Deformable Radar Polygon: A Lightweight and Predictable Occupancy Representation for Short-Range Collision Avoidance

Xiangyu Gao*, Sihao Ding, and Harshavardhan Reddy Dasari

Abstract—Inferring the drivable area in a scene is crucial for ensuring a vehicle avoids obstacles and facilitates safe autonomous driving. In this article, we concentrate on detecting the instantaneous free space surrounding the ego vehicle, targeting short-range automotive applications. We introduce a novel polygon-based occupancy representation, where the interior signifies free space, and the exterior represents undrivable areas for the ego vehicle. The radar polygon consists of vertices selected from point cloud measurements provided by radars, with each vertex incorporating Doppler velocity information from automotive radars. This information indicates the movement of the vertex along the radial direction. This characteristic allows for the prediction of the shape of future radar polygons, leading to its designation as a “deformable radar polygon.” We propose two approaches to leverage noisy radar measurements for producing accurate and smooth radar polygons. The first approach is a basic radar polygon formation algorithm, which independently selects polygon vertices for each frame, using SNR-based evidence for vertex fitness verification. The second approach is the radar polygon update algorithm, which employs a probabilistic and tracking-based mechanism to update the radar polygon over time, further enhancing accuracy and smoothness. To accommodate the unique radar polygon format, we also designed a collision detection method for short-range applications. Through extensive experiments and analysis on both a self-collected dataset and the open-source RadarScenes dataset, we demonstrate that our radar polygon algorithms achieve significantly higher IoU-gt and IoU-smooth values compared with other occupancy detection baselines, highlighting their accuracy and smoothness.

Index Terms—Automotive radar, collision avoidance, deformable, lightweight, occupancy detection, radar polygon, short-range applications.

I. INTRODUCTION

The detection of occupancy is a crucial aspect of comprehending road scenes in the context of autonomous driving. It encompasses information pertaining to the drivable area and road obstacles, playing a vital role in ensuring safe autonomous navigation. An ideal occupancy representation assigns a pixel value of 1 to locations occupied by obstacles and a pixel value of 0 to locations representing free space. In numerous short-range applications (i.e., covers less than 30-m range [1]), such as vehicle back-off planning in parking lots, the focus is on the instantaneous free space surrounding the sensor [2], [3]. To address this need, our objective is to develop a precise and lightweight occupancy representation specifically tailored for the detection of instantaneous free space in short-range applications.

Free-space detection can be achieved through various sensor modalities, including ultrasound and cameras. While cameras excel in image-based free-space detection, they are limited by challenging weather and lighting conditions (such as night, rain, and snow) [2], and they may face accuracy limitations in depth estimation using a single device. On the other hand, ultrasound sensors have been utilized for short-range occupancy guidance, but their performance can be unstable, leading to high false alarm rates due to variations in air properties, such as temperature and wind [4]. Consequently, we opt for mmWave radars [5] for free-space detection. In recent years, radars have gained prominence in autonomous driving due to their affordability, range-Doppler-angle localization capabilities [2], [3], and resilience under adverse weather conditions [6].

Various approaches have been proposed to address occupancy and free-space detection. Traditional methods involve...
occupancy grid mapping, employing Bayesian filtering and handcrafted inverse sensor model (ISM) functions \cite{7,8,9,10,11,12,13,14,15}. This approach accumulates confidence in occupancy over time for a grid map area, offering accurate estimations for static objects with known ego-vehicle motion values. However, occupancy grid techniques have limitations in highly dynamic environments. These techniques often assume a static environment, making it challenging to handle moving objects. Grid cells represent static obstacles, requiring frequent classification and updates to reflect changes, which can be computationally expensive. Without timely updates, moving objects can cause false alarms and missed regions on the grid map. Besides, occupancy grid techniques require substantial memory consumption during formation and updating over time, making it less cost-effective for short-range applications \cite{16,17}. Beyond traditional methods, recent works treat occupancy detection as a semantic segmentation problem, employing deep learning (DL) models \cite{18} to predict the occupancy status for each pixel or cell in a map. DL-based methods, often using large neural networks, surpass traditional methods in performance but require substantial labeled training sets and intensive training and computing processes \cite{12,19,20,21,22,23,24}. In addition, concerns persist regarding the generation capabilities of DL-based methods due to limitations in training and testing sets.

This article introduces a novel occupancy representation tailored for short-range applications leveraging automotive radars, termed the “deformable radar polygon.” The radar polygon format defines a polygon-shaped region where the interior signifies free space, and the exterior represents undrivable areas for the ego vehicle. This format prioritizes accurate modeling of the free space near the ego vehicle, aligning with the requirements of short-range applications that emphasize proximity to surrounding obstacles. With point clouds extracted from raw radar data (an example pipeline is shown in Fig. 1), the radar polygon comprises vertices selected from these point cloud measurements. An intriguing aspect of the radar polygon is that each vertex includes Doppler velocity information from automotive radars, indicating the movement of the vertex (shrinkage or expansion) along the radial direction. This characteristic enables the prediction of the shape of future radar polygons, offering valuable insights for downstream applications, such as route planning and collision avoidance. Due to these predictive capabilities, we refer to the radar polygon as “deformable.” Beyond its predictive property, the radar polygon is computationally and storage-efficient, requiring the storage of a limited number of vertices and their confidence levels, in contrast to processing a large number of cells across the entire map (e.g., occupancy grid).

To construct the radar polygon, we present two approaches: the first is the “basic radar polygon” algorithm, which selects vertices independently for each time slot using heuristic methods, with signal-to-noise ratio (SNR)-based evidence employed for vertex fitness verification. The second approach is the “radar polygon update” algorithm, an advancement of the basic radar polygon method that takes into account information from previous frames. To enhance the system’s performance in terms of accuracy and smoothness, we introduce a probabilistic and tracking-based mechanism for updating the radar polygon over time. Evaluation results demonstrate that the proposed radar polygon methods exhibit low memory consumption, fast processing, and high smoothness, and achieve a comprehensive and accurate representation of free space. In addition, we introduce a collision detection algorithm to identify potential collisions outside the polygon region, leveraging the odd–even principle \cite{25}.

