A Survey on 3D Object Detection Methods for Autonomous Driving Applications

Eduardo Arnold, Omar Y. Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis

***Abstract*— An autonomous vehicle (AV) requires an accurate perception of its surrounding environment to operate reliably. The perception system of an AV, which normally employs machine learning (e.g., deep learning), transforms sensory data into semantic information that enables autonomous driving. Object detection is a fundamental function of this perception system, which has been tackled by several works, most of them using 2D detection methods. However, the 2D methods do not provide depth information, which is required for driving tasks, such as path planning, collision avoidance, and so on. Alternatively, the 3D object detection methods introduce a third dimension that reveals more detailed object's size and location information. Nonetheless, the detection accuracy of such methods needs to be improved. To the best of our knowledge, this is the first survey on 3D object detection methods used for autonomous driving applications. This paper presents an overview of 3D object detection methods and prevalently used sensors and datasets in AVs. It then discusses and categorizes the recent works based on sensors modalities into monocular, point cloud-based, and fusion methods. We then summarize the results of the surveyed works and identify the research gaps and future research directions.**

***Index Terms*— Machine learning, deep learning, computer vision, object detection, autonomous vehicles, intelligent vehicles.**

I. Introduction

BETWEEN the years 2016 and 2017, the number of road casualties in the U.K. was approximately 174,510, of which 27,010 were killed or severely injured casualties [1]. As reported by the U.S. Department of Transportation, more than 90% of car crashes in the U.S. are attributed to drivers' errors [2]. The adoption of connected and autonomous vehicles is expected to improve driving safety, traffic flow and efficiency [3]. However, for an autonomous vehicle to operate safely, an accurate environment perception and awareness is fundamental.

The perception system of an Autonomous Vehicle (AV) transforms sensory data into semantic information, such as identification and recognition of road agents (*e.g.*, vehicles, pedestrians, cyclists, *etc.*) positions, velocity and class; lane marking; drivable areas and traffic signs information. Notably, the object detection task is of fundamental importance, as failing to identify and recognize road agents might lead to safety-related incidents. For instance, failing in detecting a leading vehicle can result in traffic accidents, threatening human lives [4].

One factor for failure in the perception system arises from sensors limitations and environment variations such as lighting and weather conditions. Other challenges include generalization across driving domains such as motorways, rural and urban areas. While motorways have well-structured lanes with vehicles following a standard orientation, urban areas exhibit vehicles parked at no particular orientation, more diverse classes such as pedestrians, cyclists, and background clutter such as bollards and bins. Another factor is occlusion, when one object blocks the view of another, resulting in partial or complete invisibility of the object. Not only objects' sizes can be very dissimilar, *e.g.*, comparing a truck with a dog, but objects can be very close or far away from the subject AV. The object's scale dramatically affects the sensors' readings, resulting in very dissimilar representations for the objects of the same class.

Despite the aforementioned challenges, the performance of 2D object detection methods for autonomous driving has greatly improved, achieving an Average Precision (AP) of more than 90% on the well established "KITTI" object detection benchmark [5]. While 2D methods detect objects on the image plane, their 3D counterpart introduce a third dimension to the localization and size regression, revealing depth information in world coordinates. However, the performance gap between 2D and 3D methods in the context of AVs is still significant [6]. Further research should be conducted to fill the performance gap of 3D methods, as 3D scene understanding is crucial for driving tasks. A comparison between 2D and 3D detection methods is presented in Table I.

In previous work, Ranft and Stiller [7] reviewed machine vision methods for different tasks of intelligent vehicles, including localization and mapping, driving scene understanding and object classification. In [8], on-road object detection was briefly reviewed among other perception functions, however, authors predominantly considered 2D object detection. Mukhtar *et al.* [9] reviewed 2D vehicle detection methods for Driver Assistance Systems with focus on motion and appearance-based approaches using a traditional pipeline. A traditional pipeline consists of segmentation

Manuscript received July 5, 2018; revised October 30, 2018 and December 13, 2018; accepted December 14, 2018. This work was supported in part by Jaguar Land Rover and in part by EPSRC U.K. through the Towards Autonomy: Smart and Connected Control Program under Grant EP/N01300X/1. The Associate Editor for this paper was S.-H. Kong. *(Corresponding author: Eduardo Arnold.)*

E. Arnold, O. Y. Al-Jarrah, and M. Dianati are with the Warwick Manufacturing Group, The University of Warwick, Coventry CV4 7AL, U.K. (e-mail: e.arnold@warwick.ac.uk; omar.al-jarrah@warwick.ac.uk; m.dianati@warwick.ac.uk).

S. Fallah is with the Centre for Automotive Engineering, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: s.fallah@surrey.ac.uk).

D. Oxtoby and A. Mouzakitis are with Jaguar Land Rover, Ltd., Coventry CV4 7HS, U.K. (e-mail: doxtoby@jaguarlandrover.com).

Digital Object Identifier 10.1109/TITS.2019.2892405

TABLE I
2D VERSUS 3D OBJECT DETECTION

	Advantages	Disadvantages
2D Object Detection	Well established datasets and detection architectures. Usually RGB only input can achieve accurate results in the image plane.	Limited information: lack of object's pose, occlusion and 3D position information.
3D Object Detection	3D bounding box provides object size and position in world coordinates. These detailed information allows better environment understanding.	Requires depth estimation for precise localization. Extra dimension regression increases model complexity. Scarse 3D labelled datasets.

(*e.g.,* graph-based segmentation [10] and voxel-based clustering methods [11]), hand-engineered feature extraction (*e.g.,* voxel's probabilistic features [11]) and classification stages (*e.g.,* a mixture of bag-of-words classifiers [12]).

Unlike traditional pipelines, which optimize each stage individually, end-to-end pipelines optimize the overall pipeline performance. An end-to-end detection method leverages learning algorithms to propose regions of interest and extract features from the data. The shift towards representation learning and end-to-end detection was possible by using deep learning methods, such as deep convolutional networks, which showed a significant performance gain in different applications [13], [14]. In this paper we focus on end-to-end pipelines and learning approaches, since these have become the state-of-the-art for 3D object detection and have rapidly progressed in recent years.

This paper presents an overview of 3D object detection methods and prevalently used sensors and datasets in AVs. We discuss and categorize existing works based on sensor modality into: monocular-based methods, point cloud-based methods and fusion methods. Finally, we discuss current research challenges and future research directions. The contributions of this paper are as follows:

- summarizing datasets and simulation tools used to evaluate the performance of detection models
- providing a summary of 3D object detection advancements for autonomous driving vehicles
- comparing 3D object detection methods performances on a baseline benchmark
- identifying research gaps and future research directions.

This paper is structured as follows. Section II describes commonly used sensors for perception tasks in autonomous vehicles. Section III lists well-referenced datasets used for object detection in AVs. We review 3D object detection methods in Section IV. Section V compares the performance of existing methods on a benchmark dataset and highlights research challenges and potential research opportunities. Section VI provides a brief summary and concludes this work.

II. SENSORS

Although humans primarily use their visual and auditory systems while driving, artificial perception methods rely on multiple modalities to overcome shortcomings of individual sensors. There are a wide range of sensors used by autonomous vehicles: passive ones, such as monocular and stereo cameras, and active ones, including lidar, radar and sonar. Since most research on perception for AVs focus on cameras and lidars, these two categories are described in higher detail. A more comprehensive report on current sensors for AV applications can be found in [15] and [16].

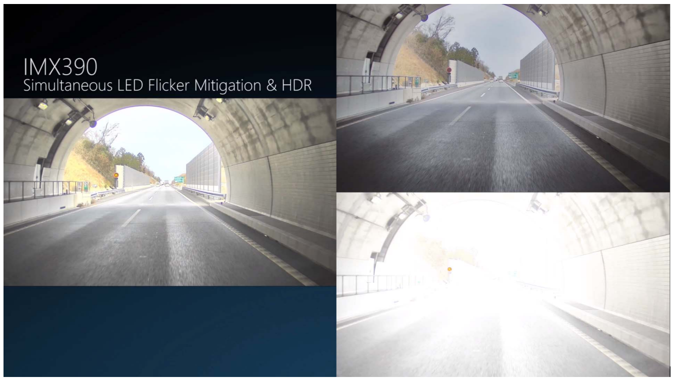

Fig. 1. IMX390 sensor sample image on a tunnel exit. The image on the left was taken with both LED flickering mitigation and High-Dynamic-Ranging (HDR) capability enabled. The top right image shows HDR functionality without LED flickering mitigation – note that the traffic sign velocity indicator does not appear. The bottom right image shows the image without any of the functionalities enabled, clearly showing the sensor capabilities. Image obtained from the Sony website [21].

A. Cameras

Monocular cameras provide detailed information in the form of pixel intensities, which at a bigger scale reveal shape and texture properties. The shape and texture information can be used to detect lane geometry, traffic signs [17] and the object class [7].

One disadvantage of monocular cameras is the lack of depth information, which is required for accurate object size and position estimation. A stereo camera setup can be used to recover depth channels. Such configuration uses matching algorithms to find correspondences in both images and calculate the depth of each point relative to the camera, demanding more processing power [18].

