Multi-Sensor Fusion Technology for 3D Object Detection in Autonomous Driving: A Review

Xuan Wang, *Member, IEEE*, Kaiqiang Li, and Abdellah Chehri, *Senior Member, IEEE*

***Abstract*— With the development of society, technological progress, and new needs, autonomous driving has become a trendy topic in smart cities. Due to technological limitations, autonomous driving is used mainly in limited and low-speed scenarios such as logistics and distribution, shared transport, unmanned retail, and other systems. On the other hand, the natural driving environment is complicated and unpredictable. As a result, to achieve all-weather and robust autonomous driving, the vehicle must precisely understand its environment. The self-driving cars are outfitted with a plethora of sensors to detect their environment. In order to provide researchers with a better understanding of the technical solutions for multi-sensor fusion, this paper provides a comprehensive review of multi-sensor fusion 3D object detection networks according to the fusion location, focusing on the most popular LiDAR and cameras currently in use. Furthermore, we describe the popular datasets and assessment metrics used for 3D object detection, as well as the problems and future prospects of 3D object detection in autonomous driving.**

***Index Terms*— Autonomous driving, smart cities, multi-sensor fusion, 3D object detection, LiDAR.**

I. Introduction

TRAFFIC congestion is a substantial hindrance to economic progress, with serious consequences for the social and economic sectors, as well as impediments to the advancement of society and sustainable cities. Significant breakthroughs in autonomous driving could bring about significant changes in human life, such as reducing the carbon emissions produced by transportation, reducing the amount of time spent commuting, improving transportation efficiency, and contributing to the development of smart cities [1].

As a result, auto manufacturers have continued to introduce vehicles with assisted driving features, which suggests that the field of autonomous driving is currently seeing rapid growth. Currently, most car companies can achieve L2 level autonomous driving, and a few can achieve L3 level autonomous driving

The identification of objects in two dimensions has seen important advances recently. Nevertheless, because 2D object detection can only provide confidence scores for 2D edges and categories of things, it cannot provide the distance information required for autonomous vehicles.

In a driving environment, self-driving cars must detect not only the distance to an object's category, but also the rotation angle and even the object's speed [2]. Autonomous vehicles must therefore have 3D object detection systems. At the top and bottom of the autonomous driving system are 3D object detection algorithms responsible for processing sensor inputs so that the autonomous car can "see" its surroundings. On the other hand, it will predict the surrounding environment based on what it observes and establish the subsequent driving trajectory to guide the vehicle's control system to do actions such as acceleration, braking, and steering.

The development of 3D object detection algorithms has produced numerous subfields. 3D object identification methods are commonly divided into two groups: a single-sensor approach and a multi-sensor fusion technique, also known as LiDAR-camera, radar-camera, LiDAR-radar-camera. The first method, "using only one sensor for 3D object recognition," refers to the practice, as its name suggests. The latter term refers to using two or more sensors working together to improve their ability to detect three-dimensional objects. Common single-modal 3D object detection algorithms are presented in Section II-B.

Detection of 3D objects has its own set of challenges. Single-sensor approaches are frequently restricted by a lack of depth information or excessive similarity of object characteristics; for instance, the radar point clouds characteristics of utility poles and individuals are remarkably similar. Methods employing simply LiDAR as a sensing device are incapable of distinguishing between them.

Despite the fact that numerous types of 3D object detection methods have been carefully summarized and compared in previous works [2], [3], [4], [5], there are relatively few review works that compare the algorithms at the level of experimental results visualization. In this study, we focus on multi-sensor fusion for 3D object detection and reproduce some representative methods to help the readers understand 3D object detection in a visual format and evaluate the performance of 3D object detection algorithms in real-world driving scenarios.

Manuscript received 13 February 2023; revised 7 June 2023 and 21 August 2023; accepted 13 September 2023. This work was supported in part by the Natural Science Foundation of Shandong Province under Grant ZR2020QF108 and Grant ZR2022QF037 and in part by the Youth Innovation Science and Technology Support Program of Shandong Province under Grant 2021KJ080. The Associate Editor for this article was C. Wen. *(Corresponding author: Abdellah Chehri.)*

Xuan Wang and Kaiqiang Li are with the School of Computer and Control Engineering, Yantai University, Yantai 264005, China (e-mail: xuanwang91@ytu.edu.cn; likaiqiang@s.ytu.edu.cn).

Abdellah Chehri is with the Department of Mathematics and Computer Science, Royal Military College of Canada, Kingston, ON K7K 7B4, Canada (e-mail: chehri@rmc.ca).

Digital Object Identifier 10.1109/TITS.2023.3317372

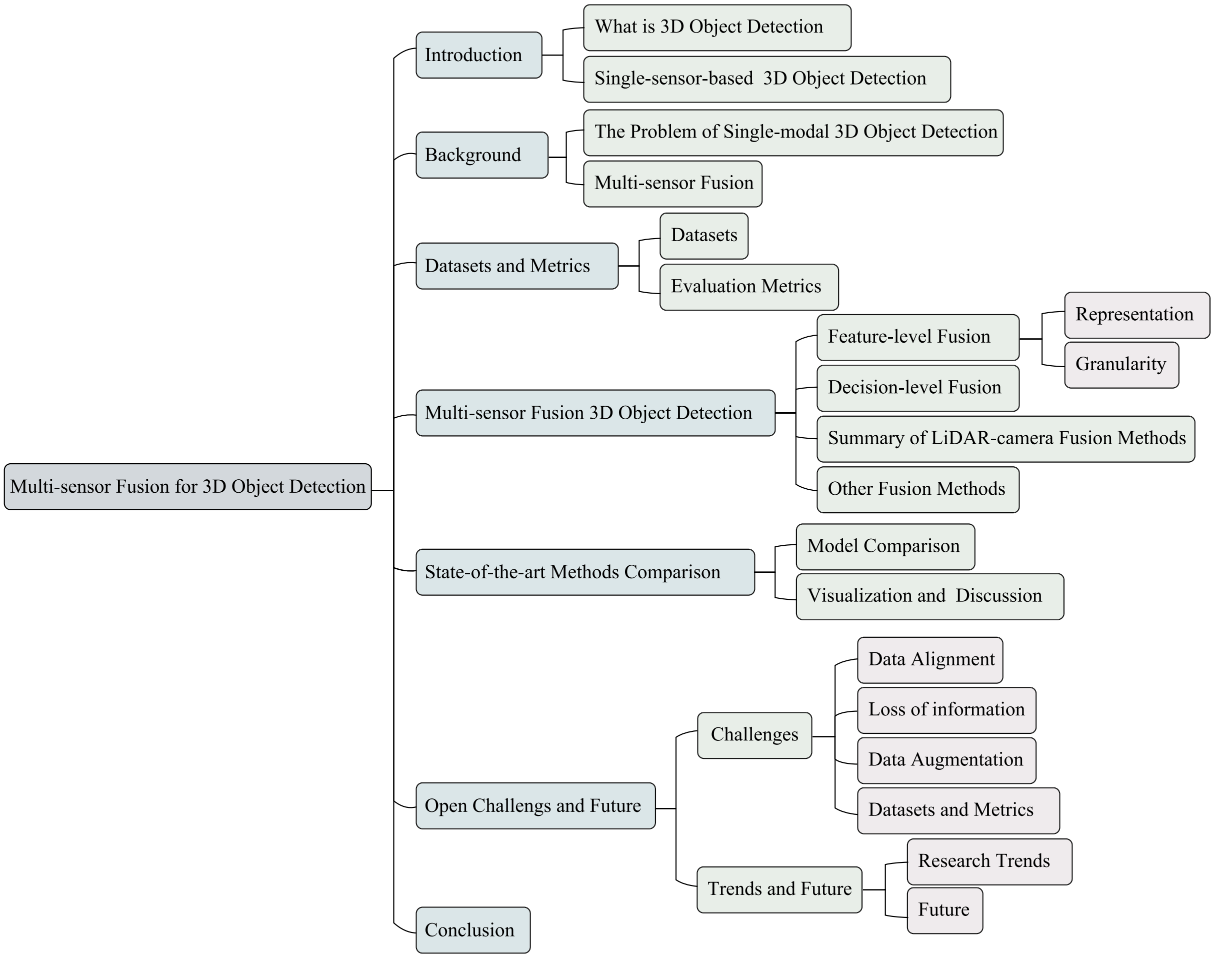

Multi-sensor Fusion for 3D Object Detection
Introduction
What is 3D Object Detection
Single-sensor-based 3D Object Detection
Background
The Problem of Single-modal 3D Object Detection
Multi-sensor Fusion
Datasets and Metrics
Datasets
Evaluation Metrics
Multi-sensor Fusion 3D Object Detection
Feature-level Fusion
Representation
Granularity
Decision-level Fusion
Summary of LiDAR-camera Fusion Methods
Other Fusion Methods
State-of-the-art Methods Comparison
Model Comparison
Visualization and Discussion
Open Challengs and Future
Challenges
Data Alignment
Loss of information
Data Augmentation
Datasets and Metrics
Trends and Future
Research Trends
Future
Conclusion

Fig. 1. Hierarchically-structured taxonomy of multi-sensor fusion 3D object detection for autonomous driving.

The contributions of this paper are as follows:

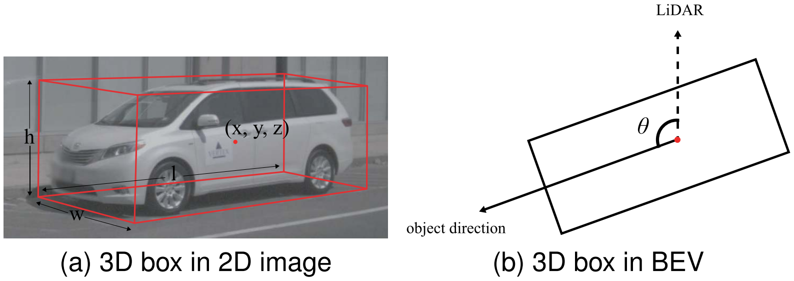
- We overview and briefly describe the most common 3D object detection datasets currently used in autonomous driving scenarios.
- We describe the most prominent 3D object detection techniques based on LiDAR-camera fusion and provide an in-depth discussion centered on the fusion position.
- We illustrate the performance of 3D object detection in a variety of settings through the depiction of the results of multiple approaches.
- We examine the challenges and future trends of 3D object identification in autonomous driving, as well as the prognosis of the impact that autonomous driving will have on the future world, with the aim that this will better stimulate future research.

The following is a brief description of the remainder of this paper. We present in Section II the data representation of sensors commonly used for autonomous driving and their representative detection networks. In Section III, we summarize the current datasets used for autonomous driving scenarios and make a brief comparison. In Section IV, we detail 3D object detection based on multi-sensor fusion. We first start with sensor devices and analyze the existing popular sensor combinations. Then, multi-sensor fusion 3D object detection is divided into two categories based on the fusion position, followed by a detailed description of the two types of schemes. In Section V, we compare popular 3D object detection schemes of recent years and present the visualization results. Finally, we conclude with a summary of the current challenges and an outlook for the future. The structure diagram of the article is shown in Figure 1.

