Abstract—Millimeter-wave (mmWave) radar has been widely used in autonomous driving due to its good performance under harsh weather conditions. In recent years, with the development of mmWave radar hardware performance, radar point clouds, as an important data format of mmWave radar, have been widely used in high-level perception tasks of mobile robots and autonomous driving. However, at present, compared to LiDAR point clouds, in common application scenes of mobile robots, mmWave radar point clouds have shortcomings such as sparsity and containing many "ghost" targets. Therefore, in this article, we analyze the reasons that cause these problems and propose a new method for point cloud generation as well as a new evaluation metric. After building a new dataset and carrying out experiments in real-world scenes, our method shows better performance on the quality of radar point clouds compared to other methods. In addition, by evaluating the performance of applying the high-quality radar point clouds to object detection tasks as well as localization and mapping tasks, the result shows that radar point clouds generated using our method can significantly improve the environment perception ability of mobile robots.

Index Terms—Millimeter-wave (mmWave) radar, point cloud, radar detector, robot environment perception.

I. INTRODUCTION

In the past few years, 77-GHz millimeter-wave (mmWave) radar technology, as a low-cost alternative to LiDAR in advanced driver assistance systems [1], has been rapidly developed and has been applied to autonomous emergency braking, adaptive cruise control, and other functions. In addition to its reasonable costs, mmWave radar is robust to different lighting and weather conditions and can provide measurements of range, azimuth angles, and instantaneous velocity [2].

Recently, with the development of integrated circuits, mmWave radar has grown to have higher integration, lower price, and better radio frequency (RF) performance [3]. Therefore, in addition to autonomous driving, single-chip mmWave radar is also widely applied to mobile robots [4], such as the wheeled robot [3], [5], [6], unmanned aerial vehicle [7], and unmanned surface vehicle [8] (as shown in Fig. 1). The single-chip mmWave radar plays an important role in the perception of mobile robots under harsh weather conditions like rain and fog [9].

As an important data format of mmWave radar, similar to LiDAR point clouds, mmWave radar point clouds [10] have gradually been applied to robot environment mapping and localization tasks and object detection tasks. However, unlike LiDAR point clouds, for applications in mobile robots, typical mmWave radar point clouds have the following three limitations (as illustrated in Fig. 2).

1) Sparsity. Compared with LiDAR point clouds, mmWave radar point clouds are relatively sparse [11]. Radar point clouds usually cannot intuitively reflect the shape of an object in detail, which results in a lack of semantic information.

2) Clutter points. Due to ground reflection and the multipath effect (as illustrated in Fig. 3), mmWave radar point clouds usually contain clutter points, leading to falsely detected ghost targets [12].

3) Viewpoint variation and temporal variation [13]. As shown in Fig. 4, on the one hand, for close viewpoint angles, the point clouds of an object may be different. On the other hand, although the viewpoint angle remains the same, the point clouds of an object may change.

mmWave signals remain adaptable under different environmental conditions because of their wavelength. However, the above disadvantages limit further applications of mmWave radar point clouds.
clouds for robot environment perception. For example, the lack of semantic information and ghost targets increases the difficulties in tasks such as object detection and semantic segmentation. The viewpoint variation and temporal variation increase the difficulties in point set registration [14], which makes simultaneous localization and mapping (SLAM) based on mmWave radar point clouds more challenging. Therefore, it is essential to overcome these limitations and generate high-quality mmWave radar point clouds for robust mobile robot environment perception.

The above limitations of mmWave radar point clouds are mainly due to two reasons [2]: the inherent physical characteristics of the hardware and the defects of the classical radar signal processing algorithms. The mmWave wavelength leads to high specularity on object surfaces. Unlike light signals, which scatter in every direction, not all reflections from the object propagate back to the mmWave receiver [5], [11]. In addition, because of the low-cost single-chip design, mmWave radar has relatively few antennas, which makes the angular resolution relatively low. In addition to the mmWave radar hardware, radar signal processing algorithms play an important role in generating radar point clouds and greatly influence the point cloud quality [13]. However, it is still a relatively unexplored field to improve the performance of radar signal processing algorithms to generate mmWave radar point clouds of higher quality, which are more suitable for applications on mobile robots.

In this case, our work mainly aimed at the generation of high-quality mmWave radar point clouds for the applications of mmWave radars in mobile robots. We first analyze the shortcomings of the classic mmWave radar signal processing chain under common application scenes of mobile robots. Then, we propose a new method to generate mmWave radar point clouds of higher quality. Our method contains two modules, the radar points detector network (RPDNet) and the radar point cloud spatiotemporal filter (RSTF). In addition, we define a new evaluation metric for the quality of mmWave radar point clouds that can be adapted to mobile robot environment perception tasks. Testing on real-world data collected in streets, parking lots, indoor environments, and water surfaces, our method shows better performance with fewer clutter points and denser real point clouds. In addition, we apply the radar point clouds generated using the proposed method to the object detection task and localization and mapping tasks. The results show that mmWave radar point clouds of higher quality generated using our method contribute to the applications of mobile robot environment perception.

In our previous work [15], we focused on the specific applications of mmWave radar in autonomous driving and proposed a primary data-driven mmWave radar detector that can improve the mmWave radar point cloud quality in this application field. In this work, we comprehensively extend the previous work and turn to meet the requirements of mobile robot environment perception applications. The application scenes include streets, indoor environments, and water surfaces. We add theoretical analysis for the challenges of radar detectors under mobile robot application scenes and propose a new method that achieves significant improvement in performance compared to existing methods. In addition, we build a prototype and carry out experiments for mobile robot environment perception tasks.

To summarize, this article mainly contributes to the following aspects.
1) To the best of our knowledge, we are the first one to focus on mmWave radar point cloud quality improvement in mobile robot application scenes. This is a basic problem for mmWave radar used in mobile robots which has not been explored before. We conduct a thorough analysis of this problem through theoretical proofs and experiments.

2) We propose a novel method that is able to generate high-quality mmWave radar point clouds for mobile robot applications to improve the performance of environment perception tasks. In addition, we propose a new evaluation metric for the quality of mmWave radar point clouds used in mobile robot applications.

3) We build a prototype platform and carry out real-world scene application experiments in which our method shows good performance. In addition, we build and release a new dataset that contains raw data collected in three types of scenes, including indoor scenes, streets, and inland waters, which can benefit researchers in the mobile robot environment perception community.

The detailed composition of the article is as follows. In Section II, we discuss the related work including radar point cloud generation and applications of mmWave radar for mobile robots. Section III introduces the generation of mmWave radar point clouds and explains the reasons for the low quality of mmWave radar point clouds in mobile robot applications through theoretical analysis and experiments. Section IV details our method, including the radar point detector network model and RSTF. In Section V, we introduce the collection and generation of our real-world dataset. The experiments and the quantitative evaluation results based on our new definition of the evaluation method as well as a comparison with other methods are also presented. In Section VI, we evaluate the impact using radar point clouds generated using our method for two mobile robot perception tasks: object detection and localization and mapping. Finally, Section VII concludes this article.

II. RELATED WORK

A. Radar Point Cloud Generation

A typical mmWave radar signal processing chain that contains two steps, including raw radar point cloud generation and point cloud postprocessing, is shown in Fig. 5. To generate effective and accurate mmWave radar point clouds, some researchers [16], [17] have tried to improve the signal processing algorithm to increase the angular resolution. The work [16] used a neural network for the direction of arrival (DOA) estimation. Qian et al. [17] applied the idea of synthetic aperture radar imaging to a two-chip cascade automotive radar to improve the side-view point cloud quality by increasing the radar point cloud angular resolution. In addition, some researchers focus on postprocessing, which means further mmWave radar point cloud processing [12], [18], [19]. The work [12] used a modified PointNet network architecture to classify real targets and ghost targets in mmWave radar point clouds. However, because much information has been lost during raw radar point cloud generation, the improvement through postprocessing is limited.

For raw radar point cloud generation, the radar detector is essential. Some researchers [20]–[23] focus on improving the performance of radar detectors. Currently, the constant false alarm rate (CFAR) detector [24], [25] is the most widely used detector in mmWave radar signal processing. Aiming at the limitations of the CFAR detector such as masking effects and computation complexity, some modified methods [20]–[22], such as cell averaging (CA)-CFAR and ordered statistic (OS)-CFAR, have been proposed. However, when applied to complex robot and autonomous driving scenes, for example, in a high-density clutter environment such as guardrails, tunnels, and soundproof walls, the performance of the CFAR detector decreases dramatically as the number of clutter points increases considerably [23].