Other camera modalities that offer depth estimation are Time-of-Flight (ToF) cameras where depth is inferred by measuring the delay between emitting and receiving modulated infrared pulses [19]. This technology has been applied for vehicle safety applications [20], but despite the lower integration price and computational complexity has low resolution when compared to stereo cameras.

Camera sensors are susceptible to light and weather conditions. Examples range from low luminosity at night-time to extreme brightness disparity when entering or leaving tunnels. The recent use of LEDs on traffic signs and vehicles brake lights creates a flickering problem. It happens as the camera sensor cannot reliably capture the emitted light due to the LEDs' switching behavior. Sony has recently announced a new camera technology designed to mitigate flickering effects and enhance colors dynamic range [21], as illustrated in Figure 1. Additionally, image degradation can occur due to

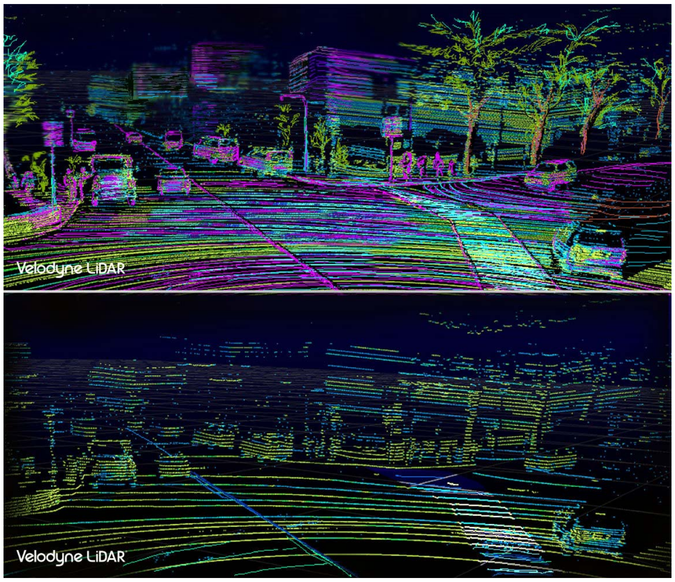

Fig. 2. The two images show the point clouds obtained by two lidar sensors on the same scene. The top image was captured using the newer VLS-128 model while the bottom one used the standard HDL-64 model. Image obtained from [24].

TABLE II
SENSORS COMPARISON

	Advantages	**Disadvantages**
Monocular Camera	Readily available and inexpensive. Multiple specifications available.	Prone to adverse light and weather conditions. No depth information provided.
Stereo Camera	Higher point density when compared to lidar. Provides dense depth map.	Depth estimation is computationally expensive. Poor performance with textureless regions or during night-time. Limited Field-of-View (FoV).
Lidar	360 degrees FoV, precise distance measurements. Not affected by light conditions.	Raw point cloud does not provide texture information. Expensive and large equipment.
Solid-State lidar	No moving mechanical parts, compact size. Large scale production should reduce final cost.	Limited FoV when compared to mechanical scanning lidar. Still under development.

rainy or snowy weather. Chen *et al.* [22] propose to mitigate this using a de-raining filter based on a multi-scale pyramid structure and conditional generative adversarial networks.

B. Lidar

Lidar sensors emit laser beams and measure the time between emitting and detecting the pulse back. The timing information determines the distance of obstacles in any given direction. The sensor readings result in a set of 3D points, also called Point Cloud (PCL), and corresponding reflectance values representing the strength of the received pulses. Unlike images, point clouds are sparse: the samples are not uniformly distributed in space. As active sensors, external illumination is not required and thus more reliable detection can be achieved considering adverse weather and extreme lighting conditions (*e.g.,* night-time or sun glare scenarios).

Standard lidar models, such as the HDL-64L [23], use an array of rotating laser beams to obtain 3D point clouds in 360 degrees and up to 120m radius. This sensor can output 120 thousand points per frame, which amounts to 1,200 million points per second on a 10 Hz frame rate. Velodyne recently announced the VLS-128 model [24] featuring 128 laser beams, higher angular resolution and 300m radius range. Figure 2 shows a comparison between the point densities of the two models. The announcement suggests that the increased point density might enhance the recall of methods using this modality but challenges real time processing performance. The primary challenge to the widespread use of lidar is its price: a single sensor can cost more than $70,000. Nevertheless, this price is expected to decrease in the following years with the introduction of solid state lidar technology [25] and large scale production.

Some methods rely on both lidar and camera modalities. Before fusing these modalities it is required to calibrate the sensors to obtain a single spatial frame of reference. Park *et al.* [26] propose to use polygonal planar boards as targets that can be detected by both modalities to generate accurate 3D-2D correspondences and obtain a more accurate calibration. However, having spatial targets makes this method laborious for on-site calibration. As an alternative, Ishikawa *et al.* [27] devised a calibration method without spatial targets using odometry estimation of the sensors w.r.t. the environment to iteratively calibrate them.

C. Discussion

Monocular cameras are inexpensive sensors, but they lack depth information which is required for accurate 3D object detection. Depth cameras can be used for depth recovery, but fail in adverse lighting conditions and textureless scenes and ToF camera sensors have limited resolution. In contrast, lidar sensors can be used for accurate depth estimation during night-time, but is prone to noise during adverse weather, such as snow and fog, and cannot provide texture information. We summarize the advantages and disadvantages of each sensor modality in Table II.

III. DATASETS

As learning approaches become widely used the need of training data also increases. The availability of large scale image datasets such as ImageNet [28] allowed fast development and evolution of image classification and object detection models. The same phenomena occurs in the driving scenario, where more data means broader scenario coverage. In particular, tasks such as object detection and semantic segmentation require finely labelled data. In this section we present common datasets for driving tasks, specifically to object detection.

One of the most used datasets in the driving context is KITTI [29], which provides stereo color images, lidar point clouds and GPS coordinates, all synchronized in time. Recorded scenes range from well-structured highways, complex urban areas and narrow countryside roads. The dataset can

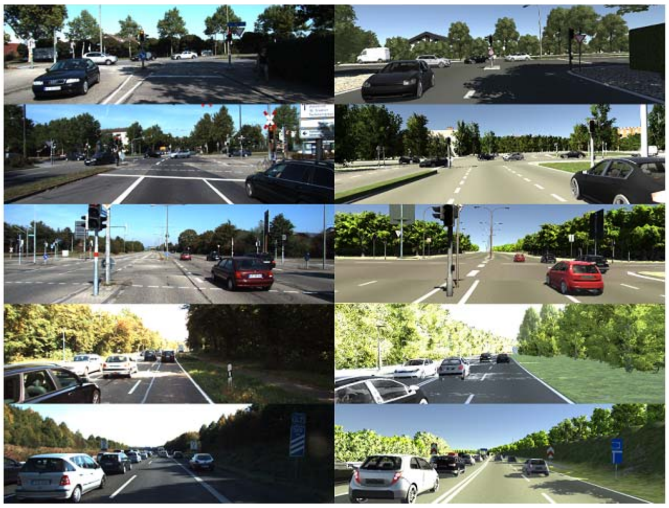

Fig. 3. Frames from 5 real KITTI videos (first column) and respective virtual clones on Virtual KITTI (second column). Image from [31].

be used for multiple tasks: stereo matching, visual odometry, 3D tracking and 3D object detection. In particular, the specific object detection dataset contains 7,481 training and 7,518 test frames, which are provided with sensor calibration information and annotated 3D boxes around objects of interest. The annotations are categorized in "easy, moderate and hard" cases, according to object size, occlusion and truncation levels.

Despite widely adopted, this dataset has several limitations. Notably, limited sensor configuration and lighting conditions: all the measurements were obtained by the same set of sensors during daytime and mostly under sunny conditions. In addition the classes frequency is highly unbalanced [30] – 75% car, 4% cyclist and 15% pedestrians. Furthermore, most scene objects follow a predominant orientation, facing the ego-vehicle. The lack of variety challenges the evaluation of current methods in more general scenarios, reducing their reliability for real-world applications.

Considering these limitations and the expensive process of obtaining and labeling a dataset, Gaidon *et al.* [31] proposed the Virtual KITTI dataset. The authors manually recreated the KITTI environment using a game-engine, 3D model assets and the original video sequences, see Figure 3. Different lighting and weather conditions, vehicles colors and models, *etc.*, were adjusted to automatically generate labelled data. They provide approximately 17,000 frames consisting of the photo-realistic images, a depth frame, and pixel-level semantic segmentation ground-truth. Additionally, the authors assessed the transferability across real and virtual domains for a tracking application (which requires detection). They evaluated a tracker trained on real images and tested on virtual ones. The results revealed that the gap in performance is minimal, showing the equivalence of the datasets. They also concluded that the best performance was obtained when training on the virtual data and fine-tuning on real data.

Simulation tools can be used to both generate training data on specific conditions or to train end-to-end driving systems [32], [33]. Using virtual data during training can enhance the performance of detection models on real environments. This data can be obtained through game-engines [34] or simulated environments [31]. CARLA [35] is an open-source simulation tool for autonomous driving that allows flexible environmental setup and sensor configuration. It provides several 3D models for pedestrians, cars and includes two virtual towns. Environmental conditions, such as weather and lighting, can be adjusted to generate unseen scenarios. The virtual sensor suite includes RGB and depth cameras with ground-truth segmentation frames and a ray-casting lidar model. Another simulation tool, Sim4CV [36] allows easy environment customization and simultaneous multi-view rendering of the driving scenes, while providing ground-truth bounding boxes for object detection purposes.

