposals to 3D. The first one is projecting bounding boxes in the image plane to the point cloud, which results in a frustum shaped 3D search space. The second method projects the point cloud to the image plane, which results in the point cloud with point-wise 2D semantics.

1) *Result-Level Fusion*: The intuition behind result-level fusion is to use off-the-shelf 2D object detectors to limit the 3D search space for 3D object detection, which significantly reduces computation and improve run-time. However, since the whole pipeline depends on the results of the 2D object detector, it suffers from the limitations of the image-based detector.

One of the early works of result-level fusion is the F-PointNets [57], where 2D bounding boxes are first generated from images and projected to the 3D space. The resulting projected frustum proposals are fed into a PointNet [31] based detector for 3D object detection. Du *et al.* [58] extended the 2D to 3D proposals generation stage with an additional proposal refinement stage, which further reduces unnecessary computation on the background point. During this refinement stage, a model fitting based method is used to filter out background points inside the seed region. Finally, the filtered points are fed into the bbox regression network. The RoarNet [59] followed a similar idea, but instead employs a neural network for the proposal refinement stage. Multiple 3D cylinder proposals are first generated based on each 2D bbox using the geometric

agreement search [60], which results in smaller but more precise frustum proposals than the F-pointNet [57]. These initial cylinder proposals are then processed by a PointNet [30] based header network for the final refinement. To summarize, these approaches assume each seed region only contains one object of interest, which is however not true for crowded scenes and small objects like pedestrians.

One possible solution towards the fore-mentioned issues is to replace the 2D object detector with 2D semantic segmentation and region-wise seed proposal with point-wise seed proposals. Intensive Point-based Object Detector (IPOD) [61] by Yang *et al.* is a work in this direction. In the first step, 2D semantic segmentation is used to filter out back-ground points. This is achieved by projecting points to the image plane and associated point with 2D semantic labels. The resulting foreground point cloud retains the context information and fine-grained location, which is essential for the region-proposal and bbox regression. In the following point-wise proposal generation and bbox regression stage, two PointNet++ [30] based networks are used for proposal feature extraction and bbox prediction. In addition, a novel criterion called PointsIoU is proposed to speed up training and inference. This approach has yielded significant performance advantages over other state-of-the-art approaches in scenes with high occlusion or many objects.

2) *Multi-Level Fusion*: Another possible direction of improvement is to combine result level fusion with feature level fusion, where one such work is PointFusion [62]. The PointFusion first utilizes an existing 2D object detector to generate 2D bboxes. These bboxes is used to select corresponding points, via projecting points to the image plane and locate points that pass through the bboxes. Finally, a ResNet [52] and a PointNet [31] based network combines image and point cloud features to estimate 3D objects. In this approach, image features and point cloud features are fused per-proposal for final object detection in 3D, which facilitates 3D bbox regression.

However, its proposal stage is still amodal. In SIFRNet [63], frustum proposals are first generated from an image. Point cloud features in these frustum proposals are then combined with their corresponding image features for final 3D bbox regression. To achieve scale-invariant, the PointSIFT [64] is incorporated into the network. Additionally, the SENet module is used to suppress less informative features.

3) *Feature-Level Fusion*: Early attempts [75], [76] of multimodal fusion are done in pixel-wise, where 3D geometry is converted to image format or appended as additional channels of an image. The intuition is to project 3D geometry onto the image plane and leverage mature image processing methods for feature extraction. The resulting output is also on the image plane, which is not ideal to locate objects in the 3D space. In 2014, Gupta *et al.* proposed DepthRCNN [75], an R-CNN [77] based architecture for 2D object detection, instance and semantic segmentation. It encodes 3D geometry from Microsoft Kinect camera in image's RGB channels, which are Horizontal disparity, Height above ground, and Angle with gravity (HHA). Gupta *et al.* extended Depth-RCNN [78] in 2015 for 3D object detection by aligning 3D CAD models, yielding significant performance improvement. In 2016, Gupta *et al.* developed a novel technique for supervised knowledge transfer between networks trained on image data and unseen paired image modality (depth image) [76]. In 2016, Schlosser *et al.* [79] further exploited learning RGB-HHA representations on 2D CNNs for pedestrian detection. However, the HHA data are generated from the LiDAR's depth instead of a depth camera. The authors also noticed that better results can be achieved if the fusion of RGB and HHA happens at deeper layers of the network.

The resolution mismatch between dense RGB and sparse depth means only a small portion of pixels have corresponding points. Therefore, to directly append RGB information to points leads to the loss of most texture information, rendering the fusion pointless. To mitigate this challenge, PointPainting [66] extract high-level image semantic before the per-point fusion. To be more specific, PointPainting [66] follows the idea of projecting points to 2D semantic maps in [61]. But instead of using 2D semantics to filter non-object points, 2D semantics is simply appended to point clouds as additional channels. The authors argued that this technique made PointPainting flexible as it enables any point cloud networks to be applied on this fused data. To demonstrate this flexibility, the fused point cloud is fed into multiple existing point cloud detectors, which are based on the PointRCNN [80], the VoxelNet [14] and the PointPillar [81]. However, this would lead to the coupling between image and LiDAR models. This requires the LiDAR model to be re-trained when the image model changes, which reduces the overall reliability and increases the development cost.

### B. 3D Proposal Based Sequential Model

In a 3D proposal based sequential model, 3D proposals are directly generated from 2D or 3D data. The elimination of 2D to 3D proposal transformation greatly limits the 3D

search space for 3D object detection. Common methods for 3D proposal generation includes the multi-view approach and the point cloud voxelization approach.

Multi-view based approach exploits the point cloud's bird's eye view (BEV) representation for 3D proposal generation. The BEV is the preferred viewpoint because it avoids occlusions and retains the raw information of objects' orientation and  $x$ ,  $y$  coordinates. These orientation and  $x$ ,  $y$  coordinates information is critical for 3D object detection while making coordinate transformation between BEV and other views straight-forward.

Point cloud voxelization transforms the continuous irregular data structure to a discrete regular data structure. This makes it possible to apply standard 3D discrete convolution and leverage existing network structures to process point cloud. The drawback is the loss of some spatial resolution, which might contain fine-grained 3D structure information.

1) *Feature-Level Fusion*: One of the pioneering and most important works in generating 3D proposals from BEV representations is MV3D [67]. MV3D generate 3D proposals on pixelized top-down LiDAR feature map (height, density and intensity). These 3D candidates are then projected to the LiDAR front view and image plane to extract and fuse region-wise features for bbox regression. The fusion happens at the region-of-interest (ROI) level via ROI pooling. The  $ROI_{views}$  of views is defined as:

$$ROI_{views} = \mathbf{T}_{3D \rightarrow views}(p_{3D}), views \in \{\text{BV, FV, RGB}\} \quad (4)$$

where  $\mathbf{T}_{3D \rightarrow views}$  represents the transformation function that project point cloud  $p_{3D}$  from 3D space to bird's eye view (BEV), front view (FV) and the image plane (RGB). The ROI-pooling  $\mathbf{R}$  to obtain feature vector  $f_{views}$  is defined as:

$$f_{views} = \mathbf{R}(x, ROI_{views}), views \in \{\text{BV, FV, RGB}\} \quad (5)$$

There are a few drawbacks of the MV3D. First, generating 3D proposals on BEV assumes that all objects of interest are captured without occlusions from this view-point. This assumption does not hold well for small object instances, such as pedestrians and bicyclists, which can be fully occluded by other large objects in the point cloud. Secondly, spatial information of small object instances is lost during the down-sample of feature maps caused by consecutive convolution operations. Thirdly, object-centric fusion combines feature maps of image and point clouds through ROI-pooling, which spoils fine-grained geometric information in the process. It is also worth noting that redundant proposals lead to repetitive computation in the bbox regression stage. To mitigate these challenges, multiple methods have been put forward to improve MV3D.

To improve the detection of small objects, the Aggregate View Object Detection network (AVOD) [68] first improved the proposal stage in MV3D [67] with feature maps from both BEV point cloud and image. Furthermore, an auto-encoder architecture is employed to up-sample the final feature maps to its original size. This alleviates the problem that small objects might get down-sampled to one 'pixel' with consecutive convolution operations. The proposed feature fusion Region Proposal Network (RPN) first extracts equal-length

feature vectors from multiple modalities (BEV point cloud and image) with crop and resize operations. Followed by a  $1 \times 1$  convolution operation for feature space dimensionality reduction, which can reduce computational cost and boost up speed. Lu *et al.* [69] also utilized an encoder-decoder based proposal network with Spatial-Channel Attention (SCA) module and Extension Spatial Upsample (ESU) module. The SCA can capture multi-scale contextual information, whereas ESU recovers the spatial information.

