Robust Multimodal Vehicle Detection in Foggy Weather Using Complementary Lidar and Radar Signals

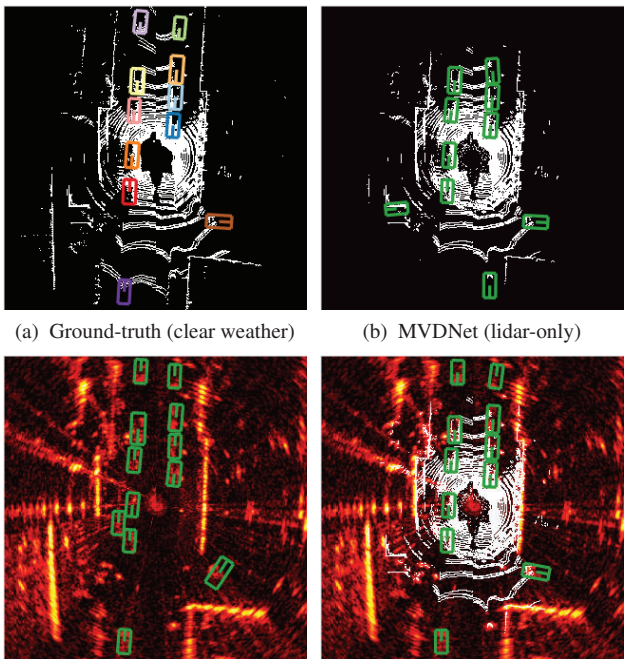
Kun Qian[1], Shilin Zhu[1], Xinyu Zhang[1], Li Erran Li[2]
[1]University of California San Diego [2]Columbia University
{kuq002,xyzhang,shz338}@eng.ucsd.edu erranli@gmail.com

Abstract

Vehicle detection with visual sensors like lidar and camera is one of the critical functions enabling autonomous driving. While they generate fine-grained point clouds or high-resolution images with rich information in good weather conditions, they fail in adverse weather (e.g., fog) where opaque particles distort lights and significantly reduce visibility. Thus, existing methods relying on lidar or camera experience significant performance degradation in rare but critical adverse weather conditions. To remedy this, we resort to exploiting complementary radar, which is less impacted by adverse weather and becomes prevalent on vehicles. In this paper, we present Multimodal Vehicle Detection Network (MVDNet), a two-stage deep fusion detector, which first generates proposals from two sensors and then fuses region-wise features between multimodal sensor streams to improve final detection results. To evaluate MVDNet, we create a procedurally generated training dataset based on the collected raw lidar and radar signals from the open-source Oxford Radar Robotcar. We show that the proposed MVDNet surpasses other state-of-the-art methods, notably in terms of Average Precision (AP), especially in adverse weather conditions. The code and data are available at https://github.com/qiank10/MVDNet.

1. Introduction

As the holy grail of autonomous driving technology, Full Driving Automation (Level 5) [20] relies on robust all-weather object detection, which provides accurate bounding boxes of surrounding objects even in the challenging adverse foggy weather condition. Nowadays, autonomous vehicles are equipped with multiple sensor modalities, such as camera, lidar, and radar [12, 6, 48, 3]. Fusing multimodal sensors overcomes any individual sensor's occasional failures and potentially yields more accurate object detection than using only a single sensor. Existing object detectors [10, 21, 52, 38] mainly fuse lidar and camera, which normally provide rich and redundant visual information. However, these visual sensors are sensitive to weather conditions and are not expected to work fully in harsh weather like fog [4, 26], making the autonomous perception systems unreliable. For example, Fig. 1a shows an example of a driving scenario with ground-truth vehicles labeled. Fig. 1b shows the detected vehicles using only lidar point cloud that is deteriorated by fog. Two farthest vehicles at the top are missing due to the occlusion of fog.



(a) Ground-truth (clear weather) (b) MVDNet (lidar-only)

(c) MVDNet (radar-only) (d) Complete MVDNet **(Ours)**

Figure 1. Performance overview of our proposed MVDNet. (a) 360° bird's eye view of the 3D lidar point cloud and ground-truth labels (colors represent different vehicles). The vehicle equipped with lidar and radar is at the center. In foggy weather, (b) lidar-only MVDNet misses vehicles at farthest range due to fog occlusion and misclassifies background points as vehicles; (c) radar-only MVDNet produces false alarms and inaccurate bounding boxes due to noisy radar data; (d) By deeply fusing lidar and radar, the complete MVDNet correctly detects vehicles.

Aside from lidar and camera, radar has been widely deployed on autonomous vehicles [6, 3, 58] and has the potential to overcome foggy weather. Specifically, radar uses millimeter-wave signals whose wavelength is much larger than the tiny particles forming fog, rain, and snow [14, 1], and hence easily penetrates or diffracts around them. However, radars in the existing autonomous driving datasets are still underexplored, mainly due to their significant data sparsity, as compared with camera and lidar. For example, the nuScenes dataset [6] has about 35K lidar points but only 200 radar points on average in each data frame. The main reason is that its radars use conventional electronically steerable

antenna array, which tends to generate beam patterns with wide beamwidth (3.2°-12.3°). In the DENSE [3] dataset, a proprietary radar is mounted on the front bumper of the vehicle. However, its angular field of view is only 35°. Fortunately, the recent Oxford Radar Robotcar [2] (ORR) deploys a radar with a rotating horn antenna, which has high directionality and much finer spatial resolution of $0.9°$, and is mechanically rotated to achieve 360° field of view. The ORR radar generates dense intensity maps, as shown in Fig. 1c, where each pixel represents the reflected signal strength. It creates a new opportunity for object detection in foggy weather condition.

Despite the richer information, the ORR radar is still significantly coarser and noisier than its visual counterpart, i.e., lidar, as showcased in Fig. 1a and 1c. As a result, if it is processed in the same way as the lidar point cloud, then false alarms and large regression errors show up. To robustly detect vehicles in foggy weather, one should take advantage of both lidar (fine granularity within visible range) and radar (immunity to foggy weather) while overcoming their shortcomings. To this end, we propose MVDNet, a multimodal deep fusion model for vehicle detection in adverse foggy weather condition. MVDNet consists of two stages. The first stage generates proposals from the lidar and radar separately. The second stage employs the adaptive fusion of the two sensors' features via attention and the temporal fusion using 3D convolutions. Such a late fusion scheme allows the model to generate sufficient proposals while focusing the fusion within the regions of interest (ROI). As shown in Fig. 1d, MVDNet can not only detect the vehicles occluded by fog in the lidar point clouds but also reject false alarms in the noisy radar intensity maps.