IV. 3D Object Detection Methods

We divide 3D object detection methods in three categories: monocular image, point cloud and fusion based methods. An overview of methodology, advantages and limitations for these methods is provided in Table III. The following subsections address each category individually.

A. Monocular Image Based Methods

Although 2D object detection is a largely addressed task that has been successfully tackled in several datasets [37], [38], the KITTI dataset offers particular settings that pose challenges to object detection. These settings, common to most driving environments, include small, occluded or truncated objects and highly saturated areas or shadows. Furthermore, 2D detection on the image plane is not enough for reliable driving systems: more accurate 3D space localization and size estimation is required for such application. This section focuses on methods that are able to estimate 3D bounding boxes based only on monocular images. Since no depth information is available, most approaches first detect 2D candidates before predicting a 3D bounding box that contains the object using neural networks [39], geometrical constraints [40] or 3D model matching [41], [42].

Chen *et al.* [39] propose Mono3D, which leverages a simple region proposal algorithm using context, semantics, hand-engineered shape features and location priors. For any given proposal, these features can be efficiently computed and scored by an energy model. Proposals are generated by exhaustive search on 3D space and filtered with Non-Maxima Suppression (NMS). The proposals are further scored by a Fast R-CNN [37] model that regresses 3D bounding boxes. The work builds upon the authors' previous work 3DOP [43], which considers depth images to generate proposals in a similar framework. Despite using only monocular images, the Mono3D model slightly improves the performance obtained by [43], which uses depth images. Pham and Jeon [44] extends the 3DOP proposal generation considering class-independent proposals, then re-ranks the proposals using both monocular images and depth maps. Their method outperforms both 3DOP and Mono3D methods, despite using depth images to refine proposals.

An important characteristic of driving environments is severe occlusion present in crowded scenes where vehicles can block the view of other agents and themselves. Xiang *et al.* introduce visibility patterns into the model to

TABLE III
COMPARISON OF 3D OBJECT DETECTION METHODS BY CATEGORY

Category		**Methodology/Advantages**	**Limitations/Drawbacks**	**Research Gaps**
Monocular		Uses single RGB images to predict 3D object bounding boxes. Predicts 2D bounding boxes on the image plane then extrapolate them to 3D through reprojection constraints or bounding box regression.	The lack of explicit depth information on the input format limits the accuracy of localization performance.	CNNs that estimate depth channels could be investigated to increase localization accuracy.
Point-cloud	Projection	Projects point clouds into a 2D image and use established architectures for object detection on 2D images with extensions to regress 3D bounding boxes.	Projecting the point cloud data inevitably causes information loss. It also prevents the explicit encoding of spatial information as in raw point cloud data.	The encoding of the input image is performed with hand-engineered features (point density, etc.). Learned input representations could improve the detection results.
	Volumetric	Generates a 3 dimensional representation of the point cloud in a voxel structure and uses Fully Convolutional Networks (FCNs) to predict object detections. Shape information is encoded explicitly.	Expensive 3D convolutions increase models inference time. The volumetric representation is sparse and computationally inefficient.	Volumetric methods have not considered region proposals, which could improve both localization accuracy and processing time.
	PointNet	Uses feed-forward networks consuming raw 3D point clouds to generate predictions on class and estimated bounding boxes.	Considering the whole point cloud as input can increase run-time. Difficult establishing region proposals considering raw point inputs.	PointNet architectures rely on region proposals to limit the number of points. Proposal methods based uniquely on point-cloud data should be investigated.
Fusion		Fuses both front view images and point clouds to generate a robust detections. Architectures usually consider multiple branches, one per modality, and rely on region proposals. Allows modalities to interact and complement each other.	Requires calibration between sensors, and depending on the architecture can be computationally expensive.	These methods represent state-of-the-art detectors. However, they should be evaluated on more general scenarios including diverse lighting and weather conditions.

mitigate occlusion effects through object reasoning. They propose the 3D Voxel Pattern (3DVP) [41] representation that models appearance through RGB intensities, 3D shape as a set of voxels and occlusion masks. This representation allows to recover which parts of the object are visible, occluded or truncated. They obtain a dictionary of 3DVPs by clustering the patterns observed on the data and training a classifier for each specific pattern given a 2D image segment of the vehicle. During the test phase the pattern obtained through classification is used for occlusion reasoning and 3D pose and localization estimation. They achieve 3D detection by minimizing the reprojection error between the projected 3D bounding box to the image plane and the 2D detection. Their pipeline is still dependent on the performance of Region Proposal Networks (RPNs).

Although some RPNs were able to improve traditional proposal methods [37] they still fail to handle occlusion, truncation and different object scales. Extending the previous 3DVP framework, the same authors propose SubCNN [45], a CNN that explores class information for object detection at the RPN level. They use the concept of subcategory, which are classes of objects sharing similar attributes such as 3D pose or shape. Candidates are extracted using convolutional layers to predict heat maps for each subcategory at the RPN level. After Region of Interest (ROI) proposal the network outputs category classification along with refined 2D bounding box estimates. Using 3DVPs [41] as subcategories for pedestrian, cyclist and vehicle classes, the model recovers 3D shape, pose and occlusion patterns. An extrapolating layer is used to improve small object detection by introducing multi-scale image pyramids.

Despite the previous 3DVP representations [41], [45] allow to model occlusion and parts appearance, they are obtained as a classification among an existing dictionary of visibility patterns common in the training set. Thus, may fail to generalize to an arbitrary vehicle pose that differs from the existing patterns. To overcome this, Deep MANTA [42] uses a many-task network to estimate vehicle position, part localization and shape based only on monocular images. The vehicle shape consists of a set of key points that characterize the vehicle 3-dimensional boundaries, *e.g.* external vertices of the vehicle. They first obtain 2D bounding regression and parts localization through a two-level refinement region-proposal network. Next, based on the inferred shape 3D model matching is performed to obtain the 3D pose.

Previous attempts performed either exhaustive search on the 3D bounding box space [39], estimated 3D pose through a cluster of appearance patterns [41] or 3D templates [42]. Mousavian *et al.* [40] first extend a standard 2D object detector with 3D orientation (yaw) and bounding box sizes regression. This is justified by the box dimensions having smaller variance and being invariant with respect to the orientation. Most models use L2 regression for orientation angle prediction. In contrast, the authors propose a Multi-bin method to regress orientation. The angle is considered to belong to one of n overlapping bins and a network estimates the confidence of the angle belonging to each bin along with a residual angle to be added to the bin center to recover the output angle.

TABLE IV
SUMMARY OF MONOCULAR BASED METHODS

Method	Methodology/Contributions	Limitations
Mono3D [39]	Improves detection performance over 3DOP that relied on the depth channel.	Poor localization accuracy given the lack of depth cues.
3DVP [41]	Novel 3DVP object representation includes appearance, 3D shape and occlusion information. Classification among an existing set of 3DVPs allows occlusion reasoning and recovering 3D pose and localization.	Fixed set of 3DVPs extracted during training limits generalisation to arbitrary object poses.
SubCNN [45]	Uses 3DVP representation to generate occlusion-aware region proposals. The proposals are refined and classified within the object representations (3DVP). Improves RPN model refinement network using CNNs.	Since the 3DVP representation is employed, this method has the same limitations as the previous one.
DeepManta [42]	CNN to predict parts localization and visibility, fine orientation, 3D localization and template, 3D template matching to recover 3D position.	Detection restricted to vehicles, ignoring other classes.
Deep3DBox [40]	Simplified network architecture by independently regressing bounding box size and angle. Then using image reprojection error minimization to obtain 3D localization.	The reprojection error is dependent on the BB size and angle regressed by the network. This dependence increases localization error.
360Panoramic [46]	Estimates depth for 360 degrees panoramic monocular images. Then adapt a CNN to predict 3D object detections on the recovered panoramic image. The only method capable of using images to detect objects at any angle around the vehicle.	Limited to vehicle detection and fails when the vehicle is too close to the camera. The resolution of the camera limits the range of detection.

The 3D box dimensions and orientations are fixed as determined by the network prediction. Then 3D object pose is recovered solving for a translation matrix that minimizes the reprojection error of the 3D bounding box w.r.t. the 2D detection box on the image plane.

All previous monocular methods can only detect objects from the front-facing camera, ignoring objects on the sides and rear of the vehicle. While lidar methods can be used effectively for 360 degrees detection, [46] proposes the first 360 degrees panoramic image based method for 3D object detection. They estimate dense depth maps of panoramic images and adapt standard object detection methods for the equirectangular representation. Due to the lack of panoramic labelled datasets for driving, they adapt the KITTI dataset using style and projection transformations. They additionally provide benchmark detection results on a synthetic dataset.