One of the problems in object-centric fusion methods [67], [68] is the loss of fine-grained geometric information during ROI-pooling. The ContFuse [70] by Liang *et al.* tackles this information lost with point-wise fusion. This point-wise fusion is achieved with continuous convolutions [83] fusion layers that bridge image and point cloud features of different scales at multiple stages in the network. This is achieved by first extracting K-nearest neighbour points for each pixel in the BEV representation of point cloud. These points are then projected to the image plane to retrieve related image features. Finally, the fused feature vector is weighted according to their geometry offset to the target 'pixel' before feeding into MLPs. However, point-wise fusion might fail to take full advantage of high-resolution images when the LiDAR points are sparse. In [73] Liang *et al.* further extended point-wise fusion by combining multiple fusion methodologies, such as signal-level fusion (RGB-D), feature-level fusion, multi-view and depth completion. In particular, depth completion upsamples sparse depth map using image information to generate a dense pseudo point cloud. This up-sampling process alleviates the sparse point-wise fusion problem, which facilitates the learning of cross-modality representations. Furthermore, the authors argued that multiple complementary tasks (ground estimation, depth completion and 2D/3D object detection) could assist the network achieve better overall performance. However, point-wise/pixel-wise fusion leads to the 'feature blurring' problem. This 'feature blurring' happens when one point in the point cloud is associated with multiple pixels in the image or the other way around, which confound the data fusion. Similarly, Wang *et al.* [72] replace the ROI-pooling in MV3D [67] with sparse non-homogeneous pooling, which enables effective fusion between feature maps from multiple modalities.

MVX-Net [71] presented by Sindagi *et al.* introduced two methods that fuse image and point cloud data point-wise or voxel-wise. Both methods employ a pre-trained 2D CNN for image feature extraction and a VoxelNet [14] based network to estimate objects from the fused point cloud. In the point-wise fusion method, the point cloud is first projected to image feature space to extract image features before voxelization and processed by VoxelNet. The voxel-wise fusion method first voxelized the point cloud before projecting non-empty voxels to the image feature space for voxel/region-wise feature extraction. These voxel-wise features are only appended to their corresponding voxels at a later stage of the VoxelNet. MVX-Net achieved state-of-the-art results and out-performed other LiDAR-based methods on the KITTI benchmark while lowering false positives and false negatives rate compared to [14].

The simplest means to combine the voxelized point cloud and image is to append RGB information as additional channels of a voxel. In a 2014 paper by Song and Xiao [82] 3D object detection is achieved by sliding a 3D detection window on the voxelized point cloud. The classification is performed by an ensemble of Exemplar-SVMs. In this work, color information is appended to voxels by projection. Song *et al.* further extended this idea with 3D discrete convolutional neural networks [84]. In the first stage, the voxelized point cloud (generated from RGB-D data) is first processed by Multi-scale 3D RPN for 3D proposal generation. These candidates are then classified by joint Object Recognition Network (ORN), which takes both image and voxelized point cloud as inputs. However, the volumetric representation introduces boundary artifacts and spoils fine-grained local geometry. Secondly, the resolution mismatch between image and voxelized point cloud makes fusion inefficient.

### C. One-Step Models

One-step models perform proposal generation and bbox regression in a single stage. By fusing the proposal and bbox regression stage into one-step, these models are often more computationally efficient. This makes them more well-suited for real-time applications on mobile computational platforms.

Meyer *et al.* [74] extended the LaserNet [85] to multi-task and multimodal network, performing 3D object detection and 3D semantic segmentation on fused image and LiDAR data. Two CNN process depth-image (generated from point cloud) and front-view image in a parallel manner and fuse them via projecting points to the image plane to associate corresponding image features. This feature map is fed into the LaserNet to predict per-point distributions of the bounding box and combine them for final 3D proposals. This method is highly efficient while achieving state-of-the-art performance.

## V. STATIONARY ROAD OBJECT DETECTION

This section focuses on reviewing recent advances in camera-LiDAR fusion based stationary road object detection methods. Stationary road objects can be categorized into on-road objects (e.g. road surfaces and road markings) and off-road objects (e.g. traffic signs). On-road and off-road objects provide regulations, warning bans and guidance for autonomous vehicles.

In Figure 6 & 7, typical model architecture in lane/road detection and traffic sign recognition (TSR) are compared. Table IV presents the comparative results of different models on the KITTI road benchmark [56] and gives summarization and comparison between these models.

### A. Lane/Road Detection

Existing surveys [93]–[95] have presented detailed reviews on traditional multimodal road detection methods. These methods [96]–[99] mainly rely on vision for road/lane detection while utilizing LiDAR for the curb fitting and obstacle masking. Therefore, this section focuses on recent advances in deep learning based fusion strategies for road extraction.

TABLE IV

COMPARATIVE RESULTS ON KITTI ROAD BENCHMARK [56] WITH SUMMARIZATION AND COMPARISON BETWEEN LANE/ROAD DETECTION MODELS.'MAXF (%)' STANDS FOR MAXIMUM F1-MEASURE ON KITTI URBAN MARKED ROAD TEST SET.  
'FL','MS' STAND FOR FUSION LEVEL AND MODEL PARAMETER SIZE

Method:	Inputs:	FL:	MaxF	MS	Highlights:	Disadvantages:
BEV-based Lane/Road Detection						
DMLD [86]	BEV Image + BEV Point Cloud	Feature	-	50M	Predict dense BEV height map from sparse depth which improves performance	Cannot distinguish between different lane types
EDUNet [87]	Fused BEV Occupation Grid	Signal	93.81	143M	Signal-level fusion reduces model complexity	Loss of dense RGB/textured information in early fusion
TSF-FCN [88]	BEV Image + BEV Point Cloud	Feature	94.86	2M	Relatively moderate computational cost	Relatively moderate performance
MSRF [89]	BEV Image + BEV Point Cloud	Feature	96.15	-	Multi-stage fusion at different network depth improves performance	Relatively high computational cost
Camera-view based Lane/Road Detection						
LidCamNet [90]	Image + Dense Depth Image	Multi	96.03	3M	Explore and compared multiple fusion strategies for road detection	Rely on image for depth up-sampling
PLARD [91]	Image + Dense Depth Image	Feature	97.05	-	Progressive fusion strategy improves performance	Relatively high computational cost
SpheSeg [92]	Image + Depth Image	Feature	96.94	1.4M	Spherical coordinate transformation reduces input size and improves speed	Road detection in 2D space rather than 3D space

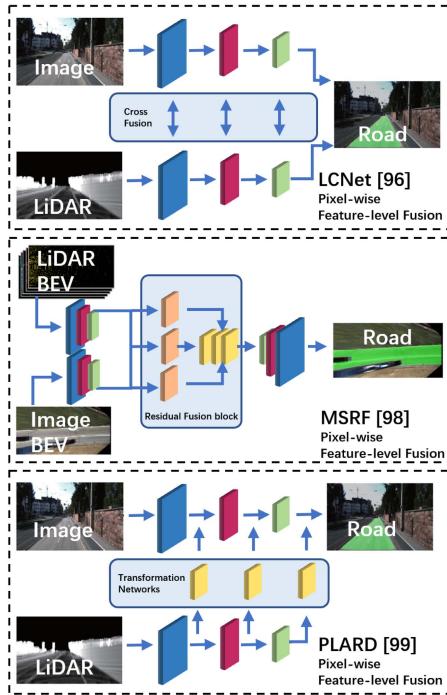


Fig. 6. Some typical model architectures and fusion methods for road/lane detection.

Deep learning based road detection methods can be grouped into BEV-based or front-camera-view-based. BEV-based methods [86]–[89] project LiDAR depth and images to BEV for road detection, which retains the objects' original  $x$ ,  $y$  coordinates and orientation. In [86], the dense BEV height estimation is predicted from the point cloud using a CNN, which is then fused with the BEV image for accurate lane detection. However, this method cannot distinguish different lane types. Similarly, Lv *et al.* [88] also utilized the BEV LiDAR grid map and the BEV image but instead processed them in a parallel manner. Yu *et al.* [89] proposed a multi-stage fusion strategy (MSRF) that combines image-depth features at

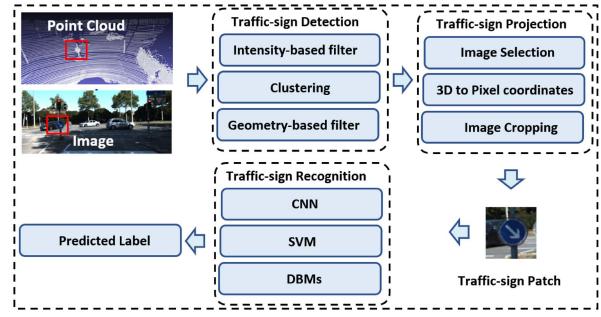


Fig. 7. A typical fusion-based traffic-sign recognition pipeline.

different network levels, which significantly improves its performance. However, this strategy also relatively increases its computational cost. Wulff *et al.* [87] used signal-level fusion to generate a fused BEV occupation grid, which is processed by a U-net based road segmentation network. However, the signal-level fusion between dense RGB and sparse depth leads to the loss of dense texture information due to the low grid resolution.

Front-camera-view-based methods [90]–[92] project LiDAR depth to the image plane to extract road surface, which suffers from accuracy loss in the translation of 2D to 3D boundaries. The LCNet [90] compared signal-level fusion (early fusion) and feature level fusion (late and cross fusion) for road detection, which finds the cross fusion is the best performing fusion strategy. Similar to [88], PLARD [91] fuses image and point cloud features progressively in multiple stages. Lee and Park [92] focused on improving speed via a spherical coordinate transformation scheme that reduces the input size. This transformed camera and LiDAR data are further processed by a SegNet based semantic segmentation network.