In summary, the key contributions are fourfold.

1) We proposed a polygon-based occupancy representation to model the free space around the ego vehicle for short-range applications. A basic radar polygon formation algorithm was proposed, which selects polygon vertices for each time slot, with SNR-based evidence employed for vertex fitness verification.

2) Beyond the basic radar polygon, we proposed the radar polygon update algorithm, which takes a probabilistic and tracking-based mechanism for updating the radar polygon over time to enhance the accuracy and smoothness.

3) We introduced the novel concept of the “deformable radar polygon,” which predicts the future shape of the free-space polygon based on the Doppler velocity of its vertices. In addition, we devised a method for collision detection on radar polygons by determining if points are situated inside the testing polygon.

4) To assess the performance of our proposed algorithms, we conducted comprehensive evaluations and analyses on both our self-collected dataset and the publicly available RadarScenes dataset. The results show significantly improved IoU-gt and IoU-smooth values compared with baseline methods.

II. RELATED WORK

The process of forming LiDAR-based occupancy grid maps \cite{12} typically involves utilizing a delta function for the ISM. This choice is influenced by the dense and accurate nature of LiDAR point data. Conversely, when creating a radar occupancy grid map, a more prevalent approach is to employ a Gaussian variant of the delta function for ISM. This adaptation accounts for the sparser and noisier characteristics of radar data. Notably, in works, such as \cite{13} and \cite{15}, the Gaussian ISM has been further enhanced by incorporating detection probability calculations, considering the plausibility of range, angle, and amplitude for each radar measurement. In the realm of ISM improvement, Prophet et al. \cite{7} introduce ego-motion velocity-dependent parameters to the algorithm.
This enhancement dynamically adjusts uncertainty ellipses based on ego velocity, assigning higher values when velocity is high and detections are scarce. To address the inherent uncertainty in radar measurements, Degerman et al. [9] propose converting SNR to the probability of detection using the Swerling-1 model within the ISM framework. Adapting to diverse application scenarios, the existing radar occupancy grid map methods have been extended to facilitate numerically efficient computations in various dimensions [9], [17]. Moving beyond traditional occupancy grid maps, Ruetz et al. [27] introduce a novel approach using 3-D mesh to represent free space. This method balances system precision and efficiency, demonstrating its versatility in handling complex scenarios. Furthermore, Hussain et al. [28] focus on predicting a drivable path for self-driving vehicles within the field of view (FOV) of a radar sensor using the dynamic Gaussian distributions for occupancy indication. In the realm of polygon-based representation, Meerpohl et al. [29] employ raycasting to delineate the boundaries of free space within a grid map, resulting in a closed polyline. Notably, their approach relies on the prior formation of a grid map, a prerequisite before executing the algorithm. Ziegler et al. [30] adopt a strategy where occupied areas on the left and right bounds of the driving corridor are defined as sets of convex hulls, particularly useful for path planning. In addition, Kuan et al. [31] present an algorithm that, given a set of polygonal obstacles in space, decomposes the free space into nonoverlapping geometric-shaped primitives suitable for path planning.

Current ISM-based algorithms heavily rely on radar detection and are predominantly handcrafted [20]. Inferring occupied space based on radar detections poses notable challenges due to data sparsity and environment-dependent noise, such as multipath reflections. Recent developments, moving away from handcrafted models, involve the utilization of DL-based methods in occupancy grid mapping, leveraging a data-driven approach [10], [12], [20], [21], [23]. DL-based ISM models consider the spatial coherence of all radar detection points to predict the scene context [21], [23]. The goal is to enhance system performance through an end-to-end model. Typically, deep ISM methods treat the occupancy status as a semantic segmentation problem, predicting the semantic class (i.e., occupied or free space) or the occupancy probability for each pixel in the occupancy grid.

Building upon the generated radar occupancy grid, various downstream applications have been explored. These applications include free-space detection through image analysis and dynamic b-spline contour tracking [32], detection of parallel-parked and cross-parked vehicles [33], and identification of available parking spaces [34], among others. Beyond occupancy mapping, there is a growing interest in vehicle perception using radars within the research community. This interest is fueled by the widespread availability of radars in off-the-shelf cars, their relatively low cost, their capability for 3-D/4-D estimation [2], [3], and their robustness under challenging weather conditions [6]. Novel applications, such as super-resolution imaging [3], vital sign monitoring [35], [36], and road object detection [5], have been proposed to leverage the full potential of automotive radars.

### III. Basic Radar Polygon Formation

#### A. Problem Formulation

In a set of radar detections denoted as $E$, each detection $e$ encompasses measurements in two dimensions $(x, y, z)$, Doppler velocity $v$, and SNR. Our goal is to select a subset of points, denoted as $E'$, from these detections to serve as vertices for constructing an occupancy polygon. The enclosed area of this polygon is considered free space. Our objective is to accurately determine the free space, minimize the number of selected points, and mitigate the interference from false alarms. To solve it, we propose the following heuristic solution.

#### B. Basic Radar Polygon Formation

In this section, we present a method for constructing a basic radar polygon in each time slot, independent of information from previous slots. This approach utilizes SNR-based evidence verification to identify suitable polygon vertices. Before initiating this process, we clean the data before feeding the radar point clouds to the algorithm by filtering out the points with very large heights (i.e., larger than $z_{\text{max}}$) and very small heights (i.e., smaller than $z_{\text{min}}$). Points with extreme heights often have less amplitude and could be false alarms resulting from the energy leakage of strong-amplitude targets. Besides, these points are outside the vehicle’s safe passing region, so disregarding them does not significantly affect the free-space detection. In our case, we have chosen $z_{\text{max}}$ to be 3 m and $z_{\text{min}}$ to be $-1.5$ m. Since our focus is on the occupancy status in a 2-D bird’s eye view (BEV), we project the processed points onto the 2-D $xy$ plane by discarding the $z$ dimension.