To validate MVDNet, we create a procedurally generated training dataset based on the raw lidar and radar signals from ORR. Specifically, we manually generate oriented bounding boxes for vehicles in the lidar point clouds, synchronize the radar and lidar with the knowledge of visual odometry, and simulate random fog effects using an accurate fog model proposed in DEF [3]. We compare MVDNet with the state-of-the-art lidar-alone detectors [55, 24, 46], or lidar and radar fusion [3]. Evaluation results show that MVDNet achieves notably better performance on vehicle detection in foggy weather condition while requiring $10\times$ less computing resource.

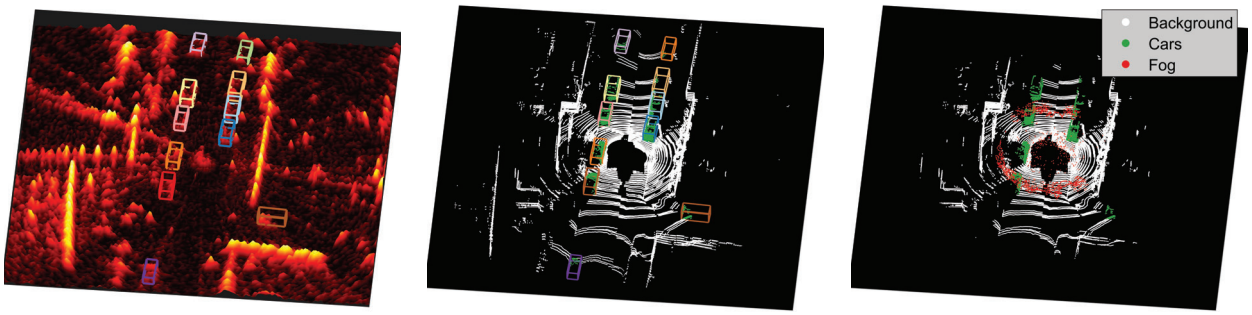
Our core contributions are two folds. First, we propose a deep late fusion detector that effectively exploits lidar and radar's complementary advantages. To our knowledge, MVDNet represents the *first vehicle detection system that fuses lidar and high-resolution* $360°$ *radar signals for vehicle detection.* Second, we introduce a labeled dataset with fine-grained lidar and radar point cloud in foggy weather condition. We assess MVDNet on the proposed dataset and demonstrate the effectiveness of the proposed fusion model.

2. Related Work

Vehicle detection from lidar signals. Depending on the representations of point clouds, lidar-based object detection falls into two categories. On the one hand, lidar data is formalized as point clouds by default and can be naturally processed by architectures designed for unordered point sets [39, 40]. Based on these architectures, end-to-end learning for raw point clouds is enabled [52, 53, 46, 24]. PointRCNN [46] extracts point-wise features with PointNet [39] and combines features at different stages to recognize foregrounds. It then generates proposals and refines final detection results. PointPillars [24] segments points into pillars, where pillar-wise features are calculated using PointNet to form a pseudo image. The image is then passed to a CNN backbone and SSD [31] detection head. However, point-wise features cannot be learned for areas occluded by adverse weather due to the absence of any point there. On the other hand, a lidar point cloud can be voxelized and processed by standard image detection models [55, 47, 57, 33]. PIXOR [55] segments points and generates an occupancy map for different heights. The voxel representation can be easily combined with other regular image data, e.g., from camera and lidar, and is exploited in MVDNet.

Denoising in foggy weather. Fog and haze reduce the data quality of visual sensors such as camera and lidar, due to loss of contrast [43, 5] and reduction in visible range [18, 4]. On the one hand, sophisticated dehazing methods [16, 11, 25, 32] for images have been proposed to benefit learning tasks [15, 43]. These methods either estimate a transmission map between foggy and clear images using hand-crafted [16, 11] or learned [25] priors or develop an end-to-end trainable model. On the other hand, little research has been done on lidar point cloud denoising [9, 17]. Due to the sparsity of lidar point cloud, existing denoising methods for dense 3D point cloud [41, 45, 19] cannot be directly applied to remove fog points. DROR [9] leverages dynamic spatial vicinity of points for denoising. Due to the lack of semantic information, it can mistakenly remove solitary reflections from objects. Heinzler *et al.* [17] proposed a CNN-based denoising model to understand and filter out fog effect. Nonetheless, existing denoising methods cannot compensate for the visibility reduction of lidar due to fog without extra information. In contrast, MVDNet combats foggy weather using high-resolution radar to complement the weather-sensitive lidar point cloud.

Vehicle detection with sensor fusion. Multimodal sensors provide redundant information, making it robust against sensor distortions due to internal noises and bad weather. Most fusion methods [21, 52, 38, 10, 27] are proposed for lidar and camera, due to their availability in common datasets [12, 48]. MV3D [10] aggregates proposals of multiple views. PointFusion [52] combines feature vectors of lidar and camera to predict 3D boxes of vehicles.



(a) (b) (c)

Figure 2. Fog effect on lidar point cloud. (a) Synchronized radar intensity map not impact by fog. (b) Raw point cloud truncated within 32 m. (c) Foggy point cloud with scattering fog points (red) and reduced visible range.

Radar is gaining traction recently as an additional modality for autonomous perception [34, 8, 35, 28, 37, 22]. In [29], the sparse and noisy radar points are projected on camera images to enhance depth estimation. In [23], the Doppler frequency shifts measured by radar is exploited to recognize pedestrians occluded in lidar's view. DEF [3] develops an early fusion detector with lidar, camera, and radar. However, DEF's radar has low quality, leading to inferior performance when radar works alone. Besides, the radar and camera of DEF have narrow angles of view, and the detector is specially designed for front-views, which is non-trivial to be adapted to 360° detection. RadarNet [54] fuses the sparse radar points and lidar point clouds at the early feature extraction stage via CNN to detect objects in 360° view and further associates sparse radar points with the detections to refine motion prediction. LiRaNet [44] also fuses sparse radar points with lidar point cloud and road map at an early stage to predict trajectories of detected vehicles. In contrast, MVDNet targets robust vehicle detection in foggy weather condition. To achieve it, we exploit a state-of-the-art imaging radar with much finer resolution than that used in RadarNet and LiRaNet, and propose an effective deep late fusion method to combine radar and lidar signals.

