Wu *et al. Intell Robot* 2022;2(2):105-29
DOI: 10.20517/ir.2021.20

Intelligence & Robotics

Review

Open Access

Deep learning for LiDAR-only and LiDAR-fusion 3D perception: a survey

Danni Wu, Zichen Liang, Guang Chen

School of Automotive Studies, Tongji University, Shanghai 201804, China.

Correspondence to: Prof. Guang Chen, School of Automotive Studies, Tongji University, 4800 Caoan Road, Shanghai 201804, China. E-mail: guangchen@tongji.edu.cn

How to cite this article: Wu D, Liang Z, Chen G. Deep learning for LiDAR-only and LiDAR-fusion 3D perception: a survey. *Intell Robot* 2022;2(2):105-29. http://dx.doi.org/10.20517/ir.2021.20

Received: 31 Dec 2021 **First Decision:** 25 Feb 2022 **Revised:** 10 Mar 2022 **Accepted:** 16 Mar 2022 **Published:** 25 Apr 2022

Academic Editors: Simon X. Yang, Lei Lei **Copy Editor:** Jin-Xin Zhang **Production Editor:** Jin-Xin Zhang

Abstract

The perception system for robotics and autonomous cars relies on the collaboration among multiple types of sensors to understand the surrounding environment. LiDAR has shown great potential to provide accurate environmental information, and thus deep learning on LiDAR point cloud draws increasing attention. However, LiDAR is unable to handle severe weather. The sensor fusion between LiDAR and other sensors is an emerging topic due to its supplementary property compared to a single LiDAR. Challenges exist in deep learning methods that take LiDAR point cloud fusion data as input, which need to seek a balance between accuracy and algorithm complexity due to data redundancy. This work focuses on a comprehensive survey of deep learning on LiDAR-only and LiDAR-fusion 3D perception tasks. Starting with the representation of LiDAR point cloud, this paper then introduces its unique characteristics and the evaluation dataset as well as metrics. This paper gives a review according to four key tasks in the field of LiDAR-based perception: object classification, object detection, object tracking, and segmentation (including semantic segmentation and instance segmentation). Finally, we present the overlooked aspects of the current algorithms and possible solutions, hoping this paper can serve as a reference for the related research.

Keywords: LiDAR, sensor fusion, object classification, object detection, object tracking, segmentation

1. INTRODUCTION

The perception system is crucial for autonomous driving, which enables the autonomous car to understand the surrounding environment with the location, velocity, and future state of pedestrians, obstacles, and other traffic participants. It provides basic and essential information for downstream tasks of autonomous driving (i.e., decision making, planning, and control system). Thus, a precise perception system is vital, which depends on breakthroughs in both hardware and software, i.e., 2D and 3D acquisition technology and perception algorithms.

Sensors equipped on the perception system generally include 2D cameras, RGB-D cameras, radar, and LiDAR. With advantages such as high angular resolution, clear detail recognition, and long-range detection, LiDAR thus becomes indispensable in autonomous driving above the L3 level. LiDAR utilizes pulses of light to translate the physical world into a 3D point cloud in real time with a high level of confidence. By measuring the propagation distance between the LiDAR emitter and the target object and analyzing the reflected energy magnitude, amplitude, frequency, and phase of the reflected wave spectrum on the surface of the target object, LiDAR can present the precise 3D structural information of the target object within centimeter level. According to the scanning mechanism, LiDAR can be divided into three categories: the standard spindle-type, solid-state LiDAR (MEMS), and flash LiDAR. Compared with the standard spindle-type LiDAR, solid-state LiDAR and flash LiDAR provide a solution to high material cost and high mass production cost; therefore, the standard spindle-type LiDAR will be replaced gradually in the future. The application of LiDAR in autonomous cars is gradually gaining market attention. According to Sullivan's statistics and forecasts, the LiDAR market in the automotive segment is expected to reach $8 billion by 2025, accounting for 60% of the total.

In recent decades, deep learning has been attracting extensive attention from computer vision researchers due to its outstanding ability in dealing with massive and unstructured data, which stimulates the growth of environment perception algorithms for autonomous driving. Depending on whether the algorithm concerns the position and pose of the object in real 3D space or just the position of the object in the reflected plane (i.e., image plane), deep learning-based perception algorithms can be divided into 3D and 2D perception. While deep learning-based 2D perception has achieved great progress and thus become a mature branch in the field of computer vision, 3D perception is an emerging topic and yet under-investigated. Relatively, 3D perception outputs abundant information, i.e., height, length, width, and semantic label for each 3D object, to restore the real state of the object in three-dimensional space. In general, the input data of 3D perception tasks contain RGB-D images from depth cameras, images from monocular cameras, binocular cameras, and multi-cameras, and point clouds from LiDAR scanning. Among them, data from LiDAR and multi-camera-based stereo-vision systems achieve higher accuracy in 3D inference. Unlike images from stereo-vision systems, LiDAR point clouds as a relatively new data structure are unordered and possess interaction among points as well as invariance under transformation. These characteristics make deep learning on LiDAR point clouds more challenging. The publication of the pioneering framework PointNet[1] together with PointNet++[2] inspires plenty of works on deep learning for LiDAR point clouds, which will promote the development of autonomous driving perception systems. Hence, this work gives a review of 3D perception algorithms based on deep learning for LiDAR point cloud. However, in real-world applications, a single LiDAR sensor always struggles in heavy weather, color-related detection, and lightly disturbed conditions, which does not fulfill the need of autonomous cars that must perceive surroundings accurately and robustly in all variable and complex conditions. To overcome the shortcomings of a single LiDAR, LiDAR-based fusion[3,4] emerges with improved perception accuracy, reliability, and robustness. Among the LiDAR-fusion methods, the fusion of LiDAR sensors and cameras including visual cameras and thermal cameras is most widely used in the area of robotics and autonomous driving perception. Hence, this paper also reviews deep learning-based fusion methods for LiDAR.

LiDAR-based 3D perception tasks take a LiDAR point cloud (or a LiDAR point cloud fused with images or

data from other sensors) as input, and then outputs the category of the target object (3D shape classification); 3D bounding box implying location, height, length, and width with the category of the target object (3D object detection); track ID in a continuous sequence (3D object tracking); segmented label for each point (3D segmentation); etc.[1]. In addition, 3D point cloud registration, 3D reconstruction, 3D point cloud generation, and 6-DOF pose estimation are also tasks worth researching.

Previous related surveys review deep learning methods on LiDAR point cloud before 2021 [5–8]. This paper reviews the latest deep learning methods on not only LiDAR point cloud but also LiDAR point cloud fusion (with image and radar). Compared with multi modality fusion surveys [9–11], which cover a wide range of sensors, this paper provides a more detailed and comprehensive review on each related 3D perception task (3D shape classification, 3D object detection, 3D object tracking, and 3D segmentation). The contribution of this paper is summarized as follows:

1. This paper is a survey that focuses on deep learning algorithms with only LiDAR point clouds and LiDAR-based fusion data (especially LiDAR point cloud fused with the camera image) as input in the field of autonomous driving. This work is structured considering four representative 3D perception tasks, namely 3D shape classification, 3D object detection, 3D object tracking, and 3D segmentation.
2. This paper gives a review of methods organized by whether fusion data are utilized as their input data. Moreover, studies and algorithms reviewed in this paper were published in the last decade, which ensures the timeliness and refer-ability of the study.
3. This paper puts some open challenges and possible research directions forward to serve as a reference and stimulate future works.

The remainder of this paper is structured as follows. Section 2 provides background knowledge about LiDAR point clouds, including representations and characteristics of LiDAR point cloud, existing LiDAR-based benchmark datasets, and corresponding evaluation metrics. The following four sections give a review of representative LiDAR-only and LiDAR-fusion methods for four 3D perception tasks: Section 3 for 3D shape classification, Section 4 for 3D object detection, Section 5 for 3D object tracking, and Section 6 for 3D semantic segmentation and instance segmentation. Some discussions about overlooked challenges and promising directions are raised in Section 7. At the end, Section 8 draws the conclusions for this paper.

2. BACKGROUND

Point clouds in the field of autonomous driving are generally generated by the on-board LiDAR. The existing mainstream LiDAR emits laser wavelengths of 905 and 1550 nm, which are focused and do not disperse over long distances. When a laser beam of LiDAR hits the surface of an object, the reflected laser carries information of the target object such as location and distance. By scanning the laser beam according to a certain trajectory, the information of the reflected laser points will be recorded. Since the LiDAR scanning is extremely fine, many laser points can be obtained, and thus a LiDAR point cloud is available. The LiDAR point cloud (point clouds mentioned in this paper refer to LiDAR point clouds) is an unordered sparse point set representing the spatial distribution of targets and characteristics of the target surface under the same spatial reference system. There are three approaches basically implemented in deep learning-based methods to process LiDAR point cloud so that processed data can be used as input data to the network: (1) multi-view-based methods; (2) volumetric-based methods; and (3) point-based methods. Multi-view-based methods represent point cloud as 2D views by projecting it onto 2D grid-based feature maps, which can leverage existing 2D convolution methods and

[1]Here, we use the term 3D to narrowly describe the tasks with 3D point clouds or 3D point cloud-based fusion data as input and information of the object in real 3D space as output (i.e., category, 3D bounding box, and semantic labels of objects). Broadly speaking, some other works explain 3D tasks as tasks inferring information of the object in real 3D space with any kind of input data.

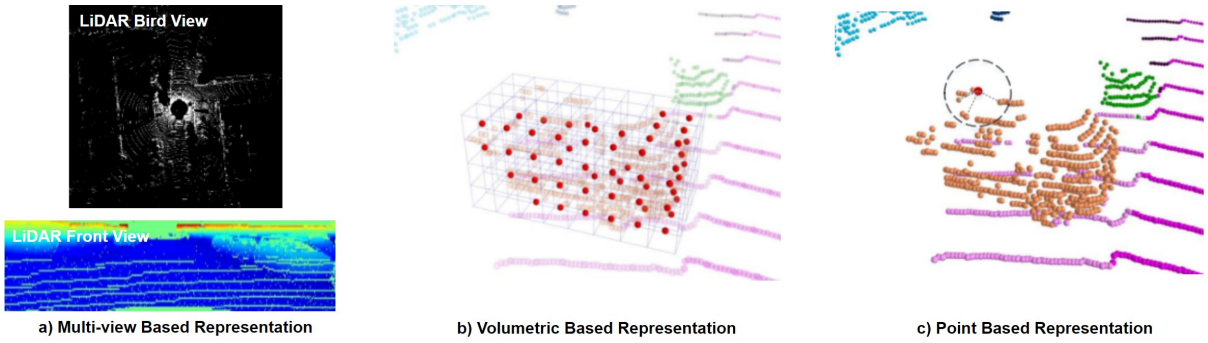

Figure 1. Three approaches for LiDAR point cloud representation: (a) multi-view-based methods; (b) volumetric-based methods; and (c) point-based methods. The image in (a) is original originally from MV3D [12]. The images in (b,c) are original originally from RPVNet [13]

view-pooling layers. Volumetric-based methods discretize the whole 3D space into plenty of 3D voxels, where each point in the original 3D space is assigned to the corresponding voxel following some specific regulations. This representation can preserve rich 3D shape information. Nevertheless, the limitation of performance is inevitable as a result of the spatial resolution and fine-grained 3D geometry loss during the voxelization. On the contrary, point-based methods conduct deep learning methods directly on the point cloud in continuous vector space without transforming the point cloud into other intermediate data representations. This approach avoids the loss caused by transformation and data quantification and preserves the detailed information of the point cloud. The visualization of the three representations is illustrated in Figure 1.

The point cloud carries point-level information (e.g., the x, y, and z coordinates in 3D space, color, and intensities) and keeps invariant under rigid transformation, scaling, and permutation. An azimuth-like physical quantity can be easily acquired from the point cloud, and thus diverse features can be generated for deep learning. Although the point cloud is less affected by the variation of illumination and scale when compared to the image, the point cloud suffers more from the intensity and often ignores sparse information reflected by the surface of objects. The laser emitted by LiDAR cannot bypass obstacles and will be greatly disturbed or even unable to work in the rain, fog, sand, and other severe weather. Thus, challenges exist when extracting features from the spatial-sparse and unordered point sets. Algorithms have evolved from hand-crafted features extraction to deep-learning ones. Among them, point-wise and region-wise methods treat different paths that lead to the same destination. Meanwhile, the cooperation with other sensors shows huge potential to improve the performance through supplementing insufficient information, which may unexpectedly lead to extra computational cost or information redundancy if not well designed. Therefore, studies focus on how to reach a compromise on the cost and the performance when conducting LiDAR-fusion tasks.

