Noname manuscript No.
(will be inserted by the editor)

Multi-Modal 3D Object Detection in Autonomous Driving: a Survey

Yingjie Wang* · **Qiuyu Mao*** · **Hanqi Zhu** · **Yu Zhang** · **Jianmin Ji** · **Yanyong Zhang**

Received: date / Accepted: date

Abstract In the past few years, we have witnessed rapid development of autonomous driving. However, achieving full autonomy remains a daunting task due to the complex and dynamic driving environment. As a result, self-driving cars are equipped with a suite of sensors to conduct robust and accurate environment perception. As the number and type of sensors keep increasing, combining them for better perception is becoming a natural trend. So far, there has been no in-depth review that focuses on multi-sensor fusion based perception. To bridge this gap and motivate future research, this survey devotes to review recent fusion-based 3D detection deep learning models that leverage multiple sensor data sources, especially cameras and LiDARs. In this survey, we first introduce the background of popular sensors for autonomous cars, including their common data representations as well as object detection networks developed for each type of sensor data. Next, we discuss some popular datasets for multi-modal 3D object detection, with a special focus on the sensor data included in each dataset. Then we present in-depth reviews of recent multi-modal 3D detection networks by considering the following three aspects of the fusion: fusion location, fusion data representation, and fusion granularity. After a detailed review, we discuss open challenges and point out possible solutions. We hope that our detailed review can help researchers to embark investigations in the area of multi-modal 3D object detection.

Keywords Multi-modal Detection · Sensor Fusion · 3D Object Detection · Autonomous Driving

*equal contribution

Yingjie Wang
University of Science and Technology of China
E-mail: yingjiewang@mail.ustc.edu.cn

Qiuyu Mao
University of Science and Technology of China
E-mail: qymao@mail.ustc.edu.cn

Hanqi Zhu
University of Science and Technology of China
E-mail: zhuhanqi@mail.ustc.edu.cn

Yu Zhang
University of Science and Technology of China
E-mail: yuzhang@ustc.edu.cn

Jianmin Ji
University of Science and Technology of China
E-mail: jianmin@ustc.edu.cn

Yanyong Zhang, corresponding author
University of Science and Technology of China
E-mail: yanyongz@ustc.edu.cn

1 Introduction

Recent breakthroughs in deep learning and computer vision have enabled the rapid development of autonomous driving, which frees the hands of the drivers, promises decreased traffic congestion, improves road safety, and reduces carbon emissions. The potential of autonomous driving is, however, not yet fully unleashed, mainly due to unsatisfactory perception performance in real-world driving scenarios. As a result, even autonomous vehicles (AVs) have seen applications in many confined and controlled environments, deploying them in urban environments still poses great technological challenges (Guo et al., 2019; Urmson et al., 2008).

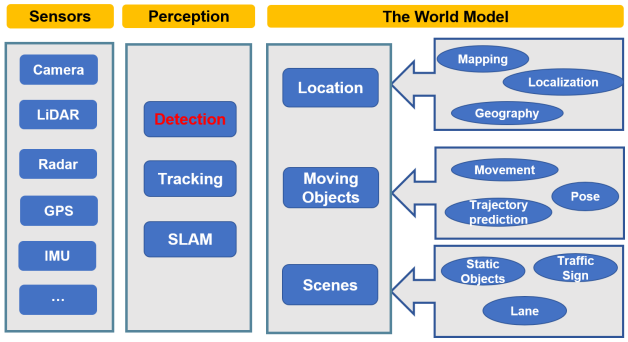


Fig. 1 The illustration of a typical perception subsystem for autonomous driving

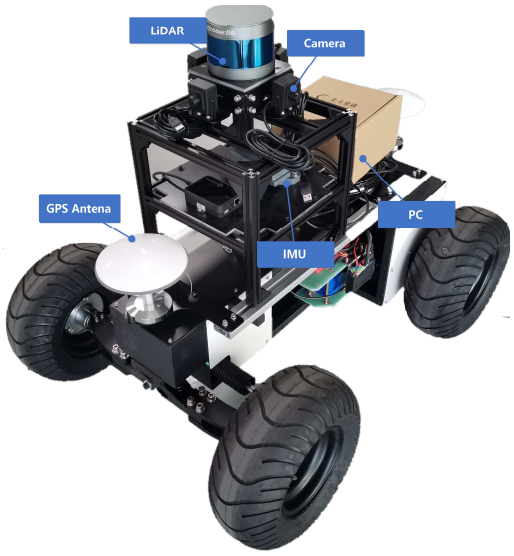


Fig. 2 The autonomous car *Sonic* with sensors including one LiDAR (Velodyne VLP-16), four cameras, one GPS, and so on. Note that the image is modified from (Zhang et al., 2020b)

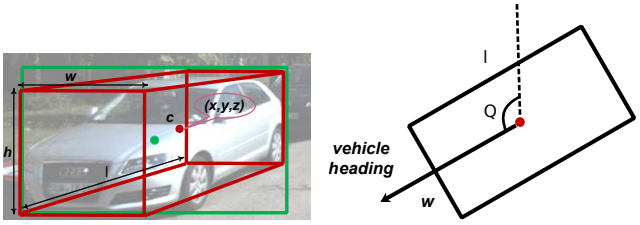
AVs are usually equipped with a perception subsystem to detect and track moving objects in real-time (illustrated in Fig. 1). The perception subsystem is to take the data from an array of sensors as input; after a series of processing steps, it outputs the learned knowledge about the environment, other objects (such as cars), as well as the AV itself. As illustrated in Fig. 2, sensors equipped on the AV usually include cameras, LiDARs (i.e., Light Detection And Ranging sensor), Radar (Radio detection and ranging), GPS (Global Positioning System), IMUs (inertial measurement units), and others (Urmson et al., 2009; Zhang et al., 2020b).

Specifically, there are three fundamental requirements for the perception subsystem. Firstly, it needs to be *accurate* and gives a precise description of the driving environment. Secondly, it needs to be *robust* and works properly under adverse weather, in situations that are encountered for the first time, and even when some sensors are degraded or even fail. Thirdly, it needs to be *real-time* and gives quick feedback. To satisfy the above requirements, the perception subsystem executes several important tasks at the same time, such as object detection, tracking, Simultaneous Localization And Mapping (SLAM), etc.

1.1 3D Object Detection through Single Sensor Modality

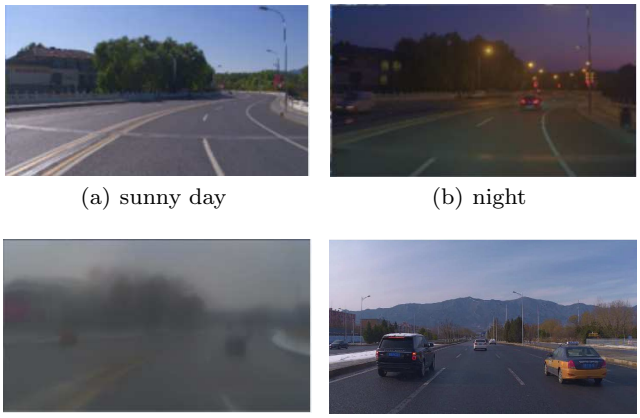
As a fundamental task in perception, *object detection* aims to identify all objects of interest in the sensor data (such as camera images) and determine their positions and categories (such as vehicles, cycles, buses, etc.). With the development of deep learning based models such as RCNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015), and Faster R-CNN (Ren et al., 2017), 2D object detection has achieved unprecedented success and popularity, and thus a variety of new models continue to emerge (Redmon et al., 2016; Liu et al., 2016; Lin et al., 2017; Lin et al., 2017). 2D detection cannot provide sufficient information that AVs needed to perceive the environment. It can only provide the 2D bounding box of the object and the confidence score of the corresponding category. In the real world, however, objects have three-dimensional shapes with rotational angles. Compared to the well-studied 2D object detection, 3D object detection helps create more accurate spatial path planning and navigation but is a more challenging task. In the task of 3d object detection, more output parameters are needed to specify the 3D-oriented bounding boxes around objects. As shown in fig. 3, we need to predict the central 3D coordinates c, length l, width w, height h, and deflection angle of the object θ, to draw the red 3D bounding box. Obviously, 2D object detection cannot meet the requirements of autonomous driving environment perception for lack of object location in the real-world coordinate system. In this paper, we focus on 3D object detection tasks towards autonomous driving, which can be further divided into the following categories according to the use of sensor type.

3D Object Detection Using Cameras. Cameras provide 2D images, which can be used for 3D object detection. A series of mature methods in 2D object detection have been developed in recent years, which can be reused in 3D object detection. In fact, it has been shown that image-based 3D object detection methods can achieve satisfactory performance at low expenses, often outperforming human experts (Silver et al., 2016; Mnih et al., 2015). However, camera-based 3D object detection has shortcomings. For example, mono-camera methods don't provide reliable 3D geometry information, which is essential for 3D object detection (Huang



(a) 2D and 3D bounding boxs (b) Bird's eye view (BEV) of the 3D box

Fig. 3 Example 2D (green) and 3D (red) object detection results



(a) sunny day (b) night

(c) foggy day (d) snowy day

Fig. 4 The camera images are to demonstrate the diversity of weather (Zhou et al., 2020b)

et al., 2017; Chen et al., 2019). Also, cameras suffer from high computational cost, high-occlusion, and textureless environments (Kim et al., 2016). Furthermore, as illustrated in Fig.4, the camera-based perception subsystem struggles under adverse conditions such as poor lighting or heavy fog, which limits their all-weather capabilities.

3D Object Detection Using LiDARs. A more popular 3D detection approach is through LiDAR point clouds. Unlike images, point clouds naturally have strong geometric information, suitable for 3D object detection. The advantages of LiDAR also lie in its great ranging ability and penetrability, which can provide high-quality spatial information without the target occlusion problem. Moreover, LiDARs are resistant to adverse lighting conditions. With the help of LiDARs, self-driving vehicles can see farther and more clearly. At present, LiDAR-based methods achieve better detection accuracy and higher recall than camera-based methods (Chen et al., 2017). As far as the KITTI 3D benchmark is concerned, the top performer MonoFlex (Zhang et al., 2021b), which takes monocular images as input, achieves 13.89% mAP for the moderate category, while quite a few LiDAR-based methods (Deng et al., 2020; Shi et al., 2020a) can obtain over 80 mAP. However, LiDAR-only algorithms are not yet ready to be widely deployed on AVs for the following reasons: 1) LiDARs are expensive and bulky, especially compared with cameras. 2) LiDARs captures the low resolution of the point clouds (ranging from 16 to 128 channels) with low refresh rate, which is insufficient to meet the requirements of real-time detection. 3) The working distance of the LiDAR is rather limited, point clouds far away from the LiDAR are extremely sparse (Zhang et al., 2021a). 4) LiDARs do not work properly under extreme severe weather conditions such as heavy rain or snow since the transmission distance of laser light is greatly affected.

3D Object Detection Using Other Sensors. It is remarkable that compared with these two, some other sensors are more resistant to environmental interference, such as millimeter wave (mmWave in short) radar sensors and infrared cameras. MmWave radar measure speed by the Doppler effect, providing long-distance and accurate measurement of the surrounding environment. They are considerably cheaper than LiDARs, resistant to adverse weather conditions, and insensitive to lighting variations. However, compared with the other two sensors, there are limited large-scale and public datasets containing mmWave radar data. Moreover, due to the low resolution and high specularity of the mmWave radar, it is hard to obtain contextual or perceptual information and cannot directly detect the shape of an object (Zewge et al., 2019; Lee, 2020). MmWave radars have a relatively poor ability to identify objects compared with LiDARs and cameras.

To further fill the gap in reliable solutions for nighttime autonomous driving, infrared cameras have become indispensable. The infrared camera adopts infrared thermal imaging technology, which is not affected by harsh environments such as night, reflective surfaces, heavy rain, etc. The infrared camera can detect objects that are more than 300 meters away (Andrai et al., 2017). With them, drivers have more time to react to sudden changes in traffic conditions, thus greatly improving driving safety. Compared with the price of tens of thousands dollars of the LiDAR sensor, infrared cameras are cost-effective.

1.2 3D Object Detection through Multi-modal Fusion

In the realistic self-driving situations, performing object detection through a single type of sensor is far from being sufficient. Firstly, each type of sensor has its own inherent shortcomings. For example, Camera only methods suffer from object occlusion; LiDAR methods

are hampered by lower input data resolution than images, especially at long ranges. Fig. 5 illustrates two cases clearly. Secondly, to realize actual autonomous driving, we need to consider a wide range of weather, road, and traffic conditions. The perception subsystem has to offer good sensing results in all different conditions, which is hard to achieve by relying upon a single type of sensor. For example, when entering a tunnel, the camera will have underexposure and overexposure problems due to the sudden change of light. The LiDAR sensors also suffer from rainy and foggy weather. It's obvious that single-sensor systems don't work well under adverse conditions.

To mitigate these challenges, many fusion-based 3D detection schemes have been proposed and studied. In these methods, data from multiple types of sensors that have complementary characteristics are used to boost performance and reduce cost. Though sensor fusion promises considerable benefits, conducting efficient and effective fusion poses severe challenges to the underlying system design. We next discuss these challenges. On one hand, sensors of different types are not synchronized either temporally or spatially. In the time domain, it is hard to guarantee that the data can be collected at the same time because the acquisition cycles of different sensors are independent of each other. In the space domain, sensors have different angles of view when they are deployed[1]. On the other hand, when devising a fusion method, we need to pay close attention to several issues. Below we list a few problems as examples:

- *Multi-Sensor Calibration and Data Alignment:* Due to the heterogeneity of multi-modal data (illustrated in Table 1), it is difficult to align them precisely either in the raw input space or in the feature space.
- *Information Loss:* To convert the sensor data into a format that can be aligned and processed at a computation cost, loss of information is inevitable.
- *Cross-modality Augmentation:* Data augmentation plays a pivotal role in 3D object detection to reduce model over-fitting (Yoo et al., 2020a), which is usually caused by insufficient training data. Augmentation strategies such as global rotation and random flip are widely applied by single-modality methods but are absent in many multi-sensor fusion methods due to concerns of multi-sensor consistency.
- *Dataset and Metrics:* There are limited amount of high-quality, publicly usable multi-modal datasets. Even existing datasets suffer from the problem of small size, class imbalance, labeling errors, and so on. What's more, there has been no metrics for the datasets that specifically evaluates the effectiveness of multi-sensor fusion so far, which brings difficulty in making comparisons between multi-sensor fusion methods.