3. The MVDNet Design

3.1. Problem Statement and Overview

The adverse effects of fog have been well measured and modeled [4, 3, 26]. Fig. 2 exemplifies the effects, where we foggify a point cloud (i.e., Fig. 1b) from the ORR lidar (Velodyne HDL-32E lidar [49]), using the fog model in [3] with fog density of 0.05 m^{-1}. Due to its lower transmissivity than clear air, fog distorts lidar point clouds in two aspects: *(i)* Lasers reflected by distant objects are attenuated and become too weak to be acquired by lidar, resulting in *reduced visible range*. *(ii)* The opaque fog backscatters laser signals, resulting in *scattering fog points* (red points in Fig. 2c). These adverse effects can cause false alarms and misdetections, as shown in Fig. 1b. In contrast, fog is almost transparent to radar [14, 1]. But radar has intrinsically lower spatial resolution than lidar due to its longer signal wavelength and wide beamwidth. Therefore, to date, radar is mostly used for motion/speed tracking (Sec. 2). Fortunately, emerging imaging radars, such as the NavTech CTS350-X [36] used in ORR, enable point clouds with comparable resolution and density as a low-grade lidar. For example, Fig. 2a shows an example bird's eye view intensity map of the ORR radar. The prominent intensity peaks correspond to main objects on the road (e.g., vehicles, walls, etc.) and match their lidar counterparts well.

MVDNet essentially deep fuses radar intensity maps with lidar point clouds, to harness their complementary capabilities. As illustrated in Fig. 3, MVDNet consists of two stages. The region proposal network (MVD-RPN) extracts feature maps from lidar and radar inputs and generates proposals from them. The region fusion network (MVD-RFN) pools and fuses region-wise features of the two sensors' frames and outputs oriented bounding boxes of detected vehicles. We now introduce the detailed design of MVDNet.

3.2. MVD-RPN Backbone

Feature extractor. MVDNet uses two feature extractors with the same structure for lidar and radar inputs. But the number of feature channels of the lidar part is doubled due to more lidar input channels (Sec. 4.2). As shown in Fig. 4a, the feature extractor first uses 4 3×3 convolution layers to extract features at input resolution. It then downsamples the output by 2× via max-pooling and further extracts features at a coarser resolution. In the bird's eye view, vehicles only occupy small areas. Specifically, the vehicles in ORR have an average size of 2.5 m×5.1 m, which only occupies a 13×26 pixels area with an input resolution of 0.2 m. Downsampling the bird's eye view map makes the region-wise features vulnerable to quantization errors in the subsequent proposal generator. MVDNet thus upsamples the coarse-grained feature map via a transposed convolution layer and concatenates the output with the fine-grained feature map via a skip link. Each feature extractor is applied to all H input frames of the corresponding sensor and generates a set of H feature maps.

Proposal generator. As illustrated in Fig. 4b, the pro-

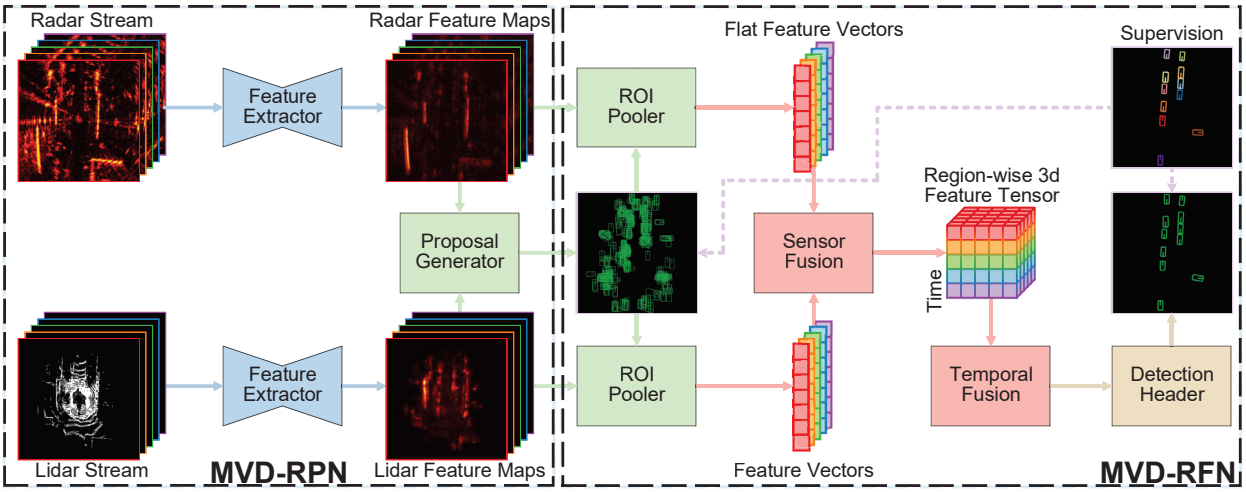


Figure 3. Overview of MVDNet. MVDNet takes the bird's eye view of both radar and lidar frames as input. It first extracts spatial feature maps of two sensors (*blue*), generates and merges oriented 2D proposals, and extracts region-wise features of two sensors via ROI pooling (*green*). A fusion network is used to combine the region-wise features for two sensors and across their temporal frames (*red*). The fused features are used to jointly detect and localize objects (*brown*).

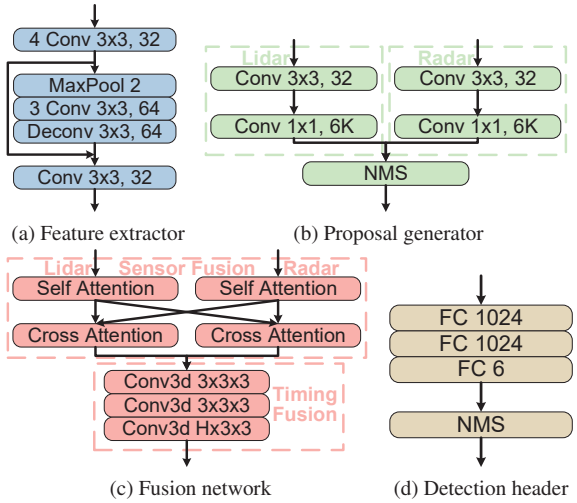


Figure 4. Details of MVDNet. The fusion network consists of a 2-stage attention block for sensor fusion and 3D CNN layers for temporal fusion. Each convolution layer is followed by batch normalization and a leaky ReLU layer.

posal generator takes the streams of H feature maps as input and generates proposals for MVD-RFN later. Since moving vehicles are at varying locations in different sensor frames, instead of generating proposals separately from the feature map of each frame, MVDNet concatenates feature maps of all frames of each sensor and fuses them via a convolution layer. To fully exploit the individual sensors, the fused feature map of each sensor is used separately to infer objectiveness scores and regress locations of proposals with K predefined anchors. Finally, the proposals generated by both sensors are merged via non-maximum suppression (NMS).