II. Background and Related Works

A. What Is 3D Object Detection?

The goal of 3D object detection is to generate accurate attribute predictions for real-world objects, such as their size, rotation angle, and other relevant characteristics. When used for autonomous driving, 3D object detection also frequently makes predictions regarding the velocity of the objects being detected. Currently, the most common applications for 3D object detection are those associated with autonomous driving, as well as 3D object detection methods for usage in interior scenarios [6], [7]. In comparison to interior contexts, driving sceneries are dynamic, complicated, and highly changing; they are also quite demanding in terms of forecast speed. As shown in 2a, in 3D object identification methods, a rectangle is typically utilized to enclose the 3D object, and this rectangle



(a) 3D box in 2D image

(b) 3D box in BEV

Fig. 2. Example of 3D object detection results.

is typically represented as follows.

$$B = \left[x_c, y_c, z_c, l, w, h, \theta, \phi, \varphi, class\right], \quad (1)$$

where (x_c, y_c, z_c) is denoted center coordinates of the rectangle, (l, w, h) is denoted the length, width, and height of the rectangular, (θ, ϕ, φ) are indicated and roll respectively. For the current autopilot stage, all objects are on the ground, so only the yaw angle θ needs to be considered. As shown in Figure 2b. $class$ indicates the class of the 3D object. In addition, in some methods, the object's speed is also predicted [8].

Commonly sensors: Unlike 2D object detection, which typically employs only cameras as input, 3D object detection can use several sensors as input to the network. Cameras, LiDAR, and radar are currently the most prevalent sensors. Following that, we will provide a quick overview of various sensors.

Cameras: Cameras are ubiquitous in our lives because they are inexpensive to manufacture, have great imaging effects, and are passive sensors. The camera can produce an $(H \times W \times 3)$ image, where (H, W) are the height and width of the image, and 3 is the number of channels per pixel, generally referring to RGB channels. The camera can acquire high-resolution images of the outside world and visualize the shape of objects, but in 3D object detection, the camera has limitations. First, cameras have poor nighttime imaging, and although some algorithms [9] enable cameras to image at night to approximate daytime levels, this is achieved by increasing exposure time at the expense of time, which is a fatal problem at autonomous driving. Second, the camera does not provide good depth information, and using 2D images to forecast depth information with a trained network frequently results in substantial inaccuracies. Furthermore, the camera is weather-sensitive, and imaging is far less effective in wet and foggy conditions than in sunny conditions.

LiDAR: Light Detection and Ranging, or LDR for short, is a common type of active sensor. In order to determine an object's precise 3D structure, LiDAR actively generates laser beams and collects information about the reflected light, in contrast to cameras, which passively take in data. Due to its high deployment cost, autonomous driving is now constrained by the use of LiDAR despite its ability to directly capture an item's 3D structure and accurate depth information. Further, because of its short wavelength, LiDAR is subject to interference from numerous types of material in the air. Hence its effectiveness will be marginally decreased in bad weather conditions.

Radar: Radar is an active sensor with the same basic principle as LiDAR, but unlike LiDAR, Radar works by generating radio waves. Because radio waves have a larger wavelength, Radar works across a longer distance. Radar has a limited resolution, and unlike LiDAR, it cannot directly acquire the contour of an object, making it ineffective for detecting small objects [10], [11].

Conclusion: 3D object detection algorithms gather sensor information and make decisions about the surrounding targets, which is an important aspect of autonomous driving. The driving environment is complicated and varied, and 3D object identification algorithms for autonomous driving scenarios demand a high level of accuracy and reliability. Each sensor has strengths and drawbacks. Thus it has become popular to combine many sensors for object detection.

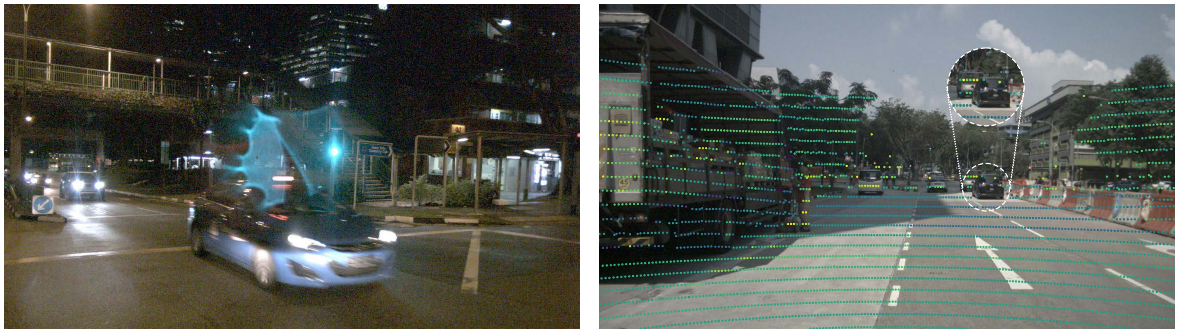
B. Single-Sensor-Based 3D Object Detection

As its name implies, single-sensor-based 3D object detection refers to the prediction of the 3D Box of a target using data from a single sensor. There are two primary popular classifications: 3D object detection via Cameras and 3D object detection via LiDAR. Due to the fact that our research focuses on multi-sensor fusion schemes, we will only briefly introduce the techniques based on the aforementioned classes without providing a comprehensive review.

1) 3D Object Detection Through Cameras: Depending on the types of cameras available, camera-based 3D object detection can be further subdivided into more specific categories. Some examples of these subcategories include monocular camera-based 3D object detection, multi-vision camera-based 3D object detection, and stereo-based 3D object detection.

a) Monocular camera-based: Monocular cameras typically feature only one lens and cannot directly determine depth in detail. They give information in the form of pixel intensities that can visually reflect an item's shape and texture information. They are a favored choice for monocular 3D object detection due to their low cost and superior imaging [12]. Camera-based 3D object detection performs depth estimation directly on the image so that it can be seen as an evolution from 2D object detection. In recent years, most camera-based 3D object detection has been done using monocular cameras [13], [14], [15], [16], [17], [18], [19].

b) Multi-vision camera-based: Currently, self-driving vehicles are typically outfitted with numerous cameras to



(a) Poor camera nighttime imaging (image from nuScenes [43])

(b) Sparse point clouds of small objects

Fig. 3. Camera night imaging effect (a) and LiDAR points projected onto the 2D image (b).

capture forward or 360° information from multiple viewpoints. 3D object detection based on multi-view cameras retrieves the same item from multiple perspectives and aggregates object properties to complete 3D object detection. This method is comparable to object re-identification [20], [21].

c) Stereo-based: Stereo cameras, often cameras with two or more image sensors, can perceive depth information by employing the human eye concept, which makes use of binocular parallax for stereo imaging [22], [23]. In contrast to monocular camera images, stereo camera images frequently appear in pairs and can be used to generate a depth image based on the correlation between the two images [24], [25], [26]. Although stereo cameras can gather more accurate depth information, 3D object detection based on stereo cameras still has a significant accuracy and performance gap compared to 3D object detection based on LiDAR. Recent years have seen an explosion in the number of 3D object identification solutions that are based on stereo cameras [15], [27], [28], [29], [30].

2) 3D Object Detection Through LiDAR: Although 3D object detection using 2D photos is very easy to implement, 2D images lack depth information, and depth information inferred using convolutional networks is frequently erroneous. With the advent of PointNet, it is now possible to conduct convolutional operations on 3D point clouds directly. Because point clouds give correct depth information, LiDAR-based 3D object detection is now the primary single-modal 3D object detection method [31].

LiDAR-based object detection is generally divided into three processes: (1) data processing, (2) feature extraction, and (3) prediction. We introduce these processes next.

a) Data processing: LiDAR will represent the acquired data as a point clouds, which is a set of data points in space, and these points can represent 3D shapes or objects. Usually, point clouds are unstructured representations, especially in the point clouds acquired by onboard LiDAR, sparsity, and its unevenness. Currently, the widely used 64-line LiDAR acquires as many as 1,152,000 data points per second, so the direct use of the original point clouds for the calculation can lead to a large computational effort. Therefore, in practical applications, the point clouds are often down-sampled.

b) Feature extract: After the initial data processing, the next step is to make the network learn features from the point clouds data. In general, several points nearby are often linked together for feature learning. Specifically, relevant points are collected within a predefined scale using ball or rectangular queries. Subsequently, with feature encoder aggregation points and contextual features, the Formation of higher-level semantic features.

c) Prediction: After successfully aggregating features from the input data, the final job is to use the new features as input to the detection head to generate the final prediction.

The LiDAR-based 3D object detection solutions in recent years are [32], [33], [34], [35]

C. The Problem of Single-Modal 3D Object Detection

Although single-sensor 3D object detection is straightforward to build and certain approaches have performed well in large dataset challenges, these solutions are not robust enough for use in real-world driving scenarios [36], [37]. Cameras and LiDAR have inherent limitations, making it difficult to meet the requirements of all-weather and real-time operational needs [38], [39]. Camera-based approaches do not give entire 3D geometric information, and the computational cost grows as camera resolution increases [40]. As illustrated in Figure 3a, weather conditions have a significant impact on the camera. Although LiDAR-based methods yield higher detection accuracy than camera-based methods [41], LiDAR-only solutions are not yet widely adopted due to high deployment costs. Furthermore, as depicted in Figure 3b, LiDAR has highly sparse point clouds on small objects at large distances [42]. Real-world driving environments are complicated and changeable, and single-sensor solutions are frequently inadequate.

D. Multi-Sensor Fusion

a) Introduction: As described in Section II-C, each type of sensor has limitations, and 3D object detection using a single sensor cannot meet the needs of all-weather driving scenarios, so fusing data from multiple sensors together for 3D object detection is a major trend, and more and more teams have proposed 3D object detection algorithms based on multi-sensor fusion, which we will discuss in detail in

TABLE I

Comparison of Popular Multi-Sensor Datasets Applicable to Autonomous Driving, Including Year, Number of Categories, Number of Scenes, and Number of 2D Annotations (n-2D) and 3D (n-3D) Annotations, as Well as Spatio-Temporal Factors (Night/Rain)

Dataset	Classes	Scenes	360°	Image	Video	n-2D	n-3D	Night	Rain	Locations
KITTI [45]	8	50	No	Yes	No	80K	80K	No	No	Germany
ApolloScape [47]	35	103	No	Yes	Yes	2.5M	70K	Yes	Yes	China
nuScenes [43]	23	1000	Yes	Yes	Yes	-	1.4M	Yes	Yes	USA, SG
Waymo [46]	4	1150	Yes	Yes	Yes	9.9M	12M	Yes	Yes	China
H3D [48]	8	160	No	Yes	No	-	1.1M	Yes	Yes	USA
Argoverse [49]	15	113	Yes	Yes	Yes	-	993K	Yes	Yes	USA
DAIR-V2X [50]	10	-	No	Yes	Yes	-	-	Yes	Yes	China

Section IV. Following that, we will briefly overview the multi-sensor fusion system and tasks. Table I depicts the general comparison of the dataset.

b) Definition: In 3D object detection, multi-sensor fusion refers to object detection using two or more sensor data; for example, LiDAR can compensate for cameras' poor imaging rate at night, and LiDAR and radar can complement each other for long-range and small objects. Multi-sensor fusion 3D object identification categorization is distinguished primarily by fusion granularity, fusion location, and fusion input. The most essential of them are distinguished by the fusion position; thus, in Section IV, we separate distinct fusion schemes primarily by the fusion position and offer a full explanation.

c) Task: In contrast to conventional 3D object recognition tasks, the multi-sensor fused 3D object detection solution requires immediate responsiveness because it is largely used in autonomous driving applications. This involves making decisions within a time frame of 50 milliseconds in order to ensure the safety of the vehicle. Additionally, it requires adaptability across varying environments and conditions, enabling predictions even in the event of a single sensor failure.