Monocular methods have been widely researched. Although previous works considered hand-engineered features for region proposals [39], most methods have shifted towards a learned paradigm for Region Proposals and second stage of 3D model matching and reprojection to obtain 3D bounding boxes. The main drawbacks of monocular based methods is the lack of depth cues, which limits detection and localization accuracy specially for far and occluded objects, and sensitivity to lighting and weather conditions, limiting the use of these methods for day time. Also, since most methods rely on a front facing camera (except for [46]), it is only possible to detect objects in front of the vehicle, contrasting to point clouds methods that, in principle, have a coverage all around the vehicle. We summarize the methodology/contributions and limitations of monocular methods in Table IV.

B. Point Cloud Based Methods

Current 3D object detection methods based on point-clouds can be divided into three subcategories: projection based, volumetric representations and point-nets. Each category is explained and reviewed below, followed by a summary discussion.

1) Projection Methods: Image classification and object detection in 2D images is a well-researched topic in the computer vision community. The availability of datasets and benchmarked architectures for 2D images make using these methods even more attractive. For this reason, point cloud (PCL) projection methods first transform the 3D points into a 2D image via plane [47], cylindrical [48] or spherical [34] projections that can then be processed using standard 2D object detection models such as [49]. The 3D bounding box can then be recovered using position and dimensions regression.

Li *et al.* [48] uses a cylindrical projection mapping and a Fully Convolutional Network (FCN) to predict 3D bounding boxes around vehicles only. The input image resulting from the projection has channels encoding the points' height and distance from the sensor. This input is fed to a 2D FCN which down-samples the input for three consecutive layers and then uses transposed convolutional layers to up-sample these maps into point-wise "objectness" and bounding box (BB) prediction outputs. The first output defines if a given point is part of a vehicle or the background, effectively working as a weak classifier. The second output encodes the vertices of the 3D bounding box delimiting the vehicle conditioned by the first output. Since there will be many BB estimates for each vehicle, an NMS strategy is employed to reduce overlapping predictions based on score and distance. The authors train this detection model in an end-to-end fashion on the KITTI dataset with loss balancing to avoid bias towards negative samples or near cars, which appear more frequently.

While previous methods used cylindrical and spherical projections, [30], [50], [51] use the bird-eye view projection to generate 3D proposals. They differ regarding the input representation: the first encodes the 2D input cells using the minimum, median and maximum height values of the points lying inside the cell as channels, while the last two use height, intensity and density channels. The first approach uses a Faster R-CNN [13] architecture as a base with an adjusted refinement network that outputs oriented 3D bounding boxes. Despite their reasonable bird-eye view results, their method performs poor orientation angle regression. Most lidar base methods use sensors with high point density, which limits the application of the resulting models on low-end lidar sensors. Beltrán *et al.* [51] propose a novel encoding that normalizes the density channel based on the parameters of the lidar being used. This normalization creates a uniform representation and allows to generalize the detection model to sensors with different specifications and number of beams.

One fundamental requirement of safety-critical systems deployed on autonomous vehicles, including object detection, is real-time operation capability. These systems must meet strict response time deadlines to allow the vehicle to respond to the environment. Complex-YOLO [30] focus on efficiency using a YOLO [52] based architecture, with extensions to predict the extra dimension and yaw angle. While classical RPN approaches further process each region for finer predictions, this architecture is categorized as a single-shot detector, obtaining detections in a single forward step. This allows Complex-YOLO to achieve a runtime of 50 fps, up to five times more efficient than previous methods, despite inferior, but comparable detection performance.

Quantifying the confidence of predictions made by an AV's object detection system is fundamental for the safe operation of such vehicle. As with human drivers, if the system has low confidence on its predictions, it should enter a safe state to avoid risks. Although most detection models offer a score for each prediction, they tend to use *softmax* normalization to obtain class distributions. Since this normalization forces the sum of probabilities to unity, it does not necessarily reflect the absolute confidence on the prediction. Feng *et al.* [53] uses a Bayesian Neural Network to predict the class and 3D bounding box after ROI pooling, which allows to quantify the network confidence for both outputs. The authors quantify epistemic and aleatoric uncertainties. While the former measures the model uncertainty to explain the observed object, the latter relates to observation noises in scenarios of occlusion and low point density. They observed an increase in detection performance when modeling aleatoric uncertainty by adding a constraint that penalizes noisy training samples.

2) Volumetric Convolutional Methods: Volumetric methods assume that the object or scene is represented in a 3D grid, or a voxel representation, where each unit has attributes, such as binary occupancy or a continuous point density. One advantage of such methods is that they encode shape information explicitly. However, as a consequence, most of the volume is empty, resulting in reduced efficiency while processing these empty cells. Additionally, since data is three dimensional by nature 3D convolutions are necessary, drastically increasing the computational cost of such models.

To this effect [54], [55] address the problem of object detection on driving scenarios using one-stage FCN on the entire scene volumetric representation. This one-stage detection differs from two-stage where region proposals are first generated and then refined on a second processing stage. Instead, one-stage detectors infer detection predictions in a single forward pass. Li [54] uses a binary volumetric input and detects vehicles only. The model's output maps represent "objectness" and BB vertices predictions, similarly to the authors' previous work [48]. The first output predicts if the estimated region belongs to an object of interest, while the second predicts its coordinates. They use expensive 3D convolutions which limits temporal performance.

Aiming at a more efficient implementation, [55] fixes BB sizes for each class but detects cars, pedestrians and cyclists. This assumption simplifies the architecture and together with a sparse convolution algorithm greatly reduces the model's complexity. L1 regularization and Rectified Linear Unit (*ReLU*) activation functions are used to maintain sparsity across convolutional layers. Parallel networks are used independently for each class during inference. The assumption of fixed BB sizes allows to train the network directly on the 3D crops of positive samples. During training they augment the data with rotation and translation transformation and employ hard negative mining to reduce false positives.

3) Point-Nets Methods: Point clouds consist of a variable number of 3D points sparsely distributed in space. Therefore, it is not obvious how to incorporate their structure to traditional feed-forward deep neural networks pipelines that assume fixed input data sizes. Previous methods attempted to either transform the point cloud raw points into images using projections or into volumetric structures using voxel representations. A third category of methods, called Point-nets, handle the irregularities by using the raw points as input in an attempt to reduce information loss caused by either projection or quantization in 3D space. We first review seminal work and then progress to driving specific applications.

The seminal work in the category is introduced by PointNet [56]. Segmented 3D PCLs are used as input to perform object classification and part-segmentation. The network performs point-wise transformations using Fully-Connected (FC) layers and aggregates a global feature through a max-pooling layer, ensuring independence on point order. Experimental results show that this approach outperforms volumetric methods [57], [58]. This model is further extended in PointNet++ [59], where each layer progressively encode more complex features in a hierarchical structure. The model generate overlapping sets of points and local attribute features are obtained by feeding each set to a local PointNet. Follow up work by Wang *et al.* [60] further generalize the PointNet architecture by considering points pair-wise relationships. More detailed information on convolutional neural networks for irregular domains is out of the scope of this paper but can be found in [61].

The seminal methods assumed segmented PCLs that contain a single object, but the gap between object classification and detection is still an open question. VoxelNet [62] uses raw point subsets to generate voxel-wise features, creating a uniform representation of the point cloud, as obtained in volumetric methods. The first step randomly selects a fixed number of points from each voxel, reducing evaluation time and enhancing generalization. Each set of points is used by a voxel-feature-encoding (VFE) layer to generate a 4D point cloud representation. This representation is fed to 3D convolutional layers, followed by a 3D region proposal network to predict BB location, size and class. The authors implement an efficient convolution operation considering the sparsity of

TABLE V

SUMMARY OF POINT CLOUD-BASED METHODS

SubCategory	Method	Methodology/Contributions	Limitations
Projection	VeloFCN [48]	Uses fully convolutional architecture with lidar point cloud bird-eye view projections. Output maps represent 3D bounding box regressions and "objectness" score, the likelihood of having an object at that position.	Detects vehicles only. Limited performance on small or occluded objects due to the loss of resolution across feature maps.
	C-YOLO [30]	Uses a YOLO based single-shot detector extended for 3D BB and orientation regression. The proposed architecture achieves 50 fps runtime, more than any previous method.	There is a tradeoff between inference time and detection accuracy. Single-shot networks underperform networks that use a second stage for refinement.
	TowardsSafe [53]	Uses variational dropout inference to quantify uncertainty in class and bounding box predictions. Aleatoric noise modelling allows the network to generalise better by reducing the impact of noisy samples in the training process.	The uncertaity estimation requires several forward passes of the network. This limits the temporal performance of this method, preventing real-time results.
	BirdNet [51]	Normalizes point cloud representation to allow detection generalisation across different lidar models and specifications.	Input image with only 3 channels encoding height, density and intensity information looses detailed information, which degrades performance.
Volumetric	3DFCN [54]	Extension of the FCN architecture to voxelized lidar points clouds. Single shot detection method.	Requires 3D convolutions, limiting temporal performance to 1 fps.
	Vote3Deep [55]	Proposes an efficient convolutional algorithm to exploit the sparsity of volumetric point cloud data. Uses L1 regularisation and Rectified Linear Unit (ReLU) to maintain sparsity.	Assumes fixed sizes for all detected objects, limiting the detection performance.
PointNet	VoxelNet [62]	Extends PointNet concept to point clouds in a scene scale. Uses raw 3D points to learn a volumetric representation through Voxel Feature Encoding layers. The volumetric features are used for 3D region proposal.	Expensive 3D convolutions limits time performance. Models are class specific, thus multiple models must be run in parallel at run time.

the voxel representation. Different voxel sizes are used for cars and pedestrians/cyclists to avoid detail loss. Models are trained independently for each class, resulting in three models that must be used simultaneously during inference. In Frustum PointNet [63] detection is achieved by selecting sets of 3D points and using a PointNet to classify and predict bounding boxes for each set. The set selection criterion is based on 2D detections on the image plane, thus this method is classified as a Fusion method, reviewed in Section IV-C.