3.3. MVD-RFN Multimodal Fusion

The proposals generated by MVD-RPN are used in RoI poolers to create region-wise features. For each proposal, the pooling operation is applied to its region in the feature map of every frame and sensor, resulting in $2H$ $C \times W \times L$ feature tensors, where C is the number of channels in the feature maps and $W \times L$ is the 2D pooling size. MVDNet then fuses the feature tensors of each proposal via two steps, i.e., *sensor fusion* and *temporal fusion*, as shown in Fig. 4c.

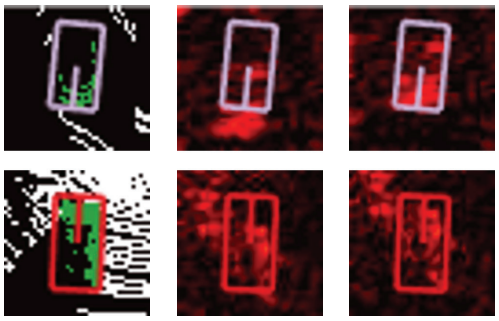
Sensor fusion. The sensor fusion merges the feature tensors of synchronized pair of lidar and radar frames. Intuitively, lidar and radar are not always of equal importance, and their contributions should be weighted accordingly. For example, a vehicle fully occluded by fog returns zero lidar points, and the lidar's feature tensors should thus be weighted less. In contrast, a strong peak of some background area in the radar intensity map may resemble the intensity peaks of vehicles. In such cases, the radar features around this area should be deweighted with the cue from the lidar features. MVDNet adaptively fuses lidar and radar features by extending the attention block in [50]. It takes as input the flattened feature tensor $\mathbf{x}_{in}$, calculates the similarity between two embedding spaces θ and ϕ, and uses the similarity to create an attention map for the third embedding space g to generate the residual output, i.e.,

$$\mathbf{x}_{out} = \sigma\left((\mathbf{W}_\theta \mathbf{x}_{in})^T \mathbf{W}_\phi \mathbf{x}_{in}\right) \mathbf{W}_g \mathbf{x}_{in} + \mathbf{x}_{in}, \quad (1)$$

where $\mathbf{W}_\theta, \mathbf{W}_\phi, \mathbf{W}_g$ are linearly transformation to the embedding spaces θ, ϕ, g respectively and σ represents the softmax function. As shown in Fig. 4c, each branch of the sensor fusion consists of two attention blocks. While the self attention applies attention within individual sensors, the cross attention further applies attention with the guidance of the counterpart sensor. Specifically, for either sensor $s_0 \in \{\text{lidar}, \text{radar}\}$ and its counterpart sensor s_1:

$$\mathbf{x}'_{s_0} = \sigma\left((\mathbf{W}_\theta \mathbf{x}'_{s_1})^T \mathbf{W}_\phi \mathbf{x}'_{s_1}\right) \mathbf{W}_g \mathbf{x}'_{s_0} + \mathbf{x}'_{s_0}, \quad (2)$$

where $\mathbf{x}'$ represents the feature vector output from the self attention of each branch. The output feature vectors from



(a) (b) (c)

Figure 5. Data preprocessing. (a) Lidar points and ground truth bounding boxes of vehicles. (b) Large offsets between radar intensity peaks and labels of vehicles due to asynchronization. (c) Accurate alignment between radar intensity peaks and labels of vehicles after data preprocessing.

the cross attention of the two sensors are reshaped back to feature tensors and concatenated for the temporal fusion.

Temporal fusion. The temporal fusion further merges the attended feature tensors of different frames. As shown in Fig. 4c, instead of using timing and memory intensive recurrent structures, MVDNet concatenates attended feature tensors of different frames along a new dimension to form 4D feature tensors and applies 3D convolution layers to allow information exchange along the time dimension. The last convolution layer compresses the time dimension and outputs the fused feature tensor. MVDNet then flattens the fused feature tensor and passes it to fully-connected layers to infer objective scores and regress locations of final detections, as shown in Fig. 4d.

4. Implementation

4.1. Loss Function and Training

Loss function. We adopt two commonly used multi-task loss functions as in [13] for both MVD-RPN and MVD-RFN. Specifically, we use binary cross-entropy loss for the classification task $L_{BCE,cls}$ and smooth l_1 loss for the bounding box regression task $L_{l1,reg}$. The regression targets are transformations from anchor boxes to proposals for MVD-RPN, and from proposals to final detections for MVD-RFN. We represent the orientation of bounding boxes with an angle in 180° and a binary direction parameter. For MVD-RPN, we omit the classification of directions to reduce the number of oriented anchors by half in order to save computing resource. For MVD-RFN, we use a second cross-entropy loss, $L_{BCE,dir}$, for direction classification. Overall, the total loss of MVDNet is:

$$\begin{aligned} L_{total} = & L^{RPN}_{BCE,cls} + L^{RPN}_{l_1,reg} \\ & + L^{RFN}_{BCE,cls} + L^{RFN}_{l_1,reg} + L^{RFN}_{BCE,dir}. \end{aligned} \tag{3}$$

Training details. The ORR dataset contains 8,862 samples, which are split into 7,071 for training and 1,791 for testing, without geographic overlapping. We set the RoI for the sensors to [-32,32] $\times$ [-32,32] m and run bird's eye view projection with a 0.2 m quantization. Similar to PIXOR [55], we set the height range to [-2.5,1] m, divide all lidar points into 35 slices with a bin size of 0.1 m, and compute an intensity channel. On the other hand, the radar only has one intensity channel. Thus, the input lidar and radar representations have dimensions of $320\times320\times36$ and $320\times320\times1$, respectively. At most 5 frames of each sensor, consisting of the current and historical 4 frames, are used as input. To train models with foggy weather, we randomly foggify the lidar point clouds in the training samples using the fog model in DEF [3] with a probability of 0.5. Specifically, for each lidar point, the fog model calculates its maximum visible distance given the fog density. If the maximum distance is smaller than the real distance, the point is either lost or relocated as a scatter point. The fog density is uniformly selected from the typical range $[0.005, 0.8]m^{-1}$.

We implement MVDNet with Detectron2 [51], an open-source codebase for RCNN-based object detectors. For MVD-RPN, the anchors are set to 3.68 m $\times$ 7.35 m, and orientations in -90^o, -45^o, 0^o and 45^o. The matching of positive and negative samples uses thresholds of 0.55 and 0.45, respectively. The IoU threshold of NMS is set to 0.7, and 1000 proposals are kept during training while 500 during inference. For MVD-RFN, the pooling size of the RoI poolers is set to 7×7. The batch size after pooling is 256. The IoU threshold of NMS is set to 0.2. We use the SGD optimizer with an initial learning rate of 0.01, decay the learning rate by a factor of 0.1 every 40K iterations, and train the model for 80K iterations from scratch. Each iteration takes the input with a batch size of 1. Besides, we train a compressed version of MVDNet, named as MVDNet-Fast, where we reduce the size of the region fusion network by $8\times$, by reducing the number of channels (features) in the convolution and FC layers by $8\times$.