Recently, Brodeski et al. [13] proposed a data-driven method that outperformed classical CFAR methods in terms of detection accuracy in an anechoic chamber with a known point target (corner reflector). However, data collected in an anechoic chamber differ from data collected in a real-world driving scene, which are more challenging with multipath reflections, interference, and attenuation. To the best of our knowledge, in mobile robot applications, generating high-quality mmWave radar point clouds is still a relatively unexplored field.

B. Applications in Robots

In recent years, single-chip and low-cost mmWave radar point clouds have been gradually used in tasks such as environment mapping and localization and object detection for mobile robots.

1) Mapping and Localization: In autonomous driving, [27] proposed a new radar SLAM method that achieves robust localization in challenging dynamic environments containing pedestrians and moving cars. The work [28] presented a real-time pose graph-based SLAM and demonstrates robustness on long tracks. In mobile robots, some studies are based on the fusion of mmWave radar and an inertial measurement unit (IMU) for ego-motion estimation [3], [5]–[7]. In some visually degraded environments, the radar-inertial method proposed in [6] achieves better motion estimation accuracy than the visual-inertial approach. In an indoor environment, Milli-RIO [3] achieves better performance than wheel-IMU odometry. MilliEgo [5] uses the
deep fusion of radar and an IMU and achieves better performance on motion estimation than vision-based methods. The work [7] proposed an extended Kalman filter based radar inertial odometry and tested the odometry performance in 3-D space using the quadcopter mounted with mmWave radar. In addition to the odometry and localization tasks, some works have also explored constructing a gridmap [29], [30].

2) Object Detection: In recent years, through the use of deep neural networks, such as PointNet [31] and VoteNet [32], some researchers have applied sparse mmWave radar point clouds to tasks such as object detection and semantic segmentation. Danzer et al. [33] presented an approach for 2-D car detection using only sparse radar point clouds based on a PointNet architecture. Nabati and Qi [34] presented a region proposal network for object detection in autonomous vehicles using mmWave radar point clouds. Some works apply mmWave radar point clouds to semantic segmentation to identify targets such as cars, pedestrians, and bike and achieve road scene understanding [35]–[37]. In addition, there are also some works that increase the accuracy and robustness of object detection by using the fusion of mmWave radar point clouds and vision [38]–[41].

Whether for localization and mapping or object detection, single-chip mmWave radar point clouds have been increasingly applied to autonomous driving and mobile robots in the last two years. However, researchers in the robotics community mainly focus on using the generated mmWave radar point clouds to solve problems in robot environment perception. For researchers in the radar signal processing community, attention is mainly paid to challenges in classic radar applications. Therefore, it is still a relatively unexplored field to improve the performance of radar signal processing and better use generated radar point clouds for mobile robot environment perception.

III. MMWAVE RADAR POINT CLOUD GENERATION

A. mmWave Radar Signal Model

In common commercial mmWave radars, a frequency-modulated continuous-wave (FMCW) is a widely used transmitting signal. For mmWave radars, a frame is a complete cycle containing transmitting signals, receiving echoes, and signal processing. A frame is composed of \( N_{\text{chirp}} \) linear frequency-modulated chirps, whose frequency varies linearly with time between the minimum frequency \( f_{\text{min}} \) and the maximum frequency \( f_{\text{max}} \). Thus, each chirp can be characterized by the duration \( T_{\text{chirp}} \) and the bandwidth \( B = f_{\text{max}} - f_{\text{min}} \). For radar signal processing, the sampled echoes are usually arranged in three dimensions and form a radar data cube [26]. For an echo signal, as shown in Fig. 6, the sampling of the same chirp at different times is the first dimension, the sampling between different chirps is the second dimension, and the sampling between different receiving antennas is the third dimension.

For data of a single receiving antenna in a radar data cube, by using the basic signal processing method [a fast Fourier transform (FFT) in two dimensions] shown in Fig. 5, a range–Doppler matrix (RDM) can be generated. As shown in Fig. 7, the range and Doppler index of the cells in the RDM correspond to the range and Doppler velocity of the target. According to the theory of digital signal processing, the range resolution can be computed by

\[
R_{\text{res}} = \frac{c}{2B}
\]

where \( c \) indicates speed of light, and the velocity resolution [42] is

\[
v_{\text{res}} = \frac{\lambda}{2T_{\text{chirp}}N_{\text{chirp}}}
\]

where \( \lambda \) represents the wavelength of the carrier. However, for mmWave radar, due to the limitations of hardware conditions such as the range of the bandwidth \( B \) and the duration \( T_{\text{chirp}} \), the detection range and the resolution of Doppler velocity are usually limited.

In RDM, the cells that contain real targets are detected using the radar detector. Therefore, the number of real targets and “ghost” targets detected by the radar is greatly affected by the performance of the radar detector. Each receiving antenna will generate an RDM. As shown in Fig. 8, the mean values of RDMs from different antennas are the input of the radar detector. For each detected cell, by using the data of the cells in the same position from different antennas, the DOA of the cell is estimated and a 4-D radar point \((x, y, z, v)\) can be generated.
Doppler velocity \( v_{q_2} \) is

\[
v_{q_2} = v_{q_1} + \Delta v = v_b \cdot \cos(\phi + \Delta \phi) \cdot \cos \theta \tag{5}
\]

where \( \Delta \phi \) denotes the elevation difference between the two points. The Doppler velocity difference is

\[
\Delta v = v_b \cdot \cos(\phi + \Delta \phi) \cdot \cos \theta - v_b \cdot \cos \phi \cdot \cos \theta. \tag{6}
\]

For mmWave radar, the maximum unambiguous Doppler is velocity

\[
v_{\text{max}} = \frac{\lambda}{4 \cdot T_{\text{chirp}}}. \tag{7}
\]

When the Doppler velocity difference reaches the Doppler resolution \( v_{\text{res}} \), suppose that \( v_b \) is equal to \( v_{\text{max}} \), then

\[
\left| \frac{\Delta v}{v_b} \right| = \frac{v_{\text{res}}}{v_{\text{max}}} = \frac{2}{N_{\text{chirp}}}. \tag{8}
\]

Using Formulas (6) and (8), it can be determined that

\[
\Delta \phi = \arccos \left( -\frac{2}{N_{\text{chirp}} \cdot \cos \theta + \cos \phi} \right) - \phi. \tag{9}
\]

Formula (9) shows that the Doppler velocity resolution can be transformed to the angular resolution to some degree, and the angular resolution is decided by \( N_{\text{chirp}} \) as well as the azimuth and elevation of the target. In Appendix A, we give a more complete theoretical proof and experimental verification. It can be proven that in common mobile robot application scenes, the angular resolution determined using the Doppler velocity resolution can be higher than the angular resolution determined using the number of virtual receiving antennas.

Therefore, according to Formula (9), by increasing the number of chirps \( N_{\text{chirp}} \), the spatial resolution of a low-cost single-chip mmWave radar can be increased. However, although denser radar point clouds can theoretically be generated by increasing the spatial resolution, in real-world mobile robots application scenes, limited by current radar detectors, the generated radar point clouds are still relatively sparse. In Section III-C, we will provide further analysis.

C. Radar Detector

A radar detector is usually based on the Neyman–Pearson theorem [43] of hypothesis testing in statistics. Therefore, under this theorem, current CFAR detectors of different types are the most widely used detectors in the field of mmWave radar signal processing [24]. CFAR is an adaptive detection algorithm that adjusts the detection threshold in line with the measured background. Specifically, a CFAR detector estimates the noise floor for the cell under test (CUT), by analyzing data from neighboring cells designated as training cells. To avoid the signal components of the CUT from leaking to the training cells, cells immediately adjacent to the CUT are normally ignored and referred to as guard cells, as shown in Fig. 9. For example, for a widely used CA-CFAR detector, the noise level \( y_{\text{nl}} \) can be computed as

\[
y_{\text{nl}} = \frac{1}{2L} \sum_{i=1}^{2L} x_i. \tag{10}
\]
where $L$ is the number of training cells on one side of the CUT and $x_j$ is the sample in each training cell. Then, the detection threshold $T$ is given by

$$T = \alpha \cdot y_{nl}$$

(11)

where $\alpha$ is a scaling factor called the threshold factor. A target is declared present in the CUT if it is greater than the detection threshold.