A polygon comprises multiple vertices, and our approach involves identifying a suitable vertex for each azimuth direction, as depicted in Fig. 2. To achieve this, we evenly sample the entire FOV using a fixed angle interval $\Delta \theta$. A vertex is then determined from radar points within each specific sampling sector (e.g., the light blue area in Fig. 2). The criteria for selecting the vertex in each sampling sector are defined as follows.

First, choose a candidate vertex from all radar points within the sampling sector. The points are sorted based on their distance $d$ defined as $d = (x^2 + y^2)^{1/2}$. The point with the shortest distance is selected as a candidate, and its validity as a valid vertex is then verified. If it does not meet the criteria,
assigns an evidence value to any location

where \( p \) extraction. The occupancy evidence map is established based

by summing these Gaussian evidences. Subsequently,

computed. The overall evidence, denoted as

evidence from all maps at the candidate vertex's location is

the candidate point, each point generates a Gaussian evidence

\( \epsilon \)
The uncertain emerging vertex will not be used for polygon formation in the current frame but will be stored and associated with future frames multiple times before being activated.

the process continues with the second-shortest-distance point, and so forth.

Verification involves examining the SNR values of neighboring points within a distance of \( \epsilon_1 \) from the candidate vertex. This step aims to prevent low-quality points, such as false alarms, from being selected as the polygon vertex. Each neighboring point is assigned a detection probability \( p_d \) based on its SNR [9] value, utilizing the Swerling-1 model

\[
p_d = P_{fa} \cdot \frac{1}{1 + \exp \left( \frac{-\left( p - \bar{p} \right)}{\sigma_p} \right)}
\]

where \( P_{fa} \) represents the false alarm rate used in point cloud extraction. The occupancy evidence map is established based on the detection probability \( p_d \) and the point location \( \mu = (x_0, y_0) \). It is defined by a Gaussian distribution, denoted as \( p_d \cdot N(\mu, \Sigma) \), with \( \Sigma \) as a fixed variance parameter [8]. To ensure that the evidence map adequately covers neighboring points, and considering that 99.7% of the energy of a normal distribution lies within the 3-sigma region, we set \( \Sigma = (\epsilon_1/3)^2 I \), where \( \epsilon_1 \) is the specified distance. The evidence map assigns an evidence value to any location \( (x, y) \) on the map according to the Gaussian distribution \( p_d \cdot N(\mu, \Sigma) \). Closeness to \( \mu \) results in a higher evidence value, approaching \( p_d \).

When multiple points are present in the \( \epsilon_1 \) neighborhood of the candidate point, each point generates a Gaussian evidence map in the \( xy \) plane, resulting in multiple overlapping evidence maps. As depicted in Fig. 2, the cumulative Gaussian evidence from all maps at the candidate vertex's location is computed. The overall evidence, denoted as \( p \), is obtained by summing these Gaussian evidences. Subsequently, \( p \) is normalized within the range \([0, 1]\) using the sigmoid function

\[
\tilde{p} = \text{sigmoid}(p) = \frac{1}{2} \left[ 1 + \frac{1}{2} \left[ 1 + \exp \left( \frac{-\left( p - \bar{p} \right)}{\sigma_p} \right) \right]^{-1} \right]
\]

where \( \tilde{p} \) and \( \sigma_p \) represent the shift and scaling factors. The sigmoid function is applied to normalize \( p \) to \( \tilde{p} \) within the range \([0, 1]\). Only when the calculated \( \tilde{p} \) exceeds the threshold \( P_{thr} \), do we confirm the candidate as the vertex. Otherwise, we select the next-shortest-distance point as the candidate and repeat the above verification process. It is important to note that the use of the \( \epsilon_1 \) distance threshold helps limit the number of points considered in the verification process, thereby alleviating computational load.

Second, following candidate verification for all points in the sampling sector, in the absence of any satisfying points, a virtual vertex is created. This virtual vertex is positioned at the intersection of the sector center and the boundary (e.g., point 2 in Fig. 2), characterized by zero Doppler velocity and zero SNR. When examining two adjacent nonvirtual vertices (e.g., points 1 and 3), if the cross-range length \( \Delta l \) (arc length) between them exceeds the predefined threshold \( l_{thr} \), a virtual vertex is not permitted in the middle sampling sector between them. This restriction is implemented to prevent the formation of small spikes associated with the virtual vertex.

Finally, all valid vertices, along with the origin point, are sequentially connected to form the basic radar polygon.

### IV. Probabilistic and Tracking-Based Algorithm for Radar Polygon Update

While the basic radar polygon formation algorithm yields an occupancy polygon around the ego vehicle for each timeslot, it lacks consideration of previous frames. To enhance confidence in the polygon boundary and address gaps in sensor coverage, we propose a probabilistic and tracking-based algorithm for updating the radar polygon over time. The underlying assumption is that vertices in each sampling sector can be tracked across timeslots when associated with the same target. Leveraging this, we employ the ISM to calculate a posterior probability or confidence for each vertex, indicating its validity. Unlike the traditional use of ISM in fixed-cell occupancy grid formation, our approach accommodates unfixed vertex locations, functioning effectively under the specified assumption. Mathematically, the problem is defined as follows: given the polygon vertices set \( E'_{t-1} \) and the confidence set \( L_{t-1} \) for the vertices at time \( t - 1 \), along with the noisy radar measurements \( E_t \) and the vehicle pose \( Z_t \) at time \( t \), the objective is to output the new polygon vertices \( E'_{t} \) and their associated confidence \( L_t \) for the current time \( t \).

