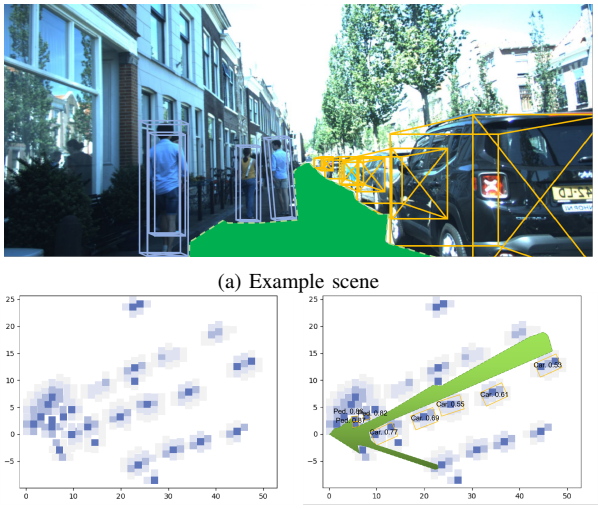
2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

MTRadSSD: A Multi-Task Single-Stage Detector for Object Detection and Free Space Analysis in Radar Point Clouds*

Yinbao Li, Songshan Yu, Dongfeng Wang and Jingen Jiao

***Abstract*— Environmental perception tasks such as object detection and free space detection based on 3+1D radar severely suffer from the disorder and sparsity of point cloud. To tackle this problem, we propose a novel Multi-Task Radar-based Single Stage Detector, termed MTRadSSD, where we adopt instance-aware sampling strategies to discover multi-class road users and propose an occupancy map tool based on kernel density estimation (KDE) to make predictions in bird's eye view (BEV). The denoised occupancy map also plays key role in generating polygon represented free space in the scene. As a result, our elaborated sampling strategies effectively retained useful semantic information and narrowed the difference of detection performance across object categories. Meanwhile, our MTRadSSD outperforms those state-of-the-art approaches in terms of real-time requirement and detection accuracy. In detail, the proposed method achieves an satisfactory speed of ˜16.7 ms per frame in experiments on the public radar point cloud dataset View-of-Delft (VOD). With IoU thresholds 0.5/0.25/0.25 the average prediction precision (AP) of easy-level objects (cars, pedestrians and cyclists) reaches at competitive 52.2%, 61.1%, 86.3%, respectively, while mean IoU of free space is 87.8%. Especially, the occupancy map also makes difference in improving prediction precision of object orientation dramatically to averaged 64.0%.**



(a) Example scene

(b) Denoised map tool (c) Model prediction

Fig. 1. Example scenario within VOD dataset. Our model only processes radar point cloud, and in the example scene, we plot 3D boxes of objects and drivable space in the camera data to enable a direct visualisation of detection results.

I. INTRODUCTION

As one of the most important sensors in autonomous driving, 3+1D mmw radar has salient strengths when compared with LiDAR and camera. Specifically, point cloud data of mmw radar carry rich information of environment, e.g. Doppler and height measurements, and its ability for space penetration facilitates wider dynamic landscapes in free space detection [1], [2]. There are many mature deep learning approaches to object detection using point cloud and they can be classified into three categories according to how they encode features of the data. In detail, the first type is voxel- or pillar-based, where voxelization is adopted to extract 3D features of points and voxels and pillarization is used to obtain 2D features; the second type is point-based and sampling or voting strategies are usually employed to select representative points; the last type is point-voxel-based methods, which make use of both point-wise and voxel-wise features. Generally speaking, voxel-based methods have intrinsic deficiency such as quantization loss [3] and point-based methods face with issues like poor memory locality [4].

*This work was supported by Joospeed Electronics.

Yinbao L., Songshan Y., Dongfeng W. and Jingen J. are with R&D Department of Jiaxing Joospeed Electronic Technology Co.,Ltd, 100020, Beijing, China {liyinbao, yusongshan, wdf, jiaojingen}@tsmtc.com
Corresponding author: Yinbao L. zhousidadi@126.com

In this paper, we firstly discuss point-based object detection methods that work with raw point cloud data only and the amount of point cloud of radar is so small that it is promising to achieve real-time performance. Afterwards, to improve detection accuracy of objects and free space, we design effective strategies to select representative points and inspired by [5], introduce occupancy evidence tools to enhance model performance. Finally, we test our Multi-Task Radar-based Single Stage Detector (MTRadSSD) on public radar dataset View-Of-Delft (VOD) [6], as shown in Fig. 1. In summary, our contributions are list as follows:

- Considering point clouds' distribution pattern of each type of specific road users: cars, pedestrians and cyclist, we designed mmw radar-specific data sampling strategies, which effectively narrowed the difference of detection performance among these objects.
- We utilized point features such as density to compute local occupancy evidence and used Kernel Density Estimation (KDE) to generate a map tool to represent global occupancy. The map tool made key contributions in multiple detection tasks.
- Our MTRadSSD realized multi-task detection in real time on public dataset with SOTA performance and specifically, the detection accuracy of object orientation was higher than those of SOTA methods.