With the development of LiDAR, increasing LiDAR point cloud datasets are available, facilitating the training and evaluation among different algorithms. Table 1 [14–28] lists datasets recorded by LiDAR-based visual system. Among them, KITTI [14] provides a comprehensive real-world dataset for autonomous driving, providing a benchmark for 3D object detection, tracking, and scene flow estimation. The evaluation metrics vary for different tasks. For 3D classification, the overall accuracy (OA) and the mean class accuracy (mAcc) are widely used. For 3D object detection, the average precision (AP) and mean average precision (mAP) are mostly-used. For 3D object tracking, precision and success are commonly used as evaluation metrics of single object tracker. Average multi-object tracking Accuracy (AMOTA) and average multi-object tracking precision (AMOTP) are used as evaluation metrics for a 3D multi-object tracker. For 3D segmentation, mean intersection over union (mIoU), OA, and mAcc are widely used for the algorithm evaluation.

Table 1. Dataset recorded by LiDAR-based visual system

Types	Dataset	Year	Data Source	Application
LiDAR-only	Sydney Urban Objects [15]	2013	LiDAR point cloud	Classification
	ScanObjectNN [16]	2019	LiDAR point cloud	Classification
	DALES [17]	2020	LiDAR point cloud	Segmentation
	LASDU [18]	2020	LiDAR point cloud	Segmentation
	Campus3D [19]	2020	LiDAR point cloud	Segmentation
	Toronto-3D [20]	2020	LiDAR point cloud	Segmentation
LiDAR-fusion	KITTI [14]	2012	RGB image + LiDAR point cloud	Majority of tasks
	RueMonge2014 [21]	2014	RGB image + RGB-D image + LiDAR point cloud	Segmentation
	Matterport3D [22]	2017	RGB-D image+ LiDAR point cloud	Segmentation
	H3D [23]	2019	RGB image + LiDAR point cloud	Detection + tracking
	Argoverse [24]	2019	RGB image + LiDAR point cloud	Detection + tracking
	Lyft_L5 [25]	2019	RGB image + LiDAR point cloud	Detection + tracking
	Waymo Open [26]	2020	RGB image + LiDAR point cloud	Detection + tracking
	nuScenes [27]	2020	RGB image + LiDAR point cloud	Detection + tracking
	MVDNet [28]	2021	RaDAR + LiDAR point cloud	Detection

3. 3D SHAPE CLASSIFICATION

Object classification on point cloud is generally known as 3D shape classification or 3D object recognition/classification. There are both inheritance and innovation when transferring 2D object classification to 3D space. For multi-view-based methods, methods for 2D images can be adopted since the point cloud is projected into 2D image planes. However, finding an effective and optimal way to aggregate features of multiple views is still challenging. For point-based methods [29,30], designing novel networks according to the characteristics of the point cloud is the key task. 3D object recognition frameworks usually follow a similar pipeline: Point clouds are first aggregated with an aggregation encoder in order to extract a global embedding. Subsequently, the global embedding is passed through several fully connected layers, after which the object category can be predicted. According to different forms of input data, 3D classifiers can be divided into LiDAR-only classifiers and LiDAR-fusion classifiers. This section reviews existing methods for 3D shape classification. A summary of the algorithms is shown in Table 2, including modalities and representations of data, algorithm novelty, and performance on ModelNet40 [31] dataset for 3D object classification.

3.1. LiDAR-only classification

In terms of diverse representations of the point cloud as input data, LiDAR-only classifiers can be divided into volumetric representation, 2D views representation, and point representation. Different from volumetric representation- and 2D views representation-based models, which preprocess point cloud into voxel or 2D multi-views by projection, point representation-based methods apply a deep learning model on the point cloud directly. Qi *et al.* [1] proposed a path-breaking architecture called PointNet, which works on raw point cloud for the first time. A transformation matrix learned by T-Net can align the input data and a canonical space in order to ensure immutability after certain geometric transformations. Therefore, a global feature can be learned through several multi-layer perceptrons (MLP), T-Net, and max-pooling. Then, the feature is utilized to predict the final classification score by MLP. Shortly after, PointNet++ [2] extracts local features that PointNet [1] ignores at diverse scales and attains deep features through a multi-layer network. It also uses two types of density adaptive layers, multi-scale grouping (MSG) and multi-resolution grouping (MRG), to deal with the feature extraction of unevenly distributed point cloud data. These two works [1,2] can be implemented simply but achieves extraordinary performance at the same time; therefore, several networks are developed on their basis. MomNet [32] is designed on the basis of a simplified version of the PointNet [1] architecture, which consequently requires relatively low computational resources. Inspired by PointNet++ [2], Zhao *et al.* [33] proposed adaptive feature adjustment (AFA) to exploit contextual information in a local region. SRN [34] builds a structural relation network in order to consider local inner interactions. Recently, Yan *et al.* [35] introduced an end-to-end network named PointASNL with an adaptive sampling (AS) module and a local-nonlocal (L-NL) module, achieving excellent performance on the majority of datasets.

While the above methods learn point-wise features through multi-layer perceptrons, some other works adopt 3D convolutional kernels to design convolutional neural networks for point clouds, which can preserve more spatial information of point clouds. One of the typical networks is PointConv[36], which uses a permutation-invariant convolution operation. As an extension of traditional image convolution, the weight functions and the density functions of a given point in PointConv are learned from MLP and kernel density estimation, respectively. Boulch *et al.*[37] built a generalization of discrete convolutions for point clouds by replacing the discrete kernels for grid sampled data with continuous ones. Relation-shape convolutional neural network (RS-CNN)[38] is a hierarchical architecture which leverages the relation-shape convolution (RS-Conv) to learn the geometric topology constraint among points from their relations with an inductive local representation. Inspired by dense connection mode, Liu *et al.*[39] introduced DensePoint, a framework that aggregates outputs of all previous layers through a generalized convolutional operator in order to learn a densely contextual representation of point clouds from multi-level and multi-scale semantics. Apart from continuous convolutional kernels, discrete convolutional kernels play a role in deep learning for point clouds as well. ShellNet[29], a convolution network that utilizes an effective convolution operator called ShellConv, achieves a balance of high performance and short run time. ShellConv partitions the domain into concentric spherical shells and conducts convolutional operation based on this discrete definition. Mao *et al.*[40] proposed InterpConv for object classification, whose key parts are spatially-discrete kernel weights, a normalization term and an interpolation function. Rao *et al.*[41] introduced an architecture named spherical fractal convolutional neural network, in which point clouds are projected into a discrete fractal spherical structure in an adaptive way. Unlike other CNN methods, a novel convolution operator[30] is proposed, which convolves annularly on point clouds and is applied in an annular convolutional neural network (A-CNN), leading to higher performance. Through specified regular and dilated rings along with constraint-based K-NN search methods, the annular convolutional methods can order neighboring points and attain the relationship between ordered points. DRINet[42] develops a dual-representation (i.e., voxel-point and point-voxel) to propagate features between these two representations, performing SOTA on the ModelNet40 dataset with high runtime efficiency.

3.2. LiDAR-fusion classification

Sensors-fusion architectures have become an emerging topic due to their balance among the compatibility with application scenarios, the complementarity of perception information, and the cost. LiDAR is fused with other sensors to deal with specific tasks for autonomous driving. For instance, point clouds and images are fused in order to accomplish the 2D object detection[43,44] and the fusion of LiDAR and radar is applied to localize and track objects more precisely in terms of 3D object detection[4,45]. However, it is desirable to carry out the point cloud based object classification as a single task with fused methods in the field of real-world self-driving cars. Generally, 3D classification is implemented as a branch of 3D object detection architecture to classify targets of a proposal region and help predict the bounding box. Moreover, since the PointNet[1] was proposed in 2017, many studies dealing directly with raw point clouds have been inspired. For 3D classification task, the overall accuracy can achieve 93.6%[16] on the generic benchmark ModelNet40, which satisfies the demand for applications of autonomous car so that 3D classification is not regarded as an independent task. On the other hand, LiDAR-based fusion methods for the object category prediction are not feasible due to the lack of corresponding image datasets aligned with existing point cloud datasets. Only a few works concentrate on the fusion method specifically for 3D classification in the field of autonomous driving. Therefore, this section focuses on the classifier integrated into the LiDAR-fusion 3D detectors or segmentators.

According to the different stages in which sensors data are fused, fusion methods can be divided into early fusion and late fusion. For early fusion, features from different data sources are fused in the input stage by concatenating each individual feature into a unified representation. This representation is sent to a network to get final outputs. For late fusion, the prediction results from the individual uni-modal streams are fused to output the final prediction. Late fusion merges results by summation or averaging in the simplest cases. Compared with early fusion, late fusion lacks the ability to exploit cross correlations among multi-modal data.

Table 2. Experiment results of 3D object classification methods on ModelNet40 benchmark. Here "I", "mvPC", "vPC", "pPC", "rm" stands for image, multiple view of point cloud, voxelized point cloud, point cloud, range map respectively. "OA" represents the overall accuracy that is the mean accuracy for all test instance; "mAcc" represents the mean accuracy that is the mean accuracy for all shape categories. Here the '%' after the number is omitted for simplicity. "-" means the result is not available

Category	Model	Modal. &Repr.	Novelty	OA	mAcc
LiDAR-Only	PointNet [1]	pPC	point-wise MLP+T-Net+global max pooling	89.2	86.2
	PointNet++ [2]	pPC	set abstraction (sampling, grouping, feature learning)+fully connected layers	90.7	90.7
	Momen(e)t [32]	pPC	MLP+max pooling+pPC coordinates and their polynomial functions as input	89.3	86.1
	SRN [34]	pPC	structural relation network(geometric and locational features+MLP)	91.5	-
	PointASNL [35]	pPC	adaptive sampling module+local-nonlocal module	92.9	-
	PointConv [36]	pPC	MLP to approximate a weight function+a density scale	92.5	-
	RS-CNN [38]	pPC	relation-shape convolution(shared MLP+channel-raising mapping)	92.6	-
	DensePoint [39]	pPC	PConv+PPooling(dense connection like)	93.2	-
	ShellNet [29]	pPC	shellconv(KNN+max pooling+shared MLP+conv order)	93.1	-
	InterpConv [40]	pPC	interpolated convolution operation+max pooling	93.0	-
	DRINet [42]	vPC+pPC	sparse point-voxel feature extraction+sparse voxel-point feature extraction	93.0	-
LiDAR-Fusion	MV3D [12]	I&mvPC	3D proposals network+region-based fusion network	-	-
	SCANet [46]	I&mvPC	multi-level fusion+spatial-channel attention+extension spatial upsample module	-	-
	MMF [47]	I&mvPC	point-wise fusion+ROI feature fusion	-	-
	ImVoteNet [48]	I&pPC	lift 2D image votes, semantic and texture cues to the 3D seed points	-	-

Classifiers integrated into two-stage LiDAR-fusion 3D detectors can be divided into two categories: (1) classifiers to distinguish the target and background; and (2) classifiers to predict the final category of the target object. Chen *et al.* [12] designed a deep fusion framework named multi-view 3D networks (MV3D) combining LiDAR point clouds and RGB images. This network designs a deep fusion scheme that alternately performs feature transformation and feature fusion, which belongs to the early fusion architecture. MV3D comprises a 3D proposal network and a region-based fusion network, both of which have a classifier. The classifier in the 3D proposal network regresses to distinguish whether it belongs to the foreground or background, and then the results along with 3D box generated by the 3D box regressor are fed to 3D Proposal Module to generate 3D proposals. The final results are obtained by a multiclass classifier that predicts the category of objects through a deep fusion approach using the element-wise mean for the join operation and fusing regions generated from multi-modal data. Motivated by deep fusion [12], ScanNet [46] proposes multi-level fusion layers fusing 3D region proposals generated by an object classifier and a 3D box regressor to enable interactions among features. ScanNet also introduces the attention mechanism in spatial and channel-wise dimensions in order to capture global and multi-scale context information. The multi-sensor fusion architecture [47] can accomplish several tasks by one framework, including object classification, 3D box estimation, 2D and 3D box refinement, depth completion, and ground estimation. In the 3D classification part, LiDAR point clouds are first projected into ground relative bird's eye view (BEV) representation through the online mapping module, and then features extracted from LiDAR point clouds, and RGB images are fused by the dense fusion module and fed into LiDAR backbone network to predict the probability of the category. This multi-task multi-sensor architecture performs robustly and qualitatively on the TOR$_4$D benchmark. For one-stage 3D fused detectors, the classifier is generally applied in a different way because the one-stage detectors aim to conduct classification and regression simultaneously. Qi *et al.* [48] proposed a one-stage architecture named ImVoteNet, which lifts 2D vote to 3D to improve 3D classification and detection performance. The architecture consists of two parts: One leverages 2D images to pass the geometric, semantic, and texture cues to 3D voting. The other proposes and classifies targets on the basis of a voting mechanism such as Hough voting. The results show that this method boosts 3D recognition with improved mAP compared with the previous best model [49].