[1] To compensate for these factors, human annotators use both the camera images together with the LiDAR point clouds to create the ground-truth bounding boxes (Geiger et al., 2012).

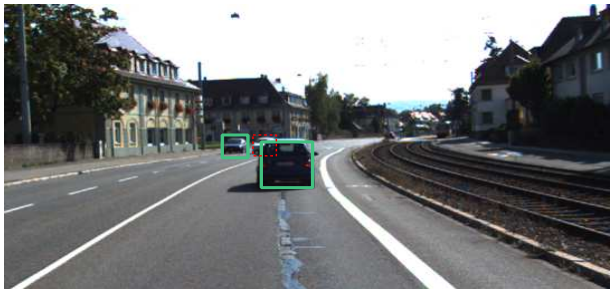
Table 1 Comparison of point clouds and RGB images

	Point Cloud	Image
dimension	3D	2D
coordinate	projective	euclidean
structure	irregular	regular
permutation	orderless	ordered
resolution	low	high

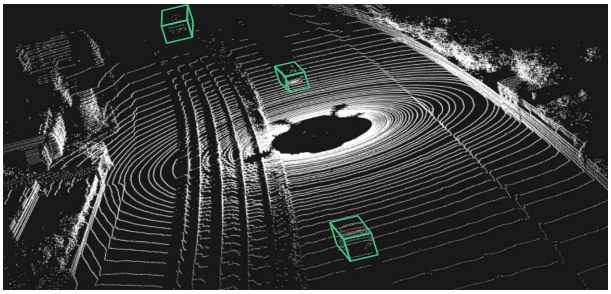
To sum up, sensor fusion has become a necessary module for the perception subsystem to achieve satisfactory performance, but many designs and implementation challenges need to be addressed before we can truly enjoy its benefits. Towards this objective, we set out to conduct a systematic review of recent fusion-based 3D object detection methods. Such a review can help pinpoint technical challenges in the sensor fusion, and help us compare and contrast various models proposed to address these challenges. In particular, since cameras and LiDARs are the most common sensors for autonomous driving, our review mainly focuses on the fusion of these two types of sensor data.

Previous surveys on deep learning based multi-modal fusion methods (Feng et al., 2021; Arnold et al., 2019; Cui et al., 2021) cover a broad range of sensors, including radars, cameras, LiDARs, Ultrasonic sensors, etc, and provide a brief review on a broad list of topics including multi-object detection, tracking, environment reconstruction, etc. While they serve as a useful guide for readers to browse through the general area, our survey serves as a distinctly different purpose: it targets at researchers who would like to carefully investigate the field of multi-modal 3D detection. As such, our survey intends to provide a deep and detailed review of recent research on this topic. Our contributions are summarized as below:

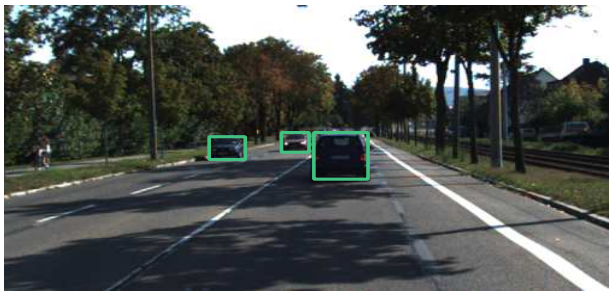
- We review multi-modal based 3D object detection methods according to various combinations of input sensor data. In particular, range images, which is an information-complete form of the LiDAR point cloud, has not been discussed in the past review articles. In addition, the representation of the pseudo-LiDARs (generated by camera images) have not been discussed either.
- We take a close look at the development of multi-modal based 3D object detection strategies from multiple perspectives. We particularly focus on key



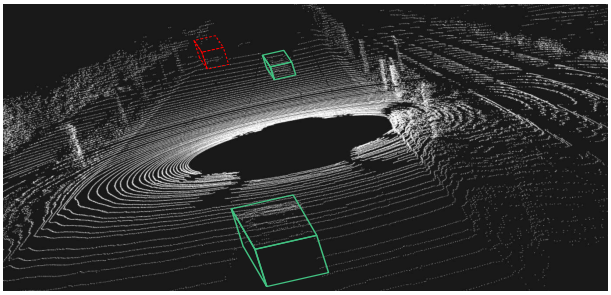
(a) Camera only detector #1



(b) LiDAR only detector #1



(c) Camera only detector #2



(d) LiDAR only detector #2

Fig. 5 Illustration of typical problems for Single-Modal detectors. For scene #1, (a) shows a single camera cannot avoid the occlusion problem while the detection result of LiDAR only detector in (b) is correct; For scene #2, camera only detector in (c) performs well while LiDAR only detector shows the difficulty of detecting faraway vehicles with just a few LiDAR points in (d). Note that dashed red boxes stand for missed objects

issues such as how these methods can achieve cross-modality data alignment, how to reduce information loss, etc.
- We present the most up-to-date review of the recent camera-LiDAR fusion-based detection methods. Meanwhile, we also summarize recent multi-modal datasets that can be used for 3D object detection.
- We carefully discuss some challenging issues in this field as well as possible solutions, which can hopefully inspire some future research.

In this paper, we first provide a brief background on several common AV sensors, their typical data representations, as well as the corresponding detection networks in Section 2. In Section 3, We present a summary of popular autonomous driving datasets. In Section 4, we categorize multi-modal fusion methods based on three factors: fusion location, representation of the fusion input, and fusion granularity. Finally, we discuss open challenges and possible solutions in Section 5.

2 Background

In this section, we provide the background overview of typical sensors used in autonomous driving, including data representation and 3D object detection methods that rely on each type of sensors. In particular, we mainly focus our discussions on cameras and LiDAR sensors. In the end, we introduce other sensors that can be used for 3D object detection.

2.1 3D Object Detection through Cameras

Cameras are the most common sensors for self-driving cars. Several types of cameras have been widely deployed, each with pros and cons. Monocular cameras provide detailed information in the form of pixel intensities, which reveal shape and texture properties (Enzweiler and Gavrila, 2009; Andriluka et al., 2010). The shape and texture information can also be used to detect lane geometry, traffic signs, type of objects, etc. On the downside, the main disadvantage of monocular cameras is the lack of depth information, which is required for accurate object size and position estimation for AVs (Carr et al., 2012). A stereo camera setup can gain higher point density and more importantly provide a dense depth map (Lee et al., 2011; Engelberg and Niem, 2009). Besides, multi-view cameras can cover different ranges of scenes through different cameras and capture the depth maps more accurately (Park et al., 2009; Kim and Woo, 2005a,b). Meanwhile, the complexity and cost will also increase exponentially. Other

camera types that offer depth estimation include Time-of-Flight (ToF) cameras infer depth of surroundings by measuring the delay between emitting and receiving modulated infrared pulses (Lee, 2014). ToF cameras have lower resolution compared to stereo cameras but have lower integration price and computational cost. ToF cameras have been applied for vehicle safety applications.

Typical Representation for Images: The most effective and common representation provided by cameras is 2D images. Images contain rich texture information of the driving surroundings. Originally, we use traditional feature extraction techniques, such as processing images to Histogram of Oriented Gradient (HOG) or Local Binary Pattern (LBP). In recent years, due to the rapid development of deep learning, powerful 2D convolutional neural networks (CNNs) are used to process images (Rojas and Crisman, 1997; Kim, 2014; Simonyan and Zisserman, 2014; He et al., 2016). In this way, features of the RGB images is extracted automatically by neural networks. Next, we present the following two popular data representations for RGB images.

feature map: A feature map, aka activation map, is the activation for the given filters. Feature maps is the output of each convolutional layer. During the deep learning-based image processing, deep neural networks take 2D images as input and extracts feature maps with a set of convolution operations (Kim, 2014; Yosinski et al., 2014). Specifically, we can obtain feature maps by applying classic pre-trained backbone networks such as VGG-16 (Simonyan and Zisserman, 2014), ResNet (He et al., 2016), DenseNet (Huang et al., 2017), and so on. As Fig.6 (b) perfectly illustrated, we can easily notice the activation on the edges and textures is bigger than other areas within the image.

mask: To directly get pixel-wise information, e.g. semantic segmentation result, we can adopt the representation of masks (Long et al., 2015). Image segmentation is a basic computer vision task, in which we should label each pixel with its category. This helps to understand the image at a much fine-grained level, i.e., the pixel level. This pixel-level information is helpful to alignment operations of deep fusion in subsequent steps. Image segmentation networks are commonly used today in autonomous driving, e.g. DeepLabV3 (Chen et al., 2018a), Mask-RCNN (He et al., 2017), and lightweight network Unet (Ronneberger et al., 2015) to name a few.

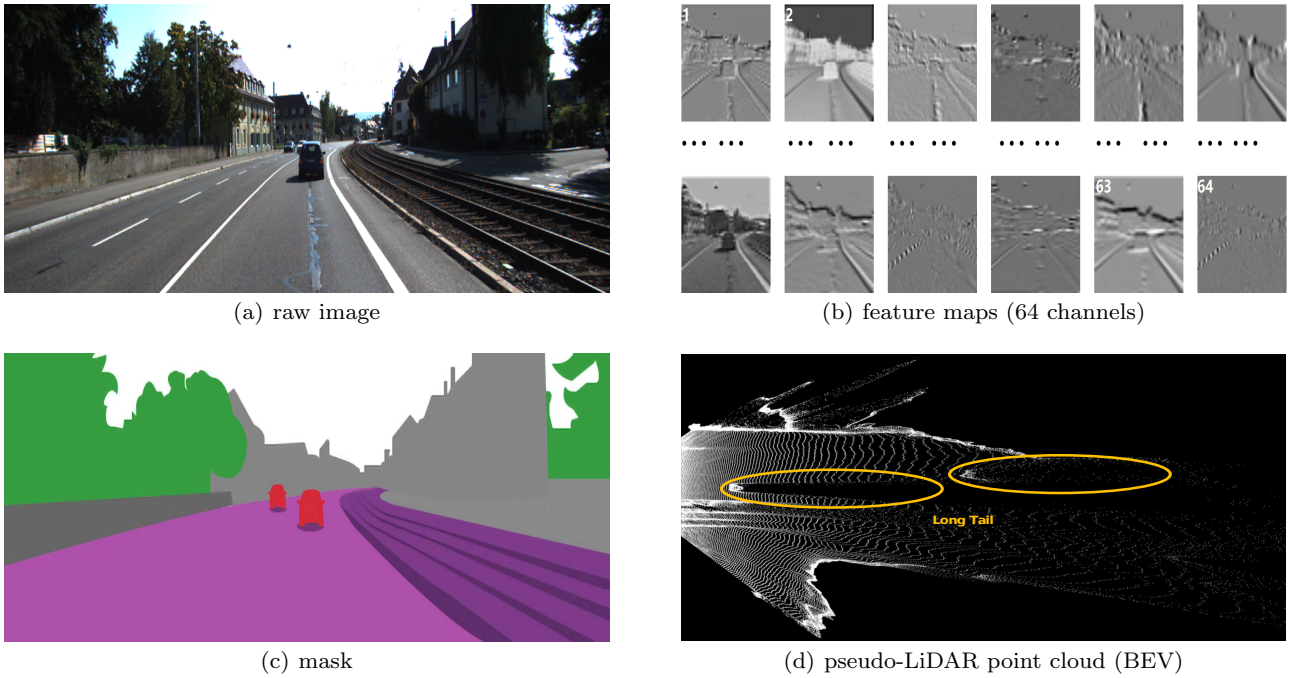
pseudo-LiDAR: Given stereo or monocular images, we can first predict the depth map (Fu et al., 2018; Chang and Chen, 2018), followed by back-projecting it into a 3D point cloud in the LiDAR coordinate system. We refer to this representation as pseudo-LiDAR, and process it exactly like real LiDAR (Wang et al., 2019; You et al., 2020). A distinction between the pseudo-LiDAR and the LiDAR point cloud is the density of the point cloud. Although a high-cost LiDAR can provide high-resolution point clouds, the number of LiDAR points is still at least one order of magnitude less than the pseudo-LiDAR point cloud. However, as shown in Fig.6 (d) with yellow circles, the pseudo-LiDAR representation often has a **long tail** problem, because the estimated depth is not accurate around the boundaries of the object.

Image-Based 3D Object Detection: Monocular cameras are readily available and inexpensive but don't provide accurate depth information. Methods can achieve real-time 3D object detection using only monocular RGB images (He and Soatto, 2019; Ma et al., 2019; Liu et al., 2020; Qin et al., 2019a; Wang et al., 2019; Chen et al., 2016; Mousavian et al., 2017). For example, Mousavian et al. (2017) first predicts the 2D bounding box, and then uses a neural network to estimate the missing depth information and then upgrades the 2D bounding box to 3D space. Chen et al. (2016) proposes to sample candidate bounding boxes in 3D and score their 2D projection based on the alignment with multiple semantic priors: shape, instance segmentation, context, and location.

In addition to monocular 3D object detection, there are also some methods that use stereo images to generate dense point clouds to conduct 3D object detection tasks (You et al., 2020; Li et al., 2019; Pon et al., 2020; Qin et al., 2019b; Chen et al., 2018b). For example, Chen et al. (2018b) focus on generating 3D proposals by encoding object size prior, ground-plane prior, and depth information (e.g., free space, point cloud density) into an energy function. 3D proposals are then used to regress the object pose and 2D boxes with the R-CNN approach. Li et al. (2019) adds extra branches after stereo Region Proposal Network (RPN) to predict sparse keypoints, viewpoints, and object dimensions, then they are combined with 2D left-right boxes to calculate coarse 3D object bounding boxes. The accurate 3D bounding boxes are recovered by a region-based photometric alignment. However, the depth estimation is computationally expensive and doesn't perform well with textureless regions or during night time.