II. RELATED WORK

In this section, we will first discuss point-based object detection networks in terms of their encoding methods and backbones & detection heads. Aiming to design a multi-task framework, we will also talk about state-of-the-art free space detection methods.

A. Point-based Object Detection

Essentially, it is challenging to use radar point cloud only in deep learning methods for object detection because of its sparsity and indistinct object characteristics. Although handful researches such as [7] attempts semantic segmentation on radar point cloud and [8] utilizes the range-azimuth-Doppler (RAD) tensor to do object detection, these researches are limited to relatively narrow selection of datasets and understanding of scenes. In environmental perception, many mature networks have been proposed to process LiDAR point cloud, which is inspiring and referential to radar point cloud research.

1) Encoding Methods: Point-based neural networks [9]-[11] directly extract and aggregate point-wise features via PointNet [12] and its variants [13]-[17]. Two-stage 3D object detection methods such as PointRCNN [10] firstly identify foreground points and then encode point-wise features to regress 3D bounding boxes within rich semantics. One-stage 3D detectors such as VoteNet [9] and 3DSSD [11] are based on point selection. Specifically, VoteNet applies voting mechanism (i.e. Hough Voting) to predict instance centroids and 3DSSD adopts sampling strategy considering the Farthest Point Sampling (FPS) on feature and Euclidean space. IA-SSD [3] is a mixture of the mentioned two types of network and an instance-aware downsampling strategy was proposed to select representative fixed-number points. Without complex processing such as voxelization of data, point-based encoding methods especially the one-stage ones are straightforward, but their efficiency is somewhat limited.

2) Backbones & Heads: Convolutional backbones and transformer are the two most popular choices for the core structure of detection networks. In point-based networks, convolutional backbones are usually implemented with a series of convolutional layers to extract multi-scale features extracted from the data and a couple of deconvolutional layers serves in model heads to detect objects. In this case, Fully Convolutional Networks (FCN) and Unet [5] are ideal representatives of this encoder-decoder structure. Specifically, feature pyramid network (FPN) [18] was designed to integrate features from all layers of such backbones and non-max suppression (NMS) allows better detection performance by combining multi-scale detection. In contrast, self-attention mechanism in transformer backbones [19] enable the network to learn object-level features, and the following feed-forward network (FFN) are used to produce predictions. To enhance detection performance, voxelization encoding approach is used prior to transformer architecture [19], [20]. Making full use of semantic information is another essential issue in improving learning performance. For example, [21] realized multi-task learning by leveraging the semantic information in their model's convolutional backbone. An inspiring attention-augmented network [22] was proposed for scene parsing via bilinear upsampling in feature map, bridging the semantic and resolution gap between multi-level features.

B. Free Space Detection

Occupancy grid mapping is the most popular approach to free space estimation because it takes advantage of inverse sensor model (ISM) and Bayesian filtering. In detail, this technique uses a grid-based representation of the environment to estimate the occupancy of each cell in the grid [23], [24]. The occupancy status of each cell is usually represented by a probability value ranging from 0 (free space) to 1 (occupied space) [25]. However, this method has limitations, such as the need for accurate sensor measurements with respect to range or angle [25], [26] and the information rectification resulted by grid generation process [27], [28]. Deformable polygon method outputs polygon represented free space by updating polygon vertices with the aid of ISM and it runs significantly faster than traditional methods [29]. NVIDIA proposed a multi-task network [5] with a specific detection head for free space, where occupancy probability map was used.

C. Multi-task Models

Research on multi-task learning mechanisms is another notable topic in autonomous driving. Different from single-task networks, multi-task method is to assist the learning of multiple tasks by sharing parameters or features in an end-to-end manner. NVRadarNet[5] realized object detection and free space estimation using only radar peak detection and its running speed dramatically faster than real-time. DRMNet[36] is a dual-resolution multi-task network that can complete the tasks of vehicle detection, lane detection, and drivable area detection in autonomous driving. In LiDAR based perception, LidarMTL[37] proposed a simple and effective multi-task network based on 3D sparse convolution and deconvolution for joint object detection and path understanding. In this work, we will draw on these designs to unify main radar based perception tasks into a single network.

III. METHOD

As shown in Fig. 2, in our MTRadSSD, we firstly adjusted sampling stratigies proposed by [3] via adding an occupancy evidence based upsampling step. Following the sampling module, Centroid Prediction and Detection modules are used to implement the whole pipeline. As the decisive element of MTRadSSD is our proposed occupancy evidence map tool, which is related to all mentioned modules, in this section, we will firstly illustrate how we designed this tool and then describe how it make difference in the whole network.