4. 3D OBJECT DETECTION

All the deep learning detectors follow a similar idea: they extract the feature from the input data with the backbone and neck of the framework to generate proposals and then classify and locate the objects with a 3D bounding box with the head part. Depending on whether region proposals are generated or not, the object

detectors can be categorized into two-stage and single-stage detectors. Two-stage detectors detect the target from the region of interests proposed from the feature map, while single-stage detectors perform tasks based on sliding dense anchor boxes or anchor points from the pyramid map directly. This section summarizes contemporary 3D object detection research, focusing on diverse data modalities from different sensors. Table 3 shows the summary for 3D object detection. Table 4 summarizes experiment results of 3D object detection methods on the KITTI test 3D object detection benchmark.

4.1. LiDAR-only detection

LiDAR-only detection generates a 3D bounding box based on networks that are only fed with a LiDAR point cloud. In general, two-stage detection processes LiDAR data with point-based representation, while single-stage detection performs the task on multiple formats, including point cloud-based, multi-viewed, and volumetric-based representations.

4.1.1. Two-stage detection

For the two-stage detection, segmentation is a widely-used method to remove noisy points and generate proposals in the first sub-module of the detection. One of the typical detection models is IPOD[50], which seeds instance-level proposals with context and local features extracted by projected segmentation. In 2019, STD[51] created point-level spherical anchors and parallel intersection-over-union (IOU) branches to improve the accuracy of the location. Following the proposal scheme of PointRCNN[52] (whose network is illustrated in Figure 2a), PointRGCN[53] introduces a graph convolutional network which aggregates per-proposal/per-frame features to improve the detection performance. Shi *et al.*[54] extended the method of PointRCNN[52] in another way, by obtaining 3D proposals and intra-object part locations with a part-aware module and regressing the 3D bounding boxes based on the fusion of appearance and location features in the part-aggregation framework. HVNet[55] fuses multi-scale voxel features point-wisely, namely hybrid voxel feature encoding. After voxelizing the point cloud at multiple scales, HVNet extracts hybrid voxel features with an attentive voxel feature encoder, and then pseudo-image features are available through scale aggregation in point-wise format. To remedy the proposal size ambiguity problem, LiDAR R-CNN[56] uses boundary offset and virtual point, designing a plug-and-play universal 3D object detector.

4.1.2. Single-stage detection

Unlike the two-stage detector that outputs final fine-grained detection results on the proposals, the single-stage detector classifies and locates 3D objects with a fully convolutional framework and transformed representation. Obviously, this method makes the foreground more susceptible to adjacent background points, thus decreasing the detection accuracy. Multiple methods emerge to solve this problem. For example, VoxelNet[57] extracts voxel-wise features from point clouds in volumetric-based representation with random sampling and normalization, after which it utilizes a 3D-CNN-based framework and region proposal network to detect 3D objects. To bridge the gap between the 3D-CNN-based and 2D-CNN-based detection, the authors of[58] applied PointNet[1] to point clouds to generate vertical-columned representation, which enables point clouds to be processed by the following 2D-CNN-based detection framework. Multi-task learning work[59] introduces a part-sensitive warping module and an auxiliary module to refine the feature extracted from the backbone network by adapting the ROI pooling from R-FCN[60] detection module. As illustrated in Figure 2c, TANet[61] designs a stacked triple attention module and a coarse-to-fine regression module to reduce the disturbance of noisy points and improve the detection performance on hard-level objects. SE-SSD[62] contains a teacher SSD and a student SSD. The teacher SSD produces soft targets by predicting relatively accurate results (after global transformation) from the input point cloud. The student SSD takes augmented input (a novel shape-aware data argumentation) as input, and then is trained with a consistency loss under the supervision of hard-level targets. 3D auto-labeling[63], which aims at saving the cost of human labeling, proposes a novel off-board 3D object detector to exploit complementary contextual information from point cloud sequences, achieving a performance on par with human labels.

Table 3. Summary of 3D object detection methods. Here "I", "mvPC", "vPC", "pPC", "RaPC" stands for image, multiple view of point cloud, voxelized point cloud, point cloud, Radar point cloud respectively

Detector	Category	Model	Modality & Representation	Novelty
Two-stage Detection	LiDAR -Only	IPOD [50]	pPC	a novel point-based proposal generation
		STD [51]	pPC	proposal generation(from point-based spherical anchors)+PointPool
		PointRGCN [53]	pPC	RPN+R-GCN+C-GCN
		SRN [34]	pPC	structural relation network(geometric and locational features+MLP)
		Part-A2 [54]	pPC	intra-object part prediction+RoI-aware point cloud pooling
		HVNet [55]	vPC	multi-scale voxelization+hybrid voxel feature extraction
		LiDAR R-CNN [56]	pPC	R-CNN style second-stage detector(size aware point features)
	LiDAR -Fusion	3D-CVF [64]	I & vPC	CVF(auto-calibrated projection)+adaptive gated fusion network
		Roarnet [65]	I & pPC	RoarNet 2D(geometric agreement search)+RoarNet 3D(RPN+BRN)
		MV3D [12]	I & mvPC	3D proposals network+region-based fusion network
		ScanNet [46]	I & mvPC	multi-level fusion+spatial-channel attention +extension spatial upsample
		MMF [47]	I & mvPC	point-wise fusion+ROI feature fusion
		Pointpainting [66]	I & pPC	image based semantics network+appended (painted) point cloud
		CM3D [67]	I & pPC	pointwise feature fusion+proposal genaration+ROI-wise feature fusion
		MVDNet [28]	RaPC & mvPC	two-stage deep fusion(region-wise feature fusion)
One-stage Detection	LiDAR -Only	VoxelNet [57]	vPC	voxel feature encoding+3D convolutional middle layer+RPN
		PointPillars [58]	pillar points	pillar feature net+backbone(2D CNN)+SSD detection head
		SASSD [59]	pPC	backbone(SECOND)+auxiliary network+PS Warp
		TANet [61]	vPC	Triple Attention module(channel-wise, point-wise, and voxel-wise attention)
		SE-SSD [62]	pPC	teacher and student SSDs+shape aware augumentation+consistency loss
		3D Auto Label [63]	mvPC	motion state classification+static object and dynamic object auto labeling
		ImVoteNet [48]	I & pPC	lift 2D image votes, semantic and texture cues to the 3D seed points
		EPNet [68]	I & pPC	two-stream RPN+LI-Fusion Module+refinement network
	LiDAR-Fusion	CLOCs [69]	I & vPC	a late fusion architecture with any pair of pre-trained 2D and 3D detectors

Table 4. Experiment results of 3D object detection methods on KITTI test 3D object detection benchmark. Average Precision (AP) for car with IoU threshold 0.7, pedestrian with IoU threshold 0.5, and cyclist with IoU threshold 0.5 is shown. "-" means the result is not available

Model	Car			Pedestrian			Cyclist		
	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
IPOD [50]	79.75%	72.57%	66.33%	56.92%	44.68%	42.39%	71.40%	53.46%	48.34%
STD [51]	79.71%	87.95%	75.09%	42.47%	53.29%	38.35%	61.59%	78.69%	55.30%
PointRGCN [53]	85.97%	75.73%	70.60%	-	-	-	-	-	-
Part-A2 [54]	85.94%	77.86%	72.00%	89.52%	84.76%	81.47%	54.49%	44.50%	42.36%
LiDAR R-CNN [56]	85.97%	74.21%	69.18%	-	-	-	-	-	-
3D-CVF [64]	89.20%	80.05%	73.11%	-	-	-	-	-	-
Roarnet [65]	83.71%	73.04%	59.16%	-	-	-	-	-	-
MV3D [12]	71.09%	62.35%	55.12%	-	-	-	-	-	-
SCANet [46]	76.09%	66.30%	58.68%	-	-	-	-	-	-
MMF [47]	86.81%	76.75%	68.41%	-	-	-	-	-	-
CM3D [67]	87.22%	77.28%	72.04%	-	-	-	-	-	-
VoxelNet [57]	77.47%	65.11%	57.73%	39.48%	33.69%	31.51%	61.22%	48.36%	44.37%
PointPillars [58]	79.05%	74.99%	68.30%	52.08%	43.53%	41.49%	75.78%	59.07%	52.92%
SASSD [59]	88.75%	79.79%	74.16%	-	-	-	-	-	-
TANet [61]	84.81%	75.38%	67.66%	54.92%	46.67%	42.42%	73.84%	59.86%	53.46%
SE-SSD [62]	91.49%	82.54%	77.15%	-	-	-	-	-	-
EPNet [68]	89.81%	79.28%	74.59%	-	-	-	-	-	-
CLOCs [69]	88.94%	80.67%	77.15%	-	-	-	-	-	-

4.2. LiDAR-fusion detection

LiDAR-fusion detection enriches the information with the aspect of data sources to improve the performance at a low cost. Its auxiliary input data include RGB images, angular velocity (acceleration), depth images, and so on.

4.2.1. Two-stage detection

The input data of the LiDAR-fusion detector vary in diverse fields with aspects of sampling frequency and data representations. Hence, simple summation or multiplication at the source side contributes little to the

improvement of the algorithm performance. In general, two-stage detection fuses the feature map before or after the proposals. To enhance the quality of proposals, 3D-CVF[64] fuses spatial features from images and point clouds in cross-wise views with the auto-calibrated feature projection. Based on PointNet[1], Roarnet[65] designs a two-stage object detection network whose input data contain RGB image and LiDAR point cloud to improve the performance with 3D pose estimation. As for the fusion of ROI-wise feature, Chen *et al.*[12] fused the feature extracted from the bird's eye view and front view of LiDAR as well as the RGB image. As shown in Figure 2b, Scanet[46] applies a spatial-channel attention module and an extension spatial up-sample module to generate proposals of RGB images and point clouds, respectively, in the first stage and then classifies and regresses the 3D bounding box with a novel multi-level fusion method. Meanwhile, some studies adopt multi-fusion methods in the proposed schemes. For instance, the authors of[47] completed a two-stage detection framework with front-end fusion and medium fusion. Its front-end fusion is to merge the sparse depth image (projected from LiDAR point cloud) and RGB image for the image backbone network to extract dense depth feature. The depth feature would be fed into the dense fusion module with LiDAR point clouds and pseudo-LiDAR points to prepare for medium fusion. Vora *et al.*[66] complemented the context information of point cloud with the semantic segmentation results of the image. Through the point painting operation, point clouds are painted by semantic scores, and then the painted point cloud is fed into a point-based 3D detector to produce final results. The pipeline[67] fuses point-wise features and couples 2D–3D anchors (which are generated from images and point clouds, respectively) to improve the quality of proposals in the first stage, after which it handles ROI-wise feature fusion in the second stage. To deal with adverse weather, MVDNet[28] exploits LiDAR and radar's potential complementary advantages. This novel framework conducts a deep late fusion, which means that proposals are generated from two sensors first and then region-wise features are fused. Moreover, MVDNet provides a foggy weather focused LiDAR and radar dataset generated from the Oxford Radar Robotcar dataset. EPNet[68] is a closed-loop two-stage detection network. Its LI-fusion module projects point cloud to images and then generates point-wise correspondence for the fusion. To form the closed-loop, EPNet achieves 3D end-to-end detection on the high definition map and estimates the map on the fly from raw point clouds. ImVoteNet[48] (which is an extension of VoteNet[49]) supplements the point-wise 3D information with the geometrical and semantic features extracted from 2D-images. In its head module, LiDAR-only, image-only, and LiDAR-fusion features all participate in the voting to improve the detection accuracy.