To address this, we initially translate the locations of previous vertices \( E'_{t-1} \) to the latest coordinates at time \( t \) using

```plaintext
Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
```
the provided vehicle pose $Z_t$. Assuming $Z_t$ is represented as $(x_t, y_t, \phi_t)$—where $x$ and $y$ denote global position, and $\phi$ is the orientation—this transformation compensates for the vehicle’s motion $(x_t - x_{t-1}, y_t - y_{t-1})$ and orientation change $\phi_t - \phi_{t-1}$. Subsequently, we integrate the updated vertices $E_{t-1}'$ into the latest radar measurements $E_t$ for further processing, i.e., $E_t = E_{t-1} \cup E_t'$. The next step involves applying the first stage of the basic radar polygon formation algorithm (as detailed in Section III-B) to $E_t$ to select candidate vertices $e_t'$ for each sampling sector. However, the processing of these candidate vertices occurs under four distinct cases.

1) **Old Vertex**: This occurs when the candidate vertex $e_t'$ coincides with a point from the previous vertex set $E_{t-1}'$. If the candidate’s confidence $\ell_{t-1}$ is negative, indicating it is no longer valid, we discard it and seek a new candidate in the same sampling sector. For positive confidence, we add it to the set $E_t'$ but impose a confidence penalty of $-\ell_{pen}$ to account for the possibility of indefinite persistence.

2) **Trackable to an Old Vertex**: This situation arises when the candidate vertex $e_t'$ is not part of $E_{t-1}'$ but is close to the latest location of the previous vertex $e_{t-1}'$ within a threshold $\epsilon_2$. In this case, we assume $e_t'$ and $e_{t-1}'$ belong to the same target and are trackable. The update of confidence or posterior probability $\ell_t$ for $e_t'$ is based on Bayes’ theorem [15] and the confidence $\ell_{t-1}(e_{t-1}')$ of the previous vertex $e_{t-1}'$

$$\ell_t(e_t') = \ell_{t-1}(e_{t-1}') + \log \frac{\hat{p}(e_t' \mid E_t, Z_t)}{1 - \hat{p}(e_t' \mid E_t, Z_t)} - \ell_o \quad (3)$$

where $\hat{p}(e_t' \mid E_t, Z_t)$ is the normalized occupancy evidence of $e_t'$ from (2), and $E_t$ and $Z_t$ describe the current measurement or vehicle pose. The initial confidence $\ell_o$ is assumed to be 0, reflecting the lack of prior knowledge about the surrounding environment before the first measurement.

3) **Uncertain Emerging Vertex**: If $e_t'$ is neither an old vertex nor trackable to an old vertex, we consider it a potential emerging vertex corresponding to a new target in the scene. All emerging vertices are initially treated as uncertain and stored in a special set $U$ for further evaluation, aiming to reduce false alarms. Once a vertex in $U$ has been consistently associated with a new vertex more than once, it is transferred from $U$ to the polygon set $E_t'$. The association is assumed to occur when the interdistance is smaller than $\epsilon_2$.

4) **Missing**: This occurs when no candidate vertex is found in a sampling sector. In such cases, a virtual vertex with zero confidence is added to $E_t'$ following the second step of the basic radar polygon formation algorithm (as outlined in Section III-B).

The diagram and pseudocodes for the above are presented in Fig. 3 and Algorithm 1, respectively.

### V. DEFORMABLE PROPERITY OF RADAR POLYGON

#### A. Deformable Polygon Prediction

The vertices comprising the radar polygon possess Doppler velocities, describing their instantaneous movement along the radial direction (i.e., the direction from the vertex to the radar). Under the assumption that the radial velocity of a vertex is constant within a short period and the ego vehicle has no significant rotation, we can estimate the vertex’s future location by calculating and adding its radial movement using the current Doppler velocity [see Fig. 4(a)]. Specifically, for a vertex with location $(x_0, y_0)$ and Doppler velocity $\mathbf{v} = (v_x, v_y)$, its radial movement $(\Delta x, \Delta y)$ within a duration $\Delta t$ is given by

$$\Delta x, \Delta y = (v_x, v_y) \times \Delta t \quad (4)$$

where $(v_x, v_y)$ is the projection of $\mathbf{v}$ along the $x$- and $y$-directions as follows:

$$v_x, v_y = \frac{(x_0 - x_v, y_0 - y_v)}{\sqrt{(x_0 - x_v)^2 + (y_0 - y_v)^2}} \times \mathbf{v} \quad (5)$$
The radar polygon update algorithm in Section IV relies on accurate vehicle pose information for projecting radar measurements from different time slots onto the same coordinate. Specifically, (3) illustrates the use of vehicle pose $Z_t = (x_i, y_i, \phi_i)$ in projecting the old radar polygon $E_{t-1}^r$ to the global coordinate of time $t$ to obtain the updated polygon. We observe that the deformable polygon prediction property—where the future shape of a radar polygon can be roughly predicted with Doppler velocity—has the potential to replace the required pose information under certain critical cases. This substitution could be particularly useful when localization sensors (e.g., inertial measurement unit and global positioning system) are unavailable, and there is no significant lateral movement. The cross-time compensation for vehicle motion $(x_i - x_{i-1}, y_i - y_{i-1})$ and orientation change $\phi_i - \phi_{i-1}$ in Section IV can be approximately replaced by moving the vertex by $v_r - \Delta t$, where $v_r - 1$ is the corresponding Doppler velocity of the vertex and $\Delta t$ is the time slot duration. It is important to note that Doppler velocity measures the relative moving speed along the radial direction. Even for stationary objects, nonzero Doppler velocities reflect relative movement when the car is in motion.

**VI. COLLISION DETECTION FOR RADAR POLYGON**

In this section, we address collision detection for the radar polygon-based occupancy representation. We assume that the polygon has $n$ vertices $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ sorted in an ascending order of azimuth. The goal is to determine whether a point $(a, b)$, which needs to be checked, collides with the region outside the polygon. This collision detection is equivalent to determining whether the point $(a, b)$ in the plane lies inside or outside the polygon.

With that being said, we adopted the even–odd rule algorithm [25] for solving the collision detection problem. To elaborate, we draw a ray to the right of point $(a, b)$ and extend it to infinity. The number of times this ray intersects with the sides of the polygon (assuming the polygon boundary is not strictly horizontal) is then counted. As shown in Fig. 4(b), if the ray intersects the sides of an even number of times, $(a, b)$ is considered outside the polygon; if it is an odd number of times, $(a, b)$ is considered inside the polygon. A distinctive case arises when the intersection between the ray and the sides of the polygon corresponds to a vertex of the polygon. In this instance, the intersection is taken into account only if the other vertex of the side lies below the ray.