It has been proven that [24] CFAR detection is a generalized likelihood ratio test (GLRT)

$$H_1 : L_G(x) = \frac{p(x; \hat{\theta}_1, H_1)}{p(x; \theta_0, H_0)} > \gamma$$

(12)

where $L_G(x)$ denotes the likelihood ratio function, $\gamma$ denotes the detection threshold and determined by the false alarm rate, $x$ is the sampled data, and $\theta_0$ and $\theta_1$ are unknown parameters under the $H_0$ and $H_1$ hypotheses. The CFAR detector estimates the statistical properties of the noise from the training cells. When the training cells are independent and identically distributed (i.i.d.), the GLRT has been proven to be a kind of optimal detection [24]. Therefore, a CFAR detector is an optimal detector in ideal environments.

However, in mobile robot application scenes, for a CFAR detector, there are two main problems that make it usually unsuitable.

First, on the one hand, the mmWave radars used in some robot application tasks are usually high-resolution, which leads to one target occupying multiple cells in the RDM. On the other hand, in some mobile robot application scenes (such as indoor scenes and streets), there are usually multiple objects with different sizes, and the sizes of real objects are different, as shown in Fig. 10. These will lead to clutter points being detected. Therefore, classic detectors cannot effectively remove a falsely detected “ghost target.”

To illustrate intuitively, we use real-world data for experiments, as shown in Fig. 11. In Fig. 11(a), we use a CA-CFAR detector. In the cyan boxes, there are objects that occupy multiple cells, which leads to the noise level estimated from the training cells being higher. Therefore, the real targets are shaded, which leads to missed detections. In the magenta box, cells that contain the multipath reflection from the ground are of higher intensities and are falsely detected, which leads to a “ghost target.” Therefore, a classic CFAR detector is not suitable for mobile robot applications.

IV. OUR METHOD

To generate radar point clouds of higher quality, we propose a new method for mmWave radar point cloud generation. In our method, we propose a novel data-driven radar point cloud detector called an RPDNet and an RSTF. An overview of the radar signal processing chain using the proposed method is shown in Fig. 12.

A. Radar Point Detector Network

In our model, the input is an RDM, and the output is a predicted label matrix in which the value of each cell denotes the possibility of containing a real target. In other words, each cell in the RDM is classified to determine whether it contains a real target. Unlike a typical RGB image semantic segmentation task in the computer vision community, there are mainly three new characteristics in our tasks.

1) In an RDM, the distribution of the cells that contain clutter has a particular preference. For example, the clutter caused by a ground multipath reflection is usually distributed in a close range and follows a certain distribution characteristics.

2) The cells that contain real targets do not form a sizeable closed shape: some scattered real target cells appear alone. Therefore, the semantic segmentation needs to be fine-grained, which means that each cell needs to be precise.

3) It is difficult to generate ground truth labels. Unlike RGB images, the two dimensions of an RDM are the range and Doppler velocity. Therefore, it is difficult to annotate an RDM manually.

According to the above characteristics, we propose a new model, RPDNet. As shown in Fig. 13, our proposed RPDNet consists of three parts: position-coding, the segmentor network $S$, and the critic network $C$. In the network input process, to better extract the spatial information in the RDM, we use position-coding. The segmentor network is a fully convolutional encoder–decoder network that generates the predicted label matrix we need from the RDM. Inspired by Luc et al. [44] using generative adversarial networks (GANs) to improve the segmentation results, we use the critic network $C$ to constrain the predicted label matrix at a higher semantic level. The input to the critic network is the RDM with the predicted label matrix and the RDM with the ground truth label matrix. The segmentor
network \( S \) and the critic network \( C \) can be regarded as the generator and a discriminator of a GAN, respectively. We constrain the predicted label matrix through focal loss to approach the ground truth label matrix. Through adversarial training, the critic network \( C \) provides additional supervision for the segmentor network \( S \).

1) Position-Coding: To provide spatial context for the model, the position-coding is processed as follows: the RDM is an \( H \times W \) matrix, where \( H \) denotes the range dimension (height) and \( W \) denotes the Doppler dimension (width). As shown in Fig. 13, the one-channel RDM is expanded to three-channel data \( 3 \times H \times W \), and the values in the two newly added dimensions are filled with horizontal and vertical coordinate information ranging from \(-1\) to \(1\) at regular intervals.

2) Segmentor network: As shown in Fig. 13, the segmentor network \( S \) has an encoder–decoder structure. It is mainly composed of convolution layers with a kernel size of \( 3 \times 3 \), stride of 1, and padding of 1. Maxpooling with a kernel size of \( 2 \times 2 \) and stride of 2 is used for downsampling in the coding phase. In the decoding phase, bilinear interpolation with scaling factor 2 is used for upsampling. Similar to the U-Net classic image segmentation network [45], we add skip-connections between the corresponding layers of the encoder and decoder.

3) Critic network: The structure of the critic network \( C \) is similar to that of the encoder in the segmentor network \( S \). It extracts the deep features of the input through a series of convolution layers. Through GAN loss, the critic network \( C \) constantly learns to distinguish whether the input is a ground truth label matrix or a predicted label matrix to provide critical information for the segmentor network \( S \). More details including the activation functions and the use of batch normalization are shown in Fig. 13.

4) Loss Function: In our task, the number of real target cells in the RDM is much less than the number of cells that do not contain a real target. The standard 0-1 cross-entropy loss cannot deal with the problem of sample imbalance. For example, when the number of negative samples is too large and accounts for most of the total loss, the optimization direction of the model is not necessarily optimal. Therefore, we use the focal loss proposed by Lin et al. [46] as the semantic segmentation loss function for the segmentor network. The focal loss is defined as follows:

\[
\mathcal{L}_{\text{Focal Loss}} = \begin{cases} 
-\alpha(1 - p)^\gamma \log(p), & \text{if } y = 1 \\
-(1 - \alpha)p^\gamma \log(1 - p), & \text{if } y = 0 
\end{cases} \tag{13}
\]

where \( p \) represents the value of cells in the predicted label matrix, \( y \) denotes the ground truth label matrix, \( \alpha \) is used to balance the number of positive and negative samples, and \( \gamma \) is used to adjust the weight of hard samples. For more arguments, please refer to [46].

In addition to the loss of semantic segmentation, we also use adversarial training to provide extra supervision. Given the \( n \)th frame RDM \( x_n \) and the corresponding ground truth label \( y_n \), the GAN loss can be defined as

\[
\min_{\theta_S} \max_{\theta_C} \mathcal{L}_{\text{GAN}}(\theta_S, \theta_C) = \frac{1}{N} \sum_{n=1}^{N} \ell_{\text{mae}} \left( f_C \left( \left\{ x_n, S(x_n) \right\}_{\text{cat}} \right), f_C \left( \left\{ x_n, y_n \right\}_{\text{cat}} \right) \right) \tag{14}
\]

where \( N \) denotes the number of training frames, \( \theta_S \) and \( \theta_C \) denote the parameters of the segmentor network \( S \) and critic network \( C \), respectively, \( \left\{ x_n, S(x_n) \right\}_{\text{cat}} \) denotes the input RDM concatenated with the predicted label matrix, \( \left\{ x_n, y_n \right\}_{\text{cat}} \) denotes concatenating the input RDM with its ground truth label matrix, \( f_C(\cdot) \) denotes the features extracted by the critic network \( C \), and \( \ell_{\text{mae}} \) is the mean absolute error.

The focal loss directly constrains the predicted label matrix, expecting it to be the same as the ground truth matrix. The GAN loss constrains the predicted label matrix on the deep feature
Fig. 13. Architecture of our model. The input to the model is composed of an RDM and two-coordinate codes ranging from –1 to 1 (horizontal and vertical coordinates, respectively). The settings for each layer of the model are marked above the Conv box. Conv3 × 3 represents that the size of the corresponding convolution kernel is 3 × 3C, S, and P represent the channel number, stride, and padding of the corresponding convolution operation, respectively. For example, the first blue box on the top left represents a convolution layer with kernel size 1 × 1 (Conv1 × 1), output channel 64 (C64), stride 1 (S1), and padding 0 (P0).

level and expects that the feature map of the predicted label matrix is the same as that of the ground truth matrix. Specifically, in the GAN loss, the segmentor network S and critic network C are playing a min–max game: S expects that there is no difference between the features of the predicted label matrix and the features of the ground truth after C, while C wants to be a strict inspector that can always distinguish the difference between the predicted label matrix and the ground truth matrix.

5) Training label generation: For data-driven supervised learning, it is essential to generate accurate ground truth labels. However, it is difficult to manually annotate the ground truth labels on an RDM. Therefore, we make use of LiDAR point clouds to automatically and efficiently generate accurate ground truth labels for the RDM. Usually, LiDAR point clouds have a very high spatial resolution (0.1°). Under normal environmental conditions, LiDAR point clouds have good stability and are easy to obtain directly. However, unlike radar, there is no Doppler dimension in LiDAR point clouds, which means that we cannot use a LiDAR point cloud to annotate an RDM directly. Therefore, in our method, we propose a new method for automatic RDM ground truth label generation based on LiDAR with the following five steps.