### B. Traffic Sign Recognition

In LiDAR scans, traffic-signs are highly distinguishable due to its retro-reflective property, but the lack of dense texture makes it difficult to classify. On the contrary, traffic sign image

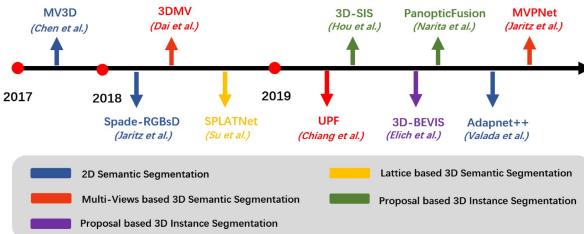


Fig. 8. Timeline of 3D semantic segmentation networks and their corresponding fusion levels.

patches can be easily classified. However, it is difficult for vision-based TSR system to locate these traffic-signs in the 3D space. Therefore, various studies have proposed to utilize both camera and LiDAR for TSR. Existing reviews [93], [100] have comprehensively covered traditional traffic sign recognition methods and part of the deep learning methods. Hence, this section presents a brief overview of traditional traffic sign recognition methods and mostly focuses on recent advances. In a typical TSR fusion pipeline [101]–[105], traffic-signs are first located in the LiDAR scan based on its retro-reflective property. These 3D positions of detected traffic-signs are then projected to the image plane to generate traffic-signs patches, which are fed into an image classifier for classification. This TSR fusion pipeline is shown in Fig. 7.

For methods that employ the typical TSR fusion pipeline, the main difference is on the classifier. These classifiers include deep Boltzmann machine (DBMs) based hierarchical classifier [102], SVMs [101] and DNN [104]. To summarize, these methods all employ result-level fusion and a hierarchical object detection model. They assume traffic-signs are visible in the LiDAR scan, which sometimes is not the case due to the occlusion. Furthermore, this pipeline is limited by the detection range of mobile LiDARs.

To mitigate these challenges, Deng and Zhou [106] combined image and point cloud to generate a colorized point cloud for both traffic-sign detection and classification. In addition, the 3D geometrical properties of detected traffic-signs are utilized to reduce false-positives. In [107] traffic-signs are detected based on prior knowledge, which includes road geometry information and traffic-sign geometry information. The detected traffic-signs patches are classified by a Gaussian–Bernoulli DBMs model. Following this ideal, Guan *et al.* [105] further improved the traffic sign recognition part using a convolutional capsule network. To summarize, these methods improve the traffic sign detection stage with multimodal data and prior knowledge. However, prior knowledge is often region-specific, which makes it difficult to generalize to other parts of the world.

## VI. SEMANTIC SEGMENTATION

This section reviews existing camera-LiDAR fusion methods for 2D semantic segmentation, 3D semantic segmentation and instance segmentation. 2D/3D semantic segmentation aims to predict per-pixel and per-point class labels, while instance segmentation also cares about individual instances. Figure 8 & 9 present a timeline of 3D semantic segmentation networks and typical model architectures.

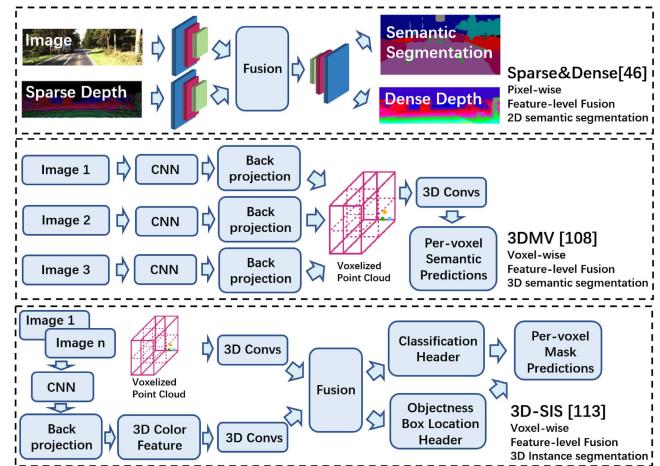


Fig. 9. Some typical model architectures and fusion methods for semantic segmentation.

### A. 2D Semantic Segmentation

1) **Feature-Level Fusion:** Sparse & Dense [46] presented a NASNet [53] based auto-encoder network that can be used for 2D semantic segmentation or depth completion leveraging image and sparse depths. The image and corresponding sparse depth map are processed by two parallel encoders before fused into the shared decoder. Valada *et al.* [55] employed a multi-stage feature-level fusion of varying depth to facilitate semantic segmentation. Caltagirone *et al.* [90] utilized up-sampled depth-image and image for 2D semantic segmentation. This dense depth-image is up-sampled using sparse depth-images (from point cloud) and images [108]. The best performing cross-fusion model processes dense depth-image and image data in two parallel CNN branches with skip-connections in between and fuses the two feature maps in the final convolution layer.

### B. 3D Semantic Segmentation

1) **Feature-Level Fusion:** Dai and Nießner [109] presented 3DMV, a multi-view network for 3D semantic segmentation which fuse image semantic and point features in voxelized point cloud. Image features are extracted by 2D CNNs from multiple aligned images and projected back to the 3D space. These multi-view image features are max-pooled voxel-wise and fused with 3D geometry before feeding into the 3D CNNs for per-voxel semantic prediction. 3DMV out-performed other voxel-based approaches on ScanNet [110] benchmark. However, the performance of voxel-based approaches is determined by the voxel-resolution and hindered by voxel boundary artifacts.

To alleviate problems caused by point cloud voxelization, Chiang *et al.* [111] proposed a point-based semantic segmentation framework (UPF) that also enables efficient representation learning of image features, geometrical structures and global context priors. Features of rendered multi-view images are extracted using a semantic segmentation network and projected to 3D space for point-wise feature fusion. This fused point cloud is processed by two PointNet++ [30] based encoders to extract local and global features before feeding into a decoder for per-point semantic label prediction. Similarly, Multi-View

PointNet (MVPNet) [112] fused multi-view images semantics and 3D geometry to predict per-point semantic labels.

Permutohedral lattice representation is an alternative approach for multimodal data fusion and processing. Sparse Lattice Networks (SPLATNet) by Su *et al.* [113] employed sparse bilateral convolution to achieve spatial-aware representation learning and multimodal (image and point cloud) reasoning. In this approach, point cloud features are interpolated onto a  $d_l$ -dimensional permutohedral lattice, where bilateral convolution is applied. The results are interpolated back onto the point cloud. Image features are extracted from multi-view images using a CNN and projected to the 3D lattice space to be combined with 3D features. This fused feature map is further processed by CNN to predict the per-point label.

### C. Instance Segmentation

In essence, instance segmentation aims to perform semantic segmentation and object detection jointly. It extends the semantic segmentation task by discriminating against individual instances within a class, which makes it more challenging.

*1) Proposal Based:* Hou *et al.* presented 3D-SIS [114], a two-stage 3D CNN that performs voxel-wise 3D instance segmentation on multi-view images and RGB-D scan data. In the 3D detection stage, multi-view image features are extracted and down-sampled using ENet [115] based network. This down-sample process tackles the mismatch problem between a high-resolution image feature map and a low-resolution voxelized point cloud feature map. These down-sampled image feature maps are projected back to 3D voxel space and append to the corresponding 3D geometry features, which are then fed into a 3D CNN to predict object classes and 3D bbox poses. In the 3D mask stage, a 3D CNN takes images, point cloud features and 3D object detection results to predict per-voxel instance labels.

Narita *et al.* [116] extended 2D panoptic segmentation to perform scene reconstruction, 3D semantic segmentation and 3D instance segmentation jointly on RGB images and depth images. This approach takes RGB and depth frames as inputs for instance and 2D semantic segmentation networks. To track labels between frames, these frame-wise predicted panoptic annotations and corresponding depth are referenced by associating and integrating to the volumetric map. In the final step, a fully connected conditional random field (CRF) is employed to fine-tune the outputs. However, this approach does not support dynamic scenes and are vulnerable to long-term post drift.

*2) Proposal-Free Based:* Elich *et al.* [117] presented 3D-BEVIS, a framework that performs 3D semantic and instance segmentation tasks jointly using the clustering method on points aggregated with 2D semantics. 3D-BEVIS first extract global semantic scores map and instance feature map from 2D BEV representation (RGB and height-above-ground). These two semantic maps are propagated to points using a graph neural network. Finally, the mean shift algorithm [118] use these semantic features to cluster points into instances. This approach is mainly constraint by its dependence on

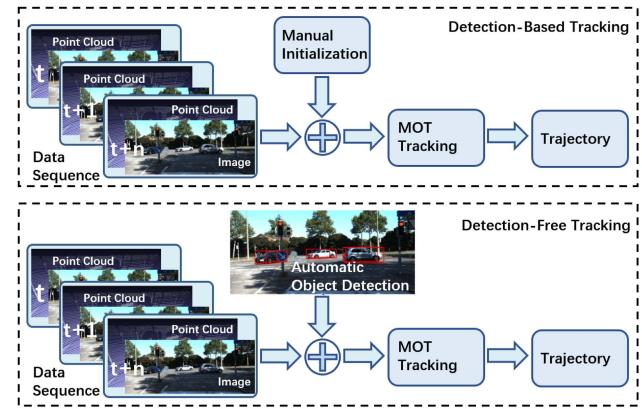


Fig. 10. A comparison between Detection-Based Tracking (DBT) and Detection-Free Tracking (DFT) approaches.

semantic features from BEV, which could introduce occlusions from sensor displacements.