2.2 3D Object Detection through LiDARs

LiDAR sensors use the laser as the light source to complete remote sensing measurement. LiDARs detect the



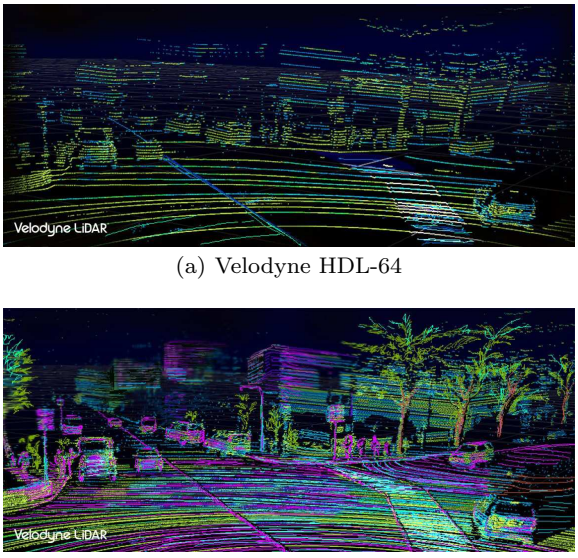
(a) raw image

(b) feature maps (64 channels)

(c) mask

(d) pseudo-LiDAR point cloud (BEV)

Fig. 6 A RGB image and its typical data representations. The raw image is from KITTI training set. We use a pretrained AlexNet to obtain feature maps of 64 channels. Notice that we adopt the BEV of the pseudo-LiDAR point cloud for better visualization



(a) Velodyne HDL-64

(b) Velodyne VLS-128

Fig. 7 Point clouds obtained by two LiDAR sensors with diffrent number of channels on the same scene (Repairer Driven News, 2018)

lightwave signal between the LiDAR sensor and the detected object (Wandinger, 2005), it continuously emits laser lights and collects the information of the reflection points to obtain a full range of environmental information. Meanwhile, the LiDAR also records the time and horizontal angle of the point. When the LiDAR rotates one circle, all the reflected point coordinates forms a point cloud. As an active sensor, external illumination is not required thus more reliable detection can be achieved under adverse weather and extreme lighting conditions. The resolution of LiDAR ranges from 16 channels to 128 channels usually. As shown in Fig.7, the popular LiDAR sensors, such as the HDL-64L or VLS-128, use an array of 64 or 128 rotating laser beams to obtain 3D point clouds. Yet, compared to cameras, LiDARs are quite costly. For example, a Velodyne HDL-64 sensor is officially priced at $80,000 each. The latest VLS-128 sensor has a higher resolution and is also more expensive. Therefore, the current deployment of self-driving vehicles are usually equipped with LiDARs of no more than 64 channels for consideration of the cost.

Typical Representation for Point Clouds: A point cloud is basically a set of points in a 3D coordinate system, commonly defined by x, y, z coordinates and the reflection intensity. Point clouds offer accurate depth measurement and can mitigate the occlusion problem that is very common for camera images (Sun et al., 2020b). Since the point cloud data is irregular and sparse, it is important to find a suitable point cloud representation for efficient processing. Most existing methods

can be divided into the three categories: voxels, points, and view.

voxels: The voxel representation requires discretizing the 3D space into 3D voxels and assigning the points to these voxels. (Zhou and Tuzel, 2018; Yang et al., 2018; Beltrn et al., 2018; Yang et al., 2018a; Lang et al., 2019; Shi et al., 2020b). For the representation, we can utilize the 3D convolution to extract features. However, this representation has some disadvantages: 1) it suffers from information loss from quantization. Point clouds within each voxel need to be subsampled. We can understand this more clearly from Fig.8 (c) that after 3D voxelization (shown by the yellow lines), the point cloud data becomes more sparse than that in (a). 2) During voxelization, a huge number of empty voxels will be produced as the LiDAR points are only on the surface of the objects. This problem is more prominent in outdoor scenes. 3) 3D convolution for voxels is often not efficient and practical in large outdoor scenes for it's expensive computational cost. In practice, 3D voxels often sacrifice their information capacity and are projected onto a 2D plane to meet limited computing resources (Lang et al., 2019).

points: Thanks to 3D point cloud processing networks, the raw 3D point cloud can be directly used as input to obtain the suitable point cloud features (Charles et al., 2017; Qi et al., 2017). In Fig.8, the yellow dots in (d) indicate the features belonging to cars. Since point-based methods directly take raw point cloud as input, they retain more information than voxel-based methods (Yang et al., 2020; Shi et al., 2020a; Qi et al., 2019). However, point-based methods are generally computationally expensive, especially when dealing with large scenes. e.g., for a widely used Velodyne LiDAR HDL-64E, it collects more than 100K points in one scan. Therefore, down-sampling point clouds is a necessary and useful method, which can reduce the computation cost but limits the performance and efficiency.

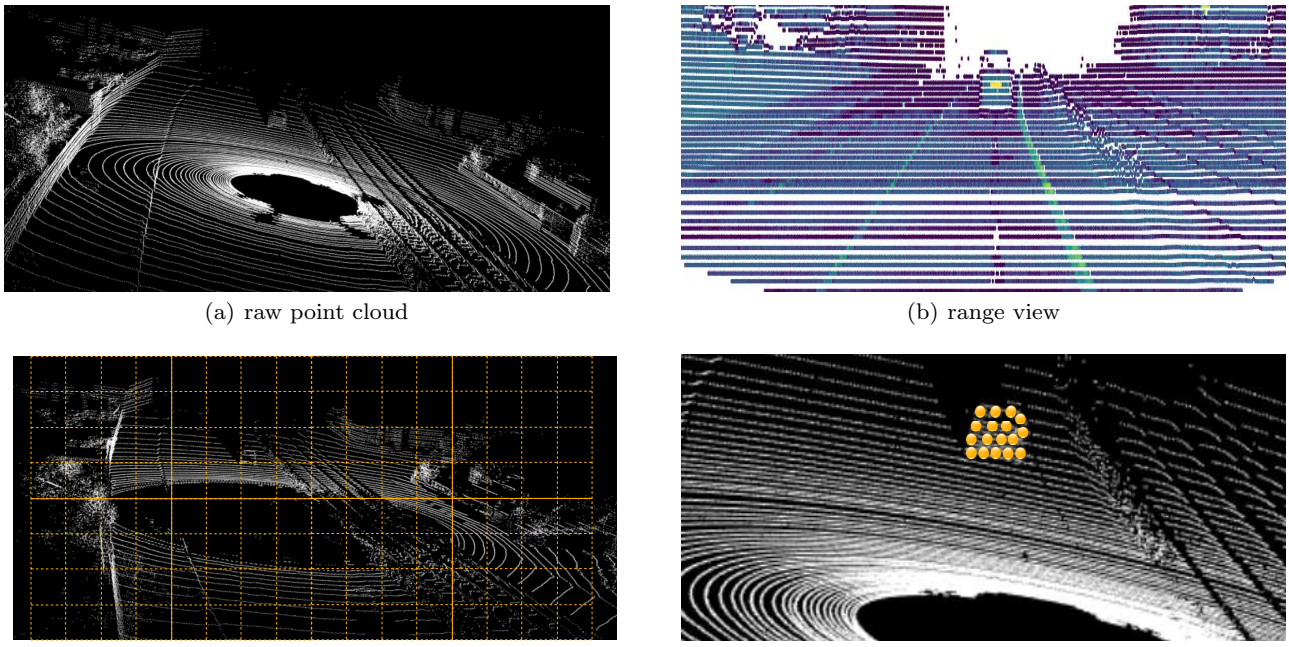
view: Another natural representation is to convert the point cloud data into some kind of 2D view, which can be processed efficiently by 2D CNNs. BEV is commonly adopted in autonomous driving for the reason that there's no overlap between objects in BEV. When encoding scenes into 2D space, they have the advantages of scale invariance and computational efficiency with 2D CNNs, but they involve information loss due to projection. Another popular view in 3D object detection models is Range view (RV), aka range image, which is a native representation of the rotating LIDAR sensor (Wang and Zhou, 2019). The dense and compact nature of RV makes it efficient for processing with powerful 2D CNNs (Liang et al., 2020; Milioto et al., 2019), but suffers from the problem of scale variation for objects.

LiDAR-Based 3D Object Detection: Most existing studies have explored the following three representations of the LiDAR data for 3D object detection:

Voxel-based Detection: Converting point cloud data into voxels is the most common treatment for the irregular organization of point clouds. Li (2017) uses a binary 3D voxel representation and feeds it to the 3D CNNs to extract features. Due to a huge number of voxels in 3D space, the whole process of detection and positioning is extremely time-consuming. In addition, traditional 3D CNNs can not learn local features of different scales well. VoxelNet (Zhou and Tuzel, 2018) extracts discriminative voxel features to speed up the model execution. SECOND (Yan et al., 2018) further overcomes the computational barrier of dense 3D convolution execution by applying sparse convolution operations. Voxel-based methods are currently widely used in 3D object detection. However, the performance upper bound of the voxel-based method is limited by the quantization error from voxelization.

Point-based Detection: We can use more recent point cloud processing networks such as PointNet, PointNet++, and other point cloud backbones (Charles et al., 2017; Qi et al., 2017; Li et al., 2018; Jiang et al., 2018; Riegler et al., 2017), to obtain spatial geometry features from point clouds. Once we get the point-based data representation, we can subsequently classify and localize objects of interest based on the extracted features. For example, Shi et al. (2019) employs PointNet++ (Qi et al., 2017) as point clouds encoder, then generates 3D proposals based on the extracted semantic and geometric features, the proposed 3D bounding boxes are refined during the second stage. PV-RCNN (Shi et al., 2020a) performs sparse convolution and Voxel Set Abstraction (VSA) on voxels to learning a power representation, which demonstrates a significant performance improvement. Specifically, a widely used Velodyne LiDAR HDL-64E collects more than 100K points in one scan. For point-based methods, this is a considerable computational challenge. To reduce the computational overhead, most of the point-based methods conduct down-sampling for raw points or feature maps when dealing with the point clouds.

View-based Detection: Many LiDAR-based methods project the LiDAR point clouds into the BEV,



(a) raw point cloud (b) range view

(c) point cloud voxels (d) point cloud points

Fig. 8 Raw point cloud and its typical data representations. We get the raw point clouds from KITTI training set

or RV, to leverage the off-the-shelf 2D CNNs. Yang et al. (2018a), which is a efficient, proposal-free single-stage detector, represents the scene in the BEV and performs 2D convolution on it. To avoid the computational bottleneck of 3D CNNs, PointPillars (Lang et al., 2019) collapses the points into vertical columns (pillars), which can convert the problem of point clouds object detection from 3D to 2D, and greatly improve the computation speed. Compact and dense RV-based methods are proposed for 3D object detection without information loss. For example, Liang et al. (2020) utilizes the dilated residual blocks to better obtain a more flexible receptive field on range images and proposes a two-stage RCNN for better 3D object detection performance.

2.3 Other sensors

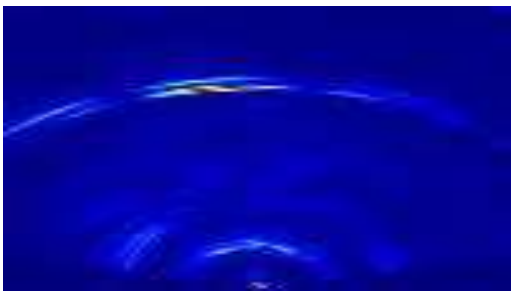
In addition to cameras and LiDARs, AVs are often equipped with other sensors such as mmWave radars, infrared cameras, etc. In particular, mmWave radar has long been used on self-driving cars due to its cost and robustness to severe weather condition, especially for adaptive cruise control and collision avoidance. We give a brief background of mmWave radars below:

Mmwave Radar Sensor: MmWave radars are active sensors that operate in the millimeter wave bands to sense the environment and measure the reflected waves to determine the location and velocity of objects (Ahmad et al., 2020). The mmWave radar has the advantages of small form factor, light weight, and high spatial resolution. It has a strong ability to penetrate fog, smoke, and dust when compared with optical sensors such as infrared sensors, LiDAR sensors, and cameras. MmWave radar has all-weather all-day characteristics. In addition, the anti-interference ability of mmWave radar is better than many other sensors.

Data Representation: The radar outputs can be organized at three levels: raw data in the form of time-frequency spectrograms, clusters from applying clustering algorithms (Kellner et al., 2012) on raw data, and tracks from performing object tracking on the clusters. However, from one representation to the next, the data sparsity and abstraction increases. We use the original collected radar data for visualization. As shown in Fig. 9, we perform two fast Fourier transforms on the raw data and then get the range-azimuth heatmap corresponding to the image. The brightness in (b) represents the strength of the signal at that location and also indicates a high probability of objects.

3D object detection through Mmwave Radars: The mmWave radar has been widely exploited in perception systems (Marchand and Chaumette, 1999). Automotive radars usually report the detected objects as 2D points in BEV and provide the azimuth angle and radial distance to the object. For each detection, the radar also reports the instantaneous velocity of the ob-

(a) RGB image



(b) mmWave radar: range-azimuth heatmap

Fig. 9 A RGB image (a) and a Mmwave Radar heatmap (b) on the same scene. The data was collected at the North Gate of the West Campus of the University of Science and Technology of China

ject in the radial direction. To the best of our knowledge, Major et al. (2019) proposes the first and currently only one radar-based deep neural network object detection with reliable results. To sum up, the radar-based 3D detectors have the following two problems. Since some radar sensors generate 3D point clouds like LiDARs, one may be enticed to apply networks similar to those used in LiDAR data processing. However, this straightforward thinking stems from the fact that the radar data has completely different properties from the point cloud. Specifically, radar data is much noisier and less accurate than LiDAR point clouds, which brings difficulties in adapting LiDAR models to radar. Another bottleneck in radar-based detectors are the lack of publicly usable data annotated with ground-truth information (Sheeny et al., 2021). In practice, mmWave radar data is more often used for fusion with other sensors. (Nabati and Qi, 2021; Yang et al., 2020).

2.4 Discussion

Cameras are highly available with low cost, but difficult to apply to complex outdoor 3D scenes. In the mean time, though LiDARs are more expensive, they have great advantages in acquiring a 3D environment information. With the help of point cloud data, we can complete the classification and identification of the surrounding more accurately.

In general, image-based methods generate less accurate 3D bounding boxes than point clouds based methods. Though LiDAR-based methods currently lead in popularity in 3D object detection, point clouds do not provide texture information for efficient class discrimination. Additionally, the density of point clouds tends to decrease quickly as the distance increases, while images can still detect faraway vehicles and objects. To improve the overall performance, an increasing number of methods try to fuse data from multiple sensors with different strategies. These methods have achieved superior performance in 3D object detection tasks compared to methods relying on single type of sensors.