Step 1. Synchronization and Calibration: To provide an accurate ground truth, first, accurate timestamps of each frame of the LiDAR and radar data are recorded. The LiDAR data have a higher frequency. For each frame of the radar data, the corresponding frame of the LiDAR data is selected by choosing the LiDAR frame that has the timestamp closest to the timestamp of the radar frame. In addition, to avoid a static time offset of timestamps between the two sensors, we use a method based on event synchronization to manually compensate the static time offset of each group of collected data, and the synchronization accuracy can reach no more than one frame.

Fig. 14. Illustration of using corner reflectors for optimization in extrinsic calibration.

The initial extrinsic calibration between the radar and LiDAR is first obtained from a structural design file. Considering the installation error, after installing, a transformation matrix T is generated by measuring the relative position between the two sensors and can reach millimeter-scale precision. Then, a rotation matrix R is optimized by setting corner reflectors [47] distributed in different ranges and angles as shown in the following steps.

Step 2. LiDAR Point Cloud Filter: For the LiDAR point clouds, the points out of the field of view (FOV) of the radar are first removed. Then, the point clouds of the ground are removed using the method in [48] based on adjacent beam comparison. By using the density-based spatial clustering of applications with noise (DBSCAN) algorithm [49] to cluster the LiDAR point clouds, a few scattered noise points are removed.

Step 3. Cell Ground Truth Preselection: In the mean value RDM of multiple RDMs generated through radar signal processing, we set a relatively low threshold for ground truth
preselection. The label of a cell is set to 0 (which means there is no target) if the energy value of the cell is lower than the threshold.

**Step 4: Radar Point Cloud Screening:** After Step 3, for the remainder, we use the DOA estimation to generate corresponding radar point clouds. For each point, if there are LiDAR points within its neighborhood in 3-D Euclidean space, the label of the corresponding cell is set to 1 (which means there is a target). If not, the label is set to 0.

**Step 5: Label Matrix Generation.** By combining the results of Step 3 and Step 4, a label matrix, which contains labels of 0/1, is generated. The sizes of the label matrix and the RDM are the same.

An illustration of the last three steps of our training label generation method is shown in Fig. 15.

### B. Radar Point Cloud Spatiotemporal Filter

We observe that the radar point clouds of real targets are usually relatively stable in adjacent frames; that is, the real target points are not so glittery. In addition, different from LiDAR point clouds, each point in mmWave radar point clouds contains Doppler velocity information. Inspired by these, to further remove some scattered clutter points and improve the quality of the generated point clouds, we propose an RSTF to process the raw radar point clouds generated by the RPDNet.

The RPDNet can detect valid target cells in the RDM. For the detected cells, through DOA estimation, mmWave radar point clouds can be generated. In this way, the transformation from the 2-D range–Doppler space of the RDM to the 4-D \((x, y, z, v)\) space of the radar point clouds can be achieved. For the RSTF module, the input is 4-D radar point clouds. The RSTF removes clutter by observing the neighborhoods of each point (spatial information) in adjacent frames (temporal information).

As shown in Fig. 16, we first utilize the Doppler velocity of each radar point in the previous frame to compensate for the movement from the previous frame to the current frame. Then, we select a region for each point of the current frame. If there is no point from the previous frame point clouds (with added compensation) in this region, this point in the current frame is regarded as a clutter point and is removed.

One minor defect is that, there will be a one frame delay when distinguishing the newly appeared real target points from the clutter points, but we think it is acceptable for most robot application scenes. By using the radial velocity, the RSTF can be adapted to scenes with different moving and static targets. The RSTF can effectively filter the clutter points in point clouds generated by the RPDNet and further improve the quality of radar point clouds.

First, to gain the spatial compensation from the previous frame to the current frame, the Doppler velocity of radar point clouds is used for platform ego-motion estimation. As radar Doppler is a measure of the radial relative motion between the sensor and target, when every target in a scene is stationary and only the sensor platform is moving with the platform, referring to [50], the Doppler velocity \(q_{i,v}^k\) of target \(i\) in the \(k\)th frame can be expressed as

\[
q_{i,v}^k = \begin{bmatrix}
\sin \theta_i^k \
\cos \theta_i^k \cos \phi_i^k \
\cos \theta_i^k \sin \phi_i^k \
\end{bmatrix}
\]

where \(\theta_i^k\) and \(\phi_i^k\) denote the azimuth and elevation of the target, respectively, and can be calculated from point \((q_{i,x}^k, q_{i,y}^k, q_{i,z}^k)\).

We define the radar point clouds in frame \(k\) as \(\Omega_k = \{q_1^k, q_2^k, \ldots, q_n^k\}\). For each point in \(\Omega_k\), Formula (15) holds, and an overdetermined equation about \((v_{x}^k, v_{y}^k, v_{z}^k)\) can be generated. Based on ordinary least squares, the radial velocity of the \(k\)th frame can be estimated. In addition, if there are dynamic targets in the environment, the dynamic target points can be removed based on random sample consensus [51]. Assuming that the velocity of the mmWave radar will not change much between two frames, a translation matrix \(T_k\) between the current frame \(k\) and the previous frame \(k-1\) can be calculated as

\[
T_k = \begin{bmatrix}
\Delta x_k \\
\Delta y_k \\
\Delta z_k
\end{bmatrix}
\cdot
\begin{bmatrix}
v_{x}^k \\
v_{y}^k \\
v_{z}^k
\end{bmatrix}
\]

where \(\Delta t_k\) denotes the time gap between frame \(k\) and frame \(k-1\). When there is an IMU equipped on the platform, the rotation matrix \(R_k\) between the two frames can be obtained from the IMU orientation measurement, and when there is not, \(R_k\) is set as the identity matrix \(I\). In most cases where the radar moves on a plane, the compensation can degrade to two dimensions, as shown in Fig. 16.
Algorithm 1: Radar Point Cloud Spatiotemporal Filter.

Input: $\Omega_R^k$, $\Omega_R^{k-1}$, $\delta_{\text{filter}}$, $T_k$, $R_k$
Output: $\Omega_R^k$

1: $\Omega_R^{k-1} \leftarrow \{q_1^{k-1}, q_2^{k-1}, \ldots, q_m^{k-1}\}$
2: for $i = 1, 2, \ldots, m$ do
3: \[
\begin{bmatrix}
q_x^i \\
q_y^i \\
q_z^i
\end{bmatrix} = R_k \begin{bmatrix}
q_x^i \\
q_y^i \\
q_z^i
\end{bmatrix}^T - T_k
\]
4: end for
5: end
6: $\Omega_R^k = \emptyset$
7: for $j = 1, 2, \ldots, n$ do
8: \[
\text{if } \exists q_j^{k-1} \in \Omega_R^{k-1}, \text{s.t. } d(q_i^{k-1}, q_j^{k-1}) \leq \delta_{\text{filter}} \text{ then}
\]
9: \[
\Omega_R^{k} = \Omega_R^{k} \cup q_j^{k-1}
\]
10: end if
11: end
12: end for
13: end

With the estimated translation matrix $T_k$ and rotation matrix $R_k$, adjacent frame spatial compensation can be performed. The radar point clouds generated by the RPDNet in frames $k$ and $k-1$ are $\Omega_R^k = \{q_1^k, q_2^k, \ldots, q_m^k\}$ and $\Omega_R^{k-1} = \{q_1^{k-1}, q_2^{k-1}, \ldots, q_m^{k-1}\}$, respectively. The output of the algorithm is the final set of radar point clouds defined as $\Omega_R^k$. For the parameter, we define the neighborhood range threshold of the region as $\delta_{\text{filter}}$. Define one radar point $q = [q_x, q_y, q_z]$. Define $d(q_1^k, q_2^k) = \sqrt{(q_1 - q_2)^2 + (q_1 - q_2)^2 + (q_1 - q_2)^2}$. Under the above definitions and with the estimated movement, the pseudocode for the RSTF is shown in Algorithm 1.

V. EXPERIMENT AND EVALUATION

In this section, we introduce our mobile acquisition platform and the composition of our dataset which includes 12 sequences collected in three types of scenes, streets, indoors, and on water surfaces. Additionally, we propose a new evaluation metric for the quality of mmWave radar point clouds and carry out quantitative evaluation for different radar point cloud generation methods. The result shows that our method generates a point cloud of higher quality. In addition, the analysis and experiment of computational efficiency show that our method can run on a real-time embedded system. Finally, we conduct several ablation studies to evaluate the contribution of each component of our method.