## VII. OBJECTS TRACKING

Multiple object tracking (MOT) aims to maintain objects identities and track their location across data frames (over time), which is indispensable for the decision making of autonomous vehicles. To this end, this section reviews camera-LiDAR fusion based object tracking methods. Based on object initialization methods, MOT algorithms can be catalogued into Detection-Based Tracking (DBT) and Detection-Free Tracking (DFT) frameworks. DBT or Tracking-by-Detection framework leverages a sequence of object hypotheses and higher-level cues produced by an object detector to track objects. In DBT, objects are tracked via data (detections sequence) association or Multiple Hypothesis Tracking. On the contrary, the DFT framework is based on finite set statistics (FISST) for state estimation. Common methods include Multi-target Multi-Bernoulli (MeMBer) filter and Probability Hypothesis Density (PHD) filter. Table V presents the performance of different models on the KITTI multi-object tracking benchmark (car) [56]. Figure 10 provides a comparison between DBT and DFT approaches.

### A. Detection-Based Tracking (DBT)

The tracking-by-Detection framework consists of two stages. In the first stage, objects of interest are detected. The second stage associates these objects over time and formulates them into trajectories, which are formulated as linear programs. Frossard and Urtasun [119] presented an end-to-end trainable tracking-by-detection framework comprise of multiple independent networks that leverage both image and point cloud. This framework performs object detection, proposal matching and scoring, linear optimization consecutively. To achieve end-to-end learning, detection and matching are formulated via a deep structured model (DSM). Zhang *et al.* [120] presented a sensor-agnostic framework, which employs a loss-coupling scheme for image and point cloud fusion. Similar to [119], the framework consists of three stages, object detection, adjacency estimation and linear optimization. In the object detection stage, image and point cloud

TABLE V

COMPARATIVE RESULTS ON KITTI MULTI-OBJECT TRACKING BENCHMARK (CAR) [56]. MOTA STANDS FOR MULTIPLE OBJECT TRACKING ACCURACY. MOPT STANDS FOR MULTIPLE OBJECT TRACKING PRECISION. MT STANDS FOR MOSTLY TRACKED. ML STANDS FOR MOSTLY LOST. IDS STANDS FOR NUMBER OF ID SWITCHES. FRAG STANDS FOR NUMBER OF FRAGMENTS

Methods	Data-association	Models	Hardware	Run-time	MOTA(%)	MOTP(%)	MT(%)	ML(%)	IDS	FRAG
DBT	min-cost flow	DSM [119]	GTX1080Ti	0.1s	76.15	83.42	60.00	8.31	296	868
	min-cost flow	mm MOT [120]	GTX1080Ti	0.02s	84.77	85.21	73.23	2.77	284	753
	Hungarian	MOTSFusion [121]	GTX1080Ti	0.44s	84.83	85.21	73.08	2.77	275	759
DFT	RFS	Complexer-YOLO [122]	GTX1080Ti	0.01s	75.70	78.46	58.00	5.08	1186	2092

features are extracted via a VGG-16 [123] and a PointNet [30] in parallel and fused via a robust fusion module. The robust fusion module is designed to work with both a-modal and multi-modal inputs. The adjacency estimation stage extends min-cost flow to multi-modality via adjacent matrix learning. Finally, an optimal path is computed from the min-cost flow graph.

Tracking and 3D reconstruct tasks can be performed jointly. Extending this idea, Luiten *et al.* [121] leveraged 3D reconstruction to improve tracking, making tracking robust against complete occlusion. The proposed MOTSFusion consists of two stages. In the first stage, detected objects are associate with spatial-temporal tracklets. These tracklets are matched and merged into trajectories using the Hungarian algorithm. Furthermore, MOTSFusion can work with LiDAR mono and stereo depth.

### B. Detection-Free Tracking (DFT)

In DFT objects are manually initialized and tracked via filtering based methods. The complexer-YOLO [122] is a real-time framework for decoupled 3D object detection and tracking on image and point cloud data. In the 3D object detection phase, 2D semantics are extracted and fused point-wise to the point cloud. This semantic point cloud is voxelized and fed into a 3D complex-YOLO for 3D object detection. To speed up the training process, IoU is replaced by a novel metric called Scale-Rotation-Translation score (SRTs), which evaluates 3 DoFs of the bounding box position. Multi-object tracking is decoupled from the detection, and the inference is achieved via Labeled Multi-Bernoulli Random Finite Sets Filter (LMB RFS).

## VIII. ONLINE CROSS-SENSOR CALIBRATION

One of the preconditions of the camera-LiDAR fusion pipeline is a flawless registration/calibration between sensors, which can be difficult to satisfy. The calibration parameters between sensors change constantly due to mechanical vibration and heat fluctuation. As most fusion methods are extremely sensitive towards calibration errors, this could significantly cripple their performance and reliability. Furthermore, offline calibration is a troublesome and time-consuming procedure. Therefore, studies on online automatic cross-sensor calibration have significant practical benefits.

### A. Classical Online Calibration

Online calibration methods estimate extrinsic in natural settings without a calibration target. Many studies [124]–[127] find extrinsic by maximizing mutual information (MI) (raw

intensity value or edge intensity) between different modalities. However, MI-based methods are not robust against texture-rich environments, large de-calibrations and occlusions caused by sensor displacements. Alternatively, the LiDAR-enabled visual-odometry based method [128] uses the camera's ego-motion to estimate and evaluate camera-LiDAR extrinsic parameters. Nevertheless, [128] still struggles with large de-calibrations and cannot run in real-time.

### B. DL-Based Online Calibration

To mitigate the aforementioned challenges, Schneider *et al.* [129] designed a real-time capable CNN (RegNet) to estimate extrinsic, which is trained on randomly decalibrated data. The proposed RegNet extracts image and depth feature in two parallel branches and concatenates them to produce a fused feature map. This fused feature map is fed into a stack of Network in Network (NiN) modules and two fully connected layers for feature matching and global regression. However, the RegNet is agnostic towards the sensor's intrinsic parameters and needs to be retrained once these intrinsic changes. To solve this problem, the CalibNet [130] learns to minimize geometric and photometric inconsistencies between the miscalibrated and target depth in a self-supervised manner. Because intrinsics are only used during the 3D spatial transformer, the CalibNet can be applied to any intrinsically calibrated cameras. Nevertheless, deep learning based cross-sensor calibration methods are computationally expensive.

## IX. TRENDS, OPEN CHALLENGES AND PROMISING DIRECTIONS

The perception module in a driverless car is responsible for obtaining and understanding its surrounding scenes. Its down-stream modules, such as planning, decision making and self-localization, depend on its outputs. Therefore, its performance and reliability are the prerequisites for the competence of the entire driverless system. To this end, LiDAR and camera fusion is applied to improve the performance and reliability of the perception system, making driverless vehicles more capable in understanding complex scenes (e.g. urban traffic, extreme weather condition and so on). Consequently, in this section, we summarize overall trends and discuss open challenges and potential influencing factors in this regard. As shown in Table VI, we focus on improving the performance of fusion methodology and the robustness of the fusion pipeline.

From the methods reviewed above, we observed some general trends among the image and point cloud fusion approaches, which are summarized as the following:

TABLE VI  
OPEN CHALLENGES RELATED TO PERFORMANCE IMPROVEMENT, RELIABILITY ENHANCEMENT

Open Challenges:	Possible solutions / directions:
Performance-related Open Research Questions	
What should be the data representation of fused data?	Point Representation + Point Convolution
How to encode Temporal Context?	RNN/LSTM + Generative Models
What should be the learning scheme?	Unsupervised + Weakly-supervised learning
When to use deep learning methods?	On applications with explicit objectives that can be verified objectively
Reliability-related Open Research Questions	
How to mitigate camera-LiDAR coupling?	Sensor-agnostic framework
How to improve all-weather / Lighting conditions?	Datasets with complex weather / lighting conditions
How to handle adversarial attacks and corner cases?	Cross modality verification
How to solve open-set object detection?	Testing protocol, metrics + new framework
How to balance speed-accuracy trade-offs?	Models developed with scalability in mind

- 2D to 3D: Under the progressing of 3D feature extraction methods, to locate, track and segment objects in 3D space has become a heated area of research.
- Single-task to multi-tasks: Some recent works [73], [122] combined multiple complementary tasks, such as object detection, semantic segmentation and depth completion to achieve better overall performance and reduce computational costs.
- Signal-level to multi-level fusion: Early works often leverage signal-level fusion where 3D geometry is translated to the image plane to leverage off-the-shelf image processing models, while recent models try to fuse image and LiDAR in multi-level (e.g. early fusion, late fusion) and temporal context encoding.