A. Occupancy Evidence Map Tool

Occupancy evidence is usually used to indicate the probability of object existence. Intuitively, point density can directly represent object existence and for LiDAR point

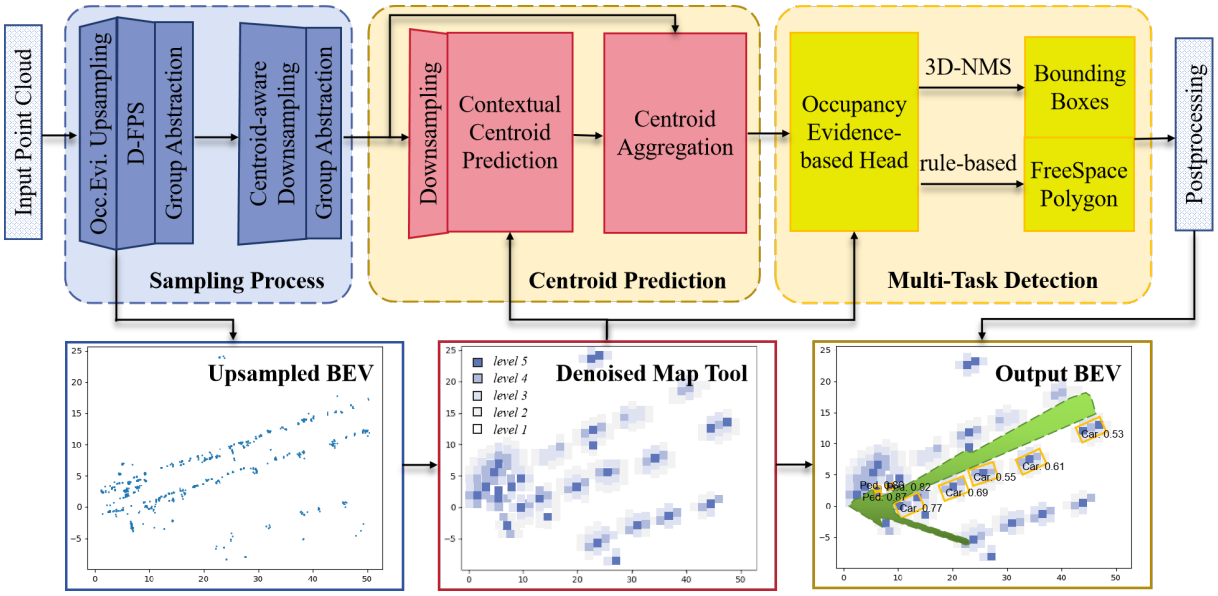


Fig. 2. Overview of the proposed MTRadSSD

cloud, point density has been demonstrated to be beneficial to object detection [19]. However, radar point cloud is much sparser than that of LiDAR and point density non-uniformly distributes in space, whose patterns dramatically varies with distance and object type. Inspiring works such as [5] and [23] offered another referential access to understand environment. [5] used pure radar data and radar cross section (RCS) was selected to determine space occupancy in free space detection. Besides RCS, signal noise ratio (SNR) is another directly or indirectly available measurement for mmw radar, and as illustrated by [30], either RCS or SNR can be transferred to the other according to

$$\text{SNR}=k \cdot \frac{\text{RCS}}{r^4} \tag{1}$$

where r is the distance of point cloud and k is a coefficient to restrict the transformation result into a specific range.

1) Occupancy Evidence: We obtain **density-based occupancy evidence** by defining each point's local density as the number of its neighbouring points within a given radius, and the larger the density is, the more possible the local space is occupied. On the other hand, we get **SNR-based occupancy evidence** according to Swerling-1 model [29], [31], where each point will be given its detection probability calculated from SNR, then we determine spatial occupancy evidence by defining

$$p_d^{\text{Swerling}}(\text{SNR})=p_{fa}^{\frac{1}{1+\text{SNR}}} \tag{2}$$

where p_{fa} is the false alarm rate in point cloud collection.

2) Map Represented Tool: We further normalize all points' occupancy evidence into a specific range such as [0, 1]. Then we adopt Kernel Density Estimation (KDE) to approximate the distribution of occupancy evidence [19], [32] in BEV by supposing all points' occupancy evidence follow Gaussian distribution in their surrounding environment. In specific, for the point set $\{\mathbf{p}_i = \{\mathrm{x}_{\mathbf{p}_i}, \mathrm{f}_{\mathbf{p}_i}\} | i = 1, ..., N\}$, where $\mathrm{x}_{\mathbf{p}_i}$ are spatial measurements of these points, $\mathrm{f}_{\mathbf{p}_i}$ are other measurements such as Doppler and SNR, and N is the number of points, we have

$$\mathcal{P}(x, y) = \frac{1}{Nh^2} \sum_{i=1}^{N} \pi_i \cdot \mathcal{N}\left(\frac{(x - x_i)(y - y_i)}{h^2}\right) \tag{3}$$

where $\mathcal{N}(\cdot)$ is the chosen Gaussian kernel, h is bandwidth and the additive weight is defined in terms of point-wise occupancy evidence p_i as follow.