A. Dataset

To collect real-world data under different scenes for training and testing, we build a mobile acquisition platform equipped with multiple sensors, which can be directly applied to indoor scenes, streets, and water surfaces. The mobile platform is shown in Fig. 17. The radar equipped on the platform is a Texas Instruments 77 GHz FMCW radar AWR1843 with an array of three transmitting antennas and four uniformly spaced receiving antennas. Therefore, by using time-division multiplexing multiple-input multiple-output (MIMO), our radar system can reach 14.3° and 57.2° azimuth and elevation angular resolutions, respectively. As mentioned in Section III, different radar waveform parameters lead to different radar detection performances. Considering the requirements of environment perception for typical mobile robots such as mobile robots used on streets, in inland waters, and indoors, we design the radar waveform parameters as listed in Table I.

A 16-beam LiDAR is used for collecting point cloud data as the ground truth. The horizontal and vertical resolution of the LiDAR are 0.18° and 2°, respectively. In addition, a global navigation satellite system (GNSS) receiver with real-time kinematic and a six-axis IMU are used to provide accurate location and pose information. In addition, an RGB camera is used to record the corresponding scenes.

Our dataset contains 12 sequences collected in different scenes including streets, indoor environments, and water surfaces. There are 28 000 frames of data in total. Detailed information on each sequence is shown in Table II. The scenes of each sequence are different. Our dataset has also been published.\footnote{Available: https://github.com/thucyw/RPDNet}

In addition, the existing ColoRadar dataset \cite{52} also provides mmWave radar, LiDAR, and IMU data. The dataset consists of three mobile robot application scenes, including indoor, outdoor, and mine environments, and there are seven subscenes in total (IR lab, ASPEN lab, EC hallways, EC courtyard, creek path, open areas of the mine, and passageways of the mine). Each subscene contains multiple sequences. We choose the first two sequences of each subscene as the training set and the other as the test set.\footnote{In the creek path sequences, the velocity of the platform sometimes exceeds twice the maximum detection velocity of the radar, which may lead to phase compensation error in the DOA estimation. In this case, we remove the frames in which the platform moves too fast in our experiment to avoid this problem.} Therefore, for the ColoRadar dataset, we use 14 sequences as the training set and 38 sequences as the test set.

B. Evaluation Metrics

In a complex environment, a well-performing radar detector means that among the 4-D radar point clouds, on the premise
that the clutter points are reduced as much as possible, the point clouds of real targets are denser [53]. At present, for mobile robots, point clouds of high quality are usually indispensable, and using multiplanar LiDAR is a convenient way to gather high-quality point clouds. We expect the 4-D point clouds of radar to be closer to the LiDAR point clouds. Thus, we use the point clouds of the 16-beam LiDAR as the ground truth and design a new evaluation metric for the quality of 4-D radar point clouds to evaluate the performance of the different methods.

We define the set of LiDAR and radar point clouds in one frame as $\Omega_{L1}$ and $\Omega_{R1}$, respectively. The LiDAR point vector is recorded as $p = [p_x, p_y, p_z]^T$, and the radar point vector is recorded as $q = [q_x, q_y, q_z]^T$. We denote the distance between two points $p$ and $q$ in Euclidean space by $d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2}$.

The set of radar clutter points $\Omega_{R2}$ which is a subset of radar point clouds $\Omega_{R1}$ is defined as

$$\Omega_{R2} = \{q \in \Omega_{R1}, s.t., d(q, p) > \delta_1, \forall p \in \Omega_{L1}\}$$

(17)

where $\delta_1$ is the distance threshold we set. Formula (17) shows that, for a radar point, if no LiDAR point is within distance $\delta_1$ of this point, the radar point is regarded as a clutter point. A lower number of clutter points means that the method has better performance in suppressing radar clutter.

We define $\Omega_{L2}$, which is a subset of $\Omega_{L1}$ as

$$\Omega_{L2} = \{p \in \Omega_{L1}, \exists q \in \Omega_{R1}, s.t., d(p, q) < \delta_2\}$$

(18)

where $\delta_2$ is the distance threshold we set. Formula (18) shows that the subset $\Omega_{L2}$ includes the LiDAR point at which there are radar points within distance $\delta_2$. To evaluate the density of the radar point clouds of real targets generated using our method, we define a radar point cloud density level (RPCDL) as

$$\text{RPCDL} = \frac{N(\Omega_{L2})}{N(\Omega_{L1})}$$

(19)

where the $N(\Omega_{L1})$ and $N(\Omega_{L2})$ are the numbers of points in $\Omega_{L1}$ and $\Omega_{L2}$, respectively. The RPCDL we defined is similar to the probability of detection for radar [54]. A higher RPCDL means that the radar point clouds are closer to the LiDAR point clouds, and the performance of the method is better.

![Fig. 18. Illustration of the point set in the evaluation metric we defined.](image)

With a higher detection threshold, the number of detected clutter points can decrease, but the number of valid detected points can also decrease. Conversely, with a lower detection threshold, the number of valid detected points may increase, but the number of clutter points can also increase. Therefore, to evaluate the performance of the detector, we use the curve of the RPCDL [defined in Formula (19)] as a function of the number of radar clutter points [defined in Formula (17)] under different thresholds. A higher RPCDL curve means better detection performance. An illustration of the evaluation metric is shown in Fig. 18. To some degree, our performance curve can be regarded as a particular receiver operating characteristic curve that is famous for radar detector performance evaluation in the signal processing community.

### C. Training Details

Because the mmWave radar waveform parameters are different between our dataset and the ColoRadar dataset, the sizes of the RDMs in our dataset and in the ColoRadar dataset are different. (The RDM in our dataset is $512 \times 128$, while the RDM in the ColoRadar dataset is $128 \times 128$.) Therefore, we train two models for the two dataset. For the two models, the training methods are the same. We use a backpropagation algorithm to train the model. In the training process, two losses (the focal loss and the GAN loss) are used alternately, and only when the GAN loss is used can the $C$ network participate in the training.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Frame</th>
<th>Type</th>
<th>Training</th>
<th>Test</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq-01</td>
<td>2000</td>
<td>Land-Parking Lot 1</td>
<td>✓</td>
<td>✓</td>
<td>multiple vehicles</td>
</tr>
<tr>
<td>Seq-02</td>
<td>2000</td>
<td>Land-Parking Lot 2</td>
<td>✓</td>
<td>✓</td>
<td>multiple vehicles</td>
</tr>
<tr>
<td>Seq-03</td>
<td>2000</td>
<td>Land-Street 1</td>
<td>✓</td>
<td>✓</td>
<td>narrow street with pedestrians</td>
</tr>
<tr>
<td>Seq-04</td>
<td>2000</td>
<td>Land-Street 2</td>
<td>✓</td>
<td>✓</td>
<td>park path</td>
</tr>
<tr>
<td>Seq-05</td>
<td>2000</td>
<td>Land-Street 3</td>
<td>✓</td>
<td>✓</td>
<td>multiple vehicle targets</td>
</tr>
<tr>
<td>Seq-06</td>
<td>4000</td>
<td>Land-Street 4</td>
<td>✓</td>
<td>✓</td>
<td>sidewalk</td>
</tr>
<tr>
<td>Seq-07</td>
<td>2000</td>
<td>Land-Street 5</td>
<td>✓</td>
<td>✓</td>
<td>park path</td>
</tr>
<tr>
<td>Seq-08</td>
<td>2000</td>
<td>Land-Indoor 1</td>
<td>✓</td>
<td>✓</td>
<td>narrow corridor and glass wall</td>
</tr>
<tr>
<td>Seq-09</td>
<td>2000</td>
<td>Land-Indoor 2</td>
<td>✓</td>
<td>✓</td>
<td>narrow corridor and glass wall</td>
</tr>
<tr>
<td>Seq-10</td>
<td>2000</td>
<td>Water-Inland Water 1</td>
<td>✓</td>
<td>✓</td>
<td>inland waters</td>
</tr>
<tr>
<td>Seq-11</td>
<td>4000</td>
<td>Water-Inland Water 2</td>
<td>✓</td>
<td>✓</td>
<td>inland waters</td>
</tr>
<tr>
<td>Seq-12</td>
<td>2000</td>
<td>Water-Inland Water 3</td>
<td>✓</td>
<td>✓</td>
<td>along banks of lake</td>
</tr>
</tbody>
</table>
Fig. 19. Evaluation result of the radar point cloud generated by different detectors on our dataset and the ColoRadar dataset. For the two thresholds used in the evaluation (see Section V-B), we set $\delta_1 = 0.5 \text{m}$ and $\delta_2 = 0.3 \text{m}$.