#### A. Performance-Related Open Research Questions

1) *What Should Be the Data Representation of Fused Data?*: The choosing of data representations of the fused data plays a fundamental role in designing any data fusion algorithms. Current data representation for image and point cloud fusion includes:

- **Image Representation:** Append 3D geometry as additional channels of the image. The image-based representation enables off-the-shelf image processing models. However, the results are also limited in the 2D image plane, which is less ideal for autonomous driving.
- **Point Representation:** Append RGB signal/features as additional channels of the point cloud. However, the mismatch of resolution between high-resolution images and low-resolution point cloud leads to inefficiency.
- **Intermediate data representations:** Translate image and point cloud features/signal into an intermediate data representation, such as voxelized point cloud [82]. However, voxel-based methods suffer from bad scalability.

Many recent works for point cloud processing have concentrated on defining explicit point convolution operations [32], [33], [35]–[38], [62], which have shown great potentials. These point convolutions are better suited to extract fine-grained per-point and local geometry. Therefore, we believe the point representation of fused data coupled with point convolutions has great potentials in camera-LiDAR fusion studies.

2) *How to Encode Temporal Context?*: Most current deep learning based perception systems tend to overlook temporal context. This leads to numerous problems, such as point cloud deformation from low refresh rate and incorrect time-synchronization between sensors. These problems cause

mismatches between images, point cloud and actual environment. Therefore, it is vital to incorporate temporal context into the perception systems.

In the context of autonomous driving, temporal context can be incorporated using RNN or LSTM models. In [131], an LSTM auto-encoder is employed to estimate future states of surrounding vehicles and adjust the planned trajectory accordingly, which helps autonomous vehicles run smoother and more stable. In [121] temporal context is leveraged to estimate ego-motion, which benefits later task-related header networks. Furthermore, the temporal context could benefit online self-calibration via a visual-odometry based method [128]. Following this trend, the mismatches caused by LiDAR low refresh rate could be solved by encoding temporal context and generative models.

3) *What Should Be the Learning Scheme?*: Most current camera-LiDAR fusion methods rely on supervised learning, which requires large annotated datasets. However, annotating images and point clouds is expensive and time-consuming. This limits the size of the current multi-modal dataset and the performance of supervised learning methods.

The answer to this problem is unsupervised and weakly-supervised learning frameworks. Some recent studies have shown great potentials in this regards [24], [43], [50], [101], [132]. Following this trend, future researches in unsupervised and weakly-supervised learning fusion frameworks could allow the networks to be trained on large unlabeled/coarse-labelled dataset and leads to better performance.

4) *When to Use Deep Learning Methods?*: Recent advances in deep learning techniques have accelerated the development of autonomous driving technology. In many aspects, however, traditional methods are still indispensable in current autonomous driving systems. Compared with deep learning methods, traditional methods have better interpretability and consume significantly less computational resources. The ability to track back a decision is crucial for the decision making and planning system of an autonomous vehicle. Nevertheless, current deep learning algorithms are not back-traceable, making them unfit for these applications. Apart from this black-box dilemma, traditional algorithms are also preferred for their real-time capability.

To summarize, we believe deep learning methods should be applied to applications that have explicit objectives that can be verified objectively.

## B. Reliability-Related Open Research Questions

**1) How to Mitigate Camera-LiDAR Coupling?:** From an engineering perspective, redundancy design in an autonomous vehicle is crucial for its safety. Although fusing LiDAR and camera improves perception performance, it also comes with the problem of signal coupling. If one of the signal paths suddenly failed, the whole pipeline could break down and cripple down-stream modules. This is unacceptable for autonomous driving systems, which require robust perception pipelines.

To solve this problem, we should develop a sensor-agnostic framework. For instance, we can adopt multiple fusion modules with different sensor inputs. Furthermore, we can employ a multi-path fusion module that take asynchronous multi-modal data. However, the best solution is still open for study.

**2) How to Improve All-weather/Lighting Conditions?:** Autonomous vehicles need to work in all weather and lighting conditions. However, current datasets and methods are mostly focused on scenes with good lighting and weather conditions. This leads to bad performances in the real-world, where illumination and weather conditions are more complex.

The first step towards this problem is developing more datasets that contain a wide range of lighting and weather conditions. In addition, methods that employ multi-modal data to tackle complex lighting and weather conditions require further investigation.

**3) How to Handle Adversarial Attacks and Corner Cases?:** Adversarial attacks targeted at camera-based perception systems have proven to be effective. This poses a grave danger for the autonomous vehicle, as it operates in safety-critical environments. It may be difficult to identify attacks explicitly designed for certain sensory modality. However, perception results can be verified across different modalities. In this context, research on utilizing the 3D geometry and images to jointly identify these attacks can be further explored.

As self-driving cars operate in an unpredictable open environment with infinite possibilities, it is critical to consider corner and edge cases in the design of the perception pipeline. The perception system should anticipate unseen and unusual obstacles, strange behaviours and extreme weather. For instance, the image of cyclists printed on a large vehicle and people wearing costumes. These corner cases are often very difficult to handle using only the camera or LiDAR pipeline. However, leveraging data from multi-modalities to identify these corner cases could prove to be more effective and reliable than from a-modal sensors. Further researches in this direction could greatly benefit the safety and commercialization of autonomous driving technology.

**4) How to Solve Open-Set Object Detection?:** Open-set object detection is a scenario where an object detector is tested on instances from unknown/unseen classes. The open-set problem is critical for an autonomous vehicle, as it operates in unconstrained environments with infinite categories of objects. Current datasets often use a background class for any objects that are not interested. However, no dataset can include all the unwanted object categories in a background class. Therefore,

the behaviour of an object detector in an open-set setting is highly uncertain, which is less ideal for autonomous driving.

The lack of open-set object detection awareness, testing protocol and metrics leads to little explicit evaluations of the open-set performance in current object detection studies. These challenges have been discussed and investigated in a recent study by Dhamija *et al.* [133], where a novel open-set protocol and metrics are proposed. The authors proposed an additional mixed unknown category, which contains known 'background' objects and unknown/unseen objects. Based on this protocol, current methods are tested on a test set with a mixed unknown category generated from a combination of existing datasets. In another recent study on the point cloud, Wong *et al.* [134] proposed a technique that maps unwanted objects from different categories into a category-agnostic embedding space for clustering.

The open-set challenges are essential for deploying deep learning-based perception systems in the real-world. And it needs more effort and attention from the whole research community (dataset and methods with emphasis on unknown objects, test protocols and metrics, etc.,).

**5) How to Balance Speed-Accuracy Trade-offs?:** The processing of multiple high-resolution images and large-scale point clouds put substantial stress on existing mobile computing platforms. This sometimes causes frame-drop, which could seriously degrade the performance of the perception system. More generally, it leads to high-power consumption and low reliability. Therefore, it is important to balance the speed and accuracy of a model in real-world deployments.

There are studies that attempt to detect the frame-drop. In [135], Imre *et al.* proposed a multi-camera frame-drop detection algorithm that leverages multiple segments (broken line) fitting on camera pairs. However, frame-drop detection only solves half of the problem. The hard-part is to prevent the performance degradation caused by the frame-drop. Recent advances in generative models have demonstrated great potentials to predict missing frames in video sequences [136], which could be leveraged in autonomous driving for filling the missing frames in the image and the point cloud pipelines. However, we believe the most effective way to solve the frame-drop problem is to prevent it by reducing the hardware workload. This can be achieved by carefully balancing the speed and accuracy of a model [137].

To achieve this, deep learning models should be able to scale down its computational cost, while maintaining acceptable performance. This scalability is often achieved by reducing the number of inputs (points, pixels, voxels) or the depth of the network. From previous studies [30], [38], [138], point-based and multi-view based fusion methods are more scalable compared to voxel-based methods.

## X. CONCLUSION

This article presented an in-depth review of the most recent progress on deep learning models for point cloud and image fusion in the context of autonomous driving. Specifically, this review organizes methods based on their fusion methodologies and covers topics in depth completion, dynamic and

stationary object detection, semantic segmentation, tracking and online cross-sensor calibration. Furthermore, performance comparisons on the publicly available dataset, highlights and advantages/disadvantages of models are presented in tables. Typical model architectures are shown in figures. Finally, we summarized general trends and discussed open challenges and possible future directions. This survey also raised awareness and provided insights on questions that are overlooked by the research community but troubles real-world deployment of the autonomous driving technology.