The pseudocode of the collision detection algorithm for radar polygon is presented in Algorithm 2 below.

![Algorithm 2 Collision Detection Algorithm for Radar Polygon](image)

It is worth noting that by utilizing the deformable polygon property, we can predict future collision detection by forecasting the new polygon shape. According to (4), the future polygon shape can be roughly predicted with the input of the current polygon shape and Doppler velocity components along $x$ and $y$. Subsequently, Algorithm 2 can be applied to the predicted polygon shape for potential collision detection in the future.

**VII. IMPLEMENTATION**

**A. Testbed Setup**

Our data collection testbed comprises four radars—two mounted on the front side and two on the rear side of the car. The radars are strategically positioned with a certain shift on boresight to collectively enhance the overall FOV, illustrated by the light green region in Fig. 5 (left). Indeed, the radar setup depicted in Fig. 5 does not cover the entire range, particularly the left-/right-hand sides of the ego vehicle. The two proposed polygon formation algorithms have different
ways of dealing with the side area. The basic radar polygon formation algorithm leaves the sides of the vehicle blank as those areas fall outside the FOV of the radars. In contrast, the probabilistic and tracking-based radar polygon update algorithm utilizes information from previous frames, allowing for the transformation of polygon vertices to the next frame. This mechanism enables the polygon to fill or observe the sides of the ego vehicle across time as the vehicle moves.

The radar configuration includes a maximum detectable range of 30 m, a range resolution of 4 cm, an azimuth FOV of ±65°, an azimuth resolution of 5°, an elevation FOV of ±60°, an elevation resolution of 5°, a Doppler resolution of 0.22 m/s, a maximum unambiguous Doppler of 7.1 m/s, and a frame rate of 10 frames/s. The experimental car is also equipped with BEV cameras to facilitate synchronized data collection and ground-truth labeling.

B. Experimental Setup

We conducted experiments in a parking lot, simulating a classic car parking scenario that includes multiple parked vehicles and various static background elements, such as walls, fences, and gates. The ego vehicle was moving at the speeds of less than 15 mph while parking in and out of spots and driving around the parking lot. The vehicle’s trajectory was highlighted with yellow curves in Fig. 5 (right). The dynamic elements in the environment included pedestrians and other moving vehicles passing by as the ego vehicle was parking. These conditions were designed to mimic real-world parking scenarios, capturing both static and dynamic elements to test the robustness of our method. During the experiment, point clouds from four radars, synchronized with camera images, were collected, each with its own timestamp. Vehicle pose and trajectory, crucial for ISM, were obtained through radar odometry techniques [3], [37]. Ground truth for occupancy status was generated by human labeling on BEV camera images, projected to radar coordinates via BEV-radar cross calibration and transformation. Hyperparameters for radar polygon formation and ISM were selected through grid search: \( \epsilon_1 = 1, \epsilon_2 = 1, \ell_{thr} = 7.5 \text{ m}, \bar{p} = 12.1, \sigma_p = 7.132, p_{thr} = 0.62, \) and \( \ell_{pen} = 0.5 \). Additional system parameters included a sampling angle of \( \Delta \theta = 2^\circ \) and a constant false alarm rate of \( p_{fa} = 10^{-3} \). For computational efficiency, we formed a front-view polygon for the two front radars and a rear-view polygon for the two rear radars, calculating the union of these polygons as the final free space. All experiments were conducted on a PC equipped with a 2.9-GHz 6-Core Intel i9 processor and 32 GB of RAM for fair algorithmic comparisons.

VIII. EVALUATION AND ANALYSIS

A. Evaluation Results on Self-Collecte d Dataset

1) Baselines: In this section, we evaluated the two proposed polygon-based algorithms on the self-collected dataset (described in Section VII-B) by measuring the accuracy and smoothness of the generated radar polygon. For simplicity of representation, we refer to the two proposed radar polygon approaches—the basic radar polygon formation algorithm, and the probabilistic and tracking-based radar polygon update algorithm—as the “basic radar polygon” and “radar polygon update,” respectively.

We chose two state-of-the-art polygon-based baselines—Meerpohl et al. [29] and Ziegler et al. [30]—for comparison with the proposed algorithms. Specifically, Meerpohl et al.’s [29] method requires forming the grid map first and then raycasting on it to create the free-space polygon. In our implementation, we form a grid map of \( 60 \times 60 \text{ m} \) with 0.3-m resolution using Werber et al.’s [13] algorithm every 30 frames. The raycasting center is chosen as the vehicle’s center, the maximum distance for the raycasting line is 30 m, the radius of the half-circle is 2 m, the total number of sampling points is 180, and the threshold for deciding an occupied pixel is 20 dB. The other baseline, Ziegler et al.’s [30] algorithm, forms two convex hulls to envelop the radar point clouds on the left- and right-hand sides of the vehicle separately. Since the inside of the convex hulls represents the occupancy area, we then customize the generation of the free-space polygon by finding the contour of the area outside the occupancy convex hulls.

2) Metrics: For numerical evaluation, we employed intersection over union (IoU) as the metric. IoU is a metric used for evaluating the similarity between two shapes, such as rectangles or polygons [38]. The IoU between two polygons, e.g., A and B, is calculated by dividing the area of their intersection by the area of their union. Based on this, we propose two metrics for evaluation: IoU-gt and IoU-smooth. IoU-gt measures the IoU between the ground-truth polygon and the radar polygon output from our proposed algorithms. IoU-smooth measures the IoU between the radar polygon outputs for two consecutive frames. For both IoU-gt and IoU-smooth, the values are averaged over all frames in the testing data to obtain the final results.