(a) Street scene (our dataset). (b) Indoor scene (our dataset). (c) Water surfaces scenes (our dataset). (d) Mine scene (ColoRadar). (e) Indoor scene (ColoRadar). (f) Outdoor scenes (ColoRadar).

Through the GAN loss, $S$ and $C$ confront each other: we first fix the segmentor network to train the critic network aiming to maximize the GAN loss and then fix the critic network to train the segmentor aiming to minimize the GAN loss.

In the training process using the focal loss, we use the Adam [55] optimizer with a learning rate of 0.001. In the training process using the GAN loss, we use the RMSProp optimizer [56] with a learning rate of 0.0001. We use an NVIDIA GeForce RTX 3090 graphics processing unit (GPU) and trained the model for 500 epochs for our dataset which took approximately about 40 h, and 500 epochs for the ColoRadar dataset, which took approximately about 10 h.

D. Quantitative Evaluation

Based on the evaluation metrics we set, we compare our method with five other methods. For radar point cloud generation methods, we compare our method with two classic radar detectors, CA-CFAR and OS-CFAR [24], and a modified CFAR detector [21]. For CFAR-based methods, the number of detected point clouds is mainly influenced by the detector threshold factor $\alpha$. Therefore, by setting different $\alpha$ values from 2 to 8 (stride is 0.5), the detection results under different numbers of points are obtained (the mean number of points varies from tens to thousands with different thresholds). In addition, we also compare our method with two other methods that apply postprocessing (DBSCAN [49] and PointNet [12]) to point clouds to remove clutter points. The work [17] requires the antenna array arrangement parallel to the moving direction of the platform and requires a two-chip cascade mmWave radar while our method is not limited by the movement and the radar hardware. Therefore, we cannot compare our method with [17] directly on our dataset and the ColoRadar dataset.

For our method, by setting different confidence thresholds, detection results under different numbers of points are also obtained. We conduct experiments with the same distance thresholds $\delta_1 = 0.5 \text{m}$, $\delta_2 = 0.3 \text{m}$ to evaluate the performance of different methods. For our dataset, we test data from streets, indoor environments, and water surfaces separately. For the ColoRadar dataset, we test data from indoor, outdoor, and mine environments separately. By setting different thresholds, the relationships between the number of clutter points and the RPCDL are obtained. The evaluation result is shown in Fig. 19. It can be observed that when the numbers of clutter points are the same, the RPCDL of our detector is higher than those of other methods, which means that the detection probability of our detector is higher. For example, in Fig. 19(a), when the number of clutter points is approximately 50, the density level of radar point clouds generated by our detector is three times the density of radar point clouds generated by the CA-CFAR detector. On the ColoRadar dataset, our method still obviously outperforms the CFAR detector. However, as the size of the RDM in the ColoRadar dataset is relatively small, the generated point clouds are sparser than the point clouds in our dataset and the RPCDLs are relatively lower than those in our dataset.
et al.

Ω is defined as the Euclidean distance and the confidence $d_{qE}$ and its nearest LiDAR point $\alpha_{C}$.

The result shows that, without the Doppler velocity compensation in the RSTF module, the average RPCDL increases by approximately 2.1%. When there is Doppler velocity compensation in the RSTF, the average RPCDL increases by approximately 2.3% compared to the result without the RSTF. Therefore, Doppler velocity compensation can be applied to fast-moving scenes and plays an important role. However, if the speed of the platform exceeds the maximum Doppler velocity, the Doppler velocity compensation will fail. In real-world applications, this problem can be avoided by designing radar waveform parameters according to the maximum platform moving speed.

E. System Efficiency

For real-world applications of mobile robots, the efficiency of a method is also very important. Therefore, we test and analyze the computational complexity of different modules of our method. For the radar point generation module, for the RPDNet, we perform experiments using an embedded platform NVIDIA Xavier NX. The average time consumption is approximately 32 ms. When the GPU in the embedded platform is disabled, the average time consumption is approximately 730 ms. In comparison, the time consumption of OS-CFAR is approximately 20 ms. When using only the CPU, the computation time of our model is significantly longer than that of OS-CFAR. However, an embedded platform with a GPU is now more widely used in mobile robots, and with a GPU, the computation time of our model is close to that of classical OS-CFAR.

In the point cloud postprocessing module, the computational complexity of the RSTF and DBSCAN are both $O(n \log n)$. However, in most cases, RSTF runs faster than DBSCAN. The test result is shown in Table III. The results show that the efficiency of the proposed method is close to that of classical methods so that the method can be used for real-time mobile robot applications.

![Fig. 20. Distribution error of radar points generated by our method and OS-CFAR on our dataset and the ColoRadar dataset. (a) Our dataset. (b) ColoRadar.](image)

In addition, to show the details of the mmWave radar point clouds, we also evaluate the spatial distribution error of the mmWave point clouds. For a point $q$ in $\Omega_{R1}$, the distribution error $\text{err}_{q}$ is defined as the Euclidean distance $d(q, p_{\text{near}})$ between $q$ and its nearest LiDAR point $p_{\text{near}}$. For a fair comparison, we set the CFAR detector threshold factor $\alpha$ and the confidence threshold for our method separately to ensure that the numbers of points generated by the two methods are approximately the same.

The evaluation results on our dataset and the ColoRadar dataset are visualized using the error distribution histogram shown in Fig. 20. The evaluation result shows that the mean distribution errors of the point clouds generated using our method are 0.44 (our dataset) and 0.31 (ColoRadar dataset). For radar point clouds generated using the CFAR detector, the mean errors are 0.79 (our dataset) and 0.54 (ColoRadar dataset). Therefore, by evaluating the point distribution error, our method also shows better performance on point cloud detection.

An example of point cloud comparison in one frame is shown in Fig. 21. The proposed detector generates denser real target point clouds with fewer clutter points compared to the CFAR detector. The evaluation result proves that, for the application of mobile robot environment perception, the radar detector we propose outperforms classical radar detectors in different mobile robot application scenes. By using our method, we can generate higher quality mmWave radar point clouds.

1) Generalization ability analysis: To test the generalization ability of our model across datasets, we use the model trained on our dataset to test on the ColoRadar dataset directly. We also use the model trained on the ColoRadar dataset to test on our dataset directly. The result is shown in Fig. 22. As can be seen, the RPCDL across datasets decreases slightly compared to using a same dataset for training and testing. Our model performs well on the generalization ability across different datasets. In addition, the evaluation results on data collected on streets, in indoor environments, and on water surfaces show that our method has good scene generalization ability. In particular, although the data of indoor scenes are not contained in the training set for the model, our method still shows good performance [as shown in Fig. 19(b)].

In the creek path subscene, the sensor moves rapidly through an outdoor environment [52]. Therefore, to verify the influence on the high-speed movement to the RSTF module, we test on these sequences for comparison experiments. The result shows that, without the Doppler velocity compensation in the RSTF module, rapid movements will cause many valid points to be falsely treated as clutter points and removed. In this case, with the same number of clutter points, compared to the points generated without the RSTF module, the average RPCDL decreases by approximately 2.1%. When there is Doppler velocity compensation in the RSTF, the average RPCDL increases by approximately 2.3% compared to the result without the RSTF. Therefore, Doppler velocity compensation can be applied to fast-moving scenes and plays an important role. However, if the speed of the platform exceeds the maximum Doppler velocity, the Doppler velocity compensation will fail. In real-world applications, this problem can be avoided by designing radar waveform parameters according to the maximum platform moving speed.

### Table III

<table>
<thead>
<tr>
<th>Method</th>
<th>Time [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RPDNet</td>
</tr>
<tr>
<td>Post-Processing</td>
<td>DBSCAN [49]</td>
</tr>
<tr>
<td></td>
<td>RSTF</td>
</tr>
</tbody>
</table>

[*] The result 730 is measured when the GPU is disabled.
Fig. 21. Comparison of point clouds generated by CFAR detector and our method. (a) Street scene. (b) Indoor scene. (c) Water surface scene.

Fig. 22. Evaluation results for generalization ability of our model across datasets. (a) Test on our dataset. (b) Test on ColoRadar.

TABLE IV
EVALUATION RESULTS FOR OBJECT DETECTION USING DIFFERENT POINT CLOUDS

<table>
<thead>
<tr>
<th></th>
<th>VoteNet [32]</th>
<th>Danzer et al. [33]*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$AP^{35}$</td>
<td>$AP^{50}$</td>
</tr>
<tr>
<td>Lidar</td>
<td>77.43</td>
<td>67.82</td>
</tr>
<tr>
<td>OS-CFAR</td>
<td>50.58</td>
<td>21.38</td>
</tr>
<tr>
<td>Ours</td>
<td>70.83</td>
<td>47.44</td>
</tr>
</tbody>
</table>

[*] $AP^{35}$ and $AP^{50}$ [%] denote the average detection accuracy when the IOU is set to 0.35 and 0.5, respectively.