3 Datasets and Metrics

Datasets are key to effectively conducting deep learning research. The availability of large-scale image datasets such as ImageNet allowed fast development and evolution of image classification and object detection models (Lin et al., 2014; Deng et al., 2009; Krizhevsky and Hinton, 2009). The same trend is also true for autonomous driving perception. In particular, tasks such as object detection require finely labelled data. In this part, we discuss some widely used datasets for 3D object detection in autonomous driving.

3.1 KITTI

One of the earliest datasets for the autonomous driving scene, KITTI (Geiger et al., 2012), provides stereo color images, LiDAR point clouds and GPS coordinates. The dataset supports multiple tasks: stereo matching, visual odometry, 3D tracking, and 3D object detection, etc. It collects data with a car equipped with a 64-channel LiDAR, 4 cameras and a combined GPS/IMU system. There are over 20 scenes in the dataset collected in cities, residential and roads. In particular, the object detection dataset contains 7,481 training and 7,518 testing frames with sensor calibration information and annotated 2D and 3D boxes around objects of interest. KITTI annotates the following classes: “Car”, “Van”, “Truck”, “Pedestrian”, “Person (sitting)”, “Cyclist”, “Tram” and “Misc”. Each annotation is categorized as “easy”, “moderate” and “hard” cases. More remarkably, in order to facilitate the development of multi-modal detection methods in autonomous driving, the KITTI development team proposes a dataset KITTI360 (Xie et al., 2016) with richer sensor information and 360° annotations. Specifically, they annotate

Table 2 Popular multi-modal dataset comparison, including year, number of LiDARs, number of LiDAR channels (we report number of channels of the top LiDAR for Waymo dataset and the maximum number of channels among 4 LiDARs for AIODrive dataset. ApolloScape's LiDARs scan with 1 beam at a high frequency to get dense point clouds), number of cameras, whether with radar, number of 2D boxes (we don't distinguish between 2D boxes and 2D instance segmentation annotation), number of 3D boxes, number of annotated classes, and location (KA: Karlsruhe; SF: San Francisco; SG: Singapore; PT: Pittsburgh)

dataset	year	n-LiDAR	n-chn	n-Cam	radar	n-2D	n-3D	n-cls	loc
KITTI	2012	1	64	4	No	80K	80K	8	KA
ApolloScape	2018	2	1	6	No	2.5M	70K	35	4x China
H3D	2019	1	64	3	No	-	1.1M	8	SF
nuScenes	2019	1	32	6	Yes	-	1.4M	23	Boston, SG
Argoverse	2019	2	32	9	No	-	993K	15	PT, Miami
Waymo	2019	5	64	5	No	9.9M	12M	4	3x USA
AIODrive	2021	4	1280	10	Yes	26M	26M	23	synthetic

3D scene elements with rough bounding primitives and then transfer this information into the image domain. As such, KITTI360 has dense semantic and instance annotations for both 3D point clouds and 2D images. In the task of object detection, KITTI requires detection of "car", "pedestrian" and "cyclist", and calculates mAP of each class. mAP is a metrics commonly used in the task of object detection. It considers a predicted box as correct if its overlap with ground-truth box is bigger than the threshold. Then the number of true positive (TP), false positive (FP), false negative (FN) is calculated. we define *precision* and *recall* as:

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

Based on the predicted and ground-truth boxes, we get a function of $p(r)$ with respect to recall r, calculation of Average Precision (AP) is as below:

$$AP = \int_0^1 p(r)dr \tag{3}$$

We calculate AP for a single class. Some datasets, which contain multiple classes, usually average the AP score of each class, denoted as mAP (mean Average Precision).

3.2 NuScenes

Developed by Motional, the nuScenes dataset is one of the largest dataset with ground-truth labels for autonomous driving (Caesar et al., 2020), which consists of 700 scenes for training, 150 scenes for validation, and 150 scenes for testing. The dataset is collected using six cameras and a 32-beam LiDAR, providing 3D annotations for 23 classes in 360 degree field of view. NuScenes is also the first AV dataset that provides radar data for perception. It is equipped with 5 radar sensors for measurement of the object velocity.

In the task of object detection, nuScenes merges similar classes and records detailed information as *attribute*. There are also some rare classes with few samples are removed. Finally, we get 10 objects for object detection. The full dataset includes approximately 1.4M camera images, 390k LiDAR sweeps, 1.4M radar sweeps and 1.4M object bounding boxes in 40k key frames. 1000 driving scenes are collected in Boston and Singapore, these two cities are known for their dense traffic and highly challenging driving situations. Additionally, nuScenes annotates object-level attributes such as visibility, activity, pose, etc.

As far as the object detection task is concerned, nuScenes allows to take historical sensor data as input. At most 6 past camera images, 6 past radar sweeps and 10 past LiDAR sweeps are provided for perception. NuScenes also annotates the velocity of objects and supports detection of object speed. The nuScenes challenge requires the detection of 10 classes, including *traffic cone*, *bicycle*, *pedestrian*, *car*, *bus*, and so on. When calculating AP for a class, instead of adopting the traditional bounding box overlap, nuScenes uses a center-distance-based metrics. When matching the prediction and ground-truth, nuScenes computes their center distance and obtain Average Precision (AP) based on a list of threshold values: 0.5, 1, 2, 4, in meters, and finally averages over the four AP as the mAP value of this class. The final metrics is the average of all the 10 classes.

Unlike KITTI, nuScenes also considers True Positive (TP)'s average translation, scale, orientation, velocity and *attribute* error with ground-truth, marked as ATE, ASE, AOE, AVE, and AAE, respectively. The final metrics, nuScenes detection score (NDS), is derived from a

weighted sum of mAP and errors, leading to a more comprehensive description of detection performance.

3.3 Waymo Open Dataset

The Waymo Open Dataset is a high-quality annotated multi-modality dataset for autonomous driving (Sun et al., 2020a). It consists of annotated data collected by Waymo's self-driving vehicles. The dataset covers a wide variety of scenes from dense urban centers to suburban landscapes. There are totally 798 scenes for training and 202 scenes for validation, which is collected by five LiDAR sensors and five pinhole cameras and annotated with 2D and 3D labels. Each scene captures 20 seconds of continuous driving. The annotations include four object categories, including: "car", "pedestrian", "cyclist" and "sign". Same as KITTI dataset, Waymo Open Dataset uses AP as metrics. In order to describe the heading accuracy of predicted boxes, Waymo Open Dataset proposes a new metrics - APH, formula of APH is as below:

$$\text{APH} = \int_0^1 h(r)dr, \tag{4}$$

where $h(r)$ is computed similar as $p(r)$ defined in Eq. (3), but each predicted bounding box is weighted by its heading accuracy.

Waymo Open Dataset also supports the challenge of domain adaptation. Domain adaptation is a popular technology that learns knowledge from one domain and then transfer to another domain, which addresses the problem of expensive annotation. In this challenge, people should perform object detection on the dataset collected from a new location with limited annotations.

3.4 Other Datasets

In addition to the above well established datasets, there are a few recent datasets that are gaining rapid popularity:

- ApolloScape (Huang et al., 2019) consists of data from 4 regions in China in various weather conditions. Data is collected with a SUV equipped with 2 LiDAR sensors, 6 video cameras and a combined IMU/GNSS system. It supports a variety of autonomous driving tasks such as scene parsing, lane segmentation, trajectory prediction, object detection, tracking, etc. The dataset contains 140K+ annotated images with annotation of lanes. For 3D object detection, it annotates 3D bounding boxes of objects in 6K+ point clouds. ApolloScape's evaluation metrics is the same as KITTI. It requires detection of vehicles, pedestrians and bicyclists. It first calculates mAP of each class and then averaging over the 3 classes as the final detection result.
- H3D (Patil et al., 2019) focuses on crowded traffic scenes in urban. The dataset collects data with 3 cameras with 260 degree field of view (FoV) in total, and a 64-line velodyne LiDAR sensor. It contains over 27K frames in 160 scenes with over 1 million objects. Each frame is annotated with 360 degree. For evaluation, it uses a similar protocol as KITTI with 0.5 IoU threshold for Car and 0.25 IoU threshold for Pedestrian. It requires detection of 360 degree FOV.
- Argoverse (Chang et al., 2019) is a dataset supporting perception and motion prediction task. It provides rich semantic annotation for maps. For sensor setup, it collects data with two 32-channel LiDARs, seven surround-view cameras and two stereo cameras. It provides rich semantic information about road infrastructure and traffic rules. It also provides HD maps for automatic map creation aka. *map automation*.
- AIODrive (Weng et al., 2020) is a large scale synthetic dataset generated by the urban driving simulator, CARLA (Dosovitskiy et al., 2017). It synthesizes data from multiple common sensors, including three high-density LiDARs, one Velodyne-64 LiDAR, five high-resolution RGB cameras, five high-resolution depth cameras, four radar, and one IMU and GPS system. All sensors collect data at a frequency of 10Hz. With the help of the simulator, it provides petty detailed annotation, including object's 2D/3D bounding boxes, trajectories, velocity, and acceleration. It also synthesizes some adverse scenes such as terrible weather condition or car accidents.

3.5 Discussion

Datasets for autonomous driving are developing very rapidly. Most of the multi-modal fusion methods we have reviewed conduct experiments on multiple datasets shown above. From Fig. 10, we observe that the size of the three popular datasets ranges from only 15,000 frames to over 230,000 frames. Compared to the image datasets in 2D computer vision, 3D datasets here are still relatively small, and the object classes are limited and unbalanced. Fig. 10 compares the percentage of car, person, and cyclist classes. There are many more objects labeled as "car" than "pedestrian" or "cyclist".

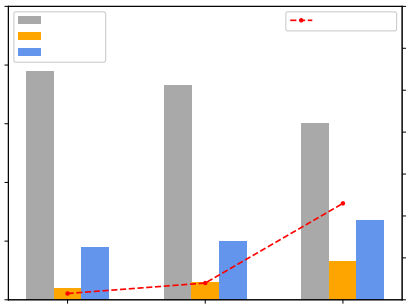


Fig. 10 Comparison of KITTI, nuScenes, and Waymo Open Dataset. We focus on the percentage of objects that belong to "car", "person", and "cyclist" classes as well as the number of frames

We make a comprehensive comparison for the discussed datasets in Tab. 2.

On the other hand, high-quality data annotation is an expensive task. A popular topic at present is reducing the dependency of ground-truth labels with the technology of transfer learning, which will be discussed in Sec. 5.

4 Deep Learning Based Multi-Modal 3D Detection Networks

In this section, we present our review of multi-modal fusion based 3D detection networks. We organize our review by considering the following three important factors in the fusion strategy: (1) *fusion location*, i.e., where the multi-modal fusion takes place in the network pipeline, (2) *fusion input*, i.e., what data representation for each sensor is used for fusion, and (3) *fusion granularity*, i.e., at what granularity the data from multiple sensors are fused for detection.

Among these three factors, fusion location is the most important factor to differentiate the methods. Generally speaking, we have two possible fusion locations as illustrated in Fig. 11: feature fusion and decision fusion. Feature fusion uses the combined features from different modalities to obtain detection results, while decision fusion attempts to combine each individual detection result. Below, we first review the feature-fusion methods, and then discuss the decision-fusion methods. Note that since the design of fusion methods is orthogonal with the choice of datasets, fusion methods from the KITTI, Waymo, and nuScenes datasets are discussed together. Among them, most multi-modal 3D detection methods are evaluated on KITTI. We can clearly identify which datasets the method applies to from the evaluation summary in Table 3.

4.1 Feature Fusion

Feature fusion mixes the modalities hierarchically in neural network layers. It allows the features from different modalities to interact with each other over layers. Feature fusion methods require interactions at feature layers, as shown in Fig. 11. These methods first adopt feature extractors for each individual modality respectively, then these features are combined to achieve multi-scale information fusion. Finally, we can obtain the detection result by feeding the fused features to a series of neural network layers.

Many fused methods fall in this category. In the rest of this subsection, we mainly focus on the fusion between cameras and LiDARs. We first look at possible combinations of fusion input representations and then look at different fusion granularity.

4.1.1 Fusion Input Representation

Here, we focus our discussion on the methods that combine LiDAR point clouds and camera images. Given the discussion in Section 2, LiDAR point clouds have three popular representations: raw points, voxel-based, point-based and view-based. Camera images have three popular representations: feature map, mask and pseudo LiDAR.

In our survey, we find that most of the existing methods employ the following combinations: point cloud view with image feature map, point cloud voxels with image feature map, LiDAR points with image feature map, LiDAR points with image mask, as well as point cloud voxels with image mask. In addition, an increasing number of fusion methods employ more than one representation for a modality, e.g. point cloud voxels and view, with image feature map, point cloud voxels with image feature map and pseudo-LiDAR representation. Below we first discuss methods that adopt dual-representation combinations in the approximate order of publication time, then discuss methods that adopt triple-representation combinations. In our discussion, we pay extra attention to the information loss in the fusion process.

Point cloud view & image feature map: Methods in this category project 3D LiDAR point clouds to the BEV, FV, or RV, and obtain image feature maps by a CNN backbone network.