4) Discussion: Among point cloud based methods, the projection subcategory has gained most attention due to the proximity to standard image object detection. Particularly, it offers a good trade-off between time complexity and detection performance. However, most methods rely on hand-engineered features when projecting the point cloud (density, height, etc.). In contrast, PointNet methods uses the raw 3D points to learn a representation in feature space. In this last category it is still necessary to investigate new forms of using a whole scene point cloud as input, as regular PointNet models assume segmented objects. Volumetric methods transform the point cloud into voxel representations where the space information in explicitly encoded. This approach causes a sparse representation which is inefficient given the need of 3D convolutions. We present a summary of point cloud-based methods in Table V.

C. Fusion Based Methods

As mentioned previously, point clouds do not provide texture information, which is valuable for class discrimination in object detection and classification. In contrast, monocular images cannot capture depth values, which are necessary for accurate 3D localization and size estimation. Additionally, the density of point clouds tends to reduce quickly as the distance from the sensor increases, while images can still provide a means of detecting far vehicles and objects. In order to increase the overall performance, some methods try to use both modalities with different strategies and fusion schemes. Generally there are three types of fusion schemes [64]:

Early fusion: Modalities are combined at the beginning of the process, creating a new representation that is dependent on all modalities.

Late fusion: Modalities are processed separately and independently up to the last stage, where fusion occurs. This scheme does not require all modalities be available as it can rely on the predictions of a single modality.

Deep fusion: Proposed in [64], it mixes the modalities hierarchically in neural network layers, allowing the features from different modalities to interact over layers, resulting in a more general fusion scheme.

Schlosser *et al.* [65] evaluate the fusion at different stages of a 3D pedestrian detection pipeline. Their model considered two inputs: monocular image and a depth frame. The authors conclude that late fusion yields the best performance, although early fusion can be used with minor performance drop.

One fusion strategy consists of using the point cloud projection method, presented in Section IV-B.1, with extra RGB channels of front facing cameras along the projected PCL maps to obtain higher detection performance. Two of these methods [6], [64] use 3D region proposal networks (RPNs) to generate 3D Regions of Interest (ROI) which are then projected to the specific views and used to predict classes and 3D bounding boxes.

TABLE VI
SUMMARY OF FUSION BASED METHODS

Method	**Methodology/Contributions**	**Limitations**
MV3D [64]	Uses bird-eye and front view lidar projections as well as monocular camera frames to detect vehicles. 3D proposal network based on the bird-eye-view. Introduces a deep fusion architecture to allow interactions between modalities.	Although far objects might be visible through the camera, the low lidar point density prevents detection of these objects. Specifically, the RPN based on the bird-eye view only limits these detections. Detects vehicles only.
AVOD [6]	Uses bird-eye lidar projection and monocular camera only. New RPN uses both modalities to generate proposals. A Feature Pyramid Network extension improves detection of small objects by up sampling feature maps. New vector representation removes ambiguities in the orientation regression. Can detect vehicles, pedestrians and cyclists.	Detection method only sensitive to objects in front of the vehicle due to the forward-facing camera used.
F-PointNet [63]	Extracts 2D detection from image plane, extrapolates detection to a 3D frustum, selecting lidar points. Uses a PointNet instance to segment background points and generate 3D detections. Can detect vehicles, pedestrians and cyclists.	Since proposals are obtained from the front view image, failing to detect objects in this view limits the detection performance. This limits the use of this method at night time, for example.

The first method, MV3D [64], uses bird-eye and front view projections of lidar points along the RGB channels of a forward facing camera. The network consists of three input branches, one for each view, with VGG [38] based feature extractors. The 3D proposals, generated based on the bird-eye view features only, are projected to each view's feature maps. A ROI pooling layer extracts the features corresponding to each view's branch. These proposal-specific features are aggregated in a deep fusion scheme, where feature maps can hierarchically interact with one another. The final layers output the classification result and the refined vertices of the regressed 3D bounding box. The authors investigate the performance of different fusion methods and conclude that the deep fusion approach obtains the best performance since it provides more flexible means of aggregating features from different modalities.

The second method, AVOD [6], is the first to introduce an early fusion approach where the bird-eye view and RGB channels are merged for region proposal. The input representations are similar to MV3D [64] except that only the bird-eye view and image input branches are used. Both modalities' feature maps are used by the RPN, achieving high proposal recall. The highest scoring region proposals are sampled and projected into the corresponding views' feature maps. Each modality proposal specific features are merged and a FC layer outputs class distribution and refined 3D boxes for each proposal. Commonly, loss of details after convolutional stages prevents detection of small objects. The authors circumvent this by upsampling the feature maps using Feature Pyramid Networks [66]. Qualitative results show robustness to snowy scenes and poor illumination conditions on private data.

A second strategy consists of using the monocular image to obtain 2D candidates and extrapolate these detections to the 3D space where point cloud data is employed. In this category Frustum Point-Net [63] generates region proposals on the image plane with monocular images and use the point cloud to perform classification and bounding box regression. The 2D boxes obtained over the image plane are extrapolated to 3D using the camera calibration parameters, resulting in frustums region proposals. The points enclosed by each frustum are selected and segmented with a PointNet instance to remove the background clutter. This set is then fed to a second PointNet instance to perform classification and 3D BB regression. Similarly, Du *et al.* [67] first select the points that lie in the detection box when projected to the image plane, then use these points to perform model fitting, resulting in a preliminary 3D proposal. The proposal is processed by a two-stage refinement CNN that outputs the final 3D box and confidence score. The detections in both these approaches are constrained by the proposal on monocular images, which can be a limiting factor due to lighting conditions, *etc*.

Fusion methods obtain state-of-the-art detection results by exploring complimentary information from multiple sensor modalities. While lidar point clouds provide accurate depth information with sparse and low point density at far locations, cameras can provide texture information which is valuable for class discrimination. Fusion of information at feature levels allow to use complimentary information to enhance performance. We provide a summary of fusion methods in Table VI.

V. EVALUATION

This section presents metrics commonly used for 3D object detection. Performance for some of the reviewed methods is also provided, followed by a comprehensive discussion of the results. Finally, we present research challenges and opportunities.

A. Metrics

For any detection or classification task that outputs a confidence y_i of sample x_i belonging to the positive class, it is possible to compute a precision/recall curve using the ranked output. Recall is defined as the proportion of all positive samples ranked above a given threshold t:

$$r(t) = P(y_i \geq t \mid x_i \in C) \tag{1}$$

where C is the set of positive samples.

Likewise, precision is the proportion of all samples above threshold t which are from the positive class:

$$p(t) = P(x_i \in C \mid y_i \geq t). \tag{2}$$

Although both precision and recall are parameterized by t, the precision/recall curve can be parameterized by the recall r.

This curve can be summarized by a single metric called Average Precision (AP) [68]:

$$\mathrm{AP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} p_{\mathrm{interp}}(r) \tag{3}$$

where $p_{\mathrm{interp}}(r) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r})$ is an interpolated version of the precision for a recall level r. This metric is the average precision at 11 different recall levels, ranging from 0 to 1 with 0.1 step size, and reduces the impact of small variations in the probabilistic output.

Most of the discussed works used the KITTI dataset for training and evaluation, which provides a consistent baseline for comparison. Detections are evaluated considering the image plane AP, hereafter called AP2D. Samples are considered true positives if the overlapping area of the estimated and ground-truth boxes exceeds a certain threshold. Specifically, the Intersection over Union (IoU) of the bounding boxes areas in the image plane should exceed 0.5 for pedestrians and cyclists and 0.7 for vehicles. The dataset guidelines suggest to evaluate 3D object detection using both $\mathrm{AP}_{\mathrm{2D}}$ and Average Orientation Similarity (AOS) metrics [29]. The latter jointly measures the 2D detection and 3D orientation performance by weighting the $\mathrm{AP}_{\mathrm{2D}}$ score with the cosine similarity between the estimated and ground-truth orientations.

Despite being employed by most monocular models, these two metrics fail to decouple the effects of localization and bounding box sizes estimation [69]. They also introduce distortion due to image plane projection. For example, two objects of different sizes at different locations can have the same bounding box projection on the image plane. To solve this, Chen *et al.* [64] project the 3D detections into the bird-eye view to compute a more meaningful 3D localization metric ($\mathrm{AP}_{\mathrm{BV}}$). To overcome the projection distortion, they also use an $\mathrm{AP}_{\mathrm{3D}}$ metric, which uses the IoU of volumes of 3D boxes. These metrics are crucial because they allow to assess localization and size regression performance that cannot be reliably captured only by the image plane AP.