III. Datasets and Metrics

The ImageNet [44] dataset is a significant dataset in the field of artificial intelligence and has driven the development of deep learning methods in the field of artificial intelligence. As the learning material of neural networks, datasets are crucial in deep learning.

Unlike ImageNet [44] for image classification, datasets for autonomous driving scenarios need to have both 2D and 3D annotations in order to facilitate research more widely. Next, we discuss some popular 3D object detection datasets for autonomous driving.

A. Datasets

The KITTI [45] dataset is a publicly available dataset created by the Karlsruhe Institute of Technology in Germany and the Toyota Institute of Technology in the U.S. that uses specialized data collection vehicles to collect data from real traffic scenes.

The KITTI [45] dataset contains data from real driving scenes in urban, rural, and highway scenarios, with up to 15 vehicles and 30 pedestrians per image, as well as various levels of The KITTI [45] dataset, contains data from a variety of sensors, such as cameras, LIDAR, and combined GPS/IMU navigation and positioning systems. The object detection dataset contains 7481 training data and 7518 test data with sensor calibration and accurate 2D frames and 3D frames with KITTI [45] annotated category labels including *car*, *Van*, *Truck*, *Pedestrian*, *Person (sitting)*, *Cyclist*, *Tram*, and *Misc (e.g., Trailers, Segways)* or *DontCare*. In addition, the KITTI [45] dataset was classified into three cases based on whether the target was obscured, the degree of obscuration, and the height of the box: *"easy"*, *"moderate"* and *"difficult"* in order to accurately determine the accuracy of a model in all aspects.

The nuScenes [43] dataset is a public large-scale dataset for autonomous driving developed by the Motional team, which collects data primarily in the Singapore and Boston areas, two cities known for their dense traffic and challenging driving conditions. In addition, nuScenes [43] manually selects 1000 scenarios by humans out of all collected scenarios, including 700 scenarios for training, 150 scenarios for validation, and 150 scenarios for testing, each lasting approximately 20 seconds.

The complete dataset includes approximately 1.4 million camera images, 390,000 m lidar scans, 1.4 million millimeter wave radar scans, and 1.4 million object-bounding boxes in 40,000 keyframes. nuScenes [43] is inspired by the groundbreaking KITTI [45] dataset. However, its use of more sensors (6 cameras, 1 lidar, 5 radar, GPS IMU) is the first dataset to offer the entire sensor suite from autonomous driving and contains seven times more object annotations than KITTI [45]. It also includes night and rain scenarios, which are unavailable in the KITTI [45] dataset.

In addition to this, nuScenes [43] annotates object-level attributes such as visibility, activity, and pose. More notably, in July 2020, nuScenes launched nuScenes-lidarseg, which uses 32 possible semantic tags to label each lidar point in keyframes for nuScenes [43] as a response to the lidar semantic segmentation task.

The Waymo Open Dataset [46] is a dataset released by Google Waymo Driverless in 2020, with 5 RGB cameras and 5 LiDARs (three in front of the car, one on the roof, and one behind the car). Its annotations include *"car"*, *"pedestrian"*, *"bicyclist"* and *"sign"*. Waymo Open also includes nighttime and rainy weather scenes.

The ApolloScape dataset [47], released by Baidu in China, comprises about 100K image frames, 80k lidar point clouds, and 1000km trajectories for urban traffic. The ApolloScape dataset was gathered under a variety of illumination conditions and traffic volumes, including highly complicated traffic flows involving vehicles, pedestrians, and riders.

H3D [48] is a dataset for autonomous directions that was released by the Honda Research Institute in 2019. It includes more than one million annotated occurrences and 160 traffic scenarios that are extremely interactive and congested.

The Argoverse [49] dataset includes recordings across seasons, weather conditions, and time of day to provide a wide and realistic range of driving scenarios. It contains a total of 113 scenes annotated with 3D tracking. Each segment is approximately 15-30 seconds in length and contains approximately 11052 tracking targets, of which 70% of the annotations are vehicles, and the remaining objects include *pedestrians*, *bicycles*, *motorcycles*, etc.

The DAIR-V2X [50] dataset is the world's first large-scale, multi-modal, multi-view dataset for research on vehicle-road cooperative autonomous driving. It is also the first dataset to achieve simultaneous spatio-temporal annotation of vehicle-road cooperative driving, with all data collected from real driving scenarios, including both 2D and 3D annotation. In addition, the DAIR-V2X dataset is the first dataset to achieve simultaneous spatio-temporal annotation of cooperative driving between vehicles and roads. It is important to point out that onboard sensors collect the data collected by DAIR-V2X. These onboard sensors comprise 300 lines of LIDAR as well as high-resolution cameras that are installed at intersections.

B. Evaluation Metrics

We will go over several essential principles in object detection before discussing the evaluation metrics currently being used.

1) IoU (Intersection Over Union): IoU means Intersection and Union Ratio, which is the ratio of the area of two intersecting rectangular boxes to the area of two intersecting rectangular boxes merging in 2D object detection, and the ratio of the volume of two intersecting rectangular boxes to the volume of two intersecting rectangular boxes merging in 3D object detection. The greater the *IoU*, the more precise the placement.

2) Precision: Precision is a measure of the probability of a classifier predicting a true example and is often expressed by the following formula:

$$precision = \frac{TP}{TP + FP}. \tag{2}$$

3) Recall: the number of positive samples correctly identified versus the number of images whose class is really a positive class, expressed by the following formula:

$$recall = \frac{TP}{TP + FN}, \tag{3}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives.

The most commonly used evaluation metric for object detection is $AP(AveragePrecision)$, which is heavily used in 2D object detection. The original approach was to extend AP in 2D object detection to 3D space.

The AP is calculated as follows:

$$AP = \int_0^1 p(r)\, dr. \tag{4}$$

In Eq. 4, $p(r)$ is the function of precision about the $recall$. Compared to AP, a more popular evaluation metric now is $mAP(MeanAveragePrecision)$, which can be expressed as the following equation:

$$mAP = \frac{\sum_{i=1}^{\mathrm{N}} AP_i}{\mathrm{N}}, \tag{5}$$

where N is the total number of classes and AP_i denotes the average precision of the *i-th* class.

In each of the above representations, $precision$ and $recall$ are for a single data, while AP is for a certain category of objects in the entire dataset, and mAP is for the entire dataset. mAP means that AP is calculated for each category and then averaged.

IV. Multi-Sensor Fusion 3D Object Detection

In this section, we provide 3D object detection based on multi-modal fusion. In addition to fusion input and granularity, the fusion position is significant in identifying different fusion procedures. As a result, in our work, we focus on two significant fusion kinds methods: feature-level fusion and decision-level fusion, depending on the fusion location, and develop a full description for each. We concentrate on two sensors, LiDAR, and a camera, which we briefly discussed in Section II. To help the reader comprehend the following fusion methods, we will first show the popular data representation of both sensors in the fusion scheme.

LiDAR: *LiDAR Points: LiDAR Points:* Figure 4a depicts the initial point clouds created by LiDAR. A point clouds is a collection of points in a 3D coordinate system that are typically described by x, y, and z coordinates as well as reflection intensity. Point clouds give correct distance information between the emitting and reflecting points, are flip-invariant and scale-invariant, and can be supplied in full geometry.

With the introduction of PointNet [31], convolutional networks can now analyze point clouds data directly. The point-based approach directly uses the original point clouds as input and can preserve the original information to the greatest extent possible.

Due to the extremely rich number of point clouds generated by LiDAR, the computational cost of directly using point clouds as raw input is extremely high, especially for complex scenes. Therefore, the point clouds are often sampled, reducing the computational cost but with some loss of information and performance.

- *Voxels:* Generally speaking, the whole point clouds space is divided into several small spaces of uniform size, which are called voxels, as shown in Figure 4b, and a voxel contains several LiDAR points [51], [52], [53], [54], [55], [56]. For voxels, the same 3D convolution can be used directly to extract features. It is worth noting that some schemes [51] divide voxels only on the plane and make the height of each voxel

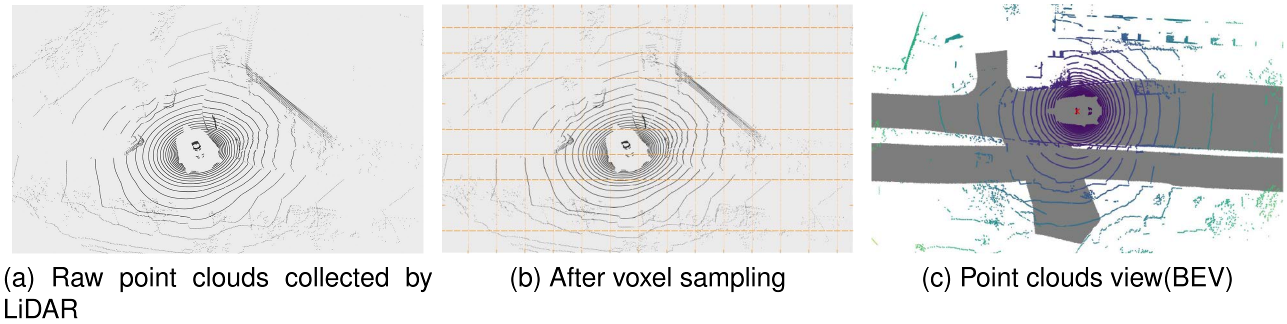

(a) Raw point clouds collected by LiDAR
(b) After voxel sampling
(c) Point clouds view(BEV)

Fig. 4. Several common representations of point clouds, each from the same scene.

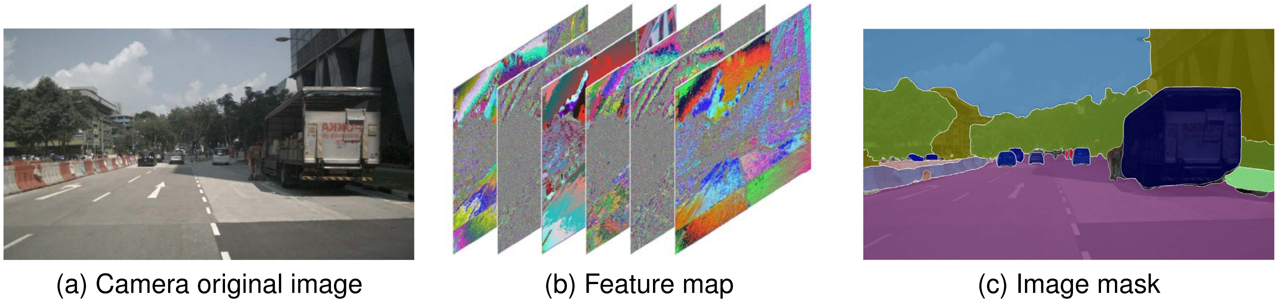

(a) Camera original image
(b) Feature map
(c) Image mask

Fig. 5. Several common data representations of the camera, each figure corresponds to the same scene.

equal to the height of the entire point clouds space; such voxels are generally called *pillar*, but in this paper we refer to them uniformly as voxels.

- *Point clouds view:* Point clouds view refers to the projection of 3D point clouds to 2D views, such as BEV, front view, etc., as shown in Figure 4c, to match with image features. After conversion to view, 2D convolution can be used directly to extract features efficiently. BEV view is usually used in autonomous driving because, at this stage of autonomous driving, objects do not overlap on the height axis [57], [58]. The conversion of 3D point clouds to 2D view projections inevitably results in information loss.