4.2.2. Single-stage detection

Single-stage detectors outperform two-stage detectors in terms of runtime due to their compact network structure. With the goal of high efficiency and accuracy, the fusion of single stage detector is placed in the post-processing stage (i.e., late fusion) in order to maintain the superior single-shot detection performance and improve through supplementary multi-sensor data at the same time. This indicates that only the results of detectors for LiDAR point cloud and other sensor data (e.g., RGB image) are fused in post-processing module without changing any network structure of detectors. CLOCs[69] builds a late fusion architecture with any pair of pre-trained image and LiDAR detectors. The output candidates of LiDAR and image are combined before the non-maximum suppression operation to exploit geometric and semantic consistencies. Individual 2D and 3D candidates are first pre-processed through specific tensor operation so that they are both in a consistent joint representation using sparse tensor. Then, a set of 2D convolution layers are utilized to fuse, which takes the sparse tensor as input and output a processed tensor. The max-pooling operation is conducted on this tensor to map it to the targets (formatted as a score map). Experiment results on the KITTI dataset show that single-stage 3D detector SECOND[70] fusion with 2D detector Cascade R-CNN[71] achieves better performance by a large margin compared to single-modality SECOND. The architecture of CLOCs is shown in Figure 2d.

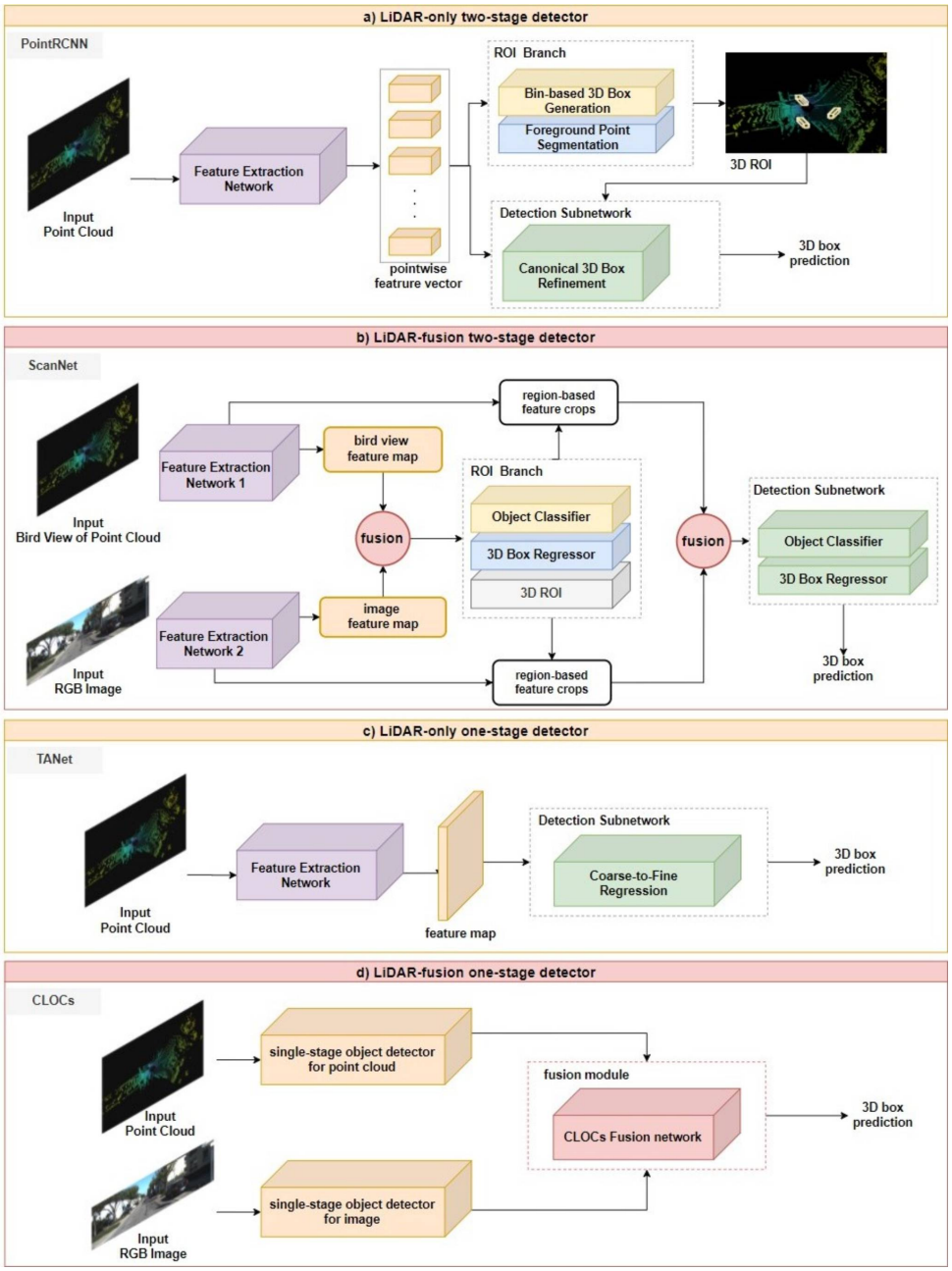

Figure 2. Typical architectures for two categories of LiDAR-based two-stage 3D detection: (a) LiDAR-only and (b) LiDAR-fusion methods. Typical networks for two categories of LiDAR-based one-stage detector: (c) LiDAR-only and (d) LiDAR-fusion methods.

5. 3D OBJECT TRACKING

All the trackers obey the same rule: they estimate the states of targets contained in the subsequent frames under the guidance of the targets in the first frame. Trackers need to overcome more difficulties, including illumination and scale variation, because trackers perform tasks with richer geometric information and context information compared to image-based trackers and LiDAR-based detectors. Unlike the isolation of single-object tracking and multi-object tracking in the field of the image, in the field of 3D tracking, both trackers are related and the former one can be regarded as a simplified version of the latter one. This section reviews two methods of achieving online 3D tracking: detection and siamese network. Table5 summarizes these works.

5.1. LiDAR-only tracking

As the temporal extension of detection, tracking achieves higher and more precise performance based on appearance similarity and motion trajectory. Tracking-by-detection is an intuitive method. For example, Vaquero *et al.*[72] fused vehicle information segmented from dual-view detectors (i.e., a front view and a bird's eye view) and then utilized extended Kalman filter, Mahalanobis distance, and motion update module to perform 3D tracking. Furthermore, Shi *et al.*[73] performed 3D tracking and domain adaption based on a variant of the 3D detection framework (i.e., PV-RCNN), which comprises temporal information incorporation and classification with RoI-wise features, and so on. In addition, detection results can be enhanced by extra target templates. As a typical example, P2B[74] first matches the proposals with augmented target-specific features and then regresses target-wise centers to generate high-quality detection results for tracking. Following CenterTrack[75], CenterPoint[76] develops an object-center-tracking network through velocity estimation and the point-based detection that views objects as points, achieving more accurate and faster performance.

As for the image-based tracking, the siamese network eliminates the data redundancy and speeds up the task through the conversion from tracking to patch matching, whose idea can be extended in the field of LiDAR-based tracking. Inspired by SAMF[77], Mueller *et al.*[78] designed a correlation filter-based tracker (i.e., SAMF_CA) which incorporates global context in an explicit way. Experiments show that the improved optimization solution achieves a better performance in the single target tracking domain. The work of Zarzar *et al.*[79] shows that the siamese network-based tracking with LiDAR-only data performs well in aerial navigation. Holding the belief that appearance information is insufficient to track, Giancola *et al.*[80] encoded the model shape and candidate shape into latent information with a Siamese tracker. Zarzar *et al.*[81] generated efficient proposals with a siamese network from the BEV representation of point clouds, after which it tracks 3D objects in accordance with the ROI-wise appearance information regularized by the latter siamese framework. PSN[82] first extracts features through a shared PointNet-like framework and then conducts feature augmentation and the attention mechanism through two separate branches to generate a similarity map so as to match the patches. Recently, MLVSNet[83] proposes conducting Hough voting on multi-level features of target and search area instead of only on final features to overcome insufficient target detection in sparse point clouds. Moreover, ground truth bounding box in the first frame can be regarded as a strong cue, enabling a better feature comparison[84], as shown in Figure 3a.

5.2. LiDAR-fusion tracking

Sensors capture data from various views, which is beneficial to supplement insufficient information for trackers. A challenge of tracking-by-detection is how to match the detection results with the context information. The simplest way is to conduct an end-fusion of the tracking results, as done by Manghat *et al.*[85]. In addition, Frossard *et al.*[86] produced precise 3D trajectories for diverse objects in accordance with detection proposals and linear optimization. Introducing the 2D visual information, Complexer-YOLO[87] first performs joint 3D object detection based on the voxelized semantic points clouds (which are fused by image-based semantic information) and then extends the model to multi-target tracking through multi-Bernoulli filter. This work demonstrates the role of scale–rotation–translation, which enables the framework to track in real time.

However, data sampled by different sensors vary in frequency and dimension, and thus it is challenging and not cost-effective to match the similarity among diverse data sources. Recent years have witnessed the emergence of ingenious algorithms while tracking based on a siamese network is still in its infancy. Developed for single object tracking, F-Siamese Tracker[88] extrudes a 2D region-of-interest from a siamese network for the purpose of generating several valid 3D proposals, which would be fed into another siamese network together with a LiDAR template. Although these studies achieve a lot, there is still a long way to go to further integrate point clouds and other sensor data (i.e., RGB images) into the siamese network for LiDAR-fusion tracking. The pipeline of F-Siamese Tracker is explained in Figure 3b.

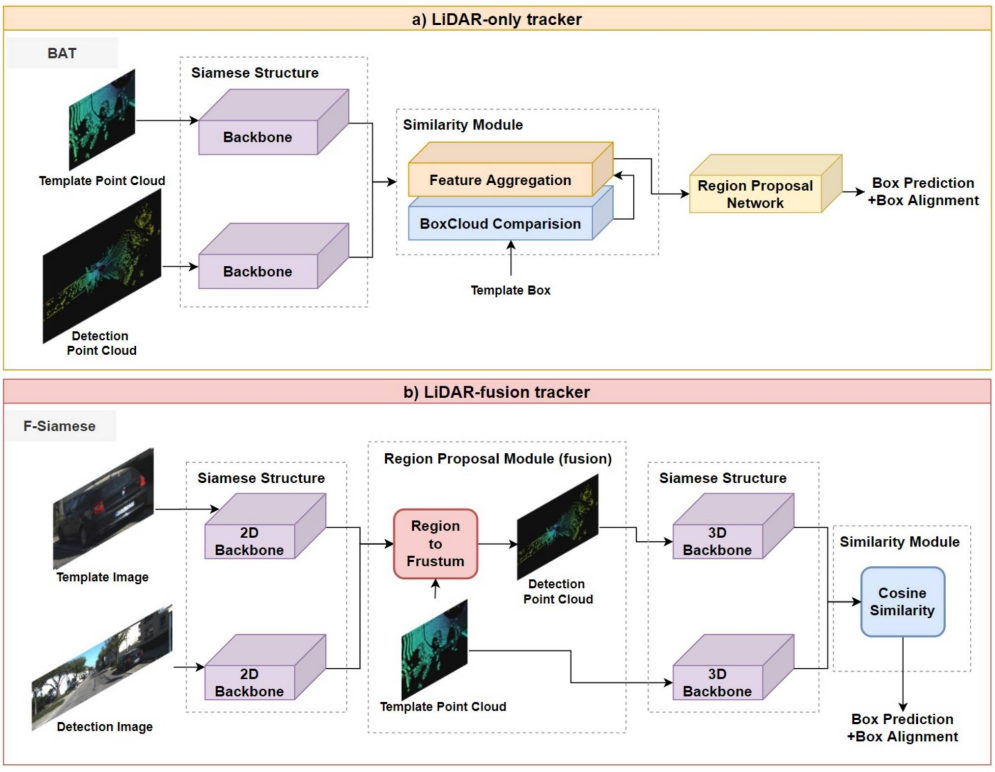

Figure 3. Typical networks for two categories of LiDAR-based tracker: (a) LiDAR-only and (b) LiDAR-fusion methods.