4.2. Dataset Preparation

The original ORR data are collected by a vehicle equipped with a NavTech CTS350-X radar [36] at the roof center, co-located with two Velodyne HDL-32E [49] lidars whose point clouds are combined. In our dataset, we manually generate the ground-truth labels based on the ORR lidar point clouds. Specifically, we create 3D bounding boxes of vehicles in one out of every 20 frames (i.e., 1 s) using Scalabel [56], an open-source annotation tool. Labels of the remaining 19 frames are interpolated using the visual odometry data provided in ORR and manually adjusted to align with the corresponding vehicles.

The ORR radar [36] scans the 360^o field of view at a step of 0.9^o every 0.25 s and lidar [49] at a step of 0.33^o every 0.05 s. The radar and lidar scanning results are transformed into a 2D intensity map and 3D point cloud, respectively. Both share the same coordinate origin. However, the radar's *considerable scanning delay* and the *lack of synchronization with the lidar* cause non-negligible misalignment especially

Method	Train	Clear+Foggy						Clear-only						#Params
	Test	Clear			Foggy			Clear			Foggy			
	IoU	0.5	0.65	0.8	0.5	0.65	0.8	0.5	0.65	0.8	0.5	0.65	0.8	
PIXOR [55]		72.76	68.25	41.15	62.59	58.89	35.74	70.97	67.15	40.62	61.77	58.27	35.70	2,135K
PointRCNN [46]		78.18	73.75	45.70	69.65	65.64	41.58	78.22	72.78	43.44	68.74	63.99	37.64	3,887K
PointPillars [24]		85.74	82.99	58.33	72.80	70.34	48.55	85.83	82.87	60.59	71.28	68.31	47.82	4,815K
DEF [3]		86.60	78.18	46.20	81.44	72.46	41.05	85.88	78.11	44.16	71.81	63.74	32.38	5,210K
DEF+MVD-RFN		87.69	85.52	67.49	82.23	79.18	61.05	86.18	84.07	69.57	71.83	69.96	56.72	13,730K
MVDNet (**Ours**)		**90.89**	**88.82**	**74.63**	**87.40**	**84.61**	**68.88**	87.22	**86.06**	**72.63**	**77.98**	**75.89**	**61.55**	8,591K
MVDNet-Fast (**Ours**)		88.99	86.20	68.30	85.58	82.25	62.76	**88.91**	85.96	68.15	76.30	73.97	56.96	977K

Table 1. Overall performance: AP of oriented bounding boxes in bird's eye view. Bold numbers represent the best score among all the methods. Underlined numbers represent the best one among the baseline methods.

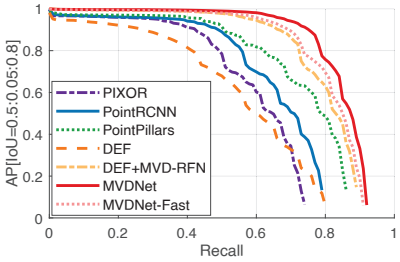


Figure 6. Precision-recall curves.

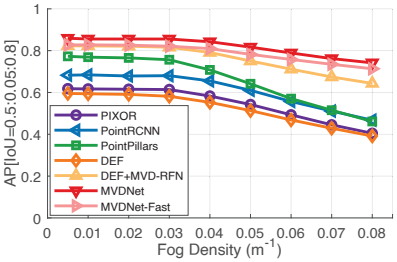


Figure 7. Impact of fog density.

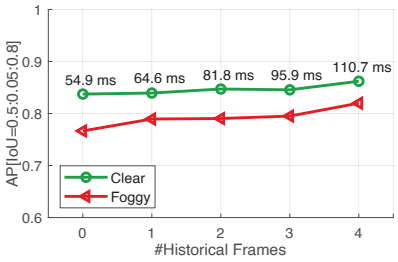


Figure 8. Impact of historical data.

when moving. For example, the lidar points and labels of two vehicles in Fig. 1a are shown in Fig. 5a and their radar intensity maps in Fig. 5b, where large offsets exist between prominent radar intensity peaks and labels.

To compensate for the misalignment, we estimate the movement $\vec{\delta_l} = (\delta_x, \delta_y)^T$ and rotation δ_θ of the radar relative to the beginning of its successive scans via SLAM [7]. When scanning a point $\vec{p_l} = (p_x, p_y)^T$ in free space, the instantaneous relative location and orientation of the radar are linearly interpolated as $t\vec{\delta_l}$ and $t\delta_\theta$ at time t, where $t \approx \frac{\angle \vec{p_l}}{2\pi} T$ is the approximated scanning time of point $\vec{p_l}$ and T is the scanning interval. We calculate the relative distance $d' = ||\vec{p_l} - t\vec{\delta_l}||$ and angle $\theta' = \angle \vec{p_l} - t\delta_\theta$ between the point $\vec{p_l}$ and the radar. The intensity at $\vec{p_l}$ is corrected as $I(p_x, p_y) = I(d', \theta')$.

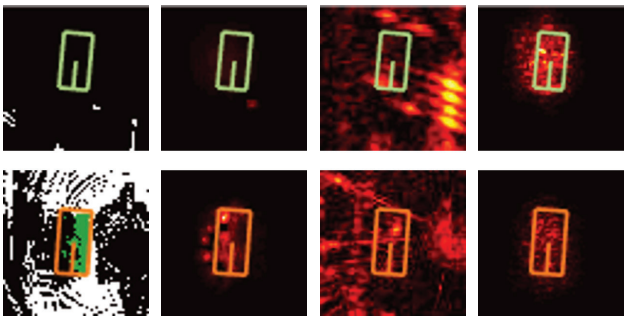
To synchronize the radar and lidar, instead of pairing each radar scan with the closest lidar scan in time, we aggregate all $N = 5$ lidar scans during this radar scan interval. Specifically, we select a sector of points from each lidar scan, where the selected sector is scanned by both lidar and radar at approximately the same time. Formally, for a radar frame and its simultaneous $N = 5$ lidar frames, a point $\vec{p}$ in the i-th lidar frame is selected only if it falls in the sector $[\frac{i-1}{N+1}\pi, \frac{i+1}{N+1}\pi]$. Finally, all selected points are transformed into the radar coordinate and combined as a lidar frame. Fig. 5c showcases the vehicles after correction, whose radar intensity peaks and labels are well aligned.