In addition to the IoU-gt and IoU-smooth metrics, we propose a new metric called confidence over time (CoT) to evaluate the confidence of polygon vertices. To calculate CoT, we sum the confidence values of all vertices that match the ground-truth polygon and then divide this sum by the number of vertices. This process is repeated over time to obtain the average CoT. A vertex is considered to match the ground-truth polygon if the distance between it and a corresponding ground-truth vertex is smaller than a specified threshold.

3) Results: We showcase a snapshot of recorded data, point cloud visualization, free-space ground truth, and the output of the two proposed polygon algorithms and two baselines in Fig. 6. From it, we can observe that the basic radar polygon
algorithm accurately delineates the contours of parked cars and the empty parking space between them. However, the left- and right-hand sides of the ego vehicle are left blank, because these areas are outside the radars’ FOV. Beyond that, the radar polygon update algorithm not only depicts the contour of the surrounding environment accurately but also fills out-of-view gaps and minimizes missing detections by utilizing temporal information. The counterpart baseline of Meerpohl et al. [29] also takes advantage of temporal information to form the free-space polygon but exhibits more false alarms of vertices due to the accumulation of confidence. Ziegler et al.’s [30] baseline, however, tends to detect smaller free space, because the occupancy convex hulls usually occupy a lot of space that indeed belongs to free space.

The numerical evaluation of the four algorithms in terms of the IoU-gt and IoU-smooth metrics is shown in Table I. The results indicate that the basic radar polygon method achieves a 66.21% IoU-gt and a perfect 86.45% IoU-smooth. The adoption of the radar polygon update significantly improves IoU-gt and IoU-smooth by 7% and 3%, respectively. However, the two baselines perform much worse than the two proposed polygon algorithms. Meerpohl et al.’s [29] baseline has a lower IoU-gt and IoU-smooth, because it takes time to accumulate confidence. Thus, for early frames, the confidence of grid pixels does not reach the threshold to form radar vertices. After accumulation over a long time, the false alarm vertices also hurt its performance. Ziegler et al.’s [30] baseline performs poorly, because the generated free-space polygon is much smaller than the real one. In terms of the CoT metric, the radar polygon update algorithm and the algorithm by Meerpohl et al. [29] show higher vertex confidence compared with the other two algorithms, as they leverage data from previous frames. Meerpohl et al.’s [29] algorithm has a CoT value that is 0.04 higher, likely due to more false alarms resulting in more vertices matching the ground truth.

| Method                  | IoU-gt | IoU-smooth | CoT |
|-------------------------|--------|------------|-----|
| basic radar polygon     | 66.21% | 86.45%     | 0.6420 |
| radar polygon update    | 73.12% | 89.68%     | 0.7774 |
| Meerpohl et al. [29]    | 40.98% | 51.11%     | 0.8153 |
| Ziegler et al. [30]     | 37.94% | 39.50%     | 0.4832 |

Table I: Evaluation Results on the Self-Collected Dataset

B. Polygon Prediction Evaluation Results

To illustrate Doppler velocity on each vertex within a radar polygon, additional data were collected for two scenarios: pedestrians passing by and vehicles passing by. Fig. 7 showcases the results formed by rear radars, with red arrows indicating measured Doppler velocities for polygon vertices. The visualizations demonstrate the polygon’s effective representation of the edges of passing pedestrians and moving vehicles, with the plotted Doppler velocities conveying the speed and direction of each vertex.

To quantitatively assess the performance of the polygon prediction discussed in Section V, we conducted a one-to-one comparison in Fig. 8 between the current-frame radar polygon and the predicted polygon from the last frame. We calculated the IoU between them for three scenarios: car backing-off, pedestrian, and vehicle passing-by experiments. In Fig. 8, current-frame radar polygons are represented by a light-blue region, while predicted polygons are depicted with a dotted line. The high IoU values above 0.9 indicate excellent prediction performance, validating the accuracy of our deformable radar polygon approach.

It is important to note that the lack of a tangential component in Doppler speed may lead to inaccuracies for vertices with significant tangential speed, whether from self-movement or relative shifts.

C. Running Time and Memory Usage Analysis

In this section, we initially assessed the execution time of the proposed radar polygon algorithms and the baselines in a PC environment. Subsequently, we conducted an analysis and measurement of the memory usage required for storing polygons and essential parameters. The results are presented in Table II.
The sampling angle, denoted as $\Delta \theta$, significantly influences system performance and memory usage. In the experiment evaluation of the basic radar polygon and radar polygon update algorithms on the self-collected dataset, varying $\Delta \theta$ values were explored. Specifically, the sampling intervals for Fig. 9 were set at $0.5^\circ$, $1^\circ$, $2^\circ$, $3^\circ$, $4^\circ$, $5^\circ$, $6^\circ$, $7^\circ$, $8^\circ$, $9^\circ$, and $10^\circ$. As depicted in Fig. 9, a smaller $\Delta \theta$ yields more complete and accurate radar polygons, leading to improved IoU-gt and IoU-smooth for both methods. While accuracy and smoothness are not dramatically compromised with $\Delta \theta$ ranging from $0.5^\circ$ to $10^\circ$, diminishing the sampling angle introduces computational complexity and memory challenges due to an increased number of vertices for calculation, storage, and updating. Consequently, finding an optimal trade-off for the $\Delta \theta$ value becomes crucial to align with specific application requirements.

### D. Impact of Sampling Angle $\Delta \theta$

The sampling angle, denoted as $\Delta \theta$, significantly influences system performance and memory usage. In the experiment...
Fig. 10. Examples in RadarScenes dataset. (a) Result of basic radar polygon (free space is in blue). (b) Result of radar polygon update algorithms. (c) Result of Meerpol et al. [29]. (d) Result of Ziegler et al. [30]. Among all figures, the green points are input radar point clouds, and the red dots are the radar polygon vertices.