Table 5. Summary of 3D object tracking. Here "I", "mvPC", "vPC", "pPC", "FrustumPC" stands for image, multiple view of point cloud, voxelized point cloud, point cloud, Frustum point cloud respectively

Category	Model	Modality & Representation	Architecture
LiDAR -Only	DualBranch [72]	mvPC	Bbox growing method + multi-hypothesis extended Kalman filter
	PV-RCNN [73]	pPC & vPC	Voxel-to-keypoint 3D scene encoding + keypoint-to-grid RoI feature abstraction
	P2B [74]	pPC	Target-specific feature augmentation + 3D target proposal and verification
	CenterPoint [76]	pillar/vPC	Map-view feature representation + center-based anchor-free head
	SC-ST [80]	pPC	Siamese tracker(resemble the latent space of a shape completion network)
	BEV-ST [81]	mvPC	Efficient RPN+Siamese tracker
	PSN [82]	pPC	Siamese tracker(feature extraction + attention module + feature augumentation)
	MLVSNet [83]	pPC	Multi-level voting+Target-Guided Attention+Vote-cluster Feature Enhancement
	BAT [84]	pPC	Box-aware feature fusion + box-aware tracker
LiDAR -Fusion	MSRT [85]	I&pPC	2D object detector-Faster-RCNN+3D detector-Point RCNN
	MS3DT [86]	I&mvPC	Detection proposals+proposals matching&scoring+linear optimization
	Complexer-YOLO [87]	I&vPC	Frame-wise 3D object detetcion+novel Scale-Rotation-Transalation score
	F-Siamese Tracker [88]	I&FrustumPC	Double Siamese network

6. 3D SEGMENTATION

3D Segmentation methods can be classified into semantic segmentation and instance segmentation, which are both crucial for scene understanding of autonomous driving. 3D Semantic segmentation focuses on per-point semantic label prediction so as to partition a scene into several parts with certain meanings (i.e., per-point class labels), while 3D instance segmentation aims at finding the edge of instances of interest (i.e., per-object masks and class labels). Since Kirillov *et al.* [89] first came up with the concept "panoptic segmentation" that combines semantic segmentation and instance segmentation, several works [90,91] inspired by this concept have been published recently, which build architectures for panoptic segmentation of point cloud. This section specifically focuses on research concerning both 3D semantic segmentation and 3D instance segmentation

tasks whose input data are divided into LiDAR point cloud data or LiDAR point cloud fused data. Summaries can be seen in Tables 6 and 7.

6.1. 3D Semantic segmentation

6.1.1. LiDAR-only semantic segmentation

PointNet[1] provides a classic prototype of point cloud semantic segmentation architecture utilizing shared MLPs and symmetrical poolings. On this basis, several dedicated point-wise MLP networks are proposed to attain more information and local structures for each point. PointNet++[2] introduces a novel hierarchical architecture applying PointNet recursively to capture multi-scale local context. Engelmann *et al.*[92] proposed a feature network with K-means and KNN to learn a better feature representation. Besides, an attention mechanism, namely group shuffle attention (GSA)[93] is introduced to exploit the relationships among subsets of point cloud and select a representative one.

Apart from MLP methods, convolutional methods on pure points also achieve some state-of-the-art performance, especially after a fully convolutional network (FCN)[94] is introduced to semantic segmentation, which replaces the fully connected layer with a convolution and thus makes any size of input data possible. Based on the idea of GoogLeNet[95] that takes fisheye cameras and LiDAR sensors data as input, Piewak *et al.*[96] proposed an FCN framework called LiLaNet aiming to label emi-dense LiDAR data point-wisely and multi-class semantically with cylindrical projections of point clouds as input data. The dedicated framework LiLaNet is comprised of a sequence of LiLaBlocks that have various kernels and a 1×1 convolution so that lessons learned from 2D semantic label methods can be converted to the point cloud domain. Recently, a fully convolutional network called 3D-MiniNet[97] extends MiniNet[98] to 3D LiDAR point cloud domain to realize 3D semantic segmentation by learning 2D representations from raw points and passing them to 2D fully convolutional neural network to attain 2D semantic labels. The 3D semantic labels are obtained through re-projection and enhancement of 2D labels.

Based on the pioneering FCN framework, an encoder–decoder framework, U-Net[99] is proposed to conduct multi-scale and large size segmentation. Therefore, several point cloud-based semantic segmentation works extend this framework to 3D space. LU-Net[100] proposes an end-to-end model, consisting of a model that extracts high-level features for each point and an image segmentation network similar to U-Net that takes the projections of these high-level features as input. SceneEncoder[101] presents an encode module to enhance the performance of global information. As shown in Figure 4a, RPVNet[13] exploits fusion advantages of point, voxel, and range map representations of point clouds. After extracting features from the encoder–decoder of three branches and projecting these features into point-based representation, a gated fusion module (GFM) is adopted to fuse features.

Due to the close relationship between the receptive field size and the network performance, a few works concentrate on expanding the receptive fields through dilated/A-trous convolution, which can preserve the spatial resolution at the meanwhile. As an extension of SqueezeSeg[102], the CNN architecture named PointSeg[103] also utilizes SqueezeNet[104] as a backbone network with spherical images generated from point clouds as input. However, PointSeg[103] takes several image-based semantic segmentation networks into consideration and transfers them to the LiDAR domain, instead of using CRF post-processing as in SqueezeSeg[104]. The PointSeg[103] architecture includes three kinds of main layers: fire layer adapted from SqueezeNet[104], squeeze reweighting layer, and enlargement layer where dilated convolutional layers are applied to extend the receptive field. Hua *et al.*[105] introduced a point-wise convolution for 3D point cloud semantic segmentation, which orders point cloud before feature learning and adopts A-trous convolution. Recently, Engelmann *et al.*[106] proposed dilated point convolutions (DPC) to systematically expand the receptive field with an awesome generalization so that it can be applied in most existing CNN for point clouds.

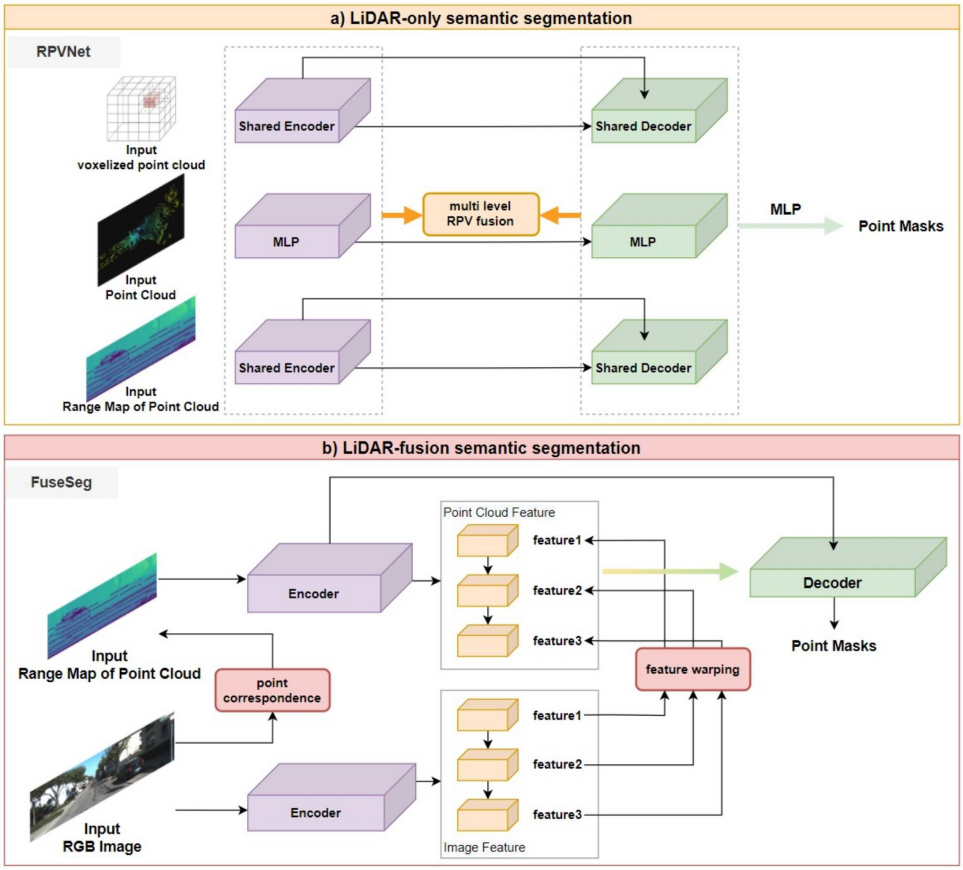

Figure 4. Typical frameworks for two categories of LiDAR-based semantic segmentation: (a) LiDAR-only and (b) LiDAR-fusion methods.

6.1.2. LiDAR-fusion semantic segmentation

One of the challenges existing in point cloud-based semantic segmentation is that the sparseness of the point cloud makes the object seem see-through, thus increasing the difficulty of discernment. Due to the different viewpoints of the RGB camera and LiDAR, RGB images can provide supplementary information about occluding objects. The fusion of RGB images and point clouds for 3D semantic segmentation is intensively researched in recent years due to the achievement of deep learning on 2D image segmentation. 3DMV [107] designs a feature-level fused joint 3D-multi-view prediction network, which combines geometric features of point clouds and color features of RGB images. This work leverages a 2D network to downsample the features extracted from full-resolution RGB input data and then leverages back-projection from a 2D feature into 3D space, rather than just mapping the RGB image on the voxel grid of point cloud. The final results are attained by the 3D convolution layers that take these back-projected 2D features and 3D geometric features as their input. As a result, 3DMV improves 3D semantic segmentation accuracy by 17.2 % in terms of the best volumetric framework at that time. Varga *et al.* [95] proposed an association of fisheye cameras and LiDAR sensors to segment feature-level 3D LiDAR point clouds. In this work, motion correction of point clouds and the undistortion and unwarping process of images are implemented first to ensure the reliability of the information. Subsequently, the undistorted fisheye image is segmented by computing the multiresolution filtered channels and deep CNN channels. Then, to transfer the pixel-wise semantic information to 3D points, the coordinates of 3D points are learned from projections of LiDAR points onto the camera image. With these coordinates, point clouds are augmented with color information and 2D semantic segmentation. Thanks to the well-settled sensor configuration, this super-sensor enables 360-degree environment perception for autonomous cars. MVPNet [108] presents a novel aggregation for feature fusion of point clouds and RGB

images. In this work, a proposed multi-view point cloud (MVPC) representation indicates a transformation from 2D image to the 3D point that expresses a discrete approximation of a ground-truth 3D surface by generating a sequence of 1-VPCs and forming predicted MVPC with their union, instead of simply combining projections. FuseSeg[3] proposes a LiDAR point clouds segmentation method that fuses RGB and LiDAR data at feature level and develops a network, whose encoder can be applied as a feature extractor for various 3D perception tasks. Figure 4b demonstrates details of its network. As an extension of SqueezeSeg[102], FuseSeg establishes correspondences between the two input modalities first and warps features extracted from RGB images. Then, the features from images and point clouds are fused by utilizing the correspondences. PMF[109] exploits supplementary advantages between appearance information from RGB images and 3D depth information from LiDAR point clouds. The two-stream network including camera-stream and LiDAR-stream extracts features from projected point cloud and RGB image, and then features from two modalities are fused by a novel residual-based fusion module into LiDAR stream. Additionally, a perception-aware loss contributes to the fusion network's ability. Unlike the ideas above, a novel permutohedral lattice representation method for data fusion is introduced[110]. SParse LATtice Networks (SPLATNet)[110] directly processes a set of points in the representation of a sparse set of samples in a high-dimensional lattice. To reduce the memory and computational cost, SPLATNet adopts a sparse bilateral convolutional layer as the backbone instead. This network incorporates point-based and image-based representations to deal with multi-modal data fusion and processing.

6.2. 3D Instance segmentation

Instance segmentation is the most challenging task of scene understanding because of the necessity to combine object detection and semantic segmentation, which focuses on each individual instance within a class.

6.2.1. LiDAR-only instance segmentation

One of the ideas is a top-down concept (also called the proposal-based method) which detects the bounding box of an instance with object detection methods first and then performs semantic segmentation within the bounding box. GSPN[111] designs a novel architecture for 3D instance segmentation named region-based PointNet (R-PointNet). A generative shape proposal network is integrated into R-PointNet to generate 3D object proposals with instance sensitive features by constructing shapes from the scene, which is converted into a 3D bounding box. The point ROIAlign module aligns features for proposals to refine the proposals and generates segmentation. Different from GSPN[111], the single-stage, anchor-free, and end-to-end 3D-BoNet[112] directly regresses 3D bounding boxes for all instances with a bounding box prediction branch. The backbone network exploits local point features and global features, which are then fed into a point mask prediction branch with a predicted object bounding box, as shown in Figure 5a.