5. Experiment

5.1. Baseline Comparison

We validate MVDNet in both clear and foggy weather conditions, in terms of average precision (AP) using the CoCo evaluation framework [30]. We compare MVDNet against existing lidar-only detectors (PIXOR [55], PointRCNN [46], and PointPillars [24]), and the lidar-radar fusion method in DEF [3]. Besides, we implement MVDNet-Fast (Sec. 4.1) with a smaller model size and DEF+MVD-RFN which combines the feature extractor of DEF with the proposal generator and region fusion network of MVDNet.

As shown in Tab. 1, MVDNet and MVDNet-Fast consistently outperform the other detectors within various training/testing settings and IoUs, thanks to the fusion model design. The trade-off between the detector's cost and performance can thus be made by selecting a proper model size. While PointPillars yields the highest performance among all three lidar-only detectors, its performance is still significantly worse than MVDNet in foggy condition due to the reduced range and scattering effect. With additional inputs from radar, DEF can detect more vehicles, especially when both training and testing sets contain foggy data. However, even with radar, DEF is still worse than PointPillars in terms of localization accuracy, as indicated by AP with higher IoU thresholds. The main reason is that DEF is specialized for front-view images. Specifically, DEF preprocesses the input images and creates local entropy maps, which embodies dense information for front-view images, but sparse and isolated information for bird's eye view images. In contrast, with MVD-RFN appended, DEF+MVD-RFN effectively fuses lidar and radar data and extracts more useful features for accurate detection. Fig. 6 further shows the fine-grained precision-recall curves of all detectors with IoU averaged over $[0.5, 0.8]$. In foggy weather condition, MVDNet shows significant advantages over other detectors,



(a) Lidar input (b) Lidar gradient (c) Radar input (d) Radar gradient

Figure 9. Comparison of contribution of lidar and radar inputs to fused features of detected vehicles with lidar's visible range reduced by fog. The top vehicle (green) is outside the lidar's visible range, while the bottom one (orange) in the lidar's visible range.

justifying the robustness of MVDNet's late fusion design.

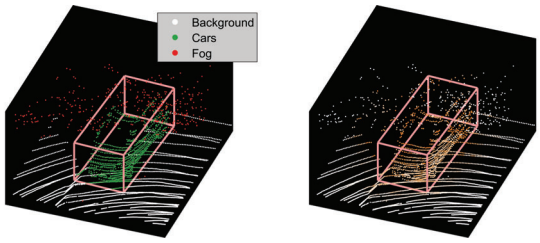
In practice, foggy weather is much rarer than normal weather, leading to an imbalance of data. We further train all detectors using only clear lidar point clouds and radar intensity maps. The results are shown in Tab. 1. Comparing to training using both clear and foggy lidar point clouds, we observe slight performance loss for clear cases and a significant loss for foggy cases. The result indicates that augmenting the clear data with foggification is crucial for robust all-weather detection.

To better understand the sensor fusion of MVDNet, we visualize the gradients of fused feature vectors of detected vehicles with respect to lidar and radar inputs. Fig. 9 compares the gradients of two vehicles in Fig. 1a that are in and outside of the visible range of the lidar respectively. For the vehicle within the visible range of the lidar (orange), both lidar and radar inputs contribute to the fused feature vector of the vehicle. In contrast, for the vehicle outside of the visible range of the lidar (green), only radar provides useful information and contributes to the fused feature vector. It validates the effectiveness of the proposed multimodal sensor fusion against the reduced visible range of lidar. Fig. 10 further compares the gradients of the feature vector of a vehicle in Fig. 1a with respect to its vehicle points and closed fog points. Most fog points contribute nearly zero gradients to the feature vector of the detected vehicle, demonstrating the denoising capability of MVDNet.

Fig. 11 shows detection results of different detectors in two scenes. All three lidar-only detectors miss some vehicles and mistakenly recognize some background lidar points as vehicles. In contrast, MVDNet detects vehicles with the highest accuracy and least regression errors, thanks to the design of deep late fusion.

5.2. Ablation on MVDNet Input

Impact of fog density. We now evaluate the performance of all detectors against various fog densities. Fog reduces the visible range of lidar. For example, the densest fog



(a) Lidar point cloud (b) Point-wise gradient

Figure 10. Comparison of the contribution of vehicle points and fog points to learned features of a detected vehicle. (a) An example vehicle (green) impacted by fog (red). (b) Vehicle points have a large contribution (orange) to the learned features of MVDNet, while fog points have little effect.

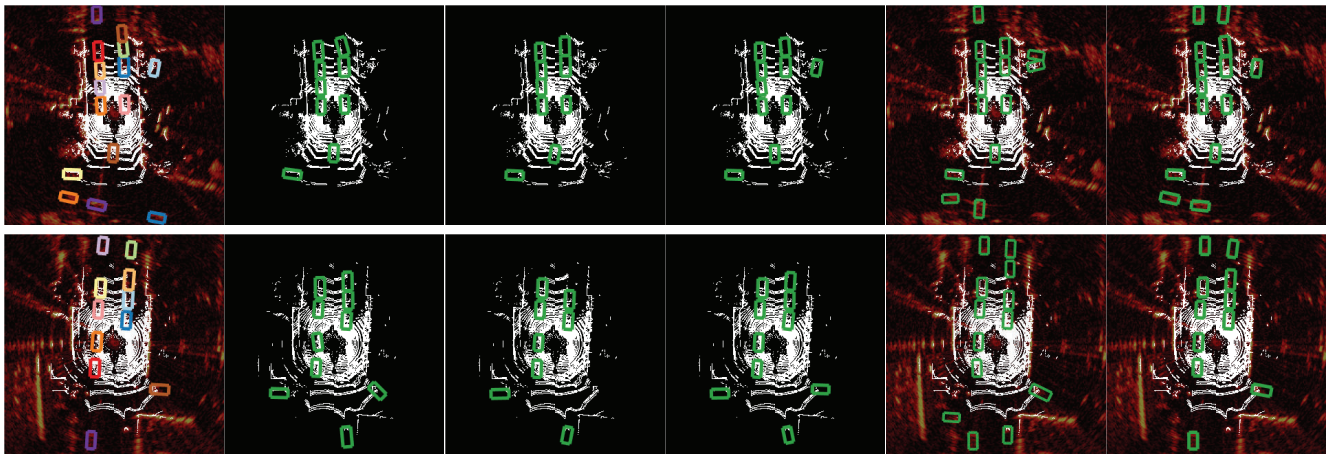
with a density of 0.08 m^{-1} reduces the visible range of the Velodyne HDL32-E lidar within only 15 m. As a result, detectors relying on lidar data will experience a significant performance drop. This is verified in Fig. 7, which shows the AP of all detectors working with common fog densities from 0.005 m^{-1} to 0.08 m^{-1}. While the performance of all detectors drops with the increase of the fog density, MVDNet has the lowest dropping rate and still maintains an AP around 0.75 with the densest fog, with the help of extra information from radar. In contrast, the 3 detectors using only the lidar have a significant performance drop where the AP reaches below 0.5 when the fog density reaches 0.08 m^{-1}.