$$\pi_i = \frac{p_i}{\sum_{j=1}^{N} p_j} \tag{4}$$

We use Eq. (3) to obtain continuous occupancy likelihood function. In this case, we grid the whole scene into cells uniformly along x- and y-axis in BEV and let the cell center represent each cell. As a result, an occupancy evidence map can be calculated by plugging the coordinate matrix of cell centers $\boldsymbol{M_{coor}}$ in function $\mathcal{P}$ as follow

$$\boldsymbol{M_{occ}} = \mathcal{P}(\boldsymbol{M_{coor}}) \tag{5}$$

B. Instance-Aware Sampling Process

Sampling process brings hazard of losing part of foreground points so a variety of sampling strategies have been proposed. It has been demonstrated that under commonly-used encoding architecture PointNet++ with 4 encoding layers, the instance recall rate (i.e., the ratio of instance retained after sampling) of farthest point sampling (FPS)-based on Euclidean distance (D-FPS) [14], or feature distance (Feat-FPS) [11], or both (FS) [11] is higher than that of random sampling [3]. Density-based sampling method such as Poisson Disk [32] is effective in recalling instances but time-consuming with small-size data. To trade off the recall rate against time consumption, we develop sampling strategies based on D-FPS.

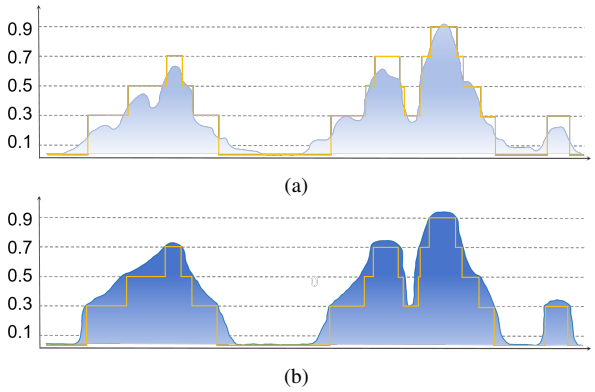


Fig. 3. Illustration of occupancy evidence-based upsampling. The shaded area in (a) indicates KDE distribution of occupancy evidence, and the yellow polygonal line represents the graded occupancy evidence. The shaded area in (b) depicts the distribution of occupancy evidence after upsampling.

1) Occupancy Evidence-based Upsampling: Before D-FPS we carry out an upsampling according to occupancy evidence as local point clouds of instances are much denser than background points. Another purpose of this operation is to avoid small objects (e.g. pedestrian) being filtered in D-FPS as they possess lesser point clouds but not necessarily smaller occupancy evidence. Nevertheless, point clouds' spatial distribution of pedestrian and cyclist are more likely to coincide with their ground truth entities while that of car tends to fall on edges, rather than the center, of the ground truth entity. As illustrated in Fig. 3, within the occupancy evidence map, we first quantify each value S in $\boldsymbol{M_{occ}}$ into 5 levels by resetting it as 0.3 if $S \in (0.15, 0.35]$, as 0.5 if $S \in (0.35, 0.55]$, as 0.7 if $S \in (0.55, 0.75]$, as 0.9 if $S \in (0.75, 1]$ and reset it to be 0 if $S \in [0, 0.15]$ aiming at denoise the map.

We secondly use Monte Carlo sampling method to generate fixed-number new points in cells whose $S \in [0.3, 0.7]$ and only when the cell's neighbors' value are consecutive to its own (e.g, 0.3 and 0.5 is consecutive while 0.3 and 0.7 is not), we retain the new points, otherwise we don't. Eventually, we obtain an upsampled point set $\{\mathbf{p}_i^{new} = \{\mathrm{x}\mathbf{p}_i, \mathrm{f}\mathbf{p}_i\} | i = 1, ..., N^{new}, ..., N^{new} + M\}$, where N^{new} is number of points within denoised point set and M is the number of newly added points. Meanwhile, the occupancy evidence map is updated as $\boldsymbol{M_{occ}^{new}}$. In particular, Monte Carlo sampling only generates coordinates of new points, and we estimate other features $\mathrm{f}\mathbf{p}_i$ of them by calculating average values of their top 5 nearest neighbours' features.

2) Centroid-Aware Sampling: Many object classification models [3] adopted class-aware sampling tools. The vanilla cross-entropy loss is as follow,

$$L_{cls} = -\sum_{c=1}^{C}(s_i log(\hat{s_i}) + (1 - s_i) log(1 - \hat{s_i})) \quad (6)$$

where C represents number of object categories, s_i is the one-hot labels and $\hat{s_i}$ is the predicted logits. This sampling strategy aims to learn point-wise semantics and MLP layers are always adopted following it to further estimate the semantic categories of each point. However, for instance-aware task, determining an object's location is as significant as identifying its category. To this end, soft point mask of instances as follow is usually calculated to assign higher weight to points that are nearer to instance center [3], [11],

$$Mask = \sqrt[3]{\frac{min(f^*, b^*)}{max(f^*, b^*)} \times \frac{min(l^*, r^*)}{max(l^*, r^*)} \times \frac{min(u^*, d^*)}{max(u^*, d^*)}} \quad (7)$$

where $f^*, b^*, l^*, r^*, u^*, d^*$ represent the distance of a point to the 6 surfaces of the 3D bounding box, respectively. It is easy to notice that the value of the mask falls in [0, 1] as once a point locates on those surfaces the numerator of the radicand will be 0. In this way, a weighted centroid-aware sampling strategy can be written as follow.