However, the top-down idea ignores the relation between masks and features and extracts masks for each foreground feature, which is redundant. Down-top methods, also named proposal-free methods, may provide a solution for these problems, which performs point-wise semantic segmentation first and then distinguishes different instances. For example, Zhou *et al.*[113] presented an instance segmentation and object detection combined architecture to exploit detailed and global information of objects. It is a two-stage network, containing a spatial embedding (SE)-based clustering and bounding box refinement modules. For instance, segmentation, semantic information is attained by an encoder–decoder network, and object information is attained by SE strategy that takes center points of objects as important information. Aside from the above ideas, utilizing conditional random fields (CRFs) as post-processing methods contributes to the refinement of the label map generated by CNN and further improves the segmentation performance. Inspired by SqueezeNet[104], SqueezeSeg[102] proposes a pioneering lightweight end-to-end pipeline CNN to solve 3D semantic segmentation for road-objects. This network takes transformed LiDAR point cloud as input and then leverages network based on SqueezeNet[104] to extract features and label points semantically, whose results are fed into CRF to refine and output final results. As an extension of SqueezeSeg[102], SqueezeSegV2[114] introduces three novel

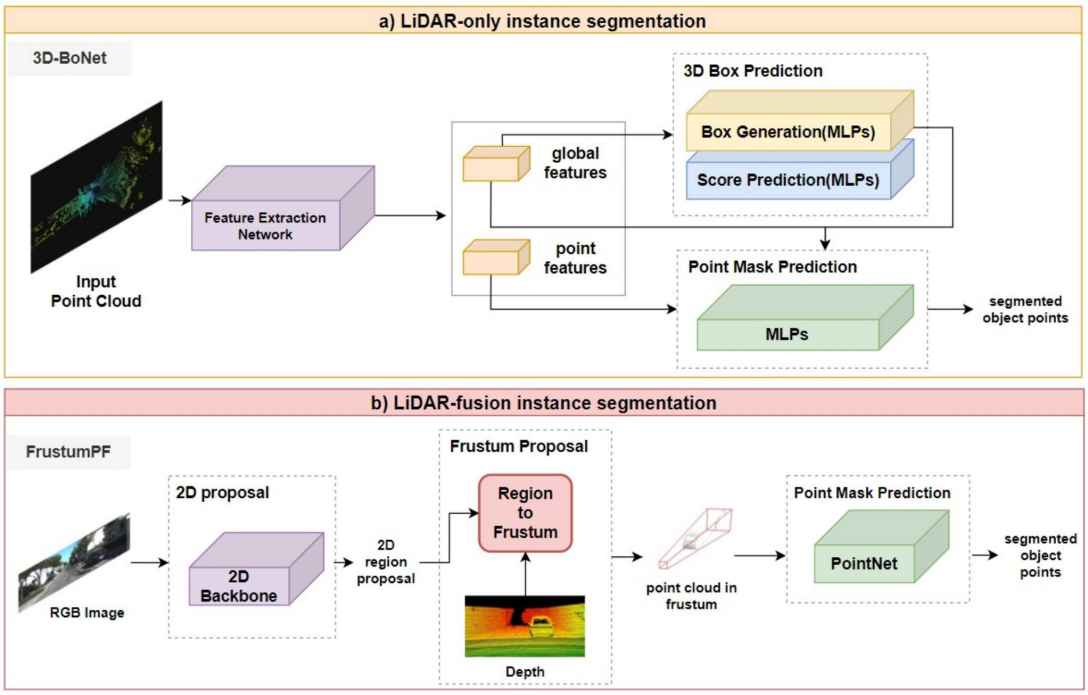

Figure 5. Typical frameworks for two categories of LiDAR-based instance segmentation: (a) LiDAR-only and (b) LiDAR-fusion methods.

modules to dropout noise and improve the accuracy.

6.2.2. LiDAR-fusion instance segmentation

Studies on LiDAR-fusion instance segmentation can also be divided into proposal-based and proposal-free. As for proposal-based methods, 3D-SIS[115] introduces a two-stage image and RGB-D data fused architecture, leveraging both geometric and color signals to jointly and semantically learn features, for instance, segmentation and detection. 3D-SIS consists of two branches, i.e., a 3D detection branch and a 3D mask workflow branch. The backbone of a 3D mask takes projected color, geometry features of each detected object, and 3D detection results as input and outputs final per-voxel mask prediction of each instance. For mask prediction, 3D convolutions with the same spatial resolutions that preserve spatial correspondence with raw point inputs are applied. Then, bounding box prediction generated from 3D-RPN is utilized to attain the key associated mask feature. The final mask of each instance is predicted by a 3D convolution which reduces the dimensionality of features. PanopticFusion[116] presents an online large-scale 3D reconstruction architecture that fuses RGB images and depth images. The 2D instance segmentation network based on Mask-CNN takes the incoming RGB frame as input and fuses both semantic and instance segmentation results to attain point-wise panoptic labels that are integrated into the volumetric map with depth data. As illustrated in Figure 5b, Qi *et al.*[117] proposed a pioneering object detection framework named Fustrum PointNets with point cloud and RGB-D fusion data as input. Frustum PointNets contains three modules: frustum proposal, 3D instance segmentation and a modal 3D box estimation, in order to fuse efficient mature 2D object detector into point cloud domain. The frustum point cloud is extracted from RGB-D data frustum proposal generation first and then is fed into set abstraction layers and point feature propagation layers based on PointNet to predict a mask for each instance by point-wise binary classification. When it comes to proposal-free methods, 3D-BEVIS[118] introduces a framework for 3D semantic and instance segmentation that transfers 2D bird's eye view (BEV) to 3D point space. This framework concentrates on both local point geometry and global context information. 3D instance segmentation network takes point cloud as input, which consists of 2D (i.e., RGB and height above ground) and 3D feature network jointly to exploit point-wise instance features and predicts final instance

Table 6. Summary of 3D semantic segmentation. "I", "mvPC", "vPC", "pPC" and "rm" stands for image, point cloud in multi-view based representation, point cloud in voxel-based representation, point cloud in point-based representation and range map separately

Category	Model	Modality & Representation	Architecture
LiDAR -Only	PointNet [1]	pPC	Point-wise MLP+T-Net+global max pooling
	PointNet++ [2]	pPC	Set abstraction (sampling, grouping, feature learning)+interpolation+skip link concatentation
	KWYND [92]	pPC	Feature network + neighbors definition + regional descriptors
	MPC [93]	pPC	PointNet++-like network+ gumbel subset sampling
	3D-MiniNet [97]	pPC	Fast 3D point neighbor search + 3D MiniNet + post-processing
	LU-Net [100]	pPC & vPC	U-Net for point cloud
	SceneEncoder [101]	pPC	Multi-hot scene descriptor + region similarity loss
	RPVNet [13]	rpc&pPC&vPC	Range-point-voxel fusion network(deep fusion + gated fusion module)
	SqueezeSeg [102]	mvPC	SqueezeNet + conditional random field
	PointSeg [103]	mvPC	SqueezeNet + new feature extract layers
	Pointwise [105]	pPC	Pointwise convolution operator
	Dilated [106]	pPC	Dilated point convolutions
LiDAR -Fusion	3DMV [107]	I & vPC	A novel end-to-end network(back propagation layer)
	SuperSensor [95]	I & mvPC	Associate architecture+360 degree sensor configuration
	MVPNet [108]	I & mvPC	Multi-view point regression network+geometric loss
	FuseSeg [3]	I & rPC	Point correspondece+feature level fusion
	PMF [109]	I & mvPC	Perspective projection+a two-stream network(fusion part)+perception-aware loss

Table 7. Summary of 3D instance segmentation. "I", "mvPC", "vPC", "pPC","FPC" and "rm" stands for image, point cloud in multi-view based representation, point cloud in voxel-based representation, point cloud in point-based representation, point cloud in Frustum representation and range map separately

Category	Model	Modality & Representation	Architecture
LiDAR-Only	GSPN [111]	pPC	Region-based PointNet(generative shape proposal network+Point RoIAlign)
	3D-BoNet [112]	pPC	Instance-level bounding box prediction + point-level mask prediction
	Joint [113]	pPC	Spatial embedding object proposal + local Bounding Boxes refinement
	SqueezeSeg [102]	mvPC	SqueezeNet + conditional random field
	SqueezeSegV2 [114]	mvPC	SqueezeSeg-like + context aggregation module
	3D-BEVIS [118]	mvPC	2D-3D deep model(2D instance feature+3D feature propagation)
LiDAR-Fusion	PanopticFusion [116]	I & vPC	Pixel-wise panoptic labels+a fully connected conditional random field
	Fustrum PointNets [117]	I & FPC	Frunstum proposal+3D instance segmentation(PointNet)

segmentation results through clustering.

7. DISCUSSION

As the upstream and key module of an autonomous vehicle, the perception system outputs its results to downstream modules (e.g., decision and planning modules). Therefore, the performance and reliability of the perception system determine the implementation of downstream tasks, thus affecting the performance of the whole autonomous system. For now, although sensor fusion (Table 8 shows a summary for LiDAR fusion architectures in this paper) can make up for the shortcomings of single LiDAR in bad weather and other aspects, there is still a huge gap between the algorithm design and practical applications in the real world. For this reason, it is necessary to be properly aware of existing open challenges and figure out possible directions to the solution. This section discusses the challenges and possible solutions for LiDAR-based 3D perception.

- **Dealing with large-scale point clouds and high-resolution images.** The need for higher accuracy has prompted researchers to consider larger scale point clouds and higher resolution images. Most the existing algorithms [2,29,36,119] are designed for small 3D point clouds (e.g., 4k points or 1 m × 1 m blocks) without good extending capability to larger point clouds (e.g., millions of points and up to 200 m × 200 m). However, larger point clouds come with a higher computational cost that is hard to afford for self-driving cars with limited computational processing ability. Several recent studies have focused on this problem and proposed some solutions. A deep learning framework for large-scale point clouds named SPG [120] partitions point

clouds adaptively to generate a compact yet rich representation by superpoint graph. RandLA-Net[121] leverages random sampling to downsample large-scale point clouds and local feature aggregation module to increase the receptive field size. SCF-Net[122] utilizes the spatial contextual features (SCF) module for large-scale point clouds segmentation. As for sensor fusion, deep learning approaches tackling the fusion of large-scale and high-resolution data should place more emphasis on point-based and multi-view based fusion approaches, which are more scalable than voxel-based ones. Overall, the trade-off between performance and computational cost is inevitable for real application of autonomous driving.

- **A robust representation of fused data.** For deep learning methods, how to pre-process the multi-modal input data is fundamental and important. Although there are several effective representations for point clouds, each of them has both disadvantages and advantages: voxel-based representation has tackled the ordering problem, but, when enlarging the scales of point cloud or increasing the resolution of voxel, the computational cost grows cubically. The quantity of point cloud that can be processed by point based representation methods is limited due to the permutation invariance and computational capacity. A consensus of a unified robust and effective representation for point clouds is necessary. For the data fused with images and point clouds, the representation approaches depend on fusion methods. Image representation-based methods mainly utilizes point clouds projected onto multi-view planes as additional branches of the image. (1) Image representation is not applicable for 3D tasks because the network output results on image plane. (2) Point representation-based methods leverages features or ROI extracted from RGB image as additional channels of point clouds. The performance of this representation is limited by the resolution differences between image (relatively high-resolution) and point clouds (relatively low-resolution). (3) Intermediate data representation methods introduce an intermediate data representation to (e.g., Frustum point cloud and voxelized point cloud). Voxel-based methods are limited in large scale, while frustum based methods have much potential to generate a unified representation based on contextual and structural information of RGB images and LiDAR point clouds.
- **Scene understanding tasks based on data sequences.** The spatiotemporal information implied in the temporally continuous sequence of point clouds and images has been overlooked for a period. Especially for sensor fusion methods, the mismatch of refresh rate between LiDAR and camera causes incorrect time-synchronization between inner perception system and surrounding environment. In addition, predictions based on spatiotemporal information can improve the performance of tasks, such as 3D object recognition, segmentation, and point cloud completion. Research has started to take temporal context into consideration. RNN, LSTM, and derived deep learning models are able to deal with temporal context. Huang *et al.*[123] proposed a multi-frame 3D object detection framework based on sparse LSTM. This work predict 3D objects in the current frame by sending features of each frame and the hidden and memory features from last frame into LSTM module. Yuan *et al.*[124] designed a temporal-channel transformer, whose encoder encodes multi-frame temporal-channel information and decoder decodes spatial-channel information for the current frame. TempNet[125] presents a lightweight semantic segmentation framework for large-scale point cloud sequences, which contains two key modules, temporal feature aggregation (TFA) and partial feature update (PFU). TFA aggregates features only on small portion of key frames with an attentional pooling mechanism, and PFU updates features with the information from non-key frame.