Impact of historical information. Historical data helps detectors by encompassing the temporal correlation of samples. To evaluate the impact of historical information in our MVDNet design, we vary the number of historical lidar and radar frames. As shown in Fig. 8, whereas MVDNet consistently receives performance gain with the increase of the number of frames on both clear and foggy testing set, the gain on the foggy case is more prominent, as the visible range of lidar is "extended" with the area visible in the past but occluded at present. The runtime of our proposed detector increases from 54.9 ms (18 FPS) to 110.7 ms (9 FPS) linearly as the number of historical frames increases from 0 to 4. Therefore, the history length can be a design knob to navigate the trade-off between runtime and performance.

Data synchronization. To validate the contribution from data synchronization in Sec. 4.2, we create the dataset without correcting the misalignment between lidar and radar. As shown in Tab. 2, the AP of MVDNet drops by about 9% on average. The larger the IoU threshold is, the more the performance drops. We conjecture that while the deep network can learn to correct misalignment implicitly, it is still insufficient to compensate for the large misalignment between lidar and radar data, as in Fig. 5. This result demonstrates the necessity of explicit data synchronization.

5.3. Ablation on MVDNet Architecture

We show an extensive ablation study of MVDNet in terms of contribution of individual fusion modules, contri-



Ground-truth PIXOR [55] (Li) PointRCNN [46] (Li) PointPillars [24] (Li) DEF [3] (Li+Ra) MVDNet (**Ours**, Li+Ra)

Figure 11. Examples of 360° detection results of different detectors. The ground-truth is in various colors while the detection is in green.

Method	IoU	0.5	0.65	0.8
No Data Sync (Sec. 4.2)		85.05	78.54	56.74
No Fusion (Sec. 3.3)		87.40	84.45	70.20
No Temporal Fusion (Sec. 3.3)		88.19	85.75	69.72
No Sensor Fusion (Sec. 3.3)		87.89	85.59	70.61
No Self Attention (Sec. 3.3)		88.19	85.88	71.41
No Cross Attention (Sec. 3.3)		88.31	85.95	70.88
MVDNet (**Ours**)		**89.15**	**86.72**	**71.76**
Radar-Only		73.04	68.27	43.25
Lidar-Only		82.28	80.72	67.83
Lidar Reconstruction		85.04	82.73	67.81

Table 2. Ablation study: AP of oriented bounding boxes in bird's eye view (averaged over both clear and fog testing sets).

bution of different sensors, and comparison between fusion at different stages. Tab. 2 shows the AP of different ablation schemes, averaged over both clear and foggy testing sets.

Individual fusion modules. MVDNet fuses lidar and radar data via both sensor fusion and temporal fusion (Sec. 3.3). To evaluate the benefits from the fusion, we replace each fusion block with a single convolution layer with the same input and output shape. As shown in Tab. 2, MVDNet achieves average gains of 1.3%, 1.2% and 1.8% with temporal fusion, sensor fusion and both, respectively. In addition, the gains from individual self and cross attention are 0.8% and 0.7% respectively, demonstrating that both attention modules help sensor fusion.

Different sensors. We then evaluate the contribution of sensors. Specifically, we keep the branch of one sensor and remove the other one in both MVD-RPN and the sensor fusion block in MVD-RFN. As shown in Tab. 2, the radar-only model has a significant performance drop comparing with the complete MVDNet, due to the coarse granularity and lack of height information of the radar. In comparison, the performance of the lidar-only model drops by 5.6% on average, mainly due to the adverse impact of fog.

Fusion at different stages. To validate the late fusion of multimodal sensors in MVDNet compared to early fusion, we prepend a standard U-Net [42] to the lidar-only model. The U-Net takes as input both radar and foggy lidar data and outputs the reconstructed lidar data. The output is then fed into the lidar-only model for detection. We first train the U-Net with a binary cross-entropy loss between the occupancy maps and a smooth l_1 loss between the intensity map of the reconstructed and clear lidar data. Then we connect the U-Net and the lidar-only model and jointly train both with the loss function in Sec. 4.1. As shown in Tab. 2, while the lidar reconstruction scheme achieves higher performance than the radar-only and lidar-only models, its AP is still lower than MVDNet by about 4%, indicating that the early fusion is less effective than the late fusion. It is mainly due to the low data quality of radar compared with lidar, making the explicit reconstruction of lidar data ineffective.

6. Conclusion

We have introduced MVDNet to enable vehicle detection under adverse foggy weather condition. MVDNet exploits complementary advantages of lidar and radar via deep late fusion across both the sensing modality and time dimensions. To evaluate MVDNet, we introduce a novel procedurally generated training dataset with spatially fine-grained mechanic radar and lidar. Experimental results show that MVDNet achieves consistently high detection accuracy than existing lidar-alone or multimodal approaches, especially in foggy weather condition. In the future, we plan to integrate a more diverse set of sensors, e.g., Doppler radar and RGBD camera, and explore network compression methods to enable real-time ($\geq$30 FPS) vehicle detection.

Acknowledgments

We sincerely thank the anonymous reviewers and ACs for their insightful feedback. This work was supported in part by the US National Science Foundation through NSF CNS-1925767, CNS-1901048.

References

[1] Nezah Balal, Gad A Pinhasi, and Yosef Pinhasi. Atmospheric and fog effects on ultra-wide band radar operating at extremely high frequencies. *Sensors*, 16(5):751, 2016. 1, 3

[2] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *Proceedings of the IEEE ICRA*, 2020. 2

[3] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE CVPR*, 2020. 1, 2, 3, 5, 6, 8

[4] Mario Bijelic, Tobias Gruber, and Werner Ritter. A benchmark for lidar sensors in fog: Is detection breaking down? In *Proceedings of the IEEE IV*, 2018. 1, 2, 3

[5] Mario Bijelic, Tobias Gruber, and Werner Ritter. Benchmarking image sensors under adverse weather conditions for autonomous driving. In *Proceedings of the IEEE IV*, 2018. 2

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE CVPR*, 2020. 1

[7] Sarah H Cen and Paul Newman. Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions. In *Proceedings of the IEEE ICRA*, 2018. 6

[8] Simon Chadwick, Will Maddetn, and Paul Newman. Distant vehicle detection using radar and vision. In *Proceedings of the IEEE ICRA*, 2019. 3

[9] Nicholas Charron, Stephen Phillips, and Steven L Waslander. De-noising of lidar point clouds corrupted by snowfall. In *Proceedings of the IEEE CRV*, 2018. 2