Still, the $\mathrm{AP}_{\mathrm{3D}}$ metric fails to precisely assess orientation estimation. This is due to the metric considering positive samples based on a threshold of the IoU metric. In this case, it will not penalize orientation error as long as there is sufficient overlapping volume. Ku *et al.* [6] penalize orientation angle by extending the AOS metric with 3D volume overlapping. They use the $\mathrm{AP}_{\mathrm{3D}}$ metric weighted by the cosine similarity of regressed and ground-truth orientations, resulting in the Average Heading Similarity (AHS) metric:

$$\mathrm{AHS} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} \max_{\tilde{r}:\tilde{r} \geq r} s(\tilde{r}) \tag{4}$$

with $s(r)$ being the orientation similarity defined for every recall r as

$$s(r) = \frac{1}{|\mathbb{D}(r)|} \sum_{i \in \mathbb{D}(r)} \mathbb{1}(\mathrm{IoU} \geq \lambda) \frac{1 + \cos(\theta - \tilde{\theta})}{2} \tag{5}$$

where θ is the orientation estimate and $\tilde{\theta}$ the ground-truth orientation, $\mathbb{D}(r)$ is the set of all detections at recall r and $\mathbb{1}(\mathrm{IoU} \geq \lambda)$ is the indicator function to consider valid detection during AHS computation. The indicator function can be shaped to compute both the 3D (using IoU of the volumes) and bird-eye (IoU of the bird-eye projections) AHS. Note that the AHS is upper bounded by $\mathrm{AP}_{\mathrm{3D}}$ or $\mathrm{AP}_{\mathrm{BV}}$, depending on the metric used.

TABLE VII

KITTI Test Set Results on 2D Object Detection for Car Class

Method	Modality	$\mathrm{AP}_{\mathrm{2D}}$			AOS		
		E	M	H	E	M	H
DeepManta [42]	Mono	96.4	90.1	80.79	96.32	89.91	80.55
SubCNN [45]	Mono	90.81	89.04	79.27	90.67	88.62	78.68
Deep3DBox [40]	Mono	92.98	89.04	77.17	92.9	88.75	76.76
DeepStOP [44]	Stereo	93.45	89.04	79.58	92.04	86.86	77.34
Mono3D [39]	Mono	92.33	88.66	78.96	91.01	86.62	76.84
3DOP [43]	Stereo	93.04	88.64	79.1	91.44	86.1	76.52
3DVP [41]	Mono	87.46	75.77	65.38	86.92	74.9	64.11
F-PointNet [63]	LIDAR+Mono	90.78	90	80.8			
MV3D [64]	LIDAR+Mono	90.53	89.17	80.16			
AVOD-FPN [6]	LIDAR+Mono	89.99	87.44	80.05	89.95	87.13	79.74
VoxelNet [62]	LIDAR	90.3	85.95	79.21			
3DFCN [54]	LIDAR	84.2	75.3	68	84.1	75.2	67.9
Vote3Deep [55]	LIDAR	76.79	68.24	63.23			
VeloFCN [48]	LIDAR	60.3	47.5	42.7	59.1	459	41.1

E, M and H stands for Easy, Moderate and Hard, respectively.

B. Performance of Existing Methods

All the reviewed methods in this paper provide 3D bounding box outputs. However, most monocular based methods directly predict 2D detections on their pipeline before generating the 3D detection. Many of these methods only provide $\mathrm{AP}_{\mathrm{2D}}$ and AOS result metrics. For this reason and considering an extensive comparison between methods, we report image plane object detection results in Table VII for the car class and Table VIII for pedestrian and cyclists classes obtained on the original papers and the KITTI online benchmark [5].

The first group in Tables VII and VIII lists monocular based methods which optimize 2D detections separately. On the other hand, methods in the second group optimize 3D bounding boxes directly. For evaluation, the latter group projects the 3D boxes onto the image plane. This projection result in 2D boxes that does not necessarily fit tightly to predictions based on the image plane directly due to the yaw angle and size predictions. This explains the disparity of results between the two groups, specifically for the Easy category.

The same tables reveal a disparity in performance between classes: cars' $\mathrm{AP}_{\mathrm{2D}}$ is at least 10% higher than pedestrians and cyclists for most methods. This effect happens for two reasons. Firstly, bigger objects are more easily detected and are more resilient to occlusion than smaller ones. Secondly, many methods only fine-tune their models for vehicles, where different classes may require another set of hyper-parameters. In addition, the intra-class performance degrades as the complexity increases, which is explained by severe occlusion in moderate and hard samples.

Despite the reasonable results on image plane projections, the first two tables fail to assess all the components

TABLE VIII
KITTI TEST SET RESULTS ON 2D OBJECT DETECTION FOR PEDESTRIANS AND CYCLISTS

Method	Modality	AP_{2D}						AOS					
		Pedestrians			Cyclists			Pedestrians			Cyclists		
		E	M	H	E	M	H	E	M	H	E	M	H
SubCNN [45]	Mono	83.28	71.33	66.36	79.48	71.06	62.68	78.45	6.28	61.36	72	63.65	56.32
DeepStereoOP [39]	Stereo	81.82	67.32	65.12	79.58	65.84	57.90	72.82	59.28	56.85	69.20	55.69	48.95
Mono3D [39]	Mono	80.35	66.68	63.44	76.04	66.36	58.87	71.15	58.15	54.94	65.56	54.97	48.77
3DOP [43]	Stereo	81.78	67.47	64.7	78.39	68.94	61.37	72.94	59.8	57.03	70.13	58.68	52.35
F-PointNet [63]	LIDAR+Mono	87.81	77.5	74.46	84.9	72.25	65.14						
AVOD-FPN [6]	LIDAR+Mono	67.32	58.4	57.44	68.65	59.32	55.82	53.36	44.92	43.77	67.61	57.53	54.16
VoxelNet [62]	LIDAR	50.61	44.08	42.84	72.04	59.33	54.72						
Vote3Deep [55]	LIDAR	68.39	55.37	52.59	79.92	67.88	62.98						

E, M and H stands for Easy, Moderate and Hard, respectively.

TABLE IX
KITTI VALIDATION SET RESULTS FOR AP_{3D} AND AHS FOR CAR CLASS. OBTAINED FROM [6] AND [64]

Method	Modality	AP_{3D}			AHS		
		E	M	H	E	M	H
Mono3D [39]	Mono	2.53	2.31	2.31			
Deep3DBox [40]	Mono	5.84	4.09	3.83	5.84	4.09	3.83
3DOP [43]	Stereo	6.55	5.07	4.1			
MV3D [64]	LIDAR+Mono	83.87	52.74	72.35	64.56	43.75	39.86
AVOD-FPN [6]	LIDAR+Mono	84.41	74.44	68.65	84.19	74.11	68.28

E, M and H stands for Easy, Moderate and Hard, respectively.

of 3D detection, *e.g.* localization, dimension and orientation regression. To this effect, Table IX obtained from [6] presents 3D metrics on three methods for the car class on the KITTI validation set with 0.7 3D IoU threshold. The AHS metric confirms that the orientation regression proposed by AVOD [6] fixes the ambiguity in the representation adopted in MV3D [64].

Monocular detection methods show very limited performance on 3D detection metrics, as evidenced by the large performance gap on 3D metrics between the two groups in Table IX. This poor performance arises from the lack of depth information in monocular images. Hence, monocular methods cannot be reliably used for 3D object detection in AVs. Moreover, this evaluation suggests that the AP_{2D} and AOS metrics are not enough to confidently assess 3D object detection methods.

Table X presents 3D metrics on the KITTI 3D object detection benchmark [5] considering the impact of localization and bounding box parameters. The results of monocular methods are not available because the 3D object detection benchmark has been established after the publication of those methods. Clearly the detection performance gap evidenced in 2D and 3D AP metrics is still large. The best performing method achieves approximately 82% AP_{3D} on the easy car class, while the image plane counterpart achieves higher than 95% AP_{2D}. This is explained by the complexity in regressing parameters for an extra dimension and also motivates further research to improve results and enable robust detection for autonomous driving applications.

TABLE X
3D OBJECT DETECTION BENCHMARK ON KITTI TEST SET. 3D IOU 0.7

Method	Time(s)	Class	AP_{3D}			AP_{BV}		
			E	M	H	E	M	H
MV3D [64]	0.36	Car	71.09	62.35	55.12	86.02	76.9	68.49
AVOD [6]	0.08		73.59	65.78	58.38	86.8	85.44	77.73
AVOD-FPN [6]	0.1		81.94	71.88	66.38	88.53	83.79	77.9
F-Pointnet [63]	0.17		81.2	70.39	62.19	88.7	84	75.33
Voxelnet [62]	0.23		77.47	65.11	57.73	89.35	79.26	77.39
C-YOLO[1] [30]	0.02		67.72	64	63.01	85.89	77.4	77.33
AVOD [6]	0.08	Ped	38.28	31.51	26.98	42.51	35.24	33.97
AVOD-FPN [6]	0.1		50.8	42.81	40.88	58.75	51.05	47.54
F-Pointnet [63]	0.17		51.21	44.89	40.23	58.09	50.22	47.2
Voxelnet [62]	0.23		39.48	33.69	31.51	46.13	40.74	38.11
C-YOLO[1] [30]	0.02		41.79	39.7	35.92	46.08	45.9	44.2
AVOD [6]	0.08	Cyc	60.11	44.9	38.8	63.66	47.74	46.55
AVOD-FPN [6]	0.1		64	52.18	46.61	68.09	57.48	50.77
F-Pointnet [63]	0.17		71.96	56.77	50.39	75.38	61.96	54.68
Voxelnet [62]	0.23		61.22	48.36	44.37	66.7	54.76	50.55
C-YOLO[1] [30]	0.02		68.17	58.32	54.3	72.37	63.36	60.27

1 The authors did not provide public test set results, only validation set
E, M and H stands for Easy, Moderate and Hard, respectively.