8. CONCLUSIONS

LiDAR captures point-wise information which is less sensitive to illumination than that of cameras. Moreover, it possesses invariance of scale and rigid transformation, showing a promising future in 3D scene understanding. Focusing on the LiDAR-only and LiDAR-fusion 3D perception, this paper first summarizes the LiDAR-based dataset as well as the evaluation metric and then presents a contemporary review of four key tasks: 3D classification, 3D object detection, 3D object tracking, and 3D segmentation. This work also points out the existing challenges and possible development direction. We always hold the belief that LiDAR-only and LiDAR-fusion 3D perception systems would feedback a precise and real-time description of the real-world

Table 8. Fusion stage and fusion methods of LiDAR-fusion tasks. Here, "I" represents image; "L" represents LiDAR point cloud; "R" represents Radar point cloud. Duplicate articles between classification and detection are merged to detection part

Task	Model	Input	FusionStage	Details of the Fusion Method
Classification	ImVoteNet [48]	I&L	Late fusion	Lift 2D image votes, semantic and texture cues to the 3D seed points
Detection	3D-CVF [64]	I&L	Early fusion	Adaptive Gated Fusion: spatial attention maps to mix features according to the region
	Roarnet [65]	I&L	Late fusion	3D detection conducts in-depth inferences recursively with candidate regions from 2D
	MV3D [12]	I&L	Early fusion	Region-based fusion via ROI pooling
	SCANet [46]	I&L	Early fusion	The multi-level fusion module fuses the region-based features
	MMF [47]	I&L	Multi fusion	Region-wise features from multiple views are fused by a deep fusion scheme
	Pointpainting [66]	I&L	Early fusion	Sequential fusion: project point cloud into the output of image semantic seg. network
	CM3D [67]	I&L	Early fusion	Two stage: point-wise feature and ROI-wise feature fusion
	MVDNet [28]	R&L	Early fusion	Region-wise features from two sensors are fused to improve final detection results
	CLOCs [69]	I&L	Late fusion	Output candidates of image and LiDAR point cloud before NMS are fused
Tracking	MSRT [85]	I&L	Late fusion	2D bbox is converted to 3D bbox that are fused to associate between sensor data
	MS3DT [86]	I&L	Early fusion	Object proposals generated by MV3D as input of the match network to link detections
	Compl.-YOLO [87]	I&L	Late fusion	Semantic Voxel Grid: project all relevant voxelized points into the semantic image
	F-Siamese [88]	I&L	Late fusion	2D region proposals are extruded into 3D viewing frustums
Semantic Seg.	3DMV [107]	I&L	Early fusion	3D geometry and per-voxel max-pooled images features are fed into two 3D conv.
	SuperSensor [95]	I&L	Late fusion	Segmentation results from the image space are transferred onto 3D points
	FuseSeg [3]	I&L	Early fusion	Fuse RGB and range image features with point correspondences and feed to net
	PMF [109]	I&L	Early fusion	Residual-based fusion modules fuse image features into LiDAR stream network
Instance Seg.	Pano.Fusion [116]	I&L	Late fusion	2D panoptic segmentation outputs are fused with depth to output volumetric map
	F-PointNets [117]	I&L	Late fusion	Frunstum proposal: extrud each 2D region proposal to a 3D viewing frustum

environment. We hope that this introductory survey serves as a step in the pursuit of a robust, precise, and efficient 3D perception system and guides the direction of its future development.

DECLARATIONS

Authors' contributions

Made substantial contributions to conception and design of the study and performed data analysis and interpretation: Wu D, Liang Z

Performed data acquisition, as well as provided administrative, technical, and material support: Chen G

Availability of data and materials

Not applicable.

Financial support and sponsorship

None.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Qi CR, Su H, Mo K, Guibas LJ. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. pp. 652–60. DOI
2. Qi CR, Yi L, Su H, Guibas LJ. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems* 2017;30. DOI
3. Krispel G, Opitz M, Waltner G, Possegger H, Bischof H. Fuseseg: Lidar point cloud segmentation fusing multi-modal data. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2020. pp. 1874–83. DOI
4. Xu D, Anguelov D, Jain A. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. pp. 244–53. DOI
5. Guo Y, Wang H, Hu Q, et al. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence* 2020. DOI
6. Li Y, Ma L, Zhong Z, et al. Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems* 2020;32:3412–32. DOI
7. Liu W, Sun J, Li W, Hu T, Wang P. Deep learning on point clouds and its application: A survey. *Sensors* 2019;19:4188. DOI
8. Ioannidou A, Chatzilari E, Nikolopoulos S, Kompatsiaris I. Deep learning advances in computer vision with 3d data: A survey. *ACM Computing Surveys (CSUR)* 2017;50:1–38. DOI
9. Feng D, Haase-Schütz C, Rosenbaum L, et al. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems* 2020;22:1341–60. DOI
10. Wang Z, Wu Y, Niu Q. Multi-sensor fusion in automated driving: A survey. *Ieee Access* 2019;8:2847–68. DOI
11. Cui Y, Chen R, Chu W, et al. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems* 2021. DOI
12. Chen X, Ma H, Wan J, Li B, Xia T. Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 1907–15. DOI
13. Xu J, Zhang R, Dou J, et al. RPVNet: a deep and efficient range-point-voxel fusion network for LiDAR point cloud segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 16024–33. DOI
14. Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE; 2012. pp. 3354–61. DOI
15. De Deuge M, Quadros A, Hung C, Douillard B. Unsupervised feature learning for classification of outdoor 3d scans. In: Australasian Conference on Robitics and Automation. vol. 2; 2013. p. 1. DOI
16. Uy MA, Pham QH, Hua BS, Nguyen T, Yeung SK. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019. pp. 1588–97. DOI
17. Varney N, Asari VK, Graehling Q. DALES: a large-scale aerial LiDAR data set for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops; 2020. pp. 186–87. DOI
18. Ye Z, Xu Y, Huang R, et al. Lasdu: A large-scale aerial lidar dataset for semantic labeling in dense urban areas. *ISPRS International Journal of Geo-Information* 2020;9:450. DOI
19. Li X, Li C, Tong Z, et al. Campus3d: A photogrammetry point cloud benchmark for hierarchical understanding of outdoor scene. In: Proceedings of the 28th ACM International Conference on Multimedia; 2020. pp. 238–46. DOI
20. Tan W, Qin N, Ma L, et al. Toronto-3D: a large-scale mobile lidar dataset for semantic segmentation of urban roadways. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops; 2020. pp. 202–3. DOI
21. Riemenschneider H, Bódis-Szomorú A, Weissenberg J, Van Gool L. Learning where to classify in multi-view semantic segmentation. In: European Conference on Computer Vision. Springer; 2014. pp. 516–32. DOI
22. Chang A, Dai A, Funkhouser T, et al. Matterport3D: Learning from RGB-D Data in Indoor Environments. In: 2017 International Conference on 3D Vision (3DV). IEEE Computer Society; 2017. pp. 667–76. DOI
23. Patil A, Malla S, Gang H, Chen YT. The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. In: 2019 International Conference on Robotics and Automation. IEEE; 2019. pp. 9552–57. DOI
24. Chang MF, Lambert J, Sangkloy P, et al. Argoverse: 3d tracking and forecasting with rich maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 8748–57. DOI
25. Kesten R, Usman M, Houston J, et al. Lyft level 5 av dataset 2019. *urlhttps://level5 lyft com/dataset* 2019. Available from: https://level5.lyft.com/dataset.
26. Sun P, Kretzschmar H, Dotiwalla X, et al. Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 2446–54. DOI
27. Caesar H, Bankiti V, Lang AH, et al. nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 11621–31. DOI
28. Qian K, Zhu S, Zhang X, Li LE. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 444–53. DOI
29. Zhang Z, Hua BS, Yeung SK. ShellNet:Efficient point cloud convolutional neural networks using concentric shells statistics. *2019 IEEE/CVF International Conference on Computer Vision* 2019:1607–16. DOI
30. Komarichev A, Zhong Z, Hua J. A-CNN: Annularly convolutional neural networks on point clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2019:7413–22. DOI
31. Wu Z, Song S, Khosla A, et al. 3d shapenets: a deep representation for volumetric shapes. In: Proceedings of the IEEE conference on

computer vision and pattern recognition; 2015. pp. 1912–20. DOI
32. Joseph-Rivlin M, Zvirin A, Kimmel R. Momen (e) t: Flavor the moments in learning to classify shapes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops; 2019. pp. 0–0. DOI
33. Zhao H, Jiang L, Fu C, Jia J. PointWeb: Enhancing local neighborhood features for point cloud processing. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 5560–68. DOI
34. Duan Y, Zheng Y, Lu J, Zhou J, Tian Q. Structural relational reasoning of point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 949–58. DOI
35. Yan X, Zheng C, Li Z, Wang S, Cui S. PointASNL: robust point clouds processing using nonlocal neural networks with adaptive sampling. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 5588–97. DOI
36. Wu W, Qi Z, Fuxin L. PointConv: Deep convolutional networks on 3D point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 9613–22. DOI
37. Boulch A. Generalizing discrete convolutions for unstructured point clouds. In: Biasotti S, Lavoué G, Veltkamp R, editors. Eurographics Workshop on 3D Object Retrieval. The Eurographics Association; 2019. pp. 71–78. DOI
38. Liu Y, Fan B, Xiang S, Pan C. Relation-shape convolutional neural network for point cloud analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 8895–904. DOI
39. Liu YC, Fan B, Meng G, et al. DensePoint: Learning densely contextual representation for efficient point cloud processing. *2019 IEEE/CVF International Conference on Computer Vision* 2019:5238–47. DOI
40. Mao J, Wang X, Li H. Interpolated convolutional networks for 3D point cloud understanding. In: 2019 IEEE/CVF International Conference on Computer Vision; 2019. pp. 1578–87. DOI
41. Rao Y, Lu J, Zhou J. Spherical fractal convolutional neural networks for point cloud recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2019:452–60. DOI
42. Ye M, Xu S, Cao T, Chen Q. DRINet: A dual-representation iterative learning network for point cloud segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 7447–56. DOI
43. Deng Q, Li X, Ni P, Li H, Zheng Z. Enet-CRF-Lidar: Lidar and camera fusion for multi-scale object recognition. *IEEE Access* 2019;7:174335–44. DOI
44. Wang H, Lou X, Cai Y, Li Y, Chen L. Real-time vehicle detection algorithm based on vision and lidar point cloud fusion. *J Sensors* 2019;2019:8473980:1–:9. DOI
45. Qi CR, Liu W, Wu C, Su H, Guibas LJ. Frustum pointnets for 3D object detection from RGB-D data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern recognition; 2018. pp. 918–27. DOI
46. Lu H, Chen X, Zhang G, et al. SCANet: spatial-channel attention network for 3D object detection. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE; 2019. pp. 1992–96. DOI
47. Liang M, Yang B, Chen Y, Hu R, Urtasun R. Multi-task multi-sensor fusion for 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 7345–53. DOI
48. Qi CR, Chen X, Litany O, Guibas LJ. Imvotenet: boosting 3d object detection in point clouds with image votes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 4404–13. DOI
49. Qi CR, Litany O, He K, Guibas LJ. Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE International Conference on Computer Vision; 2019. pp. 9277–86. DOI
50. Yang Z, Sun Y, Liu S, Shen X, Jia J. Ipod: Intensive point-based object detector for point cloud. *arXiv preprint arXiv:181205276* 2018. Available from: https://arxiv.org/abs/1812.05276.
51. Yang Z, Sun Y, Liu S, Shen X, Jia J. Std: Sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE International Conference on Computer Vision; 2019. pp. 1951–60. DOI
52. Shi S, Wang X, Li H. Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 770–79. DOI
53. Zarzar J, Giancola S, Ghanem B. PointRGCN: Graph convolution networks for 3D vehicles detection refinement. *arXiv preprint arXiv:191112236* 2019. Available from: https://arxiv.org/abs/1911.12236.
54. Shi S, Wang Z, Shi J, Wang X, Li H. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2020. DOI
55. Ye M, Xu S, Cao T. Hvnet: Hybrid voxel network for lidar based 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 1631–40. DOI
56. Li Z, Wang F, Wang N. LiDAR R-CNN: An efficient and universal 3D object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 7546–55. DOI
57. Zhou Y, Tuzel O. Voxelnet: end-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. pp. 4490–99. DOI
58. Lang AH, Vora S, Caesar H, et al. Pointpillars: fast encoders for object detection from point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 12697–705. DOI
59. He C, Zeng H, Huang J, Hua XS, Zhang L. Structure aware single-stage 3D object detection from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 11873–82. DOI
60. Dai J, Li Y, He K, Sun J. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems* 2016;29. Available from: https://proceedings.neurips.cc/paper/2016/hash/577ef1154f3240ad5b9b413aa7346a1e-Abstract.html.