$$L_{ctr} = -\sum_{c=1}^{C}(Mask_i \cdot s_i log(\hat{s_i}) + (1 - s_i) log(1 - \hat{s_i})) \quad (8)$$

3) Contextual Centroid Prediction: In centroid prediction stage, we attempt to leverage the updated occupancy evidence map to improve and balance the accuracy of instance centroid prediction across object categories. In particular, in the light of the updated occupancy evidence map, we design a score map with continuous values by smoothing the occupancy evidence map with a convolution tool. Thus, the score map $\boldsymbol{M_w}$ is calculated as Eq. (9).

$$\boldsymbol{M_w} = conv(\boldsymbol{M_{occ}^{new}}, \boldsymbol{w}) \quad (9)$$

where $\boldsymbol{w}$ is a 3×3 bilinear convolutional kernel. Finally, inspired by score-based methods such as Hough Voting [33], [3], we introduce the following loss term to optimize centroid prediction by aggregating more contextual information.

$$L_{cent} = \frac{1}{|N_{ins}|}\frac{1}{|N_g|}\sum_i\sum_j(|\Delta\hat{d_{ij}} - \Delta d_{ij}| + |\hat{d_{ij}} - \overline{d_i}|) \cdot (\boldsymbol{M_w})_{ij} \quad (10)$$

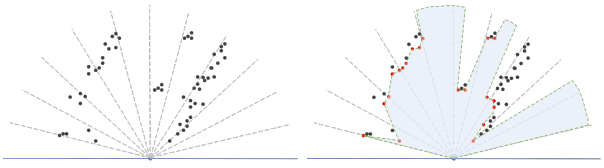
where

$$\overline{d_i} = \frac{1}{|N_g|}\sum_j \hat{d_{ij}} \quad (11)$$

where $|N_{ins}|$ is the number of candidate instances, $|N_g|$ is the number of points used to predict instance center, Δd_{ij} is the ground truth offset of the jth point to the ith instance, $\Delta\hat{d_{ij}}$ is the predicted value of this offset, and $\overline{d_i}$ is the mean destination of ith instance. Subsequent to centroid prediction, we aggregate the selected point features to extract centroids of instance via shared MLPs and symmetric functions.

C. Multi-task Detection Head

Even though the score map $\boldsymbol{M_w}$ allows us to calculate free space conveniently by searching connected space with low occupancy evidence, we set two branches in detection head because we only have ground truth for road users at hand while we only produced a limited amount of ground truth labels of free space manually for evaluation process. In this case, we apply 3D non-maximum-suppression (NMS)



(a) (b)

Fig. 4. Free space detection. (a) shows original point cloud and (b) depicts the free space polygon, where shaded region is the free space and red points are selected vertices of the polygon. Dashed lines are plotted every 15°.

to figure out bounding boxes of road users and propose a rule-based method to implement free space detection.

Similar to the work in [34], in model preparation stage, we compute the distance and orientation relative to the origin of each cell within $\boldsymbol{M_W}$ and stored them as dictionaries to facilitate the construction of free space polygon. Then as shown in Fig. 4, for non-zero elements in $\boldsymbol{M_W}$ we group them with respect to their orientations degree by degree and within each group we find the element with smallest distance to the origin. Lastly, we connect those selected elements to form the free space polygon. The rule-based free space detection also enables us to generate confidence level for each vertex of the polygon according to its score. With those mentioned dictionaries, the free space detection can be accelerated by parallel computation.

D. Training Loss Functions

The loss function of MTRadSSD method contains two parts: sampling loss and prediction loss. Besides centroid-aware sampling loss L_{ctr}, which also implies class prediction loss, and centroid prediction loss L_{cent}, as most 3D detection approaches do, we also count in losses from predictions of box center, size and orientation. In summary, the total training loss for our proposed method is

$$L_{total} = L_{ctr} + L_{cent} + L_{loc} + L_{size} + L_{ori} \qquad (12)$$

IV. EXPERIMENTS

A. Implementation Details

1) Dataset: We test our MTRadSSD on the public View-of-Delft (VOD) radar dataset in this section as the amounts of labeled objects (cars, pedestrians and cyclists) in it are more balanced than other datasets such as NuScenes. Point clouds in VOD are featured with x, y, z measurements, RCS reflectivity, v_{rel} relative radial velocity, v_r compensated radial velocity. In this dataset, objects car, pedestrian and cyclist are classified into three types, that is, "Easy", "Moderate" and "Hard" according to the levels of difficulty. The 5-frame accumulated data in VOD are used in our experiments as it is of higher resolution and we split the dataset into a training, validation, and testing set in a ratio of 59%/15%/26%.