Region proposal networks' performance is critical as they impose the upper bound detection recall for two stage detectors. These networks can be regarded as weak classifiers, which aim at narrowing down the object search space. Thus, reducing the number of possibilities that a more specific, complex network has to process. Ideally, it should retrieve all the instances in order to avoid false negatives, although it is not expected to get a high precision on these primitive proposals. Ku *et al.* [6] assess their fusion RPN scheme using the recall metric and compare it to other baseline methods, see Figure 4. These results show the performance improvement achieved by learning approaches versus hand-engineered based proposals such as in [39] and [43]. Unlike cars, pedestrians and cyclists have a significant improvement when considering the fusion scheme. These classes have smaller dimensions and cannot be completely represented exclusively in the bird-eye view, benefiting from more information obtained from the image plane.

Regarding runtime, most methods cannot operate in the real-time, considering the lidar or camera frame rate. As an exception, Complex-YOLO [30] achieves a 50 fps frame rate.

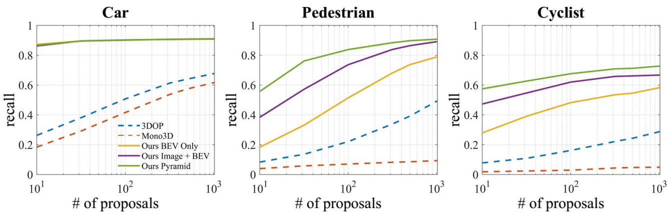

Fig. 4. Recall vs number of proposals 3D IoU threshold of 0.5 for three classes on KITTI validation set with moderate samples. Obtained from [6].

The simpler single-shot architecture reflects in a small performance drop. It must be noted that the authors did not provide public results on the KITTI test set, reporting their results on the validation set instead, which makes direct comparison to other methods dubious.

C. Research Challenges and Opportunities

We propose some further research topics that should be considered to advance the performance of 3D object detection in the context of autonomous vehicles. The topics were elaborated based on the significant performance disparity between 2D and 3D detectors and gaps found in the literature.

1) Most research in 3D object detection has focused on improving the benchmark performance of such methods. Although this is a valid goal, there is no understanding on the required detection performance levels for reliable driving applications. In this regard, a valid research opportunity is in investigating how detection performance relates to the safety of driving, measured by relevant Key Performance Indicators (KPIs).
2) The recent advances in PointNets, described in Section IV-B.3, can be explored to verify resilience to missing points and occlusion, which is still the main cause of poor performance on hard samples. More specifically, the geometrical relationships between points could be explored to obtain significant information that cannot be attained considering each point individually.
3) Many methods consider sensor fusion to improve reliability of the perception system. Considering the disparity in point density, a possible contribution would include a collaborative perception approach in a multi-agent fusion scheme. Vehicles could use V2X or LTE communication technology to share relevant perception information that could improve and extend the visibility of the environment and thus reduce uncertainty and improve performance perception methods.
4) An important limitation of the KITTI dataset is its characteristic daylight scenes and very standard weather conditions. Although [6] reports having tested their method during night-time and under snow, they only report qualitative results. Further research should be conducted to evaluate the effect of such conditions on the object detection pipeline and how to achieve reliable performance under general conditions. The simulation tools described in Section III could be used to obtain preliminary results.
5) The run time presented in Table X show that most methods can only achieve lower than 10 fps, which is the minimum rate to keep real time operation with the lidar frame rate. Significant improvement has to be done to obtain fast and reliable recognition systems operating on real environments.
6) Most methods cannot output a calibrated confidence [70] on predictions, which can lead to dangerous behaviors in real scenarios. Seminal work [53] identified this gap and proposed a method to quantify uncertainty in detection models, but failed to achieve real-time performance. More research should be conducted in this area to understand the origins of uncertainty and how to mitigate them.

VI. Conclusion

This paper reviewed the state-of-the-art of 3D object detection within the context of autonomous vehicles. We analyzed sensors technologies with their advantages and disadvantages, and discussed standard datasets. The reviewed works were categorized based on sensor modality: monocular images, point clouds (obtained through lidars or depth cameras) and fusion of both.

Quantitative results, obtained from the KITTI benchmark, showed that monocular methods are not reliable for 3D object detection, due to lack of depth information, which prevents accurate 3D positioning. On the other hand, fusion methods were used to extract the most relevant information from each modality and achieve state-of-the-art results for 3D object detection. Finally, we presented directions of future work.

References

[1] A. Dhani, "Reported road casualties in Great Britain: Quarterly provisional estimates year ending September 2017," U.K. Dept. Transport, London, U.K., Tech. Rep., Feb. 2018. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/681593/quarterly-estimates-july-to-september-2017.pdf

[2] S. Singh, "Traffic safety facts," Nat. Highway Traffic Saf. Admin., U.S. Dept. Transp., Washington, DC, USA, Tech. Rep. DOT HS 812 115, Feb. 2015. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115

[3] Atkins Ltd. "Research on the impacts of connected and autonomous vehicles (CAVs) on traffic flow," U.K. Dept. Transport, London, U.K., Tech. Rep. SO13994/3, May 2016. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/530091/impacts-of-connected-and-autonomous-vehicles-on-traffic-flow-summary-report.pdf

[4] K. Habib, "Technical report, Tesla crash," Nat. Highway Traffic Saf. Admin., U.S. Dept. Transp., Washington, DC, USA, Tech. Rep. PE 16-007, Jan. 2017. [Online]. Available: https://static.nhtsa.gov/odi/inv/2016/INCLA-PE16007-7876.PDF

[5] *KITTI 3D Object Detection Online Benchmark*. Accessed: Jun. 15, 2018. [Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d

[6] 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)*, 2018, pp. 1–8.

[7] B. Ranft and C. Stiller, "The role of machine vision for intelligent vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 8–19, Mar. 2016.

[8] S. D. Pendleton *et al.*, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, Feb. 2017. [Online]. Available: http://www.mdpi.com/2075-1702/5/1/6

[9] A. Mukhtar, L. Xia, and T. B. Tang, "Vehicle detection techniques for collision avoidance systems: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2318–2338, May 2015.

[10] D. Z. Wang, I. Posner, and P. Newman, “What could move? Finding cars, pedestrians and bicyclists in 3D laser data,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2012, pp. 4038–4044.

[11] A. Azim and O. Aycard, “Layer-based supervised classification of moving objects in outdoor dynamic environment using 3d laser scanner,” in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 1408–1414.

[12] J. Behley, V. Steinhage, and A. B. Cremers, “Laser-based segment classification using a mixture of bag-of-words,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2013, pp. 4195–4200.

[13] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 91–99.

[14] Y. Zhang *et al.*, “Towards end-to-end speech recognition with deep convolutional neural networks,” in *Proc. Interspeech*, 2016, pp. 410–414, doi: 10.21437/Interspeech.2016-1446.

[15] J. Van Brummelen, M. O’Brien, D. Gruyer, and H. Najjaran, “Autonomous vehicle perception: The technology of today and tomorrow,” *Transp. Res. C, Emerg. Technol.*, vol. 89, pp. 384–406, Apr. 2018.

[16] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, “A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications,” *IEEE Internet Things J.*, vol. 5, no. 2, pp. 829–846, Apr. 2018.

[17] M. Weber, P. Wolf, and J. M. Zöllner, “DeepTLR: A single deep convolutional network for detection and classification of traffic lights,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 342–348.

[18] S. Sivaraman and M. M. Trivedi, “Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013.

[19] S. Hsu, S. Acharya, A. Rafii, and R. New, “Performance of a time-of-flight range camera for intelligent vehicle safety applications,” in *Advanced Microsystems for Automotive Applications*. Berlin, Germany: Springer, 2006, pp. 205–219.

[20] O. Elkhalili *et al.*, “A 64× 8 pixel 3-D CMOS time of flight image sensor for car safety applications,” in *Proc. 32nd Eur. Solid-State Circuits Conf.*, 2006, pp. 568–571.

[21] *Sony IMX390CQV CMOS Image Sensor for Automotive Cameras*. [Online]. Available: https://www.sony.net/SonyInfo/News/Press/201704/17-034E/index.html

[22] Q. Chen, X. Yi, B. Ni, Z. Shen, and X. Yang, “Rain removal via residual generation cascading,” in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.