### TABLE III

| Method                      | IoU-gt | IoU-smooth | CoT   |
|-----------------------------|--------|------------|-------|
| basic radar polygon         | 23.66% | 68.91%     | 0.4402|
| radar polygon update        | 74.44% | 86.38%     | 0.8507|
| Meerpol et al. [29]         | 22.91% | 70.55%     | 0.7871|
| Ziegler et al. [30]         | 18.98% | 73.23%     | 0.2633|

Fig. 11. Evaluation results of polygon prediction on the RadarScenes dataset for two scenes (a) and (b). The light blue region is the measured radar polygon, while the dotted line forms the predicted radar polygon from the last frame. The IoU values between the measured and predicted polygon for scene (a) and (b) are 84.47% and 79.63%, respectively.

The polygon update algorithm depicts the surrounding area most accurately. The evaluation results of the polygon prediction are depicted in Fig. 11. Even in the complex city-road driving scenario, the deformable polygon demonstrates robust short-term prediction performance, exhibiting a high overlap with the current-frame radar polygon. The IoU values for two testing cases are 84.47% and 79.63%, respectively. However, a notable issue is observed, as the prediction does not effectively handle new points or objects that appear, resulting in a degradation of IoU performance due to the lack of information from previous frames.

F. Collision Detection for Route Planning

The collision detection algorithm, as discussed in Section VI, was employed to process a vehicle route planning scenario, assessing whether planned trajectories fall within the radar polygon. We adopted a radar polygon representing the free space for driving generated by our algorithms on the self-collected dataset, depicted in blue in Fig. 12. Subsequently, three trajectories were randomly generated, and each trajectory was sampled to acquire a set of locations. These locations served as input for the collision detection algorithm, determining whether each point was inside or outside the polygon. The collision detection results for the planned trajectories are depicted in Fig. 12 as red or green dotted lines, with red indicating collision in this path and green representing no collision. After a thorough evaluation, we established a 100% collision detection accuracy. This remarkable result underscores the effectiveness of the collision detection algorithm in ensuring the safety of planned vehicle trajectories.

IX. Conclusion

In this article, we introduced a radar polygon formation algorithm that utilizes radar point cloud data as input. In addition, we developed a collision detection method tailored to this new polygon representation. Both the radar polygon and the collision detection model were validated through extensive experiments using both self-collected and public datasets. Our results show that the IoU-gt and IoU-smooth metrics increased by 35% and around 40%, respectively, compared with baseline methods on the self-collected dataset, and by 52% and 13%, respectively, on the RadarScenes dataset, demonstrating superior accuracy and robustness. Furthermore, the deformable polygon concept was validated by the high IoU correlation between the predicted and generated radar polygons. Memory usage analysis also indicated significantly reduced memory requirements compared with baselines. For future work, we plan to explore approaches to adapt the radar polygon for long-range scenarios and further enhance its overall robustness.

References

[1] AVNET. (2024). Automotive Radar Systems: The Design Engineer’s Guide. AVNET. [Online]. Available: https://my.avnet.com/abacus/solutions/markets/automotive-and-transportation/automotive-comfort-infotainment-and-safety/automotive-radar/

[2] X. Gao, G. Xing, S. Roy, and H. Liu, “Experiments with mmWave automotive radar test-bed,” in Proc. 53rd Asilomar Conf. Signals, Syst., Comput., 2019, pp. 1–6.
[3] X. Gao, S. Roy, and G. Xing, “MIMO-SAR: A hierarchical high-resolution imaging algorithm for mmWave FMCW radar in autonomous driving,” IEEE Trans. Veh. Technol., vol. 70, no. 8, pp. 7322–7334, Aug. 2021.

[4] M. Toa and A. Whitehead, Ultrasonic Sensing Basics. Dallas, TX, USA: Texas Instrument, 2019. [Online]. Available: https://www.ti.com/lit/an/sla907c/sla907c.pdf

[5] X. Gao, G. Xing, S. Roy, and H. Liu, “RAMP-CNN: A novel neural network for enhanced automotive radar object recognition,” IEEE Sensors J., vol. 21, no. 4, pp. 5119–5132, Feb. 2021.

[6] X. Gao, S. Roy, G. Xing, and S. Jin, “Perception through 2D-MIMO FMCW automotive radar under adverse weather,” in Proc. IEEE Int. Conf. Auto. Syst. (ICAS), Montreal, QC, Canada, Aug. 2021, pp. 1–5.

[7] R. Prophét, H. Stark, M. Hoffmann, C. Sturm, and M. Vossiek, “Adaptions for a passive radar based occupancy gridmaps,” in IEEE MTT-S Int. Microw. Symp. Dig., Apr. 2018, pp. 1–4.

[8] P. Berthold, M. Michaelis, T. Luettel, D. Meissner, and H.-J. Wuensch, “Deriving spatial occupancy evidence from radar detection data,” in Proc. IEEE Intell. Vehicles Symp. (IV), Oct. 2020, pp. 831–836.

[9] J. Degerman, T. Pernstål, and K. Ailenjunga, “3D occupancy grid mapping using statistical radar models,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2016, pp. 902–908.

[10] F. E. Nowruzi et al., “Deep open space segmentation using automotive radar,” in IEEE MTT-S Int. Microw. Symp. Dig., Nov. 2020, pp. 1–4.

[11] N. Engelhardt, R. Pérez, and Q. Rao, “Occupancy grids generation using deep radar network for autonomous driving,” in Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), Oct. 2019, pp. 2866–2871.

[12] L. Sless, B. E. Shlomo, G. Cohen, and S. Oron, “Road scene understanding by occupancy grid learning from sparse radar clusters using semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW), Oct. 2019, pp. 867–875.

[13] K. Werber et al., “Automotive radar gridmap representations,” in IEEE MTT-S Int. Microw. Symp. Dig., Apr. 2015, pp. 1–4.

[14] C. Diehl, E. Feicho, A. Schwambach, T. Dammeyer, E. Mares, and T. Bertram, “Radar-based dynamic occupancy grid mapping and object detection,” in Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC), 2020, pp. 1–6.

[15] M. Li, Z. Feng, M. Stolz, M. Kunert, R. Henze, and F. Kicukay, “High resolution radar-based occupancy grid mapping and free space detection,” in Proc. 4th Int. Conf. Vehicle Technol. Intell. Transp. Syst., 2018, pp. 70–81.