[*] For [33], the LiDAR data input includes $x$, $y$, and $z$ of point clouds. For the mmWave radar point clouds, the results include only using radar data $x$, $y$, $z$ as input (after backslash) and using radar data $x$, $y$, $v_r$, $e$ ($e$ denotes the energy intensity) as input (before backslash).

TABLE V
EVALUATION RESULTS OF THE ICP USING DIFFERENT POINT CLOUDS

<table>
<thead>
<tr>
<th></th>
<th>Street</th>
<th>Indoor</th>
<th>Inland Water</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS-CFAR</td>
<td>0.34/4.07</td>
<td>0.21/2.24</td>
<td>0.37/4.43</td>
</tr>
<tr>
<td>Ours</td>
<td>0.22/2.63</td>
<td>0.15/1.90</td>
<td>0.23/3.03</td>
</tr>
</tbody>
</table>

[*] The values in the table denote the RTE/RRE [°/m].

Fig. 23. Ablation study for the critic network in the RPDNet. (a) Our dataset. (b) ColoRadar.

F. Ablation Study

To evaluate the contribution of different components in our method, we perform ablation studies on our dataset and the ColoRadar dataset.

1) Critic network: In the RPDNet, we add the critic network to provide stronger supervision to promote the learning of RDM characteristics and make the semantic segmentation results fine-grained. The result is shown in Fig. 23. It can be seen that the critic network improves the performance of the model. Without
In our method, the raw radar point clouds $x$, $y$, $A$, $E$, $\phi$ are processed using the RSTF postprocessing module to generate the final radar point clouds. The results of the ablation study for the RSTF module and the Doppler velocity compensation in the RSTF are shown in Fig. 24. The RSTF module can improve the quality of point clouds. By using the RSTF module, with the same number of clutter points, the RPCDL decreases approximately 4% on average.

1) **Point registration:** In our method, the raw radar point clouds generated by the RPDNet are processed using the RSTF postprocessing module to generate the final radar point clouds. The results of the ablation study for the RSTF module and the Doppler velocity compensation in the RSTF are shown in Fig. 24. The RSTF module can improve the quality of point clouds. By using the RSTF module, with the same number of clutter points in the radar point clouds decreases by approximately 40% on average. The comparison of point clouds generated with and without the RSTF module is shown in Fig. 25. The Doppler velocity compensation in the RSTF can avoid valid points being falsely removed by the RSTF. In our dataset, the moving velocity of the platform is relatively low; so the effect of the Doppler velocity compensation is small. However, in high-speed moving scenes, Doppler velocity compensation plays an important role (as shown in the experimental results in Section V-D).

VI. REAL-WORLD APPLICATION EXPERIMENT

In Section V, using the evaluation metric we define, the proposed method achieves better performance compared to classic radar detectors. To verify the effectiveness of the evaluation metric, we define and evaluate the contributions of mmWave radar point cloud quality improvement to mobile robot environment perception. We further carry out experiments on two typical mobile robot environment perception tasks: object detection and localization and mapping.

A. **Object Detection**

Object detection is an important mobile robot environment perception task. High-quality environment perception data can help improve the performance of object detection tasks. Therefore, we compare the results using the same object detection algorithm based on radar point clouds generated using our method and other typical methods, to further test the value of mmWave radar point clouds generated by our method in the applications of mobile robot environment perception.

The street scene sequences in our dataset contain vehicles of different types, ranges, and azimuths. Therefore, based on Seq-02 and Seq-05, we establish a vehicle detection task. In these sequences, there are 5813 vehicle targets. We follow a 6:4 random split for the training set and test set.

The experiment is based on two typical point cloud object detection methods, the modified PointNet [33] and VoteNet [32]. VoteNet is widely used for 3-D object detection in point clouds. The method proposed in [33] is based on PointNet [32] and is specifically designed for vehicle detection based on mmWave radar point clouds. The input to [33] includes $x$, $y$, Doppler velocity $v_d$, and radar cross-sectional values $\phi$.

We test the two methods on mmWave radar point cloud generated using the OS-CFAR detector and the proposed method. The results are shown in Table IV. As shown in Fig. 26, the clutter points in radar point clouds can lead to false alarm detection and the sparsity of point clouds can lead to missed detection of some real objects. Based on the same object detection method, the point clouds generated using the proposed method can bring significant improvement to the task of object detection.

B. Localization and Mapping

Localization and mapping is an essential task for mobile robots. High-quality environment perception data can help improve the performance of localization and mapping. Therefore, we compare the results using the same localization and mapping algorithm based on radar point clouds generated by our method and other typical methods.

1) **Point registration:** Point registration is one of the key parts of localization. The iterative closest point (ICP) algorithm is a typical point cloud registration method that is widely used for LiDAR point cloud registration. By using point cloud registration, we can obtain the translation and rotation matrix between the poses of different frames. Therefore, we take the transformation relationship generated by the ICP on the high-quality LiDAR point clouds as the ground truth. Then, we compare the results generated using the ICP on mmWave radar points with the ground truth to test the influence of radar point cloud quality on point registration.

For evaluation, referring to the evaluation metrics in [58], we use the mean relative pose error, including the mean relative translational error (RTE) and mean relative rotation error (RRE). The result is shown in Table V. For three types of scenes, streets, indoor environments, and inland waters, the RTE decreases by about 11%, 6%, and 14%, respectively. The result shows that the radar point clouds generated using the proposed method are better for point cloud registration.

2) **Self-localization:** In addition to point cloud registration, we apply mmWave radar point clouds to self-localization for mobile robots to test the influence of point cloud quality on
localization accuracy. We used two methods, the lightweight and groud-optimized lidar odometry and mapping (LeGO-LOAM) [59] which is widely used for SLAM based on LiDAR data, and a method specifically designed for mmWave radar based on FastSLAM [60] proposed by Schuster et al. [27]. We carry out experiments on four different scenes based on our dataset. For LeGO-LOAM, we use the 3-D LiDAR point clouds as input. For the radar SLAM method of Shuster et al. [27], in addition to the 3-D radar point cloud, an odometry sensor is also used. Therefore, in the experiment, we use the method proposed in [51] for ego-motion estimation based on 4-D radar point clouds and add IMU data to serve as the odometry data for [27]. Centimeter-level accuracy GNSS localization is taken as the ground truth.

The result is shown in Fig. 27. We calculate the absolute translation error (ATE) [58] of each sequence, and the results are shown in Table VI. Using point clouds generated by our method, the ATE decreases by 49% on average for the method proposed in [27] compared to using mmWave radar point clouds generated by the OS-CFAR. By using the Doppler velocity of 4-D mmWave radar data, the motion of the platform can be estimated. Therefore, in some scenes where the 3-D point cloud matching degrades, compared to using 3-D LiDAR point clouds, using the 4-D radar point clouds can achieve more accurate localization results.

3) Mapping: We also conduct experiments using different point clouds to build 3-D point cloud maps. We use high-accuracy GNSS location and IMU pose information to build maps. The mapping results are shown in Fig. 28. The 3-D point cloud map based on radar point clouds generated by our method contains denser points of real targets with fewer clutter points. Compared with radar point clouds generated by the OS-CFAR detector, the map based on point clouds generated by our method shows richer scene information.
Through the above experiments, we can see that the mmWave radar point clouds generated using our method can bring significant improvement to the task of localization and mapping.

VII. Conclusion

In this article, aiming at mmWave radar point cloud applications in mobile robot perception tasks, we proposed a novel method for radar point cloud generation, including the RPDNet and RSTF. The results of testing on the data collected in real-world scenes of streets, indoors, and water surfaces showed that our method generates radar point clouds of higher quality. The radar point clouds generated by our method contain fewer clutter points and denser point clouds of real targets. In addition, the higher quality point clouds generated by our method contribute more to the object detection and localization and mapping tasks of mobile robots and are of great value for mobile robot environment perception based on low-cost and commercial mmWave radar.

However, compared with multibeam LiDAR point clouds, there are still some shortcomings. Therefore, in the future, on the one hand, in the mmWave radar signal processing chain, in addition to a radar detector, a new method for DOA estimation can be explored for super-resolution angle estimation. On the other hand, the fusion of mmWave radar and other sensors, such as a camera, can be utilized to further increase the quality of mmWave radar point clouds.