[23] *Velodyne HDL-64E Lidar Specification*. Accessed: Apr. 10, 2018. [Online]. Available: http://velodynelidar.com/hdl-64e.html

[24] *Velodyne VLS-128 Announcement Article*. Accessed: Apr. 10, 2018. [Online]. Available http://www.repairerdrivennews.com/2018/01/02/velodyne-leading-lidar-price-halved-new-high-res-product-to-improve-self-driving-cars/

[25] *Leddar Solid-State Lidar Technology*. Accessed: Apr. 10, 2018. [Online]. Available: https://leddartech.com/technology-fundamentals/

[26] Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim, “Calibration between color camera and 3D LIDAR instruments with a polygonal planar board,” *Sensors*, vol. 14, no. 3, pp. 5333–5353, 2014.

[27] R. Ishikawa, T. Oishi, and K. Ikeuchi. (Apr. 2018). “LiDAR and camera calibration using motion estimated by sensor fusion odometry.” [Online]. Available: https://arxiv.org/abs/1804.05178

[28] 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. (CVPR)*, Jun. 2009, pp. 248–255.

[29] 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. (CVPR)*, Jun. 2012, pp. 3354–3361.

[30] M. Simon, S. Milz, K. Amende, and H. Gross. (Mar. 2018). “Complex-YOLO: Real-time 3D object detection on point clouds.” [Online]. Available: https://arxiv.org/abs/1803.06199

[31] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, “Virtual worlds as proxy for multi-object tracking analysis,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4340–4349.

[32] S. Chen, S. Zhang, J. Shang, B. Chen, and N. Zheng, “Brain-inspired cognitive model with attention for self-driving cars,” *IEEE Trans. Cogn. Devel. Syst.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/7954050, doi: 10.1109/TCDS.2017.2717451.

[33] H. Xu, Y. Gao, F. Yu, and T. Darrell, “End-to-end learning of driving models from large-scale video datasets,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3530–3538.

[34] B. Wu, A. Wan, X. Yue, and K. Keutzer. (Oct. 2017). “SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud.” [Online]. Available: https://arxiv.org/abs/1710.07368

[35] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Proc. 1st Conf. Robot Learn. (CoRL)*, Nov. 2017, pp. 1–16.

[36] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, “Sim4CV: A photo-realistic simulator for computer vision applications,” *Int. J. Comput. Vis.*, vol. 126, no. 9, pp. 902–919, Sep. 2018.

[37] R. Girshick, “Fast R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Washington, DC, USA, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[38] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Represent.*, 2015.

[39] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, “Monocular 3D object detection for autonomous driving,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2147–2156.

[40] 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.

[41] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, “Data-driven 3D voxel patterns for object category recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1903–1911.

[42] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, “Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1827–1836.

[43] X. Chen *et al.*, “3D object proposals for accurate object class detection,” in *Advances in Neural Information Processing Systems*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. New York, NY, USA: Curran Associates, Inc., 2015, pp. 424–432. [Online]. Available: http://papers.nips.cc/paper/5644-3d-object-proposals-for-accurate-object-class-detection.pdf

[44] C. C. Pham and J. W. Jeon, “Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks,” *Signal Process., Image Commun.*, vol. 53, pp. 110–122, Apr. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596517300231

[45] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, “Subcategory-aware convolutional neural networks for object proposals and detection,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 924–933.

[46] G. Payen de La Garanderie, A. Atapour Abarghouei, and T. P. Breckon, “Eliminating the blind spot: Adapting 3D object detection and monocular depth estimation to 360° panoramic imagery,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 812–830.

[47] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3D shape recognition,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

[48] B. Li, T. Zhang, and T. Xia, “Vehicle detection from 3D Lidar using fully convolutional network,” in *Proc. Robot., Sci. Syst. XII*, AnnArbor, MI, USA, Jun. 2016. [Online]. Available: http://www.roboticsproceedings.org/rss12/

[49] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer. (Nov. 2016). “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size.” [Online]. Available: https://arxiv.org/abs/1602.07360

[50] S. L. Yu, T. Westfechtel, R. Hamada, K. Ohno, and S. Tadokoro, “Vehicle detection and localization on bird’s eye view elevation images using convolutional neural network,” in *Proc. IEEE Int. Symp. Saf., Secur. Rescue Robot. (SSRR)*, Oct. 2017, pp. 102–109.

[51] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. de la Escalera. (May 2018). “BirdNet: A 3D object detection framework from LiDAR information.” [Online]. Available: http://arxiv.org/abs/1805.01195

[52] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6517–6525.

[53] D. Feng, L. Rosenbaum, and K. Dietmayer. (2018). “Towards safe autonomous driving: Capture uncertainty in the deep neural network for Lidar 3D vehicle detection.” [Online]. Available: https://arxiv.org/abs/1804.05132

[54] B. Li, “3D fully convolutional network for vehicle detection in point cloud,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1513–1518.

[55] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1355–1361.

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

[57] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[58] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2015, pp. 922–928.

[59] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*. New York, NY, USA: Curran Associates, Inc., 2017, pp. 5099–5108.

[60] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. (Jan. 2018). "Dynamic graph CNN for learning on point clouds." [Online]. Available: https://arxiv.org/abs/1801.07829

[61] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[62] Y. Zhou and O. Tuzel. (Nov. 2017). "VoxelNet: End-to-end learning for point cloud based 3D object detection." [Online]. Available: https://arxiv.org/abs/1711.06396

[63] 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 Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 918–927.

[64] 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.

[65] J. Schlosser, C. K. Chow, and Z. Kira, "Fusing LIDAR and images for pedestrian detection using convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 2198–2205.

[66] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2117–2125.

[67] X. Du, M. H. Ang, S. Karaman, and D. Rus, "A general pipeline for 3D detection of vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 3194–3200, doi: 10.1109/ICRA.2018.8461232.

[68] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[69] C. Redondo-Cabrera, R. J. López-Sastre, Y. Xiang, T. Tuytelaars, and S. Savarese, "Pose estimation errors, the ultimate diagnosis," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 118–134.

[70] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. (Aug. 2017). "On calibration of modern neural networks." [Online]. Available: http://arxiv.org/abs/1706.04599

Eduardo Arnold received the B.S. degree in electrical engineering from the Federal University of Santa Catarina, Brazil, in 2017. He is currently pursuing the Ph.D. degree with the Warwick Manufacturing Group, The University of Warwick, U.K. He was an Exchange Student with the University of Surrey, supported by the Science Without Borders Program, in 2014. His research interests include machine learning, computer vision, and connected and autonomous vehicles.

Omar Y. Al-Jarrah received the B.S. degree in computer engineering from Yarmouk University, Jordan, in 2005, the M.Sc. degree in engineering from The University of Sydney, Sydney, Australia, in 2008, and the Ph.D. degree in electrical and computer engineering from Khalifa University, United Arab Emirates, in 2016. He was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, Khalifa University. He is currently a Research Fellow with the Warwick Manufacturing Group, The University of Warwick, U.K. His main research interests include machine learning, connected and autonomous vehicles, intrusion detection, big data analytics, and knowledge discovery in various applications. He has authored/co-authored several publications on these topics. He has served as a TPC member of several conferences, including the IEEE Globecom 2018. He was a recipient of several scholarships during his undergraduate and graduate studies.

Mehrdad Dianati has worked in the industry for more than nine years as a senior software/hardware developer and as the director of R&D. He was a Professor with the 5G Innovation Centre, University of Surrey, where he is currently a Visiting Professor. He is also a Professor of autonomous and connected vehicles with the Warwick Manufacturing Group, The University of Warwick. He has been involved in a number of national and international projects as the project leader and the work-package leader in recent years. He frequently provides voluntary services to the research community in various editorial roles. He has served as an Associate Editor for the IEEE Transactions on Vehicular Technology, *IET Communications*, and *Journal of Wireless Communications and Mobile* (Wiley).

Saber Fallah was a Research Associate and a Post-Doctoral Research Fellow with the Waterloo Centre for Automotive Research, University of Waterloo, Canada, and a Research Assistant with the Concordia Centre for Advanced Vehicle Engineering, Concordia University, Montreal, Canada. He is currently a Senior Lecturer (an Associate Professor) with the University of Surrey. He is also the Director of the Connected Autonomous Vehicles (CAV) Laboratory. He is also leading and contributing to several CAV research activities funded by the U.K. and European Governments (including, EPSRC, Innovate UK, and H2020) in collaboration with companies active in this domain. He has contributed significantly to the state-of-the-art research in the areas of connected autonomous vehicles and advanced driver-assistance systems.

David Oxtoby received the B.Eng. degree in electronic engineering from the University of York, U.K., in 1993. He worked in the field of telecommunication with Nortel Networks, from 1993 to 2002, before making a career change into automotive, in 2003, first working with Nissan on audio/navigation, telephone, and camera systems. Since 2013, he has been with the Electrical Research Team, Jaguar Land Rover, where he has been working on a wide variety of projects and is currently responsible for a team delivering new electrical technologies from initial idea to concept ready for production.

Alex Mouzakitis has over 15 years of technological and managerial experience, especially in the area of automotive embedded systems. In his previous position with JLR, he has served as the Head of the Model-based Product Engineering Department, responsible for the model-based development and automated testing standards and processes. He is currently the Head of the Electrical, Electronics and Software Engineering Research Department, Jaguar Land Rover. In his current role, he is responsible for leading a Multidisciplinary Research And Technology Department dedicated to deliver a portfolio of advanced research projects in the areas of human–machine interface, digital transformation, self-learning vehicle, smart/connected systems, and onboard/off-board data platforms.