## REFERENCES

- [1] F. Duarte, "Self-driving cars: A city perspective," *Sci. Robot.*, vol. 4, no. 28, pp. 5–6, 2019. [Online]. Available: <https://robotics.sciencemag.org/content/4/28/eaav9843>
- [2] J. Guo, U. Kurup, and M. Shah, "Is it safe to drive? An overview of factors, metrics, and datasets for driveability assessment in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3135–3151, Aug. 2019.
- [3] Y. E. Bigman and K. Gray, "Life and death decisions of autonomous vehicles," *Nature*, vol. 579, no. 7797, pp. E1–E2, Mar. 2020.
- [4] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.
- [5] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: [10.1038/nature14236](https://doi.org/10.1038/nature14236).
- [6] P. Huang, M. Cheng, Y. Chen, H. Luo, C. Wang, and J. Li, "Traffic sign occlusion detection using mobile laser scanning point clouds," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2364–2376, Sep. 2017.
- [7] L. Chen *et al.*, "Surrounding vehicle detection using an FPGA panoramic camera and deep CNNs," *IEEE Trans. Intell. Transp. Syst.*, early access, Oct. 29, 2019, doi: [10.1109/TITS.2019.2949005](https://doi.org/10.1109/TITS.2019.2949005).
- [8] S. Kim, H. Kim, W. Yoo, and K. Huh, "Sensor fusion algorithm design in detecting vehicles using laser scanner and stereo vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1072–1084, Apr. 2016.
- [9] K. Park, S. Kim, and K. Sohn, "High-precision depth estimation using uncalibrated LiDAR and stereo fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 321–335, Jan. 2020.
- [10] J. Zhu, L. Fan, W. Tian, L. Chen, D. Cao, and F. Wang, "Toward the ghosting phenomenon in a stereo-based map with a collaborative RGB-D repair," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2739–2749, Jul. 2019.
- [11] J.-G. Wang and L.-B. Zhou, "Traffic light recognition with high dynamic range imaging and deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1341–1352, Apr. 2019.
- [12] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *IEEE Access*, vol. 8, pp. 2847–2868, 2020.
- [13] D. Feng *et al.*, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 17, 2020, doi: [10.1109/tits.2020.2972974](https://doi.org/10.1109/tits.2020.2972974).
- [14] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499, doi: [10.1109/cvpr.2018.00472](https://doi.org/10.1109/cvpr.2018.00472).
- [15] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.
- [16] Y. Li, L. Ma, Z. Zhong, D. Cao, and J. Li, "TGNet: Geometric graph CNN on 3-D point cloud segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3588–3600, May 2020.
- [17] C. R. Qi, H. Su, M. NieBner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5648–5656, doi: [10.1109/CVPR.2016.609](https://doi.org/10.1109/CVPR.2016.609).
- [18] W. Zeng and T. Gevers, "3DConTextNet: K-d tree guided hierarchical learning of point clouds using local and global contextual cues," in *Proc. Comput. Vis. (ECCV) Workshops*, L. Leal-Taixé and S. Roth, Eds. Cham, Switzerland: Springer, 2019, pp. 314–330.
- [19] R. Klokov and V. Lempitsky, "Escape from cells: Deep KD-networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 863–872, doi: [10.1109/ICCV.2017.99](https://doi.org/10.1109/ICCV.2017.99).
- [20] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3577–3586.
- [21] D. C. Garcia, T. A. Fonseca, R. U. Ferreira, and R. L. de Queiroz, "Geometry coding for dynamic voxelized point clouds using octrees and multiple contexts," *IEEE Trans. Image Process.*, vol. 29, pp. 313–322, 2020.
- [22] H. Lei, N. Akhtar, and A. Mian, "Octree guided CNN with spherical kernels for 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9631–9640.
- [23] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.
- [24] S. Chen, C. Duan, Y. Yang, D. Li, C. Feng, and D. Tian, "Deep unsupervised learning of 3D point clouds via graph topology inference and filtering," *IEEE Trans. Image Process.*, vol. 29, pp. 3183–3198, 2020.
- [25] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," 2015, *arXiv:1506.05163*. [Online]. Available: <https://arxiv.org/abs/1506.05163>
- [26] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3693–3702.
- [27] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," *Comput. Graph. Forum*, vol. 34, no. 5, pp. 13–23, Aug. 2015. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12693>
- [28] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Apr. 2014, pp. 1–14.
- [29] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
- [30] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.
- [31] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660, doi: [10.1109/CVPR.2017.16](https://doi.org/10.1109/CVPR.2017.16).
- [32] B.-S. Hua, M.-K. Tran, and S.-K. Yeung, "Pointwise convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 984–993.
- [33] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by extension operators," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–12, Aug. 2018, doi: [10.1145/3197517.3201301](https://doi.org/10.1145/3197517.3201301).
- [34] L. Ma, Y. Li, J. Li, W. Tan, Y. Yu, and M. A. Chapman, "Multi-scale point-wise convolutional neural networks for 3D object segmentation from LiDAR point clouds in large-scale environments," *IEEE Trans. Intell. Transp. Syst.*, early access, Dec. 27, 2019, doi: [10.1109/TITS.2019.2961060](https://doi.org/10.1109/TITS.2019.2961060).
- [35] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Advances in Neural Information Processing Systems*, vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. New York, NY, USA: Curran Associates, Inc., 2018, pp. 820–830. [Online]. Available: <https://proceedings.neurips.cc/paper/2018/file/f5f8590cd58a54e94377e6ae2ded4d9-Paper.pdf>
- [36] S. Lan, R. Yu, G. Yu, and L. S. Davis, "Modeling local geometric structure of 3D point clouds using geo-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 998–1008, doi: [10.1109/CVPR.2019.00109](https://doi.org/10.1109/CVPR.2019.00109).
- [37] F. Groh, P. Wieschollek, and H. P. A. Lenzsch, "Flex-convolution," in *Proc. 14th Asian Conf. Comput. Vis. Comput. Vis. (ACCV)*, in Lecture Notes in Computer Science, vol. 11361, C. V. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. New York, NY, USA: Springer, Dec. 2019, pp. 105–122. [Online]. Available: [http://dx.doi.org/10.1007/978-3-030-20887-5\\_7](http://dx.doi.org/10.1007/978-3-030-20887-5_7)
- [38] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6411–6420, doi: [10.1109/ICCV.2019.000651](https://doi.org/10.1109/ICCV.2019.000651).
- [39] Q. Hu *et al.*, "RandLA-net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11108–11117.