2) Baselines: Since we intend to develop a real-time multi-task detector, we take relevant state-of-the-art approaches as baselines. PointPillars [35], as a representative of voxel-based model, was selected to test the VOD dataset by [6] and it showed excellent speed. PDV [19] adopted an architecture including transformer and FFN modules and it outperformed PointPillars by considering point density. IA-SSD is chosen as a baseline because it is famous for its rapidity and it also proposed highly effective point sampling strategies. In multi-task detection aspect, we will compare our work with that of NVIDIA. All our experiments are executed on a single RTX 3080 Ti GPU.

TABLE I

DETAILED NETWORK SETTINGS OF MTRADSSD

Layer	Sampling Method	Grouping Radius	Points	Features
1	Up-D-FPS	[0.2, 0.8]	1024	64
2	Ctr	[0.8, 3.2]	512	128
3	Ctr	–	384	256
4	Vote	–	384	–
5	–	[3.2, 6.4]	384	512

3) Network Settings: Table I illustrates settings of group abstraction layers with details. In the first encoding layer we adopt D-FPS combining with the proposed density-based usampling ('Up-D-FPS'), where KDE bandwidth h=0.25. The next two layers employ the mentioned centroid-aware sampling ('Ctr') to select 256 point features, followed by voting-based centroid prediction layer ('Vote'). Instance centroids are predicted by three MLP layers and the terminal outputs for object detection with instance classification and regression are produced by another 3 MLP layers.

B. Experimental Evaluation

1) Benchmarks: We evaluate the two aspects performance, object detection and free space detection, of the MTRadSSD in this part. For object detection, we adopt benchmarks used in most SOTA models. That is, mean Average Precision (mAP) and mean Angle of Similarity (mAOS) for object detection. For free space, we employ IoU-gt, IoU-smooth and MSE to evaluate the smoothness of predicted free space along time and precision of the predictions. Specifically, multi-task model proposed by NVIDIA used F-score, so that we also adopt this metric. The mentioned IoU-gt, IoU-smooth are mathematically defined as the intersection over union (IoU) between ground truth and predicted free space polygon (IoU-gt) and that between every two successive polygon predictions, respectively [29].

2) Road User Detection: As shown in Table II, all results are evaluated by mAP with 40 recall positions via VOD evaluation server and all class-specific columns are calculated with a 3D IoU (0.5 for car, 0.25 for pedestrian and cyclist). The best result of each column are shown in bold, and the suboptimal results are underlined. As we can see, 2-stage approach PDV is severely inferior to 1-stage ones in speed and unable to satisfy real-time requirement in engineering. However, it performs superbly in cyclist detection as it benefits from considering local point density. PP-radar was proposed by [6] and this voxel-based approach is the fastest among all selected baseline models. Approaches featured single stage detection, IA-SSD and our MTRadSSD,

TABLE II

QUANTITATIVE OBJECT DETECTION PERFORMANCE OF DIFFERENT METHODS ON THE VOD 5-FRAME DATASET

Type	Method	Car			Pedestrian			Cyclist			Speed
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
2-stage	PDV	42.9	38.3	33.3	44.9	36.7	33.9	80.2	73.0	65.9	76.9
1-stage	PP-radar	39.5	44.3	37.4	42.1	35.5	31.8	60.9	52.4	45.7	**11.2**
1-stage	IA-SSD	42.5	35.2	30.1	57.2	49.2	44.1	83.1	75.8	67.6	18.9
1-stage	**MTRadSSD (Ours)**	**52.2**	**48.6**	**44.3**	**61.1**	**54.0**	**49.7**	**86.3**	**80.9**	**72.9**	16.7

TABLE III

PERFORMANCE COMPARISON AMONG BASELINES

Method	mAP	AP Std.	mAOS	F-score
PDV	49.8	16.4	46.8	0.54
PP-radar	43.3	**7.1**	38.7	0.44
IA-SSD	53.9	16.3	49.9	0.52
NVRadarNet	18.4	19.2	–	0.35
Ours	**61.1**	13.6	**64.0**	**0.57**

TABLE IV

COMPARISON OF FREE SPACE DETECTION ALGORITHMS

Method	IoU-gt	IoU-smooth	MSE	Speed
Defor. Poly.	0.73	0.90	0.13	19.6
NVRadarNet	0.59	–	–	**1.50**
Occ.-Net	0.44	–	–	–
Ours	**0.88**	**0.91**	**0.10**	16.7

generally can realize real-time detection. Main difference of the two approaches is sampling strategies and MTRadSSD outperforms in all columns as its sampling strategy is specifically designed considering distribution heterogeneity of radar point cloud across object categories. In Table III, we calculated the overall mAP and mAOS of the three types of objects of each method and it is easy to make intuitive comparison. We compute standard deviation of mAP (AP Std.) among all three object types of each method to indicate the difference in detection performance across object categories. Consequently, MTRadSSD obtained the highest mAP 61.1%, mAOS 64.0% and F-score 0.57, and relatively low AP Std. 13.6%, implying that the proposed method not only improved detection precision but also effectively balanced the detection ability among different object types. Particularly, detailed data and code of NVRadarNet method from NVIDIA are not completely available at the moment, so we cite their averaged evaluation results directly for comparison. More specifically, considering that the detection range of VOD dataset is about 50 meters, we only take NVRadarNet's experiment results within this range into comparison.