Camera: *Feature map*: A 2D image usually has three channels of RGB, and a neural network takes a 2D image as input and performs convolutional operations with a set of convolutional kernels, which then produces a multi-channel feature map, as shown in Figure 5b, in other words, what is obtained after convolutional operations of image pixels is the feature map.

In most network implementations, already pre-trained backbone networks are utilized as image feature extractors, such as ResNet [59].

- *Mask:* Image mask is an expression representing a specific image area. In computer vision, it shows a target's location and shape, containing pixel category info to distinguish foreground and background (Figure 5c).

A. Feature-Level Fusion

Feature-level fusion first involves extracting representative features from the raw observation data provided by each sensor, then fusing these features into a single feature vector. Taking LiDAR and camera as an example, the backbone network performs feature extraction on LiDAR point clouds and 2D images, respectively, and then fuses the two features and uses the fused new features for 3D object detection, the basic flow chart of which is shown in Figure 6. In the following, we briefly classify feature-level fusion methods according to the different fusion input data.

1) Fusion Input Representation: **LiDAR Points & image feature map:** Due to the irregularity of point clouds, no effective and suitable scheme could directly perform convolution operations on point clouds until PointNet [31] was proposed, Qi et al. [31] led the way to directly perform feature extraction on point clouds. Until now, direct feature extraction operations on point clouds occupy the majority of schemes.

Since the reason that the denseness of LiDAR points varies greatly, resulting in the use of LiDAR points only for 3D object detection will lead to slightly poorer detection of distant and small objects, a very common practice now is to combine LiDAR point clouds and camera images and fuse them for 3D object detection.

Xu et al. [60] used PointNet [31] to process the point clouds, and ResNet [59] was used to extract the image features from the two outputs obtained by combining the new fusion network.

PointFusion [60] fused the spatial information of the original point clouds and the texture information of the image without any information loss and took full advantage of the RGB information.

F-PointNet [61] is proposed as a multi-sensor fusion scheme by the same authors of PointNet [31], PointNet++ [62], which utilizes a very well-established object detection network in 2D images [63], [64], [65], [66], [67] to determine the 2D bounding box of an object and use the projection transformation of the camera to determine a 3D view cone and within this view,

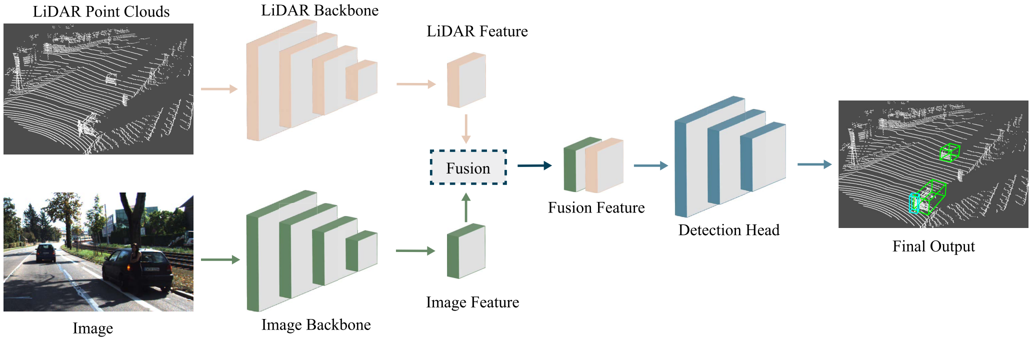


Fig. 6. Illustration of Multi-Sensor fusion 3D object detection based on feature-level fusion.

cone determines the point clouds data of the object to further regress to the 3D bounding box.

Wan et al. [68] proposed a dynamic cross-attention module (DCAN), which learns multiple offsets from the initial projection to the neighborhood to compensate for the shortcomings of common fusion schemes, thus accurately improving the ability to align LiDAR point clouds and image pixel data.

Huang et al. [69] mainly address two problems in 3D object detection: multi-sensor data fusion and settling on the inconsistency between classification confidence, which is a challenge to traditional two-stage networks. EPNet [69] performs feature enhancement on point clouds and uses image features to enhance point clouds features directly, no longer relying on image annotation.

Xu et al. [70] generate preliminary 3D proposals using SECOND [71] network and generate RoI based on the proposals. Subsequently, feature enhancement is performed on point clouds in RoI and feature extraction is performed on multi-view images using ResNet [59] and FPN.

A self-attention mechanism is subsequently used to enhance domain-specific features, and finally, fusion is performed by cross-attention to generate the final 3D proposal. It is worth mentioning that FusionRCNN [70] states that the cross-attention module is not required, which means that the two branches of FusionRCNN [70] can perform single-sensor 3D object detection separately if the cross-attention module is not applicable.

Yang et al. [72] introduced a new modal interaction strategy that learns and maintains the representation of each modality throughout the process in order to exploit their richer original features better.

LiDAR Points & image mask: Image segmentation can identify an object more accurately than 2D object detection, which can achieve pixel-level accuracy. Ordinary 2D object detection frames usually contain foreground and background, which can lead to inaccurate image features and 3D feature fusion. On the other hand, semantic segmentation can be accurate to each pixel of the object, providing more accurate image semantic information for fusion and identifying the corresponding LiDAR points more accurately.

PI-RCNN [73] varies from the preceding approach in that it employs camera image features. Image segmentation is used by PI-RCNN [73], which is made up of two sub-networks: the segmentation network and the detection network.

The segmentation network is used to segment out the objects in the image at the pixel level, and the authors also created a PACF module for connecting the two sub-networks, fusing the characteristics of the two modalities, and then feeding them to the detection network to generate 3D prediction results.

The fusion scheme proposed by Vora et al. [74] uses the fusion of 2D semantic segmentation information onto the LiDAR point clouds by projection matrix, which achieves the effect of feature enhancement of LiDAR points, and then adopts pure point clouds methods such as PointPillars [51] and PointRCNN [75] for 3D object detection.

Since LiDAR point clouds can acquire the fine structure of objects and semantic segmentation can achieve pixel-level segmentation, this can easily correspond point clouds and pixel data of the same object.

The advantage of semantic segmentation over ordinary 2D object detection is that it can accurately separate the foreground from the background and reduce the influence of the background on the fused pixels and point clouds.

Point clouds view & image feature map:

Bai et al. [76] applied the attention mechanism to the fusion framework. TransFusion [76] converts the radar point clouds into a BEV view, the first layer decoder initially predicts the initial bounding box based on the BEV view, and the second layer decoder relates object queries and image features to generate the final prediction.

Unlike mainstream methods, BEVFusion [77] mainly focuses on camera streams, complemented by LiDAR streams. Liang et al. [77] converts multi-view images into 3D representations, which in turn are represented as BEV images, and converts LiDAR points into BEV views, using a fusion module that combines camera BEV views and LiDAR BEV views, which have the same dimensionality due to both. This fusion scheme is simple to implement. Also, since BEVfusion relies mainly on camera streams, this makes BEVfusion the first framework that can cope with LiDAR failures or LiDAR data shortages.

Wang et al. [78] fused LiDAR point clouds BEV views, LiDAR point clouds distance views, and 2D images to achieve adaptive fusion using their designed attention point fusion PAF module, which adaptively selects the importance of each source using an attention mechanism.

Point clouds voxels & image feature map:

VoxelNet [52] is a pioneering work to sample point clouds as sparse voxels, which treats several point clouds as a whole, performs feature extraction, and puts the extracted features back into 3D space, turning point clouds data into ordered high-dimensional feature data.VoxelNet [52] has led a series of studies to sample point clouds as voxels.

Li et al. [79] proposed a new idea to divide the point clouds space into voxels and then use a sampler to select the information of essential regions from the image and then map them to rays and project them into the voxel field. Subsequently, the voxels on the rays are selected as the mapping of the image features, thus making full use of the contextual interactions in the 3D space.

AutoAlignV2 [80] aims to efficiently aggregate image features further to enhance the performance of the 3D object detector. Chen et al. [80] use deformable cross-attention networks to extract and aggregate features from different modalities, effectively increasing the speed of the fusion process. In addition, the *Image-Level Dropout Training Strategy* designed by the authors enables the network to support inference using only point clouds, which randomly remove image fusion features during the fusion process and fill them with zeros so that the network gradually learns to use 2D features as optional inputs, which not only speeds up the training but also improves the final performance.

Sindagi et al. [81] propose a simple and effective early fusion method that extracts features from the convolutional layer of a 2D detection network, projects the voxelized ones onto the image plane to establish an association between the point clouds and the pixels, and appends the corresponding image features to the point clouds.

Jiao et al. [82] proposed a new framework, MSMDFusion [82], which focuses on the multi-scale progression of multi-granularity LiDAR and phase features, which samples LiDAR point clouds and multi-view images as voxels and subsequently maps the obtained multi-scale LiDAR and camera features to BEV views for final prediction.

Point clouds voxels & mask:

Inspired by previous work using image masks [73], [74], Yin et al. [83] further reduced the computational effort by sampling point clouds as voxels. It also employs various data enhancement methods, such as random flips and rotations.

2) Fusion Granularity: The simplest implementation of a multi-sensor fusion operation is to fuse on the smallest data unit of each sensor, but this approach requires more computational resources. To balance detection accuracy and inference speed, depending on the network structure and parameters, all or localized regions are used for fusion. The common fusion granularities are RoI (Region of Interest), Voxel, and Point, and we will describe these three fusion granularities in detail next.

RoI-Wise: RoI (Region of Interest), delineating RoI is a typical operation in image processing similar to the attention mechanism, after dividing different RoI by algorithm, which makes the network focus mainly on this part of the region, improving learning efficiency and reducing computation.

In multi-modal fusion 3D object detection, a common operation is to delineate RoI based on 2D image object detection results, map them to 3D space to obtain 3D frustums, and use 3D detectors to process them [84].

Voxel-Wise: The RoI-wise-based scheme has too large a perceptual range and is not suitable for small object detection in some cases. In 3D space, voxels can also roughly represent the appearance of objects, so Voxel-wise can more accurately correspond 3D objects to 2D images and separate the background from the foreground compared to RoI-wise.

As mentioned in the introduction of Section IV, voxels can be considered as downsampled from point clouds, so the sparse LiDAR points can be considered as empty voxels, which can make full use of existing information and filter useless information, making Voxel-wise more accurate than RoI-wise fusion.

Point-Wise: In general, Point-wise fusion enhances the features of LiDAR points. Common point-wise feature enhancements [85], [86] are using the distance of additional LiDAR points to the center and corners of a fixed-size box, as shown in Figure 7a. Xie et al. [73] used the distances from additional LiDAR points to k proximal points as shown in Figure 7b.

In addition to the above two point-level feature enhancement methods, we can also enhance LiDAR point features with image features by corresponding image features to LiDAR points. This method can fully use the rich texture information of images. Point-wise can effectively improve the performance compared with RoI-wise and Voxel-wise [74], but this is achieved by sacrificing memory, and Point-wise implies feature enhancement for the vast majority of LiDAR points, especially if no sampling is done.