MV3D (Chen et al., 2017) is a pioneering work in this category, which combines LiDAR views with image feature maps. It takes the point cloud's FV and BEV as input, and exploits a 3D Region Proposal Network (RPN) to generate 3D proposals. MV3D

Table 3 Summary of multi-modal 3D detection methods: loc (fusion location), gran (fusion granularity), PCR (point cloud representation), IR (image representation), lat (latency), DS (dataset used for evaluation)

	loc	**gran**	**PCR**	**IR**	**Hardware**	**lat**	**DS**	**mAP**
MV3D (Chen et al., 2017)	Feature	ROI-wise	View	Feature map	Titan X	0.36s	KITTI	63.63%
AVOD (Ku et al., 2018)	Feature		View	Feature map	Titan XP	0.08s	KITTI	71.76%
PointFusion (Xu et al., 2018)	Feature		Point	Feature map	GTX1080	1.3s	KITTI	63.00%
F-Pointnet (Qi et al., 2018)	Feature		Point	Feature map	GTX1080	0.17s	KITTI	69.79%
F-ConvNet (Wang and Jia, 2019b)	Feature		Point	Feature map	-	0.1s	KITTI	75.50%
RoarNet (Shin et al., 2019)	Feature		Point	Feature map	Titan X	-	KITTI	73.04%
SCANet (Lu et al., 2019)	Feature		View	Feature map	GTX1080	0.09s	KITTI	66.30%
SIFRNet (Zhao et al., 2019)	Feature		Point	Feature map	-	-	KITTI	-
Confuse (Liang et al., 2018)	Feature	Voxel-wise	Voxel	Feature map	GTX1080	0.06s	KITTI	68.78%
IPOD (Yang et al., 2018b)	Feature	Point-wise	Point	mask	-	0.1s	KITTI	72.57%
PointPainting (Vora et al., 2020)	Feature		Point	Mask	GTX1080	0.4s	KITTI	71.70%
PI-RCNN (Xie et al., 2020)	Feature		Point	Feature map	-	0.06s	KITTI	71.70%
EPNet (Huang et al., 2020)	Feature		Point	Feature map	Titan XP	0.1s	KITTI	81.23%
Moca (Zhang et al., 2020a)	Feature		Voxel	Feature map	-	-	nuScenes	66.60%
HorizonLiDAR3D (Ding et al., 2020)	Feature		Voxel	Mask	-	-	Waymo	78.49%
PointAugmenting (Wang et al., 2021)	Feature		Voxel	Feature map	-	-	nuScenes	66.80%
CenterPointV2 (Yin et al., 2021)	Feature		Voxel	Mask	-	-	nuScenes	67.10%
FuseSeg (Sun et al., 2020c)	Feature	Pixel-wise	View	Feature map	-	-	KITTI	-
MMF (Liang et al., 2019)	Feature	Multiple	Voxel & View	Feature map	GTX1080	0.08s	KITTI	77.43%
MVX-Net (Sindagi et al., 2019)	Feature		Voxel	Feature map	-	-	KITTI	72.70%
3D-CVF (Yoo et al., 2020b)	Feature		Voxel	Feature map	-	0.06s	KITTI	80.45%
MVAF (Wang et al., 2020)	Feature		Voxel & View	Pseudo-LiDAR	-	0.06s	KITTI	78.71%
CLOCs (Pang et al., 2020)	Decision	-	-	-	-	0.1s	KITTI	82.25%

integrates proposals and multi-view features, including Bird's-Eye View (BEV) and front view of LiDAR, into the same dimension through a region of interest (RoI) pooling operation. The information loss stems from the projection of the point cloud onto BEV and FV, which is non-negligible in this case.

As such, SCANet (Lu et al., 2019) and AVOD (Ku et al., 2018) remove the LiDAR FV branch and only take LiDAR BEV and image as input, which effectively decreases computation cost and information loss. SCANet utilizes an encoder-decoder based proposal network with Spatial-Channel Attention (SCA) module to capture multi-scale contextual information and Extension Spatial Upsample (ESU) module to recover the spatial information. While AVOD (Huang et al., 2017) fuses two feature exacted by cropping and resizing operations from images and BEV feature maps. AVOD demonstrates that LiDAR BEVs and images together are sufficient to interpret information in the 3D space. In reality, however, due to the quantization problem, there is still certain loss of information.

Compared with BEV and FV, range images are compact and more importantly, intrinsic LiDAR representation. The researchers are trying to put forward fusion methods that combine range images with RGB image feature maps so that 2D convolution could be directly used without projection. FuseSeg (Sun et al., 2020c) uses the RV representation and the calibration between the image and the point cloud to establish the point-pixel correspondence relationship and leverages information from the two input modalities for detection.

point cloud voxels & image feature map: Methods that fall into this category convert point clouds into 3D voxels as well as extract feature maps from images. A dense voxel-wise correspondence is established between these two inputs.

Contfuse (Liang et al., 2018) transforms the image feature from FV to BEV and then performs continuous convolutions (Wang et al., 2018) to fuse the BEV image feature and point cloud voxels. The engagement of continuous convolution captures local information from neighboring observations and leads to less geometric information loss. Sindagi et al. (2019) takes voxels and image feature maps as input, projects non-empty voxel features generated by VoxelNet (Zhou and Tuzel, 2018) into the image and uses a pre-trained network to extract image features for each projected voxel. These image features are then concatenated with voxel features to produce more accurate 3D boxes. MVX-Net can effectively exploit multi-modal information to reduce false positives and negatives.

Very recently, MoCa (Zhang et al., 2020a) further boosts detection performance of MVX-Net by cutting point cloud and imagery patches of ground-

truth objects and pasting them into different scenes in a consistent manner while avoiding collision between objects. The experimental result ranks in the top of the nuScenes leaderboard[2] and also achieves a competitive performance on the KITTI 3D benchmark[3]. PointAugmenting (Wang et al., 2021), which also ranks top on the nuScenes dataset, decorates point clouds with corresponding point-wise CNN features extracted by pretrained 2D detection models, which is a typical point-wise fusion. PointAugmenting also benefits from an occlusion-aware point filtering algorithm, which consistently pastes virtual objects into images and point clouds during network training. In another work 3D-CVF from KITTI (Yoo et al., 2020b), the spatial attention maps (Vaswani et al., 2017) are applied to weigh each modality depending on their contributions to the detection task. Voxel-wise fusion with attention generates a strong joint camera-LiDAR feature and thereby reduces the information loss further (Liu et al., 2019). Still, the quantization problem of 3D voxels cannot be overlooked.

LiDAR points & image feature map: The advent of PointNet (Charles et al., 2017) makes it possible to process the raw point cloud directly without any projection or conversion. Consequently, PointNet inspires series of studies to combine points directly with feature maps.

In this part, earlier approaches adopt coarse-grained RoI-level fusion. They are usually limited in accuracy by the performance of 2D detectors because of the cascading nature of the structure (Xu et al., 2018; Qi et al., 2018; Wang and Jia, 2019b; Shin et al., 2019; Zhao et al., 2019). These methods leverage 2D detectors to generate 2D region proposals to narrow down the RoIs in 3D object detectors. These approaches assume each seed region only contains one object of interest, which is however not true for crowded scenes and small objects like pedestrians. As a result, the performance of these cascade networks is limited by the 2D detector's performance.

People also try to combine points with feature maps in different ways. Xie et al. (2020) conducts continuous convolution directly on 3D points and retrieves image features from feature maps with more semantic information. PI-RCNN takes the first step towards the point-wise fusion. Afterwards, Huang et al. (2020) proposes LI fusion layer that uses point cloud features to estimate the importance of corresponding image features, which reduces the influence of occlusion and depth uncertainty. However, these methods suffer from the problem of edge information ambiguity, considering that the feature corresponding to each point is obtained through interpolation. Therefore, we believe that a one-to-one correspondence can be achieved by replacing the 2D feature extractor with 2D semantic segmentation.

LiDAR points & image mask: IPOD (Yang et al., 2018b) performs semantic segmentation on images, and the output masks are projected into raw 3D point clouds to distinguish foreground and background points, then the raw point clouds and foreground point clouds are classified and regressed by PointNet++. Although the raw point cloud data can be directly used, it imposes an upper bound on recall due to the cascade network. PointPainting (Vora et al., 2020) is another typical example that belongs to this category (Note that here we choose the point-based detector PointRCNN for LiDAR branch, which refers to Painted PointRCNN). It projects the points onto the image, appends segmentation scores to the raw LiDAR point, and paints the points. PointPainting has achieved a perfect point-wise fusion by the above painting operation and has proved an effective fusion approach. More importantly, PointPainting can be applied to both point-based and voxel-based LiDAR backbones and further improves the accuracy. On the other hand, an inevitable problem is that the large 3D point clouds may find no corresponding pixel from the image because some point clouds are in the occluded part of the image.

point cloud voxels & image mask: Inspired by the successful PointPainting, CenterPointV2 (Yin et al., 2021) gets the state-of-the-art result of nuScenes, and HorizonLiDAR3D (Ding et al., 2020) ranks top on Waymo Open Dataset Challenge on the 3D detection track[4]. For the former, it uses the CenterPoint (Yin et al., 2021) detection framework. In this framework, point clouds are processed as voxels. Based on that, Yin et al. (2021) uses point-wise fusion strategy to annotate each LiDAR point with image-based instance segmentation results generated by a Cascade RCNN model trained on images. They also perform flip and rotation augmentations and use an ensemble of five models for final submission. HorizonLiDAR3D (Ding et al., 2020) improves AFDet (Ge et al., 2020) as a strong baseline in this winning solution and combines all point

[2] https://www.nuscenes.org/object-detection?externalData=all&mapData=all&modalities=Any

[3] http://www.cvlibs.net/datasets/kitti/eval_object.php

[4] https://waymo.com/open

clouds generated by all five LiDARs to fully utilize the information. Specifically, HorizonLiDAR3D designs stronger voxel-based networks and enhances the point cloud data using densification and point painting. To leverage camera information, it paints additional attributes to each point by projecting them to camera space and gathering image-based perception information.

point cloud voxels & point cloud view & image feature map: Different data representations have their own advantages, e.g., different views of point clouds have some complementary nature (Zhou et al., 2020a). We can therefore take advantage of more than one representation in a single modality. This motivates researchers to design more effective fusion frameworks to take multiple representations for each modality. Wang et al. (2020) proposes a single-stage multi-view fusion framework that takes point cloud voxels, RV, and image feature maps as input. They further estimate the importance of the three sources with attention mechanisms to achieve adaptive fusion.

point cloud voxels & image feature map & image pseudo-LiDAR: As the first method to exploit multiple related tasks for accurate multi-modal 3D object detection, MMF (Liang et al., 2019) presents an end-to-end learn-able architecture that reasons about 2D and 3D object detection as well as ground estimation and depth completion. Specifically, it uses the voxel-based LiDAR backbone, the feature map for the camera stream, and the pseudo-LiDAR obtained by depth completion. Although experiments show that all these tasks are complementary and help the network learn better representations by fusing information at various levels, there is no denying that the whole network pipeline is complex and time-consuming.

Discussion: Point cloud representations evolve from voxels, to points, and then to range images; RGB image representations evolve from feature maps to semantic segmentation results. Following this order, naturally, we have several different fusion input combinations as discussed above. Meanwhile, as the representations of the point clouds and images become increasingly diverse, it is also common to adopt multiple representations for a single modality.

4.1.2 Fusion Granularity

In this section, we discuss the fusion granularity used in the existing fusion methods. Specifically, feature fusion can be performed at different granularity, i.e., RoI-, voxel-, point-, and pixel-wise. Below we discuss these granularity in detail.

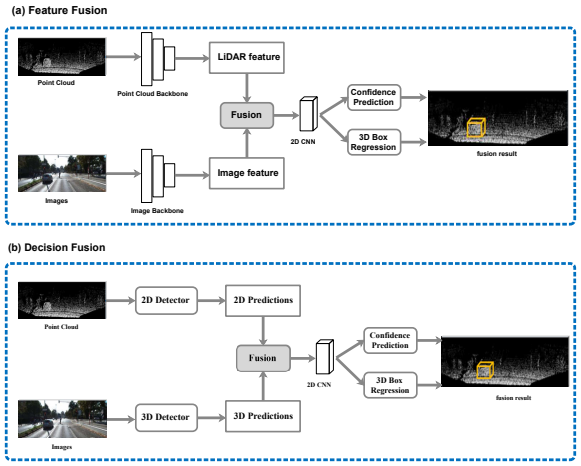


Fig. 11 Feature fusion vs decision fusion

RoI-wise: In this category, the fusion of data happens at the ROI level via operations like ROI pooling (Chen et al., 2017; Ku et al., 2018). Another common operation is to get 3D frustums from 2D RoIs by geometrical relationship and utilize 3D detector to process these frustums (Wang and Jia, 2019a; Xu et al., 2018). RoI-wise fusion appears mostly in the years of 2017 and 2018 when the multi-modal methods were just emerging. RoI-wise feature fusion is often applied before further proposal refinement (Liang et al., 2019; Yoo et al., 2020b), which is illustrated in Fig. 12. In some cases, the fusion granularity is too coarse, therefore unsuitable for fine object detection tasks. For example, the rectangular RoIs on images involve lots of background noise, and fusing these noise information will lead to unsatisfactory detection results.

Voxel-wise: In this category, voxelized point cloud data is usually projected onto the image plane, followed by feature extraction within 2D ROIs and concatenation of pooled image features at the voxel level. Voxel-wise fusion leads to the "feature blurring' problem. This "feature blurring' happens when one point in the point cloud is associated with multiple pixels in the image or the other way around, which confounds the data fusion. Therefore, interpolation is often needed to align the point cloud BEV and RGB feature maps (Liang et al., 2018; Yoo et al., 2020b). In contrast to RoI-wise fusion, this voxel-level granularity is finer and more precise. Besides, it can be easily extended to aggregate image information to empty voxels where LiDAR points are not sampled due to reasons such as low LiDAR

Table 4 Advantages and disadvantages for feature fusion and decision fusion

Categories	Advantages	Disadvantages
Feature Fusion	+ can better leverage rich intermediate features from multiple modalities + can achieve joint optimization of multiple modalities	- sensitive to inherent spatial-temporal data misalignment among sensors - high-dimension features from multiple modalities could be noisy for the detection network
Decision Fusion	+ can better leverage existing networks for each modality + easy to know whether the results from each modality is right + much simpler to build	- the performance is limited by the a single modality - less flexible than decision fusion methods

resolution or faraway objects. As such, the system provides dense information when high-resolution LiDAR points are unavailable.

Point-wise: Point-wise fusion involves projection of 3D points onto the image using a known calibration matrix (Strecha et al., 2008), followed by feature extraction from a pre-trained 2D CNN and concatenation of image or mask features at point level. Compared with the above two fusion granularity levels, it does not suffer from feature blurring. Besides, by fusing high-level image semantics to points, we can simply solve the resolution mismatch problem between dense images and sparse point clouds (Vora et al., 2020; Xie et al., 2020). Although experimental results show that point-wise strategy can effectively improve performance (Vora et al., 2020; Yin et al., 2021), there are still limitations. Firstly, images and point clouds are highly coupled and therefore may reduce overall reliability. Secondly, point-wise fusion is less efficient in terms of memory consumption compared to the voxel-wise fusion.

Pixel-wise: The range image is a native representation of the rotating LIDAR sensor in 2D space (Moosmann and Stiller, 2011). It retains all original information without any loss. Some recent fusion methods combine range images and RGB images and perform the pixel-wise fusion in 2D plane. In this way, features can be conveniently extracted through 2D CNNs. Then we can establish feature alignment between pixels. Please note that although range images are denser than point clouds, their resolution is still lower than RGB images.

Discussion: Fig. 12 shows the years in which the typical multi-modal 3D detection methods appeared. Each method's fusion granularity is also marked. We observe that the granularity was relatively coarse at first, mainly using RoI-wise and voxel-wise granularity. With the rapid development of multi-modal object detection, fusion granularity becomes finer, leading to the improvement of detection performance. As a result, it is urged to achieve real point-wise fusion directly on LiDAR points instead of performing projection or voxelization.

4.2 Decision Fusion

In decision fusion, multiple modalities are processed separately and independently up to the last stage, where fusion occurs (Fayyad et al., 2020). Generally, the idea of this kind of methods is to use neural networks to process the sensor data in parallel and then fuse all the obtained decision output to get the final result. Compared to feature fusion, decision fusion can better leverage existing networks for each modality, and we can easy to know whether the results from each modality is right.

Still, as can be seen from Table 4, a serious shortcoming that cannot be ignored is the inability to use rich intermediate but may beneficial features (Asvadi et al., 2017; Schlosser et al., 2016). Therefore, decision fusion has received little attention until very recently. Pang et al. (2020) exploits the geometric and semantic consistencies between 2D and 3D detection results and automatically learns probabilistic dependencies from training data. Specifically, it obtains 2D and 3D proposals by the detectors and then encodes all proposals into a sparse tensor. Finally, for the non-empty elements, it uses 2D CNNs to get predicted fusion scores. For experimental results, CLOCs outperforms single modality detectors and ranks the highest among all the fusion-based methods on the official KITTI leaderboard.

Compared to feature fusion methods, decision fusion has simpler network pipelines. They do not need to deal with issues such as alignment accuracy and perspective difference between pixels and point clouds. We believe that decision fusion has great potential for future research and development.

Table 4 summarizes the advantages and disadvantages of feature fusion and decision fusion. Since current researches focus on feature-level fusion, this paper will also focus on introducing the feature fusion methods.

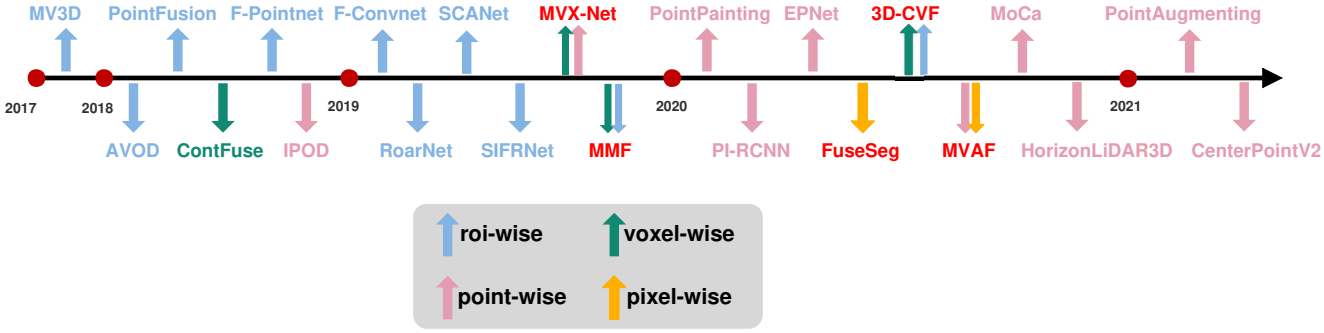


Fig. 12 Timeline of the feature fusion 3D object detection methods. We use different color to mark their fusion granularity

But we do hope that more decision-level fusion methods will emerge because of their typical advantages.

4.3 Summary of Camera-LiDAR Fusion Methods

To summarise, the majority of fusion methods are based on the KITTI 3D benchmark, but methods that rank top on the KITTI 3D object detection leaderboard are mainly LiDAR-only methods (Vora et al., 2020). On the KITTI dataset, the multi-modal approaches seem to attain only marginal improvement. On the contrary, top-ranked methods on the latest datasets such as nuScenes and Waymo Open Dataset are predominantly fusion based. One possible reason is that the LiDAR sensors used in these datasets have different resolution. KITTI uses a spinning LiDAR sensor of 64 channels, nuScenes uses a rotating 32-beam LiDAR. As a result, we can conclude that multi-modal methods are much more useful when point clouds are relatively sparse. More importantly, the recent fusion methods share some common characteristics. On one hand, they all apply the point-wise fusion granularity to effectively establish the precise mapping between LiDAR points and image pixels. On the other hand, during training the fusion network, they perform carefully designed cross-modality data alignment, which does not only accelerate network convergence but also alleviates the class imbalance problem.

To the best of our knowledge, low-resolution LiDARs are chosen by up-to-date datasets to lower the deployment cost. Therefore, we believe that the focus of the future research could be on constructing better fusion methods between lower-precision LiDARs, dense images and possibly other sensors, to achieve a balance between cost and accuracy.

4.4 Fusion with Other Sensors

So far, we have focused on LiDAR-camera fusion methods. Next, we briefly discuss fusion methods that target at other types of sensors, e.g., *Radar-Camera*, *LiDAR-Radar*, etc. Chadwick et al. (2019) projects radar detection results to the image plane to boost the object detection accuracy for distant objects. Similarly, Nabati and Qi (2019) uses radar detection results to first generate 3D object proposals, then projects them to the image plane to perform joint 2D object detection and depth estimation. Remarkably, CenterFusion (Nabati and Qi, 2021) proposes to exploit both radar and camera data for 3D object detection. It first uses a center point detection network to detect objects by identifying their center points on the image and then solves the key data association problem using a novel frustum-based method to associate the radar detection results to the corresponding objects center point. RadarNet (Yang et al., 2020) fuses radar and LiDAR data for 3D object detection. It uses a feature fusion approach to learn joint representations from the two sensors and a decision fusion mechanism to exploit the radars radial velocity evidence.

Finally, we would like to point out that it is also useful to fuse multiple sensors of the same kind. HorizonLiDAR3D (Ding et al., 2020) combines all point clouds generated by five LiDAR sensors to enrich the information of the point cloud data.

To the best of our knowledge, current multi-LiDAR fusion methods simply concatenate point clouds, with no special handling. We look forward to more and more fusion technologies that fuse multiple sensors of the same kind emerge.

5 Open Challenges and Possible Solutions

In this section we discuss the open challenges and possible solutions for multi-modal 3D object detection. We focus our discussion on how to improve the accuracy and robustness of the multi-sensor perception systems while achieving real-time performance. Table 6 summarizes our discussions.

Table 5 Summary of all the multi-modal 3D detection networks discussed in Section 4

Methods	Contributions	Limitations
MV3D	Use BEV and FV LiDAR projections and monocular camera frames to detect vehicles. Introduce a deep fusion architecture to allow interaction between different modalities.	Insufficient accuracy for small target objects.
AVOD	Only use BEV for LiDARs and monocular images for cameras. Upsample the feature map using a feature pyramid network (FPN) improving the detection of small targets.	only sensitive to objects in front of the vehicle.
PointFusion	Process the image and the original point cloud independently by the CNN and PointNet. Fuse the global feature and the single point feature together to predict the bounding box.	Fusion operation too simple.
F-Pointnet	Generate a 2D proposal from the image then extract the point cloud from the corresponding frustum. Finally perform a 3D bounding box estimation network.	Greatly limit the accuracy due to the cascade structure.
F-Convnet	Generate a sequence of frustums for each region proposal and use the obtained frustums to group local points. Can effectively prevent the failure of cascade methods when the 2D detector fails.	Not completely avoid the problems of cascading methods.
RoarNet	Estimate 3D poses from a monocular input image and derive multiple geometrically feasible candidates. Adopt a two-stage object detection framework to further refine search space from 3D point clouds.	Still a cascaded network.
SCANet	Propose a new Spatial-Channel Attention module which is capable of encoding multi-scale and global context information. Design an Extension Spatial Upsample module which uses multi-scale low-level features to guide high-level.	Only focus on car class and maybe ineffective for small objects.
SIFRNet	Focus on utilizing the front view images and frustum point clouds to generate 3D detection results. Can still obtain very satisfied results even when the point clouds are extremely sparse.	Only provide the KITTI validation set result. Also suffer from the cascade network design.
Confuse	Propose a fusion method based on KNN, bilinear interpolation and MLP. The first article that uses RGB feature maps to integrate into BEV features.	May fail to detect correctly When the point cloud is sparse.
IPOD	Seed each point with proposals, without loss of precious localization information from point cloud data. Yield much higher recall compared with voxel and projection-based methods.	Time consuming for semantic segmentation network.
PointPainting	Project LiDAR points into the output of a segmentation network and appending the scores to each point.	Same as the last one.
PI-RCNN	Directly apply continuous convolution on raw points based on Contfuse. Fuse the semantic features outputted by a segmentation model with the features of LIDAR points	Same as the last one.
EPNet	Enhance the point features with semantic image features in a point-wise manner. A consistency loss to encourage the consistency of both the localization and classification.	Observe from ablation result that the fusion operation improving little.
FuseSeg	Utilize the dense native range representation of a LiDAR sensor. Establish point correspondences between the two input modalities.	Belong to point cloud segmentation network. Can not apply to detection task directly.
MMF	Through image and LiDAR sensor fusion to learn multiple related tasks, Realize 3D and 2D target detection, ground estimation and depth completion at the same time.	Complicated and time-consuming (not end-to-end) framework.
3D-CVF	Combine the camera and LiDAR features using the cross-view spatial feature fusion strategy. Use attention map to weight the information from each modality depending on their contributions.	Not adopt the fusion strategy of point-wise.
MVAF	Estimate the importance of the three input with attention mechanisms. Achieve adaptive fusion of multi-view features in a pointwise manner.	Too many different formats of input data. Time-consuming and higher calculations.
CLOCs	Can be readily employed by any relevant already-optimized detection approaches. Automatically learn probabilistic dependencies from training data to perform fusion.	Cannot meet the real-time requirements for now.
MoCa	Cut point cloud and imagery patches of ground-truth objects and pasting them into different scenes.	Need to choose strong baseline.
HorizonLiDAR3D	Design stronger networks and enhance the point cloud data using densification and point painting. Paint additional attributes to each point by projecting them to camera space.	No new proposed fusion operation. (Inspired by PointPainting.)
PointAugmenting	Decorate point clouds with corresponding point-wise CNN features. Benefit from a novel cross-modal data augmentation algorithm.	Same as the last one. (The main improvement lies in the use of data augmentation.)
CenterPointV2	Use the CenterPoint detection framework. Use PointPainting to annotate each LiDAR point with mask.	Also refer to the PointPainting fusion operation.

Table 6 Summary for open challenges and possible Solutions

Topic	Challenge	Possible Solution
Multi-Sensor Calibration and Data Alignment	Expensive cost to calibrate	1.Assemble the LiDAR and camera as a whole 2. Use online calibration to automatically calibrate multi-sensor 3. Find balance between information and computation cost
	Quantization error during aligning multi-modal data	Use bilinear interpolation to improve the performance of multi-modal object detection network
	Decide when to fuse	1. Learn dynamic weights for the two modalities 2. Fuse multi-modal data in a multi-task fashion
Information Loss During Fusion	Fusion location	Try to fusion at feature-level or decision-level
	Fusion strategies	1. Adopt finer grained fusion operations 2. Adopt multi-fusion strategies
	Fusion architecture	1. Perform visual analysis to find the best fusion architecture 2. Use Neural Architecture Search (NAS) technology to potentially solve the problem 3. Optimize the loss function
	Fusion operation	1. Use attention mechanisms during fusion 2. Take advantage of the value of multi-task 3. Optimize operations such as continuous convolution
Multi-Modality Data Augmentation	Misalignment between multi-modality data	Cut point cloud and imagery patches of ground-truth objects and past them in a consistent manner
Dataset and Metrics	Small size, class imbalance and labeling errors	1. Adopt unsupervised and weakly-supervised learning fusion methods 2. Use synthetic datasets as auxiliary
	No metric specifically for multi-sensor fusion methods	Design a metrics focus on the extra computational overhead introduced by the fusion process, the networks robustness, and so on

5.1 Multi-Sensor Calibration and Data Alignment

5.1.1 Multi-Sensor Calibration

Multi-modal based methods require alignment of data from different sensor coordinate. For LiDARs and cameras, we need to build a map from 3D LiDAR coordinate to 2D image plane coordinate. This can be achieved by calculating a matrix based on LiDAR-camera intrinsic and extrinsic parameters, which projects 3D points to 2D pixel coordinates. Most datasets provide the projection matrix for researchers. In reality, we need to perform calibration to obtain the matrix.

Traditional calibration methods use a calibration target to derive the intrinsic and extrinsic camera parameters. Manual and cumbersome, this process inevitably leads to error. Furthermore, after the initial calibration, there still remains factors like vibration and shaking of vehicles, which can make the errors accumulate over the time. A common practice to solve this problem is to assemble the LiDAR and camera as a whole, attempting to prevent the relative displacement of LiDAR and camera to the greatest extent(Huang et al., 2019; Sun et al., 2020a).

Another solution is to develop an *online calibration* method that can continuously calibrate the LiDAR sensor and camera on the fly. Online calibration is currently an active topic of research in the community, such as the work in Pandey et al. (2012); Levinson and Thrun (2013); Schneider et al. (2017). These methods automatically calibrate among multiple sensors, without human intervention.

5.1.2 Data Alignment

Point clouds and RGB images are both from vision sensors, but hold different properties and are obtained in different perspectives. As a result, it is a challenging task to combine data from the two modalities.

Firstly, point clouds and RGB images are from different measurement space. Point clouds are a set of points indicating 3D coordinates of the objects, which is in the real world coordinate system. RGB images are matrices of pixels, each pixel's coordinate is represented as (x, y), where x, y is the pixel's row index and column index, respectively. When aligning the data streams from the two distinct measurement space, e.g., projecting 3d point clouds to the image plane based on the calibration matrix, quantization errors will occur.

To overcome the quantization error, some works (Liang et al., 2018; Xie et al., 2020) use bilinear interpolation to improve the performance of multi-modal object detection network.

A second difficulty of combining multi-modal data is to choose the right time to fuse. Point clouds and RGB images have their own strength, for example, when the light is dark, cameras can't capture a valid picture, in this case, it's important to utilize LiDAR sensors. In other cases, when objects is far from sensors, there are very few reflected points, making it hard to locate or classify an object, but RGB images, which are dense representation, perform well in this situation. As such, there is a dynamic weight for the two modalities according to different scenes. However, most existing multi-modal object detection methods take the two modality into a unified network without an adaptive weight for different scenes, making it less flexible in the constantly changing scenes.

Point clouds provide accurate real-world coordinate of objects, while RGB images contain rich feature of color, texture, etc., which have the potential to extract semantic information. A possible solution to combine the complementary information from the two modalities explicitly. For example, we can extract semantic features or masks from RGB images by semantic segmentation, and then fuse it with point clouds (Vora et al., 2020; Xie et al., 2020; Yin et al., 2021). In this way, we fuse them in a multi-task fashion, making it clear what each modality have learned.

5.2 Information Loss during Fusion

Another challenge lies in the information loss during the fusion process, especially in feature fusion. Information loss is mainly determined by fusion granularity, namely, RoI-wise, voxel-wise, point-wise, and pixel-wise. All these granularity levels lead to information loss at a certain degree. Usually, a finer granularity leads to less information loss. To compensate, we can employ another ROI-wise fusion for further refinement. In order to boost the performance of the fusion network, we can consider using Neural Architecture Search (NAS) (Liu et al., 2019; Tang et al., 2020). It aims at finding the optimal number of neurons and layers, or other *hyperparameters* in a neural network. NAS defines search space and then devises a search algorithm to propose some neural architectures, these proposals are measured by the the evaluation strategy, then a reward is fed to the search algorithm to perform optimization.

In addition, we could design loss function as supervision for the multi-modal fusion task. For example, CE loss in EPNet (Huang et al., 2020) is adopted to encourage the consistency between the classification score and IoU between the prediction and the ground-truth box. In the future research, we can strive to devise loss functions that are better suited to serve the multi-modal object detection purposes.

5.3 Multi-Modality Data Augmentation

Due to the limited number of objects in the dataset, data augmentation is a common practice to ensure the neural network to learn more efficiently and avoid overfitting during training. The existing data augmentation techniques (Luan et al., 2017) proposed for single-modality detection are equally useful for deep fusion methods. The common operations include object cut-and-paste, random flipping, scaling, rotation, and so on. We can easily perform data augmentations on RGB images or LiDAR point clouds, but not on multi-modal fusion methods. To perform multi-modal data augmentation, we need to build the mapping between data elements (such as points or pixels). Unfortunately, when performing data augmentation in each data modality, these *random* augmentation operations likely break the mapping cross modalities.

There are a few recent methods (Wang et al. (2021); Zhang et al. (2020a) trying to address this problem. Zhang et al. (2020a) presents a new multi-modality augmentation approach by cutting point cloud and imagery patches of ground-truth objects and pasting them into different scenes in a consistent manner, which prevents misalignment between multi-modality data. When projecting 3D points to 2D pixels, it performs reverse operation of translation, rotation, flip, etc., to recover the original point cloud, then gets point-pixel mapping based on the calibration information.

5.4 Dataset and Metrics

5.4.1 Dataset

As today's deep learning is quite data-driven, another bottleneck in multi-modal 3D detection is the availability of high-quality, publicly usable datasets annotated with ground-truth information. Currently, popular datasets in 3D detection have the following issues: small size, class imbalance and labeling errors, as discussed in Sec. 3. Unsupervised and weakly-supervised learning fusion frameworks could allow the networks to be trained on large unlabeled or coarse labelled dataset and leads to better performance (Caine et al., 2021). There are also emerging works on generating synthetic datasets (Gaidon et al., 2016; Ros et al., 2016; Richter et al., 2016; Dosovitskiy et al., 2017; Kar et al., 2019; Prakash et al., 2019; Weng et al., 2020). Most of the

works focus on synthetic data for RGB images, synthetic data for point clouds is yet to be developed. These synthetic datasets have the problem of domain gap between real-world datasets, which will harm the performance of models trained on them.

5.4.2 Metrics

At present, autonomous driving datasets focus on the final detection accuracy (e.g. mAP) rather than the effectiveness of the proposed fusion methods. So far, there is no metric that specifically evaluates the effectiveness of multi-sensor fusion. It is thus hard to compare different multi-modality fusion methods. We argue that, when evaluating multi-modal fusion networks, in addition to detection accuracy, we need to pay more attention to the following aspects: the extra computational overhead introduced by the fusion process, the networks robustness such as the probability-based detection quality (Hall et al., 2020), and key performance indicators (Chan and Chan, 2004), etc.

6 Conclusion

Due to the increasing importance of 3D vision in applications such as autonomous driving, this paper reviews the recent multi-modal 3D object detection networks, especially the fusion of camera images and LiDAR point clouds. We first carefully compare popular sensors and discuss their advantages and disadvantages and summarize the common problems of single-modal methods. We then provide an in depth summary of several popular datasets that are commonly used for autonomous driving. In order to provide a systematic review, we categorize the multi-modal fusion methods considering the following three dimensions: (1) where does the fusion take place in the pipeline, (2) what data representation is used for each fusion input, and (3) at what granularity is the fusion algorithm. Finally, we discuss open challenges and potential solutions in the multi-modal 3D object detection.

References

Ahmad WA, Wessel J, Ng HJ, Kissinger D (2020) Iot-ready millimeter-wave radar sensors. In: IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT), pp 1–5

Andrai P, Radii T, Mutra M, Ivoevi J (2017) Nighttime detection of uavs using thermal infrared camera. Transportation Research Procedia 28:183–190

Andriluka M, Roth S, Schiele B (2010) Monocular 3d pose estimation and tracking by detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 623–630

Arnold E, Al-Jarrah OY, Dianati M, Fallah S, Oxtoby D, Mouzakitis A (2019) A survey on 3d object detection methods for autonomous driving applications. IEEE Transactions on Intelligent Transportation Systems (TITS) 20(10):3782–3795

Asvadi A, Garrote L, Premebida C, Peixoto P, Nunes U (2017) Multimodal vehicle detection: Fusing 3d-lidar and color camera data. Pattern Recognition Letters 115

Beltrn J, Guindel C, Moreno FM, Cruzado D, Garca F, De La Escalera A (2018) Birdnet: A 3d object detection framework from lidar information. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp 3517–3523

Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, Beijbom O (2020) nuscenes: A multimodal dataset for autonomous driving. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 11618–11628

Caine B, Roelofs R, Vasudevan V, Ngiam J, Chai Y, Chen Z, Shlens J (2021) Pseudo-labeling for scalable 3d object detection. CoRR abs/2103.02093

Carr P, Sheikh Y, Matthews I (2012) Monocular object detection using 3d geometric primitives. In: Fitzgibbon A, Lazebnik S, Perona P, Sato Y, Schmid C (eds) European Conference on Computer Vision (ECCV), pp 864–878

Chadwick S, Maddern W, Newman P (2019) Distant vehicle detection using radar and vision. In: IEEE International Conference on Robotics and Automation (ICRA), pp 8311–8317

Chan A, Chan A (2004) Key performance indicators for measuring construction success. Benchmarking 11(2):203–221

Chang J, Chen Y (2018) Pyramid stereo matching network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, pp 5410–5418

Chang MF, Lambert J, Sangkloy P, Singh J, Bak S, Hartnett A, Wang D, Carr P, Lucey S, Ramanan D, Hays J (2019) Argoverse: 3d tracking and forecasting with rich maps. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 8740–8749

Charles RQ, Su H, Kaichun M, Guibas LJ (2017) Pointnet: Deep learning on point sets for 3d classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 77–85

Chen L, Zou Q, Pan Z, Lai D, Cao D (2019) Surrounding vehicle detection using an fpga panoramic camera and deep cnns. IEEE Transactions on Intelligent Transportation Systems PP(99):1–13

Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018a) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI) 40(4):834–848

Chen X, Kundu K, Zhang Z, Ma H, Fidler S, Urtasun R (2016) Monocular 3d object detection for autonomous driving. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2147–2156

Chen X, Ma H, Wan J, Li B, Xia T (2017) Multi-view 3d object detection network for autonomous driving. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1907–1915

Chen X, Kundu K, Zhu Y, Ma H, Fidler S, Urtasun R (2018b) 3d object proposals using stereo imagery for accurate object class detection. IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI) 40(5):1259–1272

Cui Y, Chen R, Chu W, Chen L, Tian D, Li Y, Cao D (2021) Deep learning for image and point cloud fusion in autonomous driving: A review. IEEE Transactions on Intelligent Transportation Systems (TITS) pp 1–18

Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 248–255

Deng J, Shi S, Li P, Zhou W, Zhang Y, Li H (2020) Voxel r-cnn: Towards high performance voxel-based 3d object detection. arXiv:201215712

Ding Z, Hu Y, Ge R, Huang L, Chen S, Wang Y, Liao J (2020) 1st place solution for waymo open dataset challenge - 3d detection and domain adaptation. CoRR abs/2006.15505

Dosovitskiy A, Ros G, Codevilla F, Lopez A, Koltun V (2017) CARLA: An open urban driving simulator. In: Proceedings of the Annual Conference on Robot Learning, pp 1–16

Engelberg T, Niem W (2009) Method for classifying an object using a stereo camera. US Patent App. 10/589,641

Enzweiler M, Gavrila DM (2009) Monocular pedestrian detection: Survey and experiments. IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI) 31:2179–2195

Fayyad J, Jaradat M, Gruyer D, Najjaran H (2020) Deep learning sensor fusion for autonomous vehicle perception and localization: A review. Sensors 20:4220

Feng D, Haase-Schütz C, Rosenbaum L, Hertlein H, Gläser C, Timm F, Wiesbeck W, Dietmayer K (2021) Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. IEEE Transactions on Intelligent Transportation Systems (TITS) 22(3):1341–1360

Fu H, Gong M, Wang C, Batmanghelich K, Tao D (2018) Deep ordinal regression network for monocular depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, pp 2002–2011

Gaidon A, Wang Q, Cabon Y, Vig E (2016) Virtualworlds as proxy for multi-object tracking analysis. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 4340–4349

Ge R, Ding Z, Hu Y, Wang Y, Chen S, Huang L, Li Y (2020) Afdet: Anchor free one stage 3d object detection. CoRR abs/2006.12671

Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3354–3361

Girshick R (2015) Fast r-cnn. In: IEEE International Conference on Computer Vision (ICCV), pp 1440–1448

Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 580–587

Guo J, Kurup U, Shah M (2019) Is it safe to drive? an overview of factors, metrics, and datasets for driveability assessment in autonomous driving. IEEE Transactions on Intelligent Transportation Systems PP(99):1–17

Hall D, Dayoub F, Skinner J, Zhang H, Miller D, Corke P, Carneiro G, Angelova A, Sünderhauf N (2020) Probabilistic object detection: Definition and evaluation. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp 1031–1040

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778

He K, Gkioxari G, Dollár P, Girshick RB (2017) Mask R-CNN. In: IEEE International Conference on Computer Vision (ICCV), pp 2980–2988

He T, Soatto S (2019) Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. In: Association for the Advancement of Artificial Intelligence (AAAI), vol 33, pp 8409–8416

Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2261–2269

Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K (2017) Speed/accuracy trade-offs for modern convolutional object detectors. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3296–3297

Huang P, Cheng M, Chen Y, Luo H, Wang C, Li J (2017) Traffic sign occlusion detection using mobile laser scanning point clouds. IEEE Transactions on Intelligent Transportation Systems 18(9):2364–2376

Huang T, Liu Z, Chen X, Bai X (2020) Epnet: Enhancing point features with image semantics for 3d object detection. In: European Conference on Computer Vision (ECCV), vol 12360, pp 35–52

Huang X, Wang P, Cheng X, Zhou D, Geng Q, Yang R (2019) The apolloscape open dataset for autonomous driving and its application. IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI) 42(10):2702–2719

Jiang M, Wu Y, Lu C (2018) Pointsift: A sift-like network module for 3d point cloud semantic segmentation. CoRR abs/1807.00652

Kar A, Prakash A, Liu MY, Cameracci E, Yuan J, Rusiniak M, Acuna D, Torralba A, Fidler S (2019) Meta-sim: Learning to generate synthetic datasets. In: IEEE International Conference on Computer Vision (ICCV), pp 4550–4559

Kellner D, Klappstein J, Dietmayer K (2012) Grid-based DBSCAN for clustering extended objects in radar data. In: IEEE Intelligent Vehicles Symposium (IV), pp 365–370

Kim K, Woo W (2005a) A Multi-view Camera Tracking for Modeling of Indoor Environment. Springer Berlin Heidelberg

Kim K, Woo W (2005b) A multi-view camera tracking for modeling of indoor environment. In: Aizawa K, Nakamura Y, Satoh S (eds) Advances in Multimedia Information Processing - PCM 2004, pp 288–297

Kim S, Kim H, Yoo W, Huh K (2016) Sensor fusion algorithm design in detecting vehicles using laser scanner and stereo vision. IEEE Transactions on Intelligent Transportation Systems 17(4):1072–1084

Kim Y (2014) Convolutional neural networks for sentence classification. Eprint Arxiv

Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto

Ku J, Mozifian M, Lee J, Harakeh A, Waslander SL (2018) Joint 3d proposal generation and object detection from view aggregation. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp 1–8

Lang AH, Vora S, Caesar H, Zhou L, Yang J, Beijbom O (2019) Pointpillars: Fast encoders for object detection from point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 12697–12705

Lee CH, Lim YC, Kwon S, Lee JH (2011) Stereo vision-based vehicle detection using a road feature and disparity histogram. Optical Engineering 50(2):027004–027004–23

Lee S (2014) Time-of-flight depth camera motion blur detection and deblurring. IEEE Signal Processing Letters 21(6):663–666

Lee S (2020) Deep learning on radar centric 3d object detection. CoRR abs/2003.00851

Levinson J, Thrun S (2013) Automatic online calibration of cameras and lasers. In: Robotics: Science and Systems, vol 2, p 7

Li B (2017) 3d fully convolutional network for vehicle detection in point cloud. In: IEEE International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 1513–1518

Li P, Chen X, Shen S (2019) Stereo r-cnn based 3d object detection for autonomous driving. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 7644–7652

Li Y, Bu R, Sun M, Wu W, Di X, Chen B (2018) Pointcnn: Convolution on x-transformed points. In: Advances in Neural Information Processing Systems (NeurIPS), pp 828–838

Liang M, Yang B, Wang S, Urtasun R (2018) Deep continuous fusion for multi-sensor 3d object detection. In: European Conference on Computer Vision (ECCV), pp 663–678

Liang M, Yang B, Chen Y, Hu R, Urtasun R (2019) Multi-task multi-sensor fusion for 3d object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 7337–7345

Liang Z, Zhang M, Zhang Z, Zhao X, Pu S (2020) Rangercnn: Towards fast and accurate 3d object detection with range image representation. CoRR abs/2009.00206

Lin T, Dollr P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 936–944

Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV), pp 740–755

Lin TY, Goyal P, Girshick R, He K, Dollr P (2017) Focal loss for dense object detection. IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI) PP(99):2999–3007

Liu H, Simonyan K, Yang Y (2019) DARTS: differentiable architecture search. In: International Conference on Learning Representations (ICLR)

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: Single shot multibox detector. In: Leibe B, Matas J, Sebe N, Welling M (eds) European Conference on Computer Vision (ECCV), pp 21–37

Liu Z, Zhao X, Huang T, Hu R, Zhou Y, Bai X (2019) TANet: Robust 3D Object Detection from Point Clouds with Triple Attention. arXiv e-prints p arXiv:1912.05163

Liu Z, Wu Z, Tth R (2020) Smoke: Single-stage monocular 3d object detection via keypoint estimation. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 4289–4298

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI) 39(4):640–651

Lu H, Chen X, Zhang G, Zhou Q, Ma Y, Zhao Y (2019) Scanet: Spatial-channel attention network for 3d object detection. In: IEEE International Conference on Acoustics, Speech and SP (ICASSP), pp 1992–1996

Luan F, Paris S, Shechtman E, Bala K (2017) Deep photo style transfer. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 6997–7005

Ma X, Wang Z, Li H, Zhang P, Ouyang W, Fan X (2019) Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In: IEEE International Conference on Computer Vision (ICCV), pp 6851–6860

Major B, Fontijne D, Ansari A, Sukhavasi RT, Gowaiker R, Hamilton M, Lee S, Grzechnik SK, Subramanian S (2019) Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors. In: IEEE International Conference on Computer Vision Workshop (ICCVW), pp 924–932

Marchand , Chaumette F (1999) An autonomous active vision system for complete and accurate 3d scene reconstruction. International Journal on Computer Vision (IJCV) 32(3):171–194

Milioto A, Vizzo I, Behley J, Stachniss C (2019) Rangenet ++: Fast and accurate lidar semantic segmentation. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp 4213–4220

Mnih V, Kavukcuoglu K, Silver D, Rusu A, Veness J, Bellemare M, Graves A, Riedmiller M, Fidjeland A, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D (2015) Human-level control through deep reinforcement learning. Nature 518:529–33

Moosmann F, Stiller C (2011) Velodyne slam. In: IEEE Intelligent Vehicles Symposium (IV), pp 393–398

Mousavian A, Anguelov D, Flynn J, Koeck J (2017) 3d bounding box estimation using deep learning and geometry. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 5632–5640

Nabati R, Qi H (2019) RRPN: radar region proposal network for object detection in autonomous vehicles. In: IEEE International Conference on Image Processing (ICIP), pp 3093–3097

Nabati R, Qi H (2021) Centerfusion: Center-based radar and camera fusion for 3d object detection. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp 1527–1536

Pandey G, McBride JR, Savarese S, Eustice RM (2012) Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In: Association for the Advancement of Artificial Intelligence (AAAI), p 20532059

Pang S, Morris D, Radha H (2020) Clocs: Camera-lidar object candidates fusion for 3d object detection. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp 10386–10393

Park JY, Chu CW, Kim HW, Lim SJ, Park JC, Koo BK (2009) Multi-view camera color calibration method using color checker chart

Patil A, Malla S, Gang H, Chen YT (2019) The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. In: IEEE International Conference on Robotics and Automation (ICRA), pp 9552–9557

Pon AD, Ku J, Li C, Waslander SL (2020) Object-centric stereo matching for 3d object detection. In: IEEE International Conference on Robotics and Automation (ICRA), pp 8383–8389

Prakash A, Boochoon S, Brophy M, Acuna D, Cameracci E, State G, Shapira O, Birchfield S (2019) Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In: IEEE International Conference on Robotics and Automation (ICRA), pp 7249–7255

Qi CR, Yi L, Su H, Guibas LJ (2017) Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems (NeurIPS), vol 30

Qi CR, Liu W, Wu C, Su H, Guibas LJ (2018) Frustum pointnets for 3d object detection from rgb-d data. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 918–927

Qi CR, Litany O, He K, Guibas L (2019) Deep hough voting for 3d object detection in point clouds. In: IEEE International Conference on Computer Vision (ICCV), pp 9276–9285

Qin Z, Wang J, Lu Y (2019a) Monogrnet: A geometric reasoning network for monocular 3d object localization. In: Association for the Advancement of Artificial Intelligence (AAAI), vol 33, pp 8851–8858

Qin Z, Wang J, Lu Y (2019b) Triangulation learning network: From monocular to stereo 3d object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 7615–7623

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 779–788

Ren S, He K, Girshick R, Sun J (2017) Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI) 39(6):1137–1149

Repairer Driven News (2018) Velodyne: Leading LIDAR price halved, new high-res product to improve self-driving cars. https://www.repairerdrivennews.com/2018/01/02/velodyne-leading-lidar-price-halved-new-high-res-product-to-improve-self-driving-cars/

Richter SR, Vineet V, Roth S, Koltun V (2016) Playing for data: Ground truth from computer games. In: Leibe B, Matas J, Sebe N, Welling M (eds) European Conference on Computer Vision (ECCV), pp 102–118

Riegler G, Ulusoy AO, Geiger A (2017) Octnet: Learning deep 3d representations at high resolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, pp 6620–6629

Rojas JC, Crisman JD (1997) Vehicle detection in color images. In: Proceedings of Conference on Intelligent Transportation Systems, pp 403–408

Ronneberger O, PFischer, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), vol 9351, pp 234–241

Ros G, Sellart L, Materzynska J, Vazquez D, Lopez AM (2016) The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3234–3243

Schlosser J, Chow CK, Kira Z (2016) Fusing lidar and images for pedestrian detection using convolutional neural networks. In: IEEE International Conference on Robotics and Automation (ICRA), pp 2198–2205

Schneider N, Piewak F, Stiller C, Franke U (2017) Regnet: Multimodal sensor registration using deep neural networks. In: IEEE Intelligent Vehicles Symposium (IV), pp 1803–1810

Sheeny M, Pellegrin ED, Mukherjee S, Ahrabian A, Wang S, Wallace AM (2021) RADIATE: A radar dataset for automotive perception. In: IEEE International Conference on Robotics and Automation (ICRA)

Shi S, Wang X, Li H (2019) Pointrcnn: 3d object proposal generation and detection from point cloud. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–779

Shi S, Guo C, Jiang L, Wang Z, Shi J, Wang X, Li H (2020a) Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 10526–10535

Shi S, Wang Z, Shi J, Wang X, Li H (2020b) From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI) pp 1–1

Shin K, Kwon YP, Tomizuka M (2019) Roarnet: A robust 3d object detection based on region approximation refinement. In: IEEE Intelligent Vehicles Symposium (IV), pp 2510–2515

Silver D, Huang A, Maddison C, Guez A, Sifre L, Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D (2016) Mastering the game of go with deep neural networks and tree search. Nature 529:484–489

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Computer Science

Sindagi VA, Zhou Y, Tuzel O (2019) Mvx-net: Multimodal voxelnet for 3d object detection. In: IEEE International Conference on Robotics and Automation (ICRA), pp 7276–7282

Strecha C, von Hansen W, Van Gool L, Fua P, Thoennessen U (2008) On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1–8

Sun P, Kretzschmar H, Dotiwalla X, Chouard A, Patnaik V, Tsui P, Guo J, Zhou Y, Chai Y, Caine B, Vasudevan V, Han W, Ngiam J, Zhao H, Timofeev A, Ettinger S, Krivokon M, Gao A, Joshi A, Zhang Y, Shlens J, Chen Z, Anguelov D (2020a) Scalability in perception for autonomous driving: Waymo open

dataset. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp 2443–2451

Sun X, Wang S, Wang M, Cheng SS, Liu M (2020b) An advanced lidar point cloud sequence coding scheme for autonomous driving. In: ACM International Conference on Multimedia (ACM MM), p 27932801

Sun Y, Zuo W, Yun P, Wang H, Liu M (2020c) Fuseseg: Semantic segmentation of urban scenes based on rgb and thermal data fusion. IEEE Transactions on Automation Science and Engineering PP(99):1–12

Tang H, Liu Z, Zhao S, Lin Y, Lin J, Wang H, Han S (2020) Searching efficient 3d architectures with sparse point-voxel convolution. In: European Conference on Computer Vision (ECCV), pp 685–702

Urmson C, Anhalt J, Bagnell D, Baker C, Bittner R, Clark M, Dolan J, Duggins D, Galatali T, Geyer C, et al. (2008) Autonomous driving in urban environments: Boss and the urban challenge. Journal of Field Robotics 25(8):425–466

Urmson C, Baker C, Dolan J, Rybski P, Salesky B, Whittaker WR, Ferguson D, Darms M (2009) Autonomous driving in traffic: Boss and the urban challenge. Ai Magazine 30(2):17–28

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Lu, Polosukhin I (2017) Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS), vol 30, p 60006010

Vora S, Lang AH, Helou B, Beijbom O (2020) Pointpainting: Sequential fusion for 3d object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 4603–4611

Wandinger U (2005) Introduction to Lidar. Brooks/Cole Pub. Co.,

Wang C, Ma C, Zhu M, Yang X (2021) Pointaugmenting: Cross-modal augmentation for 3d object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 11794–11803

Wang G, Tian B, Zhang Y, Chen L, Cao D, Wu J (2020) Multi-View Adaptive Fusion Network for 3D Object Detection. arXiv e-prints p arXiv:2011.00652

Wang J, Zhou L (2019) Traffic light recognition with high dynamic range imaging and deep learning. IEEE Transactions on Intelligent Transportation Systems 20(4):1341–1352

Wang S, Suo S, Ma W, Pokrovsky A, Urtasun R (2018) Deep parametric continuous convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2589–2597

Wang Y, Chao WL, Garg D, Hariharan B, Campbell M, Weinberger KQ (2019) Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 8437–8445

Wang Z, Jia K (2019a) Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp 1742–1749

Wang Z, Jia K (2019b) Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp 1742–1749

Weng X, Man Y, Cheng D, Park J, O'Toole M, Kitani K (2020) All-In-One Drive: A Large-Scale Comprehensive Perception Dataset with High-Density Long-Range Point Clouds. arXiv

Xie J, Kiefel M, Sun MT, Geiger A (2016) Semantic instance annotation of street scenes by 3d to 2d label transfer. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3688–3697

Xie L, Xiang C, Yu Z, Xu G, Yang Z, Cai D, He X (2020) Pi-rcnn: An efficient multi-sensor 3d object detector with point-based attentive cont-conv fusion module. In: Association for the Advancement of Artificial Intelligence (AAAI), vol 34, pp 12460–12467

Xu D, Anguelov D, Jain A (2018) Pointfusion: Deep sensor fusion for 3d bounding box estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 244–253

Yan Y, Mao Y, Li B (2018) SECOND: sparsely embedded convolutional detection. Sensors 18(10):3337

Yang B, Luo W, Urtasun R (2018) Pixor: Real-time 3d object detection from point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 7652–7660

Yang B, Luo W, Urtasun R (2018a) PIXOR: real-time 3d object detection from point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 7652–7660

Yang B, Guo R, Liang M, Casas S, Urtasun R (2020) Radarnet: Exploiting radar for robust perception of dynamic objects. In: European Conference on Computer Vision (ECCV), vol 12363, pp 496–512

Yang Z, Sun Y, Liu S, Shen X, Jia J (2018b) IPOD: intensive point-based object detector for point cloud. CoRR

Yang Z, Sun Y, Liu S, Jia J (2020) 3dssd: Point-based 3d single stage object detector. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 11037–11045

Yin T, Zhou X, Krähenbühl P (2021) Center-based 3d object detection and tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 11784–11793

Yoo J, Ahn N, Sohn K (2020a) Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 8372–8381

Yoo JH, Kim Y, Kim J, Choi JW (2020b) 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In: European Conference on Computer Vision (ECCV), pp 720–736

Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems (NeurIPS), vol 27

You Y, Wang Y, Chao W, Garg D, Pleiss G, Hariharan B, Campbell ME, Weinberger KQ (2020) Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. In: International Conference on Learning Representations (ICLR)

Zewge NS, Kim Y, Kim J, Kim JH (2019) Millimeter-wave radar and rgb-d camera sensor fusion for real-time people detection and tracking. In: 2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA), pp 93–98

Zhang H, Yang D, Yurtsever E, Redmill KA, mit zgner (2021a) Faraway-frustum: Dealing with lidar sparsity for 3d object detection using fusion. 2011.01404

Zhang W, Wang Z, Loy CC (2020a) Multi-modality cut and paste for 3d object detection. 2012.12741

Zhang Y, Zhang S, Zhang Y, Ji J, Duan Y, Huang Y, Peng J, Zhang Y (2020b) Multi-modality fusion perception and computing in autonomous driving. Journal of Computer Research and Development 57(9):1781

Zhang Y, Lu J, Zhou J (2021b) Objects are different: Flexible monocular 3d object detection. CoRR abs/2104.02323

Zhao X, Liu Z, Hu R, Huang K (2019) 3d object detection using scale invariant and feature reweighting networks. In: Association for the Advancement of Artificial Intelligence (AAAI), pp 9267–9274

Zhou Y, Tuzel O (2018) Voxelnet: End-to-end learning for point cloud based 3d object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 4490–4499

Zhou Y, Sun P, Zhang Y, Anguelov D, Gao J, Ouyang T, Guo J, Ngiam J, Vasudevan V (2020a) End-to-end multi-view fusion for 3d object detection in lidar point clouds. In: Proceedings of the Conference on Robot Learning, vol 100, pp 923–932

Zhou Y, Wan G, Hou S, Yu L, Wang G, Rui X, Song S (2020b) Da4ad: End-to-end deep attention-based visual localization for autonomous driving. In: European Conference on Computer Vision (ECCV), pp 271–289