3) Free Space Detection: Rule-based free space detection is facilitated with distance dictionary and orientation dictionary affiliating to the occupancy evidence map. When compared with other models, as shown in Table IV, our approach outperforms Deformable Polygon method proposed by [29] in terms of IoU-gt, IoU-smooth and MSE. Although the speed (in milli-second per frame) of our model is not as fast as that of NVRadarNet, it is still competitive when compared with Deformable Polygon. In particular, IoU-smooth, MSE and speed of OccupancyNet [23] are not available, so we only cite its IoU-gt in comparison.

TABLE V

RESULTS OF ABLATION EXPERIMENTS ON MTRADSSD

Condition	Car	Ped.	Cyc.	AP Std.	mAOS
no upsam.	40.2	51.1	75.0	14.5	53.5
no OEM	39.7	49.2	73.5	14.2	49.6
original	48.4	54.9	80.0	13.6	64.0

C. Ablation Studies

1) Ablation on Upsampling Strategy: We remove upsampling process in this experiment and thus we use downsampled data to generate occupancy evidence map. Then as shown in Table V, the prediction accuracy of Car declines dramatically and the mAOS also decreases ˜10%. These consequences indicate that the proposed upsampling method is effective and indeed improved the imbalance in prediction precision over object categories of the method.

2) Ablation on Occupancy Evidence Map: The occupancy evidence map contributes in contextual centroid prediction and also makes difference in detection head. As we are mainly interested in its impact on obstacle detection, we remove this tool and replace the detection head with that used in IA-SSD [3]. Consequently, the prediction precision of all objects drop down and that of Car and mAOS again falls severely by 8.7% and 14.4%, respectively. In contrast to the proposed upsampling strategy, the occupancy evidence map plays greater impact on the model. Specifically, this again proves that the map tool improved object orientation prediction.

V. CONCLUSION

We propose MTRadSSD aiming at multi-task detection with radar point cloud. The carefully designed upsampling strategy effectively improved the unbalance of detection ability among car, pedestrian and cyclist. Meanwhile, the occupancy evidence map-based centroid prediction module works superbly in promoting the detection accuracy of Car and object orientation. In terms of free space detection, the rule-based approach is proved efficient with the aid of occupancy evidence map. In our experiments on the VOD dataset, our method outperformed other SOTA approaches especially with respect to mAOS. The performance of MTRadSSD indicates that point cloud of millimeter-wave radar tends to produce reliable detection for vulnerable road users, i.e. pedestrians and cyclists. A possible future improvement of MTRadSSD is that, when free space labels are available, we can make better use of the occupancy evidence map tool in training stage by regarding free space as a special type of road user.

REFERENCES

[1] T. Zhou, M. Yang, K. Jiang, H. Wong and D. Yang, MMW radar-based technologies in autonomous driving: A review, Sensors, vol. 20, no. 24, pp.7283, 2020.

[2] H. D. Mafukidze, A. K. Mishra, J. Pidanic and S. W. Francois, Scattering centers to point clouds: a review of mmWave radars for non-radar-engineers, IEEE Access. Oct. 3, 2022.

[3] Y. Zhang, Q. Hu, G. Xu, Y. Ma, J. Wan, and Y. Guo, Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 18953-18962.

[4] Z. Liu, H. Tang, Y. Lin and S. Han, Point-voxel cnn for efficient 3d deep learning, Advances in Neural Information Processing Systems, 2019;32.

[5] A. Popov, P. Gebhardt, K. Chen and R. Oldja, Nvradarnet: Real-time radar obstacle and free space detection for autonomous driving, In 2023 IEEE International Conference on Robotics and Automation (ICRA), 2023, May 29, pp. 6958-6964.

[6] A. Palffy, E. Pool, S. Baratam, J. F. Kooij and D. M. Gavrila, Multi-class road user detection with 3+1D radar in the View-of-Delft dataset, IEEE Robotics and Automation Letters, 2022, Feb. 1, vol. 7, no. 2, pp:4961-4968.

[7] O. Schumann, M. Hahn, J. Dickmann and C. Wöhler, Semantic segmentation on radar point clouds, In 2018 21st International Conference on Information Fusion (FUSION), 2018, Jul. 10, pp. 2179-2186.

[8] Y. Sun, H. Zhang, Z. Huang and B. Liu, R2p: A deep learning model from mmwave radar to point cloud, In International Conference on Artificial Neural Networks, 2022, Sep. 6, pp. 329-341.

[9] C. R. Qi, O. Litany, K. He and L. J. Guibas, Deep hough voting for 3d object detection in point clouds, In proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9277-9286.

[10] S. Shi, X. Wang and H. Li, Pointrcnn: 3d object proposal generation and detection from point cloud, In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 770-779.