B. Decision-Level Fusion

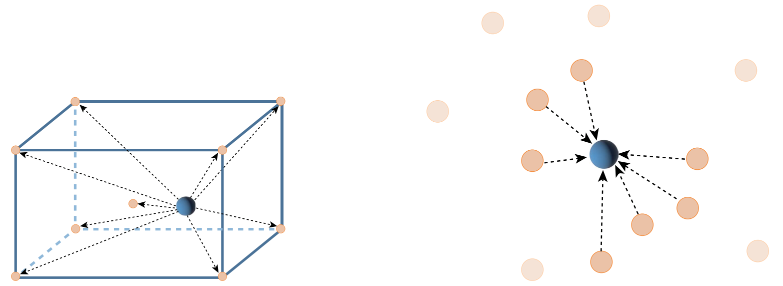
Decision-level fusion is the object detection of different data streams using single-sensor detection networks separately. Its network structure and implementation are simpler than feature-level fusion, as shown in Figure 8.

After getting the detection results of different modalities, the results are fused using the designed fusion module to adjust the single-sensor 3D proposal and generate the final 3D proposal to obtain more accurate prediction results.

Compared with feature fusion, decision-level fusion has a modular design, which makes it easy to test the designed fusion module using different detection heads. Decision-level fusion does not need to deal with the direct relationship between pixel points and LiDAR points and is less computationally intensive.

Pang et al. [87] provide a low-complexity multi-sensor fusion framework that operates before NMS (Non-maximum suppression) of candidate frames from arbitrary 2D and 3D detector outputs, exploiting their geometric consistency to produce more accurate 2D and 3D detection results.

MV3D [41] is a classical multi-sensor fusion framework that takes the bird's eye view and front view of LIDAR point clouds as well as an image as input. It first generates 3D object proposals from the point cloud's bird's-eye view, corrects the



(a) Enhance features using point-to-box distance

(b) Enhance features using the distance from a point to k-neighbor points

Fig. 7. Two common types of Point-wise feature enhancement.

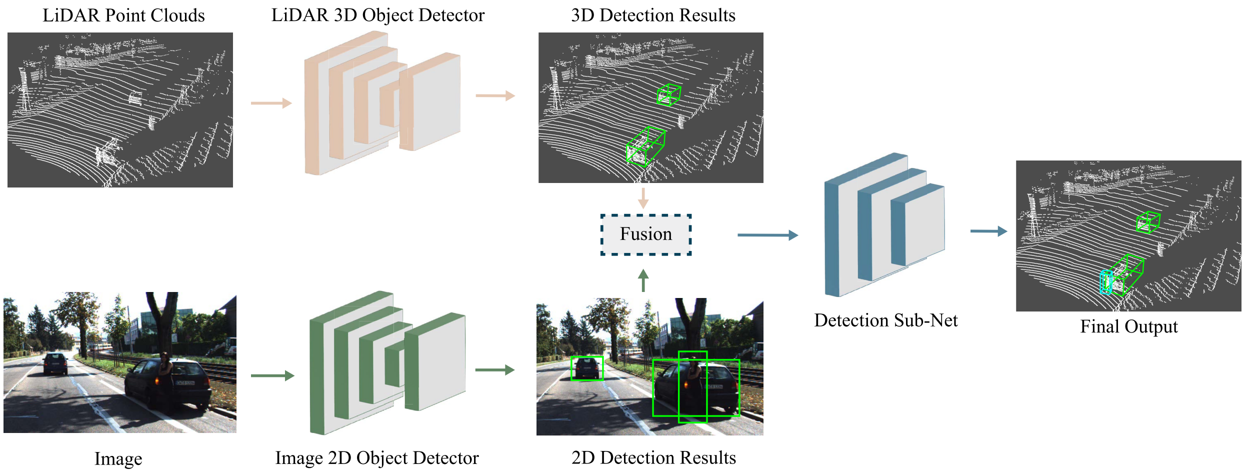


Fig. 8. Illustration of Multi-Sensor fusion 3D object detection based on decision-level fusion.

proposals using features from the other views, and finally fuses the proposals from the three views as the final prediction.

AVOD [88] is more like a combination of feature-level fusion and decision-level fusion, where BEV features from the point clouds, and camera image features are extracted separately, and the two features are fused at the feature level to produce an initial 3D proposal. The features are then used again to correct the proposal, and the fusion becomes the final prediction.

A non-negligible problem of decision-level fusion is its inability to use rich intermediate features, and its detection accuracy depends on the selected single-sensor detection network.

It is worth mentioning that most of the current experiments on algorithms related to 3D object detection are based on open-source frameworks [89], [90], which means that researchers can easily start testing using existing detection networks, which facilitates research on decision-level fusion. We believe that decision-level fusion may become a fundamental operation, both independently as a fusion solution and as an enhancement to feature-level fusion. With its parallel multi-branch structure, decision-level fusion can easily avoid the problem of fusion structures not working properly in the event of single-sensor failure, and it can combine the outputs of multiple models and multiple sensors, effectively improving robustness.

C. Summary of LiDAR-Camera Fusion Methods

Lidar and cameras can provide powerful environmental awareness and are the main sensors for self-driving cars. However, due to its own limitations, the single-sensor solution cannot accurately complete 3D object detection. The multi-sensor solution combines the advantages of each sensor to push the accuracy of 3D object detection to a new level. Specifically, cameras provide high-resolution images suitable for perceiving the appearance and texture characteristics of objects, while LiDAR provides the precise location, distance, and shape of objects. It is worth noting that the gain effect brought by camera image features is related to the degree of point cloud sparsity. When the point cloud is relatively sparse, the image features can bring greater gain; while when the point cloud itself is relatively dense, the gain effect of the image feature is not obvious.

D. Other Fusion Methods

We have described the LiDAR-camera fusion scheme in detail, below we briefly discuss other sensor fusion methods such as *radar-camera, LiDAR-radar, LiDAR-radar-camera*, etc. CenterFusion [8] uses radar and camera data for 3D object detection.

It first detects the centroids of objects on the image. It generates a heat map based on the centroids [91], and subsequently

TABLE II

COMPARISON OF SOME POPULAR MULTI-SENSOR FUSION ALGORITHMS. SOME OF THE METHODS ARE TESTED ON MULTIPLE DATASETS, AND WE ONLY LIST ONE OF THEM HERE

References	Year	Fusion location	LiDAR	Camera	Dataset	mAP
PointFusion [60]	2018	Feature	Point	Feature map	KITTI	63.00%
F-PointNet [61]	2017	Feature	Point	Feature map	KITTI	69.79%
DCAN [68]	2022	Feature	Point	Feature map	nuScenes	67.30%
EPNet [69]	2020	Feature	Point	Feature map	KITTI	81.23%
FusionRCNN [70]	2022	Feature	Point	Feature map	Waymo	70.33%
DeepInteraction [72]	2022	Feature	Point	Feature map	nuScenes	69.90%
PI-RCNN [73]	2020	Feature	Point	Mask	KITTI	71.70%
PointPainting [74]	2020	Feature	Point	Mask	KITTI	71.70%
TransFusion [76]	2022	Feature	View	Feature map	nuScenes	68.90%
BEVFusion [77]	2022	Feature	View	Feature map	nuScenes	71.70%
MVAF-Net [78]	2020	Feature	View	Feature map	KITTI	72.70%
Mvx-net [81]	2019	Feature	Voxels	Feature map	KITTI	63.22%
Voxel Field Fusion [79]	2022	Feature	Voxels	Feature map	nuScenes	68.40%
AutoAlignV2 [80]	2022	Feature	Voxels	Feature map	nuScenes	68.40%
MSMDFusion [82]	2022	Feature	Voxels	Feature map	nuScenes	71.50%
CenterPointV2 [83]	2020	Feature	Voxels	Mask	nuScenes	67.10%
CLOCs [87]	2020	Decision	-	-	KITTI	82.25%
MV3D [41]	2017	Decision	-	-	KITTI	63.63%
AVOD [88]	2018	Decision	-	-	KITTI	71.76%

uses a flat truncated head-based method to correlate radar points with the centroids of objects on the image to obtain critical depth information and finally generate predictions.

According to Kim et al. [92], radar is superior to LiDAR in terms of reliability and cost convenience; nevertheless, all of the existing radar-based fusion approaches focus on the fusion of results. Consequently, Kim et al. [92] came up with a new method for result-level early fusion that correlated image suggestions with radar points. After that, they used cross-attention-based continuous feature fusion to establish the contextual relationship between radar and camera in order to achieve robustness and attentional fusion.

The challenge of using radar is the noise and measurement ambiguity. To address this issue, Yang et al. [93] proposed the RadarNet [93] approach, which features early fusion based on voxels and later fusion based on attention mechanism.

DeepFusion [94] takes a modular design that fuses in different combinations of LiDAR, camera, and radar, with a specially designed feature extractor that can be easily switched for different device inputs, and which supports multiple combinations of LiDAR-camera, LiDAR-camera-radar, and LiDAR-radar.

V. STATE-OF-THE-ART METHODS COMPARISON

A. Model Comparison

To describe the methods in Section IV-A more intuitively, Table II makes a brief comparison of these methods to visualize the similarities and differences of LiDAR-camera-based fusion schemes.

The methods in Table II are classified overall according to fusion location and in feature-level fusion schemes with a wide range of classifiable levels, according to the feature representation in Section IV-A.

The fusion strategy design of each method is closely related to the main dataset selected, so the table summarizes the datasets selected by each method. Some of these schemes were tested on multiple datasets, and the results are shown here for only one of the datasets.

Through the comparison, we can conclude that.

- While using raw points retains complete geometry, it's computationally intensive. Final detection still hinges on fusion module design, feature enhancement, and more.
- Overall, combining image masks with LiDAR features yielded strong KITTI dataset results. Because image masks provide per-pixel semantic features and accurate representations.
- Most of the current decision-level fusion is used in combination with feature-level fusion to take full advantage of multi-sensor data and multiple models. Feature-level fusion can provide a more semantically rich feature representation, while decision-level fusion can combine the detection results from multiple sensors.

B. Visualization and Discussion

In order to visually represent the outcomes of 3D object detection and analyze the similarities and differences between multi-sensor and LiDAR-only solutions, a set of representative solutions were selected for visualization on the KITTI [45] and nuScenes [43] datasets.

First, we used Mvx-net [81] and PointPillars [51] and selected four representative scenes in the KITTI dataset for comparison, and uniformly set the threshold to 0.5. General note: the scenes in Figure 9, 10, 11, 12 are one-to-one, Figure 9 is the camera view, Figure 10 is the ground truth box, and Figure 11, 12 are the PointPillars and Mvx-net, respectively detection results. The object in the yellow circle in

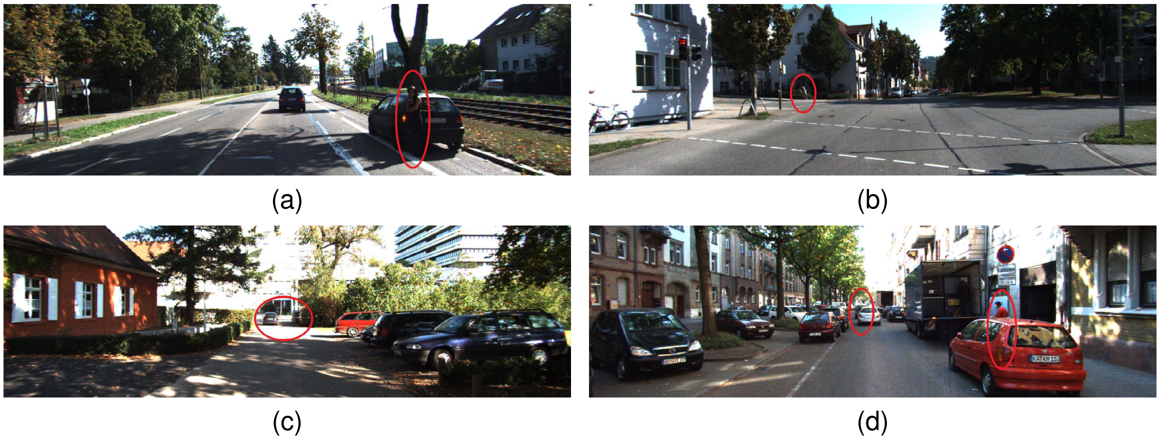


Fig. 9. Camera image from KITTI [45], the red circles are our targets of additional interest.