[10] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE CVPR*, 2017. 1, 2

[11] Raanan Fattal. Dehazing using color-lines. *ACM ToG*, 34(1):1–14, 2014. 2

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE CVPR*, 2012. 1, 2

[13] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE ICCV*, 2015. 5

[14] Yosef Golovachev, Ariel Etinger, Gad A Pinhasi, and Yosef Pinhasi. Millimeter wave high resolution radar accuracy in fog conditions-theory and experimental verification. *Sensors*, 18(7):2148, 2018. 1, 3

[15] M Hassaballah, Mourad A Kenk, Khan Muhammad, and Shervin Minaee. Vehicle detection and tracking in adverse weather using a deep learning framework. *IEEE TITS*, 2020. 2

[16] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE TPAMI*, 33(12):2341–2353, 2010. 2

[17] Robin Heinzler, Florian Piewak, Philipp Schindler, and Wilhelm Stork. Cnn-based lidar point cloud de-noising in adverse weather. *IEEE RA-L*, 5(2):2514–2521, 2020. 2

[18] Robin Heinzler, Philipp Schindler, Jürgen Seekircher, Werner Ritter, and Wilhelm Stork. Weather influence and classification with automotive lidar sensors. In *Proceedings of the IEEE IV*, 2019. 2

[19] Pedro Hermosilla, Tobias Ritschel, and Timo Ropinski. Total denoising: Unsupervised learning of 3d point cloud cleaning. In *Proceedings of the IEEE ICCV*, 2019. 2

[20] SAE international. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE International,(J3016)*, 2016. 1

[21] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *Proceedings of the IEEE/RSJ IROS*, 2018. 1, 2

[22] Hongwu Kuang, Xiaodong Liu, Jingwei Zhang, and Zicheng Fang. Multi-modality cascaded fusion technology for autonomous driving. *arXiv preprint arXiv:2002.03138*, 2020. 3

[23] Seong Kyung Kwon, Sang Hyuk Son, Eugin Hyun, Jin-Hee Lee, and Jonghun Lee. Radar-lidar sensor fusion sheme using occluded depth generation for pedestrian detection. In *Proceedings of the IEEE CSCI*, 2017. 3

[24] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE CVPR*, 2019. 2, 6, 8

[25] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE ICCV*, 2017. 2

[26] You Li, Pierre Duthon, Michèle Colomb, and Javier Ibanez-Guzman. What happens for a tof lidar in fog? *arXiv preprint arXiv:2003.06660*, 2020. 1, 3

[27] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the Springer ECCV*, 2018. 2

[28] Teck-Yian Lim, Amin Ansari, Bence Major, Daniel Fontijne, Michael Hamilton, Radhika Gowaikar, and Sundar Subramanian. Radar and camera early fusion for vehicle detection in advanced driver assistance systems. In *Proceedings of the NeurIPS Workshop on Machine Learning for Autonomous Driving*, 2019. 3

[29] Juan-Ting Lin, Dengxin Dai, and Luc Van Gool. Depth estimation from monocular images and sparse radar data. *arXiv preprint arXiv:2010.00058*, 2020. 3

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the Springer ECCV*, 2014. 6

[31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the Springer ECCV*, 2016. 2

[32] Xiaohong Liu, Yongrui Ma, Zhihao Shi, and Jun Chen. Griddehazenet: Attention-based multi-scale network for image dehazing. In *Proceedings of the IEEE ICCV*, 2019. 2

[33] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE CVPR*, 2018. 2

[34] Michael Meyer and Georg Kuschk. Deep learning based 3d object detection for automotive radar and camera. In *Proceedings of the IEEE EuRAD*, 2019. 3

[35] Ramin Nabati and Hairong Qi. Rrpn: Radar region proposal network for object detection in autonomous vehicles. In *Proceedings of the IEEE ICIP*, 2019. 3

[36] NavTech. CTS350-X radar. `https://navtechradar.com/solutions/clearway/`, 2020. 3, 5

[37] Felix Nobis, Maximilian Geisslinger, Markus Weber, Johannes Betz, and Markus Lienkamp. A deep learning-based radar and camera sensor fusion architecture for object detection. In *Proceedings of the IEEE Workshop Sensor Data Fusion: Trends, Solutions, Applications*, 2019. 3

[38] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE CVPR*, 2018. 1, 2

[39] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE CVPR*, 2017. 2

[40] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the NeurIPS*, 2017. 2

[41] Marie-Julie Rakotosaona, Vittorio La Barbera, Paul Guerrero, Niloy J Mitra, and Maks Ovsjanikov. Pointcleannet: Learning to denoise and remove outliers from dense point clouds. In *Computer Graphics Forum*, volume 39, pages 185–203, 2020. 2

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the MICCAI*, 2015. 8

[43] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *Springer IJCV*, 126(9), 2018. 2

[44] Meet Shah, Zhiling Huang, Ankit Laddha, Matthew Langford, Blake Barber, Sidney Zhang, Carlos Vallespi-Gonzalez, and Raquel Urtasun. Liranet: End-to-end trajectory prediction using spatio-temporal radar fusion. *arXiv preprint arXiv:2010.00731*, 2020. 3

[45] Ju Shen and Sen-Ching S Cheung. Layer depth denoising and completion for structured-light rgb-d cameras. In *Proceedings of the IEEE CVPR*, 2013. 2

[46] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE CVPR*, 2019. 2, 6, 8

[47] Martin Simon, Stefan Milz, Karl Amende, and Horst-Michael Gross. Complex-yolo: Real-time 3d object detection on point clouds. *arXiv:1803.06199*, 2018. 2

[48] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE CVPR*, 2020. 1, 2

[49] Velodyne. HDL-32E: High resolution real-time 3d lidar sensor. `https://velodynelidar.com/products/hdl-32e/`, 2020. 3, 5

[50] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *Proceedings of the AAAI*, 2020. 4

[51] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 5

[52] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE CVPR*, 2018. 1, 2

[53] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018. 2

[54] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. Radarnet: Exploiting radar for robust perception of dynamic objects. *arXiv preprint arXiv:2007.14366*, 2020. 3

[55] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE CVPR*, 2018. 2, 5, 6, 8

[56] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018. 5

[57] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE CVPR*, 2018. 2

[58] Julius Ziegler, Philipp Bender, Markus Schreiber, Henning Lategahn, Tobias Strauss, Christoph Stiller, Thao Dang, Uwe Franke, Nils Appenrodt, Christoph G Keller, et al. Making bertha drive—an autonomous journey on a historic route. *IEEE ITS magazine*, 6(2):8–20, 2014. 1