[11] Z. Yang, Y. Sun, S. Liu and J. Jia, 3dssd: Point-based 3d single stage object detector, In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11040-11048.

[12] C. R. Qi, H. Su, K. Mo and L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 652-660.

[13] Y. Liu, B. Fan, S. Xiang and C. Pan, Relation-shape convolutional neural network for point cloud analysis, In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 8895-8904.

[14] C. R. Qi, L. Yi, H. Su and L. J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems, 2017;30.

[15] R. Qian, D. Garg, Y. Wang, Y. You, S. Belongie, B. Hariharan, M. Campbell, K. Q. Weinberger and W. L. Chao, End-to-end pseudo-lidar for image-based 3d object detection, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5881-5890.

[16] S. Thakur and J. Peethambaran, Dynamic edge weights in graph neural networks for 3D object detection, arXiv preprint arXiv:2009.08253, 2020, Sep. 17.

[17] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein and J. M. Solomon, Dynamic graph cnn for learning on point clouds, ACM Transactions on Graphics (ToG), 2019, Oct. 10, vol. 38, no. 5, pp:1-2.

[18] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, Feature pyramid networks for object detection, In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117-2125.

[19] J. S. Hu, T. Kuai and S. L. Waslander, Point density-aware voxels for lidar 3d object detection, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8469-8478.

[20] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, In Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10012-10022.

[21] F. Zhou, B. Chaib-draa and B. Wang, Multi-task learning by leveraging the semantic information, In Proceedings of the AAAI Conference on Artificial Intelligence, 2021, May 18, Vol. 35, No. 12, pp. 11088-11096.

[22] Q. Song, K. Mei and R. Huang, AttaNet: Attention-augmented network for fast and accurate scene parsing, In Proceedings of the AAAI conference on artificial intelligence, 2021, May 18, Vol. 35, No. 3, pp. 2567-2575.

[23] L. Sless, B. S. El, G. Cohen and S. Oron, Road scene understanding by occupancy grid learning from sparse radar clusters using semantic segmentation, In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0-0.

[24] C. Lundquist, T. B. Schön and U. Orguner, Estimation of the free space in front of a moving vehicle, 2009.

[25] M. Li, Z. Feng, M. Stolz, M. Kunert, R. Henze and F. Küçükay, High Resolution Radar-based Occupancy Grid Mapping and Free Space Detection, In VEHITS, 2018, Mar., pp. 70-81.

[26] K. Werber, M. Rapp, J. Klappstein, M. Hahn, J. Dickmann, K. Dietmayer and C. Waldschmidt, Automotive radar gridmap representations, In 2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), 2015, Apr. 27, pp. 1-4.

[27] R. Prophet, H. Stark, M. Hoffmann, C. Sturm and M. Vossiek, Adaptions for automotive radar based occupancy gridmaps, In 2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), 2018, Apr. 15, pp. 1-4.

[28] M. Slutsky and D. Dobkin, Fast implementation of volumetric occupancy grids, In 2019 IEEE Intelligent Vehicles Symposium (IV), 2019, Jun. 9, pp. 750-755.

[29] X. Gao, S. Ding, K. Vanas, R. H. Dasari, and H. Soderlund, Deformable Radar Polygon: A Lightweight and Predictable Occupancy Representation for Short-range Collision Avoidance, arXiv preprint arXiv:2203.01442, 2022, Mar. 2.

[30] S. M. Ivan, Introduction to radar systems. New York: Mcgraw Hill, 1980.

[31] J. Degerman, T. Pernstål and K. Alenljung, 3D occupancy grid mapping using statistical radar models, In 2016 IEEE Intelligent Vehicles Symposium (IV), 2016, Jun. 19, pp. 902-908.

[32] P. Hermosilla, T. Ritschel, P. P. Vázquez, À. Vinacua and T. Ropinski, Monte carlo convolution for learning on non-uniformly sampled point clouds, ACM Transactions on Graphics (ToG), 2018, Dec. 4, vol. 37, no. 6, pp:1-2.

[33] S. Luo and W. Hu, Score-based point cloud denoising, In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4583-4592.

[34] Y. Li, S. Yu, M. Li, Z. Jia and Y. Song, QDTree: Quasi-density-tree accelerates free space detection with mmw radar point cloud, In Proceedings of the IEEE International Conference on Intelligent Traffic and Transportation (ICITT), 2023, unpublished.

[35] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang and O. Beijbom, Pointpillars: Fast encoders for object detection from point clouds, In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 12697-12705.

[36] J. Zhao, D. Wu, Z. Yu and Z. Gao, DRMNet: A Multi-Task Detection Model Based on Image Processing for Autonomous Driving Scenarios, In IEEE Transactions on Vehicular Technology, vol. 72, no. 12, 2023, Dec., pp. 15341-15355.

[37] D. Feng, Y. Zhou, C. Xu, M. Tomizuka and W. Zhan, A Simple and Efficient Multi-task Network for 3D Object Detection and Road Understanding, 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 2021, pp. 7067-7074.