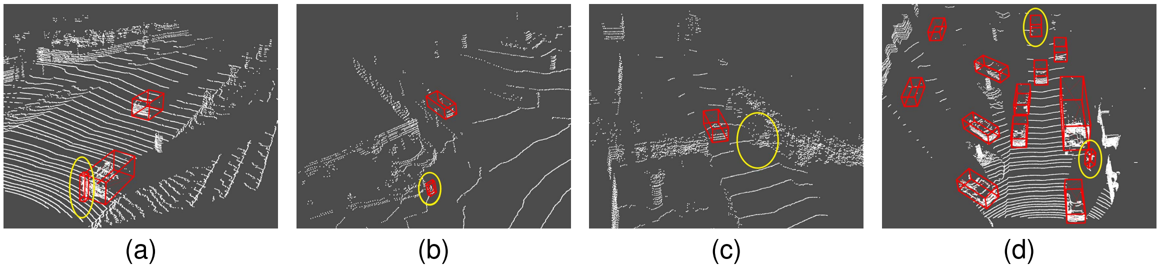


Fig. 10. Ground truth in KITTI [45] labelling.

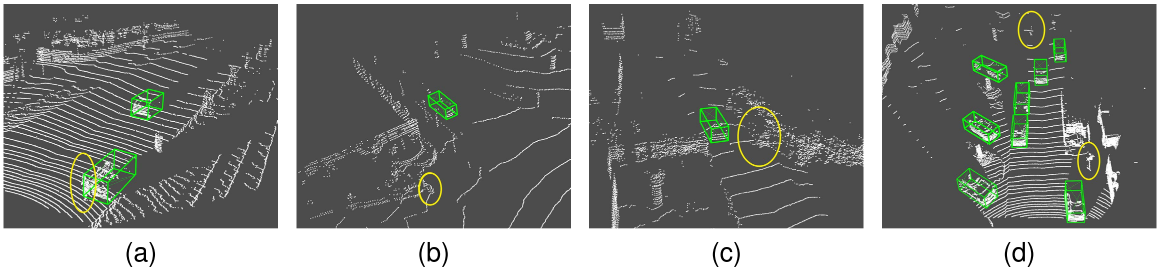


Fig. 11. Visualisation results from PointPillars [51], where the original camera image of each scene is shown in Figure 9.

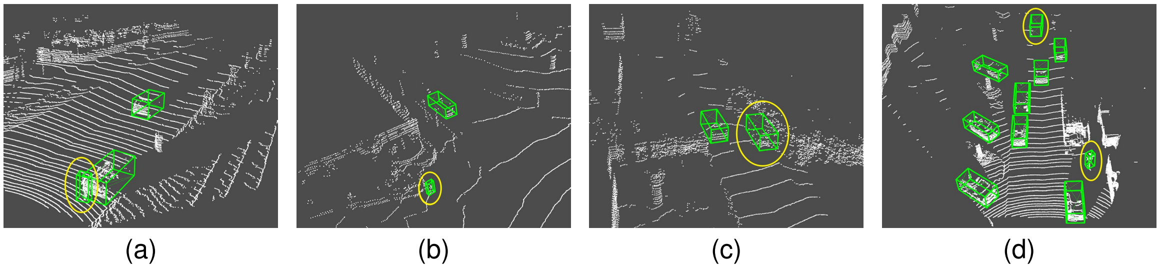


Fig. 12. Visualisation results from MVX-Net [81], where the original camera image of each scene is shown in Figure 9.

Figure 10, 11, 12 corresponds to the object in the red circle in Figure 9. The point clouds have been tilted for viewing purposes.

As shown in Figure 9a, we paid extra attention to the pedestrians within the red circles. Due to the very close proximity of the pedestrian and the vehicle behind it, it does not seem to be very easy to distinguish in point clouds space alone, which also leads to the PointPillars [51] not being successfully detected. In this case, the texture and color features from the camera acted as supplementary information to the point clouds and were able to allow successful detection by the multi-sensor scheme Mvx-net [81]. The comparative results are shown in Figure 11a, 12a.

There is a similar scene, as shown in Figure 9b. The pedestrian in the yellow circle is very close to the grass in the background, whereas in point clouds space, it is usually

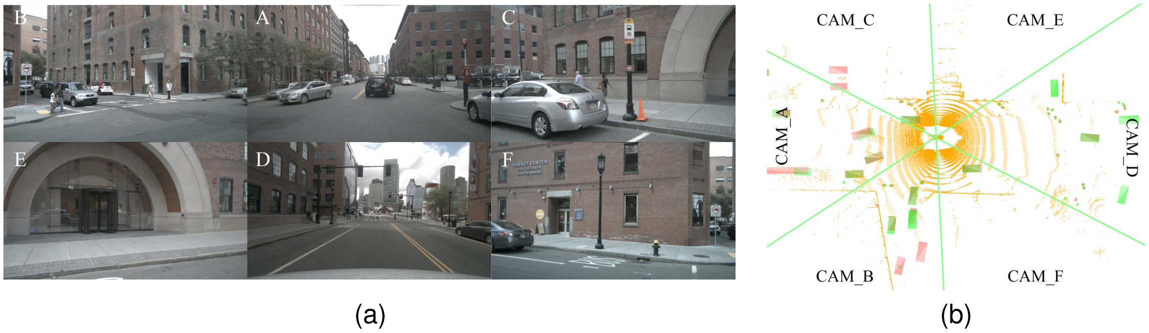


Fig. 13. (a): Image of six cameras deployed around the vehicle, image from nuScenes [43]. (b): BEV view of the 3D detection result, where the red translucent is labeled as ground truth and the green translucent is labeled as the predicted box. The areas corresponding to each of these cameras have been marked on the images.

impossible to distinguish accurately between foreground and background, which is the reason why PointPillars [51] did not identify this target. The comparative results are shown in Figure 11b, 12b.

Another problem faced by pure point cloud solutions is the difficulty of distinguishing between two objects when their point cloud information is very similar. Again, we focus on the area inside the red circle in Figure 9c. The vehicle inside the red circle and the object next to it are easily distinguishable from the camera image. However, they share a high degree of geometry in point clouds space, so this causes PointPillars [51] to incorrectly identify the clutter as a vehicle, a situation that is improved in Mvx-net [81], see Figure 11c, 12c for details of the comparison results.

Let's look at another relatively dense scene that has a large number of occlusions and truncations, as shown in Figure 9c, where we pay extra attention to the objects within the red circles. As can be seen from the comparison results in Figure 11c, 12, PointPillars [51] did not recognize any of the targets within the circles, while they were successfully detected in Mvx-net [81].

After comparing various scenarios, we conclude: 1. When the target is clear or slightly obscured, both multi-sensor and LiDAR-only methods accurately identify it; 2. For similar target point cloud shapes, multi-sensor outperforms LiDAR-only; 3. Multi-sensor excels over LiDAR-only when the target is very close to a background object.

In addition, we also show its results visualized on nuScenes [43] using BevFusion [77]. We replaced the image backbone network [95], [96] of BevFusion [77] and retrained it, and visualized some of the results. BEVFusion [77] uses LiDAR with images from six surround view cameras as input and is able to perform 3D object detection in 360°, so we show the visualization results in the BEV view, as shown in Figure 13.

As can be seen from the camera view in Figure 13a, this is a very dense scene with a lot of occlusion and truncation, and BEVFusion takes image up-dimensioning operations to map the image features into BEV space, enabling many hard-to-detect-objects-to be detected successfully, as shown in Figure 13b.

BEVFusion [77] as a representative method for 3D object detection from a BEV perspective. It maps sensor inputs into a unified BEV space in pre-processing and is applicable not only to 3D object detection but also to a variety of downstream tasks, including lane segmentation, BEV map segmentation, etc.

In addition, we visualized CenterFusion [8] for comparing LiDAR with radar, as shown in Figure 14, the number of radar points is sparser compared to the number of LiDAR points. CenterFusion [8] takes the scheme of the ROI region in the center of the heat map.

As can be seen from Figure 14, due to the characteristics of radar, radar points are not accurately projected onto the target. There are no available radar points (marked in green) on the pedestrian in the rightmost scene of Figure 14, and the radar points that should fall on the pedestrian fall on the left side of the pedestrian, which may be the projection error caused by the sensor jitter. From the result, the pedestrian is still accurately identified in the result with the correction of the camera stream.

VI. Open Challenges and Future

This section focuses on the current issues and future trends in multi-sensor fusion three-dimensional object detection.

A. Challenges

1) Data Alignment: 3D object detection based on multi-sensor fusion requires the fusion and alignment of input from different modalities [97].

The first issue is that these sensors frequently have different viewing angles; for example, due to the nature of LiDAR, it must be installed on the vehicle's top, whereas a standard camera would be mounted in front of the vehicle.

Even though the LiDAR and camera are set in exactly the same position, they have distinct viewing angles. The current common solution is to use the projection matrix to correspond LiDAR points and picture pixels one by one, and the projection matrix must be strictly calibrated according to the device parameters, device distance, and device view angle, which is a strict and time-consuming task in which errors are unavoidable.

Furthermore, the driving environment is complex, and the road surface is invariably uneven. A minor jitter might cause sensor misalignment, and if the relative sensor position changes, the calibrated projection matrix is no longer valid.