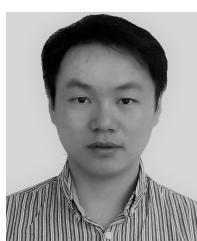
- [40] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9621–9630, doi: [10.1109/CVPR.2019.00985](https://doi.org/10.1109/CVPR.2019.00985).
- [41] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 11–20.
- [42] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–8, doi: [10.1109/ICRA.2018.8460184](https://doi.org/10.1109/ICRA.2018.8460184).
- [43] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised Sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3288–3295, doi: [10.1109/ICRA.2019.8793637](https://doi.org/10.1109/ICRA.2019.8793637).
- [44] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 108–125. [Online]. Available: [http://dx.doi.org/10.1007/978-3-030-01270-0\\_7](http://dx.doi.org/10.1007/978-3-030-01270-0_7)
- [45] X. Cheng, P. Wang, G. Chenye, and R. Yang, "Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion," in *Proc. 24th AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 10615–10622.
- [46] M. Jaritz, R. D. Charette, E. Wirbel, X. Perrotton, and F. Nashashibi, "Sparse and dense data with CNNs: Depth completion and semantic segmentation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 52–60, doi: [10.1109/3DV.2018.00017](https://doi.org/10.1109/3DV.2018.00017).
- [47] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through CNNs for guided sparse depth regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2423–2436, Oct. 2020, doi: [10.1109/TPAMI.2019.2929170](https://doi.org/10.1109/TPAMI.2019.2929170).
- [48] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning guided convolutional network for depth completion," 2019, *arXiv:1908.01238*. [Online]. Available: <https://arxiv.org/abs/1908.01238>
- [49] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy LiDAR completion with RGB guidance and uncertainty," in *Proc. 16th Int. Conf. Mach. Vis. Appl. (MVA)*, May 2019, pp. 1–6, doi: [10.23919/MVA.2019.8757939](https://doi.org/10.23919/MVA.2019.8757939).
- [50] X. Cheng, Y. Zhong, Y. Dai, P. Ji, and H. Li, "Noise-aware unsupervised deep lidar-stereo fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6339–6348, doi: [10.1109/CVPR.2019.00650](https://doi.org/10.1109/CVPR.2019.00650).
- [51] J. Zhang, M. S. Ramanagopal, R. Vasudevan, and M. Johnson-Roberson, "Listereo: Generate dense depth maps from lidar and stereo imagery," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2020, pp. 7829–7836.
- [52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [53] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710, doi: [10.1109/CVPR.2018.00907](https://doi.org/10.1109/CVPR.2018.00907).
- [54] T.-H. Wang, F.-E. Wang, J.-T. Lin, Y.-H. Tsai, W.-C. Chiu, and M. Sun, "Plug-and-play: Improve depth prediction via sparse data propagation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5880–5886.
- [55] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *Int. J. Comput. Vis.*, vol. 128, no. 5, pp. 1239–1285, Jul. 2019, doi: [10.1007/s11263-019-01188-y](https://doi.org/10.1007/s11263-019-01188-y).
- [56] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
- [57] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 918–927, doi: [10.1109/CVPR.2018.00102](https://doi.org/10.1109/CVPR.2018.00102).
- [58] X. Du, M. H. Ang, S. Karaman, and D. Rus, "A general pipeline for 3D detection of vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3194–3200, doi: [10.1109/ICRA.2018.8461232](https://doi.org/10.1109/ICRA.2018.8461232).
- [59] K. Shin, Y. P. Kwon, and M. Tomizuka, "RoarNet: A robust 3D object detection based on RegiOn approximation refinement," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 2510–2515, doi: [10.1109/IVS.2019.8813895](https://doi.org/10.1109/IVS.2019.8813895).
- [60] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7074–7082, doi: [10.1109/CVPR.2017.597](https://doi.org/10.1109/CVPR.2017.597).
- [61] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "Ipod: Intensive point-based object detector for point cloud," 2018, *arXiv:1812.05276*. [Online]. Available: <https://arxiv.org/abs/1812.05276>
- [62] D. Xu, D. Anguelov, and A. Jain, "PointFusion: Deep sensor fusion for 3D bounding box estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 244–253, doi: [10.1109/CVPR.2018.00033](https://doi.org/10.1109/CVPR.2018.00033).
- [63] X. Zhao, Z. Liu, R. Hu, and K. Huang, "3D object detection using scale invariant and feature reweighting networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 9267–9274.
- [64] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu, "Pointsift: A sift-like network module for 3d point cloud semantic segmentation," 2018, *arXiv:1807.00652*. [Online]. Available: <https://arxiv.org/abs/1807.00652>
- [65] Z. Wang and K. Jia, "Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1–8, doi: [10.1109/IROS40897.2019.8968513](https://doi.org/10.1109/IROS40897.2019.8968513).
- [66] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4604–4612.
- [67] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1907–1915, doi: [10.1109/CVPR.2017.691](https://doi.org/10.1109/CVPR.2017.691).
- [68] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–8, doi: [10.1109/IROS.2018.8594049](https://doi.org/10.1109/IROS.2018.8594049).
- [69] H. Lu, X. Chen, G. Zhang, Q. Zhou, Y. Ma, and Y. Zhao, "Scannet: Spatial-channel attention network for 3D object detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 1992–1996.
- [70] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 641–656.
- [71] V. A. Sindagi, Y. Zhou, and O. Tuzel, "MVX-net: Multimodal Voxelnet for 3D object detection," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 7276–7282, doi: [10.1109/ICRA.2019.8794195](https://doi.org/10.1109/ICRA.2019.8794195).
- [72] Z. Wang, W. Zhan, and M. Tomizuka, "Fusing Bird's eye view LiDAR point cloud and front view camera image for 3D object detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1–6.
- [73] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7345–7353.
- [74] G. P. Meyer, J. Charland, D. Hegde, A. Laddha, and C. Vallespi-Gonzalez, "Sensor fusion for joint 3d object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019.
- [75] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Computer Vision*, in (Lecture Notes in Computer Science). New York, NY, USA: Springer, 2014, pp. 345–360, doi: [10.1007/978-3-319-10584-0\\_23](https://doi.org/10.1007/978-3-319-10584-0_23).
- [76] S. Gupta, J. Hoffman, and J. Malik, "Cross modal distillation for supervision transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2827–2836, doi: [10.1109/CVPR.2016.309](https://doi.org/10.1109/CVPR.2016.309).
- [77] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587, doi: [10.1109/CVPR.2014.81](https://doi.org/10.1109/CVPR.2014.81).
- [78] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik, "Aligning 3D models to RGB-D images of cluttered scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4731–4740.
- [79] J. Schlosser, C. K. Chow, and Z. Kira, "Fusing LiDAR and images for pedestrian detection using convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 2198–2205.
- [80] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779, doi: [10.1109/CVPR.2019.00086](https://doi.org/10.1109/CVPR.2019.00086).
- [81] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705, doi: [10.1109/CVPR.2019.01298](https://doi.org/10.1109/CVPR.2019.01298).

- [82] S. Song and J. Xiao, "Sliding shapes for 3D object detection in depth images," in *Computer Vision*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 634–651.
- [83] S. Wang, S. Suo, W.-C. Ma, A. Pokrovsky, and R. Urtasun, "Deep parametric continuous convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2589–2597.
- [84] S. Song and J. Xiao, "Deep sliding shapes for amodal 3D object detection in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 808–816.
- [85] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "LaserNet: An efficient probabilistic 3D object detector for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12677–12686, doi: [10.1109/CVPR.2019.01296](https://doi.org/10.1109/CVPR.2019.01296).
- [86] M. Bai, G. Mattyus, N. Homayounfar, S. Wang, S. K. Lakshminarayanan, and R. Urtasun, "Deep multi-sensor lane detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3102–3109.
- [87] F. Wulff, B. Schaufele, O. Sawade, D. Becker, B. Henke, and I. Radusch, "Early fusion of camera and lidar for robust road detection based on U-Net FCN," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1426–1431.
- [88] X. Lv, Z. Liu, J. Xin, and N. Zheng, "A novel approach for detecting road based on two-stream fusion fully convolutional network," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1464–1469.
- [89] D. Yu, H. Xiong, Q. Xu, J. Wang, and K. Li, "Multi-stage residual fusion network for LIDAR-camera road detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 2323–2328.
- [90] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "LIDAR-camera fusion for road detection using fully convolutional neural networks," *Robot. Auto. Syst.*, vol. 111, pp. 125–131, Jan. 2019, doi: [10.1016/j.robot.2018.11.002](https://doi.org/10.1016/j.robot.2018.11.002).
- [91] Z. Chen, J. Zhang, and D. Tao, "Progressive LiDAR adaptation for road detection," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 3, pp. 693–702, May 2019.
- [92] J.-S. Lee and T.-H. Park, "Fast road detection by CNN-based camera-lidar fusion and spherical coordinate transformation," *IEEE Trans. Intell. Transp. Syst.*, early access, May 6, 2020, doi: [10.1109/TITS.2020.2988302](https://doi.org/10.1109/TITS.2020.2988302).
- [93] L. Ma, Y. Li, J. Li, C. Wang, R. Wang, and M. Chapman, "Mobile laser scanned point-clouds for road object detection and extraction: A review," *Remote Sens.*, vol. 10, no. 10, p. 1531, Sep. 2018.
- [94] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, "A review of recent advances in lane detection and departure warning system," *Pattern Recognit.*, vol. 73, pp. 216–234, Jan. 2018.
- [95] Y. Xing *et al.*, "Advances in vision-based lane detection: Algorithms, integration, assessment, and perspectives on ACP-based parallel vision," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 3, pp. 645–661, May 2018.
- [96] A. S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller, "Finding multiple lanes in urban road networks with vision and lidar," *Auto. Robots*, vol. 26, nos. 2–3, pp. 103–122, Apr. 2009.
- [97] P. Y. Shintzato, D. F. Wolf, and C. Stiller, "Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion," in *Proc. IEEE Intell. Vehicles Symp. Proc.*, Jun. 2014, pp. 687–692.
- [98] L. Xiao, R. Wang, B. Dai, Y. Fang, D. Liu, and T. Wu, "Hybrid conditional random field based camera-LIDAR fusion for road detection," *Inf. Sci.*, vol. 432, pp. 543–558, Mar. 2018.
- [99] L. Xiao, B. Dai, D. Liu, T. Hu, and T. Wu, "CRF based road detection with multi-sensor fusion," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 192–198.
- [100] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1484–1497, Dec. 2012.
- [101] M. Tan, B. Wang, Z. Wu, J. Wang, and G. Pan, "Weakly supervised metric learning for traffic sign recognition in a LIDAR-equipped vehicle," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1415–1427, May 2016.
- [102] Y. Yu, J. Li, C. Wen, H. Guan, H. Luo, and C. Wang, "Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 113, pp. 106–123, Mar. 2016.
- [103] M. Soilán, B. Riveiro, J. Martínez-Sánchez, and P. Arias, "Traffic sign detection in MLS acquired point clouds for geometric and image-based semantic inventory," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 92–101, Apr. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0924271616000368>
- [104] A. Arcos-García, M. Soilán, J. A. Álvarez-García, and B. Riveiro, "Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems," *Expert Syst. Appl.*, vol. 89, pp. 286–295, Dec. 2017.
- [105] H. Guan *et al.*, "A convolutional capsule network for traffic-sign recognition using mobile LiDAR data with digital images," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 6, pp. 1067–1071, Jun. 2020.
- [106] Z. Deng and L. Zhou, "Detection and recognition of traffic planar objects using colorized laser scan and perspective distortion rectification," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1485–1495, May 2018.
- [107] H. Guan, W. Yan, Y. Yu, L. Zhong, and D. Li, "Robust traffic-sign detection and classification using mobile LiDAR data with digital images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 5, pp. 1715–1724, May 2018.
- [108] C. Premebida, J. Carreira, J. Batista, and U. Nunes, "Pedestrian detection combining rgb and dense lidar data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 4112–4117.
- [109] A. Dai and M. Nießner, "3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)* (Lecture Notes in Computer Science), 2018, pp. 458–474. [Online]. Available: [http://dx.doi.org/10.1007/978-3-030-01249-6\\_28](http://dx.doi.org/10.1007/978-3-030-01249-6_28)
- [110] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5828–5839, doi: [10.1109/CVPR.2017.261](https://doi.org/10.1109/CVPR.2017.261).
- [111] H.-Y. Chiang, Y.-L. Lin, Y.-C. Liu, and W. H. Hsu, "A unified point-based framework for 3D segmentation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 155–163, doi: [10.1109/3DV.2019.00026](https://doi.org/10.1109/3DV.2019.00026).
- [112] M. Jaritz, J. Gu, and H. Su, "Multi-view PointNet for 3D scene understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, 2019, pp. 3995–4003, doi: [10.1109/ICCVW.2019.00494](https://doi.org/10.1109/ICCVW.2019.00494).
- [113] H. Su *et al.*, "SPLATNet: Sparse lattice networks for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2530–2539, doi: [10.1109/CVPR.2018.00268](https://doi.org/10.1109/CVPR.2018.00268).
- [114] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4421–4430, doi: [10.1109/CVPR.2019.00455](https://doi.org/10.1109/CVPR.2019.00455).
- [115] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*. [Online]. Available: <https://arxiv.org/abs/1606.02147>
- [116] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "PanopticFusion: Online volumetric semantic mapping at the level of stuff and things," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1–8, doi: [10.1109/IROS40897.2019.8967890](https://doi.org/10.1109/IROS40897.2019.8967890).
- [117] C. Elich, F. Engelmann, T. Kontogianni, and B. Leibe, "3d bird's-eye-view instance segmentation," in *Proc. German Conf. Pattern Recognit.*, 2019, pp. 48–61. [Online]. Available: [http://dx.doi.org/10.1007/978-3-03-33676-9\\_4](http://dx.doi.org/10.1007/978-3-03-33676-9_4)
- [118] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [119] D. Frossard and R. Urtasun, "End-to-end learning of multi-sensor 3D tracking by detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 635–642, doi: [10.1109/ICRA.2018.8462884](https://doi.org/10.1109/ICRA.2018.8462884).
- [120] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2365–2374, doi: [10.1109/ICCV.2019.00245](https://doi.org/10.1109/ICCV.2019.00245).
- [121] J. Luiten, T. Fischer, and B. Leibe, "Track to reconstruct and reconstruct to track," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, p. 1803–1810, Apr. 2020. [Online]. Available: <http://dx.doi.org/10.1109/LRA.2020.2969183>
- [122] M. Simon *et al.*, "Complexer YOLO: Real-time 3D object detection and tracking on semantic point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2019, pp. 1–10.
- [123] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.
- [124] A. Napier, P. Corke, and P. Newman, "Cross-calibration of push-broom 2D LIDARs and cameras in natural scenes," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3679–3684.