APPENDIX A

Proof for Radar Point Cloud Angular Resolution

A. Theoretical Analysis

According to array signal processing theory, the azimuth and elevation of a target can be determined based on the received signal phase difference between different antennas, as shown in Fig. 29. According to Formula (3), in a typical single-chip mmWave radar, $d = \lambda/2$, and the numbers of virtual receiving antennas in the horizontal and vertical directions are $N_{rx}^h = 8$ and $N_{rx}^v = 2$, respectively. Therefore, for a target of any azimuth and elevation,

$$\theta_{res} \geq 14.3^\circ,$$  \hspace{1cm} (20)

$$\phi_{res} \geq 57.2^\circ$$ \hspace{1cm} (21)

where $\theta_{res}$ and $\phi_{res}$ denote the azimuth and elevation resolution, respectively.

Using typical array signal processing technology, the inherent angular resolution of a single-chip mmWave radar point cloud is relatively low. However, we find that for mobile robot applications, the Doppler resolution can be transformed to angular resolution.

According to Formulas (2) and (7)

$$v_{max} = \frac{v_{res} \cdot N_{chirp} \cdot \Delta v}{2}$$ \hspace{1cm} (22)

where $v_{max}$ denotes the maximum unambiguous velocity, $v_{res}$ denotes the velocity resolution, and $N_{chirp}$ denotes the number of chirps in one frame of the FMCW signal.

We define the radar coordinates as shown in Fig. 30. Assume that the motion direction of the platform is along the positive direction of the y-axis in the radar coordinate system, and the speed of the mobile robot platform is $v_b$. Assume that every target in the scene is stationary, and only the sensor platform is moving. The maximum unambiguous velocity of the mmWave radar is set to the same as the velocity of the platform, that is, $v_{max} = v_b$. For two points of static targets in the environment $q_1$ and $q_2$, assume that the ranges $R$ and azimuths $\theta$ of the two points are the same. The elevations of $q_1$ and $q_2$ are $\phi$ and $(\phi + \Delta \phi)$, respectively. The Doppler velocities of $q_1$ and $q_2$ are $v_{q1}$ and $(v_{q1} + \Delta v)$, respectively. The Doppler velocity $v_{q1}$ of $q_1$ is the radial velocity of the point; so

$$v_{q1} = v_b \cdot \cos \phi \cdot \cos \theta.$$ \hspace{1cm} (23)

For $p_2$

$$v_{q1} + \Delta v = v_b \cdot \cos (\phi + \Delta \phi) \cdot \cos \theta.$$ \hspace{1cm} (24)

With Formulas (23) and (24)

$$\Delta v = v_b \cdot \cos (\phi + \Delta \phi) \cdot \cos \theta - v_b \cdot \cos \phi \cdot \cos \theta.$$ \hspace{1cm} (25)

It can be seen that, through Doppler velocity resolution, the elevation resolution can be achieved. From Formula (25), when $|\Delta v|$ reaches the Doppler velocity $v_{res}$ $(|\Delta v| = v_{res})$, the $\Delta \phi$ that can be distinguished will reach the minimum value. Therefore, with Formula (22)

$$|\Delta v| = \frac{v_{res}}{v_b} = \frac{2}{N_{chirp}}.$$ \hspace{1cm} (26)

Considering the effective FOV of mmWave radar, we focus on the azimuth and elevation in the domain $\phi \in [-\frac{\pi}{3}, \frac{\pi}{3}], \theta \in [-\frac{\pi}{4}, \frac{\pi}{4}].$
\[ \theta = 0 \cdot \phi = 0 \cdot \pi \text{ reaches the maximum value}, \text{ and } \Delta \theta < 0. \]

\[ \cos (\phi + \Delta \phi) \cdot \cos \theta - \cos \phi \cdot \cos \theta = -\frac{2}{N_{\text{chirp}}} \cdot \cos \theta. \]

Then

\[ \Delta \phi = \arccos \left( -\frac{2}{N_{\text{chirp}}} \cdot \cos \theta + \cos \phi \right) - \phi. \]

According to the above formula, by increasing \( N_{\text{chirp}} \) in the radar waveform, the angular resolution can be increased. The partial derivative of \( \Delta \phi \) with respect to \( \phi \) is

\[ \frac{\partial(\Delta \phi)}{\partial \phi} = \frac{\sin \phi}{\sqrt{1 - (\cos \phi - \frac{2}{N_{\text{chirp}}} \cdot \cos \theta)^2}} - 1. \]

The partial derivative of \( \Delta \phi \) with respect to \( \theta \) is

\[ \frac{\partial(\Delta \phi)}{\partial \theta} = \frac{2 \cdot \sin \theta}{N \cdot (\cos \theta)^2 \cdot \sqrt{1 - (\cos \phi - \frac{2}{N_{\text{chirp}}} \cdot \cos \theta)^2}}. \]

In a typical single-chip mmWave radar system, \( N_{\text{chirp}} \) can be set as 128. Therefore, according to Formulas (29) and (30), when \( \theta = \pm \frac{\pi}{3} \), \( \phi = 0 \), \( \Delta \phi \) reaches the maximum value, and \( \Delta \phi = 14.36^\circ \).

Similarly, it can be proven that, when \( \phi \in [-\frac{\pi}{3}, \frac{\pi}{3}] \) and \( \theta \in [-\frac{\pi}{3}, \frac{\pi}{3}] \), \( \Delta \theta \leq 14.36^\circ \).

It can be seen from above that the Doppler velocity resolution can be converted into azimuth and elevation resolution, and for a single-chip mmWave radar, in mobile robot application scenes, the angular resolution achieved by the Doppler velocity resolution is significantly better than the typical angular resolution achieved by typically relying on different receiving antennas.

### B. Experimental Verification

We use a TI cascade mmWave radar MMWCAS-RF-EVM [61] which contains four single-chip radars for experiments. It contains 12 transmitting antennas and 16 receiving antennas. Using MIMO technology, it can achieve 86 equivalent antennas in the horizontal direction and 7 equivalent antennas in the vertical direction. According to Formula (3), using DOA estimation, the azimuth and elevation resolution can reach 1.4° and 16.3°, respectively. On the mobile platform, we use cascaded radar to collect raw analog to digital converter (ADC) data (the range resolution is 0.05 m, and the maximum unambiguous velocity is 1.85 m/s). For the number of chirps per frame \( N_{\text{chirp}} \), \( N_{\text{chirp}} \) is set to 16, 32, and 64 to collect three sequences of data. The speed of the platform is 1.2 m/s. The comparison results for the same scene are shown in Fig. 31.

For the comparison data, the same radar CFAR detector and detection parameters are used. The single-chip radar is one from the cascade radar. From Fig. 31(b) to (d), with the increase in \( N_{\text{chirp}} \), the density of the point cloud also increases. Compared to Fig. 31(d), in Fig. 31(a), although the number of radar antenna triples and the hardware cost increase significantly, the increase in point cloud density is not larger than the density increase due to the increase in \( N_{\text{chirp}} \). Therefore, the experimental results show that the angular resolution achieved by Doppler velocity resolution is significantly better than the typical angular resolution achieved by typically relying on different receiving antennas.
et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.

et al.
Yuwei Cheng received the B.S. degree in electronics information engineering from the Northwestern Polytechnical University, Xi’an, China, in 2018. He is currently working toward the Ph.D. degree in electronic engineering, Tsinghua University, Beijing, China. His current research interests include radar signal processing, millimeter wave radar, and its applications in robotics.

Jingran Su received the B.E. degree in automation and the master’s degree in computer science from Northwestern Polytechnical University, Xi’an, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree in computer science with the Department of Computing, The Hong Kong Polytechnical University. His current research interests include adversarial attacks and defenses, computer vision, and machine learning.

Mengxin Jiang received the B.S. degree in automation from the Department of Automation, Tsinghua University, Beijing, China, in 2021. She is currently with ORCA-Uboat, Shaanxi, China. Her current research interests include robotics localization and the applications of millimeter wave radar in robotics.

Yimin Liu (Member, IEEE) received the B.S. and Ph.D. degrees (Hons.) in electronics engineering from the Tsinghua University, Beijing, China, in 2004 and 2009, respectively. From 2004, he was with the Intelligence Sensing Lab (ISL), Department of Electronic Engineering, Tsinghua University. He is currently an Associate Professor with Tsinghua University, where his field of activity is to study new concept radar and other microwave sensing technologies. His current research interests include radar theory, statistic signal processing, compressive sensing, and their applications in radar, spectrum sensing, and intelligent transportation systems.