[16] F. Homm, N. Kaempchen, J. Ota, and D. Burschka, “Efficient occupancy grid computation on the GPU with LiDAR and radar for road boundary detection,” in Proc. IEEE Intell. Vehicles Symp., Jun. 2010, pp. 1006–1013.

[17] M. Slusky and D. Dobkin, “Fast implementation of volumetric occupancy grids,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2019, pp. 750–755.

[18] X. Gao, H. Liu, S. Roy, G. Xing, A. Alansari, and Y. Luo, “Learning to detect open carry and concealed object with 77 GHz radar,” IEEE J. Sel. Topics Signal Process., vol. 16, no. 4, pp. 791–803, Jun. 2022.

[19] Z. Xie, H. Wang, S. Han, E. Schoenfeld, and F. Ye, “DeepV: A deep learning approach for RF-based vital signs sensing,” in Proc. 13th ACM Int. Conf. Bioinfomat. Comput. Biol. Health Informat., 2022, pp. 1–5.

[20] Y. Jin, M. Hoffmann, A. Deligiannis, J.-C. Fuentes-Michel, and M. Vossiek, “Semantic segmentation-based occupancy grid map learning with automotive radar raw data,” IEEE Trans. Intell. Vehicles, vol. 9, no. 1, pp. 216–230, Jan. 2024.

[21] D. Bauer, L. Kuhnert, and L. Eckstein, “Deep inverse sensor models as priors for evidential occupancy mapping,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2020, pp. 6032–6037.

[22] Z. Xie, B. Zhou, X. Cheng, E. Schoenfeld, and F. Ye, “VitalHub: Robust, non-touch multi-user vital signs monitoring using deep camera-aided UWB,” in Proc. IEEE 9th Int. Conf. Healthcare Informat. (ICHI), Victoria, BC, Canada, Aug. 2021, pp. 320–329.

[23] R. Weston, S. Cen, P. Newman, and I. Posner, “Probably unknown: Deep inverse sensor modelling radar,” in Proc. Int. Conf. Robot. Autom. (ICRA), May 2019, pp. 5446–5452.

[24] Z. Xie, B. Zhou, E. Schoenfeld, and F. Ye, “Passive and context-aware in-home vital signs monitoring using co-located UWB-depth sensor fusion,” ACM Trans. Comput. Healthcare, vol. 3, no. 4, pp. 1–31, Nov. 2022.

[25] M. Shimrat, “Algorithm 112: Position of point relative to polygon,” Commun. ACM, vol. 7, no. 8, p. 434, Aug. 1962, doi: 10.1145/368637.368653.

[26] X. Gao, S. Roy, and L. Zhang, “Static background removal in vehicular radar: Filtering in Azimuth-Elevation-Doppler domain,” 2023, arXiv:2307.01444.

[27] F. Ruetz et al., “OVPC mesh: 3D free-space representation for local ground vehicle navigation,” in Proc. Int. Conf. Robot. Autom. (ICRA), May 2019, pp. 8646–8654.

[28] M. I. Hussen, S. Azam, M. A. Rafique, A. M. Sheri, and M. Ieon, “Drivable region estimation for self-driving vehicles using radar,” IEEE Trans. Veh. Technol., vol. 71, no. 6, pp. 5971–5982, Jun. 2022.

[29] C. Meerpoohl, M. Rick, and C. Büskens, “Free-space polygon creation based on occupancy grid maps for trajectory optimization methods,” IFAC-PapersOnLine, vol. 52, no. 8, pp. 368–374, 2019.

[30] J. Ziegler, P. Bender, T. Dang, and C. Stiller, “Trajectory planning for bertha—A local, continuous method,” in Proc. IEEE Intell. Vehicles Symp., Jul. 2014, pp. 450–457.

[31] D. Kuan, J. Jamaiska, and R. Brooks, “Natural decomposition of free space for path planning,” in Proc. IEEE Int. Conf. Robot. Autom., Mar. 1985, pp. 168–173.

[32] M. Schreiter and V. Willert, “Robust free space detection in occupancy grid maps by methods of image analysis and dynamic b-spline contour tracking,” in Proc. 15th Int. IEEE Conf. Intell. Transp. Syst., Sep. 2012, pp. 514–521.

[33] R. Dubé, M. Hahn, M. Schütz, J. Dickmann, and D. Gingras, “Detection of parked vehicles from a radar based occupancy grid,” in IEEE Intell. Vehicles Symp. Proc., Jun. 2014, pp. 1415–1420.

[34] R. Prophét, M. Hoffmann, M. Vossiek, G. Li, and C. Sturm, “Parking space detection from a radar based target list,” in IEEE MTT-S Int. Microw. Symp. Dig., Mar. 2017, pp. 91–94.

[35] Z. Xie, B. Zhou, and F. Ye, “Signal quality detection towards practical non-touch vital sign monitoring,” in Proc. 12th ACM Conf. Bioinfomat. Comput. Biol. Health Informat., Aug. 2022, pp. 1–9.

[36] Z. Xie, B. Zhou, X. Cheng, E. Schoenfeld, and F. Ye, “Fusing UWB and depth sensors for passive and context-aware vital signs monitoring,” in Proc. IEEE/ACM Conf. Connected Health Appl., Syst. Eng. Technol. (CHASE), Dec. 2021, pp. 119–120.

[37] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, “Instantaneous ego-motion estimation using multiple Doppler radars,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2014, pp. 1592–1597.

[38] D. Gloudeans, X. Lu, S. Xia, and D. B. Work, “Polygon Intersection-over-Union loss for viewpoint-agnostic monocular 3D vehicle detection,” 2023, arXiv:2309.07104.

[39] O. Schumann et al., “RadarScenes: A real-world radar point cloud data set for automotive applications,” in Proc. IEEE 24th Int. Conf. Inf. Fusion (FUSION), Nov. 2021, pp. 1–8, doi: 10.23919/FUSION49465.2021.9627037.