- [125] Z. Taylor and J. Nieto, "Automatic calibration of lidar and camera images using normalized mutual information," in *Proc. IEEE Int. Conf. Proc. Robot. Autom. (ICRA)*, 2013, pp. 1–8.
- [126] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic extrinsic calibration of vision and lidar by maximizing mutual information," *J. Field Robot.*, vol. 32, no. 5, pp. 696–722, Aug. 2015.
- [127] M. Miled, B. Soheilian, E. Habets, and B. Vallet, "Hybrid online mobile laser scanner calibration through image alignment by mutual information," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. III1, pp. 25–31, Jun. 2016. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2016ISPAnIII...25M>, doi: [10.5194/isprs-annals-III-1-25-2016](https://doi.org/10.5194/isprs-annals-III-1-25-2016).
- [128] H.-J. Chien, R. Klette, N. Schneider, and U. Franke, "Visual odometry driven online calibration for monocular LiDAR-camera systems," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 2848–2853.
- [129] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "RegNet: Multimodal sensor registration using deep neural networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1803–1810, doi: [10.1109/IVS.2017.7995968](https://doi.org/10.1109/IVS.2017.7995968).
- [130] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, "CalibNet: Geometrically supervised extrinsic calibration using 3D spatial transformer networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1110–1117.
- [131] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-Sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1672–1678.
- [132] J. Weng, N. Ahuja, and T. S. Huang, "Cresceptron: A self-organizing neural network which grows adaptively," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 1, Jun. 1992, pp. 576–581.
- [133] A. R. Dhamija, M. Gunther, J. Ventura, and T. E. Boult, "The overlooked elephant of object detection: Open set," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1010–1019.
- [134] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun, "Identifying unknown instances for autonomous driving," in *Proc. Conf. Robot Learn.*, 2020, pp. 384–393.
- [135] E. Imre, J.-Y. Guillemaut, and A. Hilton, "Through-the-Lens multi-camera synchronisation and frame-drop detection for 3D reconstruction," in *Proc. 2nd Int. Conf. 3D Imag., Model., Process., Vis. Transmiss.*, Oct. 2012, pp. 395–402.
- [136] Y. Li, D. Roblek, and M. Tagliasacchi, "From here to there: Video inbetweening using direct 3D convolutions," 2019, *arXiv:1905.10240*. [Online]. Available: <http://arxiv.org/abs/1905.10240>
- [137] J. Huang *et al.*, "Speed/Accuracy trade-offs for modern convolutional object detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7310–7311, doi: [10.1109/CVPR.2017.351](https://doi.org/10.1109/CVPR.2017.351).
- [138] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920, doi: [10.1109/CVPR.2015.7298801](https://doi.org/10.1109/CVPR.2015.7298801).



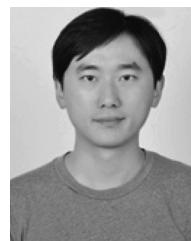
**Yaodong Cui** (Student Member, IEEE) received the B.S. degree in automation from Chang'an University, Xi'an, China, in 2017, and the M.Sc. degree in systems, control and signal processing from the University of Southampton, Southampton, U.K., in 2019. He is currently pursuing the Ph.D. degree with the Waterloo Cognitive Autonomous Driving (CogDrive) Laboratory, Department of Mechanical Engineering, University of Waterloo, Waterloo, Canada. His research interests include sensor fusion, perception for the intelligent vehicle, and driver emotion detection.



**Ren Chen** received the B.S. degree in electronic science and technology from the Huazhong University of Science and Technology, Wuhan, China, and the M.Sc. degree in computer science from the China Academy of Aerospace Science and Engineering, Beijing, China. He worked at the Institute of Deep Learning, Baidu, and the Autonomous Driving Laboratory, Tencent. His research interests include computer vision, perception for robotics, real-time dense 3-D reconstruction, and robotic motion planning.



Wenbo Chu received the B.S. degree in automotive engineering from Tsinghua University, Beijing, China, in 2008, the M.S. degree in automotive engineering from RWTH Aachen University, Germany, and the Ph.D. degree in mechanical engineering from Tsinghua University, in 2014. He is currently a Research Fellow with China Intelligent and Connected Vehicles (Beijing) Research Institute Company, Ltd., Beijing, which is also known as the National Innovation Center of Intelligent and Connected Vehicles. His research interests include vehicles, new energy vehicles, and vehicle dynamics and control.



**Long Chen** (Senior Member, IEEE) received the B.Sc. degree in communication engineering and the Ph.D. degree in signal and information processing from Wuhan University, Wuhan, China. He is currently an Associate Professor with School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His research interests include autonomous driving, robotics, and artificial intelligence, where he has contributed more than 70 publications. He serves as an Associate Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.



**Daxin Tian** (Senior Member, IEEE) is currently a Professor with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests include mobile computing, intelligent transportation systems, vehicular *ad hoc* networks, and swarm intelligent. He is a member of the IEEE Intelligent Transportation Systems Society and the IEEE Vehicular Technology Society.



**Ying Li** (Graduate Student Member, IEEE) received the M.Sc. degree in remote sensing from Wuhan University, China, in 2017. She is currently pursuing the Ph.D. degree with the Mobile Sensing and Geodata Science Laboratory, Department of Geography and Environmental Management, University of Waterloo, Canada. Her research interests include autonomous driving, mobile laser scanning, intelligent processing of point clouds, geometric and semantic modeling, and augmented reality.



**Dongpu Cao** (Member, IEEE) received the Ph.D. degree from Concordia University, Canada, in 2008. He is currently the Canada Research Chair of Driver Cognition and Automated Driving and an Associate Professor and the Director of the Waterloo Cognitive Autonomous Driving (CogDrive) Laboratory, University of Waterloo, Canada. He has contributed more than 200 articles and three books. His current research interests include driver cognition, automated driving, and cognitive autonomous driving. He received the SAE Arch T. Colwell Merit Award in 2012, the IEEE VTS 2020 Best Vehicular Electronics Paper Award, and three best paper awards from ASME and IEEE conferences. He serves as the Deputy Editor-in-Chief for *IET Intelligent Transport Systems* and an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE/ASME TRANSACTIONS ON MECHATRONICS, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, the IEEE/CAA JOURNAL OF AUTOMATICA SINICA, the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, and *Journal of Dynamic Systems, Measurement and Control* (ASME). He was a Guest Editor of *Vehicle System Dynamics*, the IEEE TRANSACTIONS ON SMC: SYSTEMS, and the IEEE INTERNET OF THINGS JOURNAL. He serves on the SAE Vehicle Dynamics Standards Committee and acts as the Co-Chair of IEEE ITSS Technical Committee on Cooperative Driving. He is also an IEEE VTS Distinguished Lecturer.