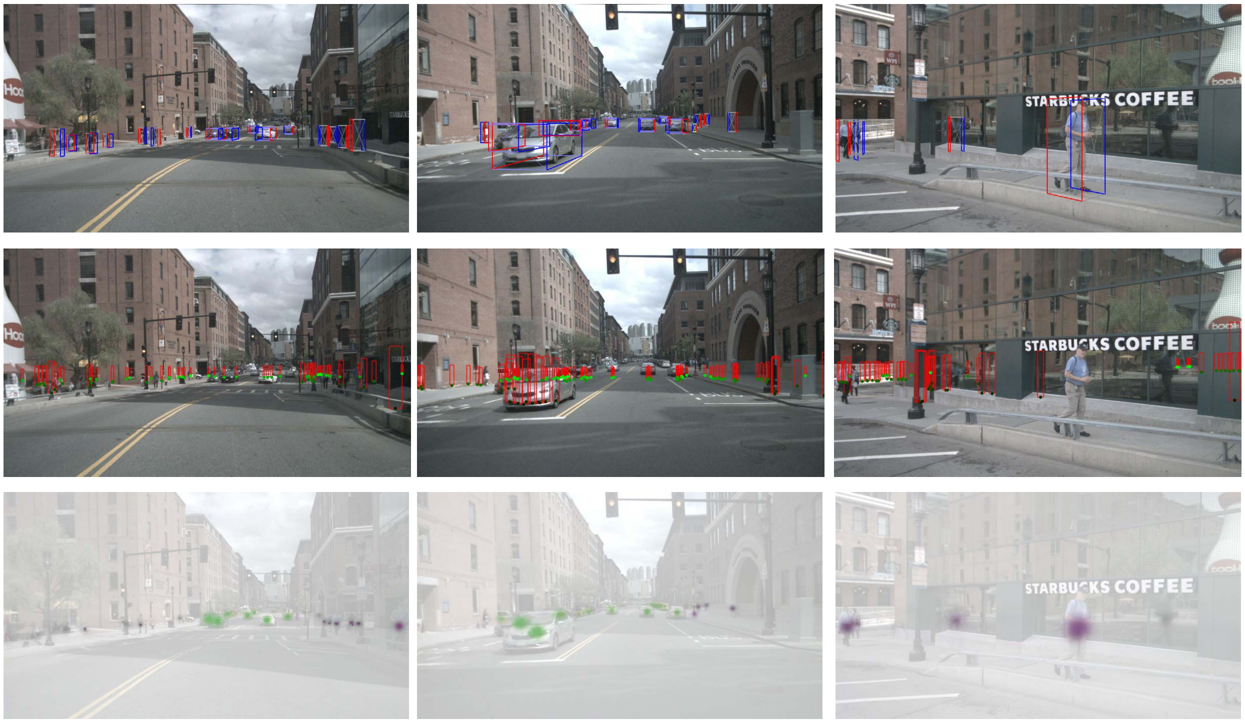


Fig. 14. CenterFusion's [8] visual output. From top to bottom, the prediction boxes on 2D images, the pillar extension visualization, and the center heat map.

One alternative approach is combining the LiDAR and the camera into a single unit. This reduces the risk of misalignment between the LiDAR and the camera to the greatest extent possible [47].

In addition, the imaging principles of the camera and the LiDAR are not the same. The camera acquires information from the real world using the "small-aperture imaging" principle, whereas the LiDAR acquires data in the real 3D world. This results in a significant difference in how the same object is represented. In addition, the dimensions of the LiDAR and the camera dimensions are different, making it challenging to combine the data from the two separate dimensions.

2) Lost Information: Another challenge is the loss of data during processing, often due to sensor differences, processing constraints, and algorithms. For instance, when projecting point clouds onto the image plane for features, the camera-LiDAR accuracy disparity, as in Figure 3b, leads to missing LiDAR projections on some pixels. Relying solely on these projected pixel features wastes image features. Also, in point cloud voxel allocation, where voxel sizes are usually not too large (e.g., [0.16,0.16,0.5] in KITTI), sampling trims computational load by discarding points, causing a loss of information. This absence of key points can reduce overall detector accuracy.

Another issue that requires more attention is the issue of sensor failure. When sensors fail and do not provide data properly, they are typically populated with zero data in order to ensure the availability of the model, but a large amount of zero data is not effective for object detection. Furthermore, self-driving vehicles can also be subject to cyber-attacks [98] or visual attacks [99], in which case the vehicle may plan the wrong path and cause a traffic accident.

We propose to introduce a dynamic selection strategy in a vehicle-grade multi-sensor fusion scheme, whereby when the computer detects abnormal input data from a sensor, it suspends the use of that sensor and places computational resources on other sensors to ensure the safety of the vehicle.

3) Data Augmentation: Data augmentation is a frequent approach in deep learning to ensure that neural networks can completely utilize current datasets, train more efficiently from limited inputs, and successfully reduce model overfitting [100].

We can efficiently perform data augmentation in detection networks that use a single modality as input information, such as scaling, rotation, flipping, object cut-and-paste, etc. For multi-sensor data augmentation, it is necessary to keep each modality synchronized for augmentation and construct mapping relationships.

Zhang et al. [101] proposed a new multi-sensor enhancement method by segmenting the point clouds and images of objects, pasting them into the scene synchronously, and then building mapping relationships based on the projection matrix. Y. Li et al. [79] build on [102] by adjusting the occlusion relationship of the copied objects to make them more realistic.

Prakash et al. [103] proposed a new method of *structured domain randomization (SDR)*, which makes the model more generalizable by randomly placing objects and distractors through the probability distribution generated by a specific problem. It is worth mentioning that Tan et al. [104] proposed a method to automatically generate realistic traffic scenes with the flexibility to insert various classes of objects in the desired scenes.

4) Datasets and Metrics: **Datasets:** The importance of datasets as the foundation of deep learning cannot be overstated. Today's popular 3D object detection datasets suffer from the following problems: incomplete categories, unbalanced categories, too few occluded and truncated objects, and too few small objects at a distance.

Despite the fact that the nuScenes [43] dataset contains 1000 scenes, these scenes are not even close to being indicative of the complex and varied driving scenarios that exist in the real world. The majority of annotations in popular 3D object detection datasets still come from cars and people, which causes the detection framework to have difficulty identifying other types of objects, such as guardrails and roadblocks.

In addition, the perspective of the onboard sensors is covered in real-world driving conditions by the majority of the objects in the environment. Due to the difficulties of labelling, the number of truncated or obscured targets is quite low among the popular datasets that are currently available.

Evaluation Metrics: It is worth noting that there are no specialized evaluation metrics for evaluating the success of a fusion scheme; rather, the mAP of the dataset is used to assess, and then the ablation experiment is utilized to establish that the fusion module is the one that accomplishes the job. As a result, a 3D object identification framework with multi-sensor fusion should be evaluated using a measure that incorporates computational fusion overhead, fusion accuracy impact, and robustness.

B. Research Trends and Future

1) Research Trends: Deep learning has achieved remarkable success in the field of object detection, and it has also been widely used in 3D object detection. In terms of multi-sensor fusion 3D object detection solutions at this stage, most approaches still focus on the design of the fusion module. We speculate that future research will continue to focus on deep learning methods and that the research will gradually move towards network structure design, proposing a backbone network specifically for 3D tasks. Additionally, large models trained through massive datasets possess significant advantages in multi-category detection tasks. However, they also require substantial computing resources and high-performance hardware for inference. With ongoing advancements in hardware technology, there is a promising outlook for the extensive application of large models in the field of autonomous driving.

In terms of data processing, converting sparse heterogeneous point clouds into ordered voxel representations can facilitate neural network handling. In addition, regions without point clouds can be very easily located based on the voxel index, and processing of the region can be skipped by strategies such as sparse convolution, thus reducing computational complexity and speeding up the network's inference. In addition to this, neighboring points in point clouds will often be in the same or immediately adjacent voxels, which helps to preserve the spatial relationships of the point clouds and build contextual information as a way to improve the algorithm's ability to perceive local and global structure. In summary, voxels have the advantages of regularised representation, simplified data processing, and ease of handling spatial relationships, and these advantages can facilitate further processing and application of point clouds data. Image data is crucial in multi-sensor fusion. Depth prediction from images gains traction, enabling applications like mapping image features to 3D space and complementing point clouds via pixel depth estimation.

2) Future: In the field of computer vision, the domain of 3D object detection is experiencing significant growth. However, there are still opportunities for further advancements. In this context, we propose several potential approaches to enhance the performance of 3D object detection systems and offer insights for future research endeavors.

- More data. In addition to LiDAR and image information, radar information is also available in some datasets, but very little work has gone into using radar information. As described in Section II-A, radar is not as high resolution as LiDAR, but radar performs more consistently in bad weather. More sensors mean a higher degree of assurance, and to improve the speed of inference, radar can be used as a backup sensor to enable radar's route of operation in the event of anomalies in the LiDAR input data.
- Time series. Frame rates often differ between sensors, and forcing the algorithm into a blocking state to synchronize different sensor data would inevitably result in a loss of inference speed. One solution uses prior frame data for reasoning, compensating for inference speed but struggles with fast-moving targets. Alternatively, in real driving scenarios, targets such as vehicles and pedestrians are constrained by the traffic topology, and information such as High-accuracy maps, lane lines, and signs predict trajectories, letting previous frame inferences apply.

VII. Conclusion

The rapid development of autonomous vehicles has led to a surge in the use of 3D object detection techniques. This study presents a survey of recent years' worth of multi-sensor fusion 3D object identification frameworks, with a particular focus on LiDAR and camera fusion approaches. First, we will quickly go through the several common sensors. The following step involves analyzing and contrasting a number of datasets that are often used for autonomous driving. We will examine in fully the LiDAR-camera fusion-based 3D object detection strategy from the fusion position so that the reader will have a better understanding of the popular multi-sensor fusion 3D object detection schemes. After that, we will provide a quick overview of the fusion strategies for the remaining sensors. At this point, we explore the challenges in multi-sensor fusion 3D object identification and the emerging developments in this field.

References

[1] J. Guo, U. Kurup, and M. Shah, "Is it safe to drive? An overview of factors, metrics, and datasets for driveability assessment in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3135–3151, Aug. 2020. [Online]. Available: https://sites.google.com/view/driveability-survey-

[2] Y. Wang et al., "Multi-modal 3D object detection in autonomous driving: A survey," 2021, *arXiv:2106.12735*.

[3] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *IEEE Access*, vol. 8, pp. 2847–2868, 2020.

[4] J. Mao, S. Shi, X. Wang, and H. Li, "3D object detection for autonomous driving: A comprehensive survey," 2022, *arXiv:2206.09474*.

[5] R. Qian, X. Lai, and X. Li, "3D object detection for autonomous driving: A survey," *Pattern Recognit.*, vol. 130, Oct. 2022, Art. no. 108796.

[6] H. Wang et al., "RBGNet: Ray-based grouping for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1100–1109.

[7] Y. Duan, C. Zhu, Y. Lan, R. Yi, X. Liu, and K. Xu, "DisARM: Displacement aware relation module for 3D detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16959–16968.

[8] R. Nabati and H. Qi, "CenterFusion: Center-based radar and camera fusion for 3D object detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1526–1535.

[9] O. Liba et al., "Handheld mobile photography in very low light," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–164, 2019.

[10] N. S. Zewge, Y. Kim, J. Kim, and J.-H. Kim, "Millimeter-wave radar and RGB-D camera sensor fusion for real-time people detection and tracking," in *Proc. 7th Int. Conf. Robot Intell. Technol. Appl. (RiTA)*, Nov. 2019, pp. 93–98.

[11] S. Lee, "Deep learning on radar centric 3D object detection," 2020, *arXiv:2003.00851*.

[12] K. Hermann, T. Chen, and S. Kornblith, "The origins and prevalence of texture bias in convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 19000–19015.

[13] Z. Liu, D. Zhou, F. Lu, J. Fang, and L. Zhang, "AutoShape: Real-time shape-aware monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15621–15630.

[14] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8551–8560.

[15] X. Ma et al., "Delving into localization errors for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4719–4728.

[16] T. Wang, Z. Xinge, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," in *Proc. Conf. Robot Learn.*, 2022, pp. 1475–1485.

[17] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5632–5640.

[18] Z. Qin, J. Wang, and Y. Lu, "MonoGRNet: A geometric reasoning network for monocular 3D object localization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 8851–8858.

[19] A. Simonelli, S. R. Bulò, L. Porzi, E. Ricci, and P. Kontschieder, "Towards generalization across depth for monocular 3D object detection," 2019, *arXiv:1912.08035*. [Online]. Available: https://research.mapillary.com

[20] I. Cortés, J. Beltrán, A. de la Escalera, and F. García, "siaNMS: Non-maximum suppression with Siamese networks for multi-camera 3D object detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct./Nov. 2020, pp. 933–938.