61. Liu Z, Zhao X, Huang T, et al. Tanet: Robust 3d object detection from point clouds with triple attention. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. pp. 11677–84. DOI
62. Zheng W, Tang W, Jiang L, Fu CW. SE-SSD: self-ensembling single-stage object detector from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 14494–503. DOI
63. Qi CR, Zhou Y, Najibi M, et al. Offboard 3D object detection from point cloud sequences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 6134–44. DOI
64. Yoo JH, Kim Y, Kim J, Choi JW. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In: 16th European Conference on Computer Vision, ECCV 2020. Springer; 2020. pp. 720–36. DOI
65. Shin K, Kwon YP, Tomizuka M. Roarnet: a robust 3d object detection based on region approximation refinement. In: 2019 IEEE Intelligent Vehicles Symposium. IEEE; 2019. pp. 2510–15. DOI
66. Vora S, Lang AH, Helou B, Beijbom O. Pointpainting: sequential fusion for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 4604–12. DOI
67. Zhu M, Ma C, Ji P, Yang X. Cross-modality 3d object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2021. pp. 3772–81. DOI
68. Huang T, Liu Z, Chen X, Bai X. Epnet: Enhancing point features with image semantics for 3d object detection. In: European Conference on Computer Vision. Springer; 2020. pp. 35–52. DOI
69. Pang S, Morris D, Radha H. CLOCs: Camera-LiDAR object candidates fusion for 3D object detection. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2020. pp. 10386–93. DOI
70. Yan Y, Mao Y, Li B. Second: Sparsely embedded convolutional detection. *Sensors* 2018;18:3337. DOI
71. Cai Z, Vasconcelos N. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2019. DOI
72. Vaquero V, del Pino I, Moreno-Noguer F, et al. Dual-Branch CNNs for Vehicle Detection and Tracking on LiDAR Data. *IEEE Transactions on Intelligent Transportation Systems* 2020. DOI
73. Shi S, Guo C, Yang J, Li H. Pv-rcnn: the top-performing lidar-only solutions for 3d detection/3d tracking/domain adaptation of waymo open dataset challenges. *arXiv preprint arXiv:200812599* 2020. Available from: https://arxiv.org/abs/2008.12599.
74. Qi H, Feng C, Cao Z, Zhao F, Xiao Y. P2B: point-to-box network for 3d object tracking in point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 6329–38. DOI
75. Zhou X, Koltun V, Krahenbuhl P. Tracking objects as points. In: European Conference on Computer Vision; 2020. pp. 474–90. Available from: https://par.nsf.gov/servlets/purl/10220677.
76. Yin T, Zhou X, Krahenbuhl P. Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 11784–93. DOI
77. Li Y, Zhu J. A scale adaptive kernel correlation filter tracker with feature integration. In: European conference on computer vision. Springer; 2014. pp. 254–65. DOI
78. Mueller M, Smith N, Ghanem B. Context-aware correlation filter tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 1396–404. DOI
79. Zarzar Torano JA. Modular autonomous taxiing simulation and 3D siamese vehicle tracking [D]. King Abdullah University of Science and Technology. Thuwal, Saudi Arabia; 2019. Available from: https://repository.kaust.edu.sa/handle/10754/644892.
80. Giancola S, Zarzar J, Ghanem B. Leveraging shape completion for 3d siamese tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 1359–68. DOI
81. Zarzar J, Giancola S, Ghanem B. Efficient tracking proposals using 2D-3D siamese networks on lidar. *arXiv preprint arXiv:190310168* 2019. Available from: https://deepai.org/publication/efficient-tracking-proposals-using-2d-3d-siamese-networks-on-lidar.
82. Cui Y, Fang Z, Zhou S. Point siamese network for person tracking using 3D point clouds. *Sensors* 2020;20:143. DOI
83. Wang Z, Xie Q, Lai YK, et al. MLVSNet: Multi-Level Voting Siamese Network for 3D Visual Tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 3101–10. DOI
84. Zheng C, Yan X, Gao J, et al. Box-aware feature enhancement for single object tracking on point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 13199–208. DOI
85. Manghat SK, El-Sharkawy M. A multi sensor real-time tracking with LiDAR and camera. In: 2020 10th Annual Computing and Communication Workshop and Conference. IEEE; 2020. pp. 0668–72. DOI
86. Frossard D, Urtasun R. End-to-end learning of multi-sensor 3d tracking by detection. In: 2018 IEEE International Conference on Robotics and Automation. IEEE; 2018. pp. 635–42. DOI
87. Simon M, Amende K, Kraus A, et al. Complexer-YOLO: real-time 3D object detection and tracking on semantic point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops; 2019. pp. 0–0. DOI
88. Zou H, Cui J, Kong X, et al. F-siamese tracker: a frustum-based double siamese network for 3d single object tracking. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2020. pp. 8133–39. DOI
89. Kirillov A, He K, Girshick R, Rother C, Dollár P. Panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 9404–13. Available from: https://openaccess.thecvf.com/content_CVPR_2019/papers/Kirillov_Panoptic_Segmentation_CVPR_2019_paper.pdf.
90. Milioto A, Behley J, McCool C, Stachniss C. Lidar panoptic segmentation for autonomous driving. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2020. pp. 8505–12. DOI
91. Behley J, Milioto A, Stachniss C. A benchmark for LiDAR-based panoptic segmentation based on KITTI. In: 2021 IEEE International

Conference on Robotics and Automation. IEEE; 2021. pp. 13596–603. DOI
92. Engelmann F, Kontogianni T, Schult J, Leibe B. Know what your neighbors do: 3D semantic segmentation of point clouds. In: Proceedings of the European Conference on Computer Vision Workshops; 2018. pp. 395–409. DOI
93. Yang J, Zhang Q, Ni B, et al. Modeling Point Clouds With Self-Attention and Gumbel Subset Sampling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2019:3318–27. DOI
94. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. pp. 3431–40. DOI
95. Varga R, Costea A, Florea H, Giosan I, Nedevschi S. Super-sensor for 360-degree environment perception: Point cloud segmentation using image features. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems; 2017. pp. 1–8. DOI
96. Piewak F, Pinggera P, Schafer M, et al. Boosting lidar-based semantic labeling by cross-modal training data generation. In: Proceedings of the European Conference on Computer Vision Workshops; 2018. pp. 0–0. DOI
97. Alonso I, Riazuelo L, Montesano L, Murillo AC. 3D-MiniNet: Learning a 2D Representation From Point Clouds for Fast and Efficient 3D LIDAR Semantic Segmentation. *IEEE Robotics and Automation Letters* 2020;5:5432–39. DOI
98. Alonso I, Riazuelo L, Murillo AC. MiniNet: an efficient semantic segmentation convnet for real-time robotic applications. *IEEE Transactions on Robotics* 2020;36:1340–47. DOI
99. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer; 2015. pp. 234–41. DOI
100. Biasutti P, Lepetit V, Aujol JF, Brédif M, Bugeau A. LU-Net: an efficient network for 3D LiDAR point cloud semantic segmentation based on end-to-end-learned 3D features and U-Net. *2019 IEEE/CVF International Conference on Computer Vision Workshop* 2019:942–50. DOI
101. Xu J, Gong J, Zhou J, et al. SceneEncoder: scene-aware semantic segmentation of point clouds with a learnable scene descriptor. In: 29th International Joint Conference on Artificial Intelligence (IJCAI 2020). International Joint Conferences on Artificial Intelligence; 2021. pp. 601–7. DOI
102. Wu B, Wan A, Yue X, Keutzer K. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In: 2018 IEEE International Conference on Robotics and Automation. IEEE; 2018. pp. 1887–93. DOI
103. Wang Y, Shi T, Yun P, Tai L, Liu M. Pointseg: Real-time semantic segmentation based on 3d lidar point cloud. *arXiv preprint arXiv:180706288* 2018. Available from: https://arxiv.org/abs/1807.06288.
104. Iandola FN, Moskewicz MW, Ashraf K, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *ArXiv* 2017;abs/1602.07360. Available from: https://arxiv.org/abs/1602.07360.
105. Hua B, Tran M, Yeung S. Pointwise convolutional neural networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018. pp. 984–93. DOI
106. Engelmann F, Kontogianni T, Leibe B. Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds. In: 2020 IEEE International Conference on Robotics and Automation. IEEE; 2020. pp. 9463–69. DOI
107. Dai A, Nießner M. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In: Proceedings of the European Conference on Computer Vision; 2018. pp. 452–68. DOI
108. Wang J, Sun B, Lu Y. Mvpnet: Multi-view point regression networks for 3d object reconstruction from a single image. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. pp. 8949–56. DOI
109. Zhuang Z, Li R, Jia K, et al. Perception-aware Multi-sensor Fusion for 3D LiDAR Semantic Segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 16280–90. DOI
110. Su H, Jampani V, Sun D, et al. SPLATNet: Sparse Lattice Networks for Point Cloud Processing. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2018:2530–39. DOI
111. Yi L, Zhao W, Wang H, Sung M, Guibas L. GSPN: Generative shape proposal network for 3D instance segmentation in point cloud. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2019:3942–51. DOI
112. Yang B, Wang J, Clark R, et al. Learning object bounding boxes for 3D Instance segmentation on point clouds. *Advances in Neural Information Processing Systems* 2019;32:6740–49. Available from: https://proceedings.neurips.cc/paper/2019/file/d0aa518d4d3bfc721aa0b8ab4ef32269-Paper.pdf.
113. Zhou D, Fang J, Song X, et al. Joint 3D instance segmentation and object detection for autonomous driving. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 1836–46. DOI
114. Wu B, Zhou X, Zhao S, Yue X, Keutzer K. SqueezeSegV2: improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. *2019 International Conference on Robotics and Automation* 2019:4376–82. DOI
115. Hou J, Dai A, Nießner M. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2019:4416–25. DOI
116. Narita G, Seno T, Ishikawa T, Kaji Y. PanopticFusion: online volumetric semantic mapping at the level of stuff and things. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 2019:4205–12. DOI
117. Qi CR, Liu W, Wu C, Su H, Guibas L. Frustum pointnets for 3D object detection from RGB-D data. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2018:918–27. DOI
118. Elich C, Engelmann F, Kontogianni T, Leibe B. 3D Bird's-eye-view instance segmentation. In: German Conference on Pattern Recognition. Springer; 2019. pp. 48–61. DOI
119. Komarichev A, Zhong Z, Hua J. A-CNN: annularly convolutional neural networks on point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 7413–22. DOI

120. Landrieu L, Simonovsky M. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. pp. 4558–67. DOI
121. Hu Q, Yang B, Xie L, et al. Randla-net: efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 11108–17. DOI
122. Fan S, Dong Q, Zhu F, et al. SCF-Net: learning spatial contextual features for large-scale point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 14504–13. DOI
123. Huang R, Zhang W, Kundu A, et al. An lstm approach to temporal 3d object detection in lidar point clouds. In: European Conference on Computer Vision. Springer; 2020. pp. 266–82. DOI
124. Yuan Z, Song X, Bai L, Wang Z, Ouyang W. Temporal-channel transformer for 3d lidar-based video object detection for autonomous driving. *IEEE Transactions on Circuits and Systems for Video Technology* 2021. DOI
125. Zhou Y, Zhu H, Li C, et al. TempNet: Online Semantic Segmentation on Large-Scale Point Cloud Series. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 7118–27. DOI