[21] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3069–3087, Nov. 2021.

[22] A. Meuleman, H. Kim, J. Tompkin, M. H. Kim, "FloatingFusion: Depth from ToF and image-stabilized stereo cameras," in *Computer Vision—ECCV 2022* (Lecture Notes in Computer Science), vol. 13661, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham, Switzerland: Springer, 2022, doi: 10.1007/978-3-031-19769-7_35.

[23] X. Lin, J. Wang, and C. Lin, "Research on 3D reconstruction in binocular stereo vision based on feature point matching method," in *Proc. IEEE 3rd Int. Conf. Inf. Syst. Comput. Aided Educ. (ICISCAE)*, Sep. 2020, pp. 551–556.

[24] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," 2018, *arXiv:1803.08669*.

[25] N. Wadhwa et al., "Synthetic depth-of-field with a single-camera mobile phone," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–13, Aug. 2018.

[26] W. Yan et al., "Collaborative structure and feature learning for multi-view clustering," *Inf. Fusion*, vol. 98, Oct. 2023, Art. no. 101832.

[27] X. Peng, X. Zhu, T. Wang, and Y. Ma, "SIDE: Center-based stereo 3D detector with structure-aware instance depth estimation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 225–234.

[28] X. Guo, S. Shi, X. Wang, and H. Li, "LIGA-stereo: Learning LiDAR geometry aware representations for stereo-based 3D detector," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3133–3143.

[29] R. Qian et al., "End-to-end pseudo-LiDAR for image-based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5880–5889.

[30] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6602–6611.

[31] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[32] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: Sparse-to-dense 3D object detector for point cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1951–1960.

[33] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11037–11045.

[34] W. Zheng, W. Tang, S. Chen, L. Jiang, and C.-W. Fu, "CIA-SSD: Confident IoU-aware single-stage object detector from point cloud," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 4, pp. 3555–3562.

[35] J. Noh, S. Lee, and B. Ham, "HVPR: Hybrid voxel-point representation for single-stage 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14600–14609.

[36] D. Huang et al., "Rethinking dimensionality reduction in grid-based 3D object detection," 2022, *arXiv:2209.09464*.

[37] D. Ye et al., "LidarMultiNet: Towards a unified multi-task network for LiDAR perception," 2022, *arXiv:2209.09385*.

[38] P. Huang, M. Cheng, Y. Chen, H. Luo, C. Wang, and J. Li, "Traffic sign occlusion detection using mobile laser scanning point clouds," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2364–2376, Sep. 2017.

[39] Y. Chen, S. Liu, X. Shen, and J. Jia, "DSGN: Deep stereo geometry network for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12536–12545.

[40] S. Kim, H. Kim, W. Yoo, and K. Huh, "Sensor fusion algorithm design in detecting vehicles using laser scanner and stereo vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1072–1084, Apr. 2016.

[41] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6526–6534.

[42] H. Zhang, D. Yang, E. Yurtsever, K. A. Redmill, and Ü. Özgüner, "Faraway-frustum: Dealing with LiDAR sparsity for 3D object detection using fusion," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 2646–2652.

[43] H. Caesar et al., "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11618–11628.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[45] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[46] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2443–2451.

[47] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.

[48] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, "The H3D dataset for full-surround 3D multi-object detection and tracking in crowded urban scenes," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9552–9557.

[49] M.-F. Chang et al., "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8740–8749.

[50] H. Yu et al., "DAIR-V2X: A large-scale dataset for vehicle-infrastructure cooperative 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 21329–21338.

[51] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12689–12697.

[52] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.

[53] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7652–7660.

[54] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. De La Escalera, "BirdNet: A 3D object detection framework from LiDAR information," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3517–3523.

[55] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2647–2664, Aug. 2021. [Online]. Available: https://github.com/sshaoshuai/PointCloudDet3D

[56] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3D architectures with sparse point-voxel convolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Glasgow, U.K.: Springer, Aug. 2020, pp. 685–702.

[57] H. Rashed, M. Essam, M. Mohamed, A. Ei Sallab, and S. Yogamani, "BEV-MODNet: Monocular camera based bird's eye view moving object detection for autonomous driving," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 1503–1508.

[58] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11135–11144.

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[60] D. Xu, D. Anguelov, and A. Jain, "PointFusion: Deep sensor fusion for 3D bounding box estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 244–253.

[61] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 918–927.

[62] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, vol. 30, Dec. 2017, pp. 5099–5108.

[63] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[64] Z. Dai, B. Cai, Y. Lin, and J. Chen, "UP-DETR: Unsupervised pre-training for object detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1601–1610.

[65] K. J. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards open world object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5826–5836.

[66] J. Wang, L. Song, Z. Li, H. Sun, J. Sun, and N. Zheng, "End-to-end object detection with fully convolutional network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15844–15853.

[67] V. S. Vibashan, V. Gupta, P. Oza, V. A. Sindagi, and V. M. Patel, "MeGA-CDA: Memory guided attention for category-aware unsupervised domain adaptive object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4514–4524.

[68] R. Wan, S. Xu, W. Wu, X. Zou, and T. Cao, "From one to many: Dynamic cross attention networks for LiDAR and camera fusion," 2022, *arXiv:2209.12254*.

[69] T. Huang, Z. Liu, X. Chen, and X. Bai, "EPNet: Enhancing point features with image semantics for 3D object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Glasgow, U.K.: Springer, Aug. 2020, pp. 35–52.

[70] X. Xu, S. Dong, L. Ding, J. Wang, T. Xu, and J. Li, "FusionRCNN: LiDAR-camera fusion for two-stage 3D object detection," 2022, *arXiv:2209.10733*.

[71] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, Oct. 2018.

[72] Z. Yang, J. Chen, Z. Miao, W. Li, X. Zhu, and L. Zhang, "DeepInteraction: 3D object detection via modality interaction," 2022, *arXiv:2208.11112*.

[73] L. Xie et al., "PI-RCNN: An efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12460–12467.

[74] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "PointPainting: Sequential fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4603–4611.

[75] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.

[76] X. Bai et al., "TransFusion: Robust LiDAR-camera fusion for 3D object detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1080–1089.

[77] T. Liang et al., "BEVFusion: A simple and robust LiDAR-camera fusion framework," 2022, *arXiv:2205.13790*.

[78] G. Wang, B. Tian, Y. Zhang, L. Chen, D. Cao, and J. Wu, "Multi-view adaptive fusion network for 3D object detection," 2020, *arXiv:2011.00652*.

[79] Y. Li et al., "Voxel field fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1110–1119.

[80] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, and F. Zhao, "AutoAlignV2: Deformable feature aggregation for dynamic multimodal 3D object detection," 2022, *arXiv:2207.10316*.

[81] V. A. Sindagi, Y. Zhou, and O. Tuzel, "MVX-Net: Multimodal Voxelnet for 3D object detection," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 7276–7282.

[82] Y. Jiao, Z. Jie, S. Chen, J. Chen, L. Ma, and Y.-G. Jiang, "MSMDFusion: Fusing LiDAR and camera at multiple scales with multi-depth seeds for 3D object detection," 2022, *arXiv:2209.03102*.

[83] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11779–11788.

[84] Z. Wang and K. Jia, "Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1742–1749.

[85] H. Shenga et al., "Improving 3D object detection with channel-wise transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2723–2732.

[86] Z. Li, F. Wang, and N. Wang, "LiDAR R-CNN: An efficient and universal 3D object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7542–7551.

[87] S. Pang, D. Morris, and H. Radha, "CLOCs: Camera-LiDAR object candidates fusion for 3D object detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10386–10393.

[88] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–8.

[89] MMDetection3D Contributors. (2020). *MMDetection3D: OpenMMLab Next-Generation Platform for General 3D Object Detection*. [Online]. Available: https://github.com/open-mmlab/mmdetection3d

[90] OpenPCDet Development Team. (2020). *OpenPCDet: An Open-Source Toolbox for 3D Object Detection From Point Clouds*. [Online]. Available: https://github.com/open-mmlab/OpenPCDet

[91] Q. Chen, L. Sun, Z. Wang, K. Jia, and A. Yuille, "Object as hotspots: An anchor-free 3D object detection approach via firing of hotspots," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Glasgow, U.K.: Springer, Aug. 2020, pp. 68–84.

[92] Y. Kim, S. Kim, J. W. Choi, and D. Kum, "CRAFT: Camera-radar 3D object detection with spatio-contextual fusion transformer," 2022, *arXiv:2209.06535*.

[93] B. Yang, R. Guo, M. Liang, S. Casas, and R. Urtasun, "RadarNet: Exploiting radar for robust perception of dynamic objects," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Glasgow, U.K.: Springer, Aug. 2020, pp. 496–512.

[94] F. Drews, D. Feng, F. Faion, L. Rosenbaum, M. Ulrich, and C. Gläser, "DeepFusion: A robust and modular 3D object detector for LiDARs, cameras and radars," 2022, *arXiv:2209.12729*.

[95] T. Liang et al., "CBNet: A composite backbone network architecture for object detection," *IEEE Trans. Image Process.*, vol. 31, pp. 6893–6906, 2022.

[96] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11963–11975.

[97] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "RegNet: Multimodal sensor registration using deep neural networks," 2017, *arXiv:1707.03167*.

[98] Y. Cao et al., "Invisible for both camera and LiDAR: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 176–194.

[99] Z. Zhu et al., "Understanding the robustness of 3D object detection with bird's-eye-view representations in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21600–21610.

[100] J. Yoo, N. Ahn, and K.-A. Sohn, "Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8372–8381.

[101] W. Zhang, Z. Wang, and C. C. Loy, "Exploring data augmentation for multi-modality 3D object detection," 2020, *arXiv:2012.12741*.

[102] C. Wang, C. Ma, M. Zhu, and X. Yang, "PointAugmenting: Cross-modal augmentation for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11789–11798.

[103] A. Prakash et al., "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 7249–7255.

[104] S. Tan, K. Wong, S. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "SceneGen: Learning to generate realistic traffic scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 892–901.

Xuan Wang (Member, IEEE) was born in Weihai, Shandong, China, in 1991. She received the B.S. and Ph.D. degrees from Traffic Information Engineering & Control, Chang'an University, China, in 2013 and 2018, respectively. She is currently a Lecturer with the School of Computer Science and Control Engineering, Yantai University. Her research interests include intelligent traffic control, artificial intelligence, and computer vision.

Kaiqiang Li was born in Weifang, Shandong, China, in 1999. He received the B.S. degree in computer science and technology from Yantai University in 2022, where he is currently pursuing the master's degree. His research interests include multi-sensor fusion 3D object detection and scene understanding for autonomous driving.

Abdellah Chehri (Senior Member, IEEE) is an Associate Professor at the Royal Military College of Canada (RMC), Kingston, ON. Before joining RMC, he was an Associate Professor at the University of Quebec (UQAC). He has an Affiliate Professor at the University of Quebec UQO, UQAT, and an Adjunct Professor at the University of Ottawa. He is a member of the IEEE Communication Society (ComSoc), the IEEE Vehicular Technology Society (VTS), the IEEE Photonics Society, the IEEE Public Safety Transportation Committee Co-Chair, and the IEEE Canadian Humanitarian Initiatives Committee. He has served as guest/associate editor for several